

Mensile di elettronica applicata, attualità scientifica, novità tecnologiche.

€ 5,00

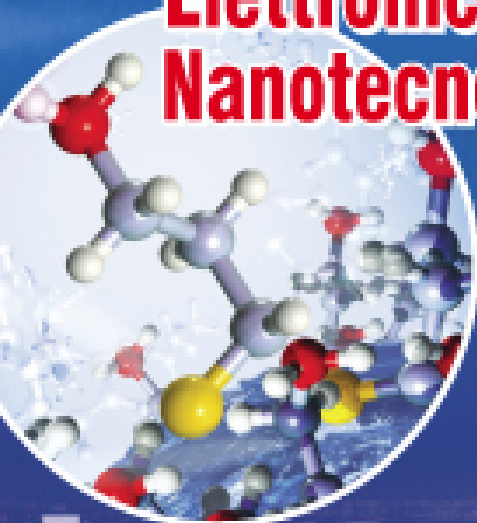
Elettronica In

oltre l'elettronica

www.elettronica.in.it

Anno XV - n. 142
Novembre 2009

Elettronica & Nanotecnologie



Radiocomando 12 CANALI



Lifter: Costruiamo un piccolo UFO

- Trasmettitore e ricevitore per radiomodelli
- Lettore di transponder stand-alone
- Misuratore di consumo di elettricità con ZigBee
- Controllo volume
- Corso FPGA

ESCLUSIVO!
Corso Wireless CPU

CONTROLLO VOLUME CON TELECOMANDO INFRAROSSI

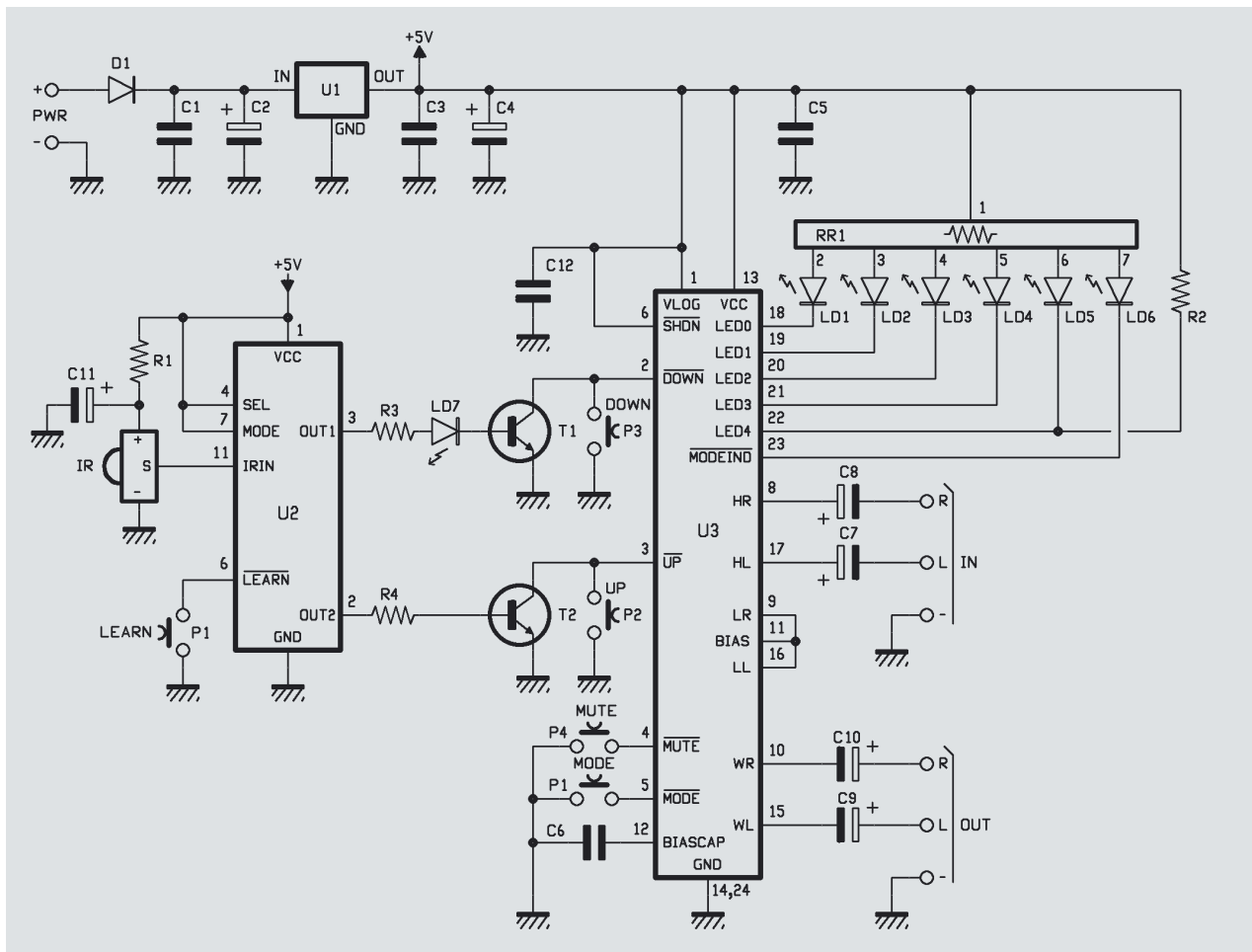
di ALESSANDRO SOTTOCORNOLA



Inserito in una linea audio mono o stereo, permette di regolare con precisione il livello e il bilanciamento tramite qualsiasi telecomando a raggi infrarossi per TV, DVD, videoregistratore.

Se avete costruito un amplificatore audio o disponete di un apparato che ha un'uscita audio di cui non è possibile regolare il volume, può certamente farvi comodo il progetto descritto in queste pagine: si tratta di un controllo di volume con telecomando a infrarossi pensato per essere inserito in una catena audio dove la regolazione non ci sia o risulti possibile solamente sul posto. Il circuito si può inserire ad esempio all'usc-

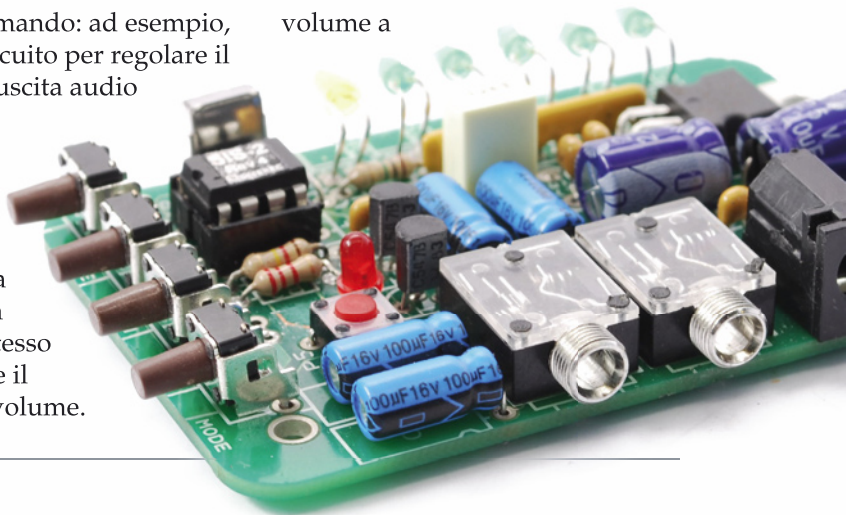
ta audio di un lettore DVD o di un videoregistratore sprovvisti di comando del volume o tra l'uscita del preamplificatore e l'ingresso del finale BF. Dato che oggi praticamente tutti i videoregistratori e comun-



que i lettori DVD o gli amplificatori audio sono stereofonici, il nostro circuito è in versione stereo e regola il volume simultaneamente sui due canali. Ma la cosa sicuramente più interessante è che per gestirlo non serve alcun trasmettitore, dato che il dispositivo qui descritto potete usarlo con qualsiasi telecomando IR che avete già in casa; infatti lo abbiamo realizzato usando uno speciale circuito integrato (già descritto nel fascicolo n° 134) capace di apprendere i codici di un massimo di due tasti (dello stesso trasmettitore o uno di un TX e uno dell'altro) di qualsiasi telecomando. Tutto quello che dovete fare per usarlo è identificare due tasti (uno per alzare e l'altro per abbassare il volume) di un telecomando (anche di uno di quelli universali) per TV, SAT, VCR

ecc., possibilmente corrispondenti a funzioni mai usate (sovente i trasmettitori hanno tasti che con alcune versioni degli apparecchi di cui sono a corredo non servono) e svolgere la procedura di abbinamento al circuito, semplice e immediata, che descriveremo più avanti. Fatto l'apprendimento, potrete telecomandare il volume semplicemente usando lo stesso telecomando: ad esempio, se usate il circuito per regolare il volume dell'uscita audio di un videoregistratore potete impiegare il tasto che comanda una funzione non usata dello stesso per effettuare il controllo di volume.

Tutto ciò grazie a un circuito estremamente semplice, realizzato da due integrati: il SIS2 della SIMEREC (<http://www.simerec.com>) progettato appositamente per funzionare da ricevitore ad autoapprendimento, in grado di riconoscere le codifiche adottate dai TX dei prodotti solitamente impiegati nelle case; il MAX5486, che è un controllo di volume a



doppio potenziometro digitale comandabile mediante livelli logici o pulsanti sul posto.

SCHEMA ELETTRICO

Ma andiamo subito a vedere com'è composto il circuito: il rispettivo schema elettrico mostra il SIS-2 nella sua classica configurazione, abbinato a un ricevitore integrato per infrarossi, che gli "passa" il segnale ricevuto per via ottica dal telecomando. Completa il tutto il MAX5486, le cui linee UP e DOWN sono gestite da due transistor pilotati in base ciascuno da un'uscita del SIS2.

Il ricevitore integrato (IR) è un modulo contenente un fotodiodo e un amplificatore a transistor che eleva il livello della tensione elettrica ricavata dal fotodiodo e rende disponibile, tra il proprio piedino S e massa (-) la componente corrispondente al codice inviato dal trasmettitore.

L'integrato SIS2 è sostanzialmente un microcontrollore programmato per analizzare il segnale elettrico in arrivo dall'uscita del modulo IR, mediante il piedino IRIN (11), quindi confrontarlo con le stringhe di dati che ha preventivamente appreso; l'apprendimento si effettua quando si preme il tasto P1 (LEARN) allorché il chip memorizza il codice ricevuto. Siccome è predisposto per comandare due distinte uscite (OUT1 e OUT2) prevede la possibilità di acquisire e memorizzare due codici distinti oppure due volte lo stesso; nel primo caso ogni codice corrisponde a un tasto del telecomando, mentre nel secondo entrambi i codici appresi riguardano il medesimo pulsante.

Il modo di funzionamento delle uscite può essere definito tramite i due piedini di configurazione, che sono 4 (MODE) e 7 (SEL): quando il primo è ad 1

logico e il secondo a zero, entrambi i canali lavorano in modo bistabile, nel senso che la rice-

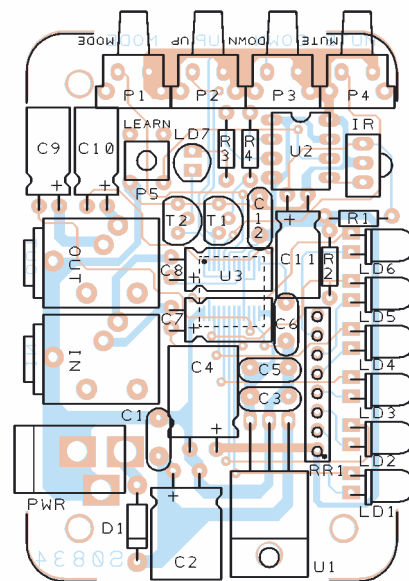
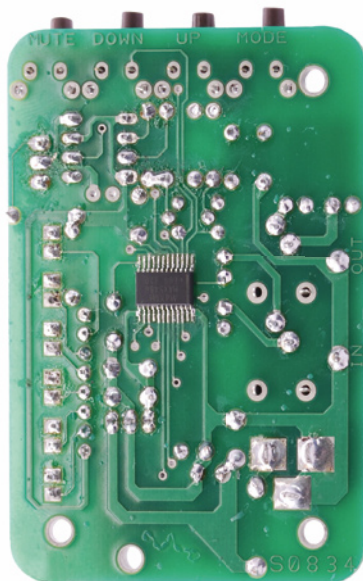
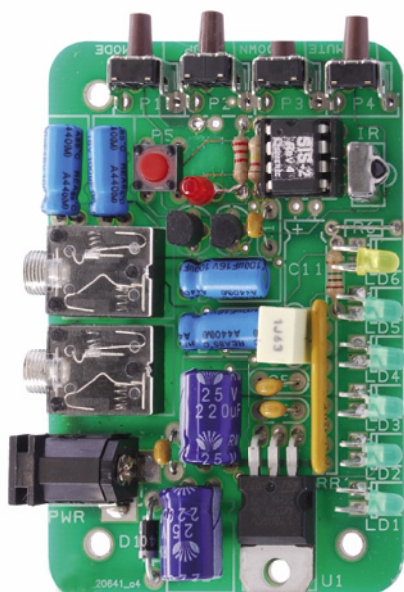
zione del codice 1 inverte ogni volta lo stato dell'OUT1 e quella del codice 2 inverte la condi-



L'iPod si affida al grammofono

Ricordate gli antichi grammofoni, precursori del moderno giradischi? Riproducevano il suono registrato nei dischi di vinile senza bisogno di corrente elettrica. Ma come funzionavano? Semplice: sfruttavano l'effetto di amplificazione sonora prodotto dalle trombe acustiche. In essi la puntina che vibrava mentre sotto le scorreva il disco era collegata ad una lamina, vibrante anch'essa; il suono così prodotto poteva essere percepito in un salone grazie al fatto che la membrana era collocata all'imboccatura di una tromba, ossia nella sua gola. Dalla parte più larga, le vibrazioni acustiche uscivano molto più intense. Questo fenomeno, che è dovuto al caricamento acustico operato dai condotti a sezione esponenziale (trombe) viene sfruttato da tempo anche negli altoparlanti professionali caricati a tromba. Da poco è stato impiegato anche dalla Mintpass (www.mintpass.com) per realizzare un dispositivo capace di amplificare il suono, ad esempio di un auricolare, senza necessità di corrente elettrica. E questo è il concetto su cui si basa Mint Horn, una tromba che ricorda quelle dei grammofoni, in perfetto stile iPod (bianco), che accoglie un auricolare (il dispositivo è venduto in coppia, per l'uso con una coppia di auricolari). Un'ottima soluzione quando serve più livello sonoro e non c'è alimentazione da rete, che però permette una fedeltà sonora limitata dal fatto che le trombe rispondono bene solo in un certo campo di frequenze.





Elenco Componenti:

R1: 100 ohm
R2: 1 Mohm
R3: 2,2 kohm
R4: 330 ohm
R5: 120 kohm
RR1: Rete resistiva 8x470 ohm + C
C1, C3, C5: 100 nF multistrato
C2, C4: 220 μ F 25 VL elettrolitico
C6: 1 μ F 63 VL poliestere
C7 $\div$ C10: 100 μ F 16 VL elettrolitico

C11: 2,2 μ F 16 VL elettrolitico
C12: 100 nF multistrato
U1: 7805
U2: SIS2
U3: MAX5486EUG+
T1, T2: BC547
LD1 $\div$ LD5: led 3 mm verde
LD6: led 3 mm giallo
LD7: led 3 mm rosso
IR: IR38DM

D1: 1N4007
P1 $\div$ P4: Microswitch 90°
P5: Microswitch

Varie:
- Zoccolo 4+4
- Presa jack 3,5mm da CS (2 pz.)
- Plug alimentazione
- Contenitore plastico WK-3.29
- Circuito stampato

zione dell'OUT2; in altre parole, trasmettendo una volta, l'uscita corrispondente si porta ad 1 logico, la seconda torna a zero, la terza va ad uno ecc.

Invece, fissando entrambe le linee a livello alto si ottiene il funzionamento impulsivo: ogni volta che arriva un codice, l'uscita corrispondente si porta ad 1 logico e vi resta fino al termine della trasmissione; è chiaro che in questa situazione non è possibile attivare contemporaneamente le uscite, perché occorrerebbe trasmettere simultaneamente

i due codici, cosa che impedirebbe al SIS2 di distinguere uno dall'altro. Una terza modalità di funzionamento del chip è quella che il costruttore chiama Discrete Button Mode, che prevede l'uso della sola OUT1 (OUT2 è inattiva): tale uscita funziona in modo bistabile e viene comandata da entrambi i codici scelti per i due canali; in breve, trasmettendo il codice del canale corrispondente ad OUT1 questa uscita passa da zero a livello alto, e torna allo stato zero quando viene ricevuto il codice del canale corrispon-

dente ad OUT2. La modalità in questione si attiva quando sia il piedino 4 che il 7 vengono posti a zero logico.

Nella nostra applicazione il SIS2 gestisce le uscite in modo impulsivo, dato che ci serve comandare a tratti o in modo continuo i piedini UP e DOWN del MAX5486. Quest'ultimo è un chip pensato proprio per il controllo di volume negli impianti hi-fi, nei moderni TV e nei lettori mp3; realizza il controllo del volume mediante due potenziometri elettronici a variazione logarit-

mica da 40 kohm ciascuno, che avanzano a passi (sono previsti 32 livelli o passi) comandati da semplicissimi pulsanti. L'integrato contiene, oltre ai potenziometri digitali, la logica di controllo (comprendente anche cinque uscite per led di segnalazione) e di interfaccia verso i tasti, che permette di regolare, oltre al volume, il bilanciamento. Dispone inoltre di un comando di MUTE, che quando viene attivato tacita l'uscita, ovvero attenua il livello del segnale di ben 90 dB.

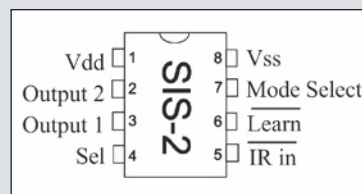
I tasti previsti nel circuito sono quattro: due per il controllo UP/DOWN in locale, uno per la tacitazione immediata (MUTE) e l'altro per decidere la modalità d'uso dei tasti UP/DOWN (MODE); quest'ultimo serve a decidere se con detti pulsanti regoliamo il volume o il bilanciamento. Per non sbagliare, un apposito led segnala quando è attiva la funzione di bilanciamento e quando, invece, UP e DOWN controllano il volume. Nella modalità bilanciamento, UP aumenta il livello del canale sinistro a discapito di quello del destro e DOWN incrementa il livello del canale destro diminuendo quello del sinistro.

Il nostro circuito è adatto a regolare il livello di segnali con ampiezza fino alla tensione di alimentazione meno 0,3 V, quindi, funzionando a 5 V singoli, ai suoi ingressi potete applicare fino a circa 1,6 Veff. Nell'applicazione qui descritta, quello che comandiamo mediante il telecomando a infrarossi è il solo volume; d'altra parte le uscite del SIS2 sono solo due. Abbiamo però previsto la possibilità di gestire localmente, tramite pulsanti, il controllo del bilanciamento e dell'eventuale MUTE. Siccome il MODE è collegato a un apposito pulsante (P1)

L'integrato SIS2

Si tratta di un riconoscitore in grado di apprendere la codifica di qualsiasi sistema di telecomando a infrarossi, progettato sia per sfruttare telecomandi di apparecchi esistenti, sia per realizzare sistemi di comando a distanza universali utilizzabili con più apparecchi.

Sostanzialmente il SIS-2 contiene un micro capace di identificare i formati più usati dai telecomandi di TV, VCR, DVD e simili e memorizzarli nella propria memoria EEPROM; lo stesso, in modalità normale quando riconosce un codice già memorizzato attiva le proprie uscite di conseguenza. Per rendere il componente più versatile sono previste tre funzioni di attivazione delle uscite, ossia monostabile, bistabile e Discrete Button Mode; il modo di funzionamento si definisce con due piedini di configurazione: 4 (MODE) e 7 (SEL). Con il primo ad 1 logico e il secondo a zero, entrambi i canali lavorano in modo bistabile, mentre con entrambe le linee a livello alto si ottiene il funzionamento impul-



sivo. La terza modalità di funzionamento (Discrete Button Mode) prevede l'uso della sola OUT1 (OUT2 è inattiva) che funziona in modo bistabile e viene comandata da entrambi i codici scelti per i due canali: trasmettendo il codice del canale corrispondente ad OUT1 passa da zero a livello alto, e torna allo stato zero quando viene ricevuto il codice del canale corrispondente ad OUT2. Questa modalità si ottiene con i due pin (4 e 7) a zero logico. Il componente è incapsulato in un contenitore plastico dip a quattro piedini per lato, si alimenta a 2÷5,5 V in continua e può erogare, da ciascuna delle due uscite di cui dispone, una corrente di 25 milliampere.

Modalità	Descrizione	dip 1	dip 1
bistabile normale	Alla ricezione di un codice già memorizzato, il canale corrispondente inverte la propria condizione rispetto a quella assunta al momento.	1	0
monostabile	Alla ricezione di un codice già memorizzato, l'uscita corrispondente si porta a livello alto; ricade quando smette la trasmissione.	1	1
Discrete Button Mode	Funziona solo OUT1; il codice 1 attiva, il codice 2 disattiva.	0	0

Modalità di funzionamento

Il ricevitore prevede tre distinte modalità di utilizzo facilmente definibili intervenendo sui dip-switch; la tabella seguente le riassume, correlate con l'impostazione da effettuare con dip1 e dip 2.

premo quest'ultimo possiamo impostare (localmente o col telecomando) il bilanciamento, quindi, fatto ciò, con una nuova pressione del P1 rimettiamo il dispositivo nella normale modalità di funzionamento, ossia nel controllo del volume. Quanto ai led, quello collegato a MODEIND segnala la modalità attiva al momento (spento=volume; acceso=bilanciamento) mentre gli altri cinque formano un indicatore della modalità di controllo del volume (tutti i led spenti corri-

spondono al minimo, mentre il massimo volume corrisponde a tutti i diodi illuminati) e un indicatore di bilanciamento nella modalità di controllo del bilanciamento (se è acceso il diodo collegato al pin LEDIND02, il bilanciamento è neutro, ossia a zero). Per abilitare i diodi luminosi, ognuno connesso a una resistenza limitatrice della corrente compresa nella rete resistiva RR1, abbiamo posto il piedino LEDIND04 al positivo mediante un resistore da 1 kohm; se non intendete usare



la barra di led corrispondente a LEDIND0÷LEDIND04, collegate a massa i pin corrispondenti. Il segnale audio stereo entra dai contatti R ed L IN, mentre quello regolato si preleva dai punti R ed L OUT; i due canali sono riferiti alla massa comune. Quanto all'alimentazione, il circuito richiede 9÷15 Vcc applicati al plug facente capo ai contatti + e - PWR; passato il diodo di protezione dall'inversione di polarità, la tensione arriva al condensatore elettrolitico C2, che insieme a C1, la filtra dai disturbi impulsivi e dal residuo d'alternata dell'alimentatore. Ad ottenere i 5 volt stabilizzati necessari al modulo ricevitore a infrarossi, all'integrato SIS2 e al MAX5486, provvede un regolatore di tensione 7805; i condensatori C3 e C4 filtrano la tensione stabilizzata di uscita del componente.

REALIZZAZIONE PRATICA

Giunti a questo punto, non ci resta che vedere gli aspetti costrut-

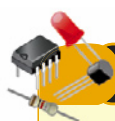
tivi, partendo dallo stampato, che va realizzato per fotoincisione seguendo la traccia lato rame che potrete scaricare gratuitamente dal nostro sito www.elettronica-in.it. Incisa e forata la basetta ramata, disponetevi e saldate i componenti in ordine di altezza, quindi partite da resistenze e diodi al silicio (prestando attenzione alla polarità di questi ultimi) e proseguite con la rete resistiva, lo zoccolo da 4+4 contatti per il micro U1, i pulsanti da c.s. e i condensatori (prima quelli non polarizzati e poi gli elettrolitici, che consigliamo di disporre sdraiati per ridurre l'ingombro). Disponete quindi i transistor e il regolatore integrato, oltre al modulo IR; quest'ultimo deve stare con la cupoletta in plastica rivolta all'esterno dello stampato. Ricordate che se utilizzate dispositivi come l'IR38 DM della Vishay dovete ridurre R1 a 10 ohm e non montare l'elettrolitico C11.

Detto ciò, inserite e saldate i led, piegando ad angolo retto i terminali di quelli che staranno sul bordo dello stampato (i verdi e il giallo...) e proseguite il montaggio con le due prese jack stereo da 3,5 mm per l'entrata e l'uscita audio e con la presa plug necessaria all'alimentazione. L'ultimo componente da montare e sicuramente quello che richiede più attenzione, è il MAX5486, dato che è in contenitore TSSOP (per montaggio superficiale, con passo dei piedini molto stretto); per riuscire nell'impresa dovete

usare un saldatore a punta sottile per integrati e del filo di lega saldante del diametro di 0,5 mm. Posizionate il MAX5486 in modo che i piedini siano al centro delle rispettive piazzole, quindi staginate almeno un pin per fermare il componente e poi proseguite con la stagnatura di tutti gli altri. Nell'inserire gli elementi polarizzati non dimenticate di dare, periodicamente, uno sguardo al piano di montaggio illustrato in queste pagine: vi aiuterà a non sbagliare.

Completato il montaggio e verificato che tutto sia al posto giusto, potete pensare a un contenitore adatto a racchiudere l'intero circuito; lo lavorerete in modo da ricavare una finestrella (da rivestire, magari, con del plexiglass trasparente rosso o violetto) in corrispondenza del sensore IR, oltre a quattro fori per i pulsanti che devono essere accessibili dall'esterno e altri sei, da 3 mm, per i led verdi e giallo che faranno da indicatore luminoso del volume o del bilanciamento. Ancora tre fori vi serviranno per far uscire le prese jack e il plug.

Una volta completato l'insieme potete provvedere all'apprendimento dei codici, ossia dei tasti del telecomando che vorrete abbinare al controllo;



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono di facile reperibilità. Il master del circuito stampato può essere scaricato gratuitamente dal sito della rivista. L'integrato Maxim MAX5486EUG+ costa 3,50 Euro, il sensore IR38DM costa 2,80 Euro mentre l'integrato SIS2 costa 15,00 Euro. I prezzi si intendono IVA compresa.

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

Il piccolo contenitore plastico deve essere forato su tre lati, per far affacciare all'esterno il plug di alimentazione, il sensore IR, i led, i pulsanti e i jack audio.

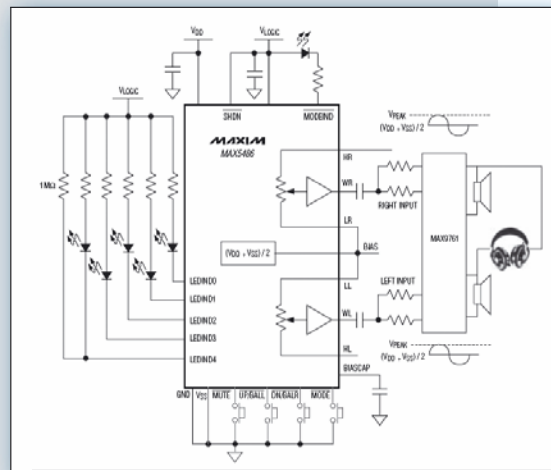
allo scopo fornite tensione al circuito inserendo nella presa il plug di un alimentatore da rete capace di erogare da 9 a 15 Vcc e una corrente di 100 mA. Fornita l'alimentazione, scelto il telecomando da abbinare e, su di esso, due tasti che normalmente non usate (comunque non quelli di accensione o spegnimento degli apparecchi cui il TX appartiene e tantomeno quelli di registrazione di videoregistratori o DVD recorder) premete P1 (LEARN) e attendete che la procedura di apprendimento abbia inizio (la ricezione è segnalata dal SIS-2 con l'attivazione di OUT1 e l'accensione dell'LD1). A questo punto lasciate P1 e premete il tasto del telecomando che volete corrisponda al codice 1 (ossia all'abbassamento del volume) rilasciandolo quando vedete che il led rosso (LD7) dopo essersi acceso si spegne. Il led tornerà illuminato dopo circa 2 secondi. Trascorso questo intervallo, ripetete altre tre volte la procedura appena descritta, ossia effettuate una nuova trasmissione con lo stesso tasto e attendete che il led torni a riposo; le ripetizioni servono a confermare al SIS2 l'abbinamento proprio con il pulsante voluto, che ha buon fine esclusivamente se per tre volte consecutive il SIS2 riceve il medesimo codice per una certa posizione (canale 1 o 2). L'apprendimento e la memorizzazione di un codice, al completamento dei tre passi, viene segnalato dal fatto che LD7 pulsa diverse volte, quindi resta acceso; quando ciò accade, il circuito è pronto per l'apprendimento del secondo codice: puntate il trasmettitore verso il circuito e premete il tasto che vo-

L'integrato MAX5486

È un chip della Maxim che realizza il controllo del volume mediante due potenziometri elettronici a variazione logaritmica da 40 kohm ciascuno, che si spostano in 32 passi (livelli di attenuazione) e sono comandati da pulsanti o livelli logici applicati agli ingressi di controllo; l'integrato contiene, anche una logica di controllo e di interfaccia verso i tasti, che permette di regolare oltre al volume il bilanciamento.

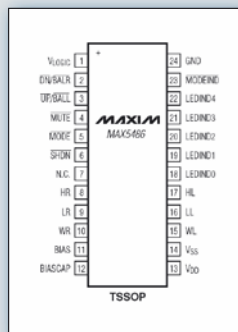
Dispone inoltre di un comando di MUTE, che attenua il livello del segnale di ben 90 dB. Gli ingressi di controllo del componente sono quattro: due per il controllo UP/DOWN, uno per il MUTE e l'altro per impostare la modalità d'uso dei tasti UP/DOWN (MODE); quest'ultimo serve a decidere se con gli ingressi regoliamo il volume o il bilanciamento. Un apposito led segnala quando è attiva la funzione di bilanciamento e quando, invece, UP e DOWN controllano il volume.

Nella modalità bilanciamento, UP aumenta il livello del canale sinistro a discapito di quello del destro e DOWN incrementa il livello



del canale destro diminuendo quello del sinistro. Ognuno degli input di controllo è provvisto internamente di un resistore di pull-up; per questa ragione tutti possono essere gestiti da pulsanti connessi verso massa, transistor (come nella nostra applicazione) o uscite a livello logico TTL-compatibili.

L'integrato funziona a tensione singola di 2,7÷5,5 V o duale di $\pm 2,7$ V ed è idoneo all'uso in apparecchi stereo ad alta fedeltà, dato che è a due canali, presenta una distorsione armonica tipica dell'ordine dello 0,03 % e garantisce una separazione dei canali di ben 100 dB.



lete corrisponda al codice 2 (volume UP) fino a vedere spegnersi LD1. Come per l'apprendimento del codice 1, anche in questo caso il led si riaccenderà dopo circa 2 secondi; quando avverrà ciò, occorrerà ripetere tre volte la trasmissione, esattamente come fatto per il primo canale. Ad apprendimento avvenuto, il diodo LD1 pulserà ripetutamente e poi resterà acceso; a questo punto il circuito sarà pronto per l'uso: trasmettendo con il tasto abbinato al codice, in base alla modalità impostata, abbasserete il volume o

sposterete il bilanciamento verso destra, mentre premendo il tasto del codice 2 alzerete il volume o sposterete il bilanciamento a sinistra. La procedura di abbinamento dei tasti del telecomando può essere ripetuta senza limiti, laddove occorresse cambiare tasti (ad esempio perché ci si è accorti che quelli assegnati servono effettivamente per l'apparecchio cui il telecomando appartiene); resta inteso che ogni volta bisogna far apprendere tutti e due i codici relativi ai tasti del telecomando scelti.

LIFTER: COSTRUIAMO UN PICCOLO UFO

..... di ALESSIO FUSARO

Costruiamo un dispositivo "antigravità" che si alza e vola silenzioso, senza parti in movimento, ma usando semplicemente un sistema di propulsione elettrocinetica.

Vi siete mai chiesti come potrebbe volare un UFO? Senza far rumore, senza razzi di propulsione, senza eliche? Una spiegazione potrebbe essere in quel sistema chiamato *Lifter*. Si tratta di un dispositivo elettrodinamico di propulsione che utilizza una corrente elettrica ad alta tensione per alimentare un "condensatore asimmetrico", fenomeno che produce una spinta più o meno intensa. Nel 1923 il dr. *Thomas Townsend Brown* e il professor *Alfred Biefel* scoprirono che un condensatore costruito in modo asimmetrico, ossia con le armatu-

re di superficie molto diversa tra loro, se alimentato ad alta tensione variava il proprio peso! A seguito di questa scoperta, nel 1952 il dottor Brown riuscì a costruire il primo dispositivo in grado di sollevarsi con una spinta superiore al proprio peso e lo chiamò "Gravitator"; nel 1953, lo stesso dr. Brown, costruì un dispositivo del diametro di 60 cm in grado di raggiungere una velocità teorica di 185 km/h.



Fig. 1

tra loro, fra le quali è interposto un dielettrico. Nei condensatori asimmetrici una delle due armature è molto più piccola, ovvero meno estesa, dell'altra. Originariamente TT Brown realizzava i lifter con una grande piastra circolare (a disco) mentre l'altra piastra era costituita come una piccola "antenna" posta al centro del disco e isolata da esso mediante un opportuno isolatore. Il design dei lifter attualmente

il lato, adiacente al filo, ripiegato di 2÷5 mm come una J rovesciata. L'aria costituisce il dielettrico. Vediamo come funziona: l'alta tensione presente nell'elettrodo sottile (il filo) polarizza le molecole circostanti che vengono attratte verso il filo. Colpendolo perdono un elettrone e gli ioni positivi che si creano vengono respinti dall'elettrodo sottile e attratti da quello negativo (il foglio d'alluminio); gli ioni positivi sbattendo sull'elettrodo negativo riprendono uno o più elettroni, tornando particelle neutre o ionizzate negativamente, che vengono respinte a loro volta dalla lamina di alluminio.

La Fig. 5 mostra un diagramma della distribuzione delle cariche e il loro movimento.

Tutto questo porta a pensare che il sollevamento del lifter sia dovuto all'effluvio elettrostatico che si verifica tra le due armature, le quali, non essendo sovrapposte e di uguali dimensioni, lasciano scoperto il vento ionico; più esattamente, verificandosi non in una zona limitata tra due armature ma in un'area esposta all'esterno, questo movimento d'aria spinge e solleva il lifter. Ciò combacia con la spiegazione ufficiale del funzionamento del lifter, dato che in effetti il vento ionico fa parte della propulsione del dispositivo. Tuttavia il lifter funziona anche nel vuoto (infatti è stato testato con successo nella camera a vuoto spinto della NASA al Marshall Space Flight Center in Huntsville - Alabama- nell'estate del 2003) e questo non si spiega; né si comprende perché la spinta aumenti alimentando il lifter con impulsi a 50÷70 Hz e malgrado il vento ionico diminuisca di intensità.

Dallo studio pubblicato dai ricercatori dell'esercito americano "Force on an Asymmetric Capacitor"

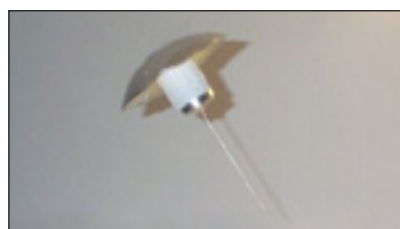
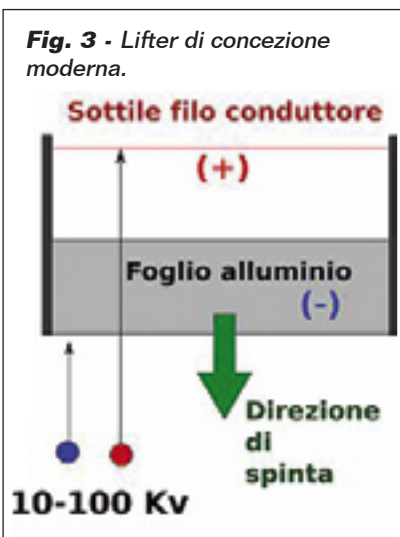


Fig. 2 - Un esempio di lifter di tipo Brown-Bahnson.



realizzati è diverso da quello utilizzato da TT Brown e presenta un rapporto peso/potenza superiore (almeno nei modelli di dimensioni ridotte); un esempio è rappresentato nella Fig. 3. In sostanza, un elettrodo è costituito da un filo di rame di diametro molto ridotto (da 0,1 a 0,3 mm) e l'altro è un foglio d'alluminio di 3÷4 cm di altezza, con

Le ricerche continuarono, seppur più lentamente, nei decenni successivi, anche in collaborazione con la NASA (interessata a studiare la propulsione elettrostatica) e diedero luogo alla registrazione di numerosi brevetti.

Veniamo al nostro secolo: nel 2000 c'è stata una "riscoperta" dell'effetto da parte di alcuni ricercatori americani, che utilizzarono, però, un modello diverso da quello di Brown. La NASA riprese le ricerche con una serie di test in aria e nel vuoto, cui hanno fatto seguito i brevetti per l'utilizzo del dispositivo nelle manovre orbitali. Tra questi, l'*Asymmetrical Capacitor Thruster* e l'*Orbital Maneuvering Propellantless Thruster*. Dal 2001 vari sperimentatori, con l'aiuto di utili informazioni reperite in internet, iniziano a replicare il dispositivo. Merita di essere segnalato il ricercatore JL Naudin (<http://jnaudin.free.fr/>) il cui sito internet era una sorta di testo sacro per chi voleva costruire un lifter. Ora, purtroppo, il sito non è più aggiornato dal 2008 e diversi preziosi link informativi non sono più attivi.

