

Electronica Innovativa In

€ 5,00

Mensile di elettronica applicata, attualità scientifica, novità tecnologiche.

125

**Datalogger USB
a 12 canali**

**Dimmer
bicanale
radiocomandato**

**Radiomodem
434/868 MHz**

- Controllo luci per automodellismo
- Bolero, localizzatore portatile programmabile
- Segnapunti, firmware e contenitore
- Progettare lo stadio di alimentazione
- Buzzer telefonico

+++ ESCLUSIVO +++
CORSO EM1000
Programmazione
modulo Tlbbio

Controllo luci per automodellismo

dell' *Ing. Mirco Segatello*

Siete appassionati di automodellismo e sempre alla ricerca di nuovi gadget per il vostro modello? Ecco il prodotto giusto: inserito tra i servocomandi e la ricevente, è in grado di attivare automaticamente gli indicatori di direzione e gli stop, riconoscendo gli ordini inviati tramite il radiocomando. E' anche in grado di accendere automaticamente i fari anabbaglianti e le luci di posizione posteriori quando diventa buio.

Gli appassionati di modellismo che si cimentano a pilotare divertenti modellini di automobili sono tantissimi: qualcuno preferisce i modelli da corsa, qualcun altro, invece, preferisce i fuori strada; da qualche tempo, poi, sono anche apparsi i primi modelli RC di macchine operatrici e di mezzi di trasporto, dalle scavatrici agli autocarri. Chi non ha mai provato a pilotare un modello RC, magari spinto da un motore a scoppio, difficilmente può capire quanto sia entusiasmante "aprire velocemente il gas dando contemporaneamente un colpo di sterzo" per innescare uno spettacolare e rumoroso testacoda. Tale

manovra, accompagnata dal rumore del motore e dal fumo dello scarico, è già molto eccitante, ma potrebbe diventarlo ancor di più se le luci del modello (qualora esso ne sia fornito) si accendessero automaticamente secondo una precisa logica. Un interessante accessorio per un modello da strada, quindi, potrebbe essere rappresentato da un controllo luci in grado di attivare automaticamente gli stop, gli abbaglianti e le frecce, il tutto perfettamente sincronizzato con i comandi impartiti via radio al modellino. La logica di funzionamento di un circuito simile è molto semplice: l'intervento del controllo di frenata deve attiva- ➤

re la luce di stop posteriore, mentre il comando di sterzo deve attivare gli indicatori di direzione laterali. Gli anabbaglianti, infine, devono essere comandati in automatico non appena viene rilevato un calo della luminosità esterna, esattamente come avviene in molte automobili da strada.

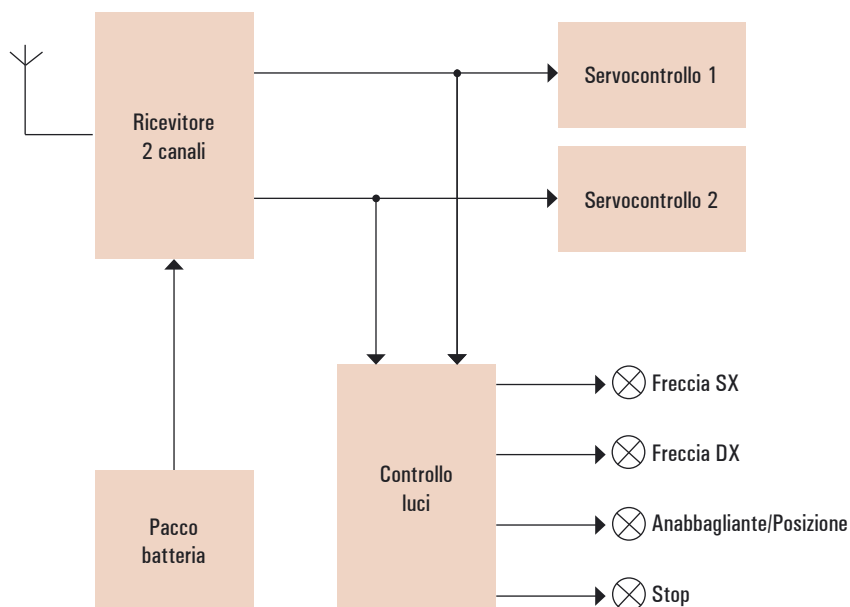
Ora, se volete dotare il vostro modello di un gadget che lo renda veramente particolare, continuate a leggere: nelle prossime pagine vi spiegheremo il funzionamento del nostro *servoluci*, nome che abbiamo coniato noi per assonanza con quello del servocontrollo.

Principio di funzionamento

Questo circuito è stato progettato per poter essere inserito facilmente in qualsiasi modello RC, intercettando le piattine trifilari con cui i servocontrolli vengono collegati all'unità ricevente. In pratica, la maggior parte delle unità riceventi bicanale dispone di due connettori pin strip (uno per canale) a tre pin e passo 2,54 mm, che portano ai servocontrolli le linee di alimentazione, massa e segnale di controllo. Come visibile in Fig. 1, il nostro circuito va collegato in parallelo tra la ricevente e i due servocontrolli tramite quattro connettori pin strip uguali a quelli della ricevente. Da essi, il dispositivo preleva alimentazione, massa e segnale di controllo di entrambi i canali, senza però apportare alcun cambiamento al segnale di controllo: notate infatti che le linee intercettate dal circuito vengono portate parallelamente ai connettori dei servocontrolli. Ciò significa che, dal punto di vista dei servo e della ricevente, il circuito è come se non ci fosse; esso, però, riceve i segnali di controllo di entrambi i servo, segnali che utilizza per accendere le luci secondo uno specifico programma: relativamente a quest'ultimo, nel riquadro dedicato al Listato 1 abbiamo riportato il codice sorgente completo, ben commentato in ogni sua istruzione.

Fig. 1 - Schema a blocchi.

Il circuito va collegato in parallelo alle linee che collegano la ricevente ai servocontrolli; in pratica preleva sia la tensione di alimentazione che gli impulsi di controllo; la durata di questi ultimi viene misurata ed utilizzata per comandare l'accensione delle luci.

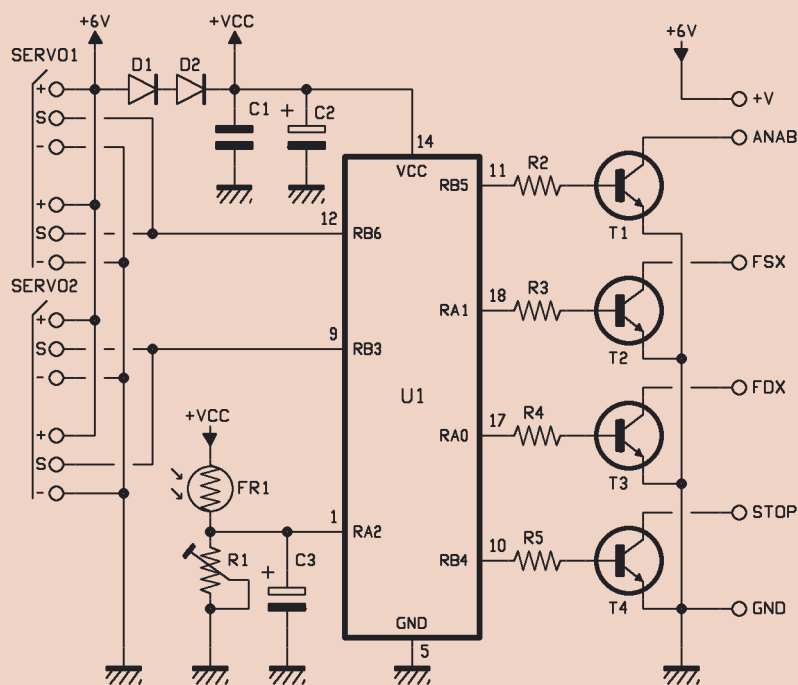


Schema Elettrico

Il circuito, molto semplice, è basato su un microcontrollore modello PIC16F88, privo di quarzo esterno dato che utilizza l'oscillatore interno. L'alimentazione viene prelevata direttamente dalla batteria che alimenta la radio, attraverso le linee di alimentazione dei servocontrolli. I due diodi in serie all'alimentazione, D1 e D2, creano una piccola caduta di tensione (1,4 V circa): l'alimentazione principale dei servo e del ricevitore, infatti, il più delle volte è costituita da quattro pile a secco da 1,5 volt, che forniscono complessivamente una tensione di 6 volt. Dato che il PIC funziona con 5,5 V massimi, la caduta di tensione su questi due diodi impedisce al micro di bruciarsi. La fotoresistenza FR1, collegata al pin RA2 in partitore col trimmer R1, viene utilizzata dal microcontrollore per rilevare l'intensità della luce esterna, indicazione che verrà poi utilizzata per attivare le luci anabbaglianti. La linea RA2 non viene usata come ingresso analogico ma come digitale, lasciando che sia la logica interna a determinare la

soglia di commutazione: visto il tipo di controllo, infatti, non è necessario conoscere con precisione il livello di luminosità ambiente, mentre è sufficiente sapere quando questa supera una determinata soglia. Il livello di commutazione può comunque essere regolato tramite il trimmer R1. Nel caso di utilizzo all'esterno in condizioni di luce diretta molto forte, è possibile coprire il sensore con del materiale semitrasparente in modo da creare una piccola schermatura ai raggi solari e poter accendere le luci anche in condizioni di media luminosità. Come spiegavamo nel paragrafo precedente, al circuito devono arrivare i segnali provenienti dalla trasmittente e destinati ai servocontrolli: i quattro connettori pin strip (due identificati come SERVO1 e gli altri due come SERVO2) vengono utilizzati per interporre il circuito tra la ricevente ed i servo. Il *servoluci* è predisposto per il controllo di quattro uscite: la luce di stop, le luci anabbaglianti e le due frecce, destra e sinistra. Le uscite sono tutte open collector e consentono di collegare

SCHEMA ELETTRICO



que, di rimanere al disotto della corrente massima ammessa per il tipo di led utilizzato che nella maggior parte dei casi è di 25÷40 mA. Inoltre, maggiore è il consumo di corrente dei led, minore è la durata del pacco batteria. Il vantaggio derivante dall'utilizzo dei led al posto delle lampadine è proprio il minor consumo in corrente, a parità di luce emessa. In commercio esistono led superluminosi di svariati colori, che permettono di personalizzare il proprio modello in modo da ottenere un effetto luci molto più accattivante.

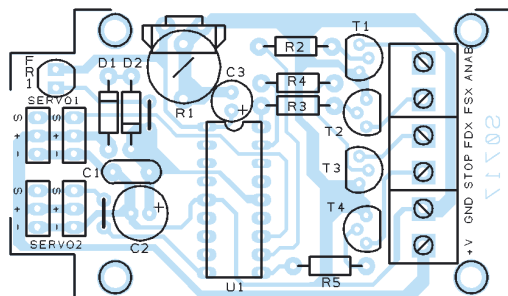
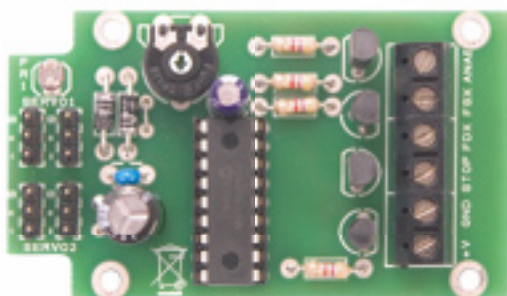
Realizzazione pratica

Come potete notare nel disegno del piano di montaggio, il circuito stampato è veramente semplice: una volta realizzata la basetta con la tecnica della fotoincisione (il master in formato 1:1 è scaricabile gratuitamente dal nostro sito www.elettronica.in.it) e forate le piazzole, potete iniziare il montaggio con i pochi componenti a basso profilo (resistenze, diodi, pin strip a passo 2,54 mm, eccetera) facendo attenzione al verso di quelli polarizzati. Potete poi montare lo zoccolo da 9+9 pin DIL per il microcontrollore, i transistor, i condensatori ed il trimmer; terminate con la ➤

carichi fino ad un massimo di 200 mA. Possono ad esempio essere collegate delle lampadine da 6 volt 0,5 W oppure dei led ad alta efficienza, opportunamente equipaggiati di una propria resistenza di caduta. In base alla nostra esperienza, i migliori risultati si ottengono proprio con i led ad alta luminosità, ormai di

facile reperibilità in negozi e rivenditori on-line. È possibile collegare a ciascuna uscita un solo led ponendo in serie ad esso una resistenza da 100÷200 ohm, oppure due led (in serie tra loro) con una resistenza da 50÷100 ohm. Valori di resistenza più basse permetteranno una maggiore luminosità; vi consigliamo, comun-

piano di MONTAGGIO



Quando installerete il circuito nel vostro modello, dovrete prestare attenzione al verso in cui i cavetti dei servocontrolli devono essere inseriti nei connettori SERVO1 e SERVO2: rispetto al piano di montaggio, il negativo è in basso, il positivo al centro e il segnale di controllo in alto.

ELENCO COMPONENTI:

R1: trimmer 10 kohm MO
R2÷R5: 4,7 kohm
FR1: Fotoresistenza 2-20 kohm
C1: 100 nF multistrato

C2: 470 µF 16 VL elettrolitico
C3: 22 µF 50 VL elettrolitico
D1, D2: 1N4007
U1: PIC16F88 (MF717)
T1÷T4: BC547

Varie:
- Morsettiere 2 poli (3 pz.)
- Zoccolo 9+9
- Strip maschio 3 vie (4 pz.)
- circuito stampato codice S717

Listato 1

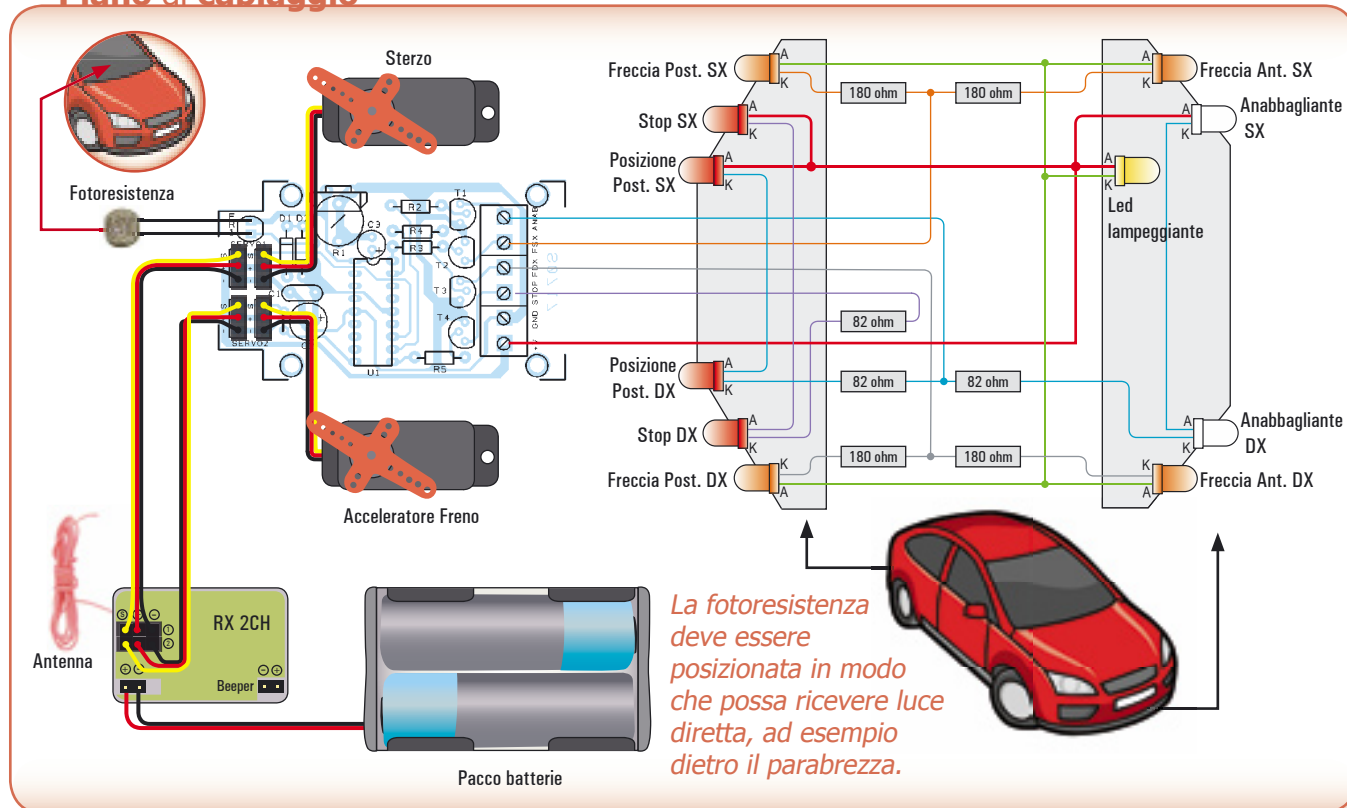
Riportiamo in questa pagina il codice sorgente completo del servoluci. Dopo aver configurato i vari registri, il programma esegue la calibrazione del neutro, ossia impara a riconoscere la larghezza degli impulsi di controllo dei due servo che corrispondono alla posizione di riposo degli stick del radiocomando. Successivamente identifica e memorizza la direzione di rotazione dei servocontrolli (quest'ultima dipende dall'impostazione del modo "reverse" sul radiocomando), in modo da riconoscere il senso di avanzamento (avanti-indietro) e la direzione (destra-sinistra). Avvia quindi il loop principale, dove misura continuamente la durata degli impulsi: quando essa differisce di 5 msec dall'impulso di neutro, attiva le luci (destra-sinistra o stop) in funzione del senso di rotazione memorizzato in precedenza. Gli anabbaglianti, invece, vengono accesi in funzione della luce ambiente.

```

-----
OPZIONI DI PROGRAMMAZIONE
-----
@      DEVICE  INTRC_OSC
@      DEVICE  WDT_OFF
@      DEVICE  PWR_T_OFF
@      DEVICE  MCLR_OFF
@      DEVICE  BOD_OFF
@      DEVICE  LVP_OFF
@      DEVICE  CPD_OFF
-----
DEFINIZIONE REGISTRI
-----
DATA @0,0,0      ' Azzerà loc. 0 e 1 EEPROM.
OPTION_REG_7=0    ' Abilita res. di pull-up.
OSCCON = %01100000 ' Clock interno.
ADCON0 = %00000000 ' Disabilita ADC.
ANSEL  = %00000000 ' Tutti pin digitali.
TRISA  = %11111100 ' RA0-RA1 out RA2-RA7 in.
TRISB  = %11001111 ' RB4-RB5 out.
DEFINE OSC 4      ' Clock a 4MHz.
symbol FDX = PORTA.0 ' Uscita freccia destra.
symbol FSX = PORTA.1 ' Uscita freccia sinistra.
symbol FR1 = PORTA.2 ' Ingresso fotoresistenza.
symbol SV2 = PORTB.3 ' Ingresso servo gas.
symbol STP = PORTB.4 ' Uscita stop.
symbol ANB = PORTB.5 ' Uscita anabbaglianti.
symbol SV1 = PORTB.6 ' Ingresso servo sterzo.
-----
DEFINIZIONE VARIABILI
-----
DIR1 var byte      ' Senso rotazione servo1 (0=antiorario).
DIR2 var byte      ' Senso dotazione servo2 (0=antiorario).
NTR1 var byte      ' Posizione di neutro servo1 (fissa).
NTR2 var byte      ' Posizione di neutro servo2 (fissa).
POS1 var byte      ' Posizione attuale servo1 (variabile).
POS2 var byte      ' Posizione attuale servo2 (variabile).
-----
IMPOSTAZIONE VARIABILI
-----
PORTA=0            ' Azzerà PORTA.
PORTB=0            ' Azzerà PORTB.
read 0, DIR1       ' Legge EEPROM locazione 0.
read 1, DIR2       ' Legge EEPROM locazione 1.
pause 1000         ' Attende 1 secondo.
-----
AUTOCALIBRAZIONE POSIZIONE DI NEUTRO
-----
Lasciare gli stick a riposo.
-----
Cal_Neutro
FDX=0              ' Etichetta della funzione.
FSX=0              ' Spegne freccia destra.
STP=0              ' Spegne freccia sinistra.
ANB=0              ' Spegne luce stop.
pause 100          ' Spegne anabbaglianti.
pulsin SV1, 1, NTR1 ' Attende 100 msec.
pulsin SV2, 1, NTR2 ' legge impulso cont. servo1.
if NTR1<120 or NTR1>180 or NTR2<100 or NTR2>200 then
    ' Legge impulso cont. servo2.
    ' Condizione di test.
    high STP        ' Un valore non è accettabile.
    pause 250       ' Avviso tre lampeggi veloci
    low STP          ' dello stop
    pause 250
    high STP
    pause 250
    low STP
    pause 250
    high STP
    pause 250
    low STP
    pause 500        ' con pause di mezzo secondo.
    goto Cal_Neutro  ' Ripete calibrazione neutro.
endif               ' Neutro calibrato. Procede.
-----
CALIBRAZIONE DELLA DIREZIONE
-----
Muovere gli stick in posizione avanti-destra.
Servo1 (SV1): sinistra-destra.
Servo2 (SV2): avanti-indietro.
-----
Cal_Dir
high STP           ' Calibrazione direzione.
pause 3000         ' Luce stop accesa
low STP            ' per 3 secondi.
pulsin SV1, 1, POS1 ' Spegne luce stop.
                    ' Legge imp. servo1, salva SV1.
-----
pulsin SV2, 1, POS2 ' Legge imp. servo1, salva SV2.
if (pos1<ntr1-20) then
    ' Se POS1 è almeno
    ' 0,2 msec minore di NTR1
    DIR1=0
    write 0, DIR1
    ' direzione servo antiorario.
    ' Salva direzione servo1.
endif
if (pos1>ntr1+20) then
    ' Prosegue.
    ' Se POS1 è almeno
    ' 0,2 msec maggiore di NTR1
    DIR1=1
    write 0, DIR1
    ' direzione servo orario.
    ' Salva direzione servo1.
    ' Prosegue.
    if (pos2<ntr2-20) then
        ' Se POS2 è almeno
        ' 0,2 msec minore di NTR2
        DIR2=0
        write 1, DIR2
        ' direzione servo antiorario.
        ' Salva direzione servo2.
    endif
    if (pos2>ntr2+20) then
        ' Se POS2 è almeno
        ' 0,2 msec maggiore di NTR2
        DIR2=1
        write 1, DIR2
        ' direzione servo orario.
        ' Salva direzione servo2.
    endif
    pause 500
    ' Prosegue.
    ' Attende mezzo secondo.
    ' Posizione stick invariata:
    ' direzione ultima memorizzata.
-----
COMANDO LUCI
-----
main
pulsin SV1, 1, POS1 ' Loop principale.
pulsin SV2, 1, POS2 ' Legge servo1, salva POS1.
                    ' Legge servo2, salva POS2.
-----
CONTROLLO SINISTRA-DESTRA
-----
if pos1>ntr1+5 then
    ' Mov. DX >6°: freccia ON.
    if DIR1=1 then
        ' Se senso orario
        high FDX
        ' accende freccia di destra.
    else
        ' Se senso antiorario
        high FSX
        ' accende freccia di sinistra.
    endif
else
    ' Mov. DX <=6°: freccia OFF.
    if DIR1=1 then
        ' Se senso orario
        low FDX
        ' spegne la freccia sinistra.
    else
        ' Se senso antiorario
        low FSX
        ' accende freccia destra.
    endif
endif
if pos1<ntr1-5 then
    ' Mov. DX <6°: freccia ON
    if DIR1=1 then
        ' Se senso orario
        high FSX
        ' accende freccia di sinistra.
    else
        ' Se senso antiorario
        high FDX
        ' accende freccia di destra.
    endif
else
    ' Mov. DX >=6°: freccia OFF
    if DIR1=1 then
        ' Se senso orario
        low FSX
        ' spegne la freccia sinistra.
    else
        ' Se senso antiorario
        low FDX
        ' spegne la freccia destra.
    endif
endif
-----
CONTROLLO AVANTI-INDIETRO
-----
if DIR2=1 then
    ' Senso orario
    if pos2<ntr2-5 then
        ' Spostamento >6° a sinistra?
        high STP
        ' Sì. Accende stop.
    else
        low STP
        ' No. Spegne stop.
    endif
else
    ' Senso antiorario?
    if pos2>ntr2+5 then
        ' Spostamento >6° a destra?
        high STP
        ' Sì. Accende stop.
    else
        low STP
        ' No. Spegne stop.
    endif
endif
-----
CONTROLLO LUCE AMBIENTE PER ACCENSIONE ANABBAGLIANTI
-----
if FR1=0 then
    ' Se manca luce
    high ANB
    ' accende anabbaglianti
else
    ' altrimenti
    low ANB
    ' spegne anabbaglianti.
endif
goto Main
' Riprende il loop principale.

```

Piano di cablaggio



morsettiera a passo 5 mm. Notate la presenza di un ponticello (realizzato con uno spezzone di reoforo) in prossimità del condensatore elettrolitico C2. Ora non resta che inserire il microcontrollore, che avrete programmato con il codice esadecimale scaricabile dal nostro sito (MF717. HEX). La fotoresistenza merita una nota a parte: come abbiamo spiegato, essa deve rilevare la luminosità dell'ambiente e pertanto va posizionata in modo appropriato. È quindi probabile che in alcuni modelli essa debba essere portata all'esterno oppure posta sotto la capote (spesso trasparente e poi verniciata, ma comunque in grado di lasciar filtrare un minimo di luce) tramite dei fili, mentre in altri potrebbe rimanere direttamente sullo stampato; questo però non siamo in grado di dirlo a priori, dato che la posizione tanto di FR1 quanto del circuito dipende esclusivamente da come è fatto il vostro modello e dal punto esatto del telaio in cui posizionerete lo stampato. A tal proposito, ricordate che sarebbe conveniente proteggere il circuito da polvere, li-

quidi, olio e vibrazioni (quasi sempre presenti in un modello RC a scoppio) inserendolo in un contenitore plastico: la nostra basetta è stata progettata per poter essere facilmente inserita in un contenitore plastico modello G404 (Futura Elettronica).

Utilizzo

Per ciò che concerne l'installazione in pratica, non possiamo sapere su quale modello lo inserirete e, pertanto, ci è impossibile definire una linea guida assoluta. Abbiamo quindi realizzato uno schema di cablaggio di massima (riportato nel riquadro in questa pagina) che, sulla base dell'esperienza nostra e di alcuni nostri collaboratori, è applicabile alla maggior parte dei modelli RC che siano gestiti tramite due servocontrolli analogici e relativa ricevente bicanale.

In questa sede, invece, spieghiamo come utilizzare il circuito dopo che lo avrete inserito nel modello e che avrete acceso tanto il radiocomando quanto la ricevente. Il PIC riceve i segnali di entrambi i servo e li analizza ciclicamente. Per poter attivare

correttamente le uscite, però, è necessario avere un riferimento di posizione di neutro ed un riferimento riguardante il senso di rotazione. Quando si installano i servocontrolli su un modello, infatti, spesso non si bada al loro senso di rotazione perché si sa a priori che esso può essere facilmente invertito impostando a dovere la funzione "reverse" del trasmettitore. Altra cosa importante è la posizione di neutro che assumono i servocontrolli quando gli stick sono lasciati in posizione centrale: logica vuole che nel nostro circuito tale situazione corrisponda ad avere tutte le uscite inattive (eccezion fatta per gli anabbaglianti, il cui stato non dipende dagli stick). Da quanto appena esposto, appare evidente che il *servoluci* necessita di una procedura (non preoccupatevi: è semplicissima) di autocalibrazione. Per prima cosa assicuratevi che il circuito sia collegato correttamente con il ricevitore ed i servocontrolli, ben fissato all'interno dell'abitacolo e protetto il più possibile da urti e polvere, evitando soprattutto infiltrazioni di liquidi che

Fig. 2 - Stick in posizione centrale, per calibrare la posizione di neutro.



Fig. 3 - Avanti-destra, per definire il senso di rotazione dei due servo.



potrebbero, tra le altre cose, ossidare i contatti dei connettori impedendo un corretto funzionamento del modellino. A questo punto accendete il trasmettitore e, successivamente, date tensione all'apparato radio del modellino. È sempre buona norma accendere prima il trasmettitore e poi il ricevitore, indipendentemente dalla presenza del nostro circuito, per evitare movimenti casuali (e spesso dannosi, soprattutto quando essi agiscono sull'acceleratore di un motore acceso) dei servocontrolli.

Quando accendete la ricevente del modello assicuratevi che gli stick

della radio siano in posizione centrale (Fig. 2) in quanto proprio durante l'accensione viene letta e memorizzata la posizione di neutro degli stick. Qualora il valore rilevato non fosse corretto, la luce di stop lampeggerà per tre volte richiedendo nuovamente questa calibrazione: ciò accade, ad esempio, quando la ricevente del modello è accesa ma la trasmittente non lo è, oppure quando gli stick si trovano in una posizione estrema. Se, invece, la calibrazione del neutro è andata a buon fine, vedrete accendersi lo stop a luce fissa per tre secondi. Durante questa fase viene impostata la direzione dello sterzo e dell'acceleratore/freno. Qualora tale calibrazione sia già stata fatta, dovrete semplicemente lasciar passare i tre secondi senza muovere gli stick e, nel momento in cui le luci si spengheranno, il circuito sarà pronto a funzionare. Se invece avrete la necessità di calibrare il senso dei servo perché, ad esempio, accelerando si accendono le luci di stop, oppure le frecce sono invertite di posizione, non dovrete fare altro che attivare lo stick corrispondente alla calibrazione da eseguire portandolo all'estremo della corsa. Il *servoluci* stabilisce la direzione apprendendo la posizione "avanti-destra" (Fig. 3): ciò significa che, durante i tre secondi in cui gli stop sono accesi a luce fissa, lo stick del gas deve essere portato tutto avanti, e quello dello sterzo deve trovarsi tutto a destra. Quando la luce di stop si spegne, la ca-

librazione è finita e potrete riportare gli stick nella posizione di neutro. Se avete la necessità di ripetere la calibrazione è sufficiente spegnere e riaccendere la ricevente (che alimenta il circuito) e ripetere la procedura. A questo punto, muovendo lo stick dello sterzo a sinistra, vedrete le frecce di sinistra accendersi. La stessa cosa accade muovendo lo stick a destra oppure lo stick del gas indietro, che determina l'accensione degli stop. Tenete presente un particolare: non abbiamo previsto il lampeggio delle frecce in quanto nei modellini radio-comandati, i movimenti dello sterzo sono sempre molto brevi e pertanto sarebbe difficile apprezzare il lampeggio delle luci. Noi, comunque, abbiamo incluso un led lampeggiante nel nostro cablaggio: in questo modo possiamo generare la cadenza delle frecce in maniera semplice ed efficace, senza appesantire il firmware, ma la massima corrente fornibile alle frecce si riduce a 20 mA circa, limite massimo oltre il quale il led lampeggiante può danneggiarsi. Per alimentare le frecce con correnti maggiori dovrete sostituire il led lampeggiante con un oscillatore dotato di uscita a relé.



per il

MATERIALE

Il materiale utilizzato in questo progetto è facilmente reperibile presso qualsiasi negozio di elettronica. Il master del circuito stampato ed il firmware col quale programmare il microcontrollore possono essere scaricati dal sito della rivista (www.elettronicaain.it). Il microcontrollore già programmato (cod. MF717) può essere acquistato al costo di 12.00 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>

Dimmer bicanale radiocomandato

di *Alessandro Sottocornola*

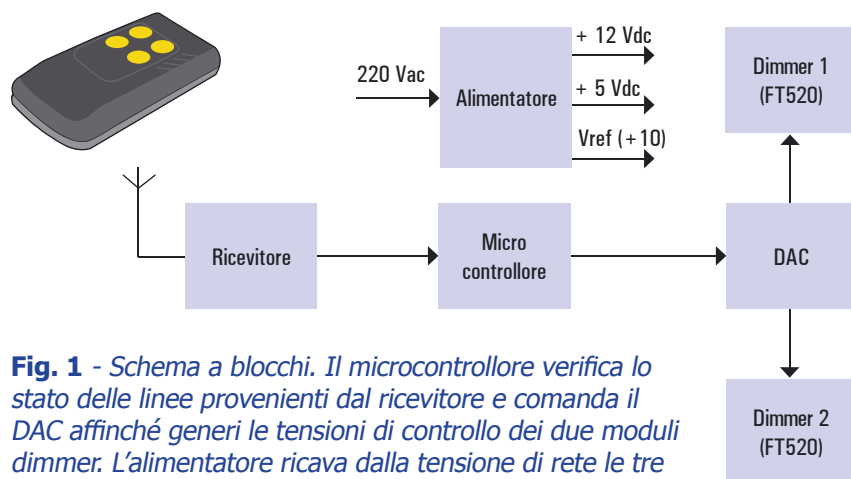
Consente di attivare due lampade ad incandescenza alimentate a tensione di rete e di variarne la luminosità.

Dispone di un telecomando a quattro pulsanti, tramite i quali è possibile sia forzare l'accensione e lo spegnimento immediato delle lampade che variare la luminosità di ognuna di esse.

Tempo fa un collaboratore ci propose l'idea di realizzare un dimmer bicanale che potesse essere controllato a distanza tramite un telecomando, sostenendo che le varie soluzioni disponibili in commercio (vendute spesso a prezzi esorbitanti), a parte l'aspetto estetico raffinato e le dimensioni compatte, dal punto di vista tecnico non offrono molto di più rispetto ad un circuito di nostra progettazione. Ebbene, l'idea ci è piaciuta ed il progetto pratico che ne è scaturito è descritto nelle pagine seguenti. Ciò che apparve chiaro fin da subito furono le linee guida di realizzazione del progetto:

il dispositivo avrebbe dovuto essere economico (nei limiti del possibile) senza però rinunciare alla qualità e alle prestazioni, nonché facile da realizzare e sicuro nell'utilizzo.

Bene, viste le premesse e dopo le dovute valutazioni, abbiamo realizzato un apparato che sicuramente non può essere disturbato da altri telecomandi presenti nell'area di copertura, è affidabile ed è in grado di controllare i carichi collegati alle uscite con grande precisione. Seguitemi con attenzione nella nostra descrizione e scoprirete tutti gli aspetti tecnici di questa interessante ➤



soluzione, ricavando spunti per la realizzazione di circuiti di controllo analoghi.

Schema a blocchi

Questo dispositivo, in realtà abbastanza semplice, può essere suddiviso in sei blocchi, come illustrato in Fig. 1. Il cuore del sistema è rappresentato dal microcontrollore, che verifica costantemente lo stato delle

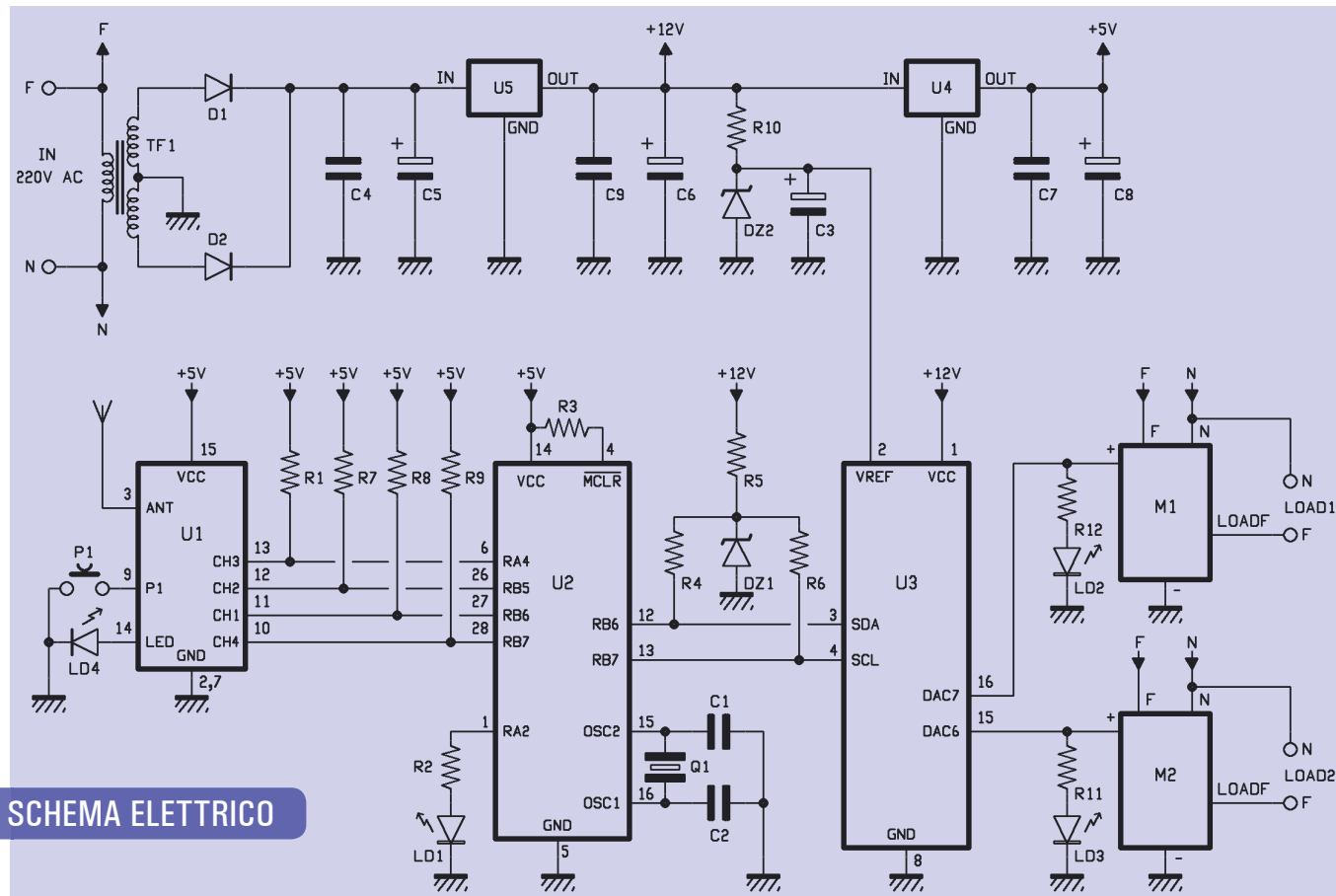
linee di comando provenienti dal ricevitore radio. Quando viene premuto uno dei pulsanti del trasmettitore, il ricevitore effettua la decodifica e comunica al microcontrollore quale pulsante è stato attivato. A questo punto il micro esegue (secondo quanto previsto dal suo programma) le elaborazioni del caso, elaborazioni che dipendono sia dal pulsante premuto sia dalla durata della pres-

sione; invia, quindi, al blocco DAC (convertitore digitale-analogico a due canali) i comandi necessari affinché questi generi sulle due uscite le tensioni di riferimento necessarie ai due circuiti di controllo del carico (Dimmer 1 e Dimmer 2). A tal proposito, vi anticipiamo che, come dimmer, abbiamo utilizzato quello da noi realizzato in passato (codice progetto FT520ANK) e pubblicato su *Elettronica In* n. 95.

L'ultimo blocco da descrivere è l'alimentatore; esso preleva la tensione di rete per ricavare tre valori di tensione continua (+5, +10 e +12 Vdc) necessari per fornire corrente alle varie parti del circuito.

Schema elettrico

Iniziamo la descrizione dello schema elettrico partendo dal ricevitore del radiocomando (U1), un RX-4M-HCS di produzione Aurel, più volte utilizzato in molti nostri progetti, ultimo in ordine di tempo il segna-



punti per tennis, pubblicato sia su questo fascicolo che sul fascicolo precedente. Proprio perché si tratta di un dispositivo ricorrente nelle nostre pagine, qui ci limitiamo a descriverne le caratteristiche principali: si tratta di un ricevitore OOK (On Off Keying) a 433,92 MHz, dotato di decodifica rolling code KeeLoq di Microchip. Esso è in grado di attivare le sue quattro uscite open collector (che possono funzionare sia in modalità monostabile che bistabile) quando risulta premuto almeno uno dei pulsanti di un trasmettitore a cui è accoppiato. La procedura di accoppiamento viene effettuata agendo sul pulsante P1, collegato direttamente al ricevitore U1. Come trasmettitore, noi abbiamo scelto un TX4M-HCS (basato su una codifica HCS301), anch'esso ospite frequente nelle nostre pagine. Per approfondire il funzionamento di questo dispositivo, potete rileggere l'articolo relativo al radiocomando a quattro canali, pubblicato sul fascicolo n. 118, oppure osservare con attenzione le tabelle da 1 a 4, in cui abbiamo riportato le fasi per far apprendere al ricevitore un telecomando (tabella 1), per cambiare il modo di funzionamento delle uscite (tabella 2 e 3), e per ripristinare la configurazione di fabbrica (tabella 4).

Torniamo ora al nostro circuito: le quattro uscite open collector di U1 vanno a pilotare le linee RA4, RB5, RB6 e RB7 (configurate come ingressi) del microcontrollore U2, un PIC16F88. Relativamente alle uscite della decodifica, abbiamo detto che esse possono operare sia in modalità monostabile che bistabile: nel nostro caso, affinché il circuito funzioni correttamente, le uscite devono operare in modalità monostabile (che corrisponde a quella predefinita). Notate che ogni linea dispone di una resistenza di pull-up, necessaria per garantire agli ingressi di U2 il livello logico alto quando le uscite open collector di U1 sono a riposo: da ciò si

Si tratta di un dimmer gestito da un microcontrollore, in grado di regolare la tensione applicata al carico (al massimo di 1 kW) variando il ritardo di innesco di un triac. Tale ritardo, calcolato sul passaggio dello zero tra una semionda negativa e una positiva (e viceversa), è proporzionale alla tensione di controllo applicata al circuito ed è compresa tra 0 e 10 volt; l'ingresso al quale viene applicata la tensione di controllo risulta isolato galvanicamente dal resto del circuito.

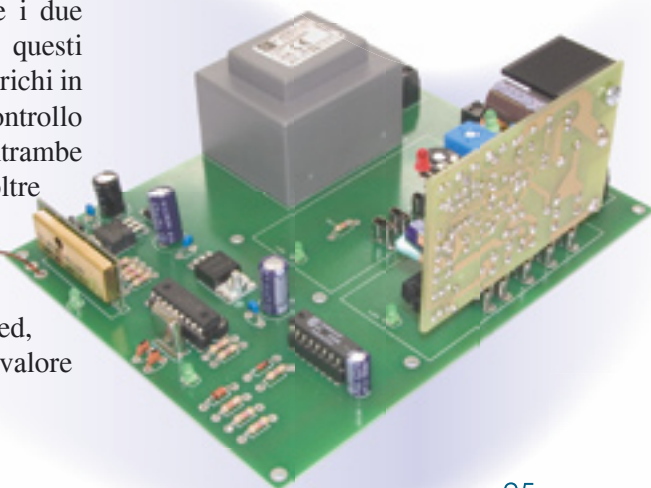


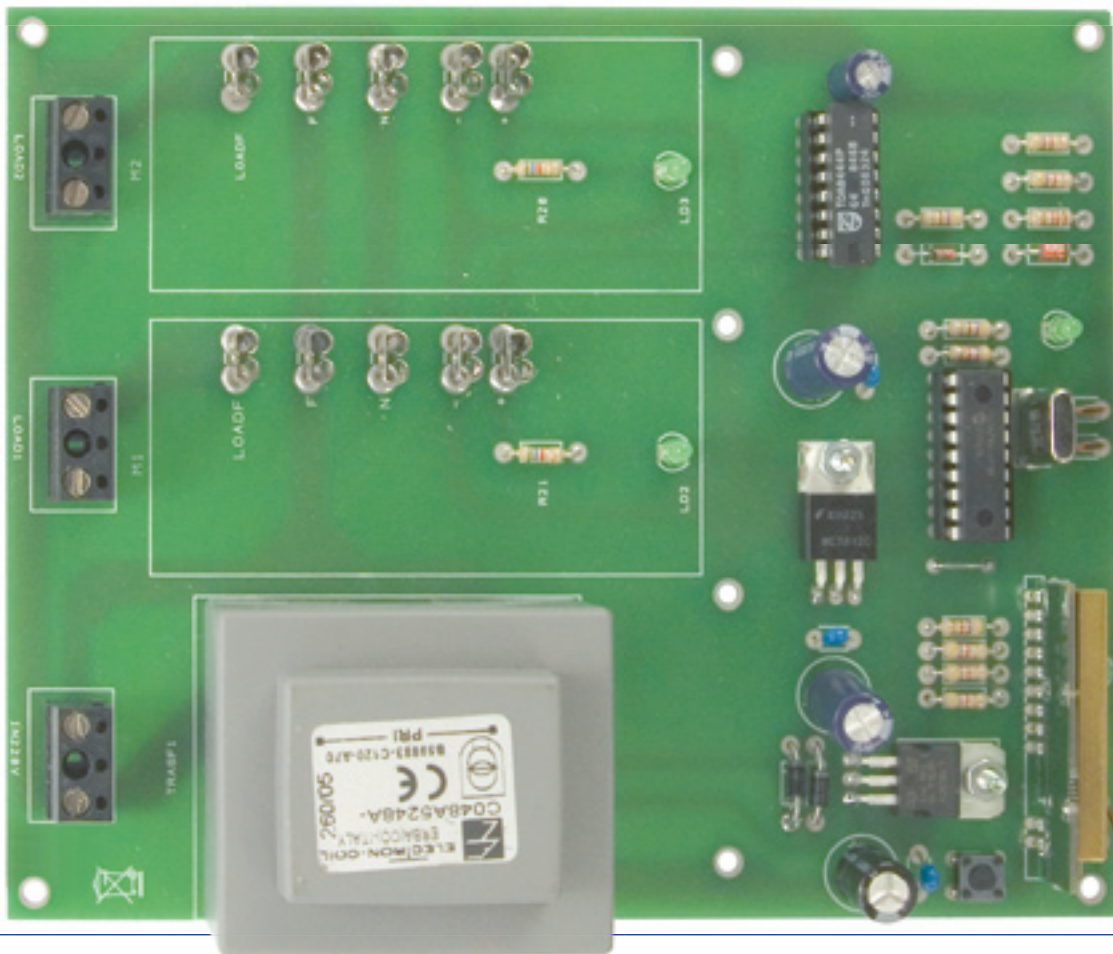
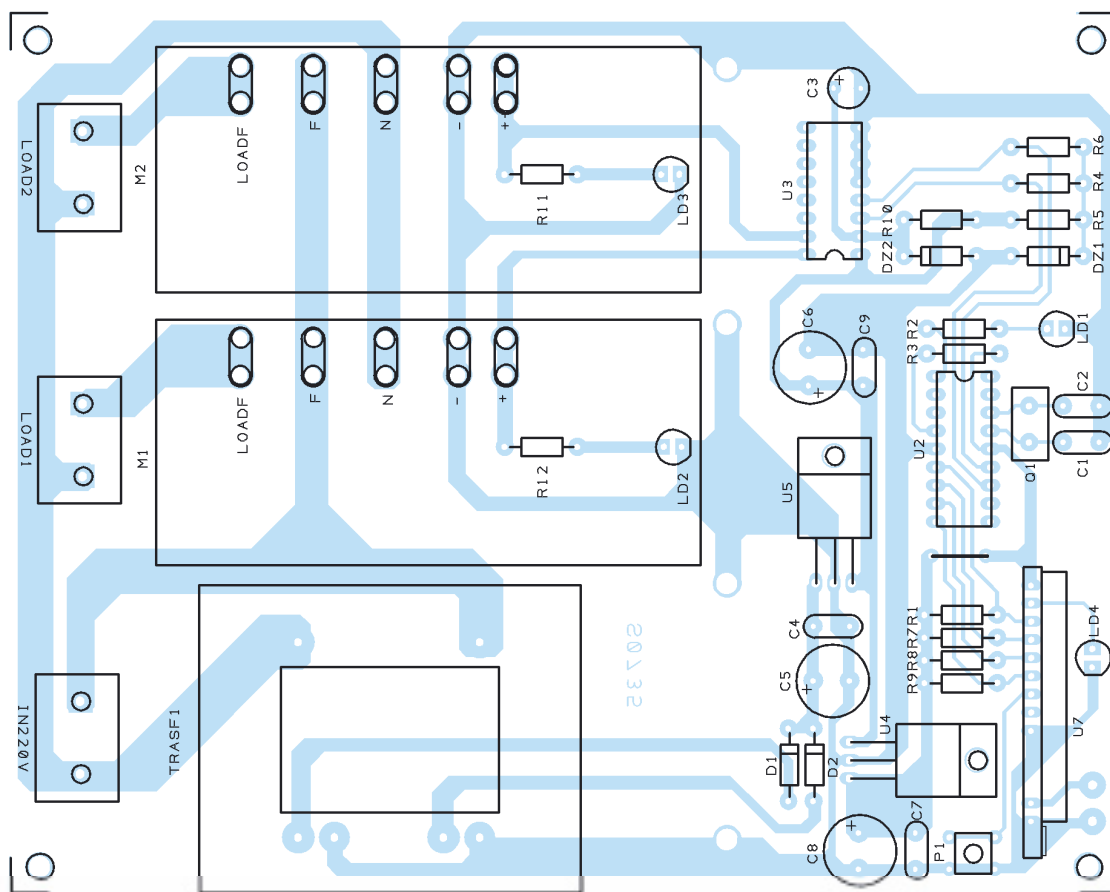
È alimentato direttamente a tensione di rete, dalla quale ricava sia la tensione di alimentazione del carico sia quella di alimentazione ai circuiti interni: quest'ultima viene ottenuta abbassando la tensione di rete per effetto della reattanza capacitiva di un condensatore di grosse dimensioni (collegato in serie alla linea della fase) che determina una caduta di tensione sufficiente per alimentare l'ingresso di uno stabilizzatore di tensione (7805); il modulo, pertanto, non necessita di alcun trasformatore di alimentazione. La figura sopra mostra il dimmer montato: notate la grossa bobina, utilizzata come filtro per i disturbi a radiofrequenza, generati dal triac quando il microcontrollore ne innesca la conduzione. Il progetto del dimmer è stato presentato sul fascicolo n. 95 di Elettronica In; di questo circuito è disponibile la scatola di montaggio completa che può essere acquistata on-line sul sito www.futurashop.it.

deduce che il microcontrollore considera una linea come attiva se questa va a livello logico basso. Quando una delle suddette linee cambia di stato, il microcontrollore, eseguendo quanto previsto dal programma (che analizzeremo più avanti), utilizza la linea per identificare tanto il pulsante premuto quanto la durata della pressione; elabora quindi i dati ricavati e, infine, invia al convertitore digitale-analogico U3 (un TDA8444) le istruzioni necessarie affinché questo generi sulle uscite DAC6 e DAC7 le tensioni di controllo necessarie per attivare i due moduli dimmer (M1 e M2); questi ultimi, infine, attiveranno i carichi in funzione della tensione di controllo applicata al loro ingresso. Entrambe le uscite del DAC pilotano, oltre all'ingresso di controllo del dimmer, anche due led (LD2 e LD3), così da fornire, a seconda della luminosità dei led, un'indicazione visiva del valore della tensione di controllo.

Il microcontrollore, inoltre, comanda la serie LD1-R2 tramite la linea RA2 (usata come uscita), in modo da comunicare differenti stati di funzionamento, come la fase di avvio, la ricezione di un comando, il cambio della tensione di riferimento ai dimmer, eccetera.

Relativamente a M1 e M2 (identici tra loro), abbiamo già anticipato che si tratta un nostro progetto cui abbiamo dedicato un articolo sulla rivista n. 95: non volendo ripetere quanto già ampiamente spiegato in quell'occasione, ci limitiamo a de- ➤





Per ragioni di spazio, i disegni a lato relativi alla basetta sono ridotti al 90%.

I due moduli dimmer sono fissati alla piastra base mediante connettori faston da stampato, che svolgono la duplice funzione di contatto elettrico e di supporto meccanico. Durante il collaudo siate molto prudenti: la tensione di rete è presente in diversi punti del circuito. Per accoppiare il trasmettitore ed il ricevitore, è sufficiente eseguire la procedura illustrata in tabella 1. In caso di errore, eseguite la cancellazione della memoria secondo la procedura di tabella 4.

ELENCO COMPONENTI:

R1, R3, R4, R6÷R9: 4,7 kohm
R2, R10: 470 ohm
R5: 1,2 kohm
R11, R12: 5,6 kohm
C1, C2: 10 pF ceramico
C3: 100 µF 25 VL elettrolitico
C4, C7, C9: 100 nF multistrato
C5, C6: 470 µF 25 VL elettrolitico
C8: 470 µF 16 VL elettrolitico
D1, D2: 1N4007
DZ1: zener 5,1V 1 W
DZ2: zener 10V 400 mW
LD1: led 3mm verde
LD2: led 3mm verde
LD3: led 3mm verde
Q1: quarzo 20 MHz
U1: RX-4M-HCS
U2: PIC16F88 (MF735)
U3: TDA8444
U4: 7805
U5: 7812
P1: Microswitch
TRASF1: Trasformatore di alimentazione 220/12+12 Vac

Varie:

- Morsettiera 2 poli passo 10 (3 pz.)
- Zoccolo 8+8
- Zoccolo 9+9
- Faston femmina da cs (10 pz.)
- vite 3 MA 8 mm (2 pz.)
- Dado 3 MA (2 pz.)
- Circuito stampato S0735

scriverne le funzioni principali nel riquadro di pagina 25.

Non resta ora che analizzare lo stadio di alimentazione (abbastanza complesso, nonostante il dispositivo sia piuttosto semplice) che, partendo dalla tensione di rete, deve fornire +5 Vdc per alimentare il microcontrollore U2 ed il ricevitore U1, +12 Vdc per alimentare il DAC U3 e, infine, +10 Vdc come tensione di riferimento al DAC.

La tensione di rete, applicata ai morsetti F e N, alimenta sia i due dimmer (M1 e M2) che il primario del trasformatore TF1. Nella configurazione da noi adottata, il doppio secondario di TF1 pilota i diodi D1 e D2 in modo da realizzare un raddrizzatore a doppia semionda con centrale del secondario a massa. L'alternata "raddrizzata" viene livellata dai condensatori di filtro C4-C5 ed applicata all'ingresso di U5, un classico regolatore di tensione a tre terminali modello 7812 che ricava 12 volt ben stabilizzati sulla sua uscita.

Questa tensione viene utilizzata per alimentare il DAC (U3) e per generare (tramite il circuito R10-DZ2-C5) la tensione di riferimento a 10 V necessaria allo stesso DAC. C6 e C9 filtrano ulteriormente la tensione di 12 V da eventuali residui di alternata, presenti all'uscita di U5.

La tensione a 12 V viene anche utilizzata per ricavare tramite U4 (un 7805) i 5 V necessari al micro e al ricevitore.

Rimane un ultimo particolare da analizzare: come può il micro (alimentato a 5 V) inviare i comandi al DAC (che funziona a 12 V) attraverso il bus I²C? I livelli logici sono teoricamente incompatibili!

Ebbene, la rete costituita da R5, R4, R6 e DZ1 (uno zener da 5,1 V), insieme alle linee RB6 e RB7 del micro (configurate come open collector) permette di adattare perfettamente i livelli elettrici sulle linee SDA e SCL di U3.

Realizzazione pratica

Per prima cosa occorre realizzare la basetta con la tecnica della fotoincisione: a tal proposito, come al solito, è possibile scaricare gratuitamente dal nostro sito www.elettronica.in.it il master in formato 1:1.

Una volta realizzata la basetta e praticati i fori necessari, potete iniziare il montaggio partendo dai componenti di piccole dimensioni quali resistenze, condensatori non polarizzati, diodi, led, eccetera.

Fate attenzione a rispettare il verso dei componenti polarizzati (aiutatevi con le illustrazioni riportate nel piano di montaggio). Proseguite quindi saldando i due zoccoli DIL (un 9+9 per il micro e un 8+8 per il DAC). Montate ora i condensatori elettrolitici, gli stabilizzatori di tensione (i cui piedini devono essere piegati a 90° come visibile in figura) avendo cura di fissarli allo stampato tramite delle viti da 3 MA. A questo punto saldate i connettori faston e le morsettiere, lasciando per ultimi il trasformatore e l'antenna, quest'ultima ricavabile con uno spezzone di filo rigido della lunghezza di 17 cm.

Prima di inserire il modulo radio U1, il micro (opportunamente programmato con il codice esadecimale MF735.HEX, scaricabile anch'esso gratuitamente dal nostro sito) ed il convertitore digitale analogico nei rispettivi zoccoli, accendere il circuito e verificate che tutte le tensioni di alimentazione (+5, +12 e +10) siano corrette, tenendo presente che una tolleranza di 0,1 V in più o in meno è accettabile.

Spegnete il circuito, attendete che i condensatori elettrolitici si siano scaricati, quindi posizionate U1, U2 e U3 nelle rispettive sedi. Potete ora accendere il circuito e procedere con la fase di accoppiamento tra ricevitore e trasmettitore (fase in cui il ricevitore U1 impara a riconoscere il pattern di dati generato dal trasmettitore), seguendo quanto riportato nelle tabelle da 1 a 4.



Prima di inserire i due dimmer sugli appositi faston, verificate che tutto funzioni correttamente. Premete un pulsante qualsiasi: se il led LD1 rimane acceso per quasi tutta la durata della pressione del pulsante, allora il circuito funziona. Come ulteriore conferma del funzionamento, pote-

te verificare che LD2 e LD3 si illuminino con differente intensità a seconda di come agite sui pulsanti, e che mantenendo premuto continuamente un pulsante qualsiasi, ad un certo punto il led LD1 inizierà ad emettere cinque lampeggi veloci intervallati da una breve pausa, segna-

landovi che avete raggiunto il limite (superiore o inferiore) di regolazione. Non vi rimane che spegnere il circuito, inserire i due dimmer, collegare due lampadine a 220 V o due carichi simili, riaccendere e verificare che la luminosità delle lampadine vari in funzione di come agite sui

Listato 1

Riportiamo qui di seguito parte del programma di gestione dei dimmer scritto utilizzando il Compilatore Basic della MicroEngineering Labs. In particolare, abbiamo selezionato la sola gestione dei pulsanti P1 e P2, dato che per P3 e P4 avviene esattamente la stessa cosa. Dopo aver configurato i vari registri e le uscite, il micro comunica di essere attivo facendo lampeggiare cinque volte il led LD1. Azzerà, quindi, la RAM ed inizia ad eseguire il loop principale, in cui verifica lo stato delle linee di comando associate ai pulsanti. Se almeno una di esse viene attivata (portata a zero logico), il programma ricerca la linea corrispondente ed avvia la parte di programma relativa ad essa. A pulsante identificato, verifica la durata della pressione del tasto, incrementando i contatori CONT e CONT2. Il primo determina il riconoscimento della pressione breve, mentre il secondo agisce sulla pressione lunga. La pressione breve richiama una subroutine che aggiorna una sola volta il DAC, generando così un'accensione o uno spegnimento immediato del carico; la pressione lunga, invece, attiva un'altra sequenza di comandi con cui il micro va ad aggiornare ciclicamente l'uscita del DAC: il risultato è una variazione della luminosità del carico (in rampa crescente o decrescente a seconda che siano premuti P1 o P2) visibile a occhio, come avviene in tutti i dimmer. Ogni ciclo di lettura del pulsante determina la memorizzazione della luminosità attualmente impostata e l'aggiornamento delle uscite del DAC, effettuata tramite la subroutine TDA, che invia sulle linee RB6 ed RB7 i comandi destinati al DAC secondo il formato I²C.

```
***** I N I Z I O *****
' Lampeggio LED: indica che la scheda è alimentata
*****
LOW LED ' Spegne LED.
FOR TMP=0 to 5 ' Avvia ciclo di cinque conteggi.
  TOGGLE LED ' Fa lampeggiare il LED.
  PAUSE 200 ' Pausa di 200 msec.
NEXT TMP ' Ripete ciclo. Al quinto esce.
LOW LED ' Spegne LED per sicurezza.
***** MAIN *****
CLEAR ' Azzerà memorie.
MAIN: ' Inizio loop principale.
' ***** VERIFICA SE UN PULSANTE PREMUTO.
IF RAD1=0 OR RAD2=0 OR RAD3=0 OR RAD4=0 THEN
  HIGH LED ' Uno dei canali radio è attivo quindi
  ' accende LED.
  CONT=0 ' Azzerà contatore1
  CONT2=0 ' Azzerà contatore2
  PAUSA=30 ' PAUSA viene impostata a 30 msec.
' ***** VERIFICA PULSANTE1 PREMUTO.
IF RAD1=0 THEN ' Pulsante1 attivo? DOWN 1
  WHILE RAD1=0 AND CONT2<1300
    ' Esegue solo se pulsante1 attivo
    ' e contatore2 inferiore a 1300.
    CONT=CONT+1 ' Incrementa contatore1
    CONT2=CONT2+1 ' Incrementa contatore2
    IF CONT>20 THEN ' Se contatore 1 maggiore di 20
      PAUSA = 5 ' abbassa pausa a 5 msec.
      CONT=0 ' e azzerà contatore1
      IF CH1>0 THEN ' Se CH1 maggiore di zero
        ' (lampada accesa)
        CH1=CH1-1 ' decrementa CH1 (dec. luminosità)
        WRITE 0,CH1 ' e memorizza stato CH1,
      ELSE ' altrimenti
        WRITE 0,CH1 ' memorizza stato CH1
        FOR TMP=0 TO 4 ' e fa lampeggiare il LED
          TOGGLE LED ' per indicare raggiunto minimo.
          PAUSE 100 ' Pausa di 100 msec.
        NEXT TMP ' Ripete ciclo.
        LOW LED ' Spegne il led. Segnalazione finit
        CONT2=1300 ' Riporta contatore2 a 1300
        ' per uscire dal ciclo di while.
      ENDIF
      GOSUB TDA ' Chiama routine per scrivere DAC.
    ENDIF
    PAUSE PAUSA ' pausa viene definito dinamicamente
    ' in funzione della durata della
    ' pressione del pulsante
    ' (pressione breve: pausa 30,
    ' pressione lunga: pausa 5).
    ' Fine ciclo.
  WEND
  IF CONT2<100 THEN ' Pulsante premuto meno di 1 s.
    READ 0,CH1 ' Ricarica CH1 da loc. mem. 0
  ENDIF
ENDIF
***** VERIFICA PULSANTE2 PREMUTO.
IF RAD2=0 THEN ' Pulsante 2 attivo? DOWN CH2
  *****
  ENDIF
***** VERIFICA PULSANTE3 PREMUTO.
IF RAD3=0 THEN ' Pulsante 3 attivo? UP CH1
  WHILE RAD3=0 AND CONT2<1300
    ' Esegue solo se pulsante3 attivo
    ' e contatore2 inferiore a 1300.
    CONT=CONT+1 ' Incrementa contatore1.
    CONT2=CONT2+1 ' Incrementa contatore2.
    IF CONT>20 THEN ' Se contatore1 maggiore di 20
      PAUSA = 5 ' PAUSA viene abbassata a 5 msec
      CONT=0 ' e contatore1 azzerato.
      IF CH1<63 THEN ' Se CH1 minore di 63 (poca luce)
        CH1=CH1+1 ' Incrementa valore di CH1
        WRITE 1,CH1 ' e lo salva in memoria.
      ELSE ' altrimenti
        CONT2=1300 ' Riporta contatore2 a 1300
        ' per uscire dal ciclo di while.
      FOR TMP=0 TO 4
        TOGGLE LED ' Fa lampeggiare il led
        PAUSE 100 ' quattro volte.
      NEXT TMP
      LOW LED ' Spegne il led'
      WRITE 1,CH1 ' Salva in memoria CH1
    ENDIF
    GOSUB TDA ' Chiama routine per scrivere DAC.
  ENDIF
  PAUSE PAUSA ' Chiama pausa.
  ' Fine ciclo.
  IF CONT2<100 THEN ' Pulsante premuto meno di 1 s.
    READ 1,CH1 ' Ricarica CH1 da loc. mem. 1
  ENDIF
  ***** VERIFICA PULSANTE4 PREMUTO.
  IF RAD4=0 THEN ' Pulsante 4 attivo? UP CH2.
    *****
    ENDIF
***** AGGIORNA DAC
GOSUB TDA ' Se nessun pulsante premuto
' va comunque a scrivere DAC.
ENDIF
LOW LED ' Spegne il led in ogni caso.
***** RIPETE CICLO.
GOTO MAIN ' Riavvia il ciclo principale.
' ***** ROUTINE SCRITTURA DAC.
TDA: ' Routine che aggiorna il DAC.
  I2Cwrite sda,scl,%01000000,%00000000,
  [0,0,0,0,0,0,CH1,CH2]
RETURN ' Termine routine.
```

Fig. 2 Funzioni corrispondenti ai pulsanti del telecomando

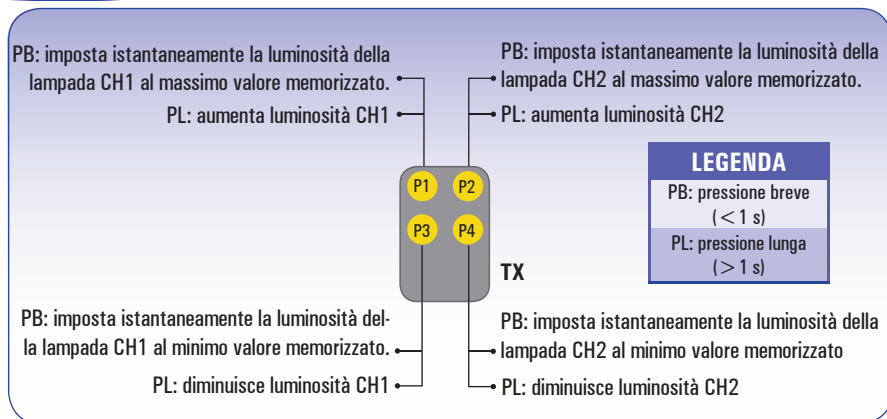


Tabella 1 - Autoapprendimento telecomando (massimo 10 telecomandi)

Fase	P1	LD4	TX	Note
1	Pressione breve	Lampeggiante per 10 secondi	Non attivo	Inizio procedura
2	Riposo	Lampeggiante per 10 secondi	Pressione di un pulsante sul TX	Al ricevitore basta riconoscere un solo pulsante del TX per decodificarli tutti
3	Riposo	Acceso luce fissa	Non attivo	Conferma apprendimento
4	Riposo	Spento	Non attivo	Procedura terminata

Tabella 2 - Impostazione del modo di funzionamento delle uscite: da monostabile (predefinito) a bistabile

Fase	P1	LD4	TX	Note
1	Pressione breve	Lampeggiante per 10 secondi	Non attivo	Inizio procedura
2	Pressione breve	Acceso luce fissa	Non attivo	Attesa pulsante premuto
3	Riposo	Acceso luce fissa	Pressione del pulsante	Premere solo il pulsante che deve comandare l'uscita in bistabile
4	Riposo	Tre lampeggi	Non attivo	Conferma cambio modo uscita
5	Riposo	Spento	Non attivo	Procedura terminata

Tabella 3 - Impostazione del modo di funzionamento delle uscite: da bistabile a monostabile (predefinita)

Fase	P1	LD4	TX	Note
1	Pressione breve	Lampeggiante per 10 secondi	Non attivo	Inizio procedura
2	Pressione breve	Acceso luce fissa	Non attivo	Attesa pulsante premuto
3	Riposo	Acceso luce fissa	Pressione pulsante	Premere solo il pulsante che deve comandare l'uscita in monostabile
4	Riposo	Due lampeggi	Non attivo	Conferma cambio modo uscita
5	Riposo	Spento	Non attivo	Procedura terminata

Tabella 4 - Cancellazione memoria (ripristino configurazione di default)

Fase	P1	LD4	TX	Note
1	Pressione breve	Lampeggiante per 10 secondi	Non attivo	Inizio procedura
2	Pressione lunga	Spento	Non attivo	Mantenere premuto almeno 5 secondi
3	Rilasciare	Cinque lampeggi	Non attivo	Conferma cancellazione memoria
4	Riposo	Spento	Non attivo	Procedura terminata. Uscite in monostabile. Nessun TX accoppiato

pulsanti, come descritto in maniera più dettagliata nella tabella riportata a lato.

Come funziona

Esaurita la parte pratica, vediamo ora come funziona il firmware, tenendo presente che le funzioni dei pulsanti stabilite da esso sono riassunte in Fig. 2. Ricordatevi inoltre che, dal momento che i due canali funzionano allo stesso modo, nel Listato 1 abbiamo riportato la sola porzione di codice sorgente relativa al controllo del canale 1, tralasciando tutto ciò che riguarda la configurazione delle uscite, la programmazione dei registri e la gestione del secondo canale.

All'avvio il microcontrollore inizializza i vari registri, azzerà il contenuto della RAM e imposta tutte le variabili di servizio, quindi segnala di aver ultimato la fase di configurazione iniziale facendo lampeggiare il led LD1 per cinque volte. A questo punto viene avviato il loop principale, in cui il micro controlla che nessuna delle linee di comando provenienti dal ricevitore U1 (si tratta - ricordiamo - di un RX-4M-HCS con codifica Rolling Code) sia attiva. Qualora almeno una di esse dovesse divenire attiva (livello logico "0"), il micro la riconosce e va ad eseguire la porzione di codice relativa ad essa. Tornate ora alla Fig. 2: i due pulsanti in alto controllano l'incremento della luminosità, mentre i due in basso agiscono sulla riduzione. Ogni pulsante, inoltre è in grado di svolgere due funzioni differenti a seconda che venga premuto brevemente o mantenuto premuto a lungo. In particolare la pressione lunga agisce sul controllo in modo dimmer mentre la pressione breve determina l'accensione o lo spegnimento immediato della lampada, ma non nel classico modo ON-OFF, bensì in un modo più raffinato che descriveremo più avanti.

Procediamo per gradi, ipotizzando ➤

di mantenere premuto P1: la lampada collegata al dimmer del primo canale si accende e si porta gradualmente alla massima luminosità, raggiunta la quale il microcontrollore fa lampeggiare il led LD1 per informare l'utente che si è raggiunto il limite massimo. Rilasciando il pulsante, la lampada rimane accesa. Quanto appena descritto accade esattamente anche quando si preme il pulsante P2, che controlla in accensione (incremento di luminosità) il carico collegato al secondo canale.

Se ora si agisce sui pulsanti P3 e P4 (mantenendoli premuti), si ottiene l'effetto inverso, ossia entrambi i carichi accesi alla massima luminosità, si spengono gradualmente. Il raggiungimento del valore di luminosità minimo viene segnalato con dei lampeggi di LD1, esattamente come visto poco fa. Rilasciando il pulsante, i carichi restano spenti.

A questo punto è evidente che la regolazione del livello di luce desiderato viene effettuata mantenendo premuti i pulsanti del telecomando per il tempo necessario.

Il programma del micro, però, al rilascio del pulsante, fa un'altra cosa molto importante: salva in memoria EEPROM un valore che corrisponde alla luminosità della lampada nell'esatto momento in cui si è rilasciato il pulsante. Questa operazione viene eseguita per ogni pulsante, in modo che il programma ricordi esattamente quattro valori distinti:

- luminosità massima impostata per il canale 1 (associata a P1),
- luminosità minima impostata per il canale 1 (associata a P3),
- luminosità massima impostata per il canale 2 (associata a P2),
- luminosità minima impostata per il canale 2 (associata a P4).

Ora, una pressione breve del pulsante P1 determina il richiamo della luminosità massima impostata per il canale 1, mentre la pressione di P3 richiama la luminosità minima.

Per il secondo canale vale esattamente quanto appena descritto. In

questo modo è possibile regolare esattamente (mantenendo premuti i pulsanti di comando) i due livelli di luminosità che vogliamo considerare come massimo e minimo, e richiamarli velocemente tramite una pressione breve degli stessi pulsanti.

Prestate attenzione ad un particolare: per realizzare un controllo di tipo ON-OFF, è necessario regolare l'intensità luminosa al massimo tramite i pulsanti P1 e P2 (ON) e regolare il minimo (luce spenta, corrispondente a OFF) con P3 e P4.



per il

MATERIALE

Il materiale utilizzato in questo progetto è facilmente reperibile presso qualsiasi negozio di elettronica. Il master del circuito stampato ed il firmware col quale programmare il microcontrollore possono essere scaricati dal sito della rivista (www.elettronica.in.it). Il microcontrollore già programmato (cod. MF735) ed il modulo dimmer FT520ANK possono essere acquistati al costo di 12.00 e 19.00 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>

Rue Royale

Bolero, localizzatore portatile programmabile

di Alessandro Orlando

Nuovo protagonista nel mercato dei sistemi di localizzazione satellitare, Bolero è un dispositivo completamente programmabile, funzionante a batteria e provvisto di cellulare GSM/GPRS integrato, ricevitore GPS a 20 canali e sensore di movimento. Comunica attraverso le interfacce Bluetooth, IEEE 802.15.4 e CAN Bus. Telecomando bidirezionale e scheda di espansione I/O sono disponibili come optional.



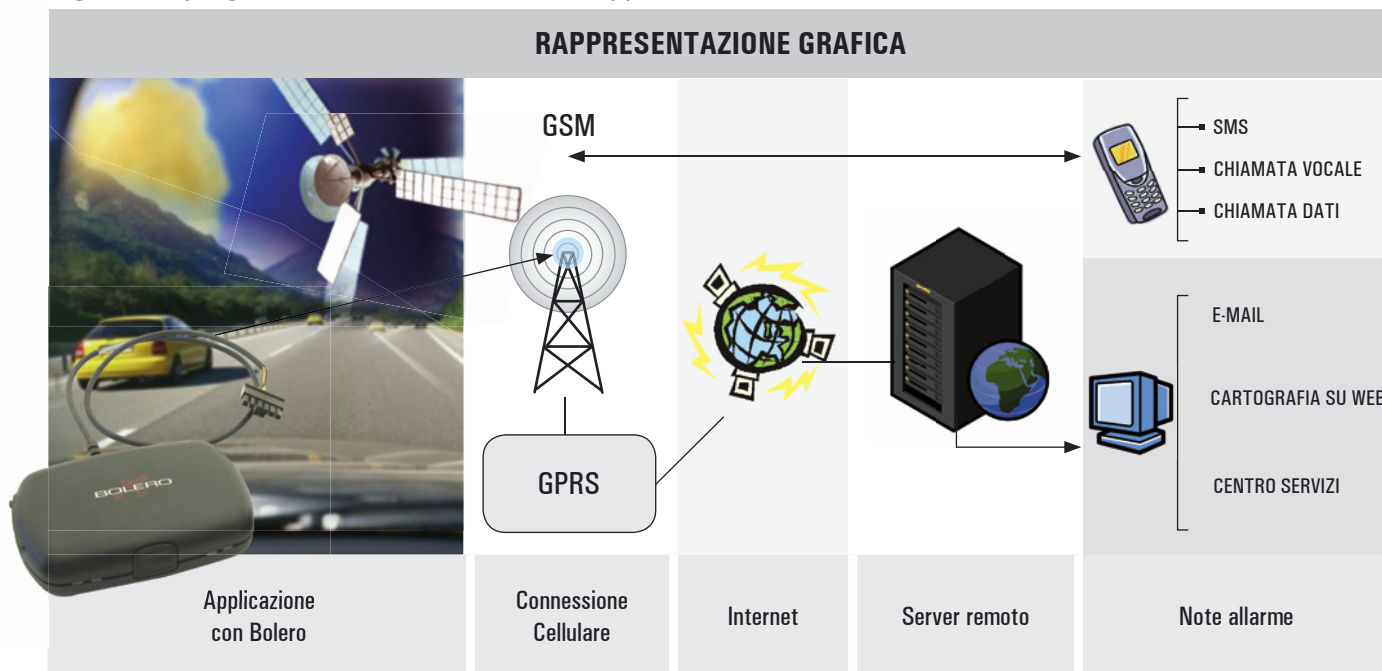
Nel corso del 2007 abbiamo dedicato ampio spazio a Mambo, un localizzatore satellitare completamente programmabile molto completo e particolarmente versatile. Chiudendo l'articolo pubblicato sulla rivista numero 122, accennavamo al fatto che Falcom (la casa produttrice di Mambo) stava per presentare una versione più evoluta del dispositivo: tale versione, chiamata Bolero, sarà l'argomento principale sia di questo sia di altri articoli, che pubblicheremo nel corso del 2008. Questo in particolare, è interamente dedicato alla descrizione del Bolero che, pur presentando numerose analogie con il

suo predecessore Mambo (dal quale eredita alcuni elementi progettuali), offre delle innovazioni tecnologiche così particolari da farne sicuramente un prodotto all'avanguardia. Scopriamo insieme tutte le caratteristiche di questo dispositivo unico nel suo genere.

Aspetti tecnici

Bolero è un dispositivo talmente integrato e sofisticato che chiamarlo semplicemente "localizzatore" è piuttosto riduttivo; esattamente come il suo predecessore, esso integra un'unità telefonica GSM/GPRS e un ricevitore GPS, >

Fig. 1 - Riepilogo dei modi di comunicazione supportati da Bolero



utilizzati per la rilevazione, la gestione e la trasmissione in remoto delle coordinate geografiche di localizzazione. Per fare ciò il sistema si avvale di un innovativo modulo GSM/GPRS Quad-Band di produzione Telit, modulo che può operare sulle frequenze GSM 850/900/1800/1900 MHz. Oltre alla sezione telefonica e al ricevitore GPS, il Bolero integra altre due tecnologie di trasmissione radio e precisamente:

- un'interfaccia radio operante in banda ISM a 2,4 GHz secondo il protocollo di livello fisico IEEE 802.15.4 (in grado di supportare le reti MESH ed utilizzato anche nella tecnologia ZigBee). Tale interfaccia viene utilizzata per comunicare in modo bidirezionale con le periferiche di Bolero (telecomando e espansioni di I/O);
- l'interfaccia in tecnologia Bluetooth, necessaria sia per la connessione al WORKBENCH (la suite di applicativi che, esattamente come avveniva per il Mambo, viene utilizzata per la gestione della configurazione) sia con l'auricolare per le connessioni audio.

L'aspetto del dispositivo (visibile nelle Fig. 3 e 4, che mostrano rispettivamente le dimensioni del dispo-

sitivo confrontate con una SIM, e col connettore Molex) è leggermente differente da quello del Mambo: l'accresciuto numero di funzioni hardware, infatti, ha reso necessario l'utilizzo di un contenitore di dimensioni leggermente superiori. In considerazione del target di impiego prettamente professionale, per questo dispositivo Falcom ha optato per un aspetto più sobrio rispetto a quello futuristico e ricercato del Mambo.

Quanto appena esposto fa del Bolero un dispositivo veramente particola-

re: attualmente è l'unico che integri quattro differenti tipi di interfaccia radio (GPS, GSM/GPRS, BT e MESH) che, come riepilogato nella Fig. 2, conferiscono al sistema una flessibilità di collegamento mai vista prima.