COMPOSIZIONE DEL LIFTER

Ma vediamo come è fatto un lifter o condensatore asimmetrico che dir si voglia. I condensatori che tutti conosciamo sono costituiti da due armature uguali

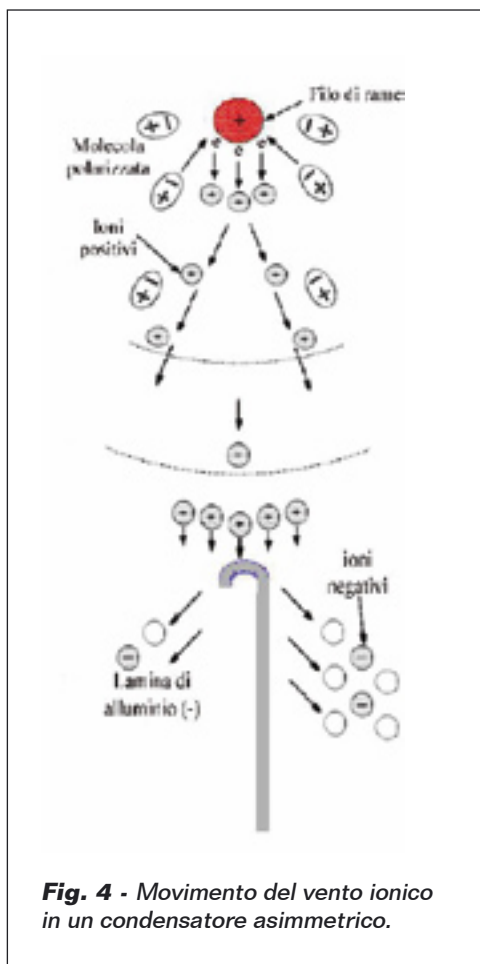


Fig. 4 - Movimento del vento ionico in un condensatore asimmetrico.

del U.S. Army Research Laboratory 2800 Powder Mill Road Adelphi, MD 20783-1197, pubblicato nel marzo 2003, si dimostra come la spinta ionica sia solo una piccola frazione della forza propulsiva del lifter. Le tecnologie classificate "anti-gravitazionali" sono state tenute nascoste al pubblico per sei decenni, mentre venivano sviluppate segretamente in ambito militare. Venne poi rivelato che nel 1992 il Bomber B2 usava cariche elettrostatiche sulle ali e sulla coda. L'Advisor per la Sicurezza Nazionale di Barack Obama, il Generale della Marina in pensione James Jones, ha giocato un importante ruolo nella pubblicazione delle tecnologie riguardanti il lifter. Secondo esperti aerospaziali,

questo confermava che il B2 usava principi elettrogravitici basati sul Biefeld-Brown Effect. La scienza non ha ancora fornito una spiegazione ufficiale e univoca al fenomeno, ma nonostante ciò, il lifter funziona.

COME COSTRUIRLO

Per realizzare un lifter fatto in casa, bisogna innanzitutto costruire un supporto per gli elettrodi. Per avere il maggiore rapporto rigidità/prestazioni, tale supporto va realizzato in forma triangolare; la struttura deve essere molto leggera, elettricamente isolante e facilmente lavorabile.

Il progetto qui descritto è stato costruito con una struttura in balsa, ma può essere realizzato anche con cannuccie di plastica (come quelle delle bibite) che però sono più difficili da lavorare e incollare. Il materiale occorrente è il seguente:

- balsa ben asciutta;
- colla cianoacrilato;
- foglio di alluminio (pellicola da cucina);
- filo di rame da 0,1÷0,3 mm di diametro.

Quest'ultimo si può ricavare sguainando un cavetto elettrico multifilare.

Una volta procurato il materiale, partendo da un'astina di balsa preparare 3 bastoncini lunghi 20 cm, della sezione 1 x 1,5 mm e altri 3 bastoncini lunghi 8 cm, della medesima sezione (Fig. 14). Di questi listelli, smussate gli angoli a 45 gradi con una lima

Fig. 5 - Distribuzione del campo elettrico e linee equipotenziali in un condensatore asimmetrico.

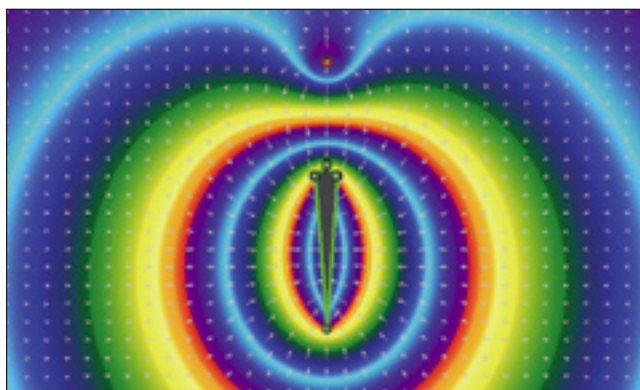
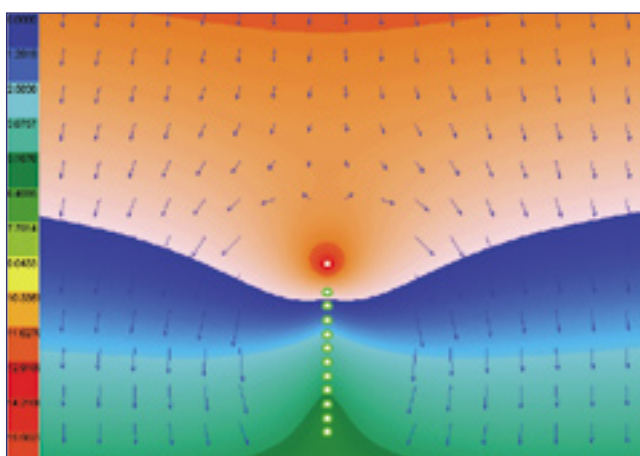


Fig. 6 - Andamento del campo elettrico attorno ad un condensatore asimmetrico.



o con carta abrasiva per poterli incollare agevolmente (Fig. 15). Incollate i 3 bastoncini per formare un triangolo equilatero, interponendo un bastoncino più corto da 8 cm come supporto verticale. I supporti orizzontali vanno incollati a 4 cm dall'estremità superiore dei supporti verticali da 8 cm. Smussate gli angoli con una lima o con carta abrasiva, in modo da ottenere una curva; questo faciliterà la successiva fase di incollaggio del foglio di alluminio e ridurrà le perdite elettriche del lifter per effetto corona. Il lifter non deve necessariamente essere fatto a triangolo: potete costruirlo con qualsiasi foggia, ma la forma triangolare ha il maggior rapporto peso/spinta ed è consigliabile per le prime

Fig. 7 - Lifter triangolare delle dimensioni di appena 1 cm.



Fig. 8 - Un lifter da 5 metri, realizzato dalla Meisei University in Giappone.



realizzazioni; poi, si potrà passare a forme più complesse che, però, richiedono una migliore equilibratura, indispensabile per farlo volare dritto.

A questo punto bisogna ritagliare un nastro dal foglio di alluminio da 5 cm di altezza per 61 cm di lunghezza, che va avvolto attorno al lifter e incollato alla struttura con del cianoacrilato; 4 o 4,5 cm del nastro devono restare avvolti e verticali e costituiranno l'elettrodo di grandi dimensioni del condensatore asimmetrico (Fig. 17). I 5÷10 mm restanti sopra i sostegni orizzontali

andranno "risvoltati" e incollati ad essi in modo da creare una falda sagomata a "J" rovesciata (Fig. 18). Avvolto e fissato il nastro, la parte superiore va avvolta e incollata alla struttura col cianoacrilato. È importante ritagliare con un cutter 5 mm di nastro in prossimità dello spigolo della struttura a triangolo, in modo da evitare scariche (per gli effetti "punta" e "corona" gli spigoli e gli angoli vivi sono la prima sede di scariche elettriche). Per effettuare i collegamenti del filo negativo conviene incollare uno spezzone di filo di rame di piccolo spessore all'interno di un risvolto del foglio di alluminio, illustrato nella Fig. 19. A questo punto bisogna avvolgere il sottile filo di rame nudo a 3 cm dal foglio di alluminio; il filo deve essere ben teso, ma fate attenzione a non rompere la struttura di balsa (Fig. 20).

COLLAUDO DI MASSIMA

Prima di entrare nel vivo del collaudo è doveroso ricordarvi l'estrema pericolosità dell'alta tensione necessaria ad alimentare il lifter: sebbene non sia in grado di generare correnti letali, il circuito proposto per l'alimen-

Fig. 9 - Un lifter a cestello.



Fig. 10 - Lifter a fisarmonica.



tazione può comunque provocare scariche molto dolorose e che possono risultare estremamente pericolose per i cardiopatici. Prese le precauzioni del caso, appoggiate il lifter su un piano pulito ed elettricamente isolante, oppure collegatelo a massa ma fate attenzione che il cavo positivo di alimentazione non si trovi a meno di 5 cm dal piano.

Per effettuare una prima prova è possibile alimentare il lifter prelevando l'alta tensione da un monitor con tubo a raggi catodici, il che potete farlo scollegando la "ventosa" che trovate applicata alla parte alta del cinescopio; in serie al contatto di questa collegate una resistenza da 200÷470 kohm, 5W, che dovrà terminare su uno dei fili di alimentazione del lifter. L'altro filo collegatelo alla massa della scheda del monitor. In questo caso è molto comodo utilizzare una ciabatta con interruttore, in modo da poter accendere e spegnere agevolmente il monitor senza mettere le mani sui contatti, cosa da evitare almeno per un bel po' dopo che il monitor è stato spento; per scaricare la capacità del lifter ed evitare di prendere la scossa, dopo aver spento il

Fig. 11 - Orville: il primo essere vivente a volare su un lifter.



UFO? No, è solo un UAV...

La maggior parte degli UAV (Unmanned Aerial Vehicles) ossia aerei in miniatura comandati a distanza, ha la forma di un aereo, che però presenta degli svantaggi, quali ridotta maneggevolezza e capacità di carico, senza contare che un UAV di questo tipo ha bisogno di una pista per decollare o atterrare. La soluzione a tali limiti sono gli UAV a decollo verticale, che usano rotori che li rendono più simili ad un elicottero. Un nuovo modello di UAV è stato costruito dall'azienda inglese AESIR: non ha parti rotanti esterne al corpo e la sua forma lo rende simili più a un UFO che a un elicottero o un aereo in miniatura. Per volare, sfrutta il cosiddetto effetto Coanda, ossia la tendenza di un flusso d'aria o liquido, che si trovi ad incontrare un corpo curvo lungo il suo percorso, a seguire la curvatura di tale corpo invece di una traiettoria rettilinea. Nel caso dell'UAV creato da AESIR, il flusso d'aria viene creato al centro del veicolo grazie ad un'elica, quindi convogliato verso il basso in modo da seguire la superficie curva del veicolo e generare una spinta verso l'alto. Il flusso d'aria causa la rotazione in senso opposto a quello dell'elica, per contrastare la quale si realizzano prese d'aria e flaps di stabilizzazione. UAV del genere possono trasportare tra 10 e 1.000 kg di carico e quindi sistemi video per operazioni di sorveglianza, ricognizione e fotografia aerea.



monitor cortocircuitate (con un cacciaviti metallico dotato di manico isolante molto spesso) i cavi che portano al lifter. Questi cavi conviene che siano "sospesi" e per il momento tenuti più corti possibile. Alimentate per un istante il monitor e osservate cosa succede: se avvengono scariche sugli angoli del lifter, togliete tensione, cortocircuitate con un cacciaviti l'alimentazione (per scaricare la capacità del lifter) e ritagliate un po' la parte alta in alluminio in prossimità allo spigolo che scarica. Se avvengono scariche tra il filo e l'alluminio, togliete tensione, cortocircuitate con un cacciaviti l'alimentazione e aumentate la distanza tra gli elettrodi del lifter, facendo scorrere verso l'alto il filo sottile lungo i supporti di balsa.

Se il lifter produce un soffio ma resta immobile, al solito togliete tensione e scaricate la linea di alimentazione con un cacciaviti, quindi avvicinate il filo di rame all'alluminio facendolo scorrere verso il basso lungo i supporti di balsa; se il filo di rame vibra come una corda, allora riavvolgetelo più teso. Qualora il lifter, pur producendo un soffio (indice del fatto che c'è l'alta tensione e che i

contatti sono alla giusta distanza) si muove ma non si alza, provate ad appoggiarne un angolo sopra ad un bicchiere rovesciato, in modo che il lifter sia inclinato: se alimentandolo in quest'ultima posizione si rovescia, vuol dire che il lifter è troppo pesante o l'alimentazione troppo "scarsa". Nel primo caso, prima di ricostruire il lifter con una sezione più sottile di balsa è possibile alleggerirlo limando o accorciando le "gambe" della struttura. Nel secondo caso, potrebbe essere insufficiente la corrente di alimentazione del generatore di alta tensione o la tensione in uscita del flyback utilizzato (o del monitor usato nella prova). Se, infine, il lifter parte e si libra nell'aria, significa che avete fatto un buon lavoro!

A questo punto non vi resta che collegare dei fili di nylon alle estremità inferiori del lifter e ancorarlo al piano di appoggio. I fili all'inizio andranno tenuti corti per verificare se il lifter è "sbilanciato"; per bilanciare la spinta è sufficiente correggere la posizione di fissaggio del filo di rame sul sostegno, avvicinando o allontanando questo dall'alluminio. Se non fosse sufficiente, è possibile alleggerire l'angolo pesante limando (assottigliando) il sostegno in balsa. Con il lifter bilanciato si potranno allungare i fili e i contatti fino a farlo volare sempre più in alto!

Con la pratica potrete provare a realizzare lifter delle più svariate forme e dimensioni. Se volete suggerimenti, potete guardare le immagini in queste pagine.

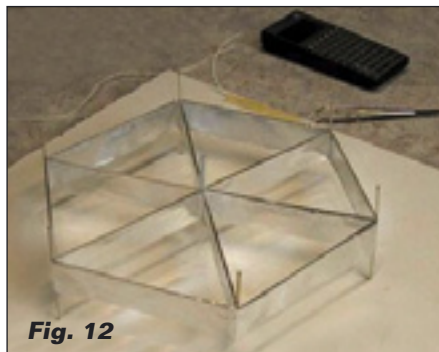


Fig. 12



Fig. 13

Guardate, ad esempio, il modello nella **Fig. 11**: realizzato nel 2003, e alimentato da 6 flyback, su di esso è stato fatto volare il primo essere vivente (il topolino *Orville*).

L'ALIMENTAZIONE DEL LIFTER

Dopo il primo collaudo, potete pensare ad un alimentatore definitivo, che deve fornire una tensione di 30 kV in continua (in

sito dell'HR, leader del settore dei trasformatori di riga compatibili (<http://www.hrshop.es>). Sul retro del monitor, solitamente, è indicata la tensione della post-accelerazione del tubo catodico; per il nostro alimentatore è bene che tale tensione sia di almeno 30 kV. Per aiutarvi, vi diciamo che nel sito dell'HR è presente un motore di ricerca che permette, conoscendo marca e modello del

monitor o TV; nel nostro caso, l'indispensabile è che contenga il diodo rettificatore dell'EAT e che la tensione in uscita sia almeno 30 kV. Il circuito da noi proposto è la più semplice fonte di alta tensione e permetterà di pilotare la quasi totalità dei trasformatori per flyback; permette di generare impulsi unidirezionali ad alta tensione, proprio grazie al rettificatore interno. Vediamo

La meccanica del lifter

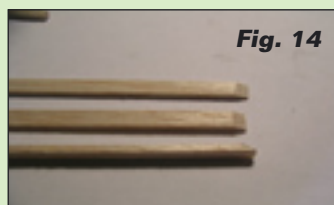


Fig. 14

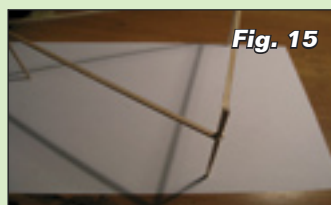


Fig. 15

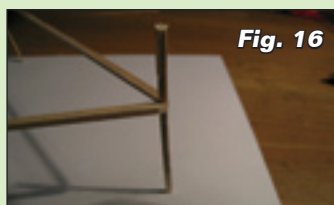


Fig. 16

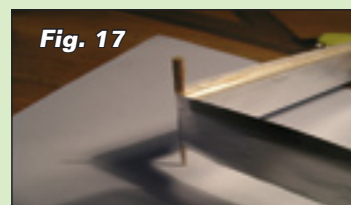


Fig. 17

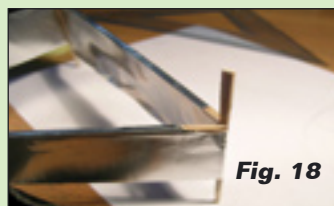


Fig. 18

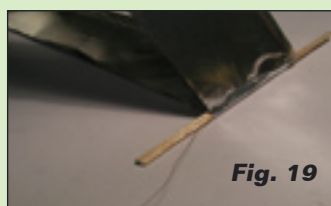


Fig. 19

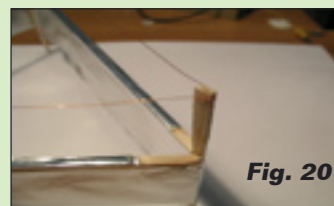


Fig. 20

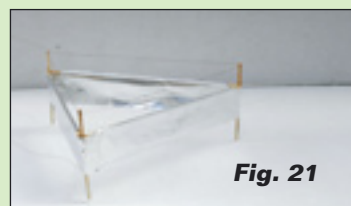


Fig. 21

Fasi costruttive del lifter: si parte da una serie di bacchette di balsa (legno leggerissimo) da tagliare a misura e smussare alle estremità, in modo che quando verranno rivestite dall'alluminio non avranno punte tali da disperdere energia a causa dell'effetto corona. Le bacchette lunghe devono formare un triangolo e le corte saranno i piedistalli. Il triangolo andrà circondato da una fascia di alluminio da cucina inferiormente e da un cavo in rame superiormente; queste due parti andranno collegate direttamente ai cavi dell'alimentatore.

teoria sono sufficienti poco più di 20 kV, ma molto dipende dalla realizzazione e dal peso della struttura). Per ottenere questa tensione, il sistema più semplice è utilizzare un trasformatore flyback (trasformatore di riga) che possiamo trovare in un monitor CRT o anche nei televisori a tubo catodico, sebbene in questi ultimi vengano utilizzate tensioni più basse (ad esempio 23 kV per una TV da 22 pollici). I trasformatori flyback possono essere "espianati" da vecchi monitor o acquistati, anche via Internet nel

monitor, di risalire alle caratteristiche del flyback ed alla sua piedinatura. Per rendere la realizzazione del progetto la più facile possibile, abbiamo disegnato un circuito semplicissimo che utilizza un trasformatore di riga e pochissimi altri componenti; con tale schema è possibile utilizzare anche un flyback con il primario bruciato o interrotto. Il trasformatore da noi utilizzato, come tutti i trasformatori di riga, contiene diversi avvolgimenti e componenti necessari al funzionamento all'interno del

subito come funziona: quando riceve alimentazione, il transistor entra in conduzione perché è polarizzato dal partitore di resistenze R1 e R2 attraverso l'avvolgimento L1 fatto sul nucleo del trasformatore. Con il transistor in conduzione, fluisce corrente dal collettore nella bobina L2, inducendo un campo magnetico nel nucleo del trasformatore, che induce a sua volta nella bobina L1 una tensione che porta all'interdizione il transistor. A questo punto si blocca bruscamente la corrente in L2 e viene generata

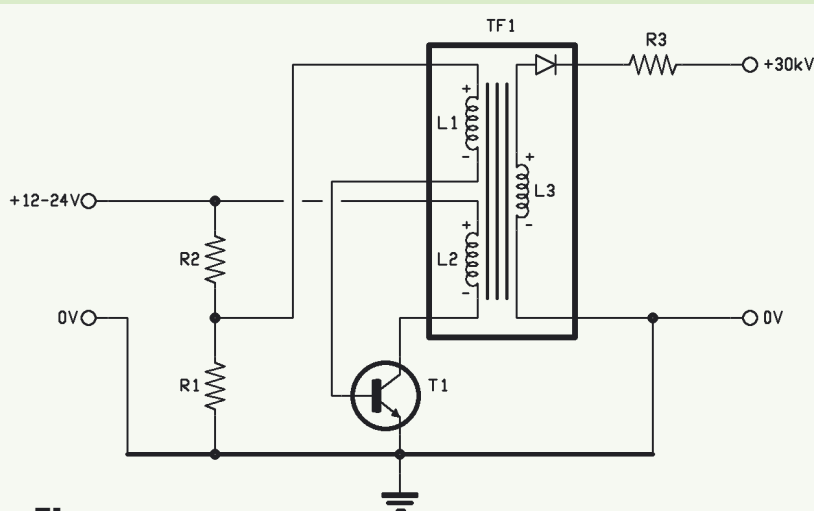
una extratensione sul secondario ad alta tensione (EAT, che nello schema è siglato L3) producendo un impulso. Adesso, non essendo tensione indotta in L1, il transistor ritorna in conduzione e il ciclo si ripete. Uno dei vantaggi di questo circuito è che la frequenza di lavoro si adatta al carico e il trasformatore lavora sempre in risonanza, con un enorme guadagno in tensione. Il circuito è molto flessibile, e può essere adattato a vari tipi di transistor di potenza e vari flyback; è importante che il transistor sopporti i picchi di extratensione prodotti dal circuito (almeno 100V di tensione di rottura) e sia montato su un generoso dissipatore. I valori di R1 e R2 non sono critici, ma è bene che tali resistenze possano dissipare 2÷5 W.

La R3 serve a proteggere l'avvolgimento EAT del trasformatore in



Gli avvolgimenti primario (L2) e secondario di reazione (L1) devono essere fatti nello stesso verso, direttamente su una colonna libera del nucleo del trasformatore EAT; quello d'alta tensione (siglato R3 nello schema) è già presente nel trasformatore e non va toccato. Per il verso di collegamento dei due, seguite lo schema qui sotto considerando che + è l'inizio avvolgimento.

[schema ALIMENTATORE]



Elenco Componenti:

R1: 27 ohm 5 W
R2: 240 ohm 5 W
R3: 220 kohm 5 W
T1: BD249C

TF1: Trasformatore di riga

Varie:
- Dissipatore
- Morsetto 2 poli

caso di cortocircuito e ad evitare che eventuali scariche sul lifter carbonizzino la struttura in balsa di quest'ultimo.

Abbiamo detto che per il nostro circuito va bene qualsiasi trasformatore capace di erogare oltre 30 kV, in quanto il primario (L2) e l'avvolgimento di reazione (L1) li prepariamo artigianalmente, disponendoli sulla parte scoperta del nucleo, come mostrato nell'apposita fotografia. L1 è costituita da 5 avvolgimenti in filo di rame smaltato o in guaina (0,5 mm di diametro) attorno

al nucleo; in realtà il numero di spire può essere variato da 3 a 5 in funzione delle caratteristiche del transistor e del trasformatore. In pratica, se il lifter non decolla mettete 5 spire, mentre se tende a scaricare pur sollevandosi, usate 4 o 5 spire. Quanto alla L2, è formata da 4 spire di filo da 1,3 mm di diametro. L3 è l'EAT del trasformatore. È importante collegare a terra e al negativo dell'alimentazione il negativo del secondario EAT, altrimenti, per effetto capacitivo c'è il rischio di ritrovarci alta tensione sul nucleo



Pericolo alta tensione

L'alimentatore per il lifter sviluppa qualche decina di kV ed è quindi necessario maneggiarlo con cura: non toccatelo e non mettete le dita sul lifter o vicino ad esso, perché potreste prendere una scossa non solo dolorosa ma anche estremamente pericolosa!



del trasformatore flyback. Nel caso avvengano scariche tra la massa e qualche altro piedino del flyback, bisognerà isolare tali piedini con delle gocce di colla a caldo o resina o silicone.

Il circuito può essere alimentato da 12 a 24 V o anche più, se la tensione di rottura del transistor lo permetterà. Siccome necessita molta corrente, sarà conveniente utilizzare un alimentatore elettronico per alogene, la cui uscita sia raddrizzata e livellata da un condensatore da almeno 47.000 μ F. Alimentandolo con un accumulatore al piombo, è possibile far volare il lifter anche fuori casa, senza bisogno di una presa di rete nelle vicinanze.

Alimentando il circuito con un alimentatore a tensione variabile è possibile variare la forza propulsiva del lifter. Completato il circuito, potete fare una prova di massima per vedere se va bene: con delle pinze isolate avvicinate i conduttori del secondario e verificate che già a 3-4 cm si verifichino delle scariche. Se ciò non accade, togliete tensione e invertite le connessioni di L1 prima e L2 dopo; ciò perché se manca tensione può essere che l'oscillatore non si inneschi. Invertendo solo L2 potrebbe innescarsi, ma non è detto che la tensione sul secondario EAT abbia la polarità giusta per superare il diodo interno al flyback. Per

Guarda il volo del
nostro lifter su YouTube!



Vuoi vedere in azione il nostro lifter? Alzarsi da terra e volare come un piccolo UFO? Nulla di più semplice: collegati alla pagina <http://www.youtube.com/watch?v=mPAwS1dULJY>. Potrai anche vedere i video di moltissimi altri nostri progetti proposti in passato.

chi volesse un alimentatore per lifter più efficiente, è possibile far commutare il primario originale del flyback con un transistor di potenza ad alta tensione, pilotato da un NE555 in configurazione astabile. In tal caso il circuito dovrà essere dimensionato per lo specifico trasformatore flyback che vorrete utilizzare e andrà alimentato con una tensione dai 70 ai 120 V e pilotato alla corretta frequenza di funzionamento del flyback; anche in questo caso, variando la tensione di alimentazione sarà possibile variare la spinta del lifter.

RM ELETTRONICA

Forniture per hobbisti, scuole e industria

Distributore autorizzato:

Il tuo punto di riferimento per l'elettronica a ROMA

Via Val Sillaro 38 • 00141 Roma • Tel: 06-8104753

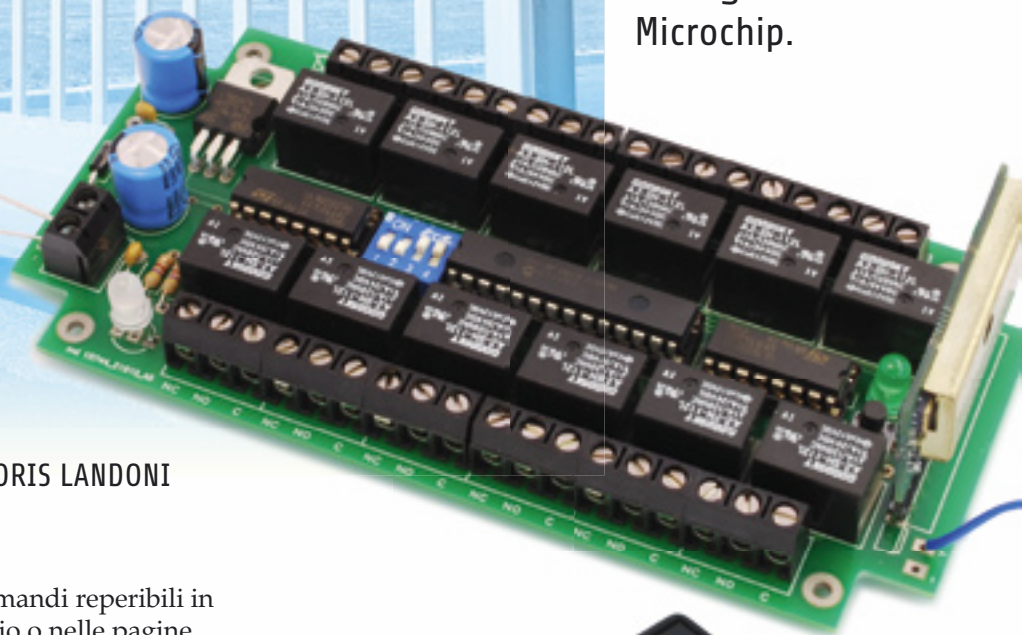
RADIOCOMANDO CODIFICATO A 12 CANALI

Dispone di 12 relé che possono essere attivati in modalità astabile o bistabile mediante un compatto telecomando. Opera a 433,92 MHz ed utilizza una codifica rolling-code HCS Microchip.

..... di BORIS LANDONI

La maggior parte dei radiocomandi reperibili in commercio, i kit di montaggio o nelle pagine delle riviste specializzate, è a due o quattro canali, perché l'esigenza più comune è di aprire e chiudere una serratura, comandare un paio di tapparelle motorizzate o un apricancello, accendere e spegnere una luce; diciamo pure che quando con un solo trasmettitore possiamo comandare tre o quattro utilizzatori di casa, è più che sufficiente. Esistono però situazioni in cui è necessario gestire anche una decina di utilizzatori ed allora doverlo fare con telecomandi bicanale o quadricanale diventa assai scomodo: infatti bisognerebbe portare con sé due, tre o più trasmettitori, che, anche se piccoli, certo creerebbero qualche problema. Per situazioni del genere abbiamo pensato di realizzare un radiocomando a 12 canali, con il quale, mediante un semplice

trasmettitore palmare, è possibile attivare e disattivare a distanza, via radio, altrettanti utilizzatori elettrici o altri dispositivi gestibili elettricamente; il nostro sistema è composto da un trasmettitore abbastanza compatto, spesso 1 cm e provvisto di tastiera



numerica, ed un ricevitore con uscite a relé, che possono essere attivate in modo sia bistabile che impulsivo. La particolarità di questo dispositivo è che la modalità ad impulso può essere attuata in due modi: nel primo il relé segue il comando, nel senso che scatta quando si preme il tasto del trasmettitore e ricade al suo rilascio, salvo che il TX non si allontani troppo da interrompere il collegamento radio; nel se-

a caso l'HCS viene usata anche negli antifurto.

Il trasmettitore è un'unità prodotta dall'Aurel (leader italiana del settore dei moduli SMD wireless) siglata TX-12CH che si compera già pronta presso, ad esempio, la ditta Futura Elettronica di Gallarate (www.futurashop.it); di questa diciamo solo che contiene un modulo trasmettente RF operante in UHF alla frequenza di 433,92 MHz, in modulazione

piano sia circuitale che costruttivo.

SCHEMA ELETTRICO

L'unità ricevente del nostro radiocomando si basa su un modulo RX Aurel anch'esso, a noi nuovo perché non l'abbiamo mai usato: si tratta dell'RX 4MHCS-4B, un ibrido supereterodina contenente un ricevitore radio UHF sintonizzato sui 433,92 MHz, provvisto di demodulatore d'ampiezza OOK e squadratore degli impulsi in uscita, completato da una decodifica che gli permette di rispondere correttamente al TX-12CH. La decodifica al suo interno è realizzata da un microcontrollore che implementa il protocollo HCS e riconosce i comandi inviati dal trasmettitore; in pratica il decoder ha 4 linee d'uscita che esprimono in formato binario su una word di 4 bit il numero del canale, ovvero il numero del tasto che è stato premuto sul TX. Ciascuna delle uscite è interfacciata con un transistor NPN, quindi è a logica invertita, ossia attiva a livello logico basso; i transistor delle uscite sono tutti configurati come open-collector. Il modulo dispone di un ingresso ed un'uscita rispettivamente per un pulsante e per connettere un led, entrambi necessari alla procedura di autoapprendimento; infatti, essendo un rolling-code, il radiocomando richiede che il ricevitore venga abbinato al trasmettitore, ovvero che apprenda il codice base e quant'altro serve per sincronizzarsi con esso. Della procedura di abbinamento ci occuperemo più avanti. Chiaramente per decifrare i comandi serve una logica, discreta o programmabile, che funzioni da decoder 4 a 16, ovvero, nel nostro caso, 4 a 12, dal momento che abbiamo dodici uscite. Per semplificare il circuit-

Il trasmettitore TX-12CH

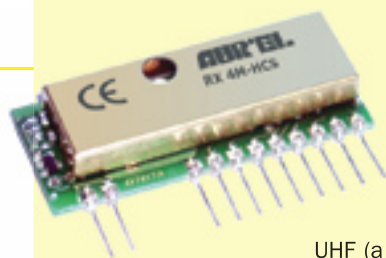
Il trasmettitore RF da noi utilizzato per comandare il ricevitore è un palmare prodotto dall'Aurel, dotato di encoder a codice dinamico (HCS301) il cui segnale d'uscita seriale modula in ampiezza un oscillatore SAW a 433,92 MHz operante con una potenza di 1 milliwatt. Le caratteristiche del dispositivo sono qui elencate.

- Frequenza operativa: 433,92 MHz
- Potenza irradiata: 1 mW
- Modulazione: OOK (AM)
- Alimentazione: 2 pile a bottone
- Assorbimento: 10 mA
- Vita della batteria: 3÷5 anni
- Dimensioni 54 x 85,5 x 10,4 mm



condo, quando si preme un tasto del trasmettitore il relé corrispondente si attiva e resta così per un tempo preimpostato, definibile a piacere. Tutte le impostazioni riguardanti il funzionamento del ricevitore si definiscono mediante un dip-switch. A ciò va aggiunto il fatto che il sistema è codificato mediante una codifica ad elevata sicurezza, perché variabile: si tratta dell'HCS della Microchip, un sistema rolling-code che già avete conosciuto negli anni passati perché impiegato in numerosi radiocomandi e capace di garantire l'esclusività del comando anche in ambienti dove operano molti radiocomandi basati sullo stesso sistema, e la tutela contro l'effrazione. Non

d'ampiezza, con una potenza di 1 milliwatt, sufficiente, con un ricevitore radio come quello impiegato nel nostro circuito RX, a coprire alcune decine di metri di distanza anche in presenza di ostacoli (purché non si tratti di pareti con armature metalliche). Della trasmettente non descriviamo l'elettronica, in quanto si tratta di un prodotto già pronto, che, per la tecnologia impiegata, è più facile comperare che costruirsi; ci limitiamo a dire che contiene un encoder HCS301 attivato da una tastiera a matrice di 4 righe per tre colonne. Invece il ricevitore si costruisce semplicemente, perché impiega componenti a montaggio normale; per questo lo descriviamo sul



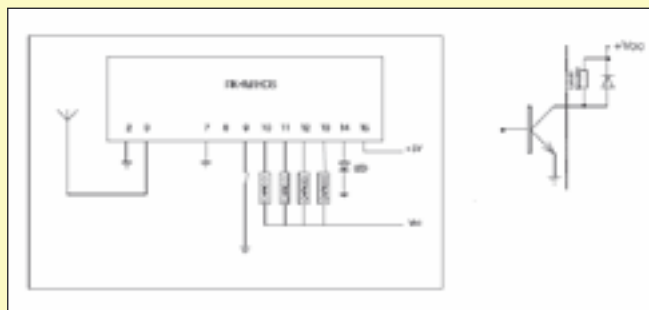
Il ricevitore RX 4MHCS-4B

Il modulo impiegato nell'unità ricevente del nostro radiocomando è un ibrido operante in UHF (a 433,92 MHz) completo di demodulatore AM, squadratore del segnale dati in uscita e decodificatore per l'algoritmo Keeloq Microchip usato dai trasmettitori con encoder HCS30x. La decodifica avviene ad opera di un piccolo microcontrollore che fornisce in uscita il numero del pulsante del trasmettitore che viene di volta in volta premuto, però in formato binario, su 4 bit. La tensione di alimentazione del modulo (pin 15) è di 5 volt. Al pin 9 deve essere collegato il pulsante per l'apprendimento dei trasmettitori (se ne possono abbinare 10) ed al pin 14 va collegato l'anodo del diodo luminoso per il controllo dell'avvenuta programmazione (la corrente di uscita è limitata internamente a circa 20 mA da una resistenza da 180 ohm).

Ogni uscita è comandata da un transistor in configurazione open-collector, in grado di sopportare una corrente massima di 100 mA: ciascuna è attiva quando il transistor è in conduzione (uscita a zero) e disattiva col transistor interdetto (uscita ad 1 logico) in interdizione. Qui di seguito è descritta la funzione di ciascun pin.

Pin-out

- 2) GND
- 3) Antenna
- 7) GND
- 8) Test Point - Uscita analogica RX
- 9) Pulsante di programmazione (connesso a massa)
- 10) Uscita Ch1 - Open collector (attivata dalla pressione del pulsante 1 nel trasmettitore)
- 11) Uscita Ch2 - Open collector (attivata dalla pressione del pulsante 2 nel trasmettitore)
- 12) Uscita Ch3 - Open collector (attivata dalla pressione del pulsante 3 nel trasmettitore)
- 13) Uscita Ch4 - Open collector (attivata dalla pressione del pulsante 4 nel trasmettitore)



14) Uscita per led (attiva a livello alto)

15) Vcc (+5 V)

La seguente tabella mostra lo stato logico dei piedini corrispondenti alle quattro uscite dati in corrispondenza della pressione di uno dei dodici tasti del trasmettitore. A riposo (ovvero se il modulo non riceve alcun segnale dal trasmettitore abbinatogli) le uscite si presentano tutte a livello alto (sono in condizione open).