Dal punto di vista pratico, Bolero è in grado di comunicare la propria posizione inviando dei messaggi SMS, oppure mediante l'utilizzo della linea dati CSD e/o della connessione a Server protetti attraverso il protocollo TCP/IP. Esso, però, offre una

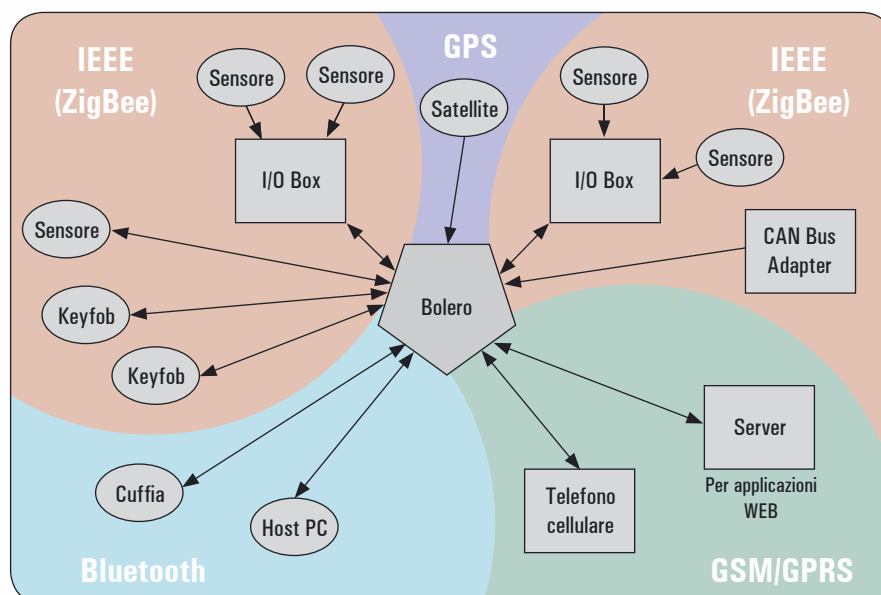


Fig. 2 - Ampia flessibilità di collegamento grazie alle quattro interfacce RF.

marcia in più: è anche in grado di inviare delle e-mail appoggiandosi al protocollo SMTP.

Espressamente studiato per soddisfare le più svariate esigenze, dal controllo delle flotte aziendali al monitoraggio dei veicoli con sensori di temperatura e/o livelli, Bolero è in grado di memorizzare ben 245.715 posizioni nella sezione di memoria Flash (960 kB) che il costruttore ha previsto come memoria di servizio per registrare dei file di log. Si pensi che, salvando per 24 ore al giorno una posizione ogni 20 secondi, sarebbe possibile memorizzare tragitti per oltre 50 giorni. Tutte le informazioni salvate possono successivamente essere scaricate da remoto e quindi eliminate, così da garantire uno spazio sufficiente alle nuove posizioni senza dover ogni volta riallocare il punto di inizio e fine del nuovo Log. A tale proposito segnaliamo un particolare curioso (improbabile ma non impossibile): una volta esaurita tutta la memoria, le nuove posizioni vanno a sovrascrivere quelle più vecchie in ordine cronologico.

Dando uno sguardo alla Fig. 1, ci rendiamo immediatamente conto di quanto il possibile ambiente di utilizzo del Bolero sia vasto: potremo comunicare via IEEE 802.15.4, BT,

GSM, e-mail, FTP e SMS agganciandoci alle due strutture di rete globalmente più diffuse (Internet e telefonia mobile); in questo modo il dispositivo sarà utilizzabile in qualsiasi parte del mondo e rintracciabile ovunque si trovi.

Se, comunque, tutto ciò non dovesse bastare ancora, Falcom ha previsto alcuni accessori che possono essere collegati via radio (ISM) al Bolero: stiamo parlando del telecomando (Keyfob) e del modulo di espansione I/O. Il telecomando (opzionale), mostrato in Fig. 3, ha tre pulsanti programmabili in base alle necessità di utilizzo; ad esempio, il tasto centrale potrebbe svolgere la funzione di SOS e avviare una chiamata antipanico o di segnalazione di avaria; uno degli altri potrebbe abilitare e disabilitare la funzione di blocco motore, funzione questa realizzabile in unione a un modulo di espansione di I/O (Fig. 11). Il terzo pulsante potrebbe attivare la sirena dell'autoveicolo, utilizzata come deterrente acustico nel caso qualcuno tenti con le minacce di rubare l'autovettura. Se ci avete seguito negli articoli dedicati al Mambo sicuramente ricorderete di quanto esso sia flessibile... Ebbene, da queste poche righe introduttive, avrete sicuramente compreso che



Fig. 3

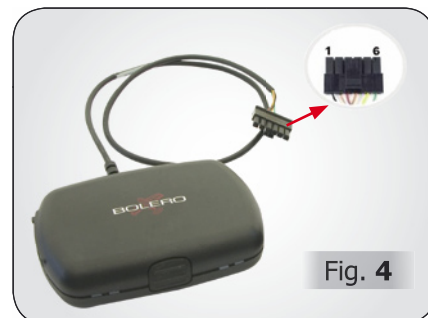


Fig. 4

il Bolero, se possibile lo è ancora di più.

Architettura interna

La Fig. 5 illustra lo schema a blocchi di Bolero: il cuore del dispositivo è rappresentato da un microprocessore ARM 7 TDMI che lavora ad una frequenza di clock di 25 MHz e che sovrintende tutte le operazioni. Ad esso fanno capo:

- il Modulo GSM, un Telit TRIUM Quad-Band;
- il ricevitore GPS, un JP13-S di >

Schema a blocchi del Bolero

Fig. 5 - Il cuore del sistema è rappresentato dal processore ARM7 TDMI a 25 MHz. Esso controlla tutte le periferiche specializzate ossia il modulo telefonico, il ricevitore GPS, le interfacce BT e IEEE 802.15.4, il sensore di movimento e il controller CAN Bus. Il blocco di alimentazione, dotato di batteria interna, alimenta separatamente i vari moduli, così da ottimizzare il consumo di energia.

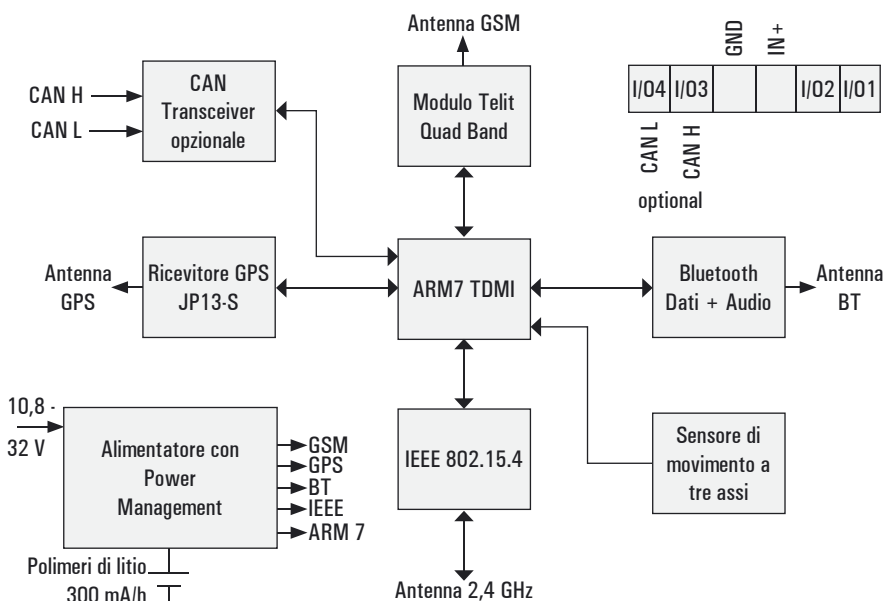


Tabella 1 - Cablaggio e funzioni di ciascun conduttore

Pin	Nome	I/O	Colore	Descrizione	Livello
1	I/O 0	I/O	Nero	Impieghi generici di gestione ingresso/uscita digitale o ingresso analogico.	Uscita digitale: open collector, 100 mA max, 10,8÷32 Vdc Ingresso analogico: fino a 32 Vdc, risoluzione a 10 bit Ingresso digitale: 10,8÷32 Vdc
2	I/O 1	I/O	Arancio	Impieghi generici di gestione ingresso/uscita digitale o ingresso analogico.	Uscita digitale: open collector, 100 mA max, 10,8÷32 Vdc Ingresso analogico: fino a 32 Vdc, risoluzione a 10 bit Ingresso digitale: 10,8÷32 Vdc
3	VCC	I	Rosso	Positivo di alimentazione. Gli ingressi NON DEVONO eccedere questo valore.	10,8÷32 Vdc positivi rispetto a massa.
4	GND		Marrone	Massa e Negativo di alimentazione	0 V. Potenziale di massa (telaio autoveicolo)
5	I/O 2	I/O	Giallo	Impieghi generici: ingresso digitale o analogico. Può essere connesso alla messa in moto del veicolo per generare gli eventi. Se il CAN Bus Controller è presente, trasporta la linea CAN H.	Ingresso digitale: 10,8÷32 Vdc Ingresso analogico: fino a 32 Vdc, risoluzione a 10 bit
6	I/O 3	I/O	Verde	Impieghi generici: ingresso digitale o analogico. Se il CAN Bus Controller è presente, trasporta la linea CAN L.	Ingresso digitale: 10,8÷32 Vdc Ingresso analogico: fino a 32 Vdc, risoluzione a 10 bit

progettazione FALCOM basato sul noto e universalmente apprezzato chip-set SIRFstarIII a 20 canali paralleli;

- il modulo Bluetooth compatibile con le classi 1.1 e 2.0, nella classica configurazione “Dati + Audio”;
- il sensore di movimento 3D che consente la gestione della funzione Motion Detection;
- il modulo radio IEEE 802.15.4 per la gestione del telecomando e delle periferiche I/O;
- l’interfaccia CAN Bus (Opzionale) per la connessione e la gestione di particolari sensori all’interno del veicolo.

Completano il dispositivo la sezione di alimentazione e “power management”, in grado di alimentare singolarmente ogni blocco, ed il connettore hardware esterno, cui fanno capo le linee di alimentazione esterna, due I/O e l’interfaccia CAN Bus opzionale.

I moduli in dettaglio

Il modulo GSM/GPRS oltre ad essere compatibile con la Fase GSM 2/2+, supporta le SIM da 3 volt, trasmette con potenza in Classe 4 (2 Watt) alle frequenze di 850/900 MHz mentre alle frequenze 1800/1900 MHz opera in Classe 1 (1 Watt). La connettività GPRS supporta la modalità multi-slot in Classe 10 il data-rate massimo è di 85,6 kbps in down-link e 42,8 kbps in up-link. È inoltre

possibile utilizzare (costi di gestione permettendo) la linea dati CSD con velocità che vanno da 2,4 a 14,4 kbps e protocollo V.110 non trasparente.

Il modulo GPS garantisce un’accuratezza della posizione compresa tra 5 e 1 m (in modalità DGPS) ed una precisione di 0,05 m/s nell’indicazione della velocità. I limiti legati alle condizioni dinamiche di utilizzo sono di 18.000 m per l’altitudine, 515 m/s per la velocità e 4 g per l’accelerazione. I tempi di acquisizione della posizione vanno da un minimo di 1 s (con partenza a caldo, hot start) ad un massimo di 42 s (con partenza a freddo, cold start). Supporta il protocollo NMEA (sentenze standard GLL, GGA, RMC, VTG, GSV, GSA) cui Falcom ha aggiunto alcune sentenze proprietarie (IOP, GSM, AREA, BIN) che, però, rispettano la sintassi ed il formato delle sentenze standard. Delle sentenze proprietarie parleremo in modo più dettagliato nel corso degli articoli successivi, quando analizzeremo la programmazione del dispositivo.

Il blocco di alimentazione è stato progettato per poter accettare in ingresso tensioni comprese tra 10,8 e 32 Volt; dispone inoltre di una batteria di backup da 3,7 volt 300 mA ai polimeri di Litio che consente l’utilizzo del Bolero anche in assenza di alimentazione (pensate, ad esempio, al taglio dei cavi della batteria durante un furto) per circa 2 ore.

Come ampiamente spiegato, la maggior parte dei contatti con il mondo esterno avviene via radio; è comunque presente un connettore Molex (Fig. 4) a sei contatti: due di essi (quelli centrali) vengono utilizzati per alimentare il dispositivo; gli altri quattro portano all’esterno due linee di I/O generiche e il collegamento CAN Bus (CAN L e CAN H). Quando il modulo CAN Bus opzionale non sia presente, le due linee prepo-

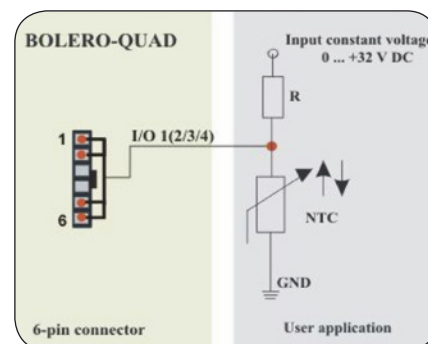


Fig. 6 - Esempio di collegamento ad un sensore termico.

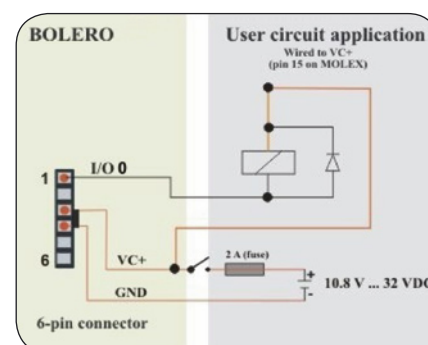


Fig. 7 - Esempio di collegamento ad un relé di potenza.

sta al bus diventano due I/O generici. I sei contatti sono stati descritti dettagliatamente nella tabella 1 alla quale rimandiamo per ulteriori approfondimenti.

Qui, invece, facciamo notare due particolari: innanzitutto, il filo di massa non è nero come sarebbe lecito aspettarsi, bensì marrone; per quanto concerne le linee di I/O, invece, esistono diverse possibilità di collegamento ed utilizzo: vediamo alcune.

Ingressi analogici

Le linee di I/O, se opportunamente programmate, possono operare come ingressi analogici ed essere collegate, ad esempio, a sensori termici, oppure a sensori di livello o di tensione: in linea di massima, l'ingresso analogico può essere utilizzato con qualsiasi tipo di sensore in grado di fornire un'uscita in tensione il cui valore dipenda dal parametro controllato. Le varie letture, operate dal Bolero in funzione della programma-

zione impostata, possono quindi dare luogo ad alcuni allarmi che vengono immediatamente inviati sfruttando uno dei mezzi di comunicazione già visti (SMS, e-mail, FTP, eccetera). La Fig. 6 mostra un tipico esempio di collegamento di un sensore analogico: la resistenza di "bias" R, collegata tra il nodo "pin/sensore" e la tensione di alimentazione, serve per alimentare il sensore qualora esso sia di tipo passivo, come la resistenza NTC usata nell'esempio. Sensori attivi (ovvero dotati di un proprio circuito di alimentazione), invece, non dovrebbero necessitare di alcuna resistenza di bias, a meno che non sia specificato sul manuale del sensore. È bene ricordare che nel Bolero i pin assumono ognuno un nome specifico e precisamente:

- Pin 1 = I/O0
- Pin 2 = I/O1
- Pin 5 = I/O2
- Pin 6 = I/O3

Uscite open collector

Le linee I/O1 e I/O2 (pin 2 e 5) possono essere programmate come uscite open collector, in grado di pilotare direttamente dei led esterni (per segnalare il verificarsi di eventi particolari) oppure dei relé. La corrente massima che le uscite sono in grado di sopportare è di 100 mA e la tensione di alimentazione dell'eventuale dispositivo collegato (relé o led che sia) non deve eccedere +VDC, che corrisponde alla tensione di alimentazione del dispositivo (10,8÷32 Vdc). Le Fig. 7 e 8 mostrano rispettivamente il modo in cui le uscite vanno collegate per pilotare un relé oppure un led.

Ingressi digitali

I pin 1, 2, 3 e 4, se opportunamente programmati, possono essere utilizzati come ingressi per generare eventi di allarme: ad esempio il tasto SOS o PANICO o altro ancora, in base alle nostre esigenze di utilizzo. Per generare l'evento, il piedino

va collegato a +VDC per almeno 1 secondo. Non si deve superare la tensione di alimentazione del dispositivo: ciò danneggerebbe l'ingresso. Tenete presente che gli ingressi non sono in grado di accendere il Bolero; essi, però, sono in grado di rilevare lo stato di alcuni parametri sotto controllo, oppure di avviare alcune operazioni a richiesta (start/stop della registrazione delle posizioni, eccetera). La Fig. 9 mostra lo schema applicativo per il controllo di un pulsante (o di qualsiasi dispositivo che comunichi un cambio di stato tramite un contatto) mentre la Fig. 10 mostra un esempio di collegamento "sotto quadro": in questo modo il dispositivo rileva l'avviamento del veicolo controllando il cambio di stato del contatto sotto chiave del blocco di accensione.

Led, pulsanti e SIM CARD.

Nella configurazione base (Fig. 12) il Bolero dispone di un solo pulsante e quattro led. Il dispositivo non ➤

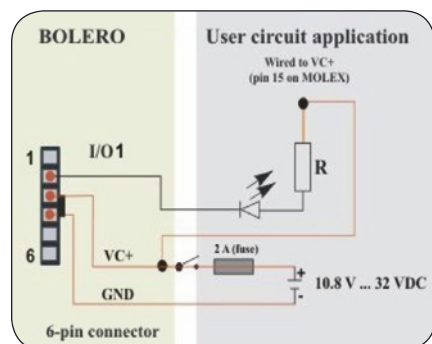


Fig. 8 - Esempio di collegamento ad un led.

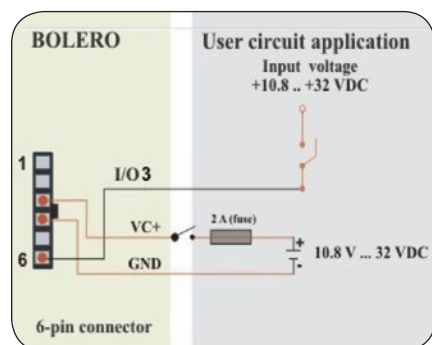


Fig. 9 - Esempio di collegamento ad un pulsante.

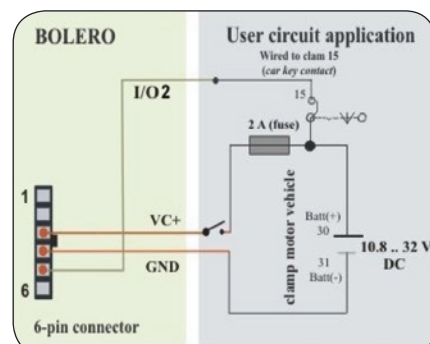


Fig. 10 - Esempio di collegamento alla messa in moto del veicolo.



Fig. 11 - I/O Box: consente di aggiungere ingressi e uscite.

si accende automaticamente una volta collegata l'alimentazione: per accenderlo bisogna mantenere premuto il tasto ON-OFF per almeno cinque secondi. Il riconoscimento del comando viene confermato dall'emissione di un tono. Prima di accenderlo, però, è necessario inserire la carta SIM nell'apposito alloggiamento laterale (Fig. 14).

Come il Mambo, anche Bolero supporta diversi eventi (liberamente programmabili) associati al tasto ON-OFF e basati sulla durata della pressione:

- Short: premuto meno di 2 s,
- Long: premuto più di 2 s,
- Double: premuto due volte entro 500 ms.

Anche per i Led vale la libera programmazione in quanto essi possono essere associati a qualsiasi evento si voglia evidenziare.

Relativamente alla gestione del pulsante e dei led, quanto detto per l'unità base vale anche per i led e i pulsanti del telecomando Bidirezionale (Keyfob, visibile in Fig. 11) e delle eventuali periferiche associate (espansione I/O, auricolare Bluetooth).

Note conclusive

Come avrete compreso leggendo queste pagine, il Bolero è un dispositivo molto flessibile che, se opportunamente programmato, si presta a molteplici applicazioni. La programmazione, però (esattamente come per il Mambo), non è sempli-

Fig. 12 - L'unico pulsante viene utilizzato sia per l'accensione (pressione di 5 s) che per avviare particolari operazioni (pressione breve, lunga e doppia). Sono presenti quattro led configurabili.



Fig. 13 - Il telecomando dispone di tre pulsanti e tre led configurabili. Il pulsante di una eventuale cuffia BT viene visto sempre come quarto pulsante.

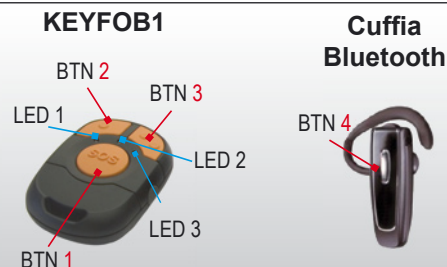


Fig. 14 - Particolare del vano porta SIM, posto sul lato sinistro (rispetto al pannello frontale) del contenitore plastico.



cissima: è necessario conoscere la sintassi dei comandi per attivare le varie funzioni, occorre conoscere a fondo l'ambiente di programmazione (Workbench) che, viste le accresciute potenzialità del Bolero, tiene conto di molti nuovi parametri, specifici per il telecomando e per la gestione delle espansioni di I/O.

Per il momento ci fermiamo qui: nel corso dei prossimi numeri presenteremo e analizzeremo l'ambiente di sviluppo e programmazione proponendo subito dopo alcuni programmi di difficoltà e complessità crescente. Continuate a seguirci per scoprire ed imparare ad utilizzare questo nuovo sistema di localizzazione.

per il

MATERIALE

Il sistema di localizzazione satellitare GPS/GSM/GPRS Bluetooth Bolero (cod. Bolero) e il radiocomando Keyfob (cod. Keyfob) costano rispettivamente 490,00 e 92,00 Euro, IVA inclusa. La confezione del Bolero comprende, oltre al localizzatore, tutto il necessario per il cablaggio nonché il CD con la documentazione ed il firmware. La confezione del Keyfob, invece, include anche un laccio per il trasporto, una staffa per il fissaggio e le batterie.

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

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

Radiomodem USB 434/868 MHz

di *Cristiano Ruggeri*

Di dimensioni molto contenute, è basato sul nuovo modulo Aurel XTR-7020A, che permette di realizzare piccole reti wireless multipunto con copertura di 300 metri circa. La frequenza di trasmissione, la potenza irradiata e la velocità di comunicazione possono essere definite tramite un set di comandi AT. Del modulo, sono disponibili due versioni pin-to-pin compatibili operanti a 868 e 434 MHz.

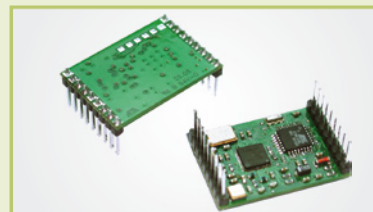
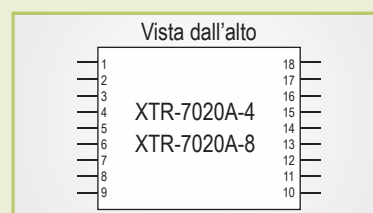
Chi ci segue da tempo sa che il mondo della connettività wireless è uno degli argomenti che seguiamo con maggior interesse, certi che questo interesse sia condiviso anche dai nostri lettori. In passato abbiamo pubblicato molti articoli dedicati a progetti di radiomodem di varia natura, talvolta basati su moduli wireless Aurel (come il progetto FT449, pubblicato sulla rivista 72), talvolta su modem GSM (progetto TDG37, rivista 111). Con queste premesse, sarebbe lecito chiedersi, quindi, per quale motivo abbiamo sentito la necessità di pubblicare il progetto di un altro radiomodem... Bene, la risposta è molto sem-

plice: perché questo progetto, basato sul nuovo modulo Aurel XTR-7020A, è qualcosa di completamente differente rispetto a quanto visto fino ad oggi. L'XTR-7020A, infatti, è un nuovissimo ricetrasmittente bidirezionale, in grado di operare su sette o dieci canali (a seconda della frequenza di trasmissione) definibili dall'utente, con potenza RF configurabile da -8 a +10 dBm (a salti di 6 dB) e gestibile, tramite un set ridotto di comandi AT, attraverso una porta seriale RS232 a livello TTL. La velocità di quest'ultima, inoltre, è configurabile sia tramite due dip-switch esterni che con un comando AT specifico. ➤

Il modulo XTR-7020A

Caratteristiche	XTR-7020A-4	XTR-7020A-8	Unità
Tensione di alimentazione	2,7 ÷ 3,6	2,7 ÷ 3,6	Vdc
Corrente consumata (RX mode)	26	26	mA
Corrente consumata (Tx mode @ -8 dBm)	20	20	mA
Corrente consumata (Tx mode @ 10 dBm)	31	31	mA
Corrente consumata (Stand-by mode)	8 ÷ 10	18 ÷ 20	μA
Tipo di modulazione	FSK	FSK	-
Sensibilità in Rx (velocità RF @ 115200 bps)	-100	-100	dBm
Potenza di trasmissione (Tx)	-8 ÷ +10	-8 ÷ +10	dBm
Bit rate seriale	9,6 · 19,2 · 38,4 57,6 · 115,2	9,6 · 19,2 · 38,4 57,6 · 115,2	kbps
Outdoor range	300	300	m
Frequenza di lavoro	434	868	MHz
Canali/canalizzazione	10 / 153,6	7 / 153,6	-/kHz

Pin	Nome	Descrizione
1, 3	RF GND	Connessione con il piano di massa della sezione RF.
2	ANT	Connessione dell'antenna a 50 ohm di impedenza.
4, 5, 6, 7, 8	-	Nella versione attuale non vengono usati.
9, 10, 18	GND	Connessione al piano di massa.
11, 15	SP1, SP2	Selezione (a modem spento) velocità seriale / modo test.
12	RSRX	Uscita (0 ÷ 3 V) seriale TTL-RS232: 1 start, 8 dati, 1 stop. Enable RS485. A 1 logico in presenza di dati su RSRX.
13	485EN/CTS	Nella versione a 868 MHz può essere usato come CTS, a seconda del valore dei registri S6 e S7.
14	RSTX	Ingresso (0 ÷ 3 V) seriale TTL-RS232: 1 start, 8 dati, 1 stop.
16	PWRDN	Attivazione modo a basso consumo: attivo a 1 logico.
17	Vcc	Alimentazione del modulo: 2,7 ÷ 3,6 Vdc.



Il modulo XTR-7020A è un radiomodem bidirezionale, prodotto da Aurel (www.aurelwireless.com) e disponibile nelle versioni a 434 (7020A-4) e 868 (7020A-8) MHz, completamente compatibili sia come pin-out che come comandi software; sono, inoltre, pin-to-pin compatibili con il modulo XTR-903A-4. Il dispositivo comunica con qualsiasi tipo di microcontrollore attraverso una seriale bidirezionale a livello TTL (0÷3V) e invia i dati via radio modulando la portante RF in FSK. La velocità della seriale può essere variata da 9,6 a 115,2 kbps sia tramite due pin (SP1 e SP2) che tramite un registro software (S5). È configurabile con un set di comandi AT che permettono di scegliere il livello di potenza RF, il canale di trasmissione e differenti modi di funzionamento.

Questo nuovo dispositivo, grazie al microprocessore e al firmware di cui dispone, per funzionare necessita di pochi componenti esterni: se proseguite nella lettura, scoprirete quanto l'XTR-7020A semplifichi la progettazione di un radiomodem USB col quale realizzare una rete wireless locale.

Il modulo XTR-7020A

Come abbiamo anticipato poco fa, ciò che contraddistingue il radiomodem pubblicato in queste pagine è l'utilizzo del nuovo ricetrasmittitore di casa Aurel, disponibile nelle due versioni (compatibili come pin-out e set di comandi) a 434 MHz (7020A-

4) e 868 MHz (7020A-8). Parlare del progetto, quindi, corrisponde in pratica a descrivere il ricetrasmittitore, al quale abbiamo aggiunto solo un minimo di hardware esterno e un semplice software di test. Tenete presente un particolare: quando abbiamo realizzato i nostri prototipi e redatto l'articolo, avevamo a disposizione solo la versione a 434 MHz, e pertanto tutto ciò che troverete nelle pagine seguenti si riferisce al modulo XTR-7020A-4. In realtà esistono delle piccole differenze tra le due versioni, ma riguardano solo il numero di canali e di registri: la versione a 868 MHz, infatti, può operare solamente su sette canali, ma dispone di due registri (S6

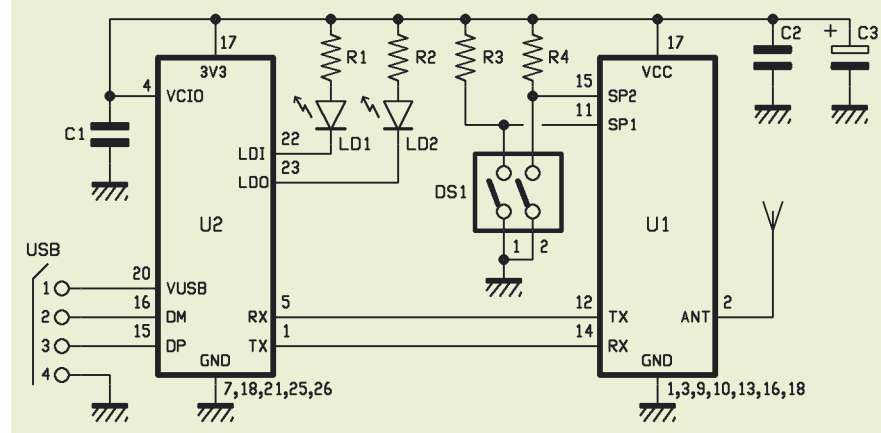
e S7) in più; nel redigere l'articolo, comunque, abbiamo tenuto conto di queste piccole differenze ed abbiamo fatto in modo di accompagnare le varie illustrazioni e tabelle con i dati relativi ad entrambi i dispositivi. Entriamo ora nel dettaglio dell'argomento facendo riferimento al riquadro in questa pagina, che riepiloga le caratteristiche tecniche e mostra la pin-out del dispositivo: l'XTR-7020A-4, è un transceiver bidirezionale half-duplex (non può trasmettere e ricevere contemporaneamente) con portante radio modulata in FSK, che consente il trasferimento di dati attraverso una porta seriale RS232 a livello TTL. Quest'ultima giunge ad

un microcontrollore che controlla costantemente il flusso in transito su di essa, distinguendo i dati dai comandi (un set di comandi AT ridotto). La velocità della seriale è configurabile a 9600, 19200, 38400, 57600 e 115200 bps a seconda della programmazione di un registro firmware (S5, di cui parleremo in un capitolo dedicato al set dei comandi) e dei pin SP1 e SP2. La trasmissione dei dati avviene secondo la tecnica “store and forward”: in pratica, i dati che entrano dalla porta seriale vengono memorizzati in un buffer, incapsulati in un pacchetto corredato di campo controllo errore (checksum, CKS) e, dopo l’arrivo dell’ultimo dato, irradiati dalla sezio-

dalla velocità delle due connessioni seriali. Se si dovessero, ad esempio, inviare dei pacchetti di 64 byte (la massima lunghezza del pacchetto è di 240 byte) ad una velocità della seriale di 115200 bps, il ritardo di trasmissione risulterebbe di 21 millisecondi circa. Il canale di comunicazione, la velocità seriale, la potenza in uscita ed il modo di funzionamento della seriale (RS232 o RS485) possono essere impostato tramite dei comandi AT specifici. Sono inoltre previsti dei comandi per monitorare il grado di occupazione del canale RF e per il salvataggio della nuova configurazione. Il modulo va alimentato con una tensione compresa tra 2,7 e 3,6 Vdc,

un normale cavo di tipo AB, preleva sia le linee dati (DM e DP) che l’alimentazione a + 5 V (VUSB), riferita alla massa del computer. Grazie al regolatore di tensione interno di U2, con uscita sul pin 17 (3V3), la tensione di 5 V viene abbassata a 3,3 V, valore questo perfettamente idoneo ad alimentare il ricetrasmittitore Aurel U1. La comunicazione tra i due dispositivi avviene attraverso le linee TX-RX incrociate (il TX di un dispositivo viene collegato all’RX dell’altro). La velocità del collegamento della porta seriale del modulo U1 viene impostata tramite il dip-switch a due vie DS1 che, insieme alle resistenze di pull-up R3 e R4, definisce il livello logico presente sulle linee SP1 e SP2. I led LD1 (verde) e LD2 (rosso), controllati dalle linee LDI e LD0 del convertitore seriale-USB e protetti dalle resistenze R1 e R2, indicano rispettivamente la presenza di dati ricevuti e di dati trasmessi. I condensatori C1, C2 e C3, ripuliscono la tensione di alimentazione a 3,3 V da eventuali residui di rumore ad alta frequenza provenienti tanto dalla porta USB del PC quanto dallo stesso ricetrasmittitore U1. Completa lo schema elettrico l’antenna a stilo, accordata alla frequenza di lavoro del modulo (nel nostro caso di 434 MHz).

SCHEMA ELETTRICO



ne RF. In fase di ricezione RF, invece, i byte vengono memorizzati e trasferiti sulla seriale soltanto dopo che il checksum è stato convalidato dall’algoritmo di controllo dell’errore: se il checksum calcolato è differente da quello contenuto nel pacchetto dati ricevuto via radio, si è in presenza di un errore di trasmissione e pertanto il pacchetto appena ricevuto viene rigettato. Questo sistema garantisce il trasferimento di dati senza errori. Visto il modo complesso (e sicuro!) in cui i dati vengono gestiti, esiste un breve ritardo tra la porta di ingresso (trasmettitore) e la porta d’uscita (ricevitore): tale ritardo dipende sia dalla lunghezza dei pacchetti inviati sia

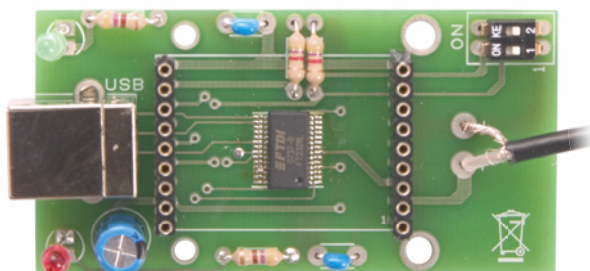
ha un consumo di corrente compreso tra 8 μ A (in modalità power-down, attivata portando a livello logico “0” il pin 16 PWRDN) e 31 mA (in trasmissione alla massima potenza, pari a +10 dBm) ed ha un ingombro di soli 23 x 33 mm.

Schema elettrico

Come avevamo già anticipato, realizzare un radiomodem USB con l’XTR-7020A è molto semplice; è stato sufficiente, infatti, collegare alla sua seriale il convertitore RS232-USB U2, un FT232R di casa FTDI (già utilizzato in passato in numerosi altri progetti). L’FT232R, collegato alla porta USB del computer tramite

Realizzazione pratica

La realizzazione di questo dispositivo non è particolarmente difficile, ma necessita di particolare attenzione soprattutto per ciò che concerne la saldatura del convertitore seriale-USB U2, disponibile solo in formato SMD. Provvedete, innanzitutto, alla realizzazione del circuito stampato tramite fotoincisione: a tal proposito, potete scaricare dal nostro sito (www.elettronica.in.it) il master in scala 1:1 del circuito stampato a doppia faccia. Se non ve la sentite di realizzarlo per vostro conto (alcune piste sono molto sottili e potrebbero tagliarsi durante il processo di asportazione del



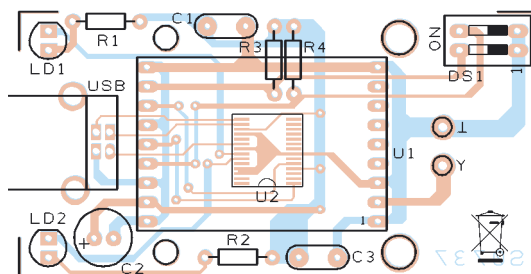
La realizzazione pratica di questo radiomodem non è particolarmente difficile; l'unico componente al quale prestare una certa attenzione è l'integrato SMD (U2, il convertitore seriale -USB). A tale scopo munitevi di un saldatore a punta fine, di fluxante in gel, di una pinzetta a becchi e di una lente di ingrandimento. Fate attenzione a non invertire i poli caldo e freddo del cavetto d'antenna.

ELENCO COMPONENTI:

R1, R2: 470 ohm
R3, R4: 4,7 kohm
C1, C2: 100 nF multistrato
C3: 220 µF 16 VL elettrolitico
U1: XTR7020A-8
U2: FT232RL
LD1: led 3 mm verde

LD2: led 3 mm rosso
DS1: Dip-switch 2 vie

Varie:
- Strip femmina 9 vie (2 pz.)
- Connettore USB-B
- Antenna 868 MHz (AG868)
- Circuito stampato (S0737)



rame), potete rivolgervi all'azienda che realizza i c.s. dei nostri prototipi (MD srl, www.mdsrl.com): il codice di questo PCB è S0737. Dopo aver realizzato il c.s e praticato tutti i fori nelle varie piazzole, munitevi di un saldatore a punta fine e potenza non superiore a 30 W, di fluxante in gel e di stagno fine (0,5 mm) di buona qualità, di pinzette a becchi e di una lampada e di una lente di ingrandimento (l'optimum sarebbe una lampada con lente incorporata). Iniziate il montaggio con il chip U2: identificato

il punto esatto dove va saldato, fate fondere un po' di stagno su due piazzole diagonalmente opposte quindi, con l'aiuto della pinzetta a becchi, posizionate U2 come mostrato nelle illustrazioni sul piano di montaggio (la tacca di riferimento va rivolta verso il basso, tenendo le piazzole del connettore USB alla vostra sinistra), avendo cura di allineare accuratamente i vari pin con le rispettive piazzole. Fate quindi fondere lo stagno depositato in precedenza, in modo che esso aderisca ai due pin del chip. Verificate gli allineamenti ed effettuate le saldature mancanti: l'uso del fluxante è indispensabile in quanto vi aiuta ad evitare cortocircuiti da eccesso di stagno tra pin adiacenti. Una volta posizionato U2, realizzate tutti i "via", ossia i collegamenti tra due piste che si trovano sui lati opposti dello stampato, con degli spezzi di filo di rame molto sottile (quello utilizzato nelle trecce del normale cavo da elettricista va benissimo). Controllate nuovamente, per essere sicuri di non aver lasciato cor-

tocircuiti e baffi di stagno sul circuito, quindi saldate le quattro resistenze, i due condensatori non polarizzati, il condensatore elettrolitico (attenzione al verso) e i due led. Relativamente a questi ultimi, se decidete di racchiudere il circuito nel contenitore da noi scelto (SC703 acquistabile presso www.futurashop.it) dovete tagliarne i reofori alla lunghezza di 11 mm, in modo che possano sporgere da due fori praticati sul coperchio del contenitore. A questo punto potete collegare l'antenna: potete utilizzare uno spezzone di cavo rigido lungo 17 cm oppure un'antenna a stilo come quella visibile nella figura qui a fianco.

Applicazioni pratiche

Questo dispositivo non nasce con lo scopo di "fare through-put ad alte prestazioni": in altre parole, non è adatto per mettere in comunicazione due o più PC o, in generale, due o più dispositivi che debbano comunicare ad alta velocità e con flusso dati continuo.

Per tale scopo esistono altri tipi di



Fig. 1 - Applicazione standard: tramite il radiomodem, due PC comunicano con due dispositivi generici equipaggiati con l'XTR-7020A.