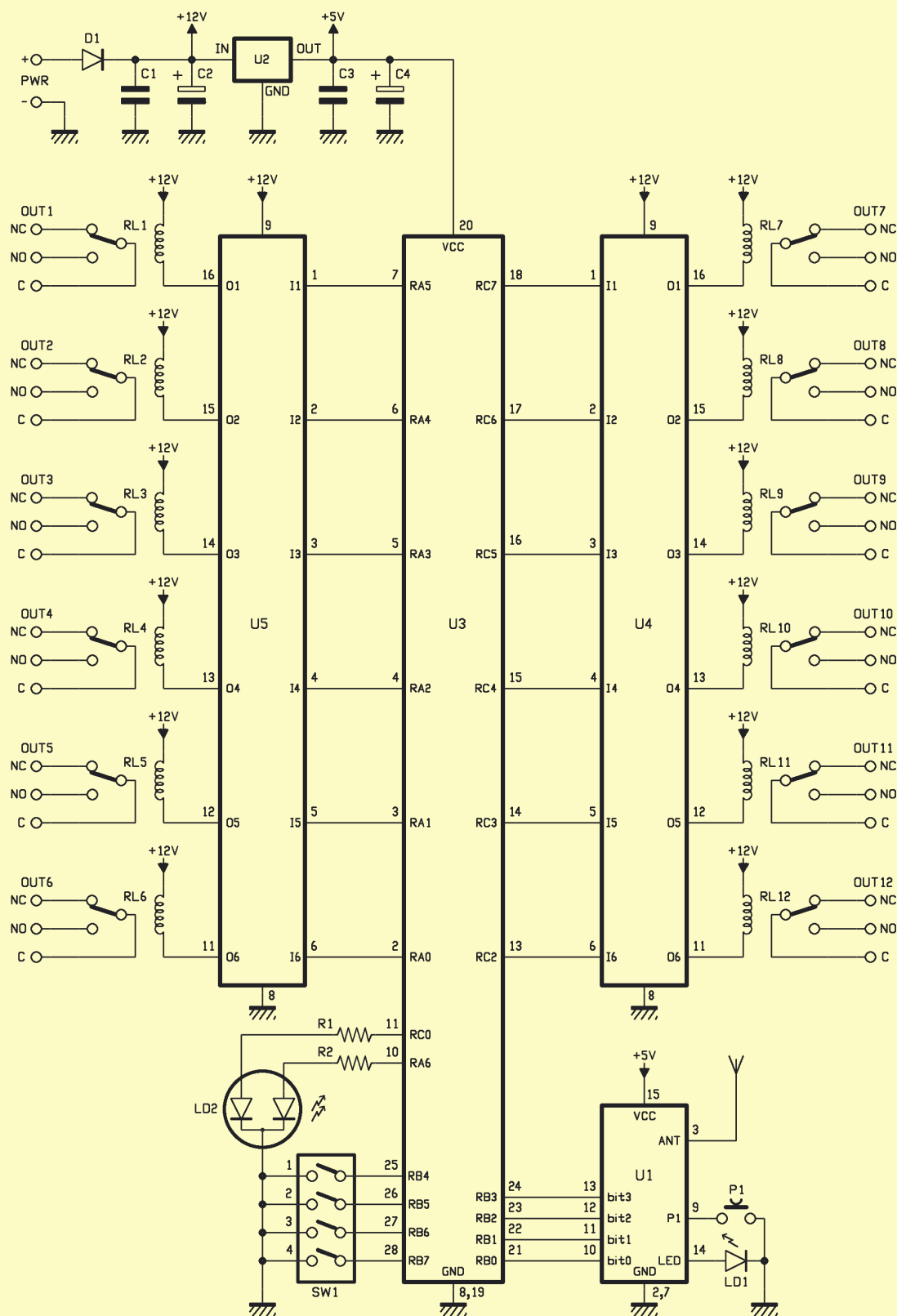
Pulsante del TX	10	11	12	13
1	0	1	1	1
2	1	0	1	1
3	0	0	1	1
4	1	1	0	1
5	0	1	0	1
6	1	0	0	1
7	0	0	0	1
8	1	1	1	0
9	0	1	1	0
0	1	0	1	0
SHIFT	0	0	1	0
ENTER	1	1	0	0

to, abbiamo lasciato il compito a un microcontrollore ancora Microchip, di tipo PIC18F2620, che dopo la sequenza di power-on e l'inizializzazione degli I/O, legge i dati in arrivo dall'ibrido Aurel e li elabora in modo da usarli per controllare i relé. Ma naturalmente non potevamo usare un microcontrollore solo per questo, in quanto per decodificare quattro bit binari ci sarebbe bastato un decoder 4 a 16. In realtà il micro ci serve sia per la decodifica, sia per ottenere tre modi di funzionamento delle uscite a relé, ossia l'attivazione bistabile, quella impulsiva e quella ad inseguimento del pulsante. Ma andiamo con ordine e vediamo tutto quello che fa il micro:

all'avvio, il firmware provvede all'inizializzazione delle linee RA0, RA1, RA2, RA3, RA4, RA5 ed RC2, RC3, RC4, RC5, RC6, RC7 come uscite, destinate al comando dei relé ed alle RB0, RB1, RB2, RB3 come input digitali (con pull-up interno); diciamo pure che la parte compresa tra questi ultimi I/O ed RA0, RA1, RA2, RA3, RA4, RA5, RC2, RC3, RC4, RC5, RC6, RC7 è il blocco decoder. Sempre all'inizializzazione, RB4, RB5, RB6 ed RB7 vengono settate come ingressi con pull-up interno e destinati alla lettura del dip-switch SW1; infine, RA6 ed RC0 vengono inizializzate come uscite e destinate al comando del led bicolore. Dopo l'inizializzazione, nel mi-

crocontrollore gira il main-program, che, come prima operazione, compie la lettura della RB4, quindi del dip 1; se questo viene trovato aperto, il circuito funziona effettivamente come ricevitore e quindi esegue i comandi in arrivo dal trasmettitore. Invece, se dip 1 è chiuso si avvia la fase di programmazione, che riguarda la definizione delle modalità operative delle uscite. Questa programmazione non va confusa con l'apprendimento, che si compie sul ricevitore ibrido mediante P1.

Vediamo dunque in cosa consiste la programmazione delle uscite, per ognuna delle quali è possibile definire la modalità di funzionamento e, limitatamente



alla modalità impulsiva pura (quella in cui il relé si eccita a seguito del comando a distanza e ricade dopo un intervallo

costante) il tempo di attivazione; dato che non potevamo disporre di una linea di impostazione per ciascun'uscita, abbiamo previsto

che la modalità sia definita da un unico comando per tutti i 12 canali e che la programmazione del tempo, nel modo impulsivo

ad inseguimento, si effettui un canale alla volta. Procediamo con ordine e vediamo il significato dei dip, eccezion fatta per il primo, che abbiamo già descritto. Il dip 2 è quello che definisce se le uscite debbano funzionare in modo bistabile o astabile: quando è chiuso, tutti i canali, ogniquale volta ricevono un comando, cambiano la loro condizione rispetto all'istante precedente, perciò se un relé è a riposo si eccita e se è eccitato torna a riposo. Lasciando aperto il dip 2 tutte le uscite lavorano ad impulso. A questo punto, chi decide come debba essere gestita la funzionalità astabile è il dip 3: se è aperto, ogni relé segue il tasto del trasmettitore; ciò significa che se sul TX premiamo il tasto 2, scatta RL2 e resta eccitato fin quando non rilasciamo detto pulsante. In realtà, mantenendo premuto un tasto, il TX trasmette al massimo per 25 secondi consecutivi. In questa modalità si può attivare un solo relé alla volta. Chiudendo il dip 3, invece, tutte le uscite lavorano in modalità monostabile pura, ovvero a seguito dell'arrivo di un comando che le riguarda, saranno attive per un tempo programmabile liberamente. La programmazione del caso si effettua chiudendo il dip 1 (l'entrata in programmazione viene segnalata dall'accensione a luce rossa fissa del led bicolore) e premendo e rilasciando sul trasmettitore il tasto corrispondente al canale da programmare; a questo punto il ricevitore inizia a far lampeggiare di rosso il led bicolore, al ritmo di una pulsazione al secondo. Quando ha contato un tempo uguale a quello per cui desiderate che il relé resti eccitato, premete nuovamente lo stesso tasto del TX, ovvero quello del medesimo canale. Ora

il conteggio ed il lampeggio si arresteranno e il ricevitore darà conferma visiva (mediante una rapida pulsazione a luce rossa del led bicolore) dell'avvenuta memorizzazione del tempo di attivazione del relé per il canale corrispondente al tasto usato. Per esempio, se vogliamo che RL3 in monostabile puro, a seguito di

del led bicolore (2 secondi) quindi premere nuovamente lo stesso pulsante 3 e verificare che il led bicolore pulsi di rosso e torni acceso a luce rossa fissa. Chiudiamo la carrellata sui dip con il dip 4, che, se viene chiuso, attiva per la modalità bistabile il ripristino automatico; questa funzione serve a memorizzare

L'algoritmo Keeloq

La codifica Keeloq è una delle più utilizzate nei radiocomandi ad elevata sicurezza perché implementa un algoritmo di variazione del codice praticamente non ripetibile e inviolabile; costruire il decoder è abbastanza semplice, perché la Microchip rende disponibile il software da implementare nel micro che svolge questa funzione. Sebbene venga utilizzato da gran parte dei fabbricanti di radiocomandi, l'HCS301 garantisce un'elevatissima sicurezza d'uso perché permette di realizzare prodotti personalizzati: infatti ogni costruttore ha la possibilità di personalizzare la propria produzione con un Manufacturer Code a 64 bit, che rappresenta uno dei dati di partenza per generare la chiave di crittografia con la quale viene generata la parte variabile del treno d'impulsi a 66 bit prodotto dall'encoder HCS301. Ogni singolo chip viene inoltre identificato da un codice univoco a 28 bit denominato, appunto, Serial Number. Anche questo dato viene memorizzato in fase di programmazione del chip da parte del costruttore del trasmettitore. In ultima analisi ciascun costruttore programma i chip con un proprio Manufacturer Code a 64 bit (lo stesso utilizzato nel software di decodifica del ricevitore) ed un Serial Number a 28 bit, univoco per ciascun HCS301.

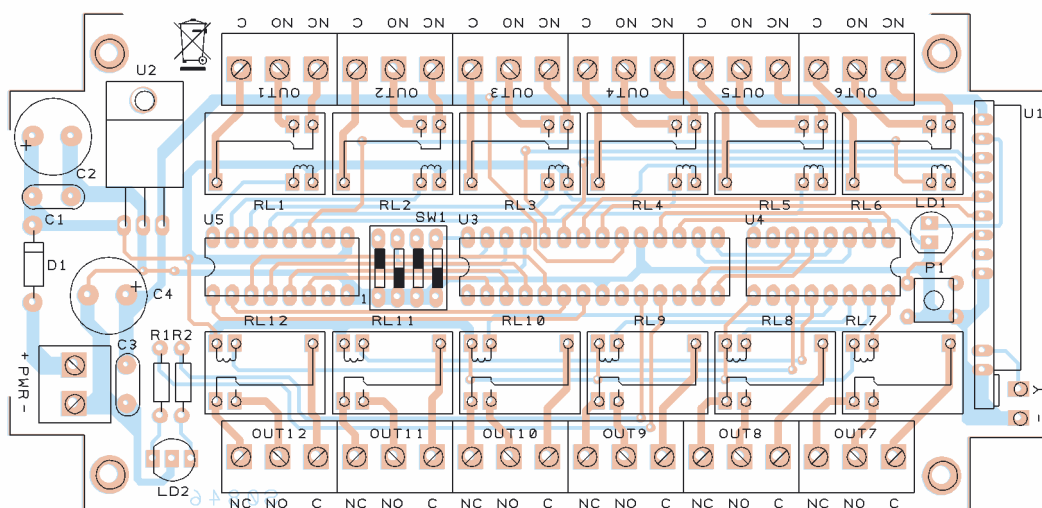
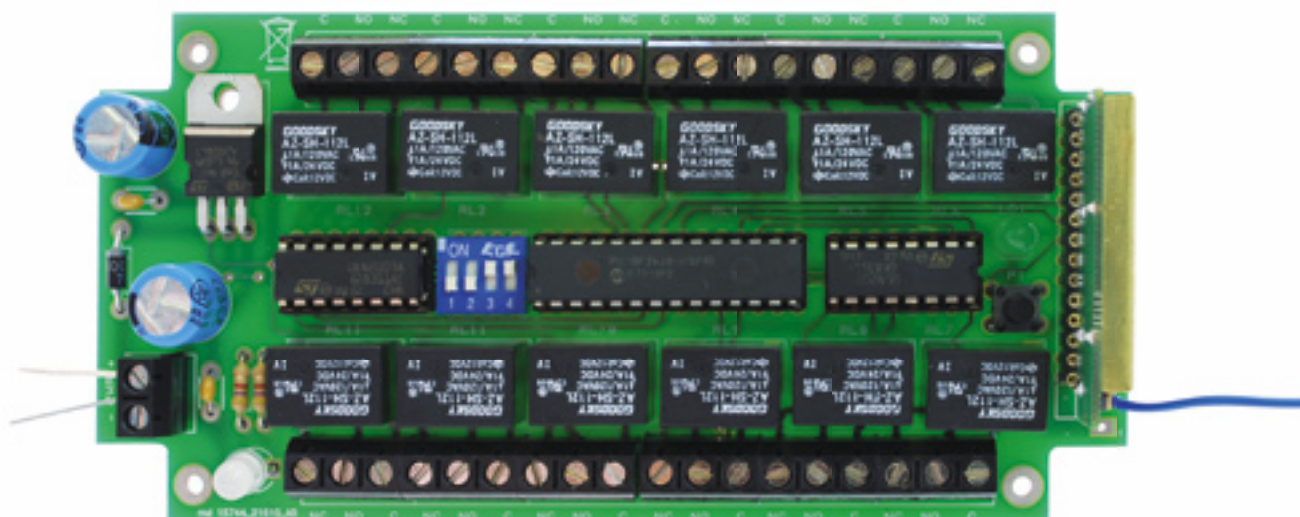
Il Manufacturer Code è un codice a 64 bit specifico di ciascun produttore (OEM per cui la Microchip produce gli encoder della serie HCS30x) che viene memorizzato sia nella EEPROM dell'encoder HCS301 che nella memoria del microcontrollore utilizzato come decoder e nel quale è implementato il software Keeloq. Per poter funzionare, un Encoder e un Decoder devono avere lo stesso Manufacturer Code, altrimenti il codice variabile che il ricevitore (il decodificatore) si aspetta sarà sempre diverso da quello generato dall'encoder. Il Serial Number è un codice progressivo a 28 bit col quale vengono differenziati i chip HCS in fase di programmazione. Il Serial Number consente dunque di distinguere un trasmettitore da un altro. In fase di accoppiamento tra TX e RX, questo codice viene appreso dal ricevitore; durante il normale funzionamento, l'encoder verifica come prima cosa che il Serial Number del TX che trasmette sia tra quelli memorizzati e solo in questo caso procede a decodificare e verificare gli altri dati contenuti nella stringa in arrivo.

Formato stringa di dati generata dal chip HCS301



ogni comando rimanga attivo per 2 secondi, dobbiamo chiudere il dip 1 e trasmettere, premendo il tasto 3, attendendo due lampeggi

la posizione assunta dai relé in modo che in caso di blackout, al ritorno della corrente assumano la condizione che avevano prima



Elenco Componenti:

R1, R2: 470 ohm
C1, C3: 100 nF multistrato
C2, C4: 470 μ F 25 VL elettrolitico
U1: RX-4MHCS-4B
U2: 7805
U3: PIC18F2620-I/SP (MF846)
U4, U5: ULN2003
LD1: Led 5 mm bicolore

LD2: Led 5 mm verde
D1: 1N4007
SW1: Dip switch 4 vie
P1: Microswitch
RL1 ÷ RL12: Relé 12 V - 1 scambio

Varie:
- Morsettiera 2 poli

- Morsettiera 3 poli (12 pz.)
- Zoccolo 8+8 (2 pz.)
- Zoccolo 14+14
- Vite 10 mm 3 MA
- Dado 3 MA
- Antenna accordata 433 MHz
- Circuito stampato

della perdita di alimentazione. Proseguiamo l'analisi dello schema elettrico vedendo la sezione di uscita: dato che la corrente erogabile da ciascun I/O del microcontrollore PIC18F2620 non è sufficiente a far eccitare correttamente i relé (si tratta di

circa 25 mA ad uscita) abbiamo ritenuto opportuno utilizzare dei line-driver ULN2003; per l'esattezza, ne usiamo due (U5 serve le prime sei uscite ed U4 i canali dal 7 al 12). Ognuno di essi è composto da sette Darlington NPN con l'uscita collegata (mediante una

resistenza) ad un input e il collettore connesso ad un output; tutti gli emettitori sono in comune e i collettori hanno un diodo di protezione collegato con l'anodo ad un punto comune, corrispondente al piedino 16. I Darlington funzionano in modo open-collector,

quindi chiudono a massa gli utilizzatori, i quali devono perciò essere collegati verso il positivo di alimentazione. I diodi interni all'ULN2003 servono per proteggere i transistor quando vengono usati per commutare utilizzatori induttivi, i quali, all'interdizione dei Darlington, generano extratensioni inverse capaci di far saltare le giunzioni di collettore. Se si vogliono usare i diodi, il piedino 16 deve essere collegato allo stesso potenziale positivo che alimenta gli utilizzatori. Il nostro ricevitore si alimenta con $12\div 15$ Vcc applicati ai punti + e - PWR ed è protetto dal diodo D1, che blocca la corrente nel caso l'alimentazione venga applicata alla rovescia; la tensione a valle del D1, filtrata da C1 e C2 (che la ripuliscono da disturbi impulsivi captati dai fili di collegamento e dal residuo d'alternata dell'alimentatore AC/DC) serve a far funzionare i circuiti di uscita, ovvero le bobine dei relé e i due line-driver. Il resto, ossia il microcontrollore e l'ibrido RX 4MHCS-4B, è completato da un regolatore integrato a tre terminali (il solito 7805) i cui 5 V di uscita sono filtrati da C3 e C4.

REALIZZAZIONE PRATICA

La costruzione del ricevitore è molto semplice perché richiede componenti a montaggio tradizionale e presenta come unica complessità la realizzazione di un circuito stampato a doppia ramatura, il quale si prepara per fotoincisione a partire dalle tracce lato rame che potete scaricare gratuitamente dal nostro sito. Nel realizzare lo stampato conviene impressionare prima una faccia della piastra ramata, quindi fare due fori per centrare la traccia della faccia opposta e poi procedere all'impressione di quest'ultima. In possesso dello

stampato, bisogna innanzitutto creare le vie, ossia infilare nelle piazzole comuni ai due lati della basetta uno spezzone di filo (o di reoforo tagliato da un componente) e stagnarli da entrambe le parti, in modo da interconnettere le due ramature. Iniziate il

montaggio con le resistenze e il diodo D1, quindi disponete gli zoccoli e il dip-switch, il pulsante P1 e i condensatori, dando la precedenza a quelli non polarizzati; proseguite con i relé e poi collocate il regolatore integrato 7805, quindi sistemate i due led e,



Ora il radiocomando ci porta anche la birra!

Sebbene l'estate sia ormai un ricordo, una birra fresca mentre guardiamo la partita in TV non è una brutta idea. Solo che come facciamo ad alzarci per andare a prenderla nel frigo senza perderci un'azione importante? Ci vorrebbe qualcuno disposto a farlo...ma se moglie o fidanzata sono fuori o se non abbiamo il maggiordomo? Per queste situazioni c'è RC Cooler, un magnifico robot radiocomandato che è insieme un contenitore per il fresco e un carrello che potrà portarci le bibite dalla cucina direttamente davanti al divano, sul bordo della piscina, sul terrazzo o in giardino. Basta mettere del ghiaccio e le bottiglie o lattine preferite (può contenere fino a 12 bottiglie da 25 cl o 10 lattine da 33 cl) nel suo cestello, quindi chiamarlo e dirigerlo con il radiocomando in dotazione per farlo arrivare dritto da noi, dopodiché non dobbiamo fare altro che tendere la mano e scegliere ciò che vogliamo bere. Indubbiamente un'originale applicazione del comando a distanza... Se volete vedere il robot all'opera, visitate <http://www.youtube.com/watch?v=betMng6DOK8>, oppure <http://www.firebox.com/product/2078/RC-Drinks-Cooler>.

Impostazione dei dip-switch.

SW1	Descrizione
	Programmazione tempo nel funzionamento astabile puro.
	Funzionamento bistabile.
	Funzionamento astabile ad inseguimento.
	Funzionamento astabile a tempo fisso (astabile puro).
	Attivazione funzione di ripristino nella modalità bistabile.

infine, il modulo ibrido. Per tutti i componenti polarizzati seguite l'orientamento indicato nel piano di montaggio; quanto all'RX 4MHCS-4B, entra nello stampato solo nel verso giusto. Per completare il ricevitore, montate delle morsettiere a passo 5 mm in corrispondenza delle piazzole OUT1÷OUT12 e di quelle siglate PWR. Completate le saldature, inserite nei rispettivi zoccoli gli ULN2003 e il microcontrollore già programmato. Ora il circuito è pronto: per utilizzarlo basta collegare ai punti ANT un'antenna, che può essere un semplice spezzone di filo di rame rigido lungo 17 cm o qualcosa di meglio: se usate lo spezzone di filo stagnato nella piazzola che porta al piedino 3 dell'ibrido, mentre se pensate di adoperare un'antenna ground-plane o una direttiva, connettetela mediante cavo schermato coassiale RG58, la cui calza metallica andrà al terminale di massa.

COME SI USA

Bene, giunti a questo punto non resta che collaudare e programmare il radiocomando: una volta in possesso del trasmettitore, alimentate il ricevitore con un alimentatore capace di fornire 12 V e una corrente continua di 500 mA, quindi verificate che tutti

i relé siano a riposo e che i led siano spenti. Affinché il ricevitore risponda al trasmettitore di cui siete in possesso, dovete abbinare i due, fermo restando che TX ed RX devono essere dello stesso lotto in quanto altrimenti cambia una parte fondamentale del codice e l'abbinamento non riesce. Acquistando trasmettitore e ricevitore dalla ditta Futura Elettronica, non ci saranno problemi. Per l'abbinamento bisogna premere e rilasciare il pulsante di apprendimento (P1); fatto ciò, il led LD1 lampeggia rapidamente per 10 secondi. Durante questo tempo, bisogna premere uno qualsiasi dei dodici tasti del trasmettitore, ovviamente restando a una distanza che permetta la ricezione ottimale del segnale (di solito l'apprendimento si fa sul posto). L'invio del comando viene segnalato dall'accensione del led sul pannello frontale del TX. L'avvenuta memorizzazione del codice base e del sincronismo del trasmettitore vengono indicate dall'LD1 con un lampeggio lungo. Notate che non serve ripetere l'operazione con gli altri tasti, perché basta trasmettere con un pulsante per essere certi che tutti gli altri vengano riconosciuti; infatti l'apprendimento acquisisce il codice base, che è unico per il trasmettitore, mentre la parte che descrive i pulsanti è quella variabile. È possibile ripetere questa procedura per fare apprendere al ricevitore RX 4MHCS-4B fino ad un massimo 10 trasmettitori TX-12CH. Per rimuovere dalla memoria del ricevitore i trasmettitori abbinati, bisogna operare la cancellazione totale della memoria (non è possibile cancellare un solo codice); questa procedura si svolge premendo e rilasciando il pulsante P1, quindi, appena il led LD1 inizia a lampeggiare, premendo nuovamente lo stesso

P1 (ciò provoca l'accensione del led a luce fissa) e tenendolo premuto fin quando il led non si spegne. Rilasciato P1, LD1 deve emettere 5 lampeggi per indicare che la memoria dei codici è stata cancellata. Questa operazione va eseguita dopo aver montato il ricevitore, prima di fare l'abbinamento col ricevitore, in modo da rimuovere eventuali dati casuali che possono trovarsi nella EEPROM dell'ibrido.

Una volta abbinato almeno un TX, impostate come volete il dip-switch SW1 e provate a premere un tasto del trasmettitore: vedrete il led bicolore pulsare a luce verde al ricevimento di ogni trasmissione.



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono di facile reperibilità. Il master del circuito stampato può essere scaricato gratuitamente dal sito della rivista così come il firmware utilizzato per programmare il microcontrollore. Quest'ultimo è anche disponibile già programmato (cod. MF846, Euro 12,00). Il ricevitore a 12 canali è disponibile in scatola di montaggio al prezzo di 56,00 Euro (cod. FT846); il kit comprende tutti i componenti, le minuterie, il micro già programmato ed il modulo Aurel. Quest'ultimo è anche disponibile separatamente (cod. RX-4MHCS-4B) al prezzo di 17,00 Euro. Il trasmettitore viene fornito già montato a 52,00 Euro (cod. TX-12CH). 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>

MISURATORE DI CONSUMO DI ELETTRICITÀ CON ZIGBEE

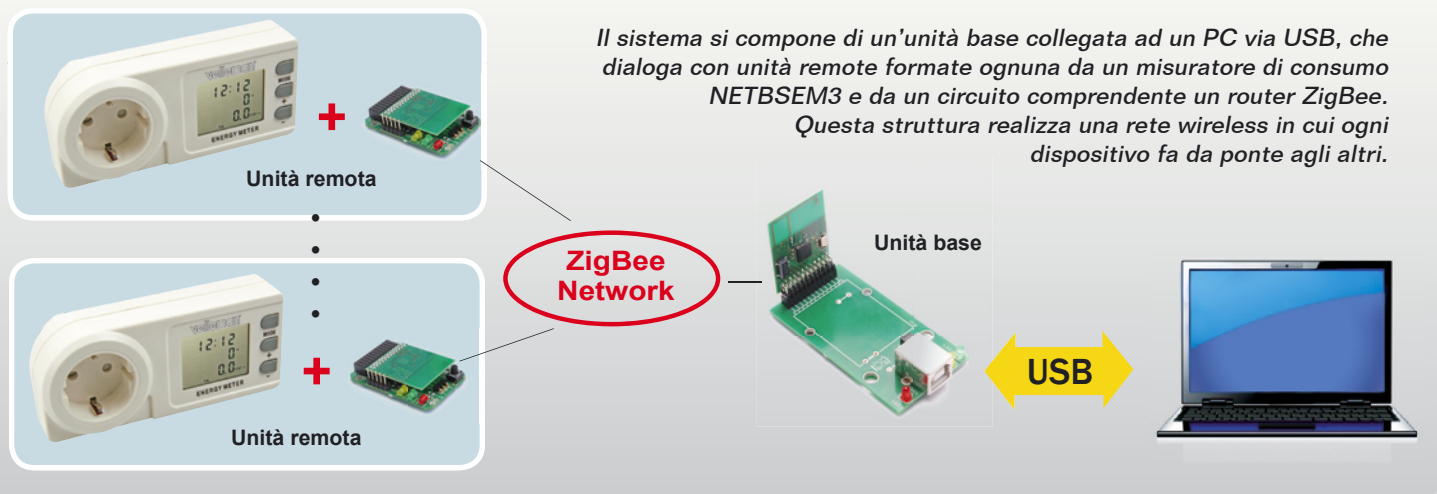
Dopo il corso sullo ZigBee, realizziamo un primo progetto con i moduli XTR-ZB1 Aurel: una rete wireless di dispositivi in grado di effettuare misure del consumo energetico delle utenze di un impianto elettrico domestico. Prima puntata.



di FRANCESCO FICILI

Una delle caratteristiche della società nella quale viviamo ai nostri giorni è la sua dipendenza dalle fonti energetiche, che si è notevolmente accentuata negli ultimi anni soprattutto nel nostro Paese, notoriamente dipendente dall'estero per l'approvvigionamento di petrolio, carbone e gas naturale, ma anche di elettricità, che comperiamo in una buona percentuale dalla Francia. Nelle nostre case, tra Personal Computer, televisori, lettori DVD, impianti Hi-Fi ed elettrodomestici di ogni genere, le utenze domestiche caratterizzate da un consumo

elettrico "importante" sono decisamente un gran numero. Da questa ed altre considerazioni nasce la necessità di monitorare e storicizzare i dati di consumo relativi agli apparati elettrici ed elettronici presenti nelle nostre case. Il dispositivo presentato in quest'articolo può essere efficacemente utilizzato per monitorare i consumi energetici di un qualsiasi impianto in bassa tensione o di semplici utilizzatori. L'idea di base è quella di utilizzare dei misuratori di consumo commerciali e, nello specifico, l'energy meter NETBSEM3, prodotto dalla Velleman e



distribuito da Futura Elettronica, illustrato nella **Fig. 1**; più esattamente, abbiamo abbinato tale dispositivo ad una piccola scheda a microcontrollore, equipaggiata con un PIC18F2550 ed un radio-modem ZigBee XTR-ZB1.

Il PIC funge da elemento di interfaccia tra il misuratore e la rete ZigBee, quindi tramite un'opportuna richiesta da parte del coordinator (a sua volta istruito dai comandi inviati dall'interfaccia Software sul bus USB), effettua la lettura del consumo energetico e manda i dati corrispondenti via radio al Coordinator.

PRINCIPIO DI FUNZIONAMENTO

La nostra proposta è realizzare un sistema modulare in cui ognuno dei moduli è un misuratore di potenza Velleman, modificato in modo da estrarre i dati sul consumo e permettere all'interfaccia

ZigBee di leggerli e trasmetterli via radio; un'unità base, provvista di interfaccia USB e ZigBee, riceve i dati dalle periferiche e li invia ad un PC, nel quale un opportuno software raccoglie le informazioni distinguendole cronologicamente per misuratore di provenienza.

Nei fascicoli da dicembre 2008 ad aprile 2009, abbiamo presentato un corso sul protocollo ZigBee, dove sono stati descritte le caratteristiche dello standard ed alcuni esempi di utilizzo dei dispositivi ZigBee Aurel denominati XTR-ZB1.

Dopo un po' di teoria sull'argomento, con il progetto di queste pagine analizziamo come può essere affrontato e risolto un problema reale, come la creazione di una rete wireless per la raccolta dei dati relativi al consumo energetico delle utenze di una abitazione.

Per prima cosa vediamo come funzionano i componenti hardware del sistema, partendo dall'energy meter NETBSEM3 della Velleman: questo dispositivo è equipaggiato con l'ADE7755, un integrato prodotto dalla Analog Devices, appositamente progettato per funzionare da misuratore

energetico. La **Fig. 2** illustra piedinatura e schema a blocchi del componente.

L'ADE7755 è un misuratore di energia elettrica ad alta risoluzione; è composto da una coppia di canali analogici d'ingresso, per il campionamento delle forme d'onda di corrente e tensione, e da una logica proprietaria che effettua la conversione del dato relativo alla potenza reale in formato digitale. Nello specifico, il valore della potenza in watt viene convertito in un treno di impulsi. L'output può essere prelevato dalla coppia di pin F1 ed F2, che forniscono l'informazione sulla potenza media, o dal pin CF, che fornisce un treno di impulsi proporzionale al valore della potenza istantanea. Il pin CF è particolarmente indicato per l'interfacciamento con microcontrollori, in quanto può essere collegato ad un pin di una porta di I/O di un microcontrollore e da questo possono essere letti gli impulsi e ricavate le informazioni sul consumo. Un esempio qualitativo delle forme d'onda in uscita dai pin F1/F2 e CF è rappresentato nella **Fig. 3**.

Il principio di funzionamento dell'ADE7755 si basa sul calcolo diretto del prodotto corrente-tensione, per risalire al valore della potenza. Per comprendere meglio il principio di funzionamento fate riferimento allo schema funzionale riportato nella **Fig. 4**. Come si vede, tramite i due canali

Fig. 1 - Unità periferica: si basa sul misuratore NETBSEM3 Velleman, che è dotato di una spina da inserire nella presa di rete e di una presa per alimentare un utilizzatore.



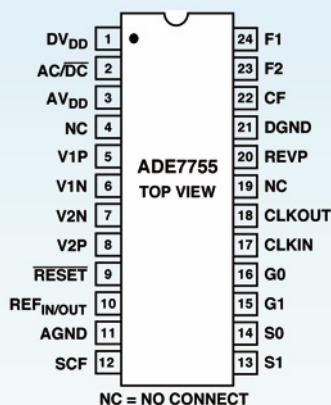


Fig. 2
Pin-out
e schema
a blocchi
dell'ADE7755.

li analogici CH1 e CH2 vengono acquisite le forme d'onda di corrente e tensione, le quali poi sono sovracampionate alla frequenza di 900 kHz e la corrente viene inoltre filtrata con un filtro passa-alto per eliminare eventuali componenti DC. A questo punto i due segnali vengono moltiplicati per ottenere la potenza istantanea. Il segnale che ne risulta viene filtrato con un filtro passa-basso per ottenere la potenza reale; la tensione corrispondente viene convertita in frequenza mediante un convertitore tensione/frequenza, generando la coppia di segnali F1-F2 ed il segnale CF. Si noti come tutte le operazioni sui segnali (dai due blocchi ADC in poi) sono realizzate nel dominio digitale, in modo da fornire la massima stabilità in temperatura ed evitare derive nel tempo. L'interfacciamento con il PIC18F2550 è stato effettuato

utilizzando il pin CF. Il segnale CF è stato collegato alla porta RA0 del PIC, configurata come ingresso digitale. Tramite un apposito firmware vengono contati gli impulsi rilevati su di essa in un determinato periodo di integrazione e così viene ricavata l'informazione sulla potenza istantanea. Chiaramente, nel caso di consumi molto bassi, poiché il segnale CF è asincrono rispetto al clock del PIC, esiste la possibilità di

perdere qualche fronte, condizione che causa una certa imprecisione e la comparsa di un'ondulazione nella misura (vedere la Fig. 5). Ecco perché il periodo di integrazione è stato portato a 4 secondi, in modo da ridurre l'errore nella misura. Una volta ottenuta l'informazione relativa al consumo di potenza dell'utenza domestica collegata, il problema si riduce al trasporto dell'informazione di ogni utenza verso un'unica centrale di raccolta dei dati, che nel nostro caso è un Personal Computer. Tale problema si adatta particolarmente bene alla soluzione con dispositivi ZigBee: infatti, poiché l'oggetto della misura è costituito dall'energia consumata da una determinata utenza elettrica, la sorgente di alimentazione è già presente sul posto. In caso contrario, il cablaggio di eventuali linee di segnale (nel caso vengano utilizzati dispositivi wired) potrebbe risultare piuttosto difficoltoso e sconsigliato dal punto di vista economico,

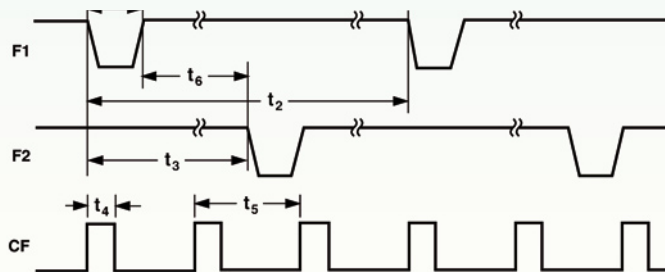


Fig. 3
Forme d'onda
in uscita dai
pin F1/F2
e CF del-
l'ADE7755.

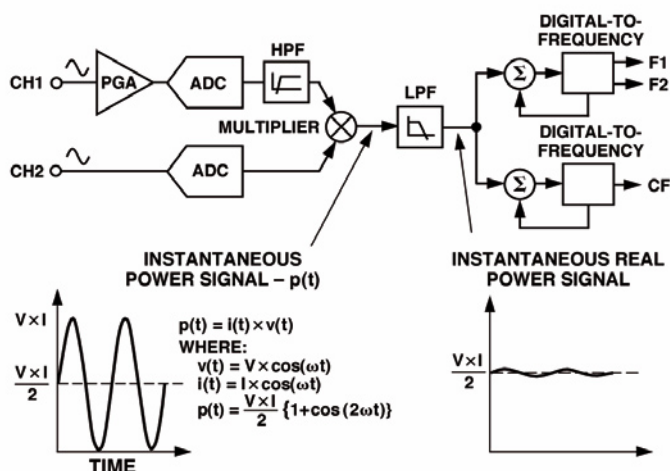
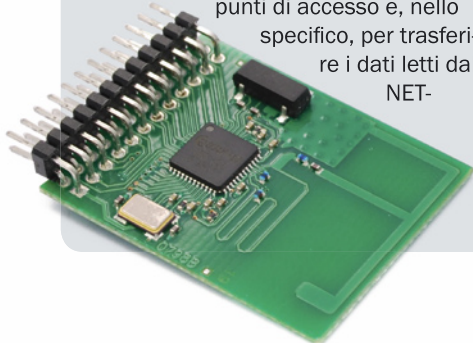


Fig. 4
Principio di
funzionamento
dell'ADE7755.