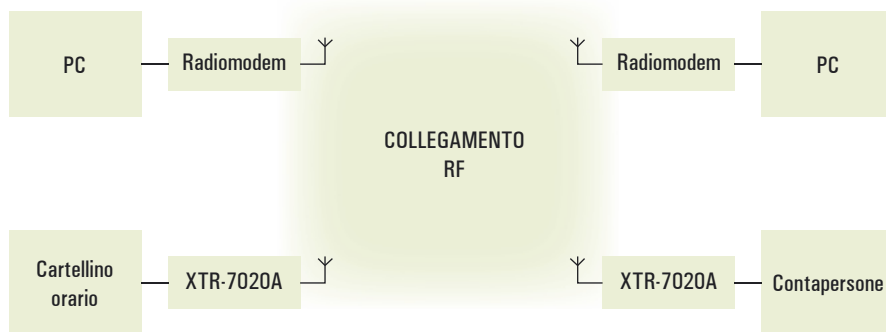


Tabella 1 - I registri

Registro	Nome	Funzione	Valori XTR-7020A-4	Valori XTR-7020A-8	R/W
S1	BANDA	Banda	0 = 433-434 MHz *	0 = 868-870 MHz *	R
S2	CANALE	Canale	0 = 433,19 MHz ¹ 1 = 433,34 MHz 2 = 433,50 MHz 3 = 433,65 MHz 4 = 433,80 MHz 5 = 433,96 MHz * 6 = 434,11 MHz 7 = 434,27 MHz 8 = 434,42 MHz 9 = 434,57 MHz ¹	0 = 868,19 MHz ¹ * 1 = 868,34 MHz 2 = 868,49 MHz 4 = 868,80 MHz 5 = 868,95 MHz 6 = 869,11 MHz 11 = 869,87 MHz ¹	R/W
S3	POTENZA	Livello RF	0 = - 8 dBm 1 = - 2 dBm 2 = + 4 dBm 3 = + 10 dBm *	0 = - 8 dBm 1 = - 2 dBm 2 = + 4 dBm 3 = + 10 dBm *	R/W
S4	RFON	Abilitazione RF	0 = Potenza RF ON * 1 = Potenza RF OFF	0 = Potenza RF ON * 1 = Potenza RF OFF	R/W
S5	SER	Banco velocità seriale	0 = banco 'slow' 1 = banco 'fast'	0 = banco 'slow' 1 = banco 'fast'	R/W
S6	CTS	Scelta sul pin 13 tra 485EN e CTS	Non applicabile	0 = 485EN* 1 = CTS	R/W
S7	SPD	Se S6 = 1, la velocità seriale dipende da S7	Non applicabile	0 = 9600* 1 = 19200	R/W
S15	VERS.	Versione FW	Variabile	Variabile	R
S16 (#)	RSSI	Livello RF ricevuto (da 0 a 9)	0 = segnale scarso 9 = segnale forte	0 = segnale scarso 9 = segnale forte	R

* = valore di default

¹ = canale non utilizzabile con potenza massima

Tabella 2 - I comandi

Comando	Nome	Funzione
+++	ESCAPE	Attiva la modalità comandi.
AT	AT	Attenzione. Inizio di ogni comando impartito.
WR	WRITE	Scrittura dei valori dei registri in EEPROM.
CC	COMMAND CLOSE	Uscita da Command Mode.

interfacce radio, come ad esempio WiFi e Bluetooth. Il nostro radiomodem, invece, è particolarmente adatto ad essere utilizzato come mezzo di comunicazione tra uno o più PC ed uno o più dispositivi di natura "non PC" come ad esempio contatori di persone, dispositivi di controllo accesso e di timbratura, cartellini orari eccetera. Fate riferimento alla Fig. 1 che schematizza un'applicazione tipica: si tratta di una piccola rete costituita da due PC, da un dispositivo per la gestione dei cartellini orari e da un contapersona (ad esempio collegato ad un tornello). Normalmente, tanto il contapersona quanto il cartellino orario, operano autonomamente memorizzando nei propri banchi di memoria tutte le informazioni relative alle timbrature e ai passaggi di persone. Periodicamente, però, uno o più PC accedono ad essi per effettuare delle operazioni: possiamo, ad esempio, ipotizzare che il primo PC (collocato in Amministrazione) acceda agli apparati per scaricare il log dei movimenti mentre il secondo PC (ad esempio nell'ufficio dell'amministratore di rete) si colleghi saltuariamente solo per effettuare la telegestione degli apparati, l'aggiornamento del firmware, la riconfigurazione, eccetera. Grazie alla canalizzazione, inoltre, possono coesistere fino a 10 reti (7 se si usa il radiomodem a 868 MHz) in grado di operare senza disturbarsi a vicenda. Tenete presente che l'esempio da noi proposto è solo uno dei tanti possibili impieghi del dispositivo: potrebbe essere utilizzato in unione a un data-logger oppure come interfaccia tra un terminale di controllo e il computer di una macchina utensile a controllo numerico, e così via.

Da quanto appena esposto, però, si deduce che il nostro radiomodem è stato realizzato appositamente per accedere, tramite un computer, ad un dispositivo qualsiasi, purché equipaggiato con un ricetrasmittitore analogo.



Impostare la velocità seriale

DIP-1	DIP-2	S5	Velocità/Modo
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

DS1

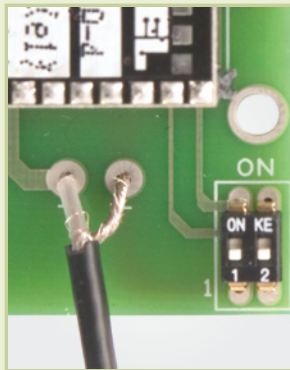
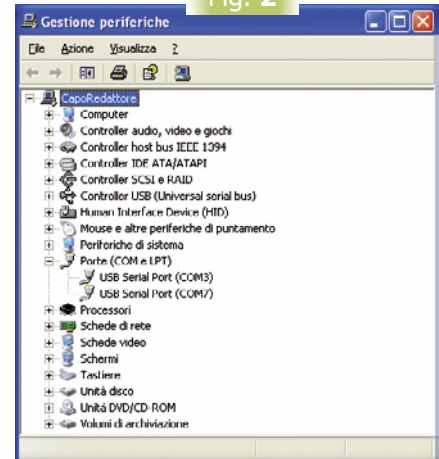


Fig. 2



Note per l'utilizzo

Dopo aver analizzato il modulo XTR-7020A-4, lo schema elettrico, i possibili campi di applicazione, ed avere realizzato il modem in pratica, siamo finalmente giunti al momento di far funzionare il dispositivo, attraverso dei semplici esercizi a scopo puramente didattico: per poter procedere, però, è necessario che abbiate a disposizione almeno due radiomodem e un PC con due porte USB libere. È inoltre necessario che abbiate installato sul PC il nostro programma "SerialControl_TEST", scaricabile gratuitamente dal sito della rivista. La messa in funzione del sistema prevede quattro fasi: la configurazione hardware e l'installazione del dispositivo, l'invio e la ricezione di alcuni dati, l'utilizzo in pratica dei comandi AT e, infine, l'esecuzione del test per verificare la copertura del segnale RF. Tenete presente che utilizzeremo solo alcuni dei comandi AT: qualora desideriate avere una visuale più ampia delle potenzialità del modulo e dei comandi, vi consigliamo di scaricare il manuale utente dal sito del produttore.

Configurazione e installazione hardware

Per prima cosa, è necessario definire la velocità della seriale del radiomodem: per far ciò, occorre impostare in modo opportuno i due interruttori del dip DS1. Nel riquadro visibile in questa pagina abbiamo riportato tut-

te le possibili configurazioni; notate due particolari:

- 1) la velocità della seriale dipende, oltre che dallo stato del dip-switch, anche dal valore (0 o 1) scritto nel registro S5, configurabile solamente attraverso i comandi AT;
- 2) se entrambi i dip sono chiusi (SP1 e SP2 a massa), il modulo XTR-7020A avvia una modalità di test in cui continua a trasmettere via radio dei dati (pseudocasuali se S5 vale 0, numerici incrementali se S5 vale 1).

La configurazione della velocità (o del modo test) deve essere effettuata a dispositivo spento in quanto la lettura delle linee SP1 e SP2 avviene solo durante l'accensione. Potete, ora, portare a ON uno dei due dip-switch, oppure lasciarli entrambi a OFF, condizione (adottata da noi) che definisce una velocità di interfaccia di 9.600 bps, sia che S5 valga 0 sia che valga 1. A questo punto i modem sono pronti per essere collegati al PC: sistemi operativi recenti, come Windows XP e Vista, sono in grado di riconoscere autonomamente il convertitore FT232R del modem; il driver di tale chip, infatti, è certificato WHQL (Windows Hardware Qualification Laboratories) e fa parte del set di driver con cui i suddetti sistemi operativi vengono forniti.

Nel caso di PC con sistemi operativi un po' datati (Win 2000, Win 98), però, può capitare che, durante l'installazione, il computer riconosca il

dispositivo ma non riesca a trovare il driver, in quanto non presente nella serie dei driver di sistema preinstallati. In questi casi è possibile scaricare dal sito di FTDI (www.ftdichip.com) i driver più recenti (V2.02.04, quando è stato redatto l'articolo) prima di collegare il primo radiomodem ad una porta USB. Dopo aver eseguito quanto appena descritto, potrete collegare il primo radiomodem al PC: questi rileva immediatamente il convertitore seriale FTDI ed avvia l'installazione del driver seguendo la classica procedura guidata di Windows.

Nel caso in cui il driver FTDI sia già presente nel sistema, la procedura termina (senza alcun ulteriore vostro intervento) con l'installazione del dispositivo e la creazione di una COM PORT virtuale, visibile accedendo alla finestra relativa alle risorse hardware del computer.

Qualora, invece, il driver non sia presente, il sistema operativo mostra la classica finestra di selezione guidata della sorgente del driver (CD, Floppy, percorso, Internet), nella quale dovete specificare il percorso (path) verso la cartella in cui avete salvato e decompresso il driver. Anche in questo caso, la procedura termina con la creazione di una COM PORT virtuale.

A prescindere dalle modalità di configurazione del primo modem, l'installazione del secondo avviene automaticamente e termina con la creazione

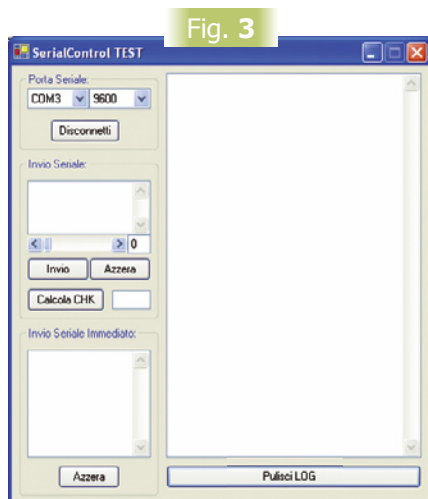


Fig. 3

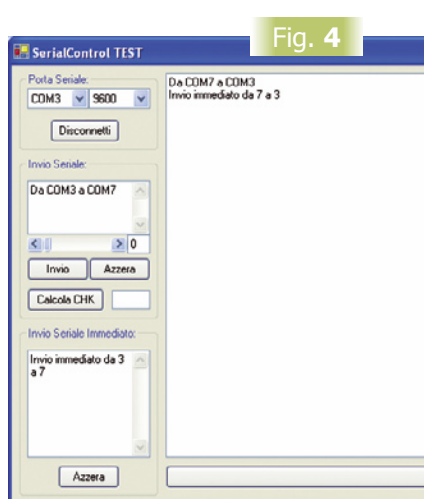


Fig. 4

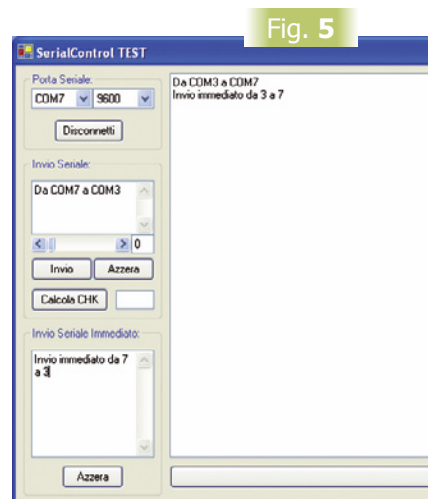


Fig. 5

di una seconda COM PORT virtuale: come visibile in Fig. 2, nel nostro caso le seriali virtuali sono le COM3 e COM7. E' ora possibile avviare due sessioni del programma SerialControl_TEST.

SerialControl_TEST: l'invio dei dati

Iniziamo, ora, ad interagire con i nostri radiomodem: per prima cosa occorre avviare due sessioni del programma, una per ogni radiomodem. La Fig. 3 mostra la schermata principale del programma di test: partendo dal lato in alto a sinistra, trovate la sezione relativa alla configurazione/apertura della porta seriale (COM3, 9.600 bps). Immediatamente sotto di essa, trovate le due sezioni *Invio Seriale* e *Invio Seriale Immediato*: la prima consente di scrivere del testo nella *text box* (ad esempio i comandi AT) e di inviarlo solo in seguito alla pressione del tasto *Invio*; la seconda, invece, invia immediatamente alla

seriale qualsiasi carattere digitato nella relativa *text box*. Tutta la sezione di destra, infine, mostra il traffico dati presente sulla seriale.

Immediatamente dopo l'avvio delle due sessioni di programma, i modem sono già pronti per comunicare; fate riferimento alle Fig. 4 e 5 che mostrano lo scambio di dati tra le due unità, effettuato sia con il comando di invio a richiesta che di invio immediato. Come è lecito aspettarsi, i caratteri inviati dalla COM3 (Fig. 4) appaiono sulla COM7 (Fig. 5) e viceversa. Potete effettuare alcune prove, ad esempio variando la velocità seriale su entrambi i modem o solamente su uno di essi: in questo caso verificherete che i dati transitino da una seriale all'altra anche se queste sono impostate con velocità differenti. È comunque necessario che tanto il modem quanto la relativa interfaccia grafica di test operino alla stessa velocità. Volendo inviare dei pacchetti dati preconfezionati, ricordatevi che

la dimensione massima dei pacchetti è di 240 byte.

SerialControl_TEST: utilizziamo i comandi AT

Quanto visto nel paragrafo precedente accade con due moduli XTR-7020A, secondo la configurazione predefinita; in pratica, però, tale configurazione deve spesso essere modificata. Per fare ciò, si utilizzano i comandi AT (riepilogati nelle Tabelle 1 e 2): nel nostro esempio, modifichiamo il registro S2 per operare sul quinto canale (frequenza di 433,8 MHz), riduciamo la potenza RF a -8 dBm variando il registro S3 e salviamo in memoria la nuova configurazione tramite il comando WR. Come illustrato nelle Fig. 6 e 7 (identiche per procedura), i comandi vengono impartiti tramite la finestra di invio immediato, mentre utilizziamo l'invio su richiesta solo per trasmettere da un modem all'altro i pacchetti dati preconfezionati "da 3 a 7" e "da 7 a >

per il

MATERIALE

I componenti necessari alla realizzazione di questo progetto possono essere acquistati presso qualsiasi rivenditore di materiale elettronico. Il master del circuito stampato può essere scaricato gratuitamente dal sito della rivista (www.elettronica.in.it). I moduli XTR-7020A-4 e XTR7020A-8 costano entrambi 38,00 Euro cadauno mentre il chip FT232RL costa 7,00 Euro. Tutti i prezzi si intendono IVA compresa.

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

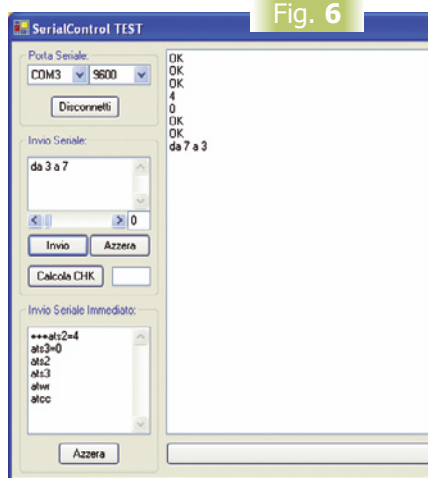


Fig. 6

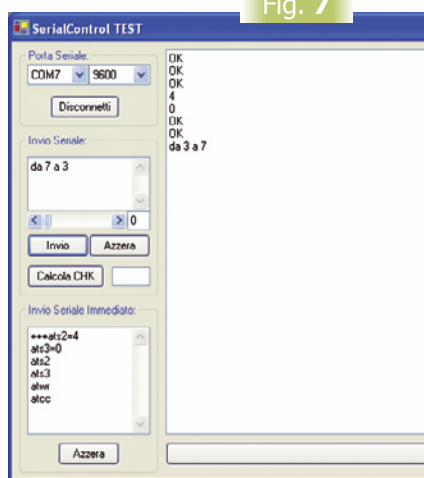


Fig. 7

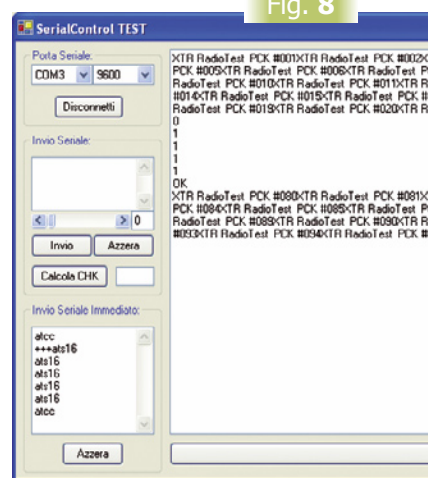


Fig. 8

3". Occorre, innanzitutto, attivare la modalità comando con la sequenza di ESCAPE +++, alla quale il radiomodem risponde OK. Vengono quindi inviati i comandi AT52=4 (imposta canale di trasmissione numero 5) e AT53=0 (potenza RF di -8 dBm), entrambi terminati con la pressione del tasto "ENTER" (CR-LF). Il modem conferma la ricezione di un comando valido con il messaggio OK; se per ipotesi avessimo inviato un comando sbagliato, come risposta avremmo ottenuto ERROR. Verifichiamo ora che i registri siano stati effettivamente modificati, inviando i due comandi AT52<CR-LF> e AT53<CR-LF>: il modem risponde con "4" al primo e "0" al secondo (esattamente secondo quanto impostato in precedenza). Salviamo la nuova configurazione con il comando ATWR<CR-LF> (il modem risponde OK) ed usciamo dalla modalità comandi inviando l'ultimo comando ATCC<CR-LF>

(risposta OK) oppure non inviando alcun comando per almeno quattro secondi. Non rimane che provare la nuova configurazione facendo clic sui pulsanti per l'invio dei due messaggi di prova e controllando che i dati vengano trasferiti nelle rispettive caselle di log.

SerialControl TEST: controllare la portante RF

Quando si realizza una rete wireless in pratica, si ha spesso il dubbio che la portante RF delle varie unità radio non riesca a garantire la copertura necessaria.

Bene, l'XTR-7020A dispone di una modalità di test che obbliga l'apparato a trasmettere continuamente dei dati casuali o dei pacchetti di dati tutti uguali, identificabili con un valore numerico crescente: in entrambi i casi, se la copertura radio è sufficiente, tali dati vengono rilevati da tutti i radiomodem operanti nella stessa

area e sullo stesso canale. Per attivare la modalità test con invio di pacchetti dati sul modem collegato alla COM7 entrate in modalità comandi tramite la sequenza di ESCAPE (+++) quindi inviate AT55=1<CR-LF> (fate riferimento al riquadro relativo all'impostazione dei dip-switch) e ATWR<CR-LF> per salvare la nuova configurazione. Chiudete ora il programma di test e scollegate per qualche istante il radiomodem dalla porta USB: una volta ricollegato, entro pochi secondi, questi inizia ad inviare i pacchetti dati di test che vengono rilevati dal radiomodem collegato alla COM 3 e (qualora ce ne fossero) da tutti i radiomodem raggiunti dalla portante RF. Fate riferimento alla Fig. 8, che mostra quanto ricevuto dal modem collegato alla COM 3: dopo la ricezione di alcuni dati abbiamo verificato la qualità del segnale RF con il comando AT516<CR-LF>.

4° SALONE dell'ELETTRONICA

con spazi dedicati:

- allo scambio del disco e CD
- a radio e grammofoni d'epoca
- all'elettronica in genere

5 - 6 aprile 2008

al CENTRO FIERE - VILLA POTENZA
MACERATA

orari: sabato 14,30 - 19,00
domenica 09,30 - 19,00

abbinato alla 21ª edizione



segreteria organizzativa:



0733.63.73.13 - 335.373.156 - www.servicepromotion.it • info@servicepromotion.it

Tabellone tennis, firmware e contenitore

di *Gabriele Daghetta*

Concludiamo la presentazione del visualizzatore per incontri di tennis descrivendo il firmware implementato nel microcontrollore ed occupandoci del contenitore metallico nel quale sono stati alloggiati tutti i circuiti elettronici e che è stato studiato nei minimi particolari per dare un aspetto professionale al dispositivo.

```
'ritardo 10 ms
'se pulsante 1 premuto
'consenso ad effettuare annulla
'consenso ad effettuare annulla
'vengono poste a 1 dal program
'emetti beep breve
'azzerà T
'Selezione punto/reset. Finché
'ritardo 5 ms
'Incrementa di 1 T
'la pressione lunga corrisponde

'se pulsante 1 premuto breve
'vai a gestione punto A
'se pulsante 1 premuto lungo
'esegui reset/test e controlla
'se P3 premuto
'  seleziona tipo gioco
'  se meglio3 vale0 gioco
'  cambia gioco
'  se meglio3 vale1 gioco meg
'  cambia gioco
'  vai a visualizza modalità
'  passa a modo visualizzazio
'  fine selezione modo megli
'  fine

endif
endif
endif
f p2=0 then
  delsetB=0
  delsetA=0
  gosub beep
  t=0
  while p2=0 and t<250
    pause 10
    t=t+1
  then
    punto_B
  'se pulsante 2 premuto gestisce
  'modo tennis/incremento (pallav
  'come P1
  'come P1
  'emetti beep breve conferma com
  'solo se sonoro attivato.
  'azzerà T
  'finché pulsante 2 premuto e t=
  'ritardo 10 ms
  'Incrementa di 1 T
  'pressione di 2,5 sec circa per
  'condizione pressione breve
  'vai a gestione punto B
  'se pulsante 2 premuto
```

Il mese scorso abbiamo iniziato la descrizione del nostro visualizzatore segnapunti universale in grado di indicare anche il punteggio degli incontri di tennis; ci siamo lasciati dopo aver ultimato la descrizione degli schemi elettrici e pratici e spiegato nel dettaglio il funzionamento dell'intera apparecchiatura. Vi domanderete, dunque: cosa manca ancora? In effetti qualcosa manca, perché nella prima puntata ci siamo limitati a descrivere l'elettronica, trascurando sia la descrizione del firmware implementato nel microcontrollore, sia la realizzazione pratica dell'apparecchiatura con particolare riferimento al

contenitore nel quale il tabellone è stato inserito. Tenendo ben presente di come e dove l'apparecchiatura viene utilizzata, abbiamo progettato e realizzato un contenitore robusto ma leggero, in metallo, resistente agli urti e completo di frontale in plexiglass rosso che rende l'indicazione del display ancora più leggibile. Per il nostro prototipo abbiamo utilizzato un contenitore in alluminio anodizzato nero del tipo di quelli impiegati per realizzare i cassonetti luminosi, realizzato con particolari profili tagliati a misura e fissati tra loro con una serie di semplici accessori. Per queste lavorazioni non sono richieste particolari capaci- ➤

Listato 1

```

PROGRAMMA PRINCIPALE (TENNIS)
' Il programma ha eseguito la fase di inizializzazione.
...
' Init...
...
' Init...
gosub vis_mod_set ' All'accensione, visualizza
goto visualizza_P ' modo set "meglio di" quindi
                    ' visualizza il punteggio.

'-----
main:                ' Inizio loop principale.
gosub mis_batt       ' Misura tensione batteria.

'-----
CERCA TASTO IN MODO TENNIS
' NOTA: la gestione dei pulsanti in modo incremento
' (pallavolo), qui non riportata, viene fatta in una
' routine differente: in questo modo è possibile
' modificare il sorgente della sola routine pallavolo
' per ottenere un contapunti differente, senza andare ad
' alterare le funzioni di segnapunti per tennis.
'-----

for rit_vis=0 to 200 ' Finché Rit_vis minore di 200
  pause 10           ' ritardo 10 ms.

'-----
Puls_1:
if p1=0 then
  delsetA=0
  delsetB=0

  gosub beep
  t=0
  while P1=0 and t<250 ' Seleziona punto/reset.
    ' Finché pulsante 1 premuto
    ' e T<250
    pause 5           ' ritardo 5 ms.
    t=t+1             ' Incrementa di 1 t.
  wend                ' La pressione lunga
                    ' corrisponde a circa 1,3 sec.
  if t<249 then
    gosub punto_A     ' Se pulsante 1 premuto breve
  else
    gosub reset       ' Se pulsante 1 premuto lungo
                    ' esegui reset/test e
                    ' controlla anche P3.
    if p3=0 then
      ' Se P3 premuto
      select case meglio3
        case 0        ' Se meglio3=0
          ' gioco meglio di 5.
          meglio3=1
        case 1        ' Se meglio3=1
          ' gioco meglio di 5.
          meglio3=0
      gosub vis_mod_set ' Visualizza modo.
      goto visualizza_P ' Passa a modo
                        ' visualizzazione.
    end select        ' Fine selezione modo set.
  endif
endif
endif                ' Fine funzioni pulsante 1.

'-----
Puls_2:
if p2=0 then
  delsetB=0
  delsetA=0
  gosub beep
  t=0

  while p2=0 and t<250 ' Finché pulsante 2 premuto
    ' e T<250
    pause 10          ' ritardo 10 ms.
    t=t+1             ' Incrementa di 1 t.
  wend                ' Pressione di 2,5 sec
                    ' circa per cambio modo.
  if t<249 then
    gosub punto_B     ' Va a gestione punto B.
  else
    goto mod_gioco    ' Se pulsante 2 premuto lungo
                    ' seleziona tennis/pallavolo.
  endif
endif                ' Fine gestione pulsante 2.

'-----
Puls_3:
if p3=0 then
  gosub beep
  t1=0
  while p3=0 and t1<200 ' Finché pulsante 3
    ' premuto e T<200
    pause 4           ' ritardo 4 ms.
    t1=t1+1           ' Incrementa di 1 T.
  wend                ' Pressione lunga circa 1 sec.
  if t1<200 then
    gosub vis_mod_set ' visualizza modo
                    ' meglio di 3 o di 5
    goto visualizza_P ' quindi visualizza punti.
  else
    goto annulla      ' altrimenti
                    ' va ad annullare il punto.
  endif
endif                ' Fine gestione pulsante 3.

'-----
Puls_4:
if P4=0 then
  gosub beep
  t=0
  while p4=0 and t<200 ' Finché pulsante 4 premuto
    ' e T<200
    pause 4           ' ritardo 4 ms.
    t=t+1             ' Incrementa di 1 T.
  wend                ' Aspetta finché T>=200
                    ' se T<200 (pressione breve)
  if t<200 then
    ' NOTA BENE: Le istruzioni sopra identificano la durata
    ' della pressione come negli altri casi.
    low LED_G         ' spegne led game.
    gosub vis_set     ' visualizza situazione set
    pause 3000        ' per 3 secondi.
    VA=VAtemp         ' Variabili temporali che
    VB=VBtemp         ' tengono conto del vantaggio
                    ' attuale in fatto di punti.
    goto visualizza_P ' Va a visualizzare
                    ' i punti quindi torna
                    ' al programma principale.
  else
    select case suono
      case 0          ' Se suono=0
        suono=1       ' abilita suono;
      case 1          ' se suono=1
        suono=0       ' disabilita suono.
    end select        ' Fine selezione suono.
    gosub okcom       ' Va a comando ricevuto
                    ' (emette beep di conferma).
    pause 100         ' Ritardo 100 ms;
    gosub okcom       ' Ripete comando ricevuto
                    ' (emette beep di conferma).
  endif
  if p4=0 then
    while p4=0
      wend
    endif
  endif
endif                ' Fine controllo pulsanti.

'-----
GESTIONE VISUALIZZAZIONE
...
' Da qui inizia il codice
' per la visualizzazione e
' per il conteggio dei punti.

```

tà, basterà procedere con precisione e ordine per essere certi di realizzare un dispositivo pratico, bello e perfettamente funzionante. Per il fissaggio del tabellone abbiamo utilizzato una catenella metallica facilmente aganciabile un po' dappertutto; nulla

vieta, tuttavia, di fare ricorso ad altri sistemi di fissaggio. L'impiego della catenella rende facilmente trasportabile il display che potremo così portare con noi ovunque andremo a giocare. In questo articolo ci occupiamo anche del firmware che è stato scritto

utilizzando un (ormai) classico compilatore basic della Micro Engineering Lab.

Ma iniziamo subito con la descrizione del montaggio meccanico dando per scontato di aver già cablato e testato le cinque schede previste.

Il software di gestione

Il firmware del nostro segnapunti è piuttosto complesso e lungo: quanti sono interessati potranno scaricarlo, nella versione eseguibile, dal nostro sito internet (per scaricare il codice sorgente è necessario essere abbonati alla rivista). A lato pubblichiamo una delle sezioni più significative del firmware: quella utilizzata per la gestione dei pulsanti (ovvero dei comandi in arrivo sui quattro canali del radiocomando o telecomando a filo) nella modalità "tennis". In ogni linea di codice è presente un commento che dovrebbe facilitare la comprensione del listato.

Dopo aver inizializzato tutti i registri, il programma visualizza il modo di gioco, imposta il display per visualizzare il punteggio, quindi avvia il loop principale, che per prima cosa controlla la tensione di batteria (eventualmente segnalando che la tensione è bassa). Viene quindi avviata la procedura per il controllo dei tasti.

Le prime righe riguardano il tasto P1 (avanzamento punteggio giocatore A / reset-test). Il codice rileva la durata della pressione del pulsante: se tale durata è inferiore a 1,25 secondi circa (pressione breve) viene incrementato il punteggio del giocatore A, altrimenti viene avviata la funzione di reset-test che, in pratica, equivale a uno spegnimento-riaccensione del dispositivo. In questa routine è interessante notare le variabili `delsetA`-`delsetB`, utilizzate dalla funzione di annullo dell'ultimo punto; per comprenderne lo scopo rammentate che la pressione del tasto P3 permette di annullare l'ultimo incremento di punteggio effettuato con P1 o P2. Stabilito ciò, va detto che se l'ultimo incremento ha fatto vincere un set, il microcontrollore deve annullare, oltre al punto, anche il game e il set appena aggiudicato, memorizzato e visualizzato. Le variabili sono normalmente a 0 e vengono poste ad 1 se l'aumento di punteggio operato con P1 ha comportato l'aggiudicazione di un set al concorrente; in caso di annullamento con P3, il software controlla lo stato della variabile e se la trova a zero annulla solo l'incremento, mentre qualora legga 1, elimina anche il set aggiudicato.

Per quanto riguarda la sezione inerente a P2 (incremento punteggio giocatore B / modo tennis-pallavolo) valgono le stesse considerazioni appena fatte: il programma verifica la durata della pressione di P2 e stabilisce di incrementare il punteggio del giocatore B in caso di pressione breve, o di passare alla modalità pallavolo con una pressione lunga.

Anche in questo caso le variabili `delsetA`-`delsetB` consentono l'annullamento, oltre che del punto, anche dell'aggiudicazione del set.

Anche per il pulsante P3 vale la discriminante del tempo: la pressione breve forza la visualizzazione del modo di gioco (meglio di 3 o di 5) mentre la pressione lunga annulla l'ultimo punto assegnato. Proprio in questo caso intervengono le variabili `delsetA` e `delsetB`.

Non rimane che descrivere la sezione dedicata alla gestione del pulsante P4 che, come gli altri, svolge due compiti distinti a seconda della durata della pressione.

La pressione breve fa sì che il display visualizzi lo stato del conteggio dei set, condizione che viene mantenuta per 3 secondi minimo, più tutto il tempo che rimane attivo P4. Durante questa fase, la G di game viene spenta e la visualizzazione dei punti viene temporaneamente sospesa (3 sec). La pressione lunga, invece, attiva e disattiva la segnalazione acustica. Di questa sezione di codice, notiamo le variabili `VA` e `VB`: normalmente a zero, vengono poste ad 1, una sola alla volta, quando, durante un gioco, il giocatore corrispondente passa in vantaggio.

Terminata la gestione dei pulsanti, il programma procede ed attiva le varie routine (qui non riportate) per il calcolo del punteggio e per la visualizzazione a display.

Il fissaggio delle basette

La prima cosa da fare è fissare meccanicamente tra loro le cinque basette che compongono il nostro dispositivo (una scheda di controllo più quattro display); le stesse andranno anche collegate elettricamente tra loro

predisponendo inoltre le connessioni alla batteria al pulsante ed alla presa di alimentazione.

Per questa operazione è necessario procurarsi innanzitutto due spezzoni di profilato in alluminio sagomato ad "L" da 15x30 o 15x40 mm, lun-

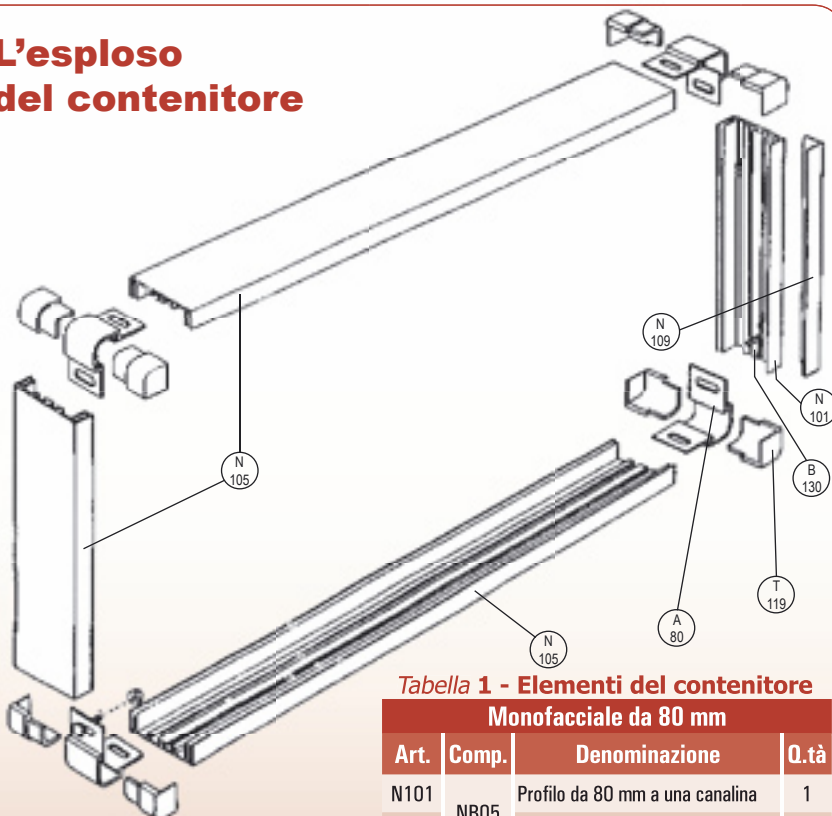
ghi ognuno circa 72 cm: a questi andranno fissate le basette. I profilati serviranno da guide di montaggio e permetteranno di costituire un insieme sufficientemente rigido che poi fisserete all'interno del contenitore.

Una volta preparati i circuiti stampati, mediante gli appositi fori li si deve fissare superiormente a uno dei profilati, sul lato corto (quello di 15 mm...) tenendoli tutti accostati, come mostrato nelle fotografie del nostro prototipo che potete vedere in queste pagine (Fig. 1). In altre parole, i quattro moduli display vanno avvicinati e accostati sul lato lungo, prevedendo in mezzo lo spazio per la basetta di controllo; quest'ultima va inserita tra i due gruppi di display, accostandola e badando che sia nel verso giusto (la scritta Tb formata dai led va posta in basso).

Dopo essere state fissate in alto, le schede vanno avvitate anche in basso, nei limiti del possibile; diciamo ciò perché quella di controllo è più bassa delle unità digit (cifre), quindi si può fissare da una sola parte: in alto o in basso. Tuttavia, per realizzare un insieme più compatto conviene centrarla in altezza e agganciarla, sfruttando i fori laterali delle unità digit, a queste ultime, ricorrendo a viti 3MA più dado e a listelli di plastica opportunamente forati. Vedete un po' voi, cosa vi riesce meglio, tenendo presente che dei ritagli di circuito stampato sono una soluzione ottimale, dato che la vetronite è robusta e facile da lavorare con lime e punte grosse.

Per il fissaggio dei circuiti stampati consigliamo di utilizzare viti 3MA con dado; in tal caso, in corrispondenza dei fori delle basette, dovrete realizzare, con un trapano e una punta da metallo, i fori da 3 mm che servono (due per ogni basetta, su ciascun profilato). Prima di montare gli stampati è comunque conveniente realizzare sulle due guide i fori (o, se preferite, delle asole) da 6 mm per fissare l'insieme al telaio del contenitore ➤

L'esploso del contenitore



Nota: i due lati lunghi e un lato corto hanno lo stesso profilo (N105). Il secondo lato corto è differente (N101): su di esso va fissato l'angolare N109, che blocca il pannello posteriore.

Tabella 1 - Elementi del contenitore

Monofacciale da 80 mm				
Art.	Comp.	Denominazione	Q.tà	
N101	NB05	Profilo da 80 mm a una canalina	1	
N109		Angolare da 25 x 20 x 1,5 mm	1	
N105		Profilo da 80 mm a due canaline	3	
A80		Angolo	4	
T119		Tappo per angolo	8	
B130		Bullone Ø 6 a testa quadra con dado	8	

trettanti corti, di uguale sezione) da unire mediante le squadrette di alluminio in dotazione e bulloni con dadi (Fig. 2, 3 e 4); le squadrette (a spigolo vivo nel nostro prototipo) sono arrotondate, nel senso che hanno gli spigoli smussati (arrotondati). Per assemblare i quattro profilati bisogna far scorrere nelle scanalature un bullone "prigioniero" per ciascuna estremità, quindi accostare a due a due i fianchi e stringere i dadi

Fig. 2



Fig. 3



tore; è necessario realizzare tanti fori quanti sono i prigionieri che farete scorrere nella guida centrale. Utilizzate una punta da 6 millimetri e realizzate i fori in maniera equidistante lungo l'intera lunghezza. Quanto alla larghezza, diciamo che, nel nostro prototipo, tutti i fori di ancoraggio al telaio stanno a circa 20 mm dallo spigolo; adottando un profilato da 15x40 mm, ciò significa a metà del lato da 40 mm, mentre se il profilato è da 15x30 mm, dovete fare i fori a 20 mm dallo spigolo (angolo). Chi desidera regolare con maggior precisione la collocazione delle schede nel senso della profondità, può realizzare vere e proprie asole o cave ad "U", anche se in tal caso serve una fresa (in alternativa, dopo aver forato bisogna lavorare di lima tonda...); il risultato è però migliore, perché si può far scorrere l'elettronica sui "prigionie-

ri" contenuti nei lati del contenitore, fino a far aderire i led al plexiglass.

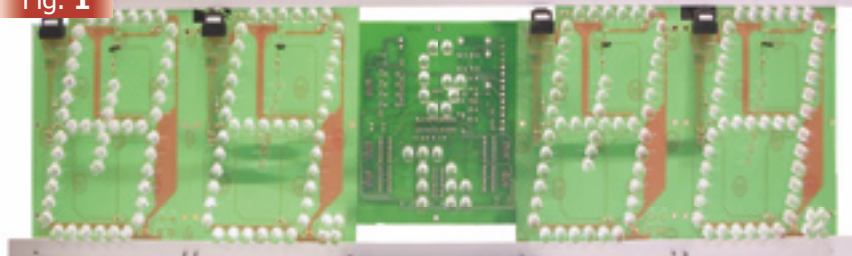
Assemblare il contenitore

Ultimata la preparazione delle guide con le cinque basette, è ora necessario predisporre il contenitore vero e proprio il cui "esploso" è visibile nel disegno in alto. Tutti i profilati sono facilmente reperibili in commercio e le misure necessarie per il taglio sono riportate nel disegno stesso. Per prima cosa montate il telaio, che consta di quattro profilati (due lunghi ed al-

in modo da fermarle. Nell'eseguire questa operazione accertatevi di aver fatto aderire perfettamente le parti. Ripetete quanto detto per tutti e quattro gli angoli ed avrete montato il telaio. La Fig. 5 mostra il primo piano di uno dei prigionieri, mentre in Fig. 6 potete notare i due prigionieri del lato sinistro del tabellone.

A questo punto va detto che prima di fissare il secondo dei profilati corti, dovete infilare il plexiglass nelle apposite scanalature; in altre parole, con il metodo appena spiegato rea-

Fig. 1



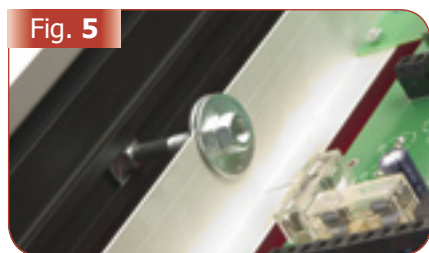
lizzate una “U”, quindi introducete il pannello di plexiglass (che costituirà il frontale del visualizzatore... Fig. 7). Sempre prima di chiudere, fate scorrere nelle scanalature centrali dei lati lunghi due o tre “prigionieri” per ciascuna scanalatura (per le teste, naturalmente...): serviranno ad agganciare il blocco contenente le schede elettroniche. Fatto anche questo, chiudete fissando l’ultimo profilato corto. Stretti i dadi, prendete gli

sare l’elettronica usate dadi adatti ai “prigionieri”, stringendoli in croce, ossia uno esterno in alto e un altro esterno (dalla parte opposta) in basso, e via di seguito; per ultimi dovete stringere i dadi interni, ossia quelli più vicini: il risultato di questa operazione è visibile nelle Fig. 6 e 7. Al fine di ottenere un fissaggio stabile, suggeriamo di posizionare i “prigionieri” e poi fermarli sulle loro guide con dei controdadi stretti alla base;

con delle fascette strette tra la guida in alluminio inferiore e le linguette di + e -, che impediscono alla batteria di scorrere longitudinalmente. Nel nostro prototipo abbiamo adottato una soluzione simile, che consiste nel bloccare longitudinalmente la batteria mediante una fascia metallica (plastificata con del termoretraibile) sagomata ad “U” e fissata alla base del telaio mediante due bulloni da 6 MA, più corti degli altri; per rendere più stabile l’insieme, la fascetta scorre in una cava appositamente realizzata nella guida inferiore in alluminio che tiene unite le schede (vedere Fig. 9).

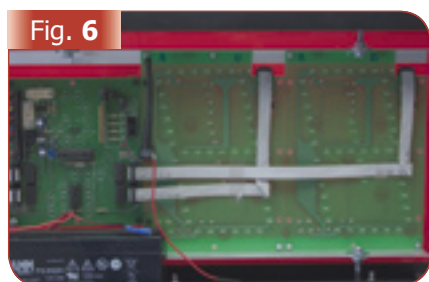
Ricordiamo che qualunque sia il metodo di fissaggio scelto, l’accumulatore deve stare in posizione standard con la base appoggiata sul lato più lungo del contenitore.

Per le prese DC di collegamento del caricabatteria e dell’eventuale alimentatore o pannello solare, vanno previsti dei fori di dimensioni adeguate. Le prese devono rimanere isolate dal metallo della cornice (alla pagina successiva vi spiegheremo il motivo) e, pertanto, vanno fissati sul contenitore attraverso un passacavo in gomma di dimensioni adeguate, in modo che la loro ghiera metallica non venga a contatto con il telaio. Riguardo l’interruttore che abbiamo scelto è del tipo anti-vandalismo ed è quindi piatto: il tasto non sporge rispetto al fianco del contenitore, quindi, oltre a non poter essere strappato, non c’è neppure il rischio che una palla da tennis o da pallavolo che lo colpisca accidentalmente possa spegnere il visualizzatore; in Fig. 10 è riportato il particolare del gruppo plug-interrut-



otto tappi in plastica e inseriteli nelle fessure dei quattro spigoli (ne vanno due per spigolo) come mostrato dall’esploso e nella Fig. 8; i tappi (qui a spigolo vivo) sono arrotondati come le squadrette di fissaggio, in modo da dare agli spigoli un profilo uniforme. Adesso introducete l’insieme dell’elettronica, posizionandolo possibilmente con i led appoggiati alla faccia interna del frontale in plexiglass; come accennato, ciò riesce meglio se avete realizzato, invece dei fori da 6 mm, delle asole o delle cave. Per fis-

fatto ciò, prima di posizionare l’elettronica avvitate un dado su ciascun prigioniero. In altre parole, collocata l’unità elettronica, mettete il controdado all’altezza giusta e serratelo con un dado dal lato opposto, dopo aver interposto delle rondelle piane. Fatto ciò, non resta che fissare la batteria e collegarla ai fili positivo e negativo in arrivo dal circuito stampato dell’unità di controllo; il fissaggio potete farlo come preferite: ad esempio mediante del silicone alla base o del biadesivo (non ottimale), ma anche



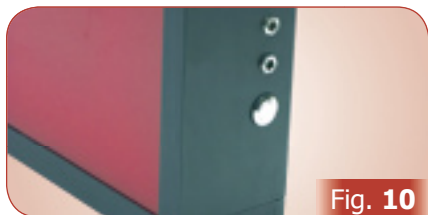


Fig. 10



Fig. 11



Fig. 12

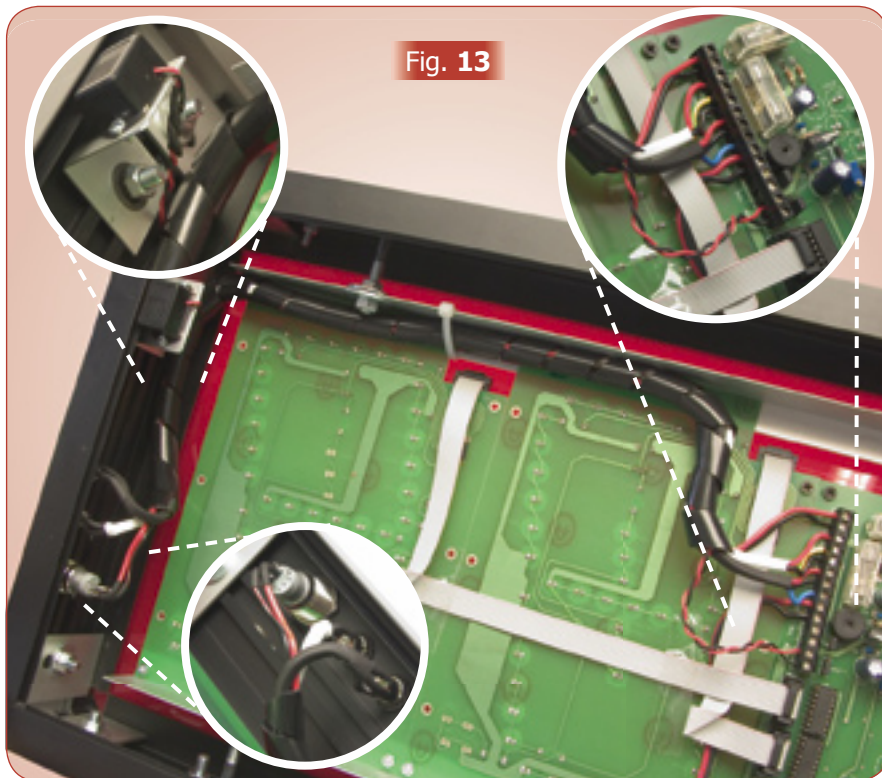


Fig. 13

tore visto dall'esterno, mentre in Fig. 11 si nota lo stesso particolare dall'interno, insieme alla squadra a 90° e ai prigionieri, usati sia per il montaggio di due lati che per il fissaggio dell'elettronica.

Per il cablaggio interno utilizzate della piattina rosso-nera da 2x0,5 mm² per collegare la batteria alla scheda di controllo (punti + e - BATT) e per connettere i morsetti SW1 di quest'ultima all'interruttore a pulsante; per le prese DC servono tre fili ciascuna, che possono essere anch'essi da 0,5 mm² di sezione. A proposito di queste prese bisogna osservare un dettaglio: la gran parte di quelle che si trovano in commercio ha il bordo esterno (vite per il fissaggio mediante l'apposito bullone piatto) elettricamente collegato al contatto S nel momento in cui la spina viene inserita; di conseguenza, anche il telaio viene a trovarsi collegato allo stesso contatto. Ora, per come sono stati previsti i collegamenti, montando direttamente sul pannello le due prese, si creerebbe un cortocircuito che vanificherebbe la funzione dei contatti, ossia l'esclusione del caricabatteria o della

batteria. Per questo motivo bisogna montare i predetti connettori isolandone almeno uno (isolare entrambi, però, è meglio) con delle rondelle isolanti o dei passacavo in gomma; in alternativa, occorre procurarsi delle prese con l'involucro elettricamente isolato dai tre contatti. In ogni caso, a cablaggio ultimato, il tabellone si presenta come in Fig. 13: notate che i cavi in uscita dalla morsettiera sono stati raggruppati e fasciati con della guaina a spirale, così da ottenere un cablaggio più ordinato. Relativamente ai plug e all'interruttore, i fili sono stati saldati e protetti con del termoretraibile. Prestate anche attenzione al particolare modo in cui il buzzer è stato fissato alla meccanica per mezzo di una squadretta a 90°: in questo modo esso può fuoriuscire dal contenitore per far sì che il suono venga percepito con più forza. Rimane, ora, il montaggio dell'antenna, il cui cavetto schermato va collegato ai morsetti ANT (Y è il conduttore centrale; l'altro è la Massa): essa può essere fissata nel modo che preferite, ma noi suggeriamo di fissarla all'interno del contenitore, ad esempio con una

piccola squadretta in plastica, come visibile in Fig. 12. Notate il filo rosso isolato, posto verticalmente in direzione opposta all'antenna, come specificato dal datasheet dell'antenna nel caso di fissaggio su supporto plastico. Per ulteriori dettagli potete fare riferimento al datasheet dell'antenna AG433 (modello T8-A Aurel). Abbiamo adottato la soluzione dell'antenna interna perché permette di raggiungere una portata di 50 metri, proteggendo contemporaneamente il delicato stilo da eventuali accidentali colpi di palla. Sistemata meccanicamente e connessa elettricamente l'elettronica, si può finalmente chiu-

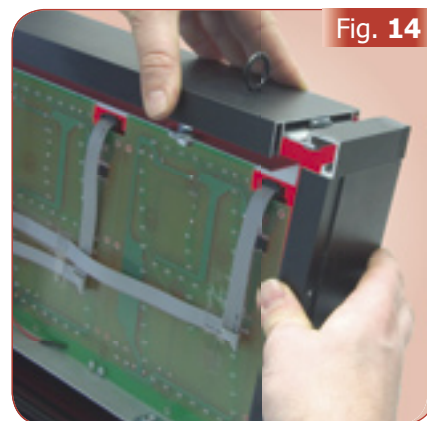


Fig. 14



Fig. 15

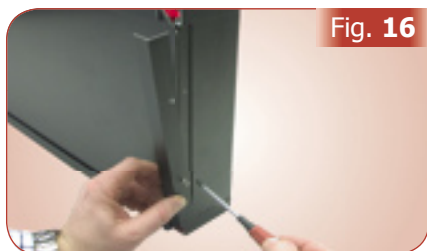


Fig. 16

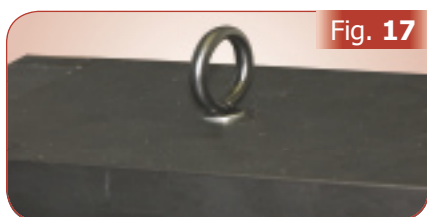


Fig. 17

Il tabellone segnapunti viene normalmente utilizzato durante gli incontri e gli allenamenti della squadra di tennis della "Futura Elettronica". Un test che è anche servito per affinare e migliorare il firmware implementato.



dere il contenitore: per prima cosa, fissate il pannello laterale ancora mancante, come visibile in Fig. 14. Prendete ora il pannello posteriore in policarbonato e fatelo scorrere nelle apposite guide dei lati lunghi, fino in fondo; prestate attenzione a un particolare: se avete deciso di fissare il buzzer come abbiamo fatto noi (Fig. 13), dovreste praticare sul pannello posteriore, come visibile in Fig. 15, una piccola apertura rettangolare così da permettere alla testa del buzzer di fuoriuscire. Fatto ciò (ossia posizionato il pannello di fondo), non resta che inserire e avvitare il profilo che fa da blocco del fondo e da chiusura

(Fig. 16). Il display segnapunti è finalmente pronto per l'uso e l'installazione. A riguardo facciamo osservare che non abbiamo previsto alcun tipo di fissaggio, per lasciare ciascuno libero di scegliere quello più adatto. In linea di massima per un montaggio a soffitto conviene appendere il visualizzatore con delle funicelle o catene; in tal caso potete comperare due ganci a vite e fissarli esattamente a metà della profondità del profilato superiore, equidistanti (circa $6 \div 10$ cm) dai lati. Se, appendendo il visualizzatore, notate che tende a stare piegato in avanti o indietro, potete bilanciarlo spostando la batteria o

aggiungendo dei contrappesi all'interno. Un altro possibile sistema di fissaggio è quello a parete: in questo caso occorrono almeno tre squadrette (ad esempio delle $60 \times 60 \times 40$ o delle $50 \times 50 \times 40$ mm, facilmente reperibili nei centri "Fai-Da-Te"), due montate sul lato lungo superiore (a 100 mm dallo spigolo) e una sul lato inferiore (al centro). In questo modo, le due squadrette superiori potranno essere fissate al muro con dei tasselli (a vite o a gancio). Non è necessario, invece, fissare al muro la squadretta inferiore che, in questo caso, serve solo per mantenere il tabellone parallelo al muro. Infine, chi desiderasse >

Tabella 1 - Alimentare il segnapunti

Alimentazione	Azione
A batteria.	Accendere il dispositivo (a batteria carica) senza collegarlo alla rete elettrica ed utilizzarlo normalmente.
A tensione di rete.	■
Ricaricare la batteria a tensione di rete. Il dispositivo si spegne.	Utilizzare un caricabatterie da 12 V, $200 \div 400$ mA. Il modello VL12-LA della Velleman (12 Vdc, 350 mA in ricarica e 20 mA in mantenimento). È necessario realizzare un semplice cavetto di adattamento da faston a presa DC.
Alimentare a tensione di rete e ricaricare la batteria a tensione di rete.	Collegare alla relativa presa DC un adattatore di tensione con uscita a 12 V 1,2 A minimo (AL08-12). Per la ricarica, utilizzare lo stesso adattatore visto alla riga precedente (ad esempio, il VL12-LA).
Ricaricare la batteria con cella solare. Il dispositivo deve essere spento.	Modo opzionale. Collegare alla morsettiera "SOL +/-" una cella solare in grado di erogare 200 mA con la tensione massima di 14 volt e lasciare il dispositivo al sole. A batteria scarica (11 V), servono almeno 8 ore di pieno sole, a patto di seguire il sole con il pannello.

Tabella 2 - I comandi per l'utilizzo in pratica

Funzione	Comando	Note
Incrementare punteggio lato sinistro.	Premere brevemente P1.	Il dispositivo emette un beep per confermare che ha ricevuto il comando. La funzione è valida sia in modo tennis che in modo pallavolo.
Incrementare punteggio lato destro.	Premere brevemente P2.	Il dispositivo emette un beep per confermare che ha ricevuto il comando. La funzione è valida sia in modo tennis che in modo pallavolo.
Visualizzare modo "al meglio di..."	Premere brevemente P3.	Il dispositivo emette un beep per confermare che ha ricevuto il comando. Questa funzione è attiva solo nel modo tennis. Il modo predefinito è al meglio di 3.
Visualizzare la situazione set.	Premere brevemente P4.	Il dispositivo emette un beep per confermare che ha ricevuto il comando. Questa funzione è attiva solo nel modo tennis.
Annullare il punto e tornare alla situazione precedente.	Mantenere premuto P3 fino a quando il dispositivo non emette un secondo beep di conferma.	Il primo beep conferma la ricezione del comando. Il secondo beep avvisa che è possibile rilasciare il pulsante. Dopo il rilascio del pulsante, viene visualizzata la situazione punti precedente (solo tennis).
Cambiare modalità di gioco tennis "al meglio di..."	Forzare un reset / test tramite P1. Durante la fase di test, mantenere premuto P3 fino a che il dispositivo non emette un suono.	Relativamente al reset / test, vedere la funzione "Effettuare reset / test sistema". Beep singolo: modo "al meglio di 5". Beep doppio: "modo al meglio di 3" (predefinito).
Passare da modo tennis a modo pallavolo e viceversa (in qualsiasi momento).	Mantenere premuto P2 fino a quando il dispositivo non esegue il reset / test.	La modalità pallavolo viene indicata con l'accensione del punto decimale della sezione di conteggio a sinistra. In modalità tennis, tale punto decimale è spento. Durante il cambio di modalità, tutti i dati vengono persi in quanto il dispositivo effettua un nuovo reset / test.
Decrementare punteggio lato sinistro.	Premere brevemente P3.	Il dispositivo emette un beep per confermare che ha ricevuto il comando. La funzione è attiva solo nel modo pallavolo.
Decrementare punteggio lato destro.	Premere brevemente P4.	Il dispositivo emette un beep per confermare che ha ricevuto il comando. La funzione è attiva solo nel modo pallavolo.
Visualizzare la versione del firmware.	Premere e mantenere premuto P1 durante la fase di accensione del dispositivo.	La versione di firmware viene visualizzata per tre secondi immediatamente dopo la fase di reset/test all'avvio.
Visualizzare la tensione di batteria.	Durante la fase di accensione del dispositivo, premere e mantenere premuto P2 fino a quando il dispositivo non emette un beep di conferma.	La tensione di batteria viene visualizzata al rilascio del pulsante; rimane visualizzata fino a quando non si ripreme brevemente P2. La taratura del valore indicato viene effettuata tramite il trimmer R7 del controller, misurando la tensione ai capi della batteria e regolando R7 in modo che l'indicazione a display corrisponda alla tensione misurata.
Attivare e disattivare l'avviso acustico.	Mantenere premuto P4 fino a quando il dispositivo non emette, a prescindere dal fatto che il suono sia attivo o no, due beep ravvicinati.	Se il suono è attivo: il primo beep conferma la ricezione del comando; i due beep successivi la disattivazione del suono. Se il suono non è attivo: il doppio beep conferma la riattivazione del suono.
Effettuare un reset / test del sistema.	In condizione di normale funzionamento, premere e mantenere premuto P1 in qualsiasi momento.	Un beep conferma il riconoscimento del comando; dopo che il display mostra tutti gli elementi luminosi accesi, è possibile rilasciare il pulsante. Un secondo beep segnala la fine del test. Nota bene: tutti i dati vengono persi.
Accoppiare il ricevitore e il trasmettitore.	Premere il tasto di programmazione P1 sul controller del segnapunti, quindi premere un pulsante qualsiasi del trasmettitore.	Alla pressione del tasto di programmazione (P1, sulla scheda controller), il led LD11 inizia a lampeggiare. Quando viene premuto un pulsante qualsiasi del trasmettitore, il led LD11 si accende a luce fissa per poi spegnersi, a segnalare l'apprendimento andato a buon fine.
Cancellare la memoria del ricevitore.	Agire sul tasto di programmazione P1 sul controller del segnapunti, come indicato nelle note.	Una prima pressione breve avvia la modalità di apprendimento (lampeggio del led LD11). Premere e mantenere premuto lo stesso tasto per almeno 5 secondi. LD11 lampeggia 5 volte per segnalare l'avvenuta cancellazione della memoria, quindi si spegne.

montare il display su un palo, potrà farlo procurandosi due collari di dimensioni adeguate e avvitandoli al centro dei profilati superiore e inferiore. Il segnapunti è ora pronto per

l'utilizzo: le Tabelle 1 e 2 riassumono le operazioni da effettuare. Prima di concludere, vogliamo segnalare un particolare: il contenitore qui descritto è stato pensato e realizzato per la-

vorare all'interno di palestre o campi da gioco coperti; in altre parole, non montatelo all'aperto perché non tiene l'acqua e l'elettronica all'interno non reagisce molto bene alla pioggia...

per il

MATERIALE

Il segnapunti è disponibile in scatola di montaggio (cod. FT720K) al prezzo di 485,00 Euro IVA compresa. Il kit comprende tutti i componenti elettronici, il microcontrollore già programmato, i circuiti stampati, la batteria, l'antenna, il ricaricatore da rete, un trasmettitore TX-4M-HCS e tutte le parti meccaniche necessarie per la costruzione del contenitore, compreso il pannello frontale in plexiglass. Ciascun trasmettitore supplementare costa 20,00 Euro.

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

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

Progettare lo stadio di alimentazione

di Matteo Bosio

Impariamo a progettare e realizzare lo stadio di alimentazione di un circuito elettronico, analizzando le esigenze in termini di tensione, corrente, dissipazione e sistemi di raffreddamento.

In questo articolo, approfondiamo le più comuni soluzioni circuitali per realizzare alimentatori lineari. Prima puntata.

Progettare un alimentatore viene spesso considerato facile, più o meno alla portata di chiunque: talvolta, e con circuiti di piccole dimensioni, quanto appena detto risulta essere vero, ma occorre tener presente che esistono molte situazioni particolari in cui l'alimentatore non può essere realizzato collegando pezzi a cassetto. Esso, infatti, deve sempre essere realizzato in base alle esigenze finali del sistema, che in alcuni casi, potrebbe necessitare di poca corrente e di tensione molto stabile, in altri il contrario, e in altri ancora di alte potenze accompagnate da basse temperature e dimensioni

contenute. Molto spesso, a livello hobbistico, i comuni 7805, 7812, LM317 (semplici, economici, affidabili e reperibili dovunque) rappresentano la soluzione ottimale, ma talvolta risultano inadeguati; da queste situazioni si può uscire prendendo due strade differenti: la prima, normalmente la più facile, porta alla ricerca (su Internet, su riviste come la nostra, su schemari) di una soluzione da copiare. La seconda porta alla progettazione vera e propria di un dispositivo che soddisfi esattamente le reali necessità e che garantisca le prestazioni richieste. Bene, in queste pagine vogliamo fornirvi le nozioni di ➤

base che vi consentiranno di progettare da soli gli stadi di alimentazione dei vostri circuiti, in funzione delle reali necessità.

Un po' di teoria

Prima di vedere come si progetta uno stadio di alimentazione, vediamo in cosa consiste questo particolare circuito elettronico.

L'alimentatore può essere definito un circuito universale, in quanto lo si trova nella maggior parte dei circuiti elettronici, dalla scheda madre di un PC alla scheda di controllo di una caldaia, fino ad un lettore DVD o un rasoio elettrico. La sua funzionalità è quella di fornire in uscita una tensione continua (con determinate caratteristiche a seconda del tipo di applicazione) partendo da una tensione alternata (generalmente i 230 V 50 Hz che troviamo nelle prese domestiche) applicata in ingresso.

Risulta chiaro quindi che il buon funzionamento di questa parte di circuito ha un'importanza fondamentale sul funzionamento corretto dell'intero sistema (pensiamo ad un micro-controllore che lavori con una tensione di alimentazione instabile, o ad una conversione AD con riferimenti di tensione instabili).

Gli alimentatori possono essere classificati in due grandi categorie: *stabilizzati* e *non stabilizzati*.

A loro volta i regolatori stabilizzati possono essere di tipo *shunt* o *serie*; questi ultimi, infine possono essere di due tipi: *lineari* o *switching*.

La principale differenza tra alimentatori stabilizzati e non stabilizzati consiste nel fatto che gli alimentatori stabilizzati mantengono stabile la tensione di uscita indipendentemente dal carico (purché compreso entro i limiti di progettazione), mentre in quelli non stabilizzati la tensione di uscita varia in funzione del carico, più precisamente diminuisce in maniera più o meno proporzionale alla corrente richiesta, con un aumento del *ripple*, fenomeno che si presenta anche negli alimentatori stabilizzati, seppur in misura minore.

Con il termine *ripple* si intende la tensione alternata residua che risulta sovrapposta alla tensione continua in uscita (Fig. 1), e rappresenta uno dei parametri che indicano la qualità di un alimentatore; per esempio, in un alimentatore la cui tensione di uscita è pari a 12 V con fattore di ripple dello 0.1%, il residuo di alternata è di 12 mV.

Un *regolatore lineare* classico è un circuito di alimentazione che si interpone tra la sorgente di ingresso e l'uscita agendo quale resistenza che suddivide la potenza elettrica per trasferirne solo l'ammontare desiderato. Una rete di retroazione controlla il valore di questa resistenza con lo scopo di mantenere la tensione in uscita costante al variare delle condizioni di carico.

Un *regolatore a commutazione*, *switching* nella terminologia inglese, invece è un circuito di alimentazione che usa una induttanza, un trasformatore oppure un condensa-

Fig. 2 - Raddrizzatore ad una semionda.

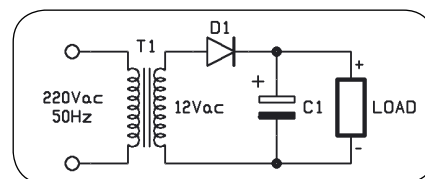
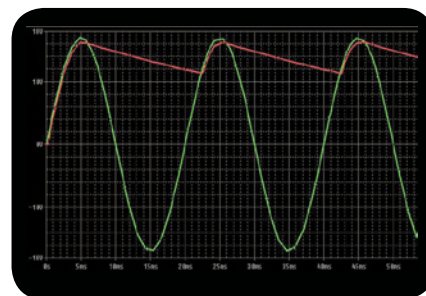


Fig. 3 - Tensioni con $C1 = 47 \mu F$.



tore, quale elemento di immagazzinamento dell'energia per compiere una funzione di trasferimento della potenza elettrica in unità discrete suddivise nel tempo. Una rete di retroazione controlla questo trasferimento di energia con lo scopo di mantenere la tensione in uscita costante al variare delle condizioni di carico.

In Tabella 1 sono riassunte le principali caratteristiche degli alimentatori stabilizzati lineari e switching: la principale differenza consiste nella dissipazione; negli alimentatori lineari infatti si sfrutta un transistor (BJT o MOSFET) in zona attiva (lineare), mentre in quelli switching un transistor (MOSFET o BJT) viene utilizzato come interruttore, o tutto aperto o tutto chiuso, con commutazioni a frequenze che vanno da pochi kHz a centinaia di kHz. Ne con-

Fig. 1 - Ripple sovrapposto ad una tensione continua.

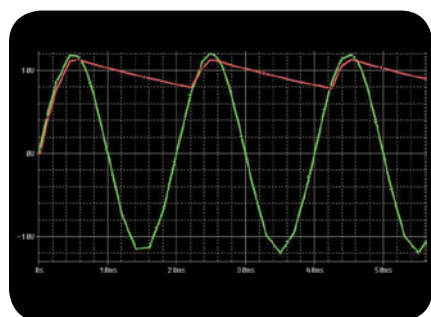


Tabella 1 - Pro e contro delle due soluzioni

	Lineari	Switching
Complessità circuitale	Bassa	Medio-alta
Tensione in uscita	Inferiore all'ingresso	Inferiore, superiore o invertita
Corrente in uscita	Anche valori elevati, ma con problemi di dissipazione	Anche valori elevati, con dissipazione contenuta
Efficienza	Medio-bassa	Alta
Rumore	Basso	Elevato (dovuto residui di commutazione)
Dimensioni	Crescono in relazione al calore da dissipare	Medio-piccole

Fig. 4 - Tensioni con $C1 = 220 \mu F$.

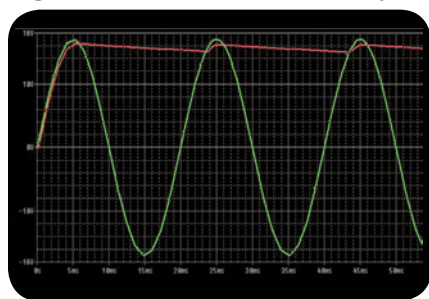
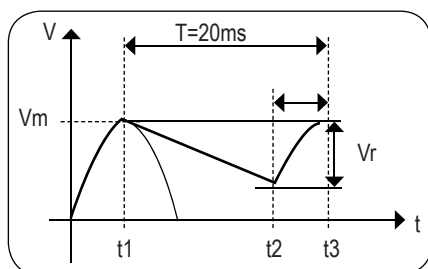


Fig. 5 - Scarica del condensatore $C1$.



Formula 1

$$v(t) = V_m \cdot e^{\frac{-(t-t_1)}{RC}}$$

Formula 2

$$\begin{aligned} V_r &= V_m - V_m \cdot e^{\frac{-(T-t_{on})}{RC}} \approx \\ &\approx V_m \left(1 - 1 + \frac{(T-t_{on})}{RC} \cdot \left(\frac{T}{T} \right) \right) \approx \\ &\approx (V_m - V_d) \cdot \left(\frac{T}{RC} \cdot \frac{(T-t_{on})}{T} \right) \end{aligned}$$

Formula 3

$$V_r = \frac{(V_m - V_d) T}{R C}$$

Formula 4

$$I_{dc} = \frac{V_m}{R}$$

Formula 5

$$V_r = I_{dc} \frac{T}{C}$$

Formula 6

$$\begin{aligned} I_{dc} &= C \frac{V_r}{T} = \\ &= C \cdot V_r \cdot f \text{ ossia } C = \frac{I_{dc}}{f \cdot V_r} \end{aligned}$$

segue che la dissipazione dei lineari è maggiore rispetto agli switching.

Alimentatori non stabilizzati

Questa categoria di circuiti può essere utilizzata per realizzare alimentatori con basse prestazioni in termini di precisione e stabilità, oppure come circuito di livellamento da anteporre agli alimentatori stabilizzati che tratteremo nel seguito dell'articolo.

Circuito raddrizzatore ad una semionda

Visibile in Fig. 2, è costituito dal trasformatore T1, dal diodo rettificatore D1 e dal condensatore C1. Esso raddrizza solo la semionda positiva, e fornisce in uscita una tensione continua con un ripple a frequenza pari alla frequenza di rete (consideriamo sempre i 50 Hz della rete elettrica italiana), come da Fig. 1.

In Fig. 3 è mostrato l'andamento della tensione su un carico da 2,2 kohm con $C1 = 47 \mu F$: si può notare un ripple di circa 3,2 V; la Fig. 4, invece, mostra la stessa tensione su un carico da 2,2 kohm ma con $C1 = 220 \mu F$: la capacità maggiore ha ridotto il ripple a 700mV. Risulta chiaro che, a parità di carico, maggiore è il valore di $C1$ minore è l'entità del ripple. Ma come si fa a dimensionare correttamente il condensatore? Occorre effettuare un'analisi approssimata del circuito in questione, facendo riferimento al ciclo di carica e scarica di $C1$ visibile in Fig. 5; si ha che $V_m = (V_{in} \times \sqrt{2}) - V_d = (12 V \times 1,414) - 0,7V = 16,268 V$ dove:

- V_m è la tensione continua ai capi di $C1$ in assenza di carico;
- V_{in} è il valore efficace della tensione alternata sul secondario del trasformatore;
- V_d è la caduta di tensione ai capi del diodo (in questo caso è stato scelto un 1N4007, che sopporta 1A e tensioni inverse di 1000 V, e ha una caduta di tensione di circa 0,7 volt).

Supponiamo che il diodo smetta di condurre nell'istante $t1$: si è in presenza di una normale scarica di un condensatore, che avviene secondo la Formula 1.

Definiamo ora:

- $t_{off} = t2 - t1$, intervallo in cui il diodo non conduce;

- $t_{on} = t3 - t2$, intervallo in cui il diodo conduce.

Ipotizziamo, quindi, che il diodo conduca per un tempo breve rispetto a tutto il periodo di 20 msec: dal punto di vista matematico, quanto sopra equivale a:

$$t_{on} \ll t_{off}$$

ossia

$$t3 - t2 \ll t2 - t1$$

Chiamiamo V_r il salto di tensione (da ora *ripple*) effettuato durante la scarica, equivalente alla tensione massima del picco diminuita della tensione raggiunta dal condensatore durante la fase in cui il diodo non conduce. Ne calcoliamo, quindi, il valore applicando la Formula 1: il risultato è visibile nella Formula 2, piuttosto complessa, ma concettualmente semplice. Essa, infatti, ci dice che il ripple è:

- inversamente proporzionale al valore della capacità,
- inversamente proporzionale alla resistenza del carico, ossia direttamente proporzionale alla corrente assorbita dal carico,
- direttamente proporzionale al tempo di off del diodo (quindi in qualche modo alla distanza tra due creste, ossia alla frequenza di rete).

Sapendo che il t_{on} è molto inferiore al periodo (di 20 msec), possiamo semplificare (con buona approssimazione) la Formula 2, ottenendo così quanto riportato in Formula 3. A questo punto, sappiamo che il ripple dipende dalla corrente assorbita dal carico, corrente descritta dalla Formula 4, per cui, per sostituzione, si ottiene la Formula 5, a conferma del fatto che maggiore è la capacità, minore è il ripple.

Dalla Formula 5, infine si può rica- ➤

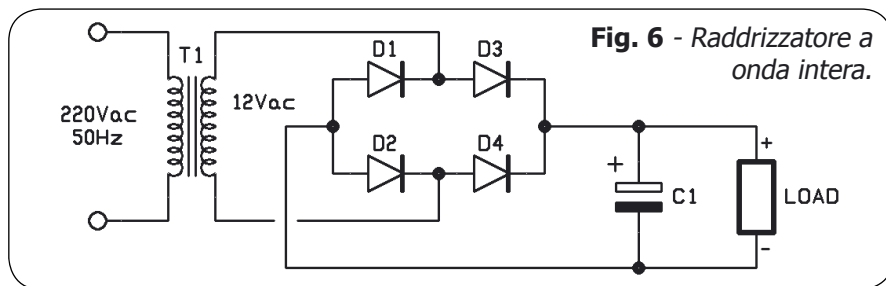


Fig. 6 - Raddrizzatore a onda intera.

Fig. 7 - Tensioni con $C1 = 47 \mu F$.

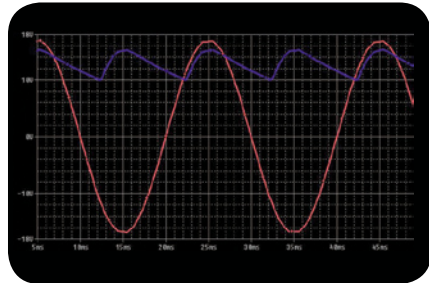
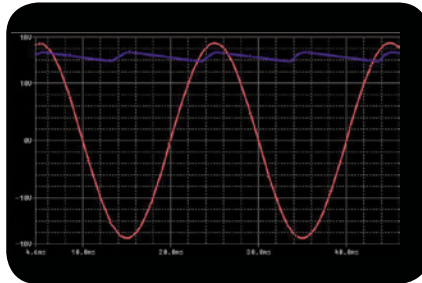


Fig. 8 - Tensioni con $C1 = 220 \mu F$.



vare la capacità C (che era esattamente il nostro obiettivo), ottenendo la Formula 6 in cui:

- C è la capacità da dimensionare (la nostra $C1$), espressa in Farad;
- I_{dc} è la corrente assorbita dal carico, espressa in ampere;
- V_r è la tensione di ripple, espressa in volt;
- f è frequenza di rete, espressa in hertz.

Tornando, quindi, ai due casi (visti come risultato ma non analizzati in precedenza) nelle Fig. 3 e 4 ed applicando le formule, otteniamo:

$I_{dc} = 16,17 \text{ V} / 2,2 \text{ kohm} = 7,35 \text{ mA}$.
Nel caso in cui $C1 = 47 \mu F$ si ottiene:

$$V_r = I_{dc} / (f \times C) = 0,00735 \text{ A} / (50 \text{ Hz} \times 0,000047 \text{ F}) = 3,12 \text{ V}.$$

mentre nel caso di $C1 = 220 \mu F$ si ottiene:

$$V_r = I_{dc} / (f \times C) = 0,00735 \text{ A} / (50 \text{ Hz} \times 0,00022 \text{ F}) = 0,668 \text{ V}.$$

Come si può notare i due risultati coincidono con i valori di Fig. 3 (circa 3,2 V) e Fig. 4 (circa 0,7 V).

Circuito raddrizzatore a onda intera

Questo tipo di circuito (visibile in Fig. 6), detto anche “raddrizzatore a

ponte di Graetz”, permette di sfruttare sia le semionde positive che quelle negative della tensione alternata in uscita dal trasformatore. Durante la semionda positiva conducono i diodi $D3$ e $D2$, mentre durante quella negativa conducono i diodi $D4$ e $D1$. In questo modo, sul condensatore collegato all’uscita del ponte sono presenti sempre e solo semionde positive. Analizziamo il circuito considerando, come nel caso del raddrizzatore a semionda, un carico da 2,2 kohm. Facendo un’analisi del circuito in questione, si possono ritenere valide gran parte delle considerazioni fatte per il raddrizzatore a semionda, a parte il fatto che la tensione massima sul carico è:

$$\begin{aligned} V_m &= (V_{in} \times \sqrt{2}) - 2V_d = \\ &= 12 \text{ V} \times 1,414 - 1,4 \text{ V} = \\ &= 15,571 \text{ V} \end{aligned}$$

dovendo sottrarre la caduta di tensione su due diodi. La frequenza, però, è il doppio della frequenza alternata in ingresso (raddrizzatore a doppia semionda). Quindi le formule viste sono ancora valide:

$$I_{dc} = V_m / R$$

per cui

$$V_r = I_{dc} \times (T/C),$$

cioè,

$$I_{dc} = C \times (V_r / T) = C \times V_r \times 2f$$

da cui si ricava

$$C = I_{dc} / (2f \times V_r).$$

Utilizzando queste formule per il circuito di Figura 6, con carico di 2,2 kohm, si ottengono i seguenti valori per la tensione di ripple:

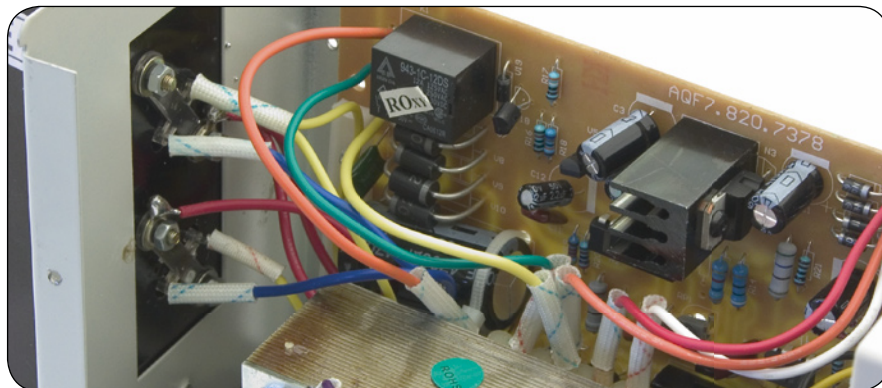
$$V_r = 1,50 \text{ V per } C1 = 47 \mu F$$

$$V_r = 0,32 \text{ V per } C1 = 220 \mu F$$

mostrati nelle Fig. 7 e 8.

Considerazione sulla variazione della tensione di uscita al variare del carico

L’inserimento del carico, come si è visto, provoca un abbassamento di tensione direttamente proporzionale alla corrente richiesta. Un buon compromesso potrebbe essere quello di dimensionare il circuito in modo che il valore di tensione con il carico inserito risulti essere circa il 90% del valore della tensione a vuoto. Quando questi alimentatori non stabilizzati vengono utilizzati come stadio di filtro per gli alimentatori stabilizzati (praticamente sempre), il condensatore di filtro può essere ridotto di circa il 50%, in quanto il circuito di regolazione di un alimentatore stabilizzato funziona come ul-



teriore filtro. Esiste, inoltre, un potenziale problema legato ai disturbi causati dalla commutazione dei diodi, che potrebbe pregiudicare, per esempio, il buon funzionamento di apparecchiature a radio-frequenza: il problema si ridimensiona a livelli tollerabili inserendo dei condensatori (di capacità dell'ordine dei 10nF) in parallelo ai diodi.

A conclusione di questa sezione dedicata agli alimentatori non stabilizzati, possiamo affermare che le caratteristiche in termini di ripple sono calcolabili solo conoscendo l'entità del carico.

Alimentatori stabilizzati lineari a componenti discreti

Cominciamo la trattazione degli stabilizzatori lineari, che sono molto utilizzati in quanto consentono di ottenere elevati fattori di stabilizzazione e residui di alternata molto bassi, a fronte però di un'elevata dissipazione di calore.