I moduli ZigBee

Per il nostro progetto impieghiamo due moduli ZigBee dell'Aurel: un router (XTR-ZB1-RLI) utilizzato per realizzare punti di accesso e, nello specifico, per trasferire i dati letti dal NET-



BSEM3 all'unità centrale; un coordinator (XTR-XB1-CLI) usato, in abbinamento a un'interfaccia USB, per realizzare l'unità base che trasferisce al computer i dati in arrivo dai router. I due moduli sono in apparenza uguali, dato che usano lo stesso hardware, ma differiscono nel firmware, in quanto uno funge da nodo della rete mesh e l'altro è, invece, il coordinatore dei router.

perché non sono quasi mai disponibili linee già cablate adatte allo scopo. L'uso di trasmettitori wireless elimina questo problema e la messa in opera della rete di monitoraggio (se ben concepita) si riduce al posizionamento dei vari nodi di misura ed al collegamento delle utenze che si intende monitorare. A questo punto la creazione ed il mantenimento della rete diventa un problema puramente software.

La struttura di rete utilizzata nel sistema che vi proponiamo è una mesh ZigBee, il cui Coordinator è collegato ad un Personal Computer tramite collegamento USB

ed oltre ad assolvere alle funzioni classiche di un Coordinator ZigBee, funge anche da elemento di raccolta dati della rete.

Ogni Router è associato ad un misuratore di consumo e quindi ad ogni nodo è associata l'informazione sul consumo di una determinata utenza elettrica. Naturalmente alle rete siffatta possono essere aggiunti anche dispositivi di tipo End Device, tuttavia questi ultimi non garantiscono l'estensione della mesh, in quanto non possono funzionare da ripetitori. Bisogna quindi fare la dovuta attenzione nel posizionamento dei vari nodi. Inoltre è bene ricordare che nell'applicazione qui proposta, la soluzione adottata (ossia una

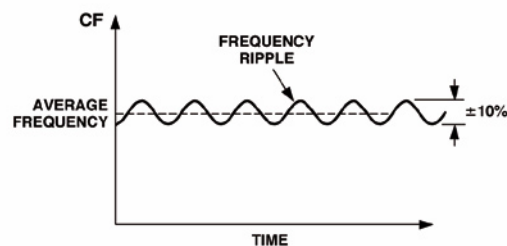


Fig. 5 - Ripple della frequenza di uscita dal pin CF e frequenza media.

rete composta da soli router) rappresenta la soluzione migliore, almeno se i nodi sono in numero limitato.

IL NODO MISURATORE

Come già accennato, il sistema proposto è costituito da una serie di nodi di misura (costituiti da router ZigBee, interfacciati con i misuratori di consumo NETBSEM3 Velleman) collegati in wireless ad un unico data-collector, il quale non è altro che il coordinator della PAN ZigBee. Il coordinator è a sua volta collegato tramite un circuito di interfaccia ad un PC, e viene gestito da quest'ultimo attraverso un software di controllo e raccolta dei dati. Come si vede, lo schema elettrico dei nodi di misura è piuttosto sempli-



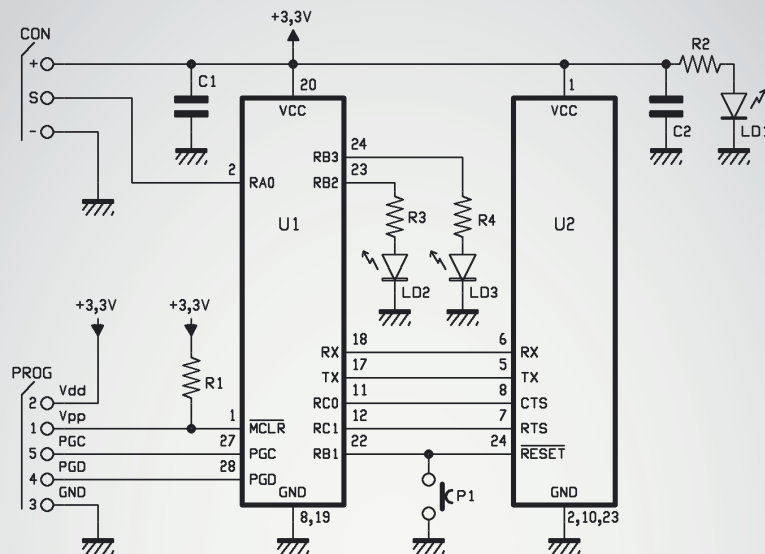
Durante i 7 mesi di ora legale (29 marzo÷25 ottobre) grazie a quell'ora di luce in più al giorno che permette di ritardare l'utilizzo della luce artificiale, sono stati risparmiati in totale 643 milioni di kilowattora: un valore pari alla metà dei consumi domestici annui del Friuli Venezia Giulia. In termini di costi, considerando che per il cliente finale 1 kilowattora è costato, nel periodo in esame, in media circa 14,5 centesimi di

Un'ora per risparmiare

euro al netto delle imposte, l'Italia ha risparmiato con l'ora legale 93 milioni di euro. Dal 2004 al 2009 il risparmio complessivo del Paese è stato di 3,7 miliardi di kilowattora, pari ad un valore di circa 500 milioni di euro. Nel dettaglio, ad aprile si è avuto un risparmio di oltre 169 milioni di kW/h, a maggio di 87 milioni di kW/h, a giugno di 34,7 milioni di kW/h, a luglio di 35,6 milioni di kW/h, ad agosto di 38 milioni di kW/h, a settembre di 85,5 milioni di kW/h e ad ottobre di quasi 193 milioni di kWh. I mesi di Aprile e Ottobre hanno fatto registrare, come di consueto, il maggior risparmio di energia elettrica. Ciò è dovuto al fatto che questi due mesi hanno giornate più "corte" rispetto ai mesi dell'intero periodo.

Il router

ce. L'unità centrale è costituita dal microcontrollore U1, un PIC18F2550 interfacciato ad un modulo Aurel XTR-ZB1-RLI, quindi ad un router ZigBee low-power e con antenna integrata. Come bus di interfaccia è stata utilizzata la porta USART del PIC (RC6 ed RC7). La porta è stata configurata per lavorare a 38.400 bps, con 8 bit di dati, nessuna parità ed 1 bit di stop, ed inoltre è stato abilitato il controllo di flusso hardware (RC0 ed RC1). Nella scheda trovano posto i connettori P1, per la programmazione ICSP del PIC, e P2, attraverso il quale viene prelevata l'alimentazione ed il segnale relativo al consumo energetico proveniente dal modulo Velleman. Sono inoltre stati aggiunti i condensatori C1 e C2, da 100 nF, il primo sulla linea di alimentazione del PIC ed il secondo sulla linea di alimentazione del modulo radio, allo scopo di filtrare eventuali disturbi che il modulo ZigBee può propagare sull'alimentazione del microcontrollore. Infine sono stati inseriti i LED LD1, LD2 e LD3 (con le relative resistenze di limitazione) ed il pulsante SW1. Il led LD1 è stato collegato direttamente alla linea di alimentazione, in modo da segnalare la presenza di tensione sulla scheda. LD2 e LD3 sono collegati sulla porta B del PIC (precisamente RB1 ed RB2) e vengono utilizzati come spie, oltre ad essere particolarmente utili in fase di debug della rete. Infine il pulsante P1 è anch'esso collegato sulla porta B (RB0, in modo da sfruttare la rete di pull-up interna) ed è utilizzato come pulsante generico. Non è stato usato alcun quarzo, in quanto per il funzionamento del circuito è più che sufficiente l'oscillatore RC interno del PIC, in grado di garantire frequenze di lavoro fino ad 8 MHz (2 MIPS).

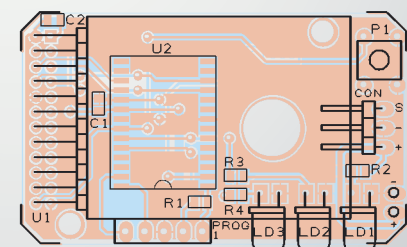
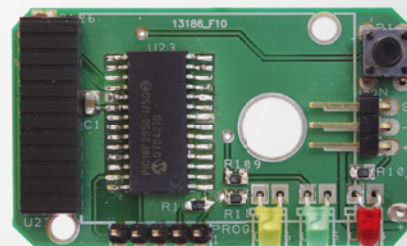


Elenco Componenti:

- R1: 4,7 kohm (0805)
- R2÷R4: 220 ohm (0805)
- C1, C2: 100 nF multistrato (0805)
- U1: XTR-ZB1-RLI
- U2: PIC18F2550-I/SO (MF813)
- LD1: Led 3 mm verde
- LD2: Led 3 mm rosso
- LD3: Led 3 mm giallo
- P1: Microswitch

Varie:

- Pin strip femmina 2x12 vie 90°
- Pin strip maschio 3 vie 90°
- Pin strip maschio 5 vie
- Circuito stampato



SCHEMA ELETTRICO DELL'INTERFACCIA USB

Vediamo adesso lo schema elettrico del circuito di interfaccia tra il coordinator ed il PC, che riceve i dati dai nodi della mesh e li invia convertiti in formato compatibile con l'USB. In questo caso il cuore del sistema è costituito dal convertitore USB-UART FT232RL, prodotto dalla compagnia scozzese FTDI. Il modulo radio utilizzato è l'XTR-ZB1-CLI, ossia un Coordinator low-power, con antenna integrata. Il modulo radio è collegato all'FT232 tramite la por-

ta UART, a 34.800 bps con l'uso di criteri di controllo hardware. A sua volta il modulo FT232RL viene collegato al connettore USB tramite le linee D+ e D-. L'uso dell'FT232RL rende molto semplice l'interfacciamento del modulo radio col bus USB, in quanto tutta la gestione del protocollo viene realizzata dal chip in maniera totalmente trasparente per l'utente. L'alimentazione del chip FT232RL e del modulo radio è affidata al regolatore LDO a 3,3 V MCP1702, insieme alla sua rete di condensatori stabilizzatori (C4 e C5). All'interno del cir-

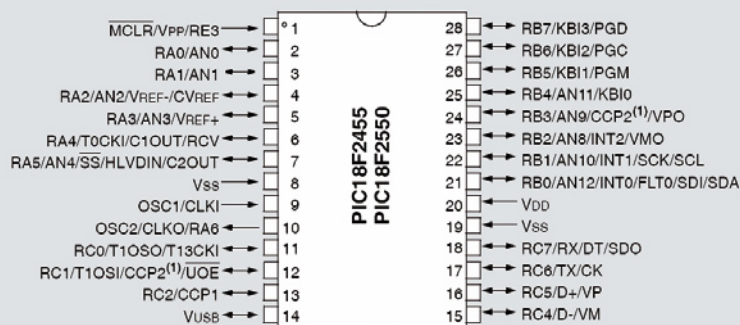
Il microcontrollore PIC18F2550

Il microcontrollore utilizzato in questo progetto è un PIC18F2550, uno dei più interessanti prodotti della famiglia PIC18F, prodotta da Microchip. Il PIC18F2550 risulta la scelta ideale per una grossa quantità di applicazioni embedded, grazie alla sua notevole quantità di memoria flash integrata ed alla ricca dotazione di memoria RAM e periferiche, tra le quali figurano porte USART, I²C, SPI, un convertitore A/D

Tipo di memoria di programma	Flash
Capacità memoria di programma	32 kB
Velocità CPU	12 (MIPS)
Capacità RAM	2.048 byte
Capacità data EEPROM	256 byte
Periferiche Digital Communication	1-A/E/USART, 1-MSSP (SPI/I ² C)
Periferiche Capture/Compare/PWM	2 CCP
Timer	1 x 8-bit, 3 x 16-bit
ADC	10 CH, 10-bit
Comparatori	2
Interfaccia USB	1 Full-speed, USB 2.0

dotato di 10 canali ed una porta USB device.

Le caratteristiche peculiari di questo RISC ad 8-bit sono elencate nella tabella illustrata qui sopra.



cuito trovano ancora posto i led LD1 ed LD2 (collegati alle linee Cbus0 e Cbus1 dell'FT232RL) che fungono da segnalatori di traffico sul bus USB, ed il led LD3, che indica la presenza di tensione sulla scheda. Infine segnaliamo la presenza dei condensatori C1, C2 e C3, tutti multistrato da 100 nF,

in parallelo alle linee di alimentazione dei due moduli ed in uscita alla tensione del bus USB, utilizzati per sopprimere i disturbi sulle linee di alimentazione.

REALIZZAZIONE E MONTAGGIO

I due circuiti presentati in quest'articolo sono realizzati quasi

interamente in tecnologia SMD, quindi spendiamo qualche riga per spiegare ai meno esperti di montaggio di questa tipologia di componenti quali accortezze usare. Iniziamo dalla scheda di interfaccia tra il Coordinator ed il PC: la scelta più conveniente consiste nel posizionare innanzi tutto i componenti passivi, ossia resistenze e condensatori SMD. Per la saldatura dei componenti SMD munitevi di una saldatore a punta sottile, un paio di pinzette per montaggi SMD, il classico rocchetto di stagno senza piombo e un po' di treccia dissaldante. Le resistenze ed i condensatori utilizzati in questo progetto sono stati scelti con case SM0805, che è relativamente facile da saldare. La tecnica utilizzata è la seguente: stagnate una delle due piazzole con una piccolissima goccia di stagno, poi prendete il componente con le pinzette, avvicinate-lo alla piazzola stagnata e fatelo aderire utilizzando il saldatore, posizionandolo in modo che l'altro terminale aderisca alla seconda piazzola; a questo punto allontanate delicatamente il saldatore e lasciate raffreddare. Completate l'opera stagnando il terminale opposto. Nel caso non abbiate mai saldato un componente in SMD, è probabile che le

Il misuratore NETBSEM3



Il misuratore di consumo utilizzato in questo progetto è l'energy meter prodotto dalla Velleman e distribuito in Italia da Futura Elettronica; questo interessante dispositivo permette di conoscere il consumo ed il costo dell'energia consumata da una qualsiasi apparecchiatura funzionante a tensione di rete. Per usarlo è sufficiente collegare il dispositivo ad una presa di corrente domestica e poi utilizzare la presa integrata sul dispositivo stesso per alimentare l'utenza (o le utenze) domestiche che si desidera monitorare. Le caratteristiche principali del NETBSEM3 sono le seguenti.

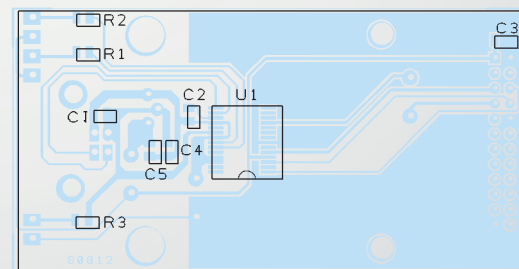
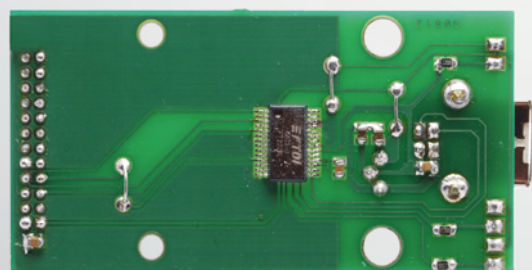
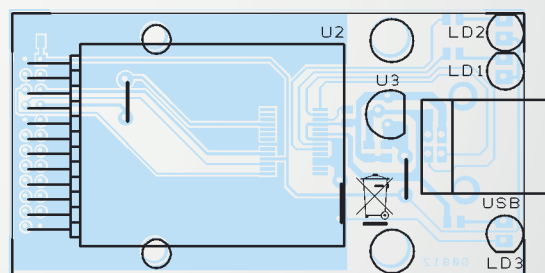
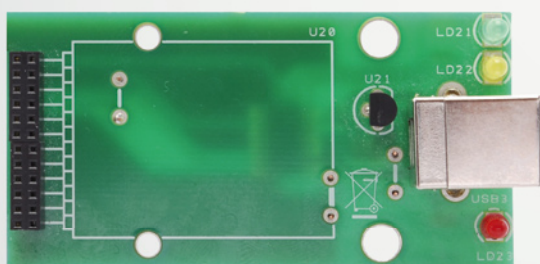
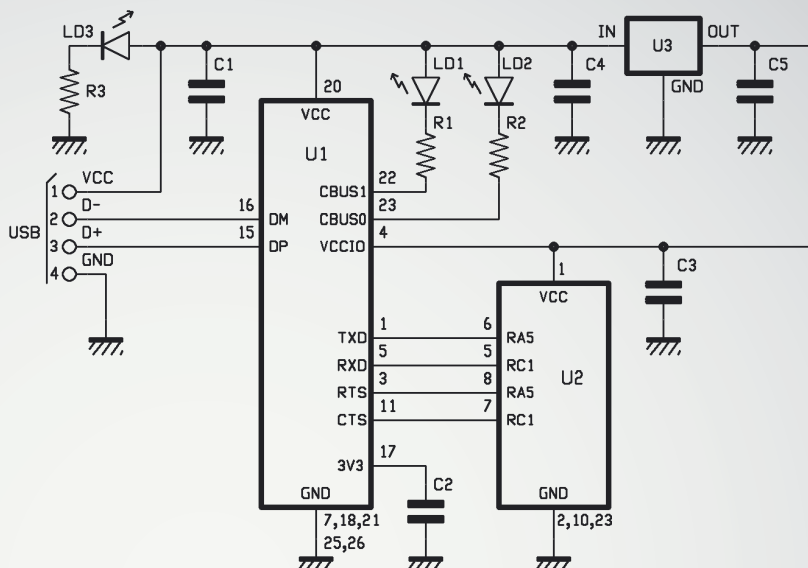
- Massimo e minimo carico applicabile: 3.600 e 5 W;
- tensione di esercizio: 220 Vca;
- temperatura d'esercizio: 10÷40 °C;
- consumo mostrato sul display: 0 ~ 999.9 kW/h;
- massimo tempo di registrazione: 9.999 h.

Elenco Componenti:

R1 ÷ R3: 220 ohm (0805)
 C1 ÷ C5: 100 nF multistrato (0805)
 U1: FT232RL
 U2: XTR-ZB1-CLI
 U3: MC1702-3302
 LD1: Led 3 mm giallo
 LD2: Led 3 mm verde
 LD3: Led 3 mm rosso

Varie:

- Pin strip femmina 2x12 vie 90°
- Connettore USB-B
- Circuito stampato



prime volte il risultato non sia esaltante, ma una volta acquisita la dovuta manualità inizierete a saldare i componenti velocemente e senza sbavature. Una volta completato il montaggio di tutte le resistenze e dei condensatori, potete passare ai led, che sono a montaggio tradizionale; anche la saldatura del connettore USB e dello zoccolo per il radiomodem XTR-ZB1 è di tipo normale, quindi non presenta particolari difficoltà. Particolare attenzione la merita il montaggio del-

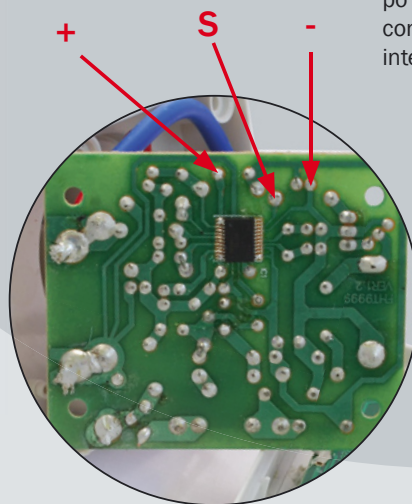
l'FT232, che è incapsulato in un package SSOP28, i cui pin sono molto vicini tra loro e quindi è necessaria un po' di cautela. La procedura è simile a quella vista in precedenza per le resistenze: per prima cosa staginate una delle piazzole alle estremità (ad esempio quella corrispondente al pin 1); una volta staginata la piazzola, posizionate il componente con le pinzette e fatelo aderire scaldando contemporaneamente la piazzola con il saldatore. Fate attenzione a non invertire

l'orientamento del chip, altrimenti ad operazione completata sarà un grosso problema rimediare a quest'errore. Ora, utilizzando le pinzette ed il saldatore, posizionate il chip in modo che tutti i pin aderiscano alle rispettive piazzole. Una volta fatto ciò, allontanate il saldatore e lasciate raffreddare, quindi saldate i rimanenti tre pin alle estremità. Ora, per completare l'operazione potete saldare ognuna delle due file di pin. Non preoccupatevi se lo stagno straborda ed invade le

Modificare il NETBSEM3

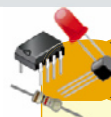
Per rilevare i consumi di elettricità, il modulo di interfaccia con router ZigBee deve prelevare dall'integrato ADE7755 del NETBSEM3 il segnale ad impulsi che indica la potenza istantanea; allo scopo dobbiamo aprire il dispositivo Velleman, svitare dal proprio alloggiamento la scheda collegata all'alimentazione di rete (quella nel semiguscio che riporta la spina) e girarla fino a mostrare il lato delle piste. A questo punto occorre identificare le piazzole dei

piedini 1 (positivo di alimentazione dell'ADE7755) 21 (massa comune) e 22 (CF); ad esse, bisogna saldare tre fili, che portano al connettore CON del circuito router. I fili vanno così collegati: il contatto del piedino 1 va sul + (per prelevare l'alimentazione occorrente al micro ed al router ZigBee) quello che viene dal pin 21 va al - (massa) e, infine, il contatto 22 dell'ADE7755 va al punto S del CON. Con questa connessione, colleghiamo l'uscita dell'ADE775 al microcontrollore del modulo router e nel contempo alimentiamo quest'ultimo con il regolatore di tensione interno al NETBSEM3.



piazzole vicine, perché una volta raffreddato potrete eliminare quello in eccesso mediante della trecciola dissaldante. Cercate di effettuare le saldature in fretta e non accanitevi sullo stesso pin troppo a lungo o rischierete di danneggiare il componente. Una volta montato l'FT232 la scheda di interfaccia USB è ultimata; a questo punto potete posizionare l'XTR-ZB1 sul suo zoccolo. Il montaggio delle schede che ospitano i router è simile a quanto descritto prima. Oltre ai soliti componenti passivi, è presente un PIC18F2550 con case SOIC28, leggermente più grande dell'FT232, per la saldatura del quale, una volta fatta un po' di pratica, non dovrete avere problemi. Completato il montaggio, è buona norma ricontrollare visivamente il tutto alla ricerca

di cortocircuiti o saldature non perfette. Nei casi dubbi servitevi sempre del vostro fido tester. Completate e verificate le unità, è possibile testare la comunicazione tra Coordinator e Router, utilizzando le funzioni di debug dell'apposito software di gestione Demo Aurel fornito in dotazione allo starter kit ZigBee Aurel. Concludiamo qui questa prima puntata: vi diamo appuntamento al prossimo mese, dove verrà analizzato il firmware con cui è stato programmato il PIC18F2550 (realizzato con l'ambiente di sviluppo Microchip MPLab IDE), per garantire l'interfacciamento con i radiomodem XTR-ZB1, oltre all'interfaccia software, realizzata in VB6, per il controllo della rete ed il monitoraggio dei consumi ricavati dai NETBSEM3. ■



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono di facile reperibilità. Il master del circuito stampato, il firmware per programmare il microcontrollore PIC18F2550 ed il software di gestione per il PC sono scaricabili gratuitamente dal sito della rivista. Il misuratore NETBSEM3 costa 17,50 Euro mentre i due dispositivi ZigBee dell'Aurel costano entrambi 35,00 Euro, router (XTR-ZB1-RLI) e coordinator (XTR-ZB1-CLI). Il convertitore FT232RL costa 7,00 Euro. I prezzi si intendono IVA compresa.

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



Su scala nanometrica la materia presenta proprietà completamente differenti rispetto alla scala macroscopica. La crescente capacità di manipolare la materia a questo livello apre nuove possibilità di impiego in numerosi campi, compreso quello elettronico e della produzione di energia. Vediamo quale impatto ha già oggi sui dispositivi elettronici l'applicazione di questa tecnologia e quali sviluppi porterà in futuro.

NANOTECNOLOGIE & ELETTRONICA

di ARSENIO SPADONI

Con i termini nanoscienza e nanotecnologia ci si riferisce ad una vasta gamma di pratiche scientifiche in diversi campi del sapere. Nella terminologia delle unità di misura, scendendo la scala delle grandezze, *nano* (dal greco *nânos* che significa minuscolo) è il prefisso che segue *micro* (*mikròs*, in greco piccolo).

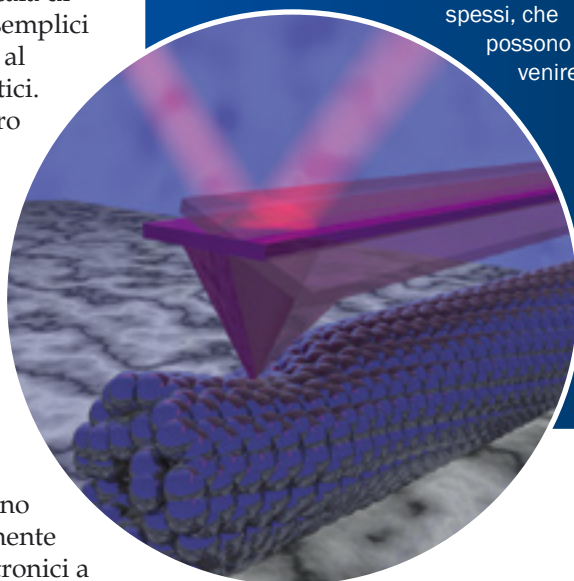
La nascita di queste discipline si fa risalire al 1959, quando il fisico Richard Feynman - futuro premio

Nobel - nel mese di dicembre tenne una conferenza presso il California Institute of Technology con un titolo bizzarro e allo stesso tempo affascinante: *There's Plenty of Room at the Bottom* ("C'è un sacco di spazio lì sotto").

Partendo dall'osservazione che nel XX secolo i fisici avevano ottenuto grandi successi nell'esplorazione dell'universo sub-atomico, ma che rimaneva ancora molto da fare nello studio delle proprietà

della materia di una scala leggermente più grande, Feynman profetizzò che questo studio avrebbe portato allo sviluppo di un numero enorme di applicazioni pratiche. Nel sistema metrico internazionale, l'unità di base per la misurazione della lunghezza è il metro (m). Dividendo il metro in mille parti si ottiene il millimetro (mm). A occhio nudo siamo in grado di distinguere oggetti distanti tra loro fino a circa 0,1 mm, perciò per osservare e lavorare su questa scala di grandezza si utilizzano semplici lenti di ingrandimento o al massimo i microscopi ottici. Se dividiamo il millimetro in mille parti otteniamo il micrometro o micron (μm). Per osservare strutture di queste dimensioni si utilizzano microscopi ottici o microscopi elettronici a scansione. Una ulteriore divisione per mille ci dà un nanometro (nm). Le strutture di queste dimensioni possono essere osservate direttamente solo con microscopi elettronici a scansione di sonda o con microscopi elettronici a trasmissione. Nel nostro viaggio verso l'infinitamente piccolo, se dividessimo il nanometro ancora una o due volte per mille (ottenendo rispettivamente un picometro e un femtometro) arriveremmo alla scala sub-atomica e ci imatteremmo nei costituenti ultimi della materia, le particelle elementari e, tra queste, quelle che compongono gli atomi: gli elettroni, i protoni e i neutroni. Il campo di ricerca delle nanoscienze e delle nanotecnologie (termine, quest'ultimo, coniato dallo scienziato giapponese Norio Taniguchi nel 1974) riguarda gli oggetti, i dispositivi ed i ma-

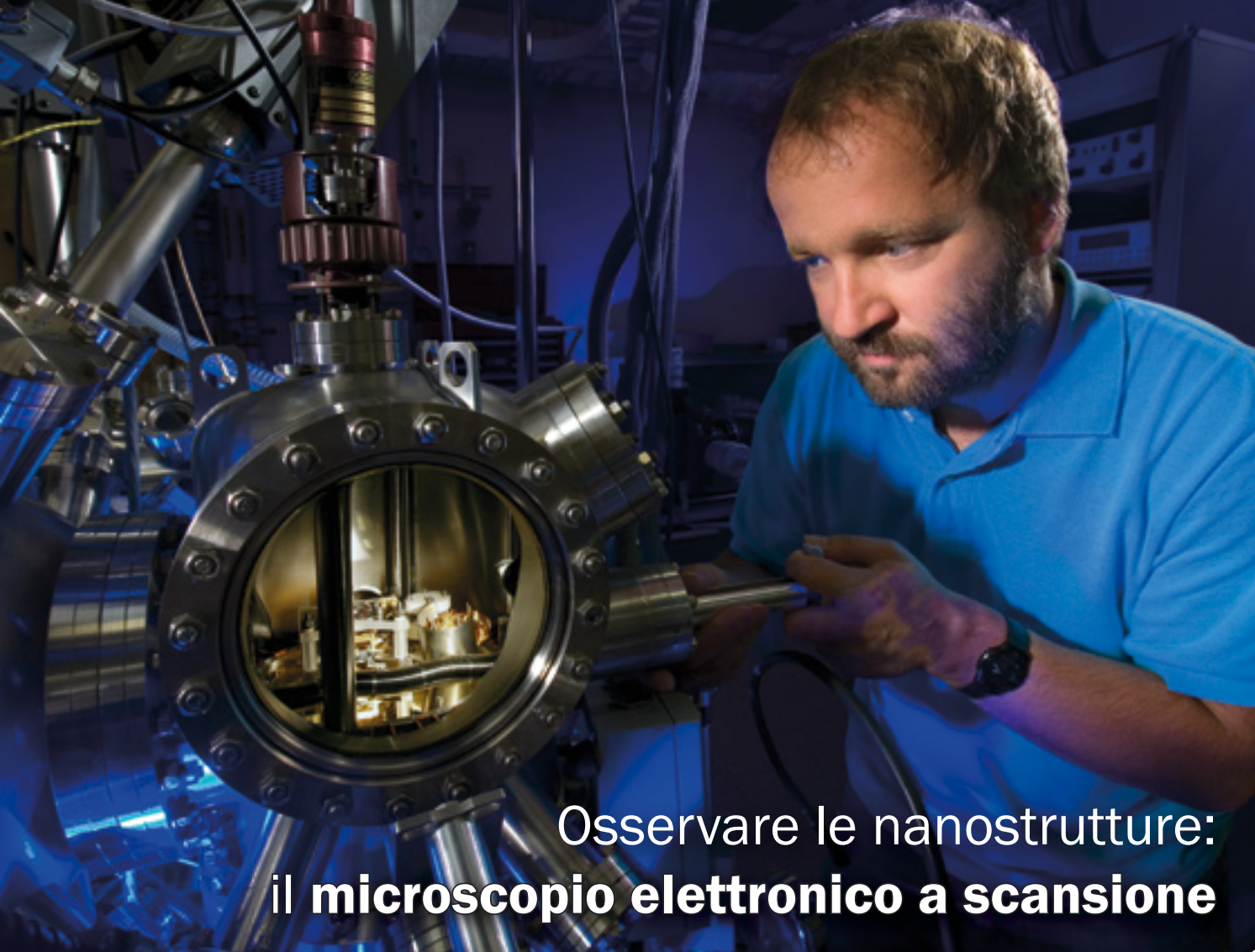
L'unico modo per osservare materiali e strutture a livello nanometrico è fare uso di microscopi elettronici, dispositivi che utilizzano un fascio di elettroni anziché la luce visibile. Ciò permette a questi strumenti di raggiungere una risoluzione superiore di vari ordini di grandezza a quella che si può ottenere con il normale microscopio ottico (che non può andare molto sotto l'ordine dei micrometri). Esistono due tipi principali di microscopi elettronici: il microscopio elettronico a scansione (SEM) e il microscopio elettronico a trasmissione (TEM). Mentre il TEM permette di osservare il campione "in trasparenza", e può quindi essere utilizzato solo con fettine ultra-sottili del campione mantenute perpendicolari al fascio di elettroni, con il SEM si osservano anche oggetti spessi, che possono venire



teriali che abbiano almeno una dimensione spaziale compresa tra 1 e 100 nanometri; sono quindi oggetto di studio di queste discipline gli aggregati di qualche centinaio di atomi, ma anche i *film sottili* di materiale di rivestimento, spessi appena qualche strato atomico ma estesi in lunghezza e larghezza su dimensioni macroscopiche, oppure i tubi di carbonio che hanno un diametro di pochi nanometri, anche se possono essere lunghi diversi millimetri. Poiché questo nuovo campo d'indagine è definito solo dalla scala dimensionale degli oggetti, non corrisponde a nessuna delle discipline in cui classica-

orientati e ingranditi durante l'osservazione. La superficie del campione viene infatti "spazzolata" da un fascio di elettroni e l'immagine viene costruita indirettamente a partire dagli elettroni riflessi e diffusi dalla superficie dell'oggetto. Appositi dispositivi consentono sia di orientare il fascio di elettroni che scansione il campione, sia il campione rispetto al fascio. L'oggetto deve però essere inserito in una camera a vuoto, per evitare il rumore causato dall'interazione degli elettroni con gli atomi che compongono i gas presenti nell'aria, nonché con la polvere ed altre impurità. Nella camera possono essere posizionati anche diversi campioni di dimensioni relativamente grandi (dell'ordine dei centimetri). Nelle tecnologie che operano sulla scala micrometrica e nanometrica, il microscopio elettronico a scansione (SEM) è uno strumento fondamentale in quanto permette di verificare molto rapidamente alcune caratteristiche principali delle strutture ottenute, come la lunghezza, il diametro, la densità, la forma, l'orientamento ed eventuali anomalie; nel caso dei nanofili, ad esempio, è molto importante capire come i vari parametri influenzino il risultato finale, perché questo rappresenta il primo passo per la realizzazione di nanofili con specifiche proprietà.

mente si suddivide il sapere; è un campo totalmente nuovo, in cui la ricerca è condotta integrando l'esperienza di molte discipline scientifiche tradizionalmente considerate come separate. In un laboratorio di nanotecnologie fisici, ingegneri, biologi, chimici e medici lavorano fianco a fianco, coinvolti in progetti di ricerca comuni; il mondo delle nanotecnologie è un territorio di ricerca in cui i problemi tecnici e teorici che si presentano di volta in volta richiedono soluzioni che necessitano competenze diversificate e un ampio bagaglio di conoscenze scientifiche multidisciplinari. Passare dalla scala del micro a quella del nano non significa solo fare un passo verso la miniaturizzazione, ma fare un vero e proprio balzo di conoscenza, scientifico e tecnologico insieme.



Osservare le nanostrutture: il microscopio elettronico a scansione

Passando dalla scala dei nano cominciano infatti a mostrare i loro limiti molte delle teorie e delle tecniche valide per lavorare a livello microscopico. Quando parliamo di dispositivi o strutture dalle dimensioni di qualche decina di nanometri stiamo considerando oggetti costituiti da poche decine o centinaia di atomi, che possono presentare proprietà (ottiche, elettromagnetiche e meccaniche) molto diverse da quelle di oggetti di dimensioni micro e macroscopiche. Un esempio di questo comportamento lo possiamo verificare nei metalli nobili (argento e oro) che, inerti su scala macroscopica, su scala nanometrica presentano una grande bio-reattività che li trasforma, rispettivamente, in un potente battericida, e in elemento dalle proprietà catalitiche nella

trasformazione del monossido di carbonio in anidride carbonica. Le sorprendenti proprietà della materia su scala nanometrica e le crescenti capacità di manipolare e aggregare atomi in modo controllato permettono già oggi di progettare nuovi materiali e realizzare dispositivi costituiti da parti piccolissime (nano componenti) con proprietà e caratteristiche completamente differenti da quelle dei materiali oggi in uso. Si apre quindi un vasto orizzonte di possibili impieghi industriali che potranno, prima o poi cambiare le nostre abitudini. Questa scommessa sulla possibilità di utilizzare in oggetti di grande diffusione qualche inusuale proprietà della materia su scala nanometrica è alla base del grande interesse suscitato oggi dalle nanotecnologie.

NANOSTRUTTURE E FISICA QUANTISTICA

Le proprietà delle nanostrutture sono descritte dalla fisica quantistica, un settore della scienza moderna nato all'inizio del ventesimo secolo proprio nel tentativo di spiegare le prime osservazioni sperimentali sulla struttura atomica della materia e del comportamento della natura su scala nanometrica; un comportamento che si rivelò talmente sorprendente da non poter essere in alcun modo ricondotto alle leggi della fisica classica valida per gli oggetti macroscopici. Nella fisica quantistica gli elettroni non sono descritti come delle minuscole sfere dotate di carica elettrica, e il loro comportamento è assimilato a quello di onde che si propagano nello spazio e nel tempo, infrangendosi contro



Fullereni & nanotubi di carbonio

Gli studi a livello nanometrico hanno portato alla scoperta di nuove strutture con caratteristiche particolari, specie nei materiali organici; in campo elettronico i primi prodotti realizzati con questa scala sono già disponibili in commercio. Un esempio sono i display a colori di alcuni telefoni cellulari che contengono tiofeni e i recenti monitor flessibili realizzati su film polimerici contenenti pentacene.

Tra i nuovi materiali, tuttavia, il posto d'onore spetta ai fullereni e, tra questi, in particolare ai nanotubi di carbonio.

Di tutte le nuove molecole organiche che stanno rivoluzionando il mondo dell'hi-tech la prima ad essere scoperta è stata il C_{60} , una delle molecole più simme-

triche esistenti in natura: ha forma di un pallone di calcio ed è costituita da 60 atomi di carbonio disposti a formare 12 pentagoni e 20 esagoni. Il diametro del C_{60} è di circa 0,7 nm: un perfetto esempio di nano particella.

Nel 1985 Richard Smalley, Harold Kroto e Robert Curl ne scoprirono l'esistenza quasi per caso; mentre Smalley studiava i livelli energetici degli orbitali molecolari dell'atmosfera di alcune stelle si accorse che mostravano la presenza di una specie sconosciuta di carbonio.

I tre ricercatori tentarono di riprodurla in laboratorio, simulando le condizioni di quella particolare atmosfera stellare: un plasma di carbonio a 10.000 °C in atmosfera di elio; fu

così che la molecola di C_{60} venne sintetizzata per la prima volta.

Negli anni seguenti si scoprirono altre molecole con strutture poliedriche simili, di varie forme e dimensioni che vennero chiamate fullereni, dal nome dell'architetto Buckminster Fuller, famoso per le sue cupole esapentagonali che somigliano molto a queste nuove molecole.

L'importanza di questa scoperta fu subito riconosciuta dalla comunità scientifica internazionale e nel 1996 i tre ricercatori ottennero il premio Nobel per la chimica. Questa nuova struttura molecolare è profondamente diversa dalle altre due forme in cui si struttura generalmente il carbonio, la grafite e il diamante, ed è un ottimo sub-compo-

nente per realizzare nuovi materiali e dispositivi con tecniche bottom-up. Il C_{60} puro è un semiconduttore ma quando viene drogato con metalli alcalini diventa superconduttore con una temperatura di transizione tra le più alte raggiunte dai cristalli di questo tipo. Siccome resistività e superconduttività dipendono anche dalla quantità di materiali droganti che vengono immessi nel C_{60} , è possibile realizzare con questa molecola materiali con caratteristiche molto diverse tra loro.

Un particolare tipo di fullereni è il nanotubo di carbonio la cui struttura cilindrica può essere a parete singola (SWCNT, Single-Wall Carbon NanoTube) o multipla (NWCNT, Multi-Wall Carbon NanoTube); scoperto nel

ostacoli, scavalcandoli, annullandosi e sommandosi, a formare onde più grandi e intense, un po' come avviene con le onde del mare. Quando però un elettrone si trova confinato all'interno di un atomo, il suo comportamento si fa ancora più strano: la sua energia è

quantizzata, può cioè assumere solamente valori discreti e ben definiti, e l'elettrone può al massimo passare da un valore energetico all'altro. Se invece i singoli atomi si organizzano in strutture più complesse, per esempio formando un reticolo cristallino, l'energia degli elet-

troni non è più discreta, ma può variare con continuità all'interno di bande. Se però gli elettroni all'interno di queste strutture si trovano confinati in regioni di dimensioni nanometriche, il loro comportamento cambia, le bande si restringono e gli elettroni tendono a comportarsi di



1991 dal ricercatore della NEC Sumio Iijima, il nanotubo di carbonio ha un corpo formato da soli esagoni, mentre le strutture di chiusura sono formate da esagoni e pentagoni, esattamente come i fullereni. Il diametro di un nanotubo è compreso tra un minimo di 0,7 nm e un massimo di 10 nm.

L'elevatissimo rapporto tra lunghezza e diametro (nell'ordine di 10^4) consente di considerarli come delle nanostrutture virtualmente monodimensionali, e conferisce a queste molecole delle proprietà particolarmente interessanti. In campo elettronico i nanotubi di carbonio sono al centro di numerose ricerche volte alla fabbricazione di nuovi elementi attivi e sistemi di interconnessione, in particolare transistor a modulazione quantica.

Tra le applicazioni già sviluppate industrialmente citiamo le memorie Nano-Ram (Nram) che utilizzano celle di memoria costituite da nanotubi di carbonio che possono assumere due stati meccanici stabili, il primo corrispondente allo stato di riposo ed il secondo indotto da una precisa tensione e che risulta anch'esso stabile per l'effetto delle interazioni di Van der Waals. Un'altra recente applicazione riguarda i film trasparenti e conduttivi della Unidym rivestiti in nanotubi in carbonio per impieghi in schermi e monitor e i prototipo di celle solari che sfruttano questo materiale messi a punto dalla Cornell University.

nuovo come gli elettroni degli atomi singoli. Per esempio, in un sottile strato di silicio di pochi nanometri gli elettroni sono obbligati a rimanere all'interno dello strato di silicio e possono spostarsi solo lungo il piano dello stesso strato perché non hanno energia sufficiente per

saltare ad un livello energetico più alto posto al di fuori di questa regione. Lo strato di silicio viene chiamato *pozzo quantico* perché per gli elettroni che lo popolano, si comporta come un vero e proprio pozzo di energia. Analogamente nei *filii quantici*, gli elettroni sono confinati all'interno di una regione filiforme e possono scorrere solo nella direzione del suo asse. Se riduciamo ancora i gradi di libertà degli elettroni e li confiniamo in una regione di dimensioni nanometriche in tutte le direzioni, ecco che abbiamo realizzato un'altra nano struttura: il *punto quantico*.

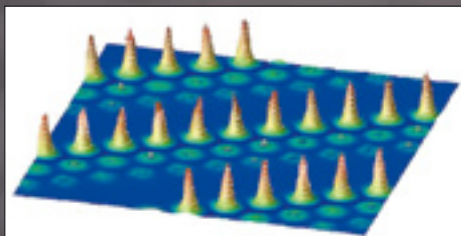
In tutti questi dispositivi gli elettroni hanno livelli energetici quantizzati, come accade agli elettroni degli atomi e delle molecole. Ovviamente i livelli energetici degli elettroni di un punto quantico sono diversi da quelli atomici, perciò i punti quantici vengono detti anche *atomi artificiali*.

Le nano strutture che si possono creare artificialmente sono sempre più numerose, diventa importante quindi predirne le proprietà elettriche e ottiche, nella speranza di poterle sfruttare per le applicazioni tecnologiche più interessanti. Attraverso la *modellistica* delle nano strutture è possibile prevedere alcune proprietà come, ad esempio, la corrente che scorre

in un nano transistor. In ultima analisi questi studi permettono di ottimizzarne le caratteristiche e quindi consentono di orientare l'approccio di fabbricazione nelle direzioni più promettenti dal punto di vista tecnologico. Le tecniche di fabbricazione delle nano strutture sono quanto mai varie: i *punti quantici* possono essere riprodotti con la litografia a fasci di elettroni oppure attraverso tecniche di auto assemblaggio elettrochimico; un *pozzo quantico* viene realizzato invece con deposizioni successive di strati atomici di materiali diversi, mentre filii quantici come i nanotubi si possono produrre con deposizione da vapori chimici. Anche i materiali magnetici possono essere utilizzati nel disegno di disposizioni nano strutturate ampliando notevolmente il campo delle possibili applicazioni tecnologiche; basti pensare alla domanda sempre crescente di memorie magnetiche ad alta densità e ad alta velocità di accesso. Questa richiesta ha portato a immaginare dispositivi sempre più piccoli in grado di sfruttare una delle proprietà quantistiche degli elettroni, il loro momento magnetico intrinseco (detto *spin*) che può assumere unicamente due valori: $+1/2$ o $-1/2$. Integrando dei nano elettrodi magnetici nella tecnologia dei semiconduttori sarà possibile scrivere su queste nano strutture magnetiche variando la direzione della magnetizzazione, e con essa il segno dello *spin* della maggioranza dei suoi elettroni. Lo sfruttamento delle proprietà intrinseche delle nano strutture investe numerosissimi campi; in queste pagine vogliamo concentrare la nostra attenzione sulle applicazioni nel campo dell'elettronica, dell'ottica e

della generazione di energia. Un settore in cui la miniaturizzazione dei dispositivi procede con successo già da alcuni decenni è quello dell'industria dei semiconduttori, il cui straordinario sviluppo ha portato negli anni ottanta a coniare la parola *microelettronica* per indicare le dimensioni tipiche dei componenti prodotti (il micrometro, appunto).

La maggior parte dei dispositivi elettronici oggi in produzione è già strutturata su scala submetrica, ma tra questi uno in particolare -il transistor MOS- è talmente importante e diffuso da essere considerato come il primo obiettivo e il banco di prova di ogni ulteriore miniaturizzazione.



Agli albori dell'elettronica le tecnologie planari permettevano di fabbricare pochi esemplari di transistor, ciascuno con dimensioni di alcuni micrometri; oggi, grazie al progresso dei processi fotolitografici, possiamo fabbricare a basso costo milioni di transistor su ogni singolo chip.

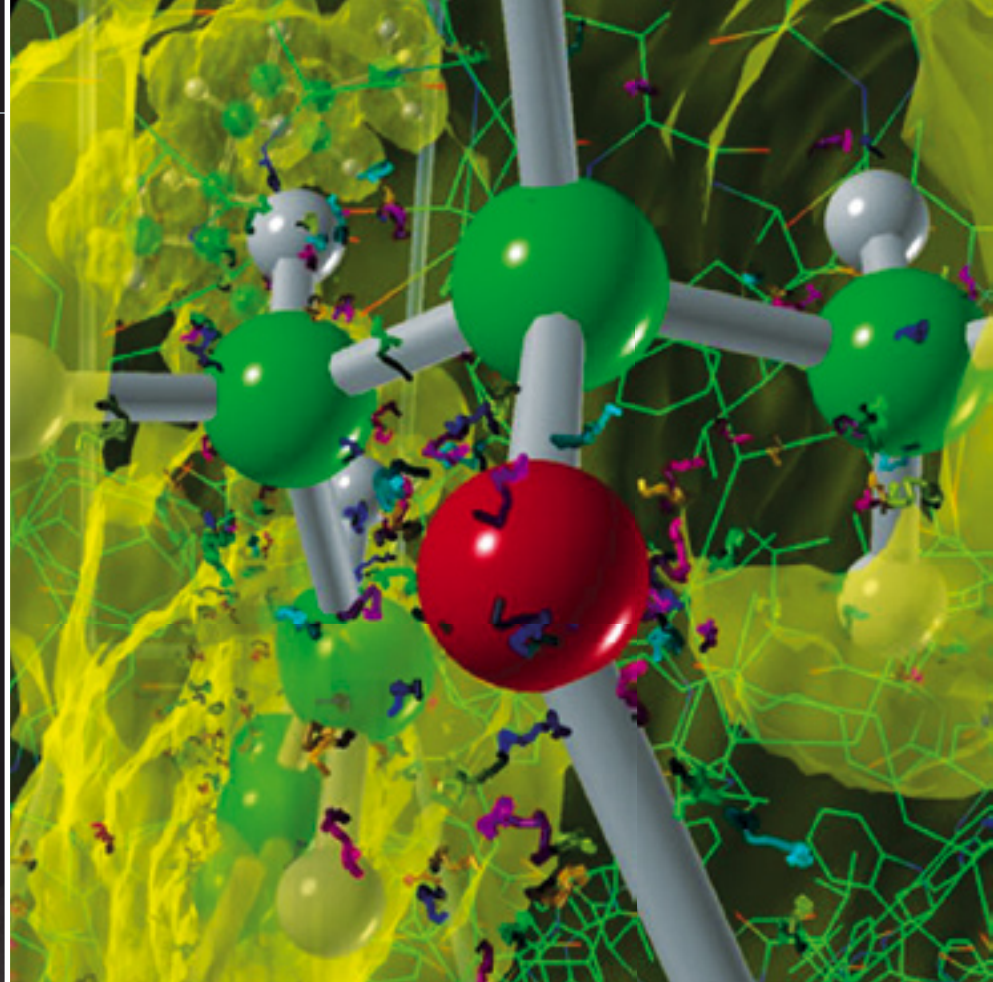
Cosa ancor più importante, un'accurata progettazione guidata da regole di *scaling*, ovvero di similitudine geometrica ed elettrostatica e l'utilizzo di nuovi materiali, hanno permesso di migliorare le prestazioni del transistor in termini di velocità e corrente riducendone nel contempo le dimensioni. I transistor MOS oggi in produzione hanno dimensioni di poche decine di nanometri, mentre prototipo di nano transistor con lunghezza di canale inferiore a 10 nm sono già stati realizzati in vari laboratori di ricerca. Alcune parti di questi transistor hanno già oggi dimensioni di circa un nanometro, corrispondenti ad una sequenza di appena una decina di atomi, e possono essere fabbricate contemporaneamente in miliardi di esemplari. Siamo dunque già scesi alla scala nanometrica, raggiungendo un livello di miniaturizzazione tale per cui le proprietà elettroniche degli atomi che compongono il dispositivo influenzano anche il comportamento del dispositivo stesso. Molte delle soluzioni individuate per ridurre ulteriormente le dimensioni dei transistor si basano su concetti derivati dallo studio sulle nanotecnologie. Unitamente ad una migliore e più efficiente progettazione dei circuiti, queste soluzioni consentiranno nei prossimi anni l'integrazione su vasta scala di miliardi di nano transistor, situazione che giusti-

ficherà sempre di più l'utilizzo della parola *nanoelettronica* al posto del termine *microelettronica*. In ogni caso, nei prossimi anni i circuiti integrati odierni che sono costruiti sulla superficie di una sola fetta di silicio monocristallino e usano soltanto uno strato sottilissimo di silicio, aumenteranno gradualmente il numero degli strati fino a arrivare a qualche migliaio. Come afferma in una famosa intervista Federico Faggin, l'inventore del microprocessore, "È come se oggi sapessimo fare soltanto fabbricati a un piano. Quando i fabbricati diventano troppo estesi bisogna imparare a costruirli con più di un piano. Alla fine impareremo a fare i grattacieli." Applicando queste nuove tecniche, i chip più complessi che realizzeremo tra qualche anno utilizzeranno 100.000 miliardi di transistor in dimensioni pari a quelle di un circuito integrato attuale. Al giorno d'oggi la produzione di chip è ancora basata sul silicio e sull'approccio top-down alla fabbricazione; avvalendosi della fotolitografia questo approccio consente di ottenere simultaneamente molti esemplari funzionanti dello stesso circuito, con caratteristiche riproducibili ed una considerevole riduzione del costo di ciascun esemplare. La tecnica della litografia nell'UV estremo può teoricamente raggiungere una risoluzione di 13 nm e dimensioni ancora più piccole sono ottenibili attraverso metodi di asportazione selettiva dei materiali. Visti i limiti della fotolitografia UV, e i costi ancora eccessivi della litografia a raggi X, sono allo studio nuove tecnologie di fabbricazione che si basano sull'approccio bottom-up. Un settore dell'elettronica ricco di innovazioni legate alle

nanotecnologie è quello delle memorie non volatili, cioè di quelle memorie che trattengono i dati anche in assenza di tensione di alimentazione. Ad esempio, si sta lavorando alla messa a punto di memorie in cui l'informazione binaria è costituita dalla presenza o assenza di pochissimi elettroni, al limite uno solo, all'interno di minuscoli cristalli di pochi nanometri di diametro immersi in uno strato isolante. Data l'esiguità della carica che rappresenta l'informazione, queste memorie potrebbero essere lette e scritte con un dispendio di energia minimo; inoltre l'integrazione nelle tecnologie del silicio di nuovi materiali a cambiamento di fase, ferroelettrici e ferromagnetici, permetterà di moltiplicare notevolmente la densità delle informazioni memorizzate.

Tra i dispositivi elettronici più interessanti che potrebbero essere realizzati con nuovi processi nano tecnologici vi sono quelli basati sui nanotubi di carbonio e sulle loro proprietà. L'impiego più naturale vista la loro elevata conducibilità e la capacità di condurre elevate densità di corrente senza fondere, sembra essere quello di utilizzarli come nanofili per connettere dispositivi realizzati con tecniche più convenzionali. Proprio recentemente è stata dimostrata la possibilità di indurre la crescita controllata di nanotubi di carbonio in punti specifici di un wafer di silicio. Inoltre i nanotubi, come gli altri fullereni e anche altri polimeri, drogati opportunamente come si fa con i semiconduttori, possono essere utilizzati come parte integrante di transistor e circuiti.

Un obiettivo ancora più ambizioso è quello di sviluppare l'elettronica molecolare. Già da

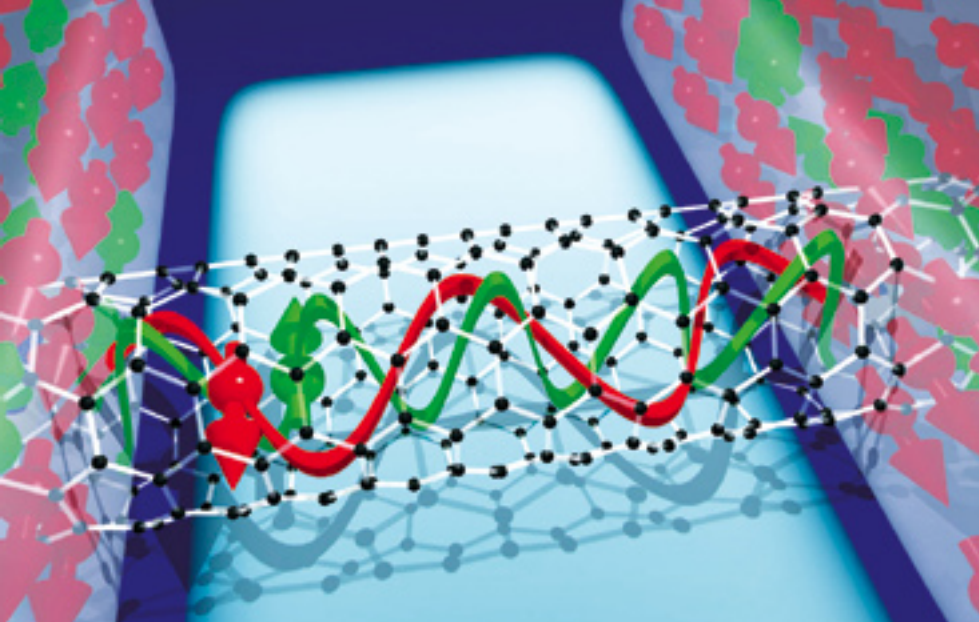


diversi anni è noto che alcune molecole possono compiere le funzioni elementari dell'elettronica, come la rettificazione di un segnale, l'immagazzinamento di dati e semplici funzioni di logica binaria.

Si tratterebbe quindi di realizzare dispositivi molecolari capaci di sostituire i transistor ad effetto di campo e le memorie attualmente utilizzate nell'elettronica al silicio. In effetti un simile obiettivo appare oggi troppo ambizioso, anche perché troppi problemi irrisolti separano i tentativi di laboratorio dallo sviluppo di una vera tecnologia competitiva con quella del silicio; tuttavia non è da escludere che in tempi ragionevolmente brevi specifiche molecole vengano utilizzate per realizzare le memorie o come parti di biosensori elettronici.

Poiché con l'elettronica molecolare ci siamo spinti fin quasi alle soglie della fantascienza, possiamo ancora far un passo

avanti ed accennare a quello che oggi sembra l'ultima rivoluzione possibile: il calcolo quantistico. In questo caso si tratta di sfruttare le leggi della meccanica quantistica all'interno di dispositivi nanostrutturati che non operino più in logica binaria, ma secondo una logica basata sulla sovrapposizione delle funzioni d'onda elettroniche del dispositivo. Il computer quantico invece di operare sui bit di informazione - l'uno e lo zero, il sì e il no, del computer classico - opera su *qubits*, bit di informazione che sono simultaneamente sia uno che zero, sia sì che no: un paradosso nella logica tradizionale. La possibilità teorica di creare un computer quantico è già stata dimostrata sperimentalmente, almeno per alcune operazioni elementari. Durante le operazioni di calcolo del computer quantico non è possibile sapere nulla su quanto avviene all'interno del computer, tranne che i *qubits* diventano



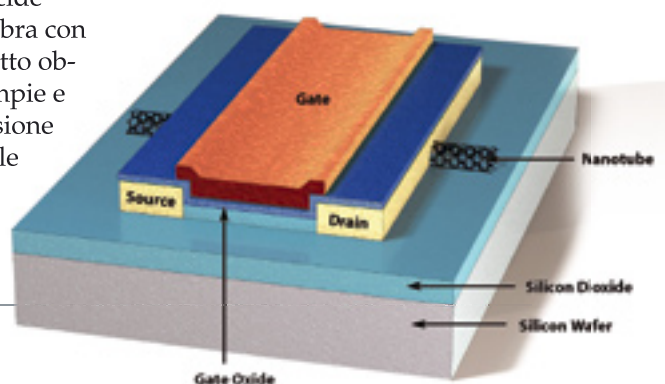
bit convenzionali quando è pronta la risposta finale. La ricerca per tale tecnologia è già iniziata e potrebbe anche fornire una strada alternativa per sostituire la tecnologia del silicio, quando quest'ultima arriverà alla fine della sua evoluzione.

I transistor non sono gli unici componenti nanoelettronici odierni. Altri dispositivi, come per esempio il laser a semiconduttore, contengono parti con dimensioni dell'ordine di pochi decine di nanometri e funzionano solo perché queste parti sono così piccole. Una direzione molto promettente, nella quale le nanotecnologie potranno contribuire a compiere ulteriori progressi nell'utilizzo industriale dei semiconduttori, sembra essere quella che conduce all'integrazione dei circuiti basati sui chip di silicio con nuovi tipi di dispositivi optoelettronici che utilizzano fasci di luce per il trasporto di dati anziché cariche elettriche. Questa idea non aveva avuto molto successo in passato, perché doveva essere implementata su semiconduttori composti come il fosforo di indio o l'arseniuro di gallio che sono poco utilizzati nell'elettronica di consumo. Recentemente, attraverso il drogaggio con terre rare si è riusciti a conferire proprietà ottiche simili a quelle dei semiconduttori del transistor MOS, rendendo quindi possibile l'integrazione delle fun-

zioni ottiche sul circuito integrato. La via ottica risulta conveniente per il trasporto dell'informazione su lunghe distanze poiché una delle sue caratteristiche principali è quella di trasportare il segnale senza dispersione per lunghi tratti. Le applicazioni della fotonica vanno dai laser e dai diodi emettitori di luce (LED) fino al cavo in fibra ottica con cui sono cablati i sistemi di comunicazione a rete fissa. In particolare la fibra ottica ha rivoluzionato la tecnologia delle comunicazioni grazie alla elevata velocità di trasmissione del segnale, che oggi è di 10 gigabit al secondo per singolo canale; sfruttando la possibilità di trasmettere luce a diverse lunghezze d'onda (luce di colori diversi) con gli attuali standard si possono trasmettere sulla stessa fibra fino a 200 segnali contemporaneamente (multiplexing). Una delle limitazioni della fibra ottica è data dal fatto che la luce percorre traiettorie rettilinee tra una riflessione e l'altra, e resta all'interno della fibra ottica (dove viaggia per riflessione sulle pareti interne) solo se il fascio incide sulla pareti interne della fibra con angoli piccoli; questo aspetto obbliga a curvature molto ampie e ad una immediata conversione del segnale ottico in segnale elettrico al termine della fibra. L'ideale sarebbe poter disporre di una

guida ottica che consenta alla luce che vi viaggia di compiere svolte (riflessioni) ad angolo retto senza essere assorbita; simili dispositivi sono allo studio e prendono il nome di *cristalli fotonici*.

Nel settore dell'immagazzinamento e della conversione dell'energia i materiali nano strutturati consentiranno un incremento delle prestazioni con riduzioni di costo significative. Per quanto riguarda le batterie al litio, oggi le più performanti, si stanno utilizzando materiali nano porosi che presentano una superficie maggiore dove immagazzinare il litio, e dove sia possibile sviluppare la reazione chimica in modo più efficiente. Lo sforzo della ricerca è indirizzato anche ad individuare materiali che possano rallentare e ridurre la passivazione degli elettrodi e allo sviluppo di materiali più sicuri per l'ambiente. Per quanto riguarda lo sfruttamento dell'energia solare con conversione diretta in elettricità, gli studi per forme di conversione più efficienti effettuati utilizzando materiali nano strutturati, organici e inorganici, hanno portato alla realizzazione di celle solari in certi casi più economiche ma per ora ancora meno efficienti del silicio cristallino. E' opinione diffusa tra i ricercatori, però, che la strada imboccata sia quella giusta e che con queste nuove tecniche nel giro di una decina d'anni si potranno ottenere a costi molto più contenuti, risultati migliori di quelli attuali. ■



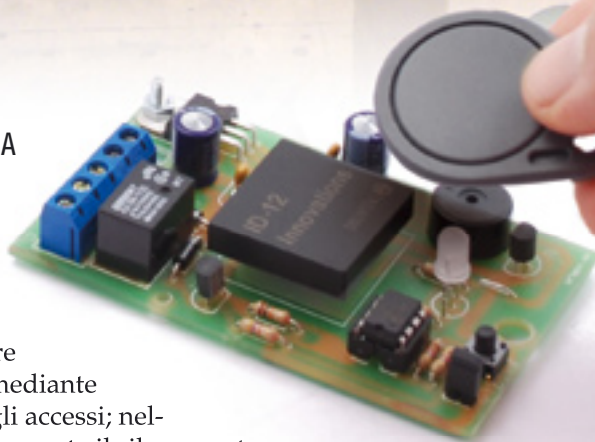
LETTORE DI TRANSPONDER STAND-ALONE

Dispone di un'uscita a relé che si attiva al riconoscimento di uno dei 127 tag RFid memorizzabili e permette di comandare elettroserrature o dispositivi di sicurezza.

di ALESSANDRO SOTTOCORNOLA

I tag per RFid vengono usati sempre più spesso in tutte quelle applicazioni dove occorre identificare un oggetto mediante un codice o controllare gli accessi; nell'ambito della logistica consente il rilevamento di articoli di varia natura che passano, ad esempio, su di un nastro trasportatore, mentre nei supermercati servono ad evitare i furti. Un grande ruolo lo giocano sicuramente nella regolamentazione degli

accessi di persone e automobili, nel rilevamento delle presenze dei lavoratori e nella realizzazione di chiavi di accesso a sistemi elettronici (ad esempio



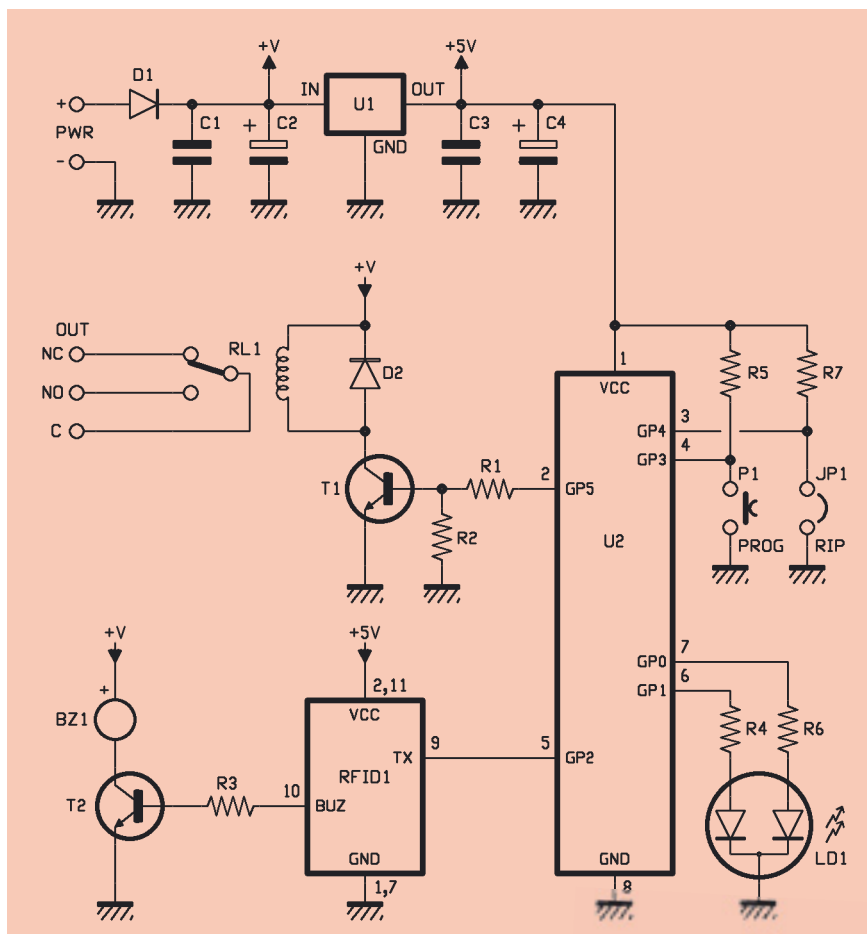
computer). Proprio per l'importanza che riveste l'RFid, nelle pagine di Elettronica In abbiamo dato ampio spazio a progetti di lettori di tag, proponendo vari schemi ognuno finalizzato a una specifica applicazione. Dopo il lettore low-cost descritto in luglio/agosto e quello con

di grande qualità, molto pratico perché incorpora tutto quel che serve a realizzare un lettore di transponder: l'ID-12 della Innovations Technology. Questo componente integra il generatore per il campo elettromagnetico eccitatore, la logica di lettura dell'assorbimento e di discrimina-

solo se questi sono codificati secondo tre standard: ASCII, Wiegand26 e Magnetic ABA Track2. Proprio perché provvede da solo al riconoscimento dei transponder, l'ID-12 è il dispositivo ideale per la realizzazione di molti sistemi, tra i quali varchi ad accesso elettronico, registratori di passaggi, classificatori e catalogatori di merci, ecc. Il tutto sollevando l'eventuale computer o microcontrollore che deve leggerne i dati dai compiti di decifrazione e scarto: il componente legge i transponder e scarta i dati di quelli fuori formato o letti male, passando solo le stringhe integre.

Degli 11 piedini di cui dispone, il 7 è quello che definisce il formato dei dati da riconoscere e va disposto a zero logico se si desidera che l'ID-12 legga transponder ASCII, collegato al piedino 11 per identificare dispositivi Magnet ABA Track2, ovvero alla linea di alimentazione per leggere transponder Wiegand26.

Il componente dispone di un'interfaccia seriale a livello TTL, accessibile dai piedini 9 (TX o D0) e 8, dalla quale escono i dati letti; c'è poi un'uscita (piedino 10) che pulsa rapidamente (alla frequenza di 3 kHz) ogni volta che il modulo rileva il codice di un tag codificato come previsto e che serve a pilotare un cicalino senza



elettronica o un diodo luminoso, dai quali ottenere la segnalazione di transponder identificato. Resta inteso che il modulo non è in grado di memorizzare codici, ma avvisa solamente del fatto che ha riconosciuto un tag codificato come si aspetta; poi, a decidere se il tag debba far attivare il circuito provvede il resto dell'elettronica, che andiamo subito a descrivere.

SCHEMA ELETTRICO

Il circuito è sostanzialmente una serratura elettronica ed è reso semplice dall'adozione dell'ID-12 della Innovations Technology, che nello schema vediamo interfacciato con un microcontrollore PIC12F683 Microchip (un micro ad 8 bit, ad architettura RISC, con memoria Flash) al quale passa i dati che legge dai tag e con un transistor NPN (T2) che usiamo come amplificatore di corrente necessario a pilotare un cicalino. L'ID-12, una volta alimentato è pronto all'uso; quando gli si avvicina un tag, fornisce in uscita i dati letti da esso.

Nel nostro caso il pin 7 è connesso a massa, quindi il circuito riconoscerà solo i tag codificati ASCII e restituirà in uscita proprio caratteri ASCII, che il microcontrollore verificherà e preparerà per la memorizzazione (in apprendimento) o la verifica con i codici già in memoria (fun-

zionamento da serratura).

Nel circuito, l'ID-12 è interfacciato con un microcontrollore che provvede all'apprendimento dei tag e al loro riconoscimento, quindi alle azioni correlate all'identificazione di uno dei transponder presenti in memoria. All'inizializzazione, il firmware del PIC imposta GP0 e GP1 come uscite dedicate al comando diretto dei diodi del led bicolore LD1, usato per segnalare le condizioni operative. Un'altra uscita è GP5, utilizzata per comandare, mediante il transistor T1 (polarizzato in base dal partitore R1/R2) il relé del circuito. Le linee GP2, GP3 e GP4 sono iniziate come ingressi e servono, alla lettura, rispettivamente, dei dati in arrivo dal modulo ID-12, dello stato del pulsante P1 (programmazione) e della condizione del ponticello JP1.

L'intero circuito funziona con i 5 volt continui e stabilizzati che il regolatore U1 ricava dalla tensione di alimentazione applicata ai punti + e - PWR, che può essere compresa tra 11 e 15 volt, in continua. Il diodo D1 protegge il circuito nel caso si applichi l'alimentazione con la polarità invertita, C1 e C2 filtrano la tensione d'ingresso rispettivamente dai disturbi impulsivi e dal residuo d'alternata (ripple); C3 e C4 filtrano i 5 volt.

La gestione dei tag

Durante la lettura, il modulo ID-12 restituisce il codice esadecimale in formato ASCII del tipo A1234B567C (10 byte) della tessera transponder. Di tale codice vengono presi in considerazione gli ultimi 4 caratteri, che vengono quindi memorizzati dal microcontrollore nella propria EEPROM; pertanto ogni codice occupa solo 2 byte. Quindi, considerando l'esempio sopra riportato, in una locazione di partenza viene memorizzato 0x56 e nella locazione successiva 0x7C. La parte precedente del codice viene ignorata in quanto solitamente è uguale per ogni transponder. Così facendo è ipoteticamente possibile trovare un tag con lo stesso codice non prima di 216 (65.536) tag, in quanto vengono presi in considerazione solo due byte del codice.



CARATTERISTICHE TECNICHE

- Fino a 127 tag memorizzabili.
- Possibilità di memorizzazione\cancellazione di un singolo tag e cancellazione totale della memoria.
- Organizzazione dinamica della memoria: ogni tag viene memorizzato nella prima locazione disponibile.
- Funzionamento dell'uscita monostabile (attivazione per 1 secondo) oppure bistabile.
- Buzzer segnalatore del passaggio di tag sul lettore.

COME FUNZIONA

Una volta messo il circuito sotto tensione, quando si avvicina un tag codificato ASCII al modulo lettore ID-12, il piedino 10 di quest'ultimo genera per un secondo un treno di impulsi a 3 kHz (che fa suonare il cicalino comandandolo mediante il T2) e il micro si predispone alla lettura dei dati che RFID1 gli passa lungo il canale dati TX; letti i dati, il microcontrollore estrae il codice



Arriva il passaporto Rfid

Anche in Italia è arrivato il passaporto elettronico basato su Rfid già collaudato in U.S.A. e in UE; le prime città ad averlo emesso sono state Grosseto e Potenza. Il nuovo passaporto incorpora un transponder al cui interno risiedono diverse informazioni fra cui la propria immagine e i dati biometrici; quindi per la prima volta verranno prese le impronte digitali a tutti coloro che richiederanno il nuovo passaporto, minori di 12 anni esclusi. La procedura è prevista da un regolamento europeo del 2004 secondo cui tutti i nuovi passaporti devono contenere in formato elettronico fotografia e impronte digitali del titolare; la maggior parte degli stati membri dell'Unione si è già adeguata. Il passaporto elettronico manterrà lo stesso aspetto di quello tradizionale; però incorporerà il tag Rfid operante in modalità protetta; così non ci sarà alcun rischio per i dati sensibili. Solo le questure e i commissariati saranno abilitati alla rilevazione delle impronte, mentre rimarranno invariate le competenze di comuni e Carabinieri per l'accettazione delle domande e dei relativi passaporti. I dati raccolti al momento della richiesta saranno trasferiti nei CED del Ministero degli Interni, dove resteranno memorizzati solo fino alla stampa del passaporto. Infatti il controllo al posto di frontiera sarà effettuato direttamente tra l'immagine dell'impronta immagazzinata nel microchip Rfid del documento e l'impronta del polpastrello del viaggiatore rilevata da un'apposito lettore.

in essi contenuto e lo confronta con quelli che ha in memoria. A questo punto possono accadere due cose: se il codice non è di quelli dei transponder appresi il micro attiva l'allarme, corrispondente al lampeggio del led bicolore a luce rossa; dunque, il relé rimane a riposo, però suona il cicalino e il led si accende di rosso per un secondo, poi torna nella condizione iniziale. Se il

circuito lavora nella modalità ad uscita bistabile e il relé è stato eccitato da un transponder valido, il diodo, acceso a luce, verde diventa rosso e poi torna verde. Se invece il transponder è uno di quelli già appresi, ciò che accade dipende dalla modalità di attivazione dell'uscita: in monostabile il micro fa pulsare RL1 per un secondo; questo relé servirà per attivare un'elettroserratura,

un apricancello, per liberare la sbarra di un tornello ecc. Nel contempo il led bicolore pulserà a luce verde, per poi spegnersi al rilascio del relé.

In modalità bistabile, invece, il relé cambia di stato ogni volta che all'ID-12 si avvicina un tag valido: la prima volta RL1 scatta e resta eccitato, condizione evidenziata dall'illuminazione a luce verde fissa del led bicolore. Notate che il firmware del PIC è stato progettato in modo che il circuito segnali comunque il passaggio di tag, anche se non sono tra quelli abbinati (memorizzati): infatti se all'ID-12 si avvicina un transponder non ASCII il led bicolore resta immutato, mentre se si avvicina un tag ASCII, la segnalazione è diversa a seconda che lo stesso sia abbinato (luce verde) o non abilitato (luce rossa). Chiaramente il circuito può funzionare così se nella sua memoria si trova almeno il codice di un tag Rfid; la memorizzazione del caso si conduce nel modo seguente: dopo aver alimentato la scheda, premere il pulsante P1 e mantenerlo premuto fino a che il led non si accende a luce arancione. A questo punto bisogna rilasciare il pulsante e passare la tessera transponder che si desidera memorizzare. Se il led si spegne e riaccende, la tessera è stata memorizzata; se invece il led rimane acceso, significa che il codice del transponder è già presente in memoria, oppure che lo spazio in memoria è esaurito. A questo punto è possibile replicare la procedura per abbinare altri tag. Invece, se desiderate uscire dalla procedura di memorizzazione, tenete premuto il pulsante P1 fino a che il led non inizia a lampeggiare; ciò segnerà il ritorno al normale funzionamento (chiave).