Un regolatore lineare può essere paragonato ad un resistore variabile inserito in serie al carico. Tale resistore è in grado di cambiare "dinamicamente" il proprio valore in funzione del carico e delle variazioni della tensione in ingresso. Questo a patto che le variazioni di corrente richiesta in uscita e di tensione in ingresso siano comprese entro dei limiti determinati in fase di progetto.

Circuito con diodo zener e transistor

Questo circuito, mostrato in Fig. 9, viene utilizzato con correnti modeste (meno di 500 mA).

Per un corretto dimensionamento dei valori, bisogna conoscere la corrente massima richiesta dal carico e la tensione d'uscita stabilizzata.

Dalla Fig. 9 si ricava:

$$V_{out} = V_z - V_{be}$$

dove V_z è la tensione di Zener, mentre V_{be} è la caduta di tensione tra base e emettitore del transistor, generalmente attorno a 0,7 V.

La tensione V_{in} di ingresso è quella prelevata da un precedente stadio di alimentazione non stabilizzato, come potrebbe essere quello di Fig. 2 o di Fig. 6.

Attraverso la resistenza $R1$ passano le correnti necessarie alla polarizzazione sia della base di TR1 che dello Zener. Ne consegue che la resistenza dipende anche dal guadagno del transistor. Proviamo, ora, a dimensionare un circuito in grado di fornire una corrente massima di 300 mA con una tensione di 12 V.

Supponiamo di partire dalla tensione del circuito di Fig. 2 con $C1 = 220 \mu F$, e cioè con un ripple pari a circa 0,7 V: la tensione oscilla quindi tra 15,37 V e 16,17 V.

La potenza massima che deve dissipare il transistor è dunque:

$$P = (16,17 \text{ V} - 12 \text{ V}) \times 0,3 \text{ A} = 1,25 \text{ W}$$

Il transistor adatto allo scopo è il BD139, che sopporta tensioni di collettore di 80 V e correnti (I_c) fino a 1,5 A, garantendo nel contempo un guadagno (H_{fe}) di 40 e una potenza massima dissipabile di 12,5 W. Noto l' H_{fe} (che per definizione è circa il rapporto tra la corrente di collettore I_c e la corrente di base I_b), ricaviamo la corrente di base massima necessaria:

$$I_b = I_c / H_{fe} = 300 \text{ mA} / 40 = 7,5 \text{ mA}$$

Scegliamo ora il diodo Zener, il cui valore deve essere

$$V_z = V_{out} + V_{be} = 12 \text{ V} + 0,7 \text{ V} = 12,7 \text{ V}$$

Non essendo 12,7 V un valore com-

merciale per uno Zener, si ripiega su uno da 13 V, il BZX55 ($I_z = 5 \text{ mA}$, $P = 400 \text{ mW}$). Il circuito che si ottiene ha un'uscita stabilizzata di valore 12,3 V.

Calcoliamo ora il valore di $R1$:

$$R1 = (V_{in_{min}} - V_z) / (I_{dz} + I_b) = (15,37 \text{ V} - 13 \text{ V}) / (5 \text{ mA} + 7,5 \text{ mA}) = 189,5 \text{ ohm}$$

La potenza massima che deve dissipare la resistenza è:

$$P = R \times I^2 = 189 \text{ ohm} \times (12,5 \text{ mA})^2 = 30 \text{ mW}$$

Per cui scegliamo una resistenza da 180 ohm 1/8 W.

Un vantaggio di questo circuito risulta dal fatto che il condensatore $C1$ forma con il transistor TR1 un filtro elettronico, per cui in parallelo all'uscita risulta una capacità "virtuale" C_v di valore:

$$C_v = C1 \times H_{fe}$$

Quindi con un condensatore $C1$ da 100 μF in uscita si vede una capacità di 4.000 μF .

Nel caso ci fosse bisogno di una maggiore corrente in uscita, è possibile utilizzare lo stesso circuito di Figura 9, ma con due transistor connessi in configurazione Darlington (Fig. 10), che si comportano come un unico transistor, il cui guadagno totale è però il prodotto dei guadagni dei singoli transistor:

$$I_{b2} = I_{c2} / H_{fe2}$$

da cui

$$I_{b2} = I_{c1} = I_{b1} \times H_{fe1}$$

ovvero

$$I_{c2} = H_{fe1} \times H_{fe2} \times I_{b1}$$

La tensione in uscita vale

$$V_{out} = V_z - V_{be1} - V_{be2}$$

>

Fig. 9 - Stabilizzatore con diodo Zener e transistor.

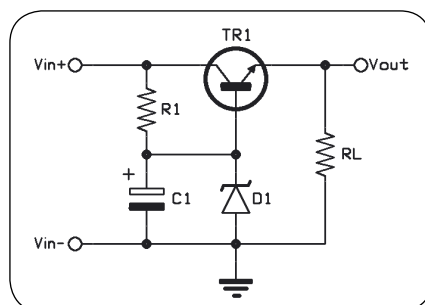


Fig. 10 - Due transistor in configurazione Darlington.

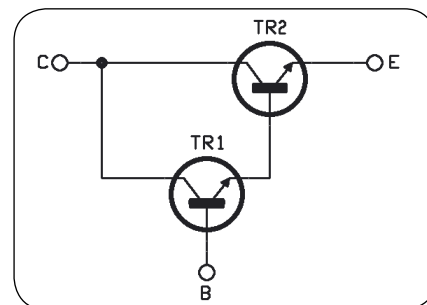
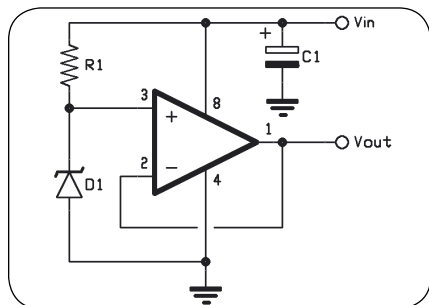


Fig. 11 - Alimentatore stabilizzato con operazionale.



Circuito con amplificatore operazionale

Il circuito di Fig. 11 è in grado di fornire una tensione molto stabile, però con una corrente bassissima (per intenderci, attorno ai 10 mA).

Il funzionamento è molto semplice, in quanto l'operazionale è configurato come inseguitore di tensione e la tensione di uscita è pari alla tensione V_z dello Zener.

Unica nota di questo circuito riguarda la tensione di zener V_z , che non deve essere inferiore ai 3 V, in quanto l'operazionale funziona ad alimentazione singola.

Un discorso diverso può essere fatto per il circuito di Fig. 12, in cui l'operazionale è alimentato con una tensione duale (in questo modo la V_z può avere qualsiasi valore inferiore alla tensione massima del-

l'operazionale), e in uscita si ha un transistor, utilizzato come amplificatore di corrente.

In questo circuito, infatti, la corrente di uscita è pari alla corrente di emettitore del transistor, per cui:

$$I_{out} = I_{e1} = I_{c1} + I_{b1} = I_{b1} \times (H_{fe1} + 1)$$

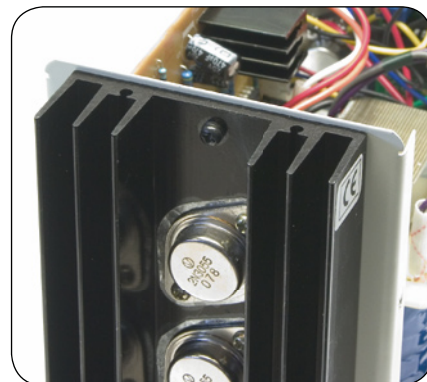
La corrente di base del transistor, però, è la corrente di uscita dell'operazionale, anche qui in configurazione di inseguitore di tensione, che al massimo può avere un valore attorno ai 10 mA. Utilizzando, per esempio, un transistor con $H_{fe} = 40$ (il nostro BD139 di prima), si possono prelevare in uscita al massimo 400 mA. Riguardo la tensione di uscita, essa vale

$$V_{out} = V_z - V_{be}.$$

Anche in questo caso è possibile aumentare la corrente erogabile, ricorrendo a una configurazione Darlington.

Alimentatori stabilizzati lineari mediante regolatori di tensione

Fino a questo punto sono stati analizzati circuiti di alimentazione realizzati con componenti discreti; esistono però dei circuiti integrati, a tre o più terminali, concepiti appositamente per realizzare degli alimentatori stabilizzati, in grado di fornire



tensioni fisse o variabili a seconda del caso. Le diverse caratteristiche dei molti tipi di circuiti integrati regolatori di tensione sono ricavabili dai datasheet facilmente reperibili sul web. Di seguito vediamo brevemente il significato di alcune caratteristiche comuni a tutti:

- **V_o – Output Voltage:** indica la tensione di uscita del regolatore.
- **ΔV_o – Line Regulation:** indica la variazione di uscita al variare della tensione di ingresso.
- **ΔV_o – Load Regulation:** indica la variazione di uscita al variare della corrente in uscita.
- **I_q – Quiescent Current:** è la corrente che circola a riposo nell'integrato (cioè in assenza di corrente assorbita dal carico), dell'ordine dei mA.
- **V_n – Output Noise Voltage:** indica l'ampiezza (in μV) dei disturbi sovrapposti alla tensione di uscita (di solito vengono indicate anche le frequenze di tali disturbi).
- **$\Delta V_{in}/\Delta V_{out}$ – Ripple Rejection:** indica (in dB) la capacità di sopprimere in uscita il residuo di alternata presente in ingresso; maggiore è il valore, migliore è l'integrato.
- **R_o – Dropout Voltage:** indica la minima differenza che deve esserci tra la tensione di ingresso e quella di uscita. Nei lineari generalmente questo valore è attorno ai 2÷3 V, anche se esistono dei componenti detti "low dropout", in cui questa differenza di tensione è attorno a 0,7 V.
- **V_{in} – Input Voltage:** indica la ten-

Fig. 12 - Alimentatore con operazionale e transistor amplificatore di corrente.

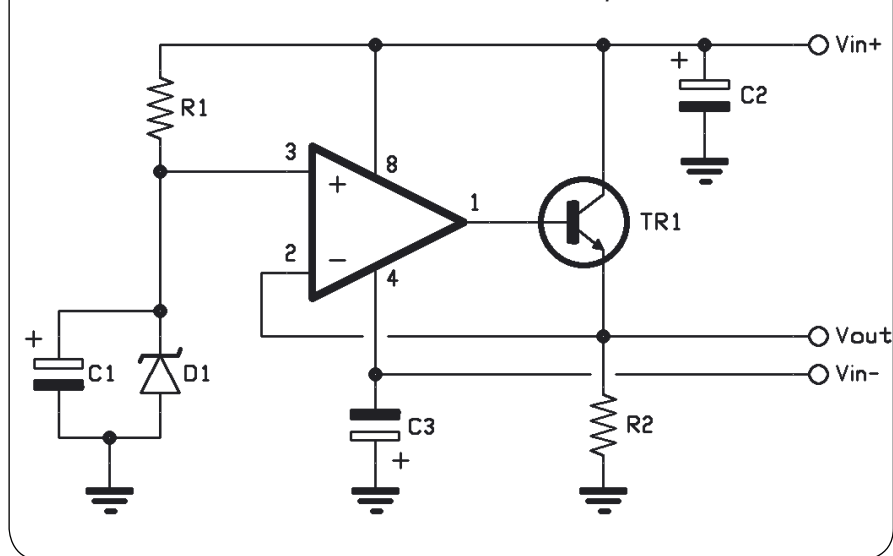


Fig. 13 - Diversi tipi di package.

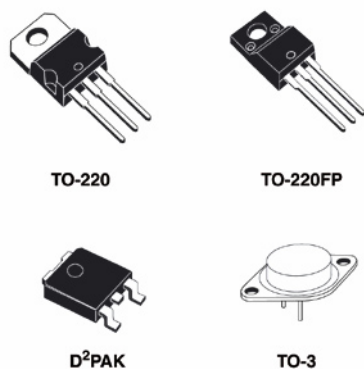
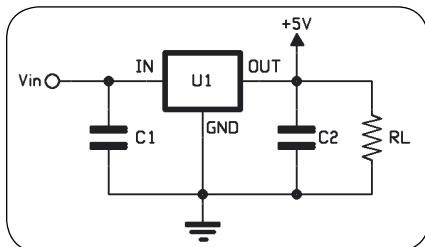
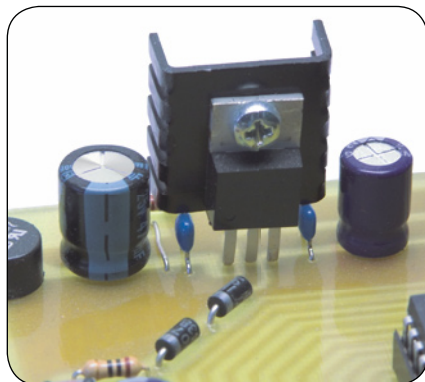


Fig. 14 - Circuito con LM7805.



sione che l'integrato accetta in ingresso.

- **Output Current:** è la massima corrente erogabile dall'integrato.
- **Short Circuit Current:** indica la corrente che circola nell'integrato in caso di corto circuito dell'uscita;
- **Power Dissipation:** indica la massima potenza dissipabile dall'integrato; solitamente questi integrati sono "internally limited", cioè limitano automaticamente la corrente erogata, al raggiungimento di una certa soglia; inoltre sono protetti termicamente, cioè la corrente viene limitata quando la temperatura di esercizio supera un



certo limite. Quando una delle due limitazioni è attiva si ha un abbassamento della tensione di uscita.

- **Maximum Voltage:** indica la massima tensione che l'integrato può accettare in ingresso senza danneggiarsi.

Regolatori a tensione fissa positiva: famiglia 78XX

La famiglia 78XX (probabilmente la più conosciuta) è costituita da una serie di regolatori di tensione a tre terminali con diverse tensioni di uscita positive fisse (5, 6, 8, 9, 10, 12, 15, 18 e 24V), utilizzabili in diversi campi di applicazione.

Esistono due sotto-serie della famiglia: la 78XX stessa, in grado di erogare fino a 1 A e disponibile nei formati TO-3 in alluminio, TO-220 e D-PAK (Fig. 13), e la famiglia 78LXX, in grado di fornire fino a 100 mA e disponibile nel formato tradizionale TO-92 e SMD SOT-89 e SO-8. Questi componenti necessitano di pochi componenti esterni (vedere Fig. 14) e ciò, probabilmente, rappresenta il loro più grande vantaggio.

Circuito con LM7805

Il circuito di Fig. 14 mostra un regolatore di tensione a 5 V, in grado di erogare fino a 1 A. Si può notare la presenza dei condensatori non polarizzati C1 e C2, di valore compreso tra 100 nF e 330 nF, il cui compito è quello di evitare che il regolatore possa auto-oscillare.

Vediamo alcune caratteristiche del regolatore LM7805:

- **Vo – Output Voltage:** minimo 4,8 V, tipico 5 V, massimo 5,2 V;
- **ΔV_o – Line Regulation:** tipico 3 mV, massimo 50 mV;
- **ΔV_o – Load Regulation:** tipico 10 mV, massimo 50 mV;
- **I_q – Quiescent Current:** massimo 8 mA;
- **V_n – Output Noise Voltage:** Tipico 40 μV ;
- **$\Delta V_{in}/\Delta V_{out}$ – Ripple rejection:**

minimo 62 dB, tipico 80 dB;

- **Ro – Dropout Voltage:** tipico 2,0 V;
- **V_{in} – Input Voltage:** tipico 7,5 V;
- **Output current:** 1,0 A;
- **Maximum voltage:** 35 V.

Supponiamo di avere in ingresso una tensione raddrizzata di 10 V con ripple di 200 mV, e di dover erogare una corrente di 150 mA.

In questo caso, ai capi del regolatore cadrebbe una tensione pari a:

$$\Delta V_r = V_{out} - V_{in} = (10 - 5) V = 5 V$$

che, con una corrente di 150 mA, comporterebbe una dissipazione di:

$$P_r = \Delta V_r \times I_{out} = 5 V \times 150 mA = 750 mW.$$

Essendo il *Ripple Rejection* pari a 80 dB, e avendo in ingresso un ripple di 200 mV, si ottiene in uscita un ripple pari a:

$$R_{out} = R_{in} / 10^4 = 200 mV / 10000 = 20 \mu V.$$

Circuito con LM7812 e transistor

Quando sono richieste correnti superiori ai limiti dell'integrato, si può utilizzare lo schema di Fig. 15, in cui la maggior parte della corrente erogata fluisce attraverso il transistor TR1, mentre in U1 scorre una corrente di pochi mA. Una cosa fondamentale è che R1 sia tale da consentire che in U1 scorra una corrente maggiore della Quiescent current dell'integrato, pari a 8 mA.

Le caratteristiche del LM7812 sono simili a quelle del LM7805, a differenza di:

- **Vo – Output Voltage:** minimo 11,5 V, tipico 12 V, massimo 12,5 V; ➤

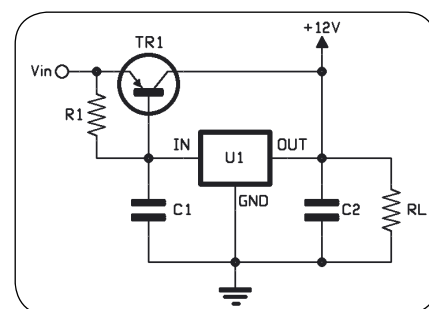
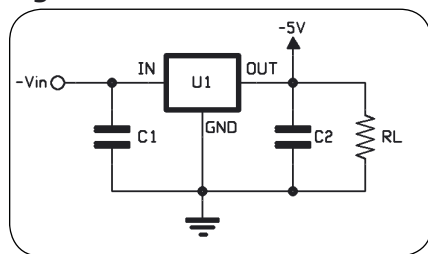


Fig. 15- Circuito con LM7812 e transistor.

Fig. 16 - Circuito con LM7905.



• V_{in} – Input Voltage: tipico 14,6 V. Vediamo il funzionamento del circuito, ipotizzando di avere in ingresso una tensione di 17 V con ripple di 300 mV, e una corrente di uscita che varia.

Se la corrente richiesta è tale che la tensione su R1 sia minore di 0,7 V, TR1 è interdetto, e U1 fornisce la corrente al carico. Appena la corrente erogata è tale da far sì che la tensione su R1 superi gli 0,7 V, TR1 entra in conduzione, e il resto della corrente passa tutta attraverso di esso. Supponiamo di volere che U1 eroghi al massimo 100 mA, e di volere in uscita una corrente di 1,5 A: bisogna dimensionare R1 in modo tale che attraverso U1 passino 100 mA e attraverso TR1 passino 1,4 A. La corrente che attraversa R1 è:

$$I_{R1} = I_{reg} + I_q + I_{b1}$$

Essendo R1 in parallelo alla giunzione base-emettitore di TR1, quando TR1 è in conduzione su R1 cadono 0,7V, quindi:

$$R1 = 0,7 \text{ V} / (I_{reg} + I_{b1} + I_q)$$

dove $I_{reg} = 100 \text{ mA}$, I_q (quiescent current) $\geq 8 \text{ mA}$ e I_{b1} è la corrente di base di TR1.

Mancano due dati: la corrente di base di TR1 e R1; supponiamo di avere una corrente di base di 50 mA, si ottiene:

$$R1 = 0,7 \text{ V} / (100 \text{ mA} + 8 \text{ mA} + 50 \text{ mA}) = 4,45 \text{ ohm}$$

che può essere approssimato al valore standard 4,7 ohm.

Come transistor scegliamo un BDX54C, un PNP con le seguenti caratteristiche:

- formato TO-220;
- V_{cbo} 100 V;
- V_{ceo} 100 V;

- I_c 5 A;
- I_b 200 mA;
- $V_{ce(sat)}$ 2 V (carico 3 A);
- P_{tot} 60 W.

In esso deve passare una corrente di 1,4 A; considerando una $V_{ce(sat)}$ di 2 V, dissipa:

$$PTR1 = 2 \text{ V} \times 1,4 \text{ A} = 2,8 \text{ W}$$

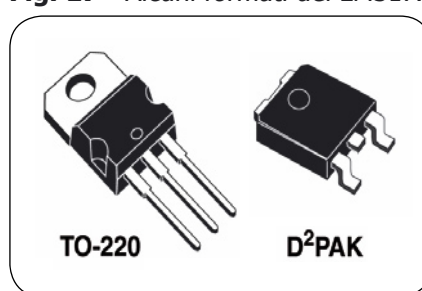
Vediamo il regolatore 7812; ai suoi capi cadono:

$$17 \text{ V} - 12 \text{ V} = 5 \text{ V}$$

con una corrente di 100 mA, per cui dissipa:

$$PU1 = 5 \text{ V} \times 100 \text{ mA} = 500 \text{ mW}$$

Fig. 17 - Alcuni formati del LM317.

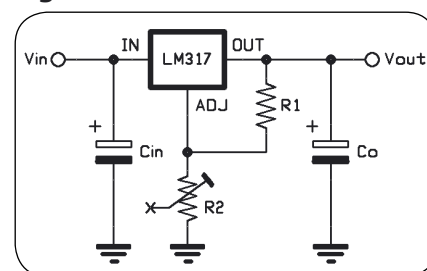


Regolatori a tensione fissa negativa: famiglia 79XX

I regolatori di questa famiglia permettono di ottenere delle tensioni negative stabilizzate a valori fissi, ossia -5, -6, -8, -9, -10, -12, -15, -18 e -24 V. Anch'essi sono suddivisi in due sottoclassi, la 79XX che eroga fino a 1,5 A e la 79LXX, in grado di fornire fino a 100 mA.

Per i regolatori di questa famiglia vale quanto detto per quelli della famiglia 78XX (sono, infatti, i loro complementari), con la differenza che la tensione di uscita è negativa e la corrente è entrante nel pin 3.

Fig. 18 - Circuito con LM317.



In Fig. 16 è raffigurato un circuito di esempio, praticamente identico (polarità a parte) a quello visibile in Fig. 14.

Regolatori a tensione variabile positiva: LM317

I regolatori di tipo LM317 (disponibili anch'essi in diverso formati, Fig. 17) permettono di ottenere delle tensioni positive variabili a piacimento, cambiando semplicemente il valore di due resistenze. Ecco le principali caratteristiche di questo tipo di regolatori:

- corrente di uscita fino a 1,5 A;
- tensione in uscita variabile tra 1,2V e 37 V;
- protezione interna sia sul sovraccarico che sul corto circuito dell'uscita;
- dropout tra 1,5 V (a 500 mA) e 2,2 V (1,5 A);
- ripple in uscita: -80 dB (1/10000 del ripple in ingresso)

In Fig. 18 è mostrato il tipico circuito elettronico completo di un regolatore di tensione positiva variabile realizzato con un LM317. Notate che il condensatore Cin è necessario se il regolatore è distante

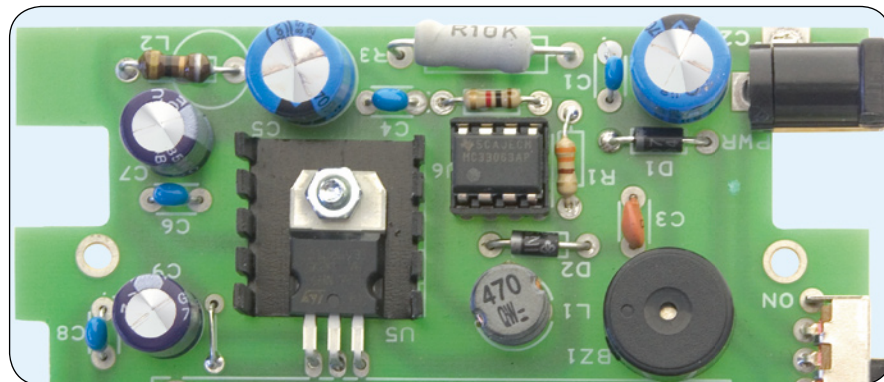
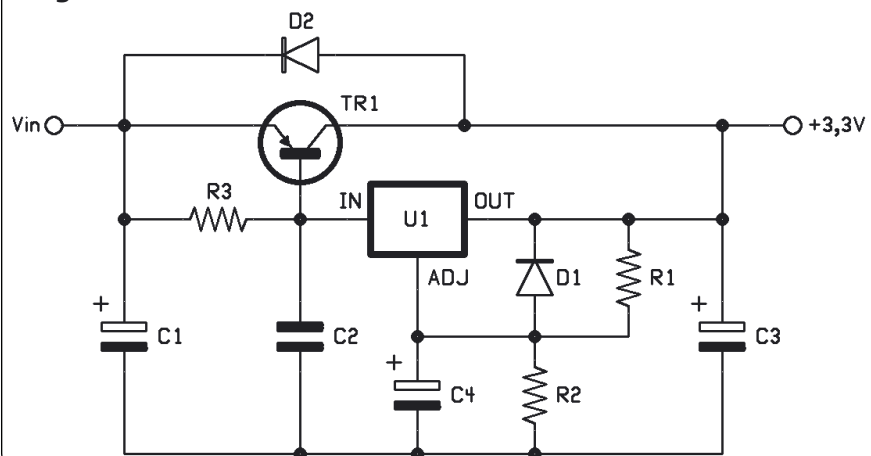


Fig. 19 - Circuito con LM317 e transistor.



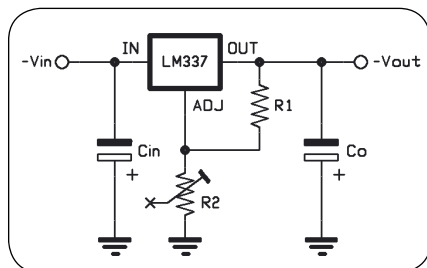
dal filtro sull'alimentazione.

Esiste una formula per dimensionare le due resistenze R1 e R2 in modo da ottenere in uscita la tensione voluta, in funzione della tensione di riferimento presente sul pin 1, tipicamente pari a 1,25 V.

Tale formula è:

$$V_{out} = 1,25 \times (1 + R2/R1) + (I_{adj} \times R2).$$

Fig. 20 - Circuito con LM337.



Essendo I_{adj} minore di 0,1 mA, spesso si considera solo

$$V_{out} = 1,25 \times (1 + R2/R1).$$

Qualora si volesse avere in uscita

una corrente superiore agli 1,5 A che l'integrato LM317 è in grado di fornire, è possibile ricorrere al circuito di Fig. 19, che utilizza il transistor TR1 di potenza, per esempio un BDX54C visto in precedenza. Il principio di funzionamento è praticamente identico a quello del circuito di Figura 14, per cui restano valide le considerazioni fatte e anche l'analisi del funzionamento circuitale.

Regolatori a tensione variabile negativa: LM337

I regolatori di tipo LM337 permettono di ottenere delle tensioni negative variabili a piacimento, cambiando il valore di due resistenze.

Ecco le principali caratteristiche di questo tipo di regolatori:

- corrente di uscita fino a 1,5 A;
- tensione di uscita variabile tra -1,2 V e -37 V;
- protezione interna sul sovraccarico

e sul cortocircuito in uscita;

- limitazione interna sul corto circuito dell'uscita;
- dropout tra 1,5 V (a 500 mA) e 2,2 V (a 1,5 A);
- ripple in uscita: -80dB (1/10000 del ripple in ingresso).

In Fig. 20 è mostrato il tipico circuito elettronico completo di un regolatore di tensione positiva variabile fatto con LM337, circuito che, a parte il verso delle correnti, è identico a quello visto in Fig. 17.

Relativamente al calcolo della tensione in uscita in funzione delle resistenze R1 e R2, e all'incremento della massima corrente fornibile al carico, vale esattamente quanto visto per l'LM317.

Note conclusive

Qui si conclude la prima parte del nostro minicorso sulla progettazione degli stadi di alimentazione.

Nella seconda parte, conclusiva, parleremo sia degli alimentatori switching che del calcolo per il corretto dimensionamento dei dissipatori.

Vi proporremo, inoltre, il progetto di un piccolo alimentatore switching di facile realizzazione e utilizzabile in molte applicazioni hobbistiche.

L'appuntamento è quindi tra un mese in edicola.



**Tutti i nostri progetti,
i master, i firmware
e tanto altro ancora su**

www.elettronica.in.it

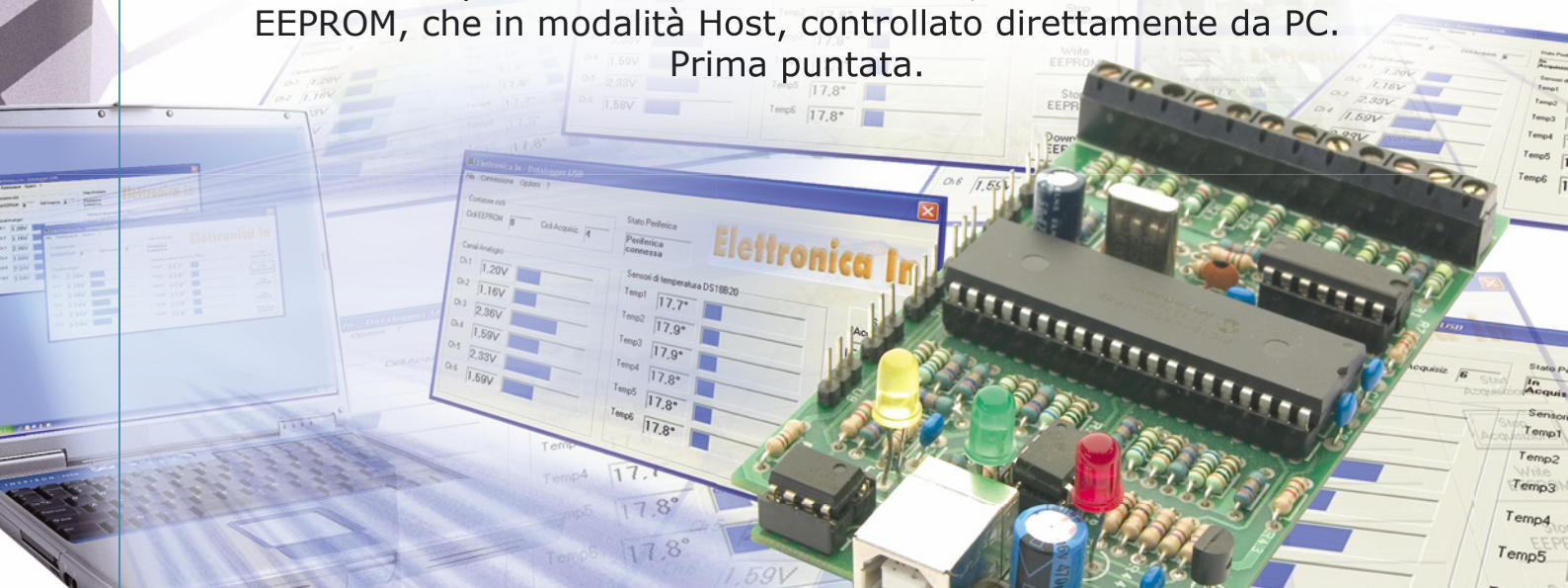
Collegati subito!

Datalogger USB a 12 canali

di *Francesco Ficili*

Sistema di acquisizione dati dotato di sei ingressi analogici e sei sensori di temperatura one-wire DS18B20. Viene gestito tramite un PC cui è collegato mediante la porta USB, per mezzo di un applicativo in ambiente Windows. Può operare sia in modo autonomo, salvando i dati sulla sua EEPROM, che in modalità Host, controllato direttamente da PC.

Prima puntata.



Anche se non sono sempre così evidenti, esistono delle applicazioni particolari in cui può essere necessario controllare costantemente con un solo dispositivo dei parametri di varia natura. Pensate, per esempio, alla serra tropicale di un giardino botanico: in tale situazione, si deve riprodurre il microclima di una particolare zona geografica in cui temperatura, umidità e illuminazione vanno mantenute entro livelli precisi per non alterare il delicato equilibrio tra flora e fauna, ricreando così un microcosmo il più vicino possibile alla realtà.

Bene: in situazioni come questa, così come in altre (ri-

levazione di parametri ambientali quali inquinamento, PM10, misura dell'opacità di fumi e acque, eccetera), si deve ricorrere ad un datalogger, particolare dispositivo in grado di controllare contemporaneamente diversi sensori (tipicamente un sensore per ogni canale) e di memorizzare ad intervalli prestabiliti le rilevazioni effettuate da ciascun sensore.

Ciò che vi presentiamo in questo articolo è proprio il progetto di un datalogger a dodici canali, sei collegati ad altrettanti sensori di temperatura digitali e sei in grado di rilevare tensioni comprese tra 0 e 30 Vdc. Il dispositivo ➤

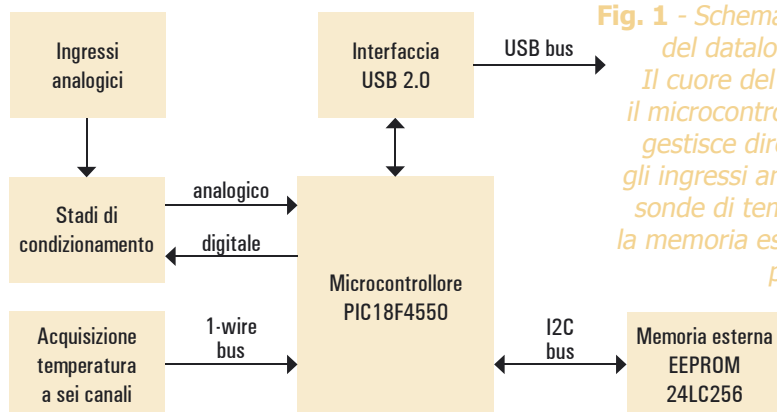


Fig. 1 - Schema a blocchi del datalogger USB. Il cuore del sistema è il microcontrollore, che gestisce direttamente gli ingressi analogici, le sonde di temperatura, la memoria esterna e la porta USB.

può essere gestito tramite qualsiasi PC (collegato ad una porta USB) su cui sia installato uno specifico software di gestione, che sarà argomento della puntata successiva.

Relativamente agli ingressi in tensione, è previsto il controllo automatico del guadagno per ognuno di essi, così da garantire sempre la miglior risoluzione in lettura.

L'acquisizione della temperatura, invece, viene effettuata tramite sei sonde 1-wire® modello DS18B20, di produzione Maxim-Dallas. Il tutto è gestito da un microcontrollore PIC18F4550, dispositivo Microchip di medio taglio che, tra le varie risorse hardware di cui dispone, include una porta USB 2.0 completa. Vediamo, ora, tutti i dettagli tecnici di questo progetto.

Schema a blocchi

Lo schema a blocchi del datalogger USB, riportato in Fig 1, evidenzia le sei parti in cui può essere suddiviso il circuito.

Il cuore dell'intero sistema è rappresentato dal microcontrollore PIC18F4550 che gestisce tutte le periferiche presenti sulla scheda e si occupa di stabilire e mantenere la connessione con l'host USB (PC). È dotato di 32 kbyte di memoria Flash, 2 kbyte di RAM, 256 byte di EEPROM interna e di svariate periferiche. Dispone di un circuito di oscillazione a PLL integrato, che gli consente di lavorare ad una frequenza massima di 48 MHz (12 MIPS). Il micro controlla direttamente gli stadi di condizionamento (secondo blocco): si tratta di sei circuiti ugua-

li tramite i quali il segnale applicato agli ingressi analogici viene opportunamente attenuato/amplificato, in modo da garantire sempre la massima risoluzione nella lettura del valore di tensione. Il guadagno di ogni canale viene variato dal microcontrollore.

Relativamente agli ingressi analogici (terzo blocco), si tratta di sei partitori di tensione passivi, ad ognuno dei quali può essere collegata l'uscita di sensori di temperatura, di umidità, di fumo, eccetera, purché forniscano in uscita una tensione minima di 0 Vdc e massima di 30 Vdc.

Il quarto blocco è il circuito di acquisizione della temperatura: il datalogger è predisposto per la connessione con sei sensori di temperatura DS18B20. Questi dispositivi, come già accennato, comunicano con il micro attraverso il protocollo proprietario 1-wire®, che, come si evince dal nome, utilizza un bus a singolo filo bidirezionale (più alimentazione e massa).

Rimangono, infine, la memoria esterna e la porta USB: la prima è rappresentata da una EEPROM 24LC256 (prodotta da Microchip) su cui vengono memorizzati i dati provenienti dagli ingressi, quando la scheda opera in modo autonomo; può contenere fino a 32 kbyte di dati.

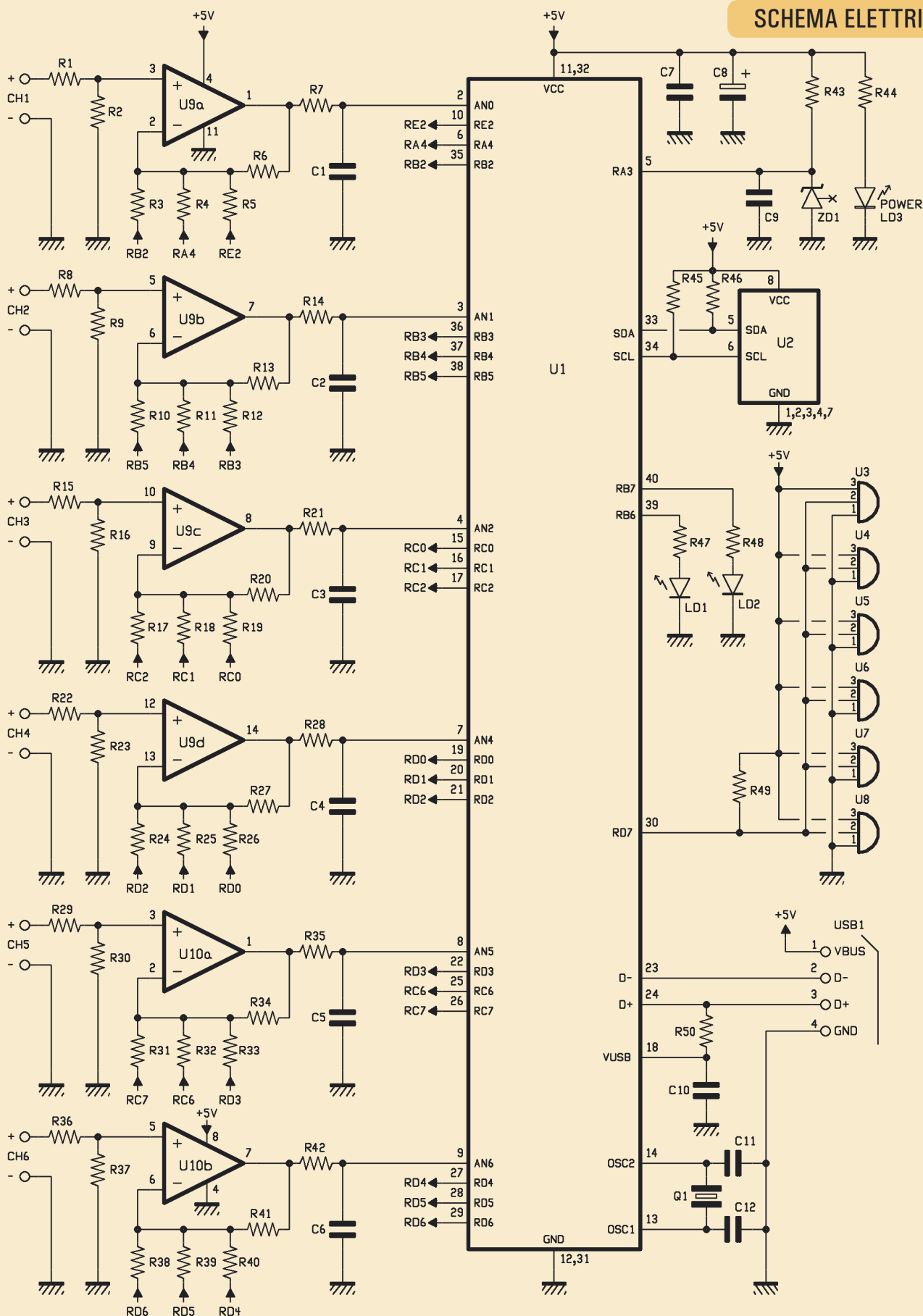
La seconda, invece (interna al micro e compatibile alla versione 2.0 full-speed del protocollo) viene utilizzata sia per la gestione tramite PC che per il trasferimento dei dati quando il datalogger funziona in modalità host.