CANCELLAZIONE TESSERA

In ogni momento è possibile rimuovere un codice dalla memoria, ossia disabilitare il codice corrispondente a un tag per RFid; allo scopo bisogna alimentare la scheda, premere il pulsante P1 e mantenerlo premuto fino a che il led rosso non si accende. A questo punto bisogna rilasciare

di eventuali altri: basta passarli uno dopo l'altro. Per abbandonare la procedura, tenete premuto il pulsante P1 fino a che il led non inizia a lampeggiare, condizione che evidenzia il rientro nel normale funzionamento (chiave).

CANCELLAZIONE TOTALE

Per cancellare in un solo colpo

pratica il lettore di transponder, per il quale abbiamo previsto un circuito stampato che potete facilmente autocostruire (visto che la basetta è monofaccia) scaricando la traccia lato rame (da cui ricavare la pellicola per la fotoincisione) dal nostro sito web www.elettronica.in.it.

Incisa e forata la basetta, dispo-

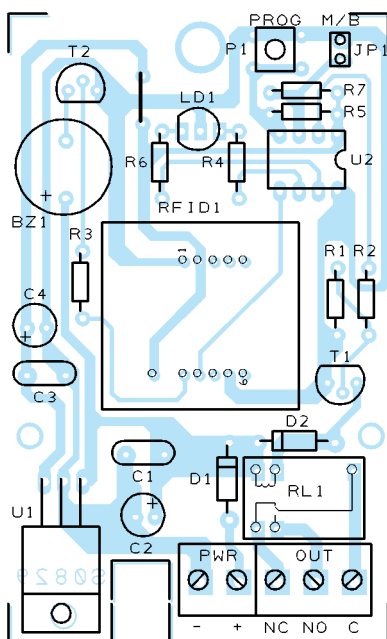
[piano di MONTAGGIO]

Elenco Componenti:

R1: 10 kohm
R2, R3, R5, R7: 4,7 kohm
R4, R6: 470 ohm
C1, C3: 100 nF multistrato
C2: 220 μ F 25 VL elettrolitico
C4: 220 μ F 16 VL elettrolitico
C5, C6: 15 pF ceramico
BZ1: Cicalino 12 V con elettronica
LD1: Led 5 mm rosso/verde
P1: Microswitch
RFID1: Modulo ID-12
RL1: Relé 12 V
T1, T2: BC547
U1: 7805
U2: PIC12F683 (MF829)

Varie:

- Morsettiera 2 poli passo 5,08 mm
- Morsettiera 3 poli passo 5,08 mm
- Zoccolo 4+4
- Vite 10 mm 3 MA
- Dado 3 MA
- Strip maschio 2 poli
- Jumper
- Contenitore plastico SC-700
- Circuito stampato



il pulsante e passare il tag che si desidera cancellare; se il led si spegne e riaccende, la tessera è stata cancellata, mentre se, invece, il led rimane acceso, significa che state tentando di rimuovere dalla memoria un transponder che non è mai stato memorizzato o che attualmente non risulta abbinato al lettore. Cancellato un tag, è possibile ripetere la procedura per rimuovere i codici

tutta la memoria di servizio, ossia i codici di tutti i tag abbinati, bisogna alimentare il circuito con premuto il pulsante P1. Quando il led rosso si accende, dovete rilasciare il pulsante perché la cancellazione è avvenuta completamente.

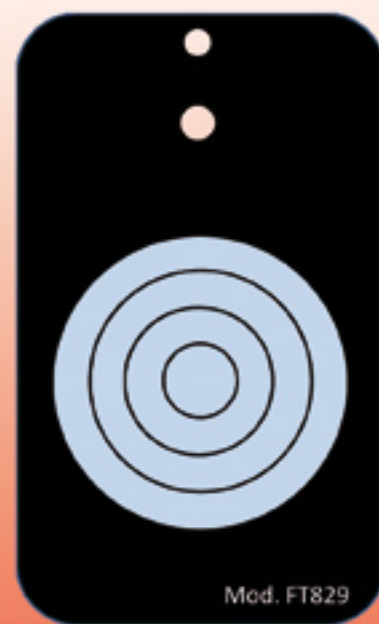
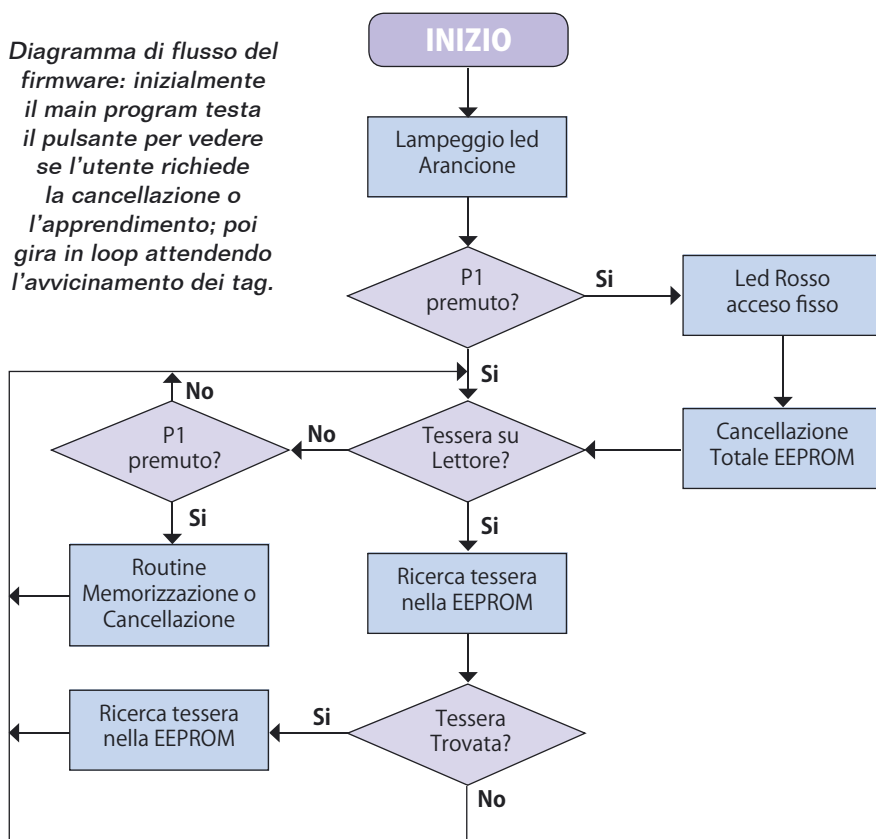
REALIZZAZIONE PRATICA

Bene, a questo punto possiamo vedere come si costruisce in

nete i componenti partendo dalle resistenze e dai diodi al silicio, quindi sistemate lo zoccolo per il microcontrollore; quanto al modulo ID-12, potete montarlo su uno zoccolo "arrangiato" ottenuto tagliando due strip di contatti da 5 pin l'una da uno zoccolo per integrati.

Proseguite il montaggio con i condensatori, dando la precedenza a quelli non polarizzati e

Diagramma di flusso del firmware: inizialmente il main program testa il pulsante per vedere se l'utente richiede la cancellazione o l'apprendimento; poi gira in loop attendendo l'avvicinamento dei tag.



Mascherina da applicare alla scatola che conterrà il circuito e che indicherà dove avvicinare i tag.

rispettandola polarità indicata per gli elettrolitici. Sistemate poi il pin-strip a due punte per realizzare il jumper JP1, il piccolo pulsante da c.s., i due transistor e i led bicolore, rammentando che lo smusso corrisponde al terminale di anodo della giunzione rossa; comunque basta tenere il diodo con la

tacca rivolta come indicato nel piano di montaggio. Procedete collocando il relé miniatura, il cicalino (essendo senza elettronica, questo non ha polarità) le morsettiere a passo 5 mm per le connessioni di alimentazione e dello scambio del relé, quindi sistemate il regolatore 7805, che potete montare adagiato e fissato

con un piccolo bullone 3MA più dado dopo averne piegato i terminali ad angolo retto.

Se desiderate tenere il modulo più sollevato dallo stampato, ad esempio perché dopo aver collocato la scheda sotto un pannello nell'uso vi accorgete che i tag vengono letti a fatica, potete prolungare i contatti dello zoccolo usando degli strip femmina a passo 2,54 mm dell'altezza desiderata.

Completato il tutto, verificate con il disegno di disposizione componenti illustrato in queste pagine che ogni componente sia al posto giusto e orientato correttamente; inserite gli integrati nei rispettivi zoccoli e l'unità è così completata. Per alimentarla, utilizzate un qualsiasi alimentatore in grado di fornire una tensione continua, possibilmente stabilizzata, del valore di 11÷15 V e una corrente di almeno 150 milliampere.



per il MATERIALE

Tutti i componenti utilizzati nel progetto descritto in queste pagine sono facilmente reperibili in commercio. Dal sito della rivista è possibile scaricare gli unici componenti non disponibili sul mercato, ovvero il master del circuito stampato ed il firmware col quale programmare il microcontrollore. Quest'ultimo è disponibile anche già programmato (cod. MF829) al prezzo di 12,00 Euro. Per quanto riguarda gli altri componenti, il modulo Rfid ID12 costa 33,00 Euro mentre i transponder, disponibili sia in versione carta di credito (TAG2) che portachiavi (TAG1), costano 3,00 Euro cadauno. 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>



Corso FPGA

Un primo progetto pratico da realizzare con il sistema di sviluppo per la serie SPARTAN-3AN Xilinx.

di
MARIANO SEVERI

Nella precedente puntata abbiamo visto in dettaglio l'architettura dei componenti FPGA della serie Spartan-3AN della Xilinx ed un interessante sistema di sviluppo che può essere utilizzato per imparare a progettare con i dispositivi logici programmabili di nuova generazione. Abbiamo anche provato ad eseguire alcuni degli esempi di riferimento che vengono forniti a corredo, acquistando così un po' di dimestichezza con questi nuovi dispositivi. Ora è giunto il momento di imparare a realizzare un semplice progetto partendo da zero.

Proveremo ad impiegare il nostro sistema di sviluppo come cronometro; utilizzeremo il display a caratteri per visualizzare il tempo ed i pulsanti per controllare il cronometro.

Purtroppo il display è piuttosto lento nell'aggiornare le informazioni, per cui nella misura del tempo abbiamo scelto come risoluzione 1 secondo; trattandosi di un semplice esempio dimostrativo per iniziare ad utilizzare i dispositivi FPGA, ciò è più che sufficiente. Tutti i file inerenti a questo progetto sono disponibili per il download.

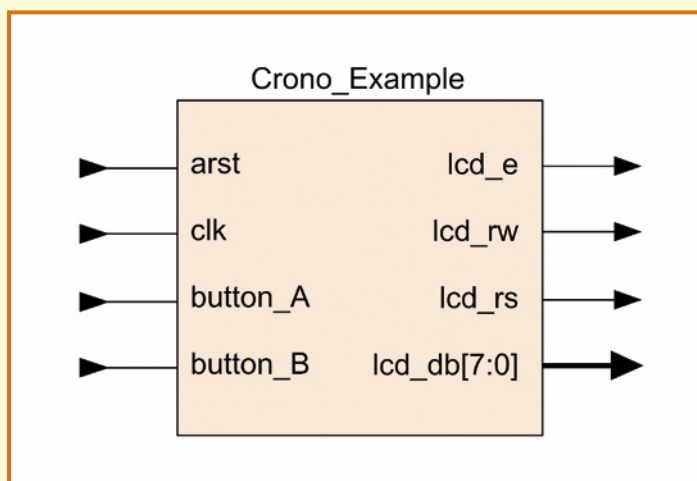


Fig. 1 - Un simbolo di alto livello per il nostro cronometro.

bile sulla scheda di sviluppo ha una frequenza di 50 MHz. I segnali **time_incr** e **time_rst** abilitano il conteggio ed azzerano, rispettivamente, il contatore. Sono generati dal modulo **CronoCtrl**, che implementa una semplice macchina a stati la cui evoluzione è dettata dagli ingressi **button_A** e **button_B**, come sarà descritto in seguito. **Debouncer** è un classico circuito di debouncing (anti-rimbalzo) che consente di rilevare la pressione ed il rilascio del pulsante;

SCHEMA A BLOCCHI DEL CIRCUITO

Cominciamo col rappresentare il nostro progetto mediante il simbolo mostrato nella Fig. 1, che riporta i segnali di ingresso ed uscita. Il nostro esempio ha come ingressi il segnale di clock, di reset asincrono ed i pulsanti per gestire il cronometro; come uscite sono previsti invece i segnali di controllo/dati verso il display LCD.

Disegniamo poi un semplice schema a blocchi dell'architettura interna del dispositivo, come vedete nella Fig. 2.

In esso, **TimeCounter** è il contatore di ore, minuti e secondi, caratterizzato da una risoluzione di 1 secondo, dettata dal segnale di sincronismo **clk_ce** generato mediante un semplice prescaler, indicato in figura come **TimePrescaler**; il segnale di clock disponi-

sono usati due moduli identici per gestire i pulsanti per il controllo del nostro cronometro. In corrispondenza di una variazione del tempo misurato, **TimeUpdate** genera il segnale che comanda l'aggiornamento del tempo sul display a caratteri. **LCDInterface** è l'interfaccia di controllo di questa periferica.

DESCRIZIONE IN VHDL

Il VHDL è un linguaggio utilizzato per descrivere i circuiti logici; qui lo utilizziamo per definire le funzioni del nostro progetto. Elementi fondamentali per una descrizione in VHDL sono le entità e le architetture. L'entità corrisponde al simbolo di più alto livello del nostro circuito e contiene la descrizione delle interfacce. Facendo riferimento al simbolo illustrato nella Fig. 1, l'entità nel caso del nostro contatore appare come riportato nel Listato 1.

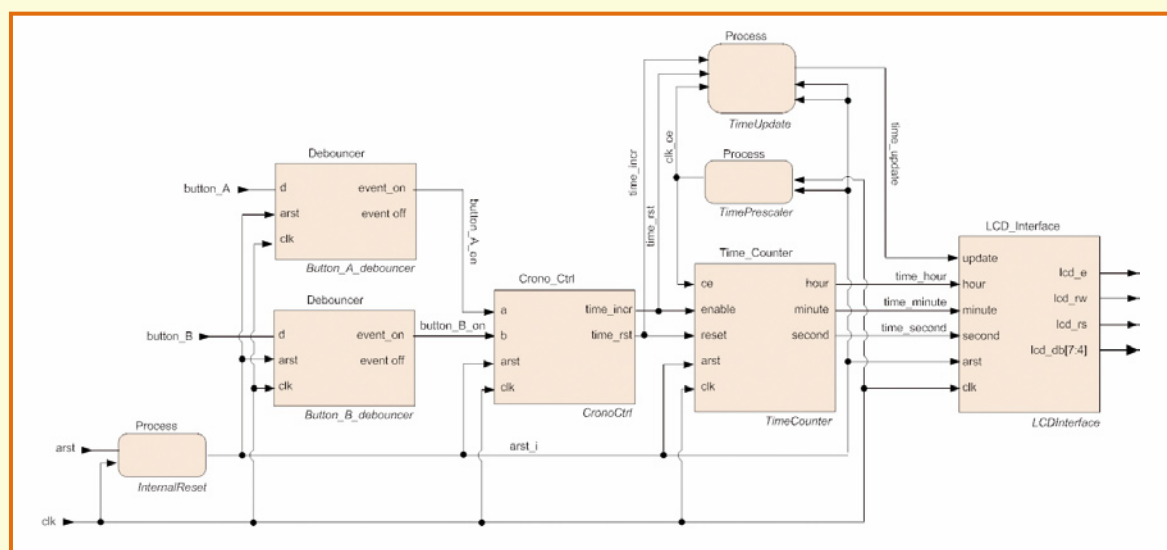


Fig. 2 - Una possibile realizzazione del cronometro.

Listato 1

```

entity crono_example is
port
(
-- clock and reset -----
arst      : IN std_logic;
clk       : IN std_logic;
-- control input -----
button_A  : IN std_logic;
button_B  : IN std_logic;
-- LCD interface -----
lcd_e     : OUT std_logic;
lcd_rw    : OUT std_logic;
lcd_rs    : OUT std_logic;
lcd_db    : OUT std_logic_vector(7 downto 0)
);
end entity crono_example;

```

Entity è la parola chiave del linguaggio che definisce l'entità e *crono_example* il nome che abbiamo dato al nostro progetto. L'istruzione **port** definisce l'insieme dei segnali di ingresso/uscita. Per ogni segnale sono specificate la direzione (ingresso od uscita) e il tipo. Il VHDL, in effetti, è un genere di linguaggio fortemente tipizzato; senza voler entrare nei dettagli, basti per ora osservare che *std_logic* è il tipo che serve per descrivere i segnali elettrici e può assumere, tra gli altri, i valori 0, 1 e Z per indicare i livelli logici, rispettivamente, basso, alto e tri-state. Il tipo *std_logic_vector* è un array di *std_logic*. Questi tipi non sono nativi del VHDL ma vengono dichiarati all'interno del package *std_logic_1164* della libreria *ieee*. Affinché il compilatore riconosca correttamente questi tipi, dobbiamo pertanto far precedere la dichiarazione dell'entità dalla definizione delle librerie e dei package che intendiamo collegare al nostro progetto. Ecco le istruzioni da aggiungere in testa al nostro file:

```

library ieee;
use ieee.std_logic_1164.all;

```

Se l'entità è la definizione delle interfacce di un modulo, l'architettura ne è l'implementazione e quindi, riprendendo la corrispondenza già evidenziata in precedenza, equivale allo schema a blocchi del nostro circuito. Nel **Listato 2** è riportata la descrizione dell'architettura del nostro cronometro.

In esso, *Architecture* è la parola chiave che definisce l'architettura, *struct* il nome che abbiamo dato alla corrispondente implementazione del nostro cronometro. La prima sezione dell'architettura fino all'istruzione *begin* è quella dichiarativa e contiene, nel nostro caso, la definizione dei segnali interni, ovvero quelli che

consentiranno di collegare i diversi moduli. La successiva sezione, che potremo chiamare assertiva, descrive invece quali moduli sono presenti all'interno del sistema e quindi come questi sono connessi. Per ognuno dei moduli usati, evidentemente, dovremmo aver già definito entità ed architettura da qualche altra parte; in seguito vedremo i dettagli. Per istanziare un componente all'interno di un'architettura si usa una sintassi del tipo:

```

InstanceName : entity work.EntityName
port map
(
    port_name_0 => signal_name_0, ...
);

```

dove *InstanceName* è il nome dell'istanza corrente, *EntityName* il nome dell'entità corrispondente (ovvero del modulo nella sua accezione astratta), *port_name* quello della porta dell'entità e *signal_name* quello del segnale ad essa connesso. All'interno di un sistema possono evidentemente essere presenti due istanze di uno stesso componente. Nel nostro caso, ad esempio, sono istanziati due circuiti di anti-rimbalzo per la gestione dei pulsanti; è sufficiente che le due istanze all'interno dell'architettura abbiano nomi diversi, pur facendo riferimento alla stessa entità. Alla luce di questi pochi elementi, se riguardiamo la descrizione dell'architettura del nostro cronometro e lo schema a blocchi riportato nella **Fig. 2**, la maggior parte del codice che abbiamo riportato nel **Listato 2** per descrivere l'architettura del nostro cronometro ora dovrebbe risultarci piuttosto chiara.

PROCESSI IN VHDL E CIRCUITI COMBINATORI E SEQUENZIALI

Oltre a istanziare dei componenti, all'interno della nostra architettura sono anche descritte esplicitamente delle funzioni logiche, come ad esempio quelle per generare il segnale di *TimeUpdate*; altre saranno presenti nelle descrizioni dei moduli che compongono il nostro cronometro, come ad esempio la macchina a stati o la logica di interfaccia del display a caratteri. Vediamo come queste possono essere rappresentate.

In base alla descrizione che abbiamo dato

Listato 2

```

=====
architecture struct of crono_example is
=====
-- internal signal
-----
signal arst_i      : std_logic;
signal clk_ce      : std_logic;
signal time_incr   : std_logic;
signal time_rst    : std_logic;
signal time_hour   : std_logic_vector(7 downto 0);
signal time_minute : std_logic_vector(7 downto 0);
signal time_second : std_logic_vector(7 downto 0);
signal time_update  : std_logic;

signal button_A_on : std_logic_vector(1 downto 0);
signal button_B_on : std_logic_vector(1 downto 0);
-----

begin
-- Internal Reset
-----
InternalReset : process (arst, clk)
    variable v : std_logic_vector(3 downto 0);
begin
    if arst='1' then
        v:=x"f";
    elsif clk'event and clk='1' then
        v(3):=v(2); v(2):=v(1); v(1):=v(0); v(0):='0';
    end if;
    arst_i <= v(3);
end process InternalReset;
-----

-- Controller
-----
ButtonA_Debouncer : entity work.debounce
port map
(
    arst => arst_i, clk => clk,
    d => button_A, event_on => button_A_On, event_off => open
);

ButtonB_Debouncer : entity work.debounce
port map
(
    arst => arst_i, clk => clk,
    d => button_B, event_on => button_B_On, event_off => open
);

CronoCtrl : entity work.crono_ctrl
port map
(
    arst => arst_i, clk => clk,
    A => button_A_On, B => button_B_On,
    time_incr => time_incr, time_rst => time_rst
);
-----

-- Time Prescaler
-----
TimePrescaler : process (arst_i, clk)
    variable cnt_v : integer;
begin
    if arst_i='1' then
        cnt_v :=0;
        clk_ce <= '0';
    elsif clk'event and clk='1' then
        if cnt_v=4999999 then -- count 1 second
            clk_ce<='1'; cnt_v:=0;
        else
            clk_ce<='0'; cnt_v:=cnt_v+1;
        end if;
    end if;
end process TimePrescaler;
-----

-- Time Counter
-----
TimeCounter : entity work.time_counter
port map
(
    arst => arst_i, clk => clk,
    ce => clk_ce, en => time_incr, reset => time_rst,
    hour => time_hour, minute => time_minute, second => time_se-
cond
);
-----

-- LCD Interface
-----
TimeUpdate : process (arst, clk)
begin
    if arst='1' then
        Time_Update <= '0';
    elsif clk'event and clk='1' then
        if (time_incr='1' and clk_ce='1') then
            Time_Update <= '1';
        elsif time_rst='1' then
            Time_Update <= '1';
        else
            Time_Update <= '0';
        end if;
    end if;
end process TimeUpdate;

LCDInterface : entity work.LCD_Interface
port map
(
    arst => arst_i, clk => clk,
    hour => time_hour, minute => time_minute, second => time_se-
cond,
    update => time_update,
    lcd_e => lcd_e, lcd_rw => lcd_rw, lcd_rs => lcd_rs, lcd_db =>
lcd_db(7 downto 4)
);

    lcd_db(3 downto 0) <= x"f";
end architecture struct;
=====

```

inizialmente per il comportamento del nostro circuito, vogliamo che il segnale *TimeUpdate* sia attivo per un impulso di clock in seguito ad un'operazione di incremento (che avviene quando *time_incr*='1' e *clk_ce*='1') o di reset del contatore (corrispondente all'evento *time_rst*='1'). Come mostrato nella Fig. 3, quindi, il segnale può essere generato mediante un semplice flip-flop di tipo D il cui ingresso è dato da una funzione booleana dei segnali *time_incr*, *clk_ce*, e *time_rst*.

La sintassi per descrivere un flip-flop in lin-

guaggio VHDL è la seguente:

```

flip-flop : process (arst, clk)
begin
    if arst='1' then
        q<='0';
    elsif clk'event and clk='1' then
        q <= d;
    end if;
end process flip-flop;

```

dove *arst* è il segnale di reset asincrono, *clk* il

clock e d e q , rispettivamente ingresso e uscita. Come si vede, abbiamo usato una nuova istruzione *process* che serve a definire i processi. Un processo non è altro che un contenitore per una serie di istruzioni che il simulatore esegue sequenzialmente; all'interno di un'architettura sono eseguiti diversi processi in parallelo, il che consente di rappresentare strutture concorrenti. In seguito avremo modo di vederne molti. La sintassi più semplice per definire un processo generico è del tipo:

```
process_name : process (sensitivity_list)
begin
-- sequential instruction
end process process_name;
```

dove *sensitivity_list* è l'insieme dei segnali in corrispondenza delle cui variazioni il processo viene eseguito.

Se ora torniamo alla descrizione del segnale *TimeUpdate*, in base a quanto appena detto, ci basterà scrivere la seguente parte di codice per avere il comportamento atteso:

```
TimeUpdateReg : process (arst, clk)
begin
if arst='1' then
Time_Update <= '0';
elsif clk'event and clk='1' then
Time_Update <= (Time_Incr and Clk_ce) or Time_Rst;
end if;
end process TimeUpdateReg;
```

Qui è il caso di osservare che l'istruzione $q \leq d$ (che assegna l'ingresso all'uscita) può in genere essere sostituita da una serie di istruzioni sequenziali più complesse che contengono, ad esempio, costrutti *if...then...else*. Ciò consente di descrivere i circuiti combinatori che generano l'ingresso del nostro registro senza ricorrere direttamente alla espressione booleana, ma lasciando al tool di sintesi il compito di estrarla dalla nostra descrizione. Nel caso, ad esempio, del segnale *TimeUpdate* che stiamo considerando ed in accordo con la funzione logica che vogliamo realizzare, potremmo usare un costrutto del tipo:

```
if (time_incr='1' and clk_ce='1') then
Time_Update <= '1';
```

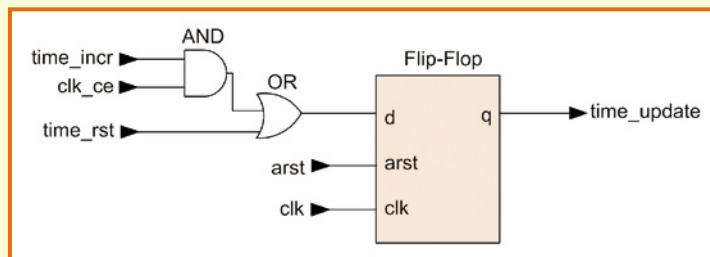


Fig. 3 - Generazione del segnale *TimeUpdate*.

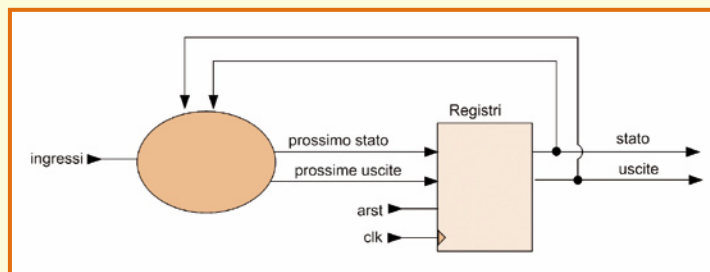


Fig. 4 - Schema generico di una macchina a stati.

```
elsif time_rst='1' then
Time_Update <= '1';
else
Time_Update <= '0';
end if;
```

DESCRIZIONE DI UNA MACCHINA A STATI

Una volta visto come istanziare dei moduli, in che modo connetterli e come creare dei processi per generare logica sequenziale e combinatoria, forse l'ultimo tassello che ci manca per avere in mano gli strumenti di base necessari a descrivere in VHDL un circuito elettrico è il modo per rappresentare una macchina a stati. Per colmare questa lacuna prenderemo ad esempio la descrizione della macchina a stati contenuta in *CronoCtrl*, che controlla il comportamento del nostro cronometro.

Prima di esaminarla in dettaglio, vediamo in generale come si può realizzare in hardware una macchina a stati. La Fig. 4 ne mostra un esempio. La logica combinatoria descrive le equazioni di prossimo stato e prossima uscita, le quali determinano appunto il prossimo valore delle uscite e degli stati in funzione dello stato corrente e del valore attuale dei segnali di ingresso e di uscita. Quindi sono istanziati dei registri che rappresentano l'elemento di ritardo che consente di sincronizzare nella realtà il funzionamento della macchina a stati.

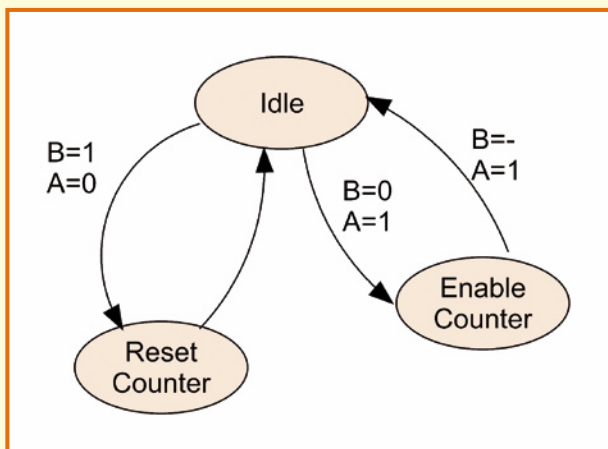


Fig. 5 - Diagramma a bolle di CronoCtrl.

Nella descrizione VHDL avremo quindi due processi: uno puramente combinatorio che rappresenta l'equazione di prossimo stato ed uno sequenziale che, invece, inferisce i registri; quest'ultimo abbiamo già visto come descriverlo. Nel nostro esempio il codice conterrà quindi un processo del tipo:

```

SMSeq : process (arst, clk)
begin
  if arst='1' then
    state_r <= idle;
    time_incr <= '0';
    time_rst <= '0';
  elsif clk'event and clk='1' then
    state_r <= state_c;
    time_incr <= time_incr_c;
    time_rst <= time_rst_c;
  end if;
end process SMSeq;

```

dove *state_r* è l'attuale stato della macchina (*state_c* il prossimo stato), *time_incr* e *time_rst* le due uscite (*time_incr_c* e *time_rst_c* le prossime uscite).

Per descrivere uno stato generico avremmo potuto usare un vettore di *std_logic*, definendo quindi esplicitamente, per ogni stato, la codifica. Il realtà il VHDL consente di definire dei tipi utenti tra cui quelli enumerativi che in questo caso trovano interessante applicazione. L'istruzione è la seguente:

```
type enum_type is (enum_list);
```

dove *enum_type* è il nome del tipo ed *enum_list* la lista dei valori che può assumere (separati

mediante virgola). Durante la fase di sintesi, sarà il compilatore a trovare la migliore realizzazione di questo tipo in termini di bit e la migliore codifica per i diversi valori, consentendo così al progettista di risparmiare tempo. Una delle proprietà principali del linguaggio di descrizione hardware, in effetti, è proprio la possibilità di portare la rappresentazione ad un livello più alto di astrazione che semplifichi il processo di progetto e consenta il riutilizzo di applicazioni già sviluppate.

Listato 3

```

SMComb : process (state_r, A, B)
begin
  case state_r is

    when idle =>
      if B='1' then
        state_c <= reset_counter;
      elsif A='1' then
        state_c <= enable_counter;
      else
        state_c <= idle;
      end if;
      time_incr_c <= '0';
      time_rst_c <= '0';

    when enable_counter =>
      if A='1' then
        state_c <= idle;
      else
        state_c <= enable_counter;
      end if;
      time_incr_c <= '1';
      time_rst_c <= '0';

    when reset_counter =>
      state_c <= idle;
      time_rst_c <= '1';
      time_incr_c <= '0';

  end case;
end process SMComb;

```

Torniamo alla nostra macchina a stati; nella Fig. 5 ne è mostrato lo "schema a bolle". Seguendo quanto anticipato in precedenza, definiamo il tipo per descrivere gli stati e quindi i due segnali per il prossimo stato e lo stato corrente :

```

type State is (idle, enable_counter, reset_counter);
signal state_c : state;
signal state_r : state;

```

Infine scriviamo un processo per le equazioni di prossimo stato e prossima uscita usando

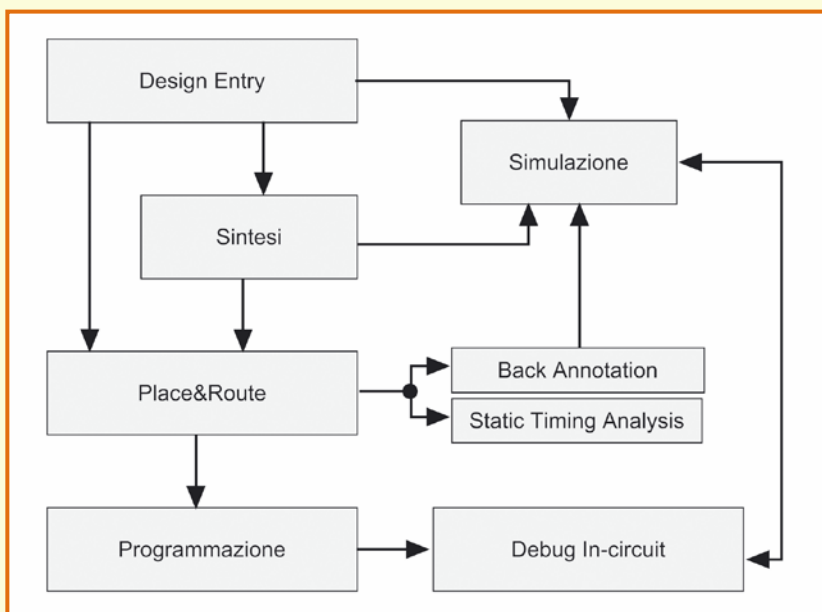


Fig. 6 - Il flusso di progetto con FPGA.

Graphics. Una versione free (http://www.xilinx.com/ise/optional_prod/mxe.htm) con alcune limitazioni relative al numero di linee di codice eseguibile simulabili ed al tipo di dispositivo supportato è già inclusa nell'ambiente di sviluppo per FPGA Xilinx ISE WebPack che descriveremo in seguito.

In alternativa, in Internet si possono trovare dei simulatori open-source tra cui GHDL (<http://ghdl.free.fr/>)

L'istruzione *case...when* come mostrato nel **Listato 3**. Come si vede, viene dapprima definito (*case state_r is*) il segnale da valutare per selezionare le istruzioni successive da eseguire; quindi per ogni possibile valore del segnale (*when ... =>*) viene definito l'insieme di queste. L'istruzione ritorna (*end case;*) non appena una delle condizioni viene eseguita. Per avere chiara la corrispondenza tra quanto appena spiegato e la descrizione hardware, tenete sott'occhio il **Listato 3** e il diagramma a bolle visibile nella Fig. 5.

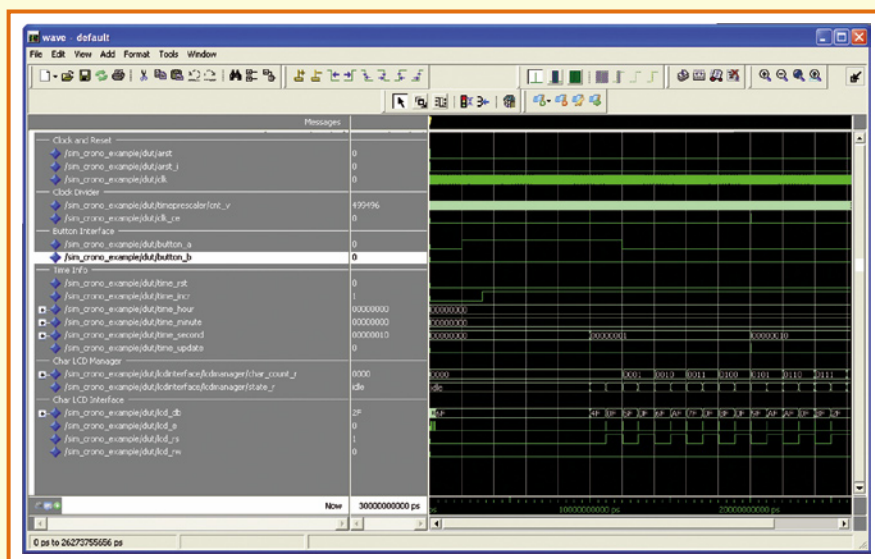


Fig. 7 - Simulazione del nostro progetto in ModelSim.

IMPLEMENTIAMO IL NOSTRO CIRCUITO

Nella Fig. 6 è riportato come promemoria lo schema del flusso di progetto che viene seguito nella realizzazione di FPGA e che avevamo descritto nella prima puntata del nostro tutorial.

SIMULAZIONE

Dopo aver scritto il codice sorgente, dobbiamo evidentemente simularlo per controllare che risponda al comportamento atteso; allo scopo esistono diversi ambienti, tra i quali uno dei più diffusi è ModelSim di Mentor

basato sulla catena di compilazione GNU. Detto ciò, precisiamo che simulare un progetto vuol dire in pratica generare gli ingressi e verificare che le uscite seguano il comportamento atteso. Nel nostro caso significa generare i segnali di clock e reset per il contatore ed i controlli provenienti dai pulsanti. Il test-bench di simulazione che abbiamo creato (si veda il file **sim_crono_example.vhd**) definisce un modulo **sim_crono_example** che non ha porte di ingresso ed uscita ma è semplicemente un contenitore per il nostro progetto da simulare e per i processi che servono a definire l'andamento dei segnali di ingresso. Di seguito

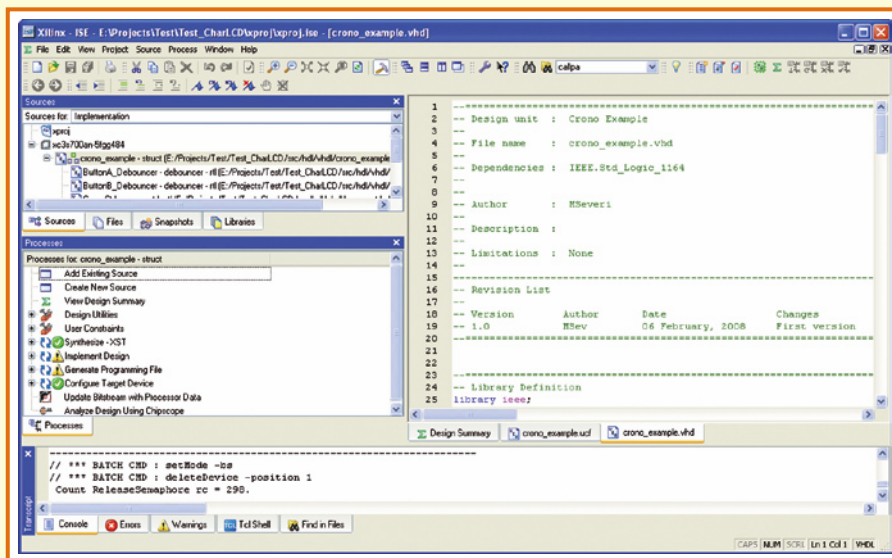


Fig. 8 - L'ambiente di sviluppo ISE Foundation.

è riportato come esempio il processo per la descrizione del segnale di clock :

```
ClkProcess : process
begin
  clk <= '0'; wait for 10 ns;
  clk <= '1'; wait for 10 ns;
end process ClkProcess;
```

Si noti che il processo non dispone di `sensitivity_list`; questo significa che viene rieseguito automaticamente al termine della lista di istruzioni contenute. L'istruzione `wait for ... ns`, in particolare, è una semplice istruzione di attesa per il tempo specificato. Mediante questi semplici costrutti si riesce così a creare una forma d'onda con periodo 20 ns (frequenza 50 MHz) corrispondente al segnale di clock del nostro kit di sviluppo.

Nella Fig. 7 è mostrato il viewer di forme d'onda di ModelSim al termine della simulazione; come si vede, in corrispondenza della pressione del pulsante A, la linea si attiva. La macchina a stati manda a livello alto la linea `time_incr` ed in corrispondenza dell'impulso di conteggio `clk_ce` il valore del contatore si incrementa. Vengono generati l'impulso `time_update` che richiede l'aggiornamento del display e, in seguito a questo, gli opportuni segnali di controllo per la periferica. Si noti che, per ridurre i tempi di simulazione, alcuni dei parametri del progetto (come ad esempio di tempi per la procedura di inizializzazione

del display a carattere od il fattore di divisione del clock di sistema) sono stati ridotti.

SINTESI E PLACE&ROUTE

La fase successiva è quella di sintesi e place&route in cui la nostra descrizione in linguaggio VHDL viene tradotta automaticamente in porte logiche, che saranno associate alle risorse disponibili nel dispositivo selezionato; qui verranno definite le interconnessioni ed infine sarà generato il file di programmazione per l'FPGA. ISE Design Suite è il tool proprietario di Xilinx che

consente di gestire l'intero flusso. Nel kit di sviluppo è compreso il CD di installazione del software; la versione più aggiornata può essere scaricata direttamente dal sito Xilinx all'indirizzo: http://www.xilinx.com/ise/logic_design_prod/webpack.htm.

Tale versione WebPack è gratuita, ma presenta alcune limitazioni (in termini di funzionalità e tipi di dispositivi supportati) che tuttavia per la maggior parte dei progetti e soprattutto per il nostro caso, non hanno interesse. In alternativa può essere scaricata in valutazione – interamente funzionante per un periodo di 60 giorni – la versione completa a partire dall'indirizzo: http://www.xilinx.com/ise_eval/index.htm.

La Fig. 8 mostra come appare l'ambiente di sviluppo. La finestra in alto a sinistra riporta i file inclusi nel progetto, mostrando la gerarchia di quest'ultimo; la finestra immediatamente sottostante illustra, invece, i processi che possono essere eseguiti. In basso è riportata una finestra di dialogo con indicazioni di messaggi ed errori oltre ad una shell Tcl; nel riquadro a destra un'area per finestre accessorie come ad esempio un editor di testo integrato.

L'utilizzo del tool non è molto diverso, almeno nelle linee di principio, dalla maggior parte degli ambienti di sviluppo hardware o software.

Dando per scontato che abbiate dimestichezza con uno di essi, di seguito riportiamo solo i

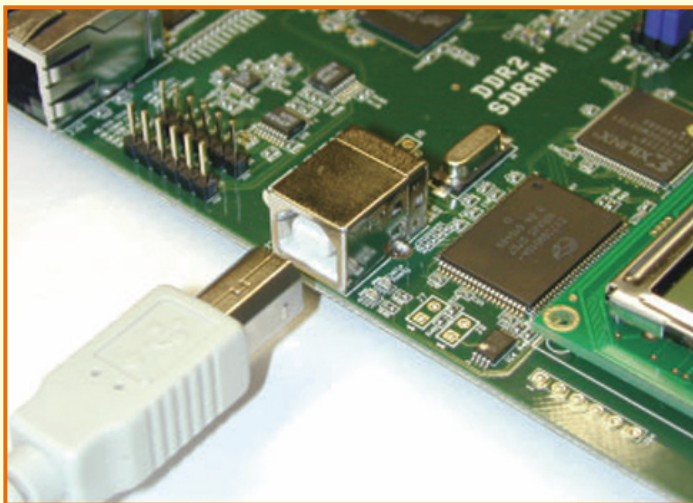


Fig. 9 - Connessione del kit di sviluppo al PC per la programmazione della FPGA.

punti essenziali per arrivare in pochi passi alla generazione del file di configurazione della FPGA.

Iniziamo creando un nuovo progetto: dal menu *File* selezionate *New Project*; quindi nella finestra pop-up che appare, definite la locazione del progetto sul disco dove desiderate crearlo ed il nome che volete dargli.

La finestra successiva consente di selezionare il tipo di dispositivo da utilizzare; nel nostro caso definiamo i parametri corrispondenti alla FPGA del kit di sviluppo:

Family -> Spartan3A and Spartan3AN
Device -> XC3S700AN

Proseguendo, è possibile creare nuovi file da aggiungere al progetto o inserirne di esistenti; nel nostro caso salteremo la prima aggiungendo direttamente i file per la descrizione del contatore che abbiamo già scritto (e che si trovano nel download allegato all'articolo). Oltre a quelli di descrizione, è presente un file di constraint (*crono_example.ucf*) che consente di associare ogni segnale di uscita del nostro sistema ai pin fisicamente connessi alla scheda di sviluppo; le informazioni a riguardo possono essere ricavate dal manuale utente o direttamente dallo schema elettrico.

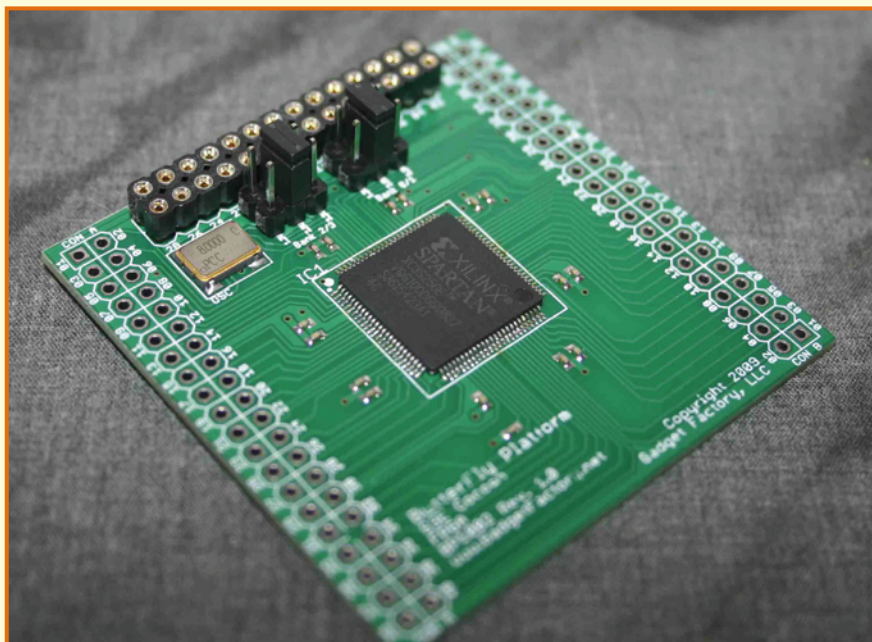
L'ultima finestra riporta un riassunto delle impostazioni che abbiamo selezionato. Confermando, se non abbiamo commesso errori, viene finalmente creato il progetto. ISE ritorna

alla finestra principale che appare come mostrato nella **Fig. 8**. Selezionando *Implement top module* dal menu *Process*, viene avviato il processo di sintesi e place&route. Grazie ad una serie di script predefiniti, il flusso procede automaticamente e se, come dovrebbe, non si verificano errori, il processo termina con la generazione del bitstream. Accanto ad alcune delle icone che rappresentano le diverse fasi, potrebbero evidenziarsi dei punti esclamativi che richiamano l'attenzione del progettista segnalando la presenza di warning. Nel nostro caso possiamo trascurare queste indicazioni.

PROGRAMMAZIONE DEL DISPOSITIVO E TEST

L'ultimo passo è la programmazione del dispositivo: connettiamo il kit di sviluppo al PC mediante il cavo USB a corredo, come mostrato nella **Fig. 9** e alimentiamo quindi la scheda. La prima volta ci verrà chiesto di installare dei driver per poter gestire la programmazione; sul kit di sviluppo è infatti presente un dispositivo con interfaccia USB che svolge tali funzionalità. È necessario consentire l'installazione per poter proseguire; nella maggior parte dei casi la richiesta di installazione viene ripetuta tre volte prima di restituire sulla barra degli strumenti di Windows l'indicazione che il nuovo hardware è stato installato correttamente ed è pronto per l'uso; ciò è abbastanza normale.

A questo punto selezioniamo in ISE l'opzione *Configure Target Device*; facendo clic sul pulsante destro del mouse, richiamiamo il menu di pop-up e quindi selezioniamo *run*. Viene aperta una nuova finestra nel riquadro a destra per eseguire l'applicativo *Impact*; per default dovrebbe essere selezionata la voce *Configure Device using Boundary-Scan (JTAG)*. Confermate per procedere oltre. Tramite la connessione USB, il software esegue la scansione dei dispositivi presenti sulla scheda e riconosce l'esistenza di una FPGA Spartan3AN. Richiederà quindi di associare al dispositivo il file di programmazione; selezionate, allora, *crono_example.bit*, presente all'interno della directory del progetto ISE. Nel caso in cui la procedura non venisse eseguita automa-



le memorie non volatili del kit sono già programmate con dati necessari all'esecuzione della demo che abbiamo descritto nella precedente puntata; riscrivendo una di queste difficilmente sarà possibile eseguirla nuovamente a meno di non ripristinare correttamente il contenuto delle memorie sovrascritte. Già nella precedente puntata avevamo sottolineato questo aspetto ed indicato come, nel manuale utente e sul sito, siano riportate tutte le indicazioni ed i file necessari a tale procedura di ripristino.

ticamente, selezionate l'icona della FPGA che appare nella finestra, aprite il menu di pop-up cliccando con il tasto destro sul menu ed eseguite *Assign New Configuration File*. Infine, per iniziare la procedura di programmazione della FPGA, nel riquadro *Processes* presente sulla sinistra della finestra selezionate la voce *Program FPGA Only*. Se tutto è stato eseguito correttamente, siamo finalmente in grado di provare il nostro circuito.

Premendo il pulsante EAST sulla scheda, si inizia il conteggio; premendolo nuovamente lo si arresta. In questa condizione, premendo il pulsante WEST il cronometro viene azzerato. Sul display a carattere dovremmo vedere aggiornato costantemente, una volta al secondo, il tempo misurato.

Tenete conto che la procedura descritta prevede la programmazione della sola FPGA che, come chiarito nelle precedenti puntate, è un dispositivo volatile; quindi spegnendo la scheda di sviluppo la configurazione del dispositivo si perde e, per poter eseguire nuovamente il nostro esempio di riferimento, sarà necessario ripetere la procedura precedente. In alternativa potete programmare una delle memorie non volatili presenti sulla scheda (compresa ad esempio la flash interna della FPGA) e configurare il kit affinché all'accensione l'FPGA carichi automaticamente da questa il bitstream di configurazione. Nel manuale utente del kit sono spiegate in dettaglio le modalità su come procedere in questi casi. Fate solo attenzione al fatto che

CONCLUSIONI

In questa puntata abbiamo mostrato come è possibile realizzare un semplice progetto con FPGA partendo da zero usando il nostro kit di sviluppo. Chiaramente ci sono molti aspetti del problema che non sono stati affrontati ed altri sono solo stati accennati ma riteniamo che con le poche indicazioni fornite e seguendo l'esempio di riferimento proposto, ci siano ampi margini per poter iniziare a lavorare realizzando una propria applicazione. Alcune cose saranno più chiare solo con l'uso; su altre avremo modo di tornare in futuro.

Come anticipato in precedenza, una delle caratteristiche più interessanti delle moderne FPGA, è la disponibilità di una collezione di blocchi funzionali (per i quali si usa il termine di IP – Intellectual Property) già sviluppati e che possono essere inseriti all'interno del proprio progetto. Alcuni sono disponibili gratuitamente, altri sono proprietari e distribuiti direttamente dai produttori di logiche programmabili o da terze parti. Tra questi moduli, vi sono ad esempio core di processori, UART, interfacce PCI, MAC Ethernet, moduli DMA, eccetera. Si può così realizzare all'interno della FPGA un vero e proprio sistema complesso, più o meno corrispondente a quello che qualche anno fa si faceva con una scheda intera! Nella prossima puntata vedremo un esempio di ciò basandoci sempre sul nostro kit ed usando l'ambiente di sviluppo Embedded Development Kit di Xilinx che supporta il microprocessore proprietario MicroBlaze. ■



TRASMETTITORE E RICEVITORE 8 CH PER RADIOMODELLI

Gestisce quattro canali proporzionali (destinati a servocontrolli, driver motori e servorelé proporzionali) e quattro ON-OFF, per attivare carichi a 6 V/500 mA. Il display di cui è dotato il TX permette di configurare facilmente tutti i parametri di funzionamento quali inversione del canale, regolazione del neutro e sensibilità.

..... di CRISTIANO RUGGERI

Negli anni abbiamo pubblicato molti sistemi di controllo e comando remoto via radio: ricorderete certamente le varie applicazioni di telecomando GSM oppure Bluetooth, così come i tanti radiocomandi (a 1-2-4 e più canali) realizzati con i moduli radio e di decodifica prodotti dall'Aurel. Però alcuni lettori ci hanno fatto notare che, nonostante le tante pubblicazioni di "oggetti funzionanti via etere", non avevamo ancora realizzato un sistema trasmettitore-ricevitore a controllo proporzionale, ossia dotato di uscite il cui

stato fosse variabile in proporzione al valore del comando inviato dal trasmettitore. In altre parole, non avevamo ancora presentato un sistema TX-RX utilizzabile in ambito modellistico, dove il solo controllo di tipo ON-OFF non è sufficiente se non per comandare carichi che funzionino in modo "acceso/spento". Ora colmiamo questa lacuna proponendo un telecomando proporzionale composto da un'unità trasmettente ed una ricevente. Il progetto, pur essendo relativamente semplice sul piano hardware, ha richiesto molto impegno per lo sviluppo del

Trimming, Reverse e Gain dei servocontrolli

Vediamo tre parametri fondamentali nella gestione dei servocontrolli, ipotizzando di utilizzarli su una macchina radiocomandata, con motore a scoppio, dotata di un servocontrollo per lo sterzo ed uno che agisce sull'acceleratore da una parte e sul freno dall'altra.

Trimming

Dopo la fase di accensione del modello, la situazione tipica prevede che trasmettente, ricevente e motore a scoppio siano tutti accesi, con i joystick in posizione di neutro; il modello deve rimanere assolutamente fermo, con sterzo al centro, motore al minimo e freno a riposo o leggermente in tiro. In pratica, però, date le tolleranze meccaniche sia dei servocontrolli sia dei leveraggi che agiscono sullo sterzo e sul gruppo acceleratore/freno, potrebbe capitare che le ruote sterzanti puntino leggermente a destra o sinistra invece che al centro. Analogamente, il servo che controlla sia il carburatore del motore a scoppio sia il freno, potrebbe tenere il motore leggermente accelerato oppure il freno azionato.

Il parametro trimming serve proprio per correggere questi errori meccanici (normalmente calibrati in fase di costruzione entro i limiti del possibile) spostando il valore della posizione di neutro del servocontrollo in modo da compensare e correggere ulteriormente gli errori dovuti alle tolleranze meccaniche.

Reverse

Dopo aver calibrato le posizioni di neutro agendo sul parametro di trimming, occorre verificare che il verso di movimento dei servocontrolli sia quello giusto: capita, infatti, che spostando a destra il controllo dello sterzo le ruote si muovano a sinistra, oppure spostando in avanti il controllo velocità in realtà si agisca sul freno invece che sulla valvola del carburatore.

Ciò non è dovuto a problemi o errori progettuali bensì alle costrizioni meccaniche del modello, normalmente piccolo, in cui i servocontrolli e i relativi leveraggi vengono, il più delle volte, posizionati meccanicamente "come e

dove è possibile" invece che "come e dove si vorrebbe". I nostri amici modellisti, infatti, ci hanno spiegato che la posizione dei vari attuatori meccanici influisce molto sul bilanciamento del modello, specialmente quando si muove in un ambiente fluido (navi e sommergibili) o aereo (aerei ed elicotteri) e che, quasi sempre, la posizione scelta per il montaggio di un servocontrollo è legata a sperimentazioni, esperienza personale e, ultimo ma non ultimo, allo spazio di cui si dispone. Il parametro reverse serve appunto per invertire la direzione di spostamento del servocontrollo, ovviando in maniera elettronica ad una limitazione meccanica obbligatoria.

Nota: a questo punto, talvolta potrebbe essere necessario ripetere la procedura di trimming.

In altri casi potrebbe essere necessario effettuare il reverse prima del trimming.

Gain

Stabilito che le posizioni di neutro siano a posto e che i servocontrolli vengano mossi nell'esatta direzione dai joystick della trasmettente, occorre impostare il guadagno, ossia il parametro che permette di variare l'entità della sterzata e i giri motore in funzione dello spostamento dei joystick.

Un modellista alle prime armi, infatti, potrebbe voler fare esperienza associando alla massima escursione dei joystick una sterzata massima piuttosto larga e dei giri motore bassi, così da avere un modello lento e dalla sterzata morbida: la situazione è certamente poco entusiasmante, ma permette di fare esperienza nel pilotaggio RC limitando i rischi di far "sbattere" rovinosamente il modello contro un palo o il bordo di un marciapiede.

Al contrario, un modellista esperto preferirà avere tra le sue dita il controllo totale sia della sterzata che della velocità/frenata, così da poter decidere, in base alla sua esperienza e capacità, i regimi adatti momento per momento. Il parametro gain consente appunto di variare la sensibilità necessaria per controllare il modello.

i parametri fondamentali che devono essere configurabili.

LE SPECIFICHE DEL PROGETTO

Nel procedere con le varie argomentazioni, diamo per scontato che sappiate come funziona un servocontrollo o un servomotore per uso modellistico; in caso contrario, suggeriamo di leggere l'articolo relativo al "tester per servocontrolli" pubblicato sul fascicolo n° 126. Veniamo ora al nostro sistema: è noto come il trasmettitore di un radiocontrollo per modellismo disponga di almeno due joystick che, a seconda del numero dei canali proporzionali, si possono muovere solo sull'asse x, solo sull'asse y, oppure su entrambi gli assi, esattamente come avviene nei joystick per computer. Quale che sia l'asse di movimento, quando la leva dei joystick viene rilasciata fa ritorno verso una posizione di neutro prossima al centro (o incrocio) degli assi: tale posizione è nota come *posizione di neutro* (anche di centro o di zero) e rappresenta la situazione in cui il modello rimane fermo (sterzo al centro, motori fermi, timone al centro, eliche ferme eccetera).

Relativamente alla posizione di neutro e all'escursione dei joystick, esistono molti parametri configurabili (legati al tipo di modello che si vuole comandare) e, dei tanti, i più importanti sono il valore di zero (trimming), l'inversione (reverse) e il guadagno (gain) che abbiamo commentato in un riquadro a parte, ipotizzando di "guidare" una macchinina con motore a scoppio con due soli controlli proporzionali, ossia "sterzo" e "giri motore / freno". I radiocomandi professionali dispongono di altri parametri configurabili, che permettono ad esempio di variare la rampa di risposta del servo nei riguardi

firmware, dato che un sistema di controllo proporzionale deve poter essere completamente configurato dall'utilizzatore, al contrario dei sistemi ON-OFF che, il più delle volte, richiedono

solo l'apprendimento della codifica del trasmettitore da parte del ricevitore. Prima di progettare il nostro sistema, anche con l'aiuto di alcuni nostri amici esperti modellisti, abbiamo identificato

dello spostamento del joystick, oppure di compensare il tremore delle mani di chi sta pilotando, o miscelare due canali su un solo canale, così da comandare due servocontrolli tramite un solo asse del joystick e molto altro ancora, ma la realizzazione di quanto sopra complicherebbe notevolmente tanto l'hardware quanto il firmware.

Noi abbiamo preferito rivolgerci ad un pubblico meno esigente, che potrebbe utilizzare il nostro sistema con piccole imbarcazioni, macchinine elettriche, piccoli robot e, in linea di massima, tutti quegli oggetti dinamici che non corrono troppi rischi di danneggiamento di cose e persone durante il normale uso.

SCHEMA ELETTRICO

Prima di descrivere gli schemi elettrici di TX ed RX, è necessario puntualizzare tre esigenze di progetto specifiche e precisamente:

- possibilità di leggere dal trasmettitore, via radio, lo stato della batteria del ricevitore;
- cambio del canale RF senza sostituire il quarzo o senza agire su entrambe le schede;
- cambio della potenza RF di entrambi i dispositivi.

Dal primo punto appare evidente che la sezione radio del ricevitore debba essere in grado, oltre che di ricevere i comandi dal trasmettitore, di trasmettere a sua volta i dati relativi alla batteria e, di conseguenza, anche la sezione radio del trasmettitore deve poter operare in modo bidirezionale. Dai punti successivi, invece, si deduce che, oltre ad essere bidirezionale, la sezione radio di entrambi gli apparati deve essere abbastanza evoluta da poter operare con canali e potenze RF differenti. Dopo svariate verifiche, abbiamo deciso di utilizzare in

entrambi i circuiti il radiomodem XTR-7020A-4 di casa Aurel. Questo "oggetto" risponde per-

di configurazione attraverso la porta seriale di cui dispone. Nel nostro progetto i comandi

CARATTERISTICHE TECNICHE TRASMETTITORE

- **Numero canali:**
 - 4 proporzionali su due joystick; previsti controllo di gain, trimming e reverse.
 - 4 on-off a interruttore, con controllo di reverse.
 - Interruttori: a zero centrale, posizione fissa da una parte e con ritorno dall'altra.
- **Frequenza di trasmissione:** 433 MHz circa.
- **Canali radio utilizzabili:**
 - 10 (1: 433,19 - 2: 433,34 - 3: 433,50 - 4: 433,65 - 5: 433,80 - 6: 433,96 - 7: 434,11 - 8: 434,27 - 9: 434,42 - 10: 434,57 predefinita).
- **Potenza emessa:**
 - 4 livelli di potenza RF (- 8 dBm, - 2 dBm predefinita, + 4 dBm, + 10 dBm).
- **Controllo stato batteria:** sia del trasmettitore che del ricevitore.
- **Visualizzazione parametri:** display 2 linee per 16 caratteri + 1 led bicolore.
 - Retro-illuminazione impostabile su tre livelli basso-medio-alto e su controllo automatico in funzione della luce ambiente.
 - Parametri modificabili: trimming, gain, reverse, canale RF, potenza RF, retro-illuminazione display, abilitazione/disabilitazione toni di segnalazione.
 - Parametri di sola lettura: batteria trasmettitore e ricevitore.
- **Impostazione parametri:** tramite menu a display e sei pulsanti di comando, due in centro ai joystick + quattro interruttori di comando dei canali on-off.
- **Segnalazione attività radio e situazioni di errore:** tramite led bicolore, messaggi e buzzer.
- **Alimentazione:** 5,5÷9 Vdc (ottimale: sei pile AA NiMH da 1,2 V / 2,5 A/h).
- **Assorbimento:** 200 mA a 7,2 V, con potenza RF e retro-illuminazione a livello massimo.
- **Autonomia:** 12 ore circa con batterie a piena carica e utilizzo nelle condizioni peggiorative.

CARATTERISTICHE TECNICHE RICEVITORE

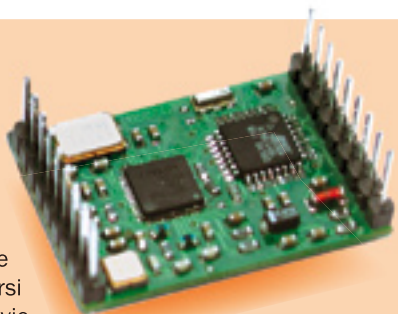
- **Numero canali:**
 - 4 proporzionali;
 - 4 on-off con selettore di modo proporzionale o logica on-off.
 - Modalità logica: gestisce direttamente carichi come relé e lampadine (6 V massimi).
 - Modalità proporzionale: ON = impulsi da 1,8 ms a 40 Hz; OFF = impulsi da 1,2 ms a 40 Hz.
 - Il selettore di modo viene configurato tramite un registro software.
- **Frequenza di ricezione:** 433 MHz circa.
- **Canali radio:**
 - 10 (1: 433,19 - 2: 433,34 - 3: 433,50 - 4: 433,65 - 5: 433,80 - 6: 433,96 - 7: 434,11 - 8: 434,27 - 9: 434,42 - 10: 434,57 predefinita).
- **Controllo stato batteria.**
- **Connettore uscite e alimentazione:** compatibile con i servocontrolli, i servomotori e i pacchi batterie delle marche più diffuse.
- **Impostazione parametri:** tramite trasmettitore. I parametri vengono memorizzati in eeprom e riletta ad ogni accensione.
- **Segnalazione attività radio/errori:** tramite led bicolore.
- **Alimentazione:** 3,5÷6 Vcc, pacco batterie singolo o doppio.
- **Assorbimento:** 50 mA a 6 V, con potenza RF massima, per il solo ricevitore; nessun servo o relé collegato.
- **Autonomia:** legata al consumo dei dispositivi collegati alle uscite.

fettamente alle nostre esigenze, dato che supporta 10 canali RF, trasmette con quattro livelli di potenza, è bidirezionale e (cosa apprezzabile) riceve tanto i dati da trasmettere quanto i comandi

vengono inviati al radiomodem tramite il noto set AT (facile, intuitivo ed efficace) che semplifica l'interazione con i microcontrollori del ricevitore e del trasmettitore.

Il radiomodem

Il modulo utilizzato nelle unità trasmettente e ricevente è un ricetrasmittitore, ossia un modem via radio: si tratta dell'XTR-7020A-4 dell'Aurel. Il componente opera in UHF a 433 MHz e può sintonizzarsi su 10 diversi canali a seconda di come viene impostato; trasmette con quattro livelli di potenza, è bidirezionale e dispone di un'interfaccia seriale TTL da cui riceve tanto i dati da trasmettere quanto i comandi di configurazione. La velocità di comunicazione seriale può essere scelta tra 9.600, 19.200, 38.400, 57.600 e 115.200 bps. La potenza RF irradiata si può impostare da -8 a +10 dBm.



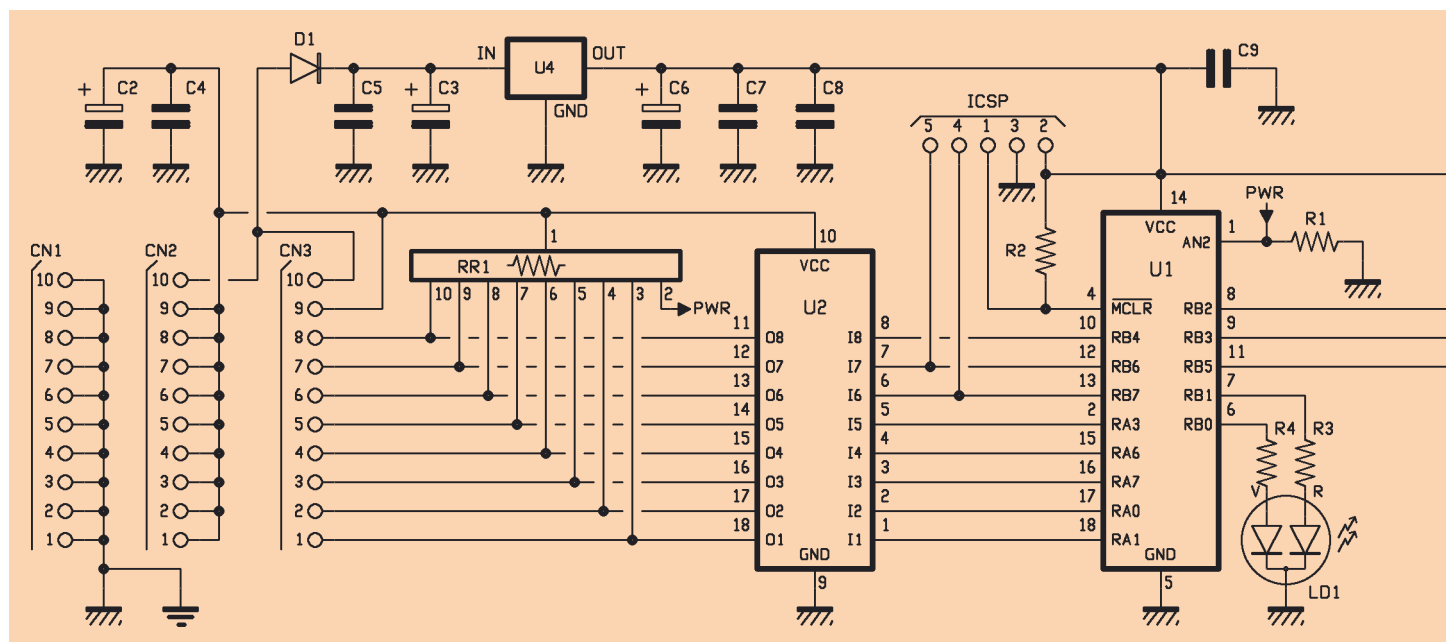
RICEVITORE

Si tratta di un circuito abbastanza semplice, che orbita attorno al microcontrollore U1, un PIC16F88 scelto perché dispone di tutte le periferiche necessarie, ossia una seriale hardware bidirezionale, utilizzata per comunicare con il radiomodem, un A/D converter a 10 bit, con cui leggere lo stato della batteria, un numero di GPIO digitali sufficiente per gestire le otto uscite, nonché il led bicolore di segnalazione ed un preciso oscillatore interno a 8 MHz che ci permette di evitare l'uso di un quarzo esterno. Ma

procediamo per gradi, partendo dal segnale radio captato dal radiomodem U3: esso viene decodificato e trasformato in caratteri seriali trasferiti sulla linea di ricezione RSRX (pin 12), a sua volta collegata alla linea di ricezione RX (pin 9) del micro U1. I dati ricevuti vengono elaborati in modo differente a seconda che si tratti di comandi di configurazione o controllo delle uscite. Nel primo caso i comandi generano delle azioni interne al micro che, una volta terminati i vari processi, risponde attraverso la linea TX (pin 11) collegata alla trasmissio-

ne del radiomodem (RSTX, pin 14). Quest'ultimo trasmette via radio i dati ricevuti dal micro, siano essi il valore della tensione di batteria o la conferma/richiesta di esecuzione del comando. A questo punto il trasmettitore, che diamo per scontato essere in una condizione di attesa, riceve la risposta da parte del ricevitore, la elabora e si prepara per le successive operazioni. Nel secondo caso, invece, si tratta di comandi alle uscite e pertanto il PIC genera sui pin 18 (CH1) - 17 (CH2) - 16 (CH3) - 15 (CH4), che sono tutti uscite proporzionali, una serie di impulsi di durata legata al comando ricevuto via radio. Per quanto riguarda le uscite logiche poste sui pin 2 (CH5) - 13 (CH6) - 12 (CH7) - 10 (CH8) il discorso è leggermente differente: possono funzionare in modo sia ON-OFF che proporzionale, a seconda dello stato della costante RELE_MODE, modificabile via software. Se RELE_MODE vale 0, allora le uscite da 5 a 8 lavorano in modo logico, ossia ON-OFF; se, invece, RELE_MODE vale 1, le uscite funzionano in modo

[schema elettrico RICEVITORE]



[piano di montaggio **RICEVITORE**]

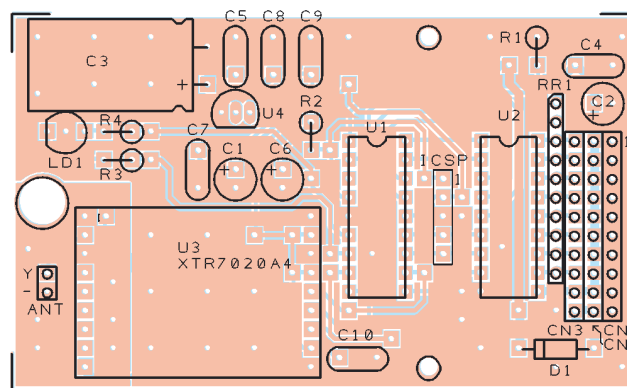
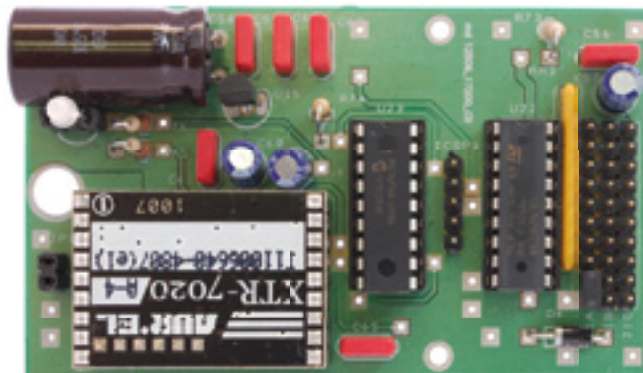
proporzionale, assegnando degli impulsi da 1,2 ms allo stato di OFF e degli impulsi da 1,8 ms allo stato di ON. Tutte le linee di GPIO vengono amplificate in corrente dai driver open-collector di U2 (ULN2803), le cui uscite sono collegate sia al connettore JP4, sia alla rete resistiva RR1, utilizzata per il pull-up necessario a stabilire l'uno logico all'ingresso dei servocontrolli o dei servo-relé. Una delle resistenze di RR1, insieme alla resistenza R1, realizza un partitore resistivo il cui nodo centrale, collegato all'ingresso analogico AN2 (pin 1 di U1), permette di tenere sotto controllo la tensione di batteria. Il led bicolore LD1, gestito dalle linee 6 e 7 del PIC, fornisce varie indicazioni sullo stato operativo del ricevitore, segnalate da lampeggi e colori differenti, come vedremo più avanti nei paragrafi dedicati all'utilizzo in pratica. Relativamente allo stadio di alimentazione, si tratta del classico regolatore lineare serie che, invece di impiegare un comune 78L05, fa uso di un Microchip MCP1702 (U4) nella versione a 3 V, scelto in

Elenco Componenti:

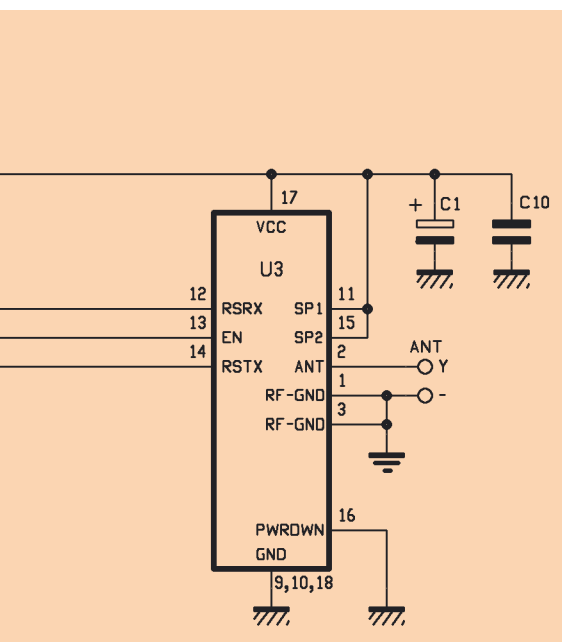
R1: 1,8 kohm
R2: 10 kohm
R3, R4: 330 ohm
RR1: Rete Resistiva 9 x 2,2 kohm + Comune
C1, C2, C6: 10 μ F 16 VL elettrolitico
C3: 2200 μ F 16 VL elettrolitico
C4, C5, C7÷C10: 100 nF multistrato
U1: PIC16F88-I/P (MF839L/MF839P)
U2: ULN2803
U3: XTR7020A-4
U4: MCP1702-3002E
LD1: led 3 mm bicolore
D1: 1N4007