Schema elettrico

Ad una prima analisi, appare evidente che lo schema elettrico ripropone esattamente le varie sezioni viste nello schema a blocchi, quindi non sarà difficile comprenderlo. Iniziamo la descrizione dai sei canali analogici con i relativi stadi di condizionamento. Questi ultimi sono realizzati dagli integrati U9 (TLV274) e U10 (TLV272), degli amplificatori operazionali (rispettivamente quadruplo

- Ingressi analogici: 6, con sensibilità di $0 \div 30$ Vdc e autorange.
- Risoluzione campionamento a 8 bit.
- Ingressi temperatura: fino a 6 sensori digitali DS18B20.
- sensibilità $-25 \div 125$ °C, risoluzione a 12 bit (0,0625 °C).
- Interfaccia PC: USB full-speed 2.0, software di gestione Windows.
- Alimentazione: via USB.
- Possibilità di memorizzazione dati di temperatura/tensione su EEPROM (fino a 32 Kbyte). Possibilità di regolare gli intervalli di salvataggio tramite PC.
- Indicatori visivi: interfaccia grafica su PC + 3 LED di stato.
- Software di gestione con diverse modalità di funzionamento:
 - Acquisizione real-time e salvataggio dati su file;
 - Memorizzazione su EEPROM interna;
 - Possibilità di impostare i guadagni dei canali analogici;
 - Regolazione degli intervalli di salvataggio;
 - Possibilità di impostare allarmi sui vari ingressi.

SCHEMA ELETTRICO



e doppio) a singola alimentazione e uscita rail-to-rail (in grado, cioè, di assumere valori compresi tra 0 V e la tensione di alimentazione) prodotti da Texas Instruments. Essi vengono utilizzati nella configurazione “amplificatore non invertente” e hanno in ingresso un partitore di tensione con rapporto di circa 1/12 (vedere, ad esempio R1 ed R2, rispettivamente da 1 Mohm e da 91 kohm), che consente agli ingressi analogici di tollerare tensioni d’ingresso comprese tra 0 e 30 Vdc, nonostante l’alimentazione sia a 5 Vdc. Dicevamo che gli operazionali vengono utilizzati dal micro per attenuare o amplificare opportunamente il segnale in ingresso: il controllo del guadagno viene effettuato tramite una rete di tre resistenze che vengono collegate a massa una alla volta a seconda dello stato degli I/O del microcontrollore, seguendo una logica di funzionamento che verrà descritta nel paragrafo successivo. Completano gli stadi d’ingresso sei filtri passa-basso di tipo RC (R7, R14, R21, R28, R35, R42 e C1 ÷ C6) con frequenza di taglio di circa 1 kHz, collegati ognuno all’uscita di un operazionale. I sei segnali in uscita ai filtri, infine, vengono collegati agli ingressi AN0, AN1, AN2, AN4, AN5, AN6 del convertitore AD del microcontrollore.

In ingresso al pin 5 (RA3, Vref+) è stato inserito un circuito formato da ZD1 (LM385), dalla resistenza R43 e dal condensatore C9, che serve per dare un riferimento di 1,2 V (molto preciso e stabile) al convertitore AD del PIC, così da svincolare il risultato della conversione AD da eventuali (seppur minime) fluttuazioni della tensione di alimentazione.

Oltre che dai canali analogici, il data-logger può acquisire i dati forniti da sei sensori di temperatura (U3÷U8, DS18B20), in grado di effettuare rilevazioni termiche con una risoluzione massima di 12 bit, interfacciati alla logica di controllo del micro tramite il bus proprietario denominato

PIC18F4550

Fig. 2 - La periferica USB del PIC18F4550: necessita solo di una resistenza e un condensatore esterni.

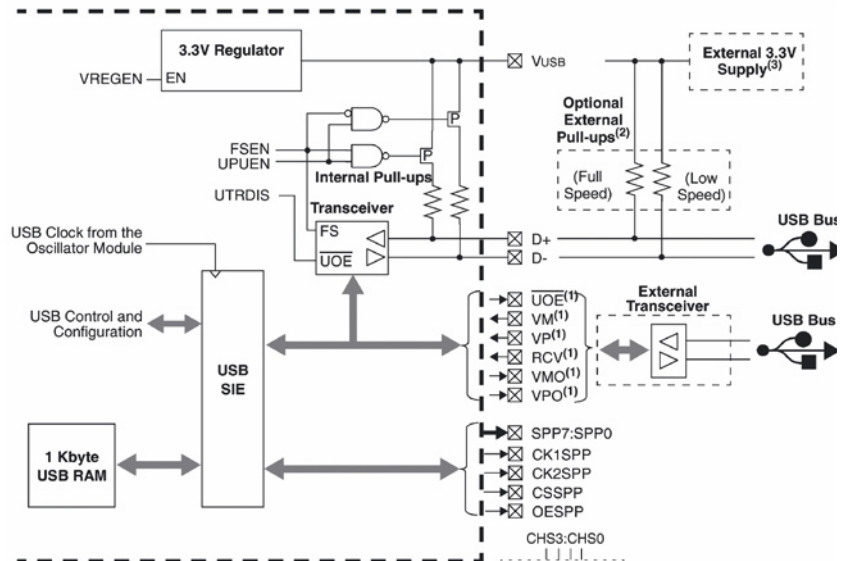


Fig. 3 - Schema a blocchi dell'ADC: dispone di 13 canali multiplexati, campionati con risoluzione di 10 bit.

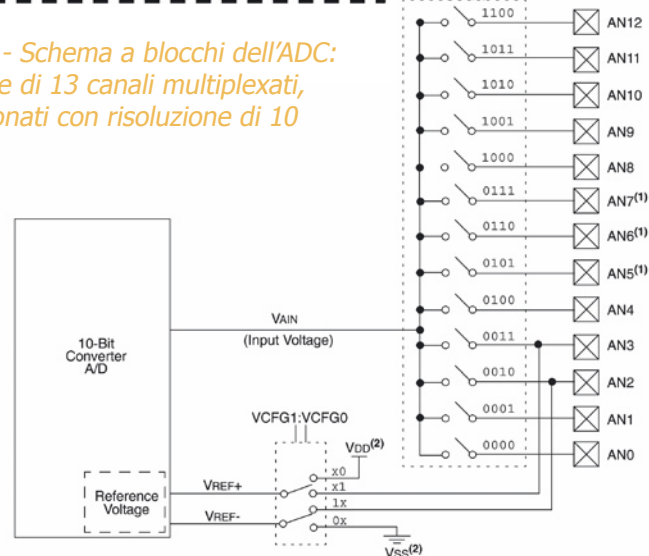
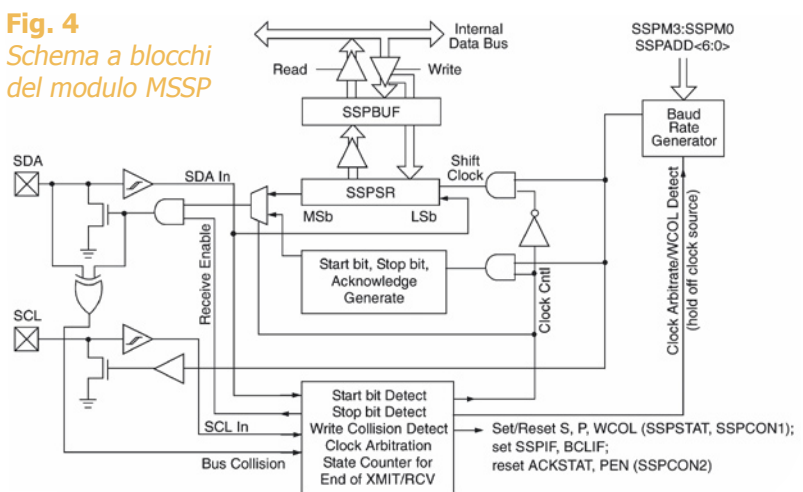


Fig. 4 - Schema a blocchi del modulo MSSP



1-wire®; esso consente di realizzare comunicazioni di tipo *master-slave* tra un microcontrollore e un numero

elevato di dispositivi impiegando un singolo filo per i dati, più le linee di alimentazione e massa.

Nel caso dei DS18B20 (e di altri dispositivi che utilizzano il protocollo 1-wire®) è stata anche prevista una modalità di funzionamento (denominata *Parasite Power*) che permette di impiegare solo due fili (dati e massa). Dal punto di vista hardware la connessione di un dispositivo 1-wire® ad un microcontrollore richiede un singolo pin di una porta (con uscita open-collector o three-state) ed una resistenza di pull-up. Nel nostro caso si è scelto di utilizzare il pin 7 della porta D, ed una resistenza da 10 kohm (R49).

Passiamo ora al blocco delle “memorie”, costituito, in realtà da una sola EEPROM (U2, una 24LC256): per gestire la comunicazione tra quest’ultima e il micro sono stati usati i due pin SCK e SDO del modulo MSSP (Master Synchronous Serial Port, Fig. 4) di cui è dotato il PIC18F4550, che viene utilizzato come interfaccia I²C master (insieme alle due resistenze di pull-up R45 ed R46). Per quel che riguarda l’alimentazione della scheda, viene ricavata direttamente dalla porta USB: notate che nel circuito sono stati aggiunti i due condensatori C7 (100 nF multistrato) e C8 (470 µF elettrolitico) per sopprimere eventuali disturbi residui e sopperire ai picchi di assorbimento. Sono inoltre stati previsti tre led per fornire indicazioni visive immediate sullo stato di funzionamento del dispositivo. LD3, pilotato direttamente dal positivo dell’alimentazione attraverso la resistenza di limitazione R44, indica la presenza di tensione sulla scheda. LD1 e LD2, invece, sono pilotati dal PIC attraverso le linee RB6 ed RB7 e le resistenze R47 ed R48: essi forniscono indicazioni sullo stato della connessione USB, come si vedrà più in dettaglio nella puntata dedicata al firmware.

Il circuito di oscillazione del microcontrollore è stato realizzato con il quarzo Q1, da 20 MHz e i due condensatori C11 e C12 da 10 pF. L’interfacciamento al PC avviene

sfruttando il modulo USB interno del microcontrollore: i PIC della famiglia X550, infatti, dispongono sia di un USB Serial Interface Engine (SIE) compatibile con le modalità low-speed e full-speed della specifica USB 2.0, sia di un complesso di periferiche che permettono di incrementare le prestazioni e ridurre al minimo l’hardware esterno.

In Fig. 2 è riportato lo schema a blocchi del modulo USB della famiglia X550: il set di periferiche hardware dedicate che compongono il modulo USB comprende:

- un transceiver USB integrato;
- un regolatore interno a 3,3 V;
- due resistenze di pull-up interne;
- alcuni circuiti logici di controllo;
- 1 kbyte di “dual port RAM” utilizzata come buffer di ricezione e trasmissione.

Tutto ciò consente di collegare il microcontrollore direttamente alla porta USB del PC, aggiungendo solo un condensatore, una resistenza e il connettore esterno: il condensatore C10 (4,7 µF) stabilizza la tensione sul bus USB, tra il pin 18 (Vusb) e la massa, mentre la resistenza di pull-up R50 (1,5 kohm) permette al PC di classificare il dispositivo come full-speed.

Principio di funzionamento

Analizziamo ora in modo più approfondito alcuni dettagli relativi al funzionamento delle vari sezioni che compongono il datalogger, dando maggiore risalto ad alcune scelte effettuate in fase di progettazione.

La prima sezione esaminata è il blocco di canali analogici d’ingresso e dei relativi stadi di condizionamento, realizzati con sei amplificatori operazionali in configurazione non invertente e collegati ad opportune reti di retroazione resistive.

Definendo le specifiche del progetto, ci siamo imposti che ogni ingresso analogico dovesse poter accettare tensioni comprese tra 0 e 30 V e, per fare ciò, su ogni stadio di amplificazione sono state aggiunte tre re-

sistenze di retroazione tra l’ingresso invertente dell’operazionale e tre pin di I/O del PIC.

Impostando uno di questi pin come output e forzandolo a massa, la resistenza ad esso collegata diventa un componente della rete di retroazione, mentre impostando il pin come input l’effetto di quest’ultima viene neutralizzato dal buffer three-state della porta del microcontrollore, che si pone in stato di alta impedenza. In questo modo è possibile “programmare” il guadagno dell’operazionale con una opportuna combinazione dei registri di controllo delle porte.

Chiariamo meglio questo concetto analizzando lo stadio U10b, il cui guadagno dipende dalle resistenze R38, R39 e R40. Ipotizziamo che il micro imponga la linea RD6 (che controlla R38) come uscita a 0 logico e le linee RD5 e RD4 come ingresso (alta impedenza). In questo caso, R39 e R40 non concorrono all’amplificazione di U10b, il cui guadagno in tensione è dato da

$$A_v = 1 + (R41/R38).$$

Sostituendo i valori noti nella formula, si ottiene

$$A_v = 1 + (27\text{ k} / 3\text{ k}) = 10.$$

Lo stesso discorso è valido per le resistenze R39 (6,8 kohm) e R40 (27 kohm), che determinano un guadagno di 5 e di 2.

Ricordando che il partitore di ingresso attenua 12 volte e che la tensione di alimentazione è di 5 V, si può risalire alla tensione massima applicabile all’ingresso con la formula

$$V_{i_{\max(A_vN)}} = (V_{u_{\max}} \times 12) / A_v$$

in cui $V_{i_{\max(A_vN)}}$ rappresenta la massima tensione in ingresso in funzione del guadagno, $V_{u_{\max}}$ corrisponde alla

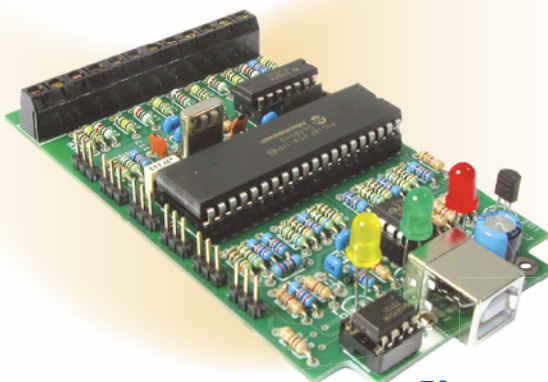
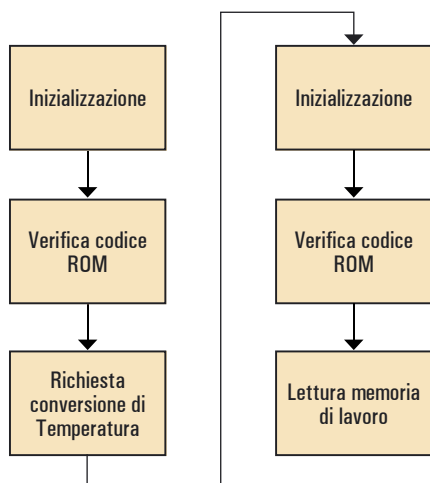


Fig. 5 - Flusso delle operazioni di lettura della temperatura.



massima tensione in uscita all'operazione (pari a 5 Vdc) mentre A_v è il guadagno in tensione imposto dal micro. Sostituendo nuovamente i valori noti alla formula, si ottiene:

$$V_{i_{max(Av2)}} = (5 \times 12) / 2 = 30 \text{ Vdc}$$

$$V_{i_{max(Av5)}} = (5 \times 12) / 5 = 12 \text{ Vdc}$$

$$V_{i_{max(Av10)}} = (5 \times 12) / 10 = 6 \text{ Vdc}$$

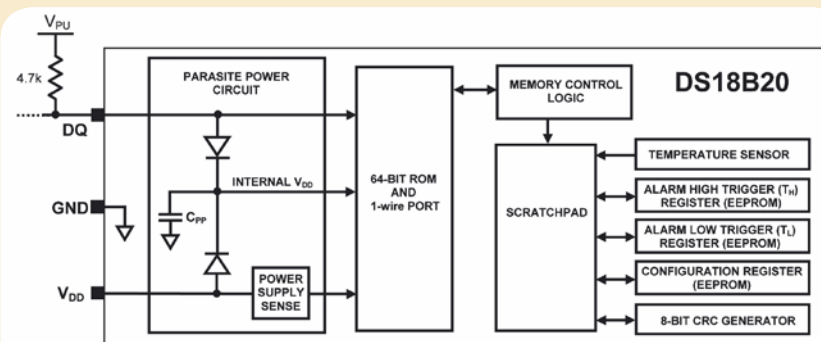
che definiscono l'escursione della tensione in ingresso in funzione del guadagno imposto dal microcontrollore. Torniamo, ora, ai sei canali analogici d'ingresso che vengono acquisiti in multiplexing dal convertitore A/D del microcontrollore (in grado di acquisirne fino a 13, con una risoluzione massima di 10 bit, Fig. 3). Nel progetto in esame, però, si è scelto di impostare la risoluzione del convertitore ad 8 bit (che nel range di tensione 0-5 V fornisce una risoluzione di 19,5 mV) per alleggerire il traffico dati tra datalogger e software su PC. Vediamo ora come abbiamo progettato il circuito di acquisizione delle temperature. Il problema principale che si pone nella gestione dei DS18B20 (le cui caratteristiche sono riportate in un riquadro a parte) è quello di riuscire a comunicare con i vari dispositivi collegati al bus evitando che si generino collisioni. Per far ciò viene utilizzato il codice univoco a 64 bit contenuto nella memoria ROM di ogni sensore. Il concetto che viene utilizzato è il seguente:

DS18B20

- Range di temperatura: $-55 \div 125^\circ\text{C}$.
- Accuratezza: $\pm 0,5^\circ\text{C}$.
- Risoluzione: programmabile da 9 a 12 bit.
- Codice identificativo: a 64 bit memorizzato su on-board ROM.
- Range di alimentazione: $3,0 \div 5,5 \text{ Vdc}$.
- Tempo di conversione: 750 ms con risoluzione a 12 bit.
- Allarmi di temperatura: fino a 2.
- Alimentazione: sia diretta che derivata dalla linea dati (Parasite Power).



Fig. 6 - Schema a blocchi del sensore di temperatura DS18B20 e formato del dato



LS Byte	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
	2^3	2^2	2^1	2^0	2^1	2^2	2^3	2^4
MS Byte	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
	S	S	S	S	S	2^8	2^5	2^4

durante ogni transazione (scambio di dati con una periferica) il master, in una prima fase, comunica con tutti

con un DS18B20, utilizzando il protocollo 1-wire®, si compone di tre step, precisamente:

Tabella 1 - Alcuni esempi di frequenza di campionamento e durata massima di acquisizione

Frequenza di campionamento	Durata massima acquisizione
1 acquisizione ogni 5 secondi	150 minuti
1 acquisizione ogni 10 secondi	300 minuti
1 acquisizione ogni 30 secondi	15 ore
1 acquisizione ogni minuto	30 ore
1 acquisizione ogni 30 minuti	900 ore

i dispositivi *slave*, inviando sul bus il codice ROM del dispositivo che intende indirizzare. A questo punto il dispositivo indirizzato risponde, mentre tutti gli altri si mettono in attesa fino all'inizio della successiva comunicazione. La sequenza per effettuare una singola transazione dati

1. Inizializzazione
2. ROM Command
3. DS18B20 Function Command

Lo step 1 è standard, mentre gli step 2 e 3 contengono un comando che varia a seconda dei casi (match, lettura o salto della ROM, conversione della temperatura, impostazione degli al-

larmi, eccetera) e sono solitamente seguiti da uno scambio di dati tra il microcontrollore e il DS18B20. Nel nostro caso, per ottenere una lettura di temperatura, sono necessarie due

precedente. In Fig. 5 è riportato lo schema logico delle transazioni che avvengono tra il microcontrollore ed il DS18B20 per la lettura di un dato di temperatura. La procedura

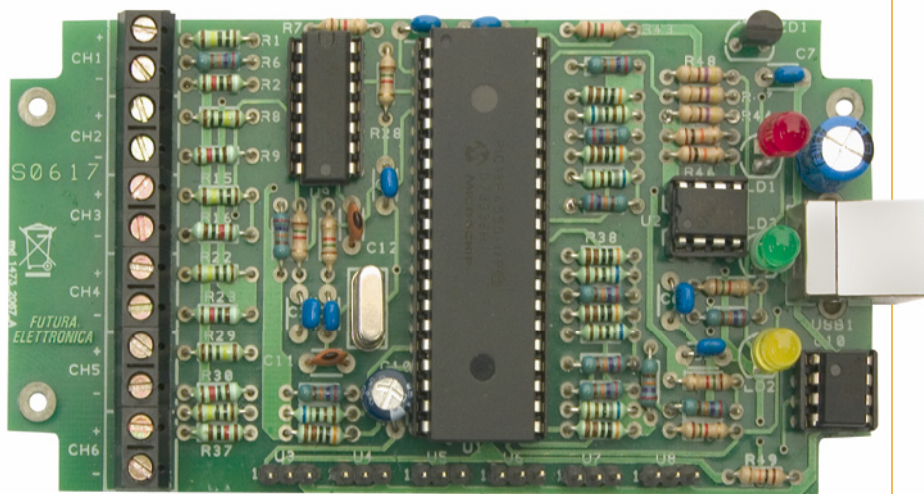
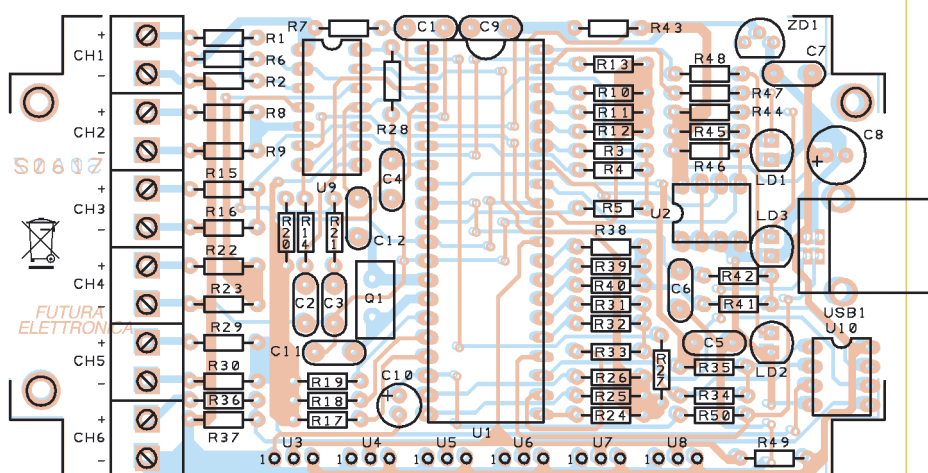
impiegando una singola resistenza di pull-up, e ciò non è cosa da poco. In Fig. 6 è riportato lo schema a blocchi del DS18B20, insieme al formato del pacchetto dati. Il datalogger è anche

piano di MONTAGGIO

Il datalogger è realizzato con componenti ad inserzione e, nella realizzazione in pratica, non presenta particolari difficoltà.

ELENCO COMPONENTI:

R1, R8, R15: 1 Mohm 1%
 R2, R9, R16: 91 kohm 1%
 R3, R10, R17: 3 kohm 1%
 R4, R11, R18: 6,8 kohm 1%
 R5, R6, R12, R13: 27 kohm 1%
 R7, R14, R21, R28: 1,5 kohm
 R19, R20, R26, R27: 27 kohm 1%
 R22, R29, R36: 1 Mohm 1%
 R23, R30, R37: 91 kohm 1%
 R24, R31, R38: 3 kohm 1%
 R25, R32, R39: 6,8 kohm 1%
 R33, R34, R40, R41: 27 kohm 1%
 R35, R42, R43, R50: 1,5 kohm
 R44, R47, R48: 470 ohm
 R45, R46, R49: 10 kohm
 C1÷C7, C9: 100 nF multistrato
 C8: 470 µF 16 VL elettrolitico
 C10: 4,7 µF 16 VL elettrolitico
 C11, C12: 10 pF ceramico
 U1: PIC18F4550
 U2: 24LC256
 U3÷U8: DS18B20
 U9: TLV274
 U10: TLV272
 ZD1: LM385
 LD1: Led 5 mm rosso
 LD2: Led 5 mm giallo
 LD3: Led 5 mm verde
 Q1: Quarzo 20 MHz



Varie:

- Connettore USB-B
- Zoccolo passo doppio 20+20
- Zoccolo 4+4 (2 pz.)

- Zoccolo 7+7
- Morsettiera 2 poli (6 pz.)
- Strip femmina 3 pin (6 pz.)
- Circuito stampato codice S0617

transazioni: nella prima si identifica il dispositivo ed il *master* chiede allo *slave* di eseguire una conversione di temperatura; nella seconda si identifica nuovamente il dispositivo ed il *master* effettua la lettura della memoria di lavoro (*Scratchpad Memory*) contenente il dato richiesto allo step

può sembrare, a prima vista un po' macchinosa (e in effetti non è molto semplice): però questo sistema consente di collegare un numero decisamente elevato di dispositivi (in linea di principio, pari a tutte le possibili combinazioni del codice ROM a 64 bit) sul medesimo bus a singolo filo,

dotato di una memoria EEPROM I²C dove può memorizzare i dati quando opera in modalità autonoma; resta inteso che è possibile effettuare il download dei dati in qualsiasi momento. La memoria scelta è la 24LC256 da 256 kbit (32 kByte), che comunica con il microcontrollore at-

traverso il modulo hardware MSSP (Master Synchronus Serial Port, il cui schema a blocchi è riportato in Fig. 3) configurato in modalità I²C Master. L'immagazzinamento dei dati nella EEPROM viene gestito attraverso un comando software, che consente l'impostazione della frequenza di campionamento in base alle esigenze dell'utente. Nella Tabella 1 sono elencate alcune delle possibili scelte di frequenza di campionamento con le relative durate massime dell'acquisizione prima che saturi la EEPROM. Chiaramente, il problema della saturazione della EEPROM non si pone utilizzando il datalogger collegato al PC e salvando i dati su hard disk (dove si ha molta più memoria a disposizione). È stata aggiunta comunque la possibilità di scegliere la frequenza di campionamento, così da non generare file troppo grandi senza che ci sia la reale necessità.

Realizzazione pratica

Una volta realizzato il c.s. tramite il master doppia faccia, scaricabile dal nostro sito (www.elettronicaain.it), iniziate il montaggio con i componenti a basso profilo, ossia i passivi (resistenze, condensatori, quarzo), facendo attenzione alla polarità dei condensatori elettrolitici. Proseguite quindi con i tre led, anche in questo caso facendo attenzione a posizionare correttamente i reofori dell'anodo e del catodo ed il riferimento di tensione ZD1-LM385 (nel package TO-92), controllando che i tre piedini sia-

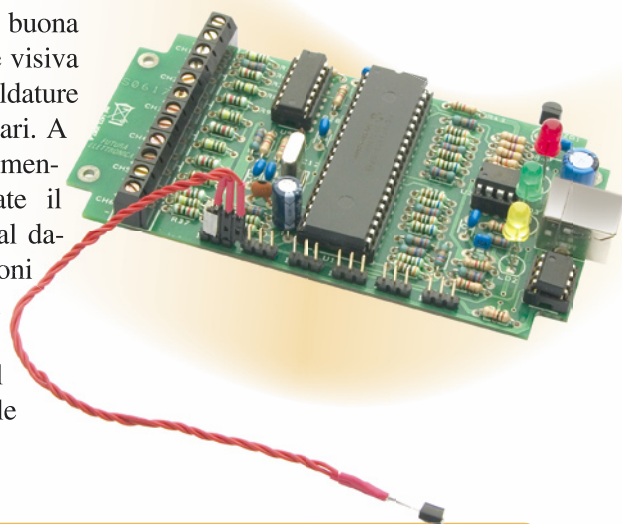
no inseriti nel verso corretto (l'anodo del diodo zener deve essere rivolto a massa). Nel montaggio dei led è consigliabile lasciar sporgere quanto più possibile i reofori dallo stampato (o almeno un centimetro) in modo da poterli eventualmente piegare, qualora si decidesse di inserire la scheda all'interno di una scatola (il datalogger è stato progettato per l'inserimento in un contenitore modello COFFER2). A questo punto si possono saldare sul PCB gli zoccoli dei quattro integrati, il quarzo ed il connettore USB tipo B femmina. Una volta saldati i quattro pin del connettore è consigliabile piegare accuratamente con una pinza le due alette di ancoraggio, in modo da fornirgli un solido aggancio allo stampato, in quanto questo componente sarà sollecitato meccanicamente da diversi cicli di inserimento ed estrazione del relativo plug. Ultimate il montaggio con le morsettiere da stampato a passo 5,08 mm e con le sei linee di pin strip a tre poli (passo 2,54 mm) cui verranno collegati i sensori di temperatura. Esaurite tutte le operazioni di saldatura, è buona norma effettuare un'ispezione visiva della scheda, alla ricerca di saldature fatte male o di cortocircuiti vari. A questo punto potete dare alimentazione alla scheda: collegate il cavetto USB (sia al PC che al datalogger) e controllate le tensioni di alimentazione sugli zoccoli degli integrati; dovete misurare circa 5 Vdc. Scollegato il circuito dal PC, ora è possibile

inserire la memoria EEPROM e i due integrati TLV272 e TLV274 nei relativi zoccoli, facendo attenzione a non invertire i due lati e controllando accuratamente che tutti i piedini entrino nelle rispettive sedi. Prima di inserire il PIC18F4550 nel rispettivo zoccolo a 40 pin, è necessario programmarlo con il firmware scaricabile gratuitamente dal sito della rivista. L'operazione può essere eseguita utilizzando un qualsiasi programmatore per PIC come i nostri FT284 e FT386.

Note conclusive

Giunti a questo punto, il datalogger USB è pronto per essere collegato al PC ed utilizzato: dovete però pazientare fino al mese prossimo.

Nella puntata conclusiva, interamente dedicata ai programmi, vi parleremo dettagliatamente sia del firmware a bordo del dispositivo, sia del software in ambiente Windows che abbiamo realizzato appositamente per esso. L'appuntamento è quindi tra un mese in edicola, con il fascicolo 126: non mancate!



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 da caricare sul PC possono essere scaricati gratuitamente dal sito della rivista (www.elettronicaain.it). Il microcontrollore è disponibile anche già programmato (cod. MF617) al prezzo di 20,00 Euro.

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

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

Segnalatore supplementare di chiamata telefonica

di Davide Scullino

Fa suonare un cicalino e pulsare un diodo luminoso all'arrivo di una telefonata; ideale come avvisatore per luoghi rumorosi e abitazioni composte da molte stanze, può essere utilizzato anche come rivelatore di chiamata.

Quando un telefono viene installato in ambienti molto rumorosi a causa delle lavorazioni che vi si svolgono o perchè particolarmente affollati, difficilmente si riesce a sentire l'arrivo di una telefonata; per aggirare l'ostacolo bisogna collegare alla linea un dispositivo ausiliario che può essere una suoneria molto potente oppure un avvisatore ottico. In altre parole, un ripetitore di chiamata.

Pensando alle situazioni descritte e ad altre ancora, abbiamo sviluppato il semplice dispositivo descritto in queste pagine: si tratta di un avvisatore ottico e ac-

stico ideale per avvertire dell'arrivo di una telefonata; la segnalazione ottica è realizzata con l'ausilio di un led ad altissima efficienza (ben 10 candele di intensità luminosa!) mentre quella acustica è ottenuta da un cicalino piezoelettrico.

Il diodo luminoso è particolarmente utile nei luoghi rumorosi, nei quali difficilmente si può udire una suoneria, per quanto forte possa essere; il cicalino serve invece nei fabbricati composti da molte stanze, dove può essere conveniente avere più suonerie collocate in punti che permettano di essere avvertiti ovunque ci troviamo. ➤

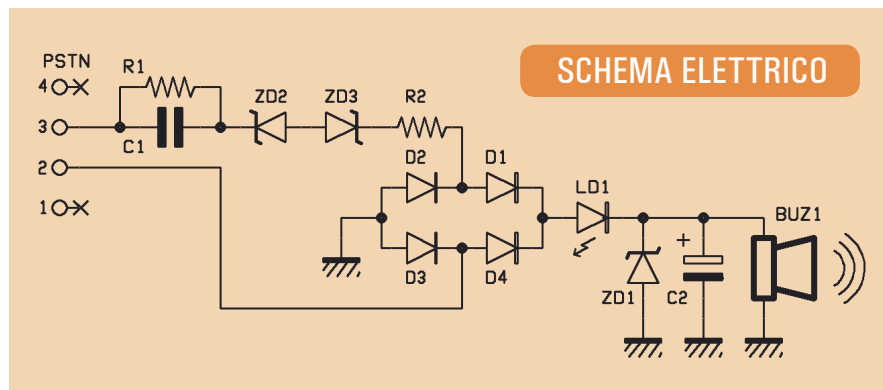
Dunque, quello che vi proponiamo è un ripetitore bivalente, utilizzabile in ogni situazione pratica, anche in condizioni particolari: ad esempio, in realtà dove operano e vivono non udenti consente a costoro di accorgersi dell'arrivo di telefonate e di avvertire chi deve andare a rispondere, laddove non si fosse accorto della chiamata in arrivo. Ma non solo: il circuito può servire semplicemente anche come avvisatore dell'arrivo di una chiamata su una certa linea, quindi da apparecchio di test nella ricerca di guasti; oppure da trigger per registratori di telefonate o per intercettazioni ambientali.

Schema elettrico

Il ripetitore di chiamata, che presenta alcune caratteristiche simili a quello già pubblicato sul fascicolo 123, è ottenuto con un circuito molto semplice, composto da pochissimi componenti elettronici; è inoltre (cosa particolarmente apprezzabile) passivo, nel senso che non richiede alcuna alimentazione esterna, in quanto ricava la poca corrente che necessita dallo stesso segnale di chiamata immesso sul doppino telefonico dalla centrale. Come già spiegato nel fascicolo scorso, infatti, per avvisare un utente dell'arrivo di una chiamata, la centrale telefonica del gestore della linea manda sul doppino una tensione alternata alla frequenza di 25/50 Hz, dell'ampiezza di 70

Veff. Tale tensione viene inviata a tratti: per l'esattezza, a cicli di 1 secondo intervallati da 4 secondi di pausa. Si utilizza l'alternata perché la parte di fonia del telefono è accoppiata e sempre alimentata dalla tensione continua presente ai capi della linea; la suoneria, che deve attivarsi esclusivamente in presenza di chiamata, deve quindi essere disaccoppiata dalla continua. Separandola con un condensato-

un rilevatore dell'alternata di chiamata, dotato di quei pochi componenti che occorrono a far lampeggiare un diodo luminoso e a far suonare un cicalino o ronzatore. La connessione alla linea viene effettuata mediante un canonico connettore RJ11; dai contatti 2 e 3 (i due interni) si preleva la linea, che viene applicata all'ingresso di un raddrizzatore a ponte di Graetz (D1, D2, D3, D4) preceduto da una cop-



re, la suoneria rimane inerte fino a quando il telefono è a riposo o in conversazione, situazioni in cui sulla linea possono essere presenti solamente segnali alternati in banda fonica (300÷3400 Hz) a livello molto basso (inferiori a 0 dBm) e perciò non in grado di attivare la suoneria.

Quando si sgancia la cornetta, la centrale telefonica rileva la risposta e sospende l'invio dell'alternata, che altrimenti si udirebbe nel ricevitore.

Il nostro circuito è sostanzialmente

pia di diodi Zener e da una rete R/C. Scopo del ponte è trasformare l'alternata di chiamata in una serie di impulsi a frequenza doppia rispetto all'alternata stessa, con i quali viene alimentato il cicalino e con esso il led LD1, che si trova in serie all'uscita del ponte; avremmo anche potuto risparmiarci il raddrizzatore, in quanto il led avrebbe esso stesso raddrizzato l'alternata, conducendo per un solo semiperiodo, tuttavia con la circuitazione adottata sfruttiamo al meglio l'energia fornita dalla chiamata, così da ottenere un

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). Il dispositivo è contraddistinto dal codice K8087 e costa 16,00 Euro, IVA compresa. La scatola di montaggio comprende tutti i componenti, la basetta forata e serigrafata, il contenitore plastico con frontale trasparente, i connettori nonché tutte le minuterie meccaniche.

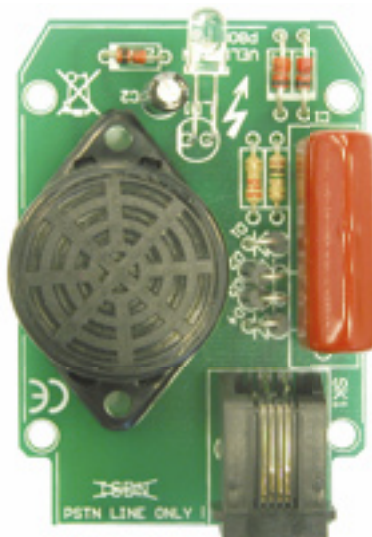
Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)

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

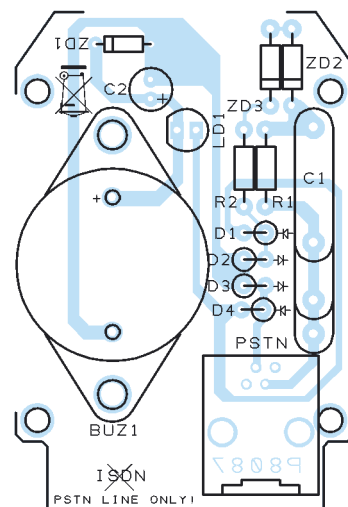
Il circuito montato. Qualora venga inserito in un contenitore plastico come quello da noi utilizzato (Fig. 1), i terminali di LD1 vanno piegati a 90° in modo da orientare la lente del led verso la finestra rossa trasparente. Per l'utilizzo in ambienti particolarmente rumorosi, potrebbe essere necessario realizzare dei fori sul contenitore, in modo da agevolare l'emissione sonora.

ELENCO COMPONENTI:

R1: 1 Mohm
R2: 2,2 kohm
C1: 1 μ F 275 VL poliestere
C2: 10 μ F 35 VL elettrolitico
D1: 1N4007
D2: 1N4007



D3: 1N4007
D4: 1N4007
ZD1: zener 12V 400mW
ZD2: zener 20V 400mW
ZD3: zener 20V 400mW
LD1: led 5 mm rosso



BUZ1: Buzzer da CS

Varie:

- Connettore telefonico RJ11 da CS
- Contenitore plastico
- Circuito stampato

suono che sia il più forte possibile. I due diodi Zener, montati contrapposti, servono a introdurre una soglia inferiore di tensione, ovvero ad evitare che il nostro avvisatore di chiamata venga attivato da segnali alternati la cui ampiezza sia inferiore a 45 volt.

In tal modo siamo certi che l'innesco avvenga solo in presenza dell'alternata di chiamata inviata dalla centrale. La predetta soglia è dovuta alla somma delle cadute di tensione sui vari elementi in serie ossia:

- 20,7 V circa su DZ2 e DZ3 che, essendo collegati in antiserie, sono sempre uno in conduzione diretta e l'altro in conduzione inversa;
- 1,4 V sul ponte raddrizzatore;
- 2 V sul led ad alta efficienza;
- 12 V sul diodo zener ZD1 (che alimentano il buzzer);
- 2,2 V sulla resistenza R2 e 6,4 V ai capi di C1 (che a 25 Hz presenta un'impedenza di circa 6400 ohm) ipotizzando che tutta la se-

rie sia percorsa da una corrente di 1 mA.

per una soglia di circa 44,7 V (avevamo detto 45 V, quindi...).

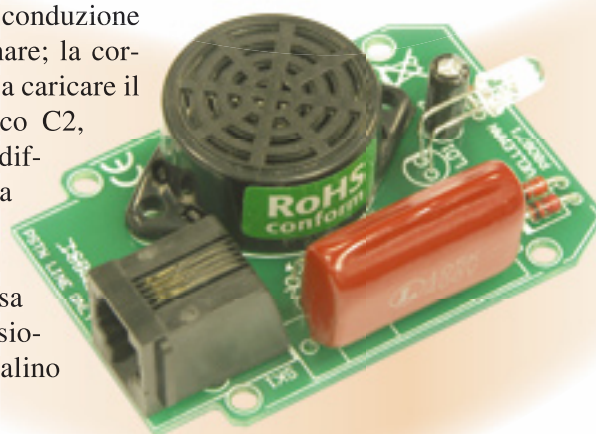
Il condensatore C1 serve a disaccoppiare in continua il rilevatore dell'alternata di chiamata, mentre la resistenza R1 è stata inserita per scaricarlo, così da evitare che, scollegando il circuito, si "prenda la scossa"; è stato scelto un elevato valore resistivo per evitare che il circuito assorba, in continua, una corrente abbastanza intensa da simulare l'impegno della linea.

In chiamata, ogni impulso ricavato dall'alternata manda in conduzione LD1, facendolo illuminare; la corrente uscente dal led va a caricare il condensatore elettrolitico C2, ai cui capi si trova una differenza di potenziale la cui ampiezza è limitata dal terzo diodo Zener: DZ1. Quest'ultimo fissa a 12 V la massima tensione che alimenta il cicalino BUZ1.

Realizzazione pratica

Giunti a questo punto non c'è altro da dire, almeno sul piano teorico; possiamo quindi vedere come costruire il ripetitore e metterlo in opera: tutti i componenti prendono posto su un piccolo circuito stampato facilmente realizzabile sia per fotoincisione che con la tracciatura diretta; volendo, vista la semplicità, nulla vieta di utilizzare un pezzetto di "millefori".

Dopo l'incisione e la foratura, disponete, nell'ordine, le resistenze, i diodi al silicio e i condensatori, dando la precedenza a quelli non ➤



CD3+
Disturbatore Cellulari
All Bands
23W totali min.

Vendita soggetta a limitazioni

NEW

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

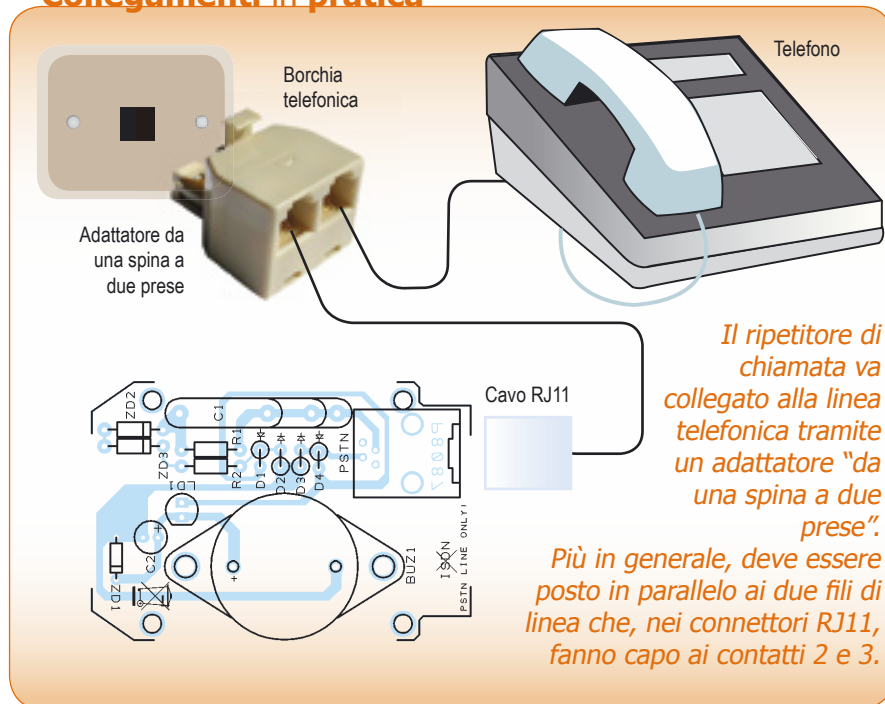
DESPY

Ideale per la ricerca di microspie fino a 3 Ghz

BIAS s.a. www.bias.sm

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

Collegamenti in pratica



polarizzati. Sistemate poi il led e la presa RJ11 da circuito stampato, necessaria a collegare il ripetitore alla linea del telefono.

Durante tutte le fasi del montaggio, per il corretto orientamento dei componenti polarizzati (diodi ed elettrolitici) fate riferimento al piano generale di cablaggio visibile in queste pagine.

L'installazione

Una volta terminato il montaggio, racchiudete il circuito in un piccolo contenitore plastico forato in modo da lasciare accessibile la presa plug RJ11 nella quale inserire il connettore del cavetto telefonico.

Prevedete anche un foro dal quale dovrà spuntare il led ad alta efficienza ed un altro (o una feritoia) per far uscire il suono del cicalino piezoelettrico.

Per il collegamento alla linea, è previsto l'uso di una classica piattina per telefoni, terminante con connettori RJ11; se l'impianto cui si intende applicare il nostro circuito non ha una presa libera, procuratevi un adattatore "da una spina a due prese".

Sia l'adattatore che il cavetto RJ11 già cablato sono facilmente reperibili nei negozi di elettronica di consumo, ma anche nei grandi supermercati e centri commerciali.

Fig. 1





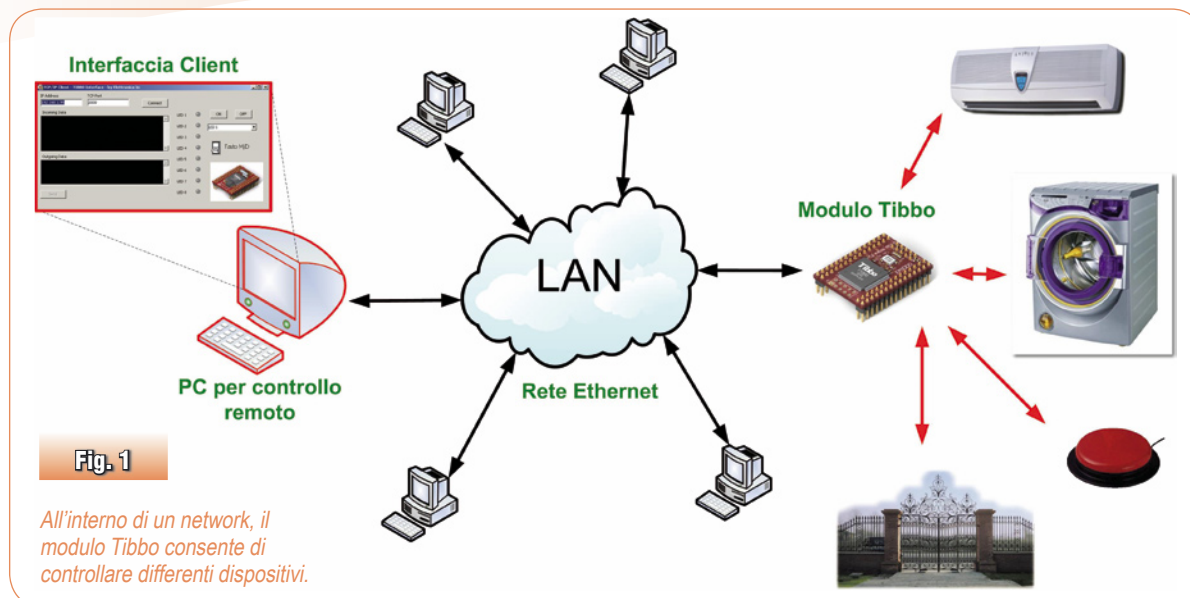
Corso di programmazione modulo Tibbo EM1000

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 presentiamo il primo di una serie di casi applicativi, in cui il modulo EM1000 è utilizzato per remotizzare telecomandi e telecontrolli di vario genere, il tutto utilizzando una pratica interfaccia Client in grado di comunicare con il modulo stesso. Quarta puntata.

Dopo aver illustrato i potenziali campi di applicazione del modulo Tibbo, averne descritto le caratteristiche hardware (EM1000) ed il relativo ambiente di sviluppo (TIDE), entriamo ora nel vivo della programmazione e sperimentazione con un primo caso pratico: la gestione di telecomandi e telecontrolli. Tramite il dispositivo sarà possibile inviare dei comandi da una qualunque postazione PC su cui è utilizzato un software

in grado di comunicare ed inviare al modulo Tibbo dei comandi remoti per l'accensione e più in generale per il controllo di differenti dispositivi. Possibili applicazioni pratiche sono il controllo di un relè per l'apertura di un cancello oppure l'accensione del condizionatore di casa; sarà inoltre possibile ricevere informazioni sullo stato dei pin di ingresso dell'EM1000 collegati, ad esempio, a sensori di temperatura, pressione, o altro ancora. ➤



La Figura 1 mostra come l'EM1000 può essere utilizzato, grazie alla sua interfaccia Ethernet 10/100BaseT, come un qualunque altro host su una rete LAN. In questo modo qualsiasi dispositivo (elettrodomestico, sensore, servocomando) potrà facilmente essere interfacciato ad una rete Ethernet e quindi controllato da remoto.

La semplice pressione di un tasto dal PC (evidenziato in rosso in Figura 1) potrebbe, ad esempio, determinare l'accensione del condizionatore o l'apertura del cancello, oppure rilevare lo stato del pulsante (premuto o meno).

Descrizione del codice

Poiché si tratta della prima applicazione pratica, abbiamo deciso di iniziare con qualcosa di semplice ma con un senso logico, ossia un esempio di codice che consente di interagire, tramite una semplice interfaccia Client, con il modulo EM1000. Il Listato 1 riporta il suddetto codice, mentre in Fig. 2 è riportato il diagramma di flusso che fornisce una prima idea su come è strutturato il flusso delle operazioni eseguite.

Le parti fondamentali si possono riassumere come segue:

- configurazione iniziale del modulo,
- attesa di comandi inviati dal software Client,
- attesa di variazioni delle eventuali linee configurate come input.

In questa prima applicazione abbiamo utilizzato fondamentalmente gli oggetti relativi alla gestione dei dati su protocollo TCP/IP, per la ricezione ed il trasferimento: stiamo parlando degli oggetti *Net* e *Sock* dei quali, tra poco, descriveremo il funzionamento, in modo da capire meglio a cosa

servono e come possono essere utilizzati all'interno del codice.

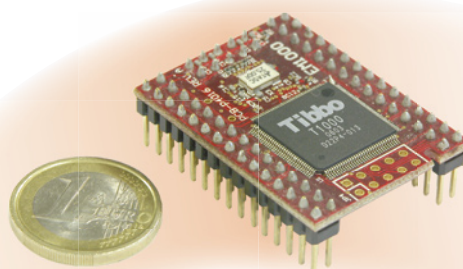
L'oggetto Net

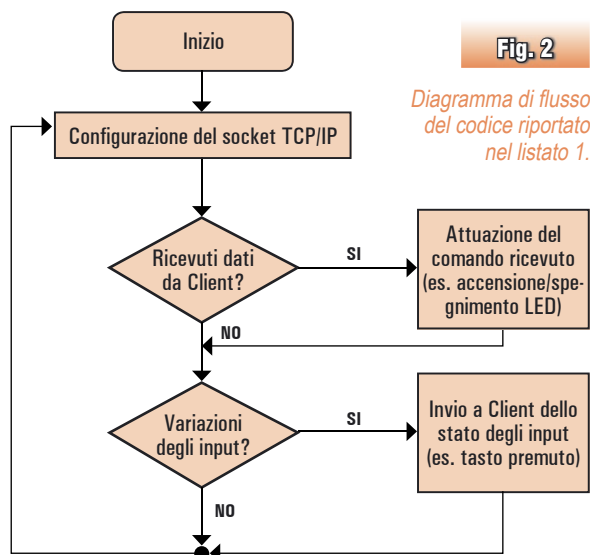
Rappresenta l'interfaccia Ethernet del dispositivo e ne specifica i relativi parametri. Esso, comunque, non è responsabile della trasmissione e ricezione dei datagrammi (pacchetti TCP/IP). Più precisamente, con l'oggetto *Net* è possibile:

- verificare se il NIC (Network Interface Controller) funziona correttamente;
- recuperare l'indirizzo MAC del dispositivo;
- impostare l'indirizzo IP dell'interfaccia;
- definire il gateway predefinito;
- avere notifica sullo stato di collegamento del cavo Ethernet e verificarne lo stato del link;
- avere notifica di un eventuale overflow nel controllore di rete (NIC).

L'oggetto Sock

Ciascuna connessione creata tra il modulo Tibbo e qualunque applicazione di gestione residente su un PC è chiamata *socket*. Per gestire tale connessione è utilizzato l'oggetto *Sock*. Con il TiOS (Tibbo Operative System – il sistema operativo che “gira” sul modulo) sono consentite fino ad un massimo di 16 connessioni (leggasi anche *socket*), TCP o UDP, contemporanee. Il valore





Listato 1

Di seguito riportiamo il codice sorgente del dispositivo: dopo aver inizializzato registri e parametri, il programma esegue il "main loop" che mantiene la posizione del servo, controlla gli ingressi e fa eseguire i movimenti.

```

1.  sub on_sys_init
2.  net.ip="192.168.1.95"
3.  net.netmask="255.255.255.0"

4.  sock.num=0
5.  sock.rxbufrrq(4)
6.  sock.txbufrrq(4)

7.  sys.buffalloc

8.  sock.num=0

9.  sock.protocol=PL_SOCKET_PROTOCOL_TCP
10. sock.localportlist="2000,3000"

11. sock.inconmode=PL_SOCKET_INCONMODE_ANY_IP_ANY_PORT
12. end sub

13. sub on_sock_data_arrival
14. stato = sock.getdata(6)
15. num_led = val(stato)

16. io.num = 24 + num_led
17. io.enabled = YES
18. if (num_led >= 4096) then
19.     io.state = HIGH
20. else
21.     io.state = LOW
22. end if
23. beep.divider = num_led
24. beep.play("B-B-B", PL_PAT_CANINT)
25. end sub

26. sub on_button_pressed
27. sock.setdata("Tasto Premuto")
28. sock.send
29. end sub
  
```

potrebbe, comunque, essere inferiore a seconda dell'utilizzo della memoria. Ogni socket ha tutte le impostazioni che tradizionalmente ci si aspetta di trovare, come numero della porta di destinazione, protocollo ed altri parametri simili. Oltre a ciò, presenta molte altre funzionalità che riducono significativamente la quantità di codice che

è necessario scrivere: tra le tante, citiamo come esempio la possibilità di restringere il numero di connessioni effettuate verso il dispositivo, filtrando automaticamente i messaggi in ingresso.

Il socket implementa anche le funzionalità di Web Server (che utilizzeremo nelle prossime puntate, in cui vedremo degli esempi di applicazioni di gestione tramite interfaccia web). Infatti, ogni socket può trasportare sia una connessione dati "normale" che una basata su protocollo HTTP (Hyper Text Transfer Protocol, usato pesantemente nelle pagine Web). È importante non confondere l'oggetto *Sock* con quello *Net*; quest'ultimo, come si è già detto, rappresenta l'interfaccia Ethernet (ossia il *livello fisico*, ovvero solo la parte di circuito che si occupa del collegamento elettrico), mentre l'oggetto *Sock* è responsabile della comunicazione IP (TCP o UDP, ossia il *livello di link*). Di seguito abbiamo riportato le caratteristiche principali dell'oggetto *Sock*:

- supporta i protocolli TCP, UDP e HTTP;
- dispone di un ampio numero di proprietà che definiscono quali host possono connettersi all'EM1000, se il broadcast (messaggio indirizzato a tutta la rete) deve essere supportato, quali sono le porte in ascolto associate con il socket;
- fornisce informazioni dettagliate relativamente allo stato del socket, descrivendoli con ben trenta differenti stati associati;
- supporta il funzionamento asincrono ed è in grado di gestire eventi separati sia per la ricezione che per la trasmissione dei dati;
- rileva automaticamente la condizione di buffer overflow;
- permette di variare le dimensioni dei buffer di ricezione e trasmissione, in modo da ottimizzare l'uso della RAM;
- dispone della funzione di "buffer shorting", che permette di scambiare velocemente dei dati tra l'oggetto *Sock* e altri, come l'oggetto *Ser* (necessario per la gestione delle porte seriali).

Configurazione iniziale: l'evento *on_sys_init()*

Da questo momento è necessario fare riferimento al Listato 1: notate che abbiamo voluto assegnare un identificativo numerico ad ogni riga di codice, in modo da facilitare l'identificazione dei vari comandi, impartiti durante l'esecuzione del programma.

La configurazione del modulo è eseguita nella subroutine *on_sys_init* (riga 1), richiamata automaticamente dal dispositivo ogni volta che viene avviato; essa è utilizzata, in generale, per ➤

la configurazione di qualsiasi oggetto utilizzato. Nel caso in esame serve ad inizializzare gli oggetti *Net* e *Sock*. La prima istruzione eseguita (riga 2) è relativa all'assegnazione dell'indirizzo IP del modulo, tramite la *proprietà IP* dell'oggetto *Net*. L'informazione deve essere fornita come una stringa, nella forma classica; ad esempio, se si utilizza l'EM1000 all'interno di una *subnet* 192.168.1.x, si potrà utilizzare uno qualsiasi degli indirizzi compresi tra 192.168.1.1 e 192.168.1.254 (a patto, ovviamente, che tale indirizzo non sia già utilizzato all'interno della subnet). Se il valore specificato non è corretto, il compilatore provvede a correggerlo automaticamente al valore più vicino: ad esempio, se fosse specificato "192.168.1.0", dopo la correzione il valore diventerebbe "192.168.1.1". È importante ricordare, anche se potrebbe sembrare scontato, che la variazione dell'indirizzo può essere effettuata solo se l'oggetto si trova nello stato *idle* (PL_SSTS_CLOSED), ossia di inattività.

Un'altra proprietà importante nella configurazione di un host all'interno di una rete LAN è la *Subnet Mask*: essa è fondamentale per un dispositivo (ad esempio un PC o l'EM1000) quando questi deve comunicare con un altro indirizzo IP; la subnet mask, infatti, fa capire al dispositivo se deve instradare i pacchetti verso il gateway della sua rete locale oppure usare l'indirizzo di rete locale del destinatario.

La riga 3 riporta l'istruzione utilizzata per impostare questa proprietà: il suo valore deve essere lo stesso di quello utilizzato nella propria sottorete (nell'esempio è riportato 255.255.255.0, ma potrebbe anche essere 255.255.0.0 o qualsiasi altro, in funzione della sottorete), altrimenti il trasferimento del pacchetto dati non può essere effettuato.

Si è detto che si possono impostare fino ad un massimo di 16 socket: essi sono enumerati a partire dal valore 0 (che è anche quello predefinito).

La proprietà per selezionare, ad esempio, il socket 0 è la seguente:

sock.num=0

Si è detto anche che il numero massimo di connessioni disponibili può essere condizionato anche dalla quantità di memoria. Per conoscere il valore esatto è possibile utilizzare la proprietà:

sock.numofsock - 1

Le istruzioni alle righe 5 e 6 servono per impostare la dimensione del buffer in ricezione e trasmissione. Con il metodo *Rxbuffrq* si decide quante pagine riservare per la ricezione; la stessa cosa vale per il metodo *Txbuffrq* che, però, agisce sulla trasmissione. La dimensione di una pagina è pari a 256 byte e nel Listato 1 sono state riservate quattro pagine ossia $4 \times 256 \text{ byte} = 1 \text{ kbyte}$. In realtà, questa non è la dimensione netta utilizzabile, poiché è necessario togliere 16 byte destinati a variabili interne e pertanto la memoria effettivamente utilizzabile è di 1008 byte. In ogni caso, l'allocazione vera e propria della memoria avviene solo quando si richiama il metodo *sys.buffalloc* (riga 7). Il programma procede con l'impostazione delle proprietà principali del controllo *Sock*. Notate che nel codice è stata inserita nuovamente l'istruzione per selezionare il socket (riga 8), anche se in questo caso non vi era alcuna necessità. Questo non è un errore: è infatti buona norma selezionare il socket prima di impostarne qualsiasi valore o richiamare qualsiasi metodo che agisca su di esso, poiché nella gestione contemporanea di più oggetti *Sock* si potrebbero commettere facilmente degli errori. Le istruzioni 9, 10 e 11 consentono rispettivamente di selezionare il tipo di protocollo utilizzato (TCP), le porte su cui vengono effettuate le connessioni (2000 e 3000) e specificare da quali host il modulo accetta tali connessioni. Se si utilizza la costante "PL_SOCK_INCONMODE_ANY_IP_ANY_PORT", qualunque host

Tabella 1 - Costanti che si possono specificare nella proprietà *Incommode* dell'oggetto *Sock*

Costante	Descrizione
PL_SOCK_INCONMODE_NONE	Nessuna connessione accettata
PL_SOCK_INCONMODE_SPECIFIC_IPPORT	Connessioni accettate solo dall'indirizzo e porta specificata nelle proprietà <i>sock.targetip</i> e <i>sock.targetport</i> .
PL_SOCK_INCONMODE_SPECIFIC_IP_ANY_PORT	Connessioni accettate da qualsiasi IP.
PL_SOCK_INCONMODE_ANY_IP_ANY_PORT	Connessioni accettate da qualsiasi IP e porta.

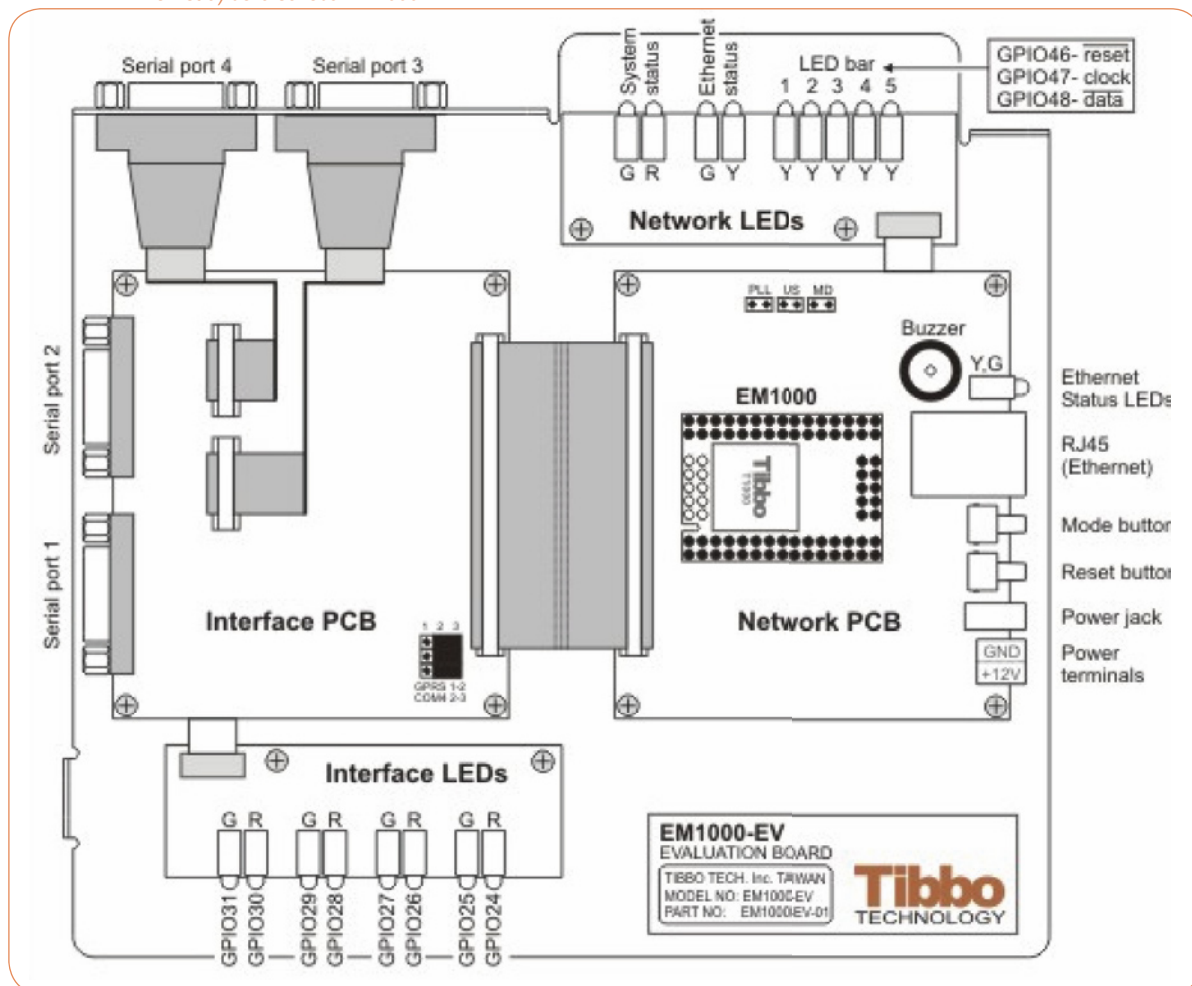
potrà effettuare l'accesso a Tibbo. La Tabella 1 riporta tutte le altre possibili costanti, con il relativo significato. Particolarmente interessante è la possibilità di limitare le connessioni solo a specifici indirizzi IP: ciò, oltre che limitare il sovraccarico del modulo, garantisce una maggiore sicurezza, infatti potranno avere accesso alla gestione del modulo solo gli indirizzi IP autorizzati. Va detto che non si tratta di una sicurezza totale poiché non è difficile per un host assumere l'indirizzo IP che ha il permesso ad accedere: per aumentare la sicu-

attesa di due eventi, secondo quanto riportato nel Listato 1 ossia:

- la ricezione di un carattere da rete Ethernet; in risposta a questo evento, il modulo modificherà lo stato dei led presenti sulla scheda di sviluppo, generando opportuni suoni;
- la rilevazione della pressione del tasto M/D presente sulla demoboard, che determina l'inizio dell'evento al Client con cui è stata stabilita la connessione; in questo modo si realizzerà un semplice "conta impulsi" che, dal punto di vi-

Fig. 3

I led comandati dal software Client sono quelli riportati nella sezione "Interface LED" (in basso a sinistra) della scheda EM1000 EV.



rezza possono essere implementate altre procedure quali l'impiego di una password di accesso al sistema. A questo punto, comunque, il modulo è configurato per accettare una connessione di tipo TCP, sulle porte 2000 o 3000 da qualsiasi host presente su un rete con subnet 192.168.1.x e mask 255.255.255.0.

Ricezione dati da rete:

l'evento on_sock_data_arrival ()

Dopo aver configurato il dispositivo, esso resta in

sta didattico è di facile realizzazione mentre da quello pratico, potrebbe avere applicazioni reali quali il conteggio delle persone che transitano attraverso un accesso controllato.

Gestire un evento con Tibbo BASIC è molto semplice ed intuitivo: è sufficiente selezionare, nella finestra *Project* e precisamente nella sezione *Platform Events*, la voce *on_sock_data_arrival*. In questo modo viene automaticamente creata una sezione in cui è possibile inserire il codice per gestire l'evento di ricezione dati dal socket ➤

Tabella 2

Caratteri speciali inviati dal software Client per indicare al modulo EM1000 quale led accendere o spegnere

Carattere	Funzione	Carattere	Funzione
&h0000	Accensione led 1	&h1000	Spegnimento led 1
&h0001	Accensione led 2	&h1001	Spegnimento led 2
&h0002	Accensione led 3	&h1002	Spegnimento led 3
&h0003	Accensione led 4	&h1003	Spegnimento led 4
&h0004	Accensione led 5	&h1004	Spegnimento led 5
&h0005	Accensione led 6	&h1005	Spegnimento led 6
&h0006	Accensione led 7	&h1006	Spegnimento led 7
&h0007	Accensione led 8	&h1007	Spegnimento led 8

configurato. Il software Client, dopo aver stabilito la connessione con l'EM1000, invierà (su richiesta dell'utente) dei caratteri speciali per comunicare quale led deve accendere o spegnere. I led a cui si fa riferimento sono quelli riportati nell'interfaccia a led della scheda (vedere Fig. 3).

Il fulcro di tutta la subroutine *on_sock_data_arrival* è costituito dal metodo *get_data* dell'oggetto *sock*. Esso, in pratica, si occupa di estrarre i dati dal buffer di ricezione (si ricordi che la dimensione massima utilizzabile di questa area di memoria è stata impostata a 1008 byte). Il prototipo della funzione è il seguente:

sock.getdata(maxinplen as word) as string

dove *maxinplen* rappresenta il numero di byte da estrarre dal buffer. Attenzione ad un particolare: i dati sono estratti in maniera permanente dal buffer e ciò significa che vengono fisicamente rimossi dal buffer per liberare spazio.

Le operazioni successive contenute nell'evento di ricezione sono usate per decodificare la stringa ricevuta e attuare il comando, secondo quanto riportato nella Tabella 2. Per comandare le linee di I/O della demoboard (e quindi i led) si utilizza l'oggetto *io*, già descritto nella terza puntata del corso. Anche l'utilizzo di questo oggetto è davvero semplice: dopo aver selezionato la linea con la proprietà *num*, si abilita la linea e infine si decide il suo stato (*HIGH* o *LOW*).

Intercettazione degli input:

l'evento on_button_pressed

Dopo aver esaminato come avviene il flusso dati dal Client al modulo, esaminiamo ora il caso inverso, ossia l'invio dati dal modulo al Client.

Il listato del nostro esempio (Listato 1) potrebbe sembrare molto banale (si tratta della pressione di un tasto e della relativa intercettazione da parte del Client), ma in realtà la stessa filosofia può essere attuata per inviare lo stato di un sensore o qualunque altra informazione che si vuole comunicare (periodicamente oppure in maniera asincrona) ad un centro di raccolta dati. Anche in questo caso si può aggiungere l'evento *on_button_pressed*, con la stessa procedura seguita precedentemente.

Questo caso ci permette di approfondire l'uso di un altro metodo dell'oggetto *Sock*: *setdata*. Il relativo prototipo è:

sock.setdata(byref txdata as string)

La funzione trasferisce il contenuto di una stringa direttamente nel buffer di trasmissione. È importante notare che se il buffer TX non dispone di sufficiente spazio, la stringa verrà troncata e quindi la trasmissione sarà corrotta. Il fatto che i dati siano trasferiti nell'area di trasmissione non significa, però, che verranno inviati in rete immediatamente: la trasmissione, infatti, avviene solo quando si invoca il metodo *send*.

Come sarà illustrato di seguito, in questo modo è molto semplice realizzare un conta impulsi.

Generazione di suoni: l'oggetto beep ()

L'evaluation kit oggetto di questo articolo è anche dotato di un buzzer che consente al modulo EM1000 di generare suoni all'interno della banda audio. Tali segnalazioni possono essere utilizzate per gli scopi più diversi.

L'oggetto incaricato di gestire tali suoni è chiamato *beep*. È costituito da una *proprietà* e un *metodo*, rispettivamente *divider* e *play*. *Divider* im-



Fig.4 - Schermata iniziale per la creazione di un nuovo progetto con Tibbo IDE.

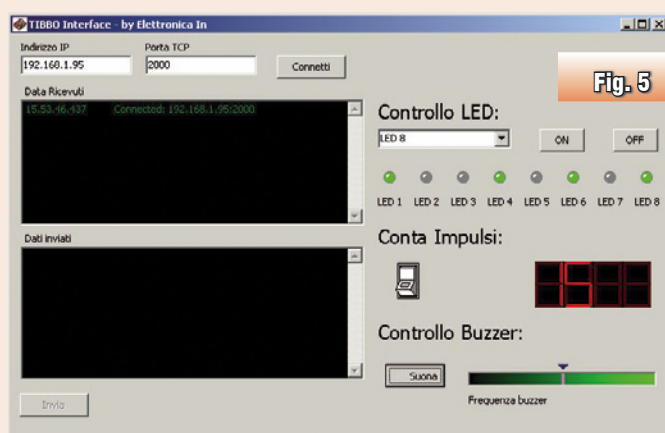


Fig.5 - Interfaccia dell'applicazione Client per il telecontrollo del modulo EM1000 (vedi Listato 2).

posta la frequenza del suono, secondo la formula seguente:

$$base_freq/(2*beep.divider)$$

dove *base_freq* rappresenta la frequenza propria del modulo.

La funzione *play* ha la seguente espressione:

beep.play(byref pattern as string, patint as pl_beep_int)

Un esempio di utilizzo è riportato all'interno dell'evento *on_sock_data_arrival*, alle righe di codice 21 e 22. Per configurare la sequenza dei suoni da effettuare, si passa alla funzione un *pattern* speciale costituito dai seguenti caratteri:

- '-': il buzzer è spento;
- 'B' oppure 'b': il buzzer è acceso;
- '~': il buzzer è mandato in esecuzione ciclica;
- '*': serve a raddoppiare la velocità di esecuzione del pattern.

Prove pratiche: creazione progetto e interfaccia Client

L'applicazione descritta, seppure molto semplice, rappresenta il punto di partenza per lo sviluppo di un codice molto più complesso e completo. Quello che sorprende è il numero ridotto di righe di codice necessarie per realizzare un'applicazione completamente funzionante su rete Ethernet.

È stato sufficiente realizzare una routine di inizializzazione, scrivere il codice necessario per la gestione di due eventi e imparare le nozioni base per l'uso di qualche oggetto. A completamento di ciò è da evidenziare anche l'ottima organizzazione dell'help in linea che fornisce in maniera molto pratica una guida a tutti i metodi, proprietà ed

eventi. Per provare l'applicazione descritta, è necessario aprire l'ambiente di sviluppo Tibbo IDE e creare un nuovo progetto così come descritto nella terza puntata del corso. Come promemoria, riportiamo di seguito la sequenza di operazioni da eseguire, evitando però di spiegarne il significato. Nella schermata *New Project* (Fig. 4):

- selezionare piattaforma *EM1000*;
- definire *Empty Project* come tipo di progetto;
- assegnare un nome al progetto;
- selezionare una cartella di destinazione;
- selezionare il modulo EM1000 tramite il tasto *Select*: tale operazione è possibile solo dopo aver collegato un cavo di rete Ethernet tra il PC e modulo. Ricordate che è buona norma (semplicemente per questione di logica) usare un cavo diritto per effettuare un collegamento modulo-switch/router e uno incrociato per un link tra modulo-PC, nonostante l'EM1000 sia in grado di riconoscere automaticamente il tipo di cavo;
- copiare il Listato 1 ed incollarlo nel progetto appena creato;
- selezionare *Project Build and Upload*. In questo modo il codice viene trasferito sul modulo;
- avviare l'applicazione selezionando *Debug Run*.

A questo punto manca solo l'applicazione Client per collegarci tramite Ethernet al modulo EM1000. Si può utilizzare qualsiasi software di test, realizzato con un qualunque linguaggio di programmazione. Quello proposto nell'articolo, da installare sul vostro PC, è stato realizzato in Delphi, utilizzando semplici controlli. La Fig. 5 mostra la schermata del software: relativamente ad esso, parte del sorgente è riportato nel riquadro

di questa pagina (Listato 2); il sorgente completo e il file eseguibile, invece, sono scaricabili gratuitamente dal nostro sito. Per stabilire la connessione occorre inserire l'indirizzo IP 192.168.1.95 nel campo *IP Address*, inserire come *Porta TCP* di comunicazione 2000 (oppure 3000) e fare clic sul tasto *Connect*. Quando la connessione è stata stabilita, compare la scritta *Connected* nel campo *Incoming Data*. Da questo momento l'EM1000 è pronto ad accettare i comandi descritti in seguito. È possibile provare ad inviare il comando di accensione, ad esempio del led 1:

- selezionare LED 1 dalla relativa *combo box*;
- fare clic sul tasto ON. A questo punto si accenderà il LED 1 sulla scheda di sviluppo.

Premendo il pulsante M/D presente sulla scheda di sviluppo si può verificare il funzionamento del conta impulsi. Infatti, sul display a 7 segmenti viene visualizzato il numero di volte che è stato premuto il pulsante. Inoltre, si può apprezzare anche il controllo buzzer. Prima di premere il tasto

Suona è necessario scegliere il valore di frequenza da generare, tramite l'apposito cursore.

Note conclusive

Giunti a questo punto, avete appreso le basi per utilizzare il modulo EM1000 come gestore remoto di ingressi e uscite: come abbiamo già avuto modo di sottolineare, il linguaggio Tibbo BASIC è stato ottimizzato al massimo, in modo da permettere la realizzazione di applicazioni complesse senza dover scrivere migliaia di righe di codice, come avveniva ad esempio in Assembler. Nel corso delle prossime puntate, impareremo a realizzare un convertitore seriale-Ethernet. Tale dispositivo è essenziale se si vuole connettere porte seriali su rete intranet o Internet. Mantenendo la stessa semplicità di programmazione, analizzeremo altri oggetti disponibili in TIDE per il controllo della seriale: in particolare, impareremo l'uso dell'oggetto *ser* ed effettueremo esempi di trasmissione da Ethernet a seriale e viceversa.

Listato 2

Abbiamo qui riportato alcune parti significative (il sorgente completo, scaricabile gratuitamente dal nostro sito www.elettronica.in.it, è molto lungo) del codice Delphi utilizzato per la realizzazione dell'interfaccia Client descritta nell'articolo (Fig. 5). In particolare, vi mostriamo alcune routine impiegate nella gestione dei socket TCP. *ConnectActExecute* apre e chiude un socket quando viene premuto un pulsante; *TcpClientReceive* scrive in memoria un dato che deve essere inviato. *SendActExecute* invia il dato alla porta remota; *TcpClientConnect* impedisce di cambiare l'indirizzo IP se dei socket sono attivi. Infine *FormClose Event* chiude tutte le procedure attive quando la schermata principale del programma viene chiusa.

```
{-----*
* ConnectActExecute:
* Connect/disconnect when button is pushed *
*-----*
procedure TfrmTCPMain.ConnectActExecute
  (Sender: TObject);
begin
  if TcpClient1.Connected
  then TcpClient1.Disconnect
  else
  begin
    TcpClient1.RemoteHost := self.IP_Address.Text;
    TcpClient1.RemotePort := self.Port.Text;
    screen.Cursor := crhourglass;
    try
      ConnectAct.Enabled := FALSE;
      TcpClient1.Connect;
    finally
      ConnectAct.Enabled := TRUE;
      screen.Cursor := crdefault;
    end;
  end;
end;
{-----*
* TcpClient1Receive:
* Place new data into the memo
*-----*
procedure TfrmTCPMain.TcpClient1Receive
  (Sender: TObject; Buf: PAnsiChar;
   var DataLen: Integer);
var
  Astr: string;
  Achar: Pchar;
  index: integer;
begin
  Astr := '';
  Achar := buf;
  for index := 1 to dataLen do
  begin
    Astr := Astr + char(Achar^);
    INC(Achar);
  end;
  self.incomingData.Lines.add(Astr);
end;
end;
{-----*
* SendActExecute:
* Send data to the remote port
*-----*
procedure TfrmTCPMain.SendActExecute
  (Sender: TObject);
begin
  TcpClient1.SendBuf
    (pchar(self.SendData.Text)^,
     length(SendData.Text), 0);
  SendData.Clear;
end;
{-----*
* TcpClient1Connect:
* Let's make it so that you can't change
* the IP address and port while connected *
*-----*
procedure TfrmTCPMain.TcpClient1Connect
  (Sender: TObject);
begin
  IP_Address.ReadOnly := TRUE;
  Port.ReadOnly := TRUE;
  IncomingData.Lines.add
    (formatdatetime('HH:NN:SS.zz', now)
     + #9 + 'Connected:
     + TcpClient1.RemoteHost
     + ':' + tcpClient1.RemotePort);
  WSASyncSelect(tcpClient1.Handle,
    self.Handle,
    WM_SOCKET_CLOSED, FD_CLOSE);
end;
{-----*
* FormClose Event:
* When the form is closed make sure
* everything is disconnected
*-----*
procedure TfrmTCPMain.FormClose(Sender:
  TObject; var Action: TCloseAction);
begin
  if tcpClient1.Connected
  then tcpClient1.Disconnect;
end;
```