Varie:

- Zoccolo 9+9 (2 pz.)
- Strip maschio 5 vie
- Strip maschio 10 vie (3 pz.)
- Strip femmina 3 vie



- Strip femmina 2 vie
- Strip femmina 9 vie (2 pz.)
- Cavaliere passo 2,54mm
- Circuito stampato
- Contenitore SC700

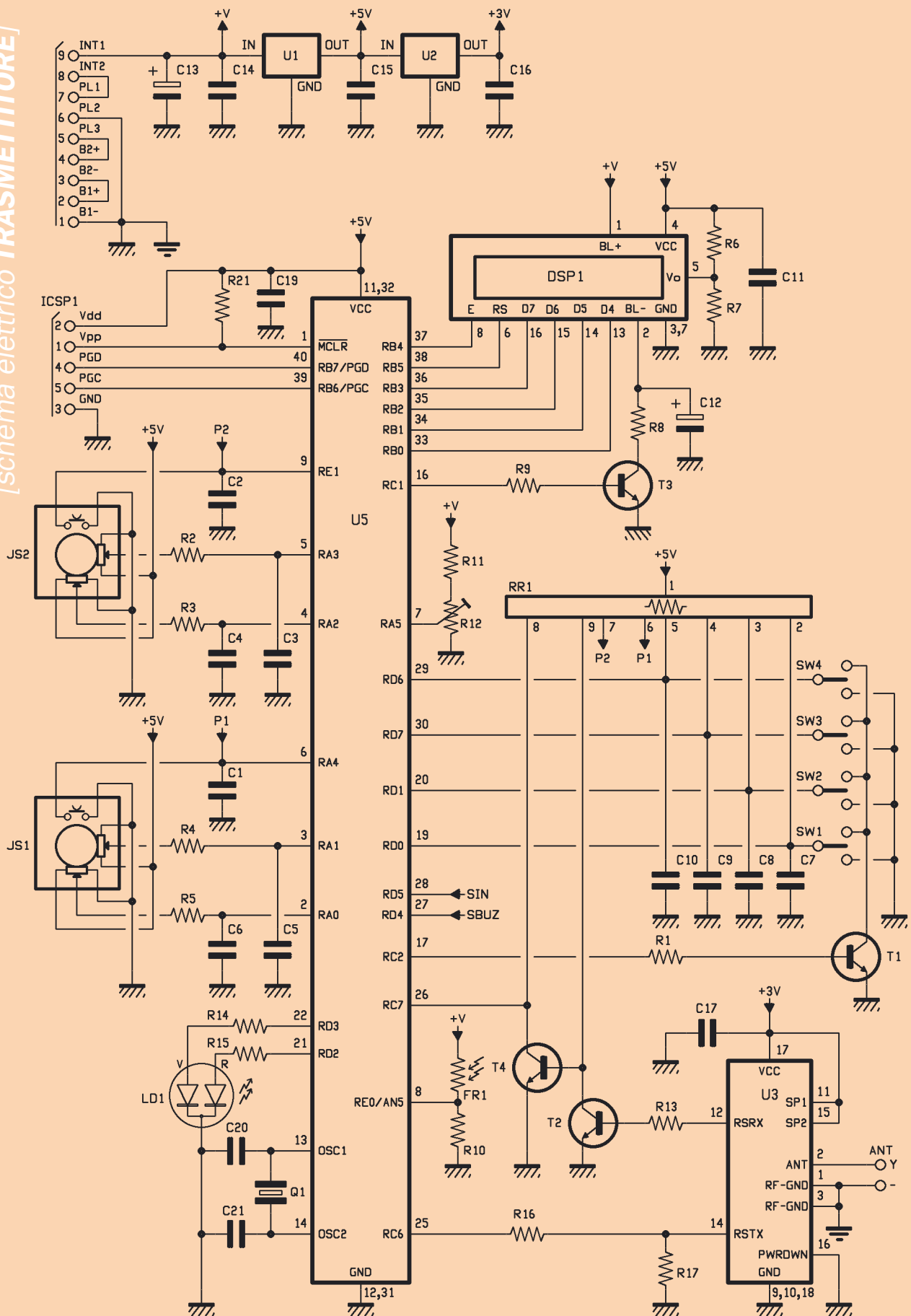


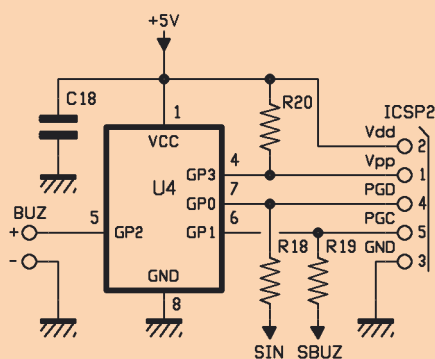
quanto si tratta di uno dei pochi regolatori lineari low-drop-out a 3 V in case TO-92 in grado di erogare 250 mA continui. Questo stadio può funzionare sia con singola che con doppia batteria (3,5÷5,5 V). Nel primo caso, i pin 9 e 10 di CN3 devono essere cortocircuitati tramite un cavaliere a passo 2,54 mm; la batteria può essere collegata indifferentemente sui pin 10 di CN2 (barra del positivo) e CN1 (barra del negativo) o sui pin 9. A doppia batteria, invece, i pin 9-10 di CN3 devono rimanere aperti: la batteria collegata ai pin 9 di CN1 e CN2 alimenta la sola sezione di potenza (ULN2803-U2 e uscite) mentre quella collegata ai pin 10

(sempre di CN1 e CN2) alimenta la sezione elettronica di processo (micro e radiomodem). Il filtro realizzato dal diodo D2 e dal condensatore C3 da 2.200 μ F, riduce drasticamente il rumore elettrico generato dai motori dei servocontrolli in movimento, rumore che potrebbe influenzare il funzionamento del microcontrollore, soprattutto quando si verificano azionamenti contemporanei di più servocontrolli. Completa il circuito il connettore ICSP, necessario per la programmazione in-circuit del PIC.

TRASMETTITORE

Lo schema del trasmettitore è alquanto complesso, poiché il



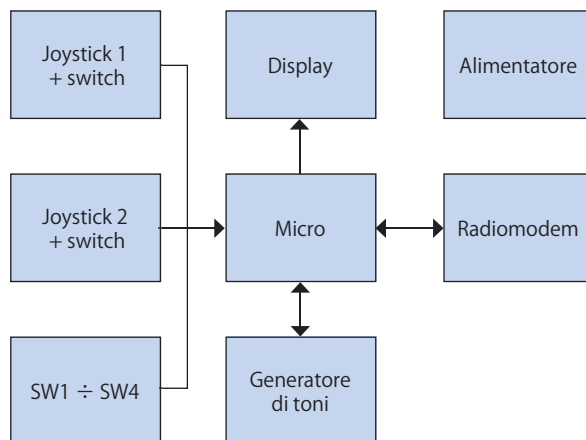


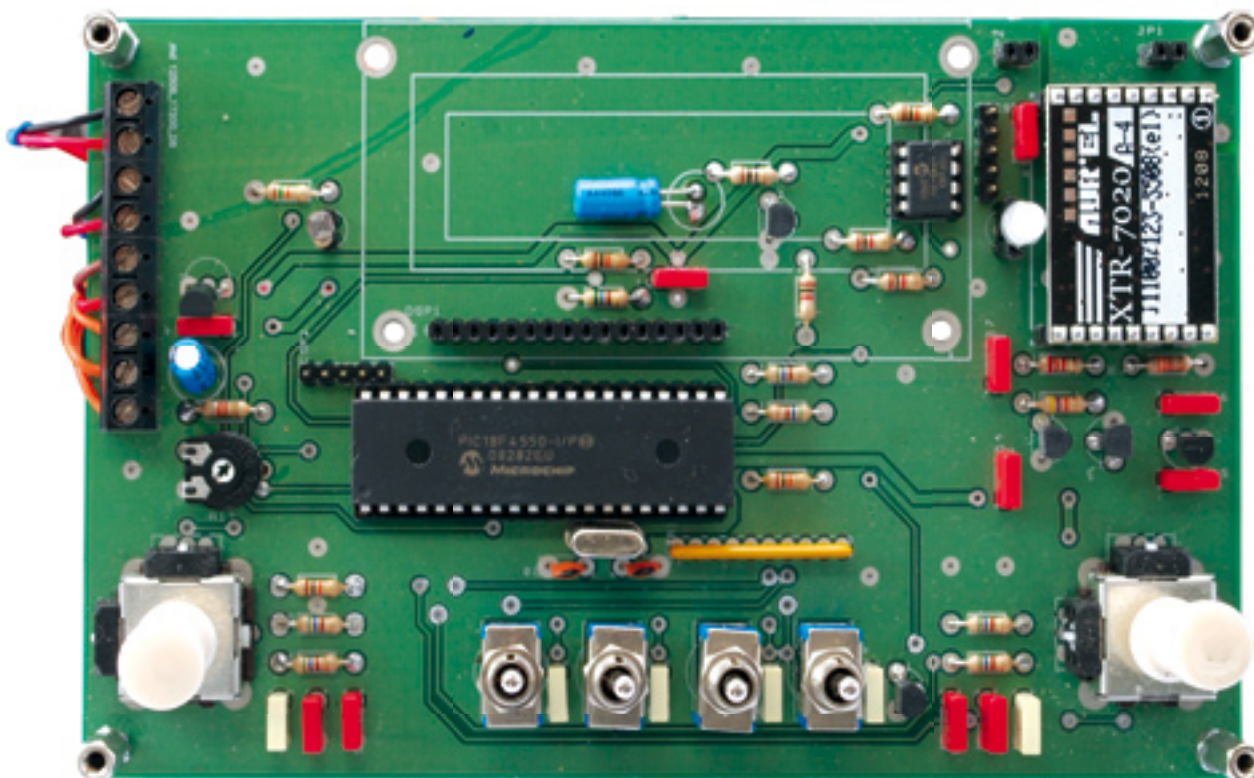
numero di compiti che deve svolgere è assai elevato; è pertanto meglio analizzarlo prima lo schema a blocchi. Ancora una volta, il cuore del circuito è il microcontrollore, cui fanno capo i due joystick a due assi e i sei switch di comando (due nei joystick e quattro aggiunti esternamente). I segnali provenienti dai joystick e dai pulsanti vengono processati dal micro, che aggiorna il display, gestisce il generatore di toni (è lo stesso pubblicato nel fascicolo 139) ed elabora i dati da inviare al radiomodem e destinati al ricevitore. Se necessario, si pone in una condizione di attesa di risposta dal ricevitore, condizione che, a seconda della risposta ricevuta, può semplicemente forzare un aggiornamento del display oppure mandare il trasmettitore in una nuova condizione di attesa di istruzioni da parte dell'utente. Il microcontrollore adottato, U5 (funzionante a 20 MHz) è un PIC18F4550 in formato DIL 40 pin, scelto sia per l'elevato numero di I/O, sufficienti per tutte le esigenze del circuito, sia per la memoria programma da 32 kB. Dispone (tra le altre periferiche) anche di una porta USB device e di un PLL in grado di elevare la velocità del clock a 50 MHz, che qui non sono stati utilizzati. Tutti i segnali provenienti dai joystick e dagli switch convergono al micro. I joystick dispongono di due potenziometri da 10 kohm, uno per l'asse x e uno per l'asse y, e di un pulsante normalmente aperto, il cui contatto si chiude premendo la leva. Da ognuno di

essi vengono prelevati tre segnali di controllo, ossia un segnale logico proveniente dal pulsante (0 V = attivo; 5 V = a riposo) e due segnali analogici di ampiezza compresa tra 0 e 5 V circa, con valore di centro (o neutro o riposo) di $2,2 \div 2,8$ V (i potenziometri hanno una tolleranza del 30 %). Tutti questi segnali vengono filtrati opportunamente: facendo riferimento a JS1, la linea pulsante viene "pulita" dal condensatore C1, mentre le due linee analogiche vengono filtrate dalle celle RC R5-C6 e R4-C5. In questo modo i segnali elettrici visti dagli ingressi del micro sono puliti dai rumori elettrici, causati dal rimbalzo dei contatti durante la commutazione del pulsante e dallo sfregamento del cursore dei potenziometri sull'elemento resistivo. Naturalmente la stessa cosa vale per JS2. I quattro switch SW1÷SW4 sono utilizzati sia per attivare le uscite logiche da 5 a 8 sia per impostare il sistema durante le varie fasi di configurazione: si tratta di quattro deviatori a levetta di tipo 1-0-1 con ritorno; con la levetta al centro nessun contatto è chiuso, spostando la levetta da una parte si chiude un contatto in modo permanente e spostandola dall'altra parte, il contatto rimane chiuso fino a che non si rilascia la levetta. In questo modo, durante il normale funzionamento del sistema TX-RX è possibile attivare le uscite sul ricevitore sia in maniera stabile che impulsiva. Durante la fase di configurazione degli apparati, invece, è possibile utilizzare solo il comando impulsivo degli switch: dallo schema elettrico, infatti si nota che i contatti con ritorno sono collegati direttamente alla massa

del circuito, mentre quelli stabili sono collegati ad una linea di massa virtuale, controllata dal transistor T1. Durante il normale funzionamento, la base di T1 è polarizzata e pertanto la linea di massa virtuale alimentata dal collettore del transistor è virtualmente collegata a massa. Durante la configurazione, invece, T1 viene interdetto e pertanto la linea di massa virtuale è in alta impedenza, quindi i deviatori funzionano solamente sul lato monostabile. La rete resistiva RR1, insieme ai condensatori C7÷C10, realizza sia la funzione di pull-up per alcuni circuiti che di filtro RC, necessario per ripulire dai disturbi di commutazione i segnali elettrici che dagli switch vanno al micro. Quest'ultimo controlla direttamente il display (2 righe x 16 caratteri, modello DCL4162 Velleman) sulle linee di I/O RB0÷RB5, ma non solo: attraverso la seriale hardware bidirezionale (RX – pin RC7 e TX – pin RC6) colloquia con il radiomodem U3 (XTR-7020A-4) e tramite le linee RD5 e RD4 (che emulano una seconda seriale hardware bidirezionale) gestisce il generatore di toni, che a sua volta controlla direttamente il buzzer piezoelettrico collegato al connettore BUZ.

Schema a blocchi del trasmettitore





Elenco Componenti:

R1, R6, R20, R21: 10 kohm
R2÷R5, R7, R14, R15: 560 ohm
R8: 10 ohm
R9, R18, R19: 1 kohm
R10: 15 kohm
R11: 8,2 kohm
R12: Trimmer 10 kohm

R13: 47 kohm
R16: 2,2 kohm
R17: 3,3 kohm
RR1: Rete Resistiva 9 x 10 kohm
+ Comune
C1, C2: 1 nF multistrato
C3÷C6, C11, C14÷C19:

100 nF multistrato
C7÷C10: 1 nF multistrato
C12, C13: 100 µF 16 VL elettrolitico
C20, C21: 12 pF disco
U1: MCP1702-5002E
U2: MCP1702-3002E
U3: XTR7020A-4

A questo punto sorge un piccolo problema: il micro e il display funzionano a 5 V, mentre il radiomodem funziona a 3 V; per questa ragione il partitore resistivo R16-R17 (2,2 kohm e 3,3 kohm) abbassa a 3 V il segnale a 5 V proveniente dalla linea di trasmissione del PIC.

Analogamente, il circuito realizzato con R13, T2, T4 e due resistenze di RR1, funge da convertitore (traslatore) di livello logico alto (da 3 a 5 V) per la linea di ricezione dal radiomodem verso il micro.

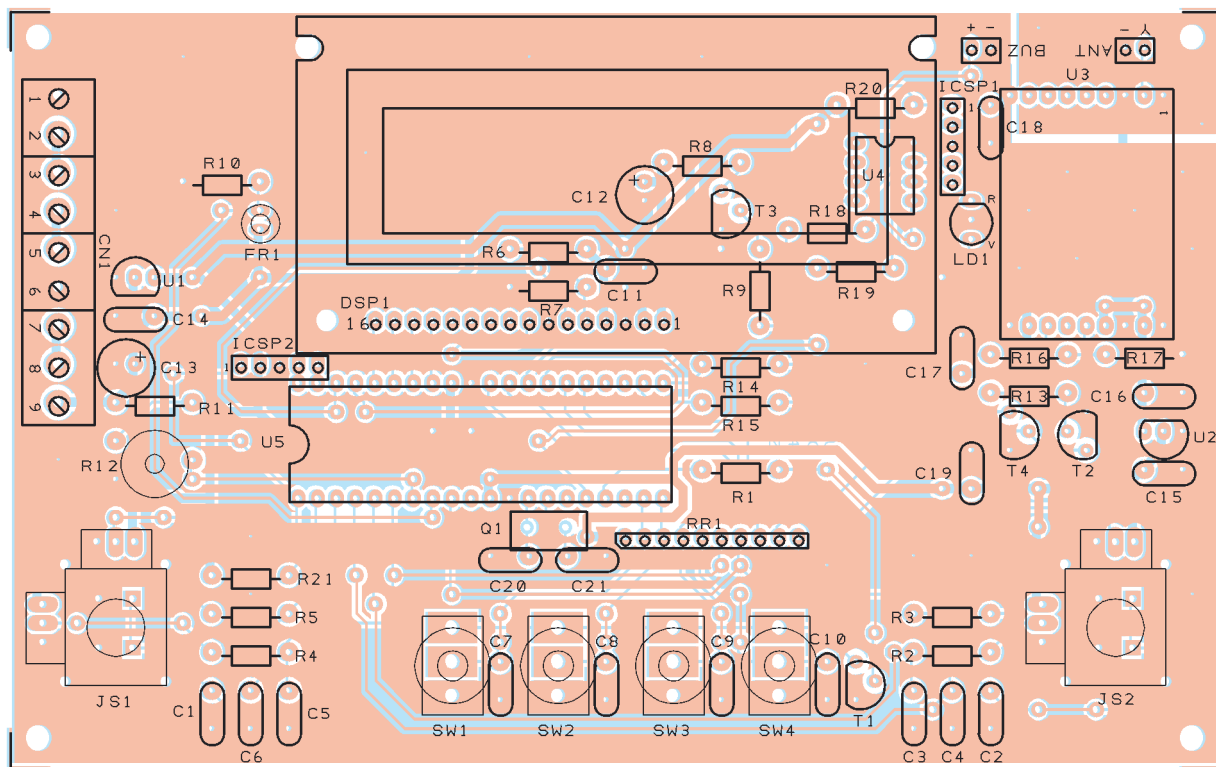
Da ciò si deduce che il blocco di alimentazione deve fornire due tensioni distinte (3 e 5 V): infatti, come visibile dallo schema, impiega due regolatori low-drop-

out MCP1702, uno nella versione a 5 V (U1) e uno a 3 V (U2). Quanto esposto fino ad ora riassume le funzioni principali del trasmettitore; esistono però delle funzioni secondarie che coinvolgono altre parti di circuito: innanzi tutto il led bicolore LD1, controllato direttamente dal micro attraverso le resistenze R14-R15, scandisce le varie fasi dell'attività radio, fasi segnalate con lampeggi di colore e durata differenti (che vedremo più avanti nelle note per l'utilizzo in pratica).

Il circuito composto da R9-R8-T3-C12, invece, realizza il driver PWM per il controllo della retro-illuminazione del display: in pratica, la base di T3 viene pilotata

dal generatore PWM del micro ad una frequenza di 10 kHz attraverso la resistenza R9. Il Duty-Cycle (DC) del generatore di PWM (che fa capo al pin 16) varia secondo il livello di retro-illuminazione desiderato: è possibile scegliere il livello basso (DC 6 %), medio (DC 25 %), alto (DC 95 %) o automatico.

In quest'ultimo caso, il valore del DC varia tra il 4 e il 95 % circa in funzione sia della luce rilevata dalla fotoresistenza R1, collegata a partitore insieme alla R10 e letta dall'ingresso analogico AN5 (pin 8) del microcontrollore, sia della tensione di batteria. Notate, infatti, che FR1 preleva la tensione direttamente dalla batteria invece che dall'alimentazione stabilizza-



U4: PIC12F683-I/P (MF818A)
 U5: PIC18F4550-I/P (MF840)
 Q1: Quarzo 20 MHz
 T1÷T4: BC547
 JS1, JS2: 252A103A60TB
 LD1: Led bicolore 5 mm
 DSP1: Display LCD 16x2
 (CDL4162)
 FR1: Fotoresistenza 2-20 kohm

SW1÷SW4: Deviatore a levetta da
 cs 1-0-1 con ritorno

Varie:

- Morsettiere 2 poli (3 pz.)
- Morsettiere 3 poli
- Zoccolo 4+4
- Zoccolo 20+20
- Strip maschio 2 vie (2 pz.)

- Strip maschio 5 vie (2 pz.)
- Strip maschio 16 vie
- Strip femmina 2 vie (2 pz.)
- Strip femmina 9 vie (2 pz.)
- Strip femmina 16 vie
- Antenna accordata 433 MHz
- Circuito stampato
- Contenitore plastico Teko
- Pult 363

ta: così facendo è possibile legare il livello di retroilluminazione del display sia alla luce ambiente sia al livello di carica delle batterie. In pratica, la condizione "luce forte con batterie parzialmente cariche" determina una retroilluminazione inferiore rispetto a quella che si avrebbe con "luce forte e batterie completamente cariche", permettendo, così di ridurre i consumi. Relativamente a FR1, va detto che le fotoresistenze hanno un'alta tolleranza: nelle stesse condizioni di luce una può avere valore resistivo di 10 kohm e un'altra uguale di meno di 1 kohm.

Durante la taratura occorrerà, pertanto, verificare sperimentalmente se la condizione di batte-

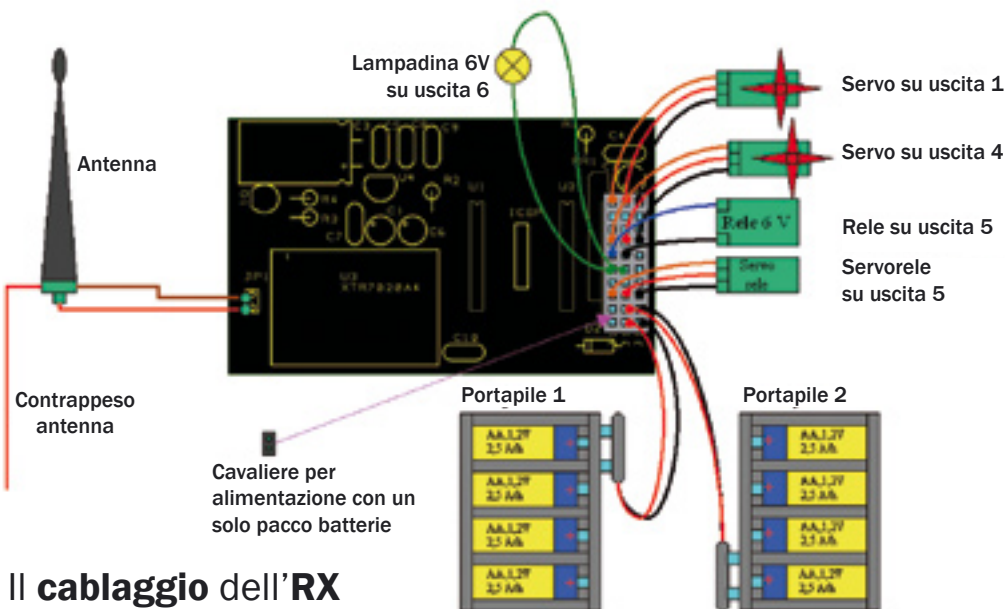
ria carica e illuminazione forte genererà sull'ingresso del micro una tensione superiore a 5 V; nel caso, sarà necessario aggiungere in serie alla fotoresistenza una resistenza di compensazione di valore opportuno, che vedremo come calcolare.

Completano il circuito il partitore per il controllo della tensione di batteria (R11 insieme al trimmer R12), i due connettori ICSP (uno per il micro U5 e uno per il micro U4, che è il generatore di toni), il connettore per l'antenna (ANT), quello per il buzzer (BUZ), ed una morsettiere a 9 morsetti cui è possibile collegare due pacchi batteria da 3,6 V, una presa a tre contatti per ricaricare le batterie, ed un interruttore di accensione.

REALIZZAZIONE PRATICA

Per prima cosa scaricate dal sito della rivista www.elettronica.in i master (doppia faccia in scala 1:1) tramite i quali potete realizzare i due circuiti stampati con la tecnica della fotoincisione.

Forate le piazzole ed effettuate i collegamenti delle vie (piazzole che trasferiscono una linea elettrica da un lato del circuito all'altro). Iniziate il montaggio partendo dai componenti a basso profilo ossia resistenze e condensatori, connettori pin strip a passo 2,54 mm, zoccoli degli integrati eccetera, procedendo via via con i componenti di altezza crescente, quali transistor, condensatori, interruttori e joystick (nel trasmettitore) ecc. Potete aiutarvi con



Il cablaggio dell'RX

i piani di montaggio. Nel caso vogliate inserire i due dispositivi in un contenitore, il trasmettitore può essere facilmente alloggiato in un PULT363 mentre il ricevitore è stato progettato per essere inserito in un SC700, entrambi di casa Teko e acquistabili on-line presso www.futurashop.it. Vi consigliamo di non saldare i led e la fotoresistenza direttamente sullo stampato: è meglio montarli su dei connettori pin strip femmina, in modo da poter facilmente regolare la lunghezza dei reofori in funzione del contenitore. I radiomodem vanno posizionati su due pin strip da 9 pin per lato, dopo avere accorciato di tre mm circa tutti i pin dei radiomodem. Il display del trasmettitore va montato anch'esso su un

pin strip femmina (saldare un pin strip maschio sul c.s. del display dalla parte opposta dell'area visibile) e fissato meccanicamente ai fori sottostanti con delle torrette o dei distanziali da 15 mm circa, che abbiano filetto da 2,6 MA. Fate riferimento agli schemi di cablaggio per collegare il plug, l'interruttore e le batterie nei due circuiti; un'ultima verifica alla ricerca d'errori e/o imprecisioni, e siete pronti per collaudare il sistema.

COLLAUDO TRASMETTITORE

Assicuratevi che i due microcontrollori, il radiomodem e il display non siano montati, quindi alimentate il circuito con una tensione di 9 V (applicata tra i morsetti 1 e 4 della morsettiera) e verificate che sui pin 32 di U5 e 17 di U3 siano presenti le tensioni rispettivamente di 5 e 3 V. Prendete nota del valore esatto della tensione a 5 V (ipotiz-

ziamo 5,15 V). Regolate, quindi, il trimmer R12 in modo da leggere sul suo cursore (pin 7 di U5) una tensione di circa 3,63 V. Per i più pignoli, la formula per calcolare la tensione da leggere sul cursore è la seguente:

$$V_{\text{cursore}} = V_{5V} * [(V_{1-4} * 20) / 255]$$

La quale nel nostro caso specifico diventa:

$$V_{\text{cursore}} = 5,15 * [(9 * 20) / 255] = 5,15 * [180 / 255] = 5,15 * 0,706 = 3,63 \text{ V}$$

Controllate che la fotoresistenza sia montata e illuminata con una sorgente luminosa forte (ad esempio una lampada alogena da tavolo); misurate la tensione presente sul pin 8 di U5: se non supera i 5,15 V non dovete fare nulla. In caso contrario, dovete saldare in serie a uno dei pin della fotoresistenza una resistenza di compensazione, il cui valore deve essere calcolato con i due passaggi descritti qui di seguito.

1) Calcolo della resistenza della fotoresistenza con la formula:

$$R_{\text{fotoresistenza}} = [V_{1-4} / (V_{\text{pin8}} / 15000)] - 15000$$

Se, ad esempio, misurassimo 6 V sul pin 8, otterremmo:

$$R_{\text{fotoresistenza}} = [9 / (6 / 15000)] - 15000 = [9 / 0,0004] - 15000 = 22500 - 15000 = 7500 \text{ ohm.}$$

2) Calcolo della resistenza di compensazione con la formula:

$$R_{\text{comp}} = [(V_{1-4} - V_{5V}) / (V_{5V} / 15000)] - R_{\text{fotoresistenza}}$$

Quindi, con la nostra fotoresistenza da 7500 ohm e la tensione di alimentazione sul +5 di 5,15 V otteniamo:

$$R_{\text{comp}} = [(9 - 5,15) / (5,15 / 15000)] - 7500 = [3,85 / 0,000343] - 7500 = 11213 - 7500 = 3713 \text{ ohm,}$$

Dell'antenna dell'RX si collega al c.s. il solo stilo; alla sua massa collegate uno spezzone di filo lungo uguale, che servirà a "contrappeso".



valore che possiamo approssimare a 3900 ohm.

Con i valori ottenuti, la tensione teoricamente misurabile ora sul pin 8 di U5 vale:

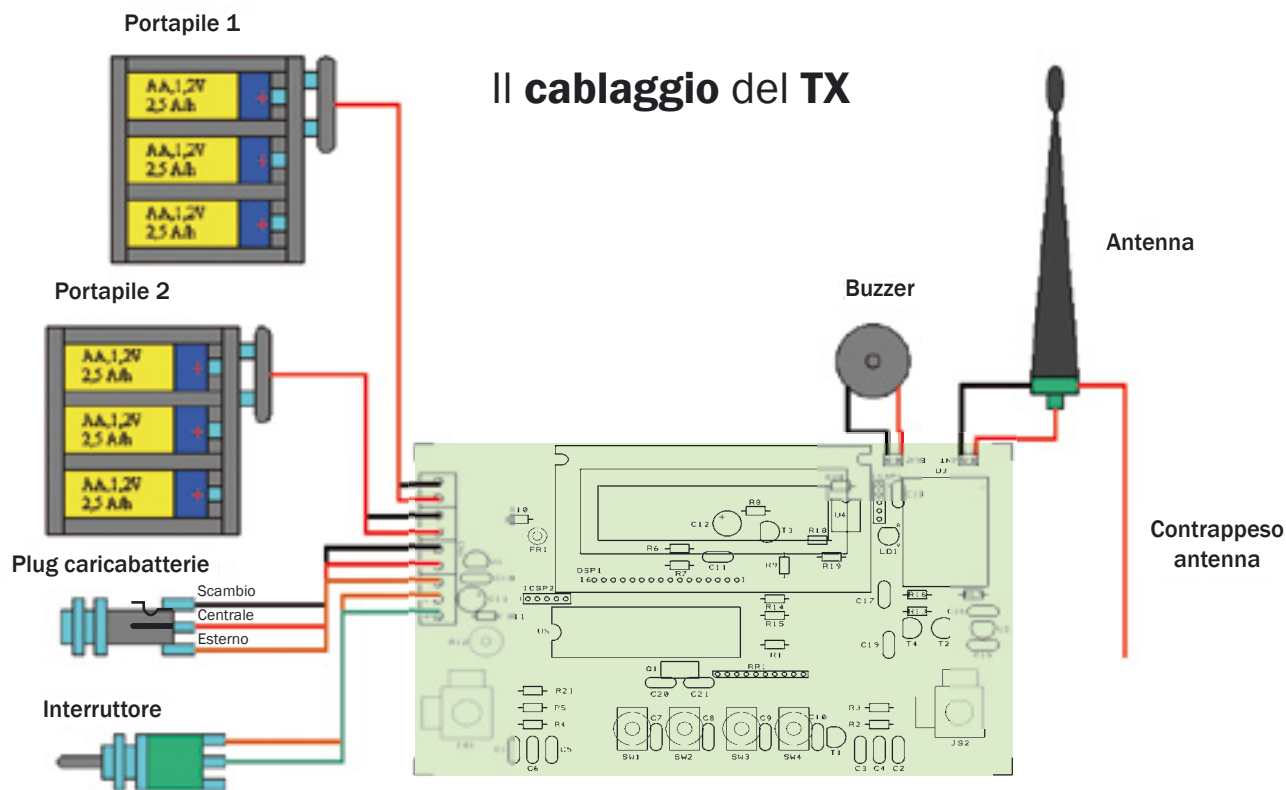
$$V_{pin8} = [V_{1-4} / (R_{comp} + R_{fotoreistenza} + 15000)] * 15000 = [9 / (3900 + 7500 + 15000)] * 15000 = [9 / 26400] * 15000 = 5,11 \text{ V.}$$

Questo valore è inferiore alla nostra tensione di alimentazione (5,15 V) quando, a batterie completamente cariche, il circuito

la programmazione in-circuit (tramite i due connettori ICSP) dei microcontrollori; i file da caricare in U5 (MF840.hex) e U3 (MF819.hex) li scaricate dalla stessa sezione del nostro sito *www.elettronica.in.it*, dove trovate anche i master dei circuiti stampati. Spegnete e riaccendete il trasmettitore, quindi verificate che appaiano in sequenza i messaggi di diagnostica e che sia completata la sequenza di avvio; notate che ad ogni cambio di schermata si

- un lampeggio rosso seguito da uno verde: il modem ha ricevuto il comando di configurazione (rosso) e l'ha processato (verde);
- un lampeggio rosso seguito da uno verde: il modem ha ricevuto il comando di salvataggio/uscita dalla fase di configurazione (rosso) e l'ha processato (verde).

Relativamente alle segnalazioni luminose, la **Tabella 1** riassume tutte le situazioni che si possono

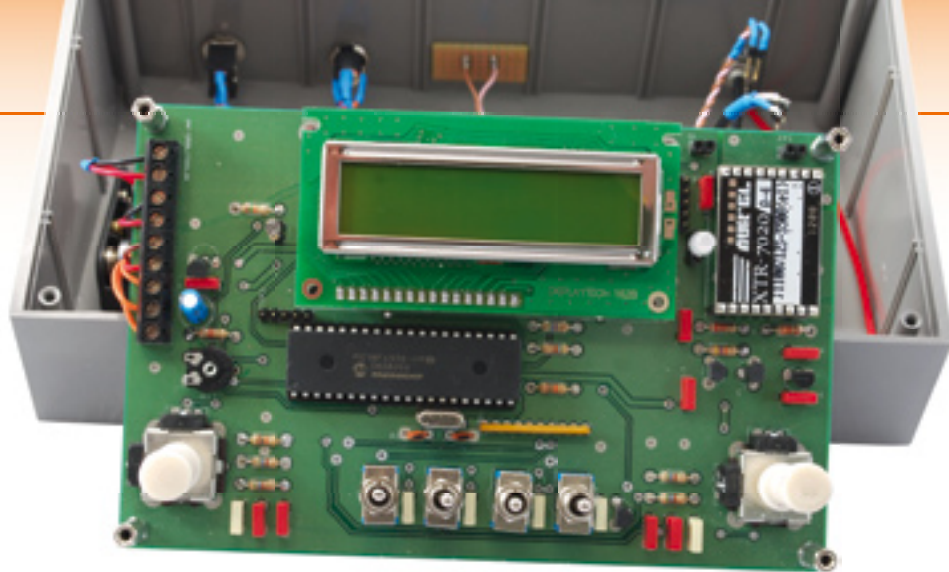


viene alimentato con una tensione di 9 V (6 x 1,5 V). Ora spegnete il trasmettitore, attendete un paio di minuti che i condensatori si scarichino, quindi montate i due microcontrollori, il radiomodem e il display. Buzzer e antenna devono essere già collegati. Accendete nuovamente l'apparato e procedete con

ode un tono acuto breve. Particolare attenzione va posta durante la configurazione del radiomodem: il led lampeggia velocemente di colore giallo, per segnalare che sta iniziando la configurazione della radio quindi si hanno, in sequenza:

- un lampeggio rosso: radiomodem allertato;

verificare, sia sul TX che sull'RX. A questo punto si odono tre toni in scala crescente contemporaneamente al messaggio RADIO OK. Potrebbe, però, apparire un messaggio che segnala che la sezione radio non è stata trovata, accompagnato da una sirena di allarme e da un lampeggio di errore: in questo caso il programma



rimane bloccato e dovete pertanto rivedere il montaggio alla ricerca di errori ed eventuali baffi di stagno tra piste adiacenti. Viene quindi letta la posizione di neutro dei joystick che, durante questa fase, non devono essere toccati. Dopo qualche istante, il trasmettitore si pone in condizione di invio dei valori di neutro alla ricevente, in cui rimane in attesa della richiesta di neutro da parte della ricevente. Dato che quest'ultima, al momento, non è presente, la trasmittente rimane sempre in attesa, senza variare la schermata a display. Premete contemporaneamente i due joystick per forzare l'uscita dalla fase di invio del neutro e passare alla schermata di normale operatività, in cui sono visibili a display i valori numerici delle posizioni dei joystick. Se avete visto tutto

ciò, al 99% il trasmettitore è stato montato correttamente; in caso contrario, dovete riverificare il montaggio.

COLLAUDO RICEVITORE

Senza montare nelle rispettive sedi il micro, il driver d'uscita e il modem, alimentate il dispositivo con una tensione di 6 V in continua, applicata agli ingressi batteria come visibile nel piano di cablaggio (il cavaliere per l'alimentazione singola deve essere inserito) e verificate che siano presenti:

- 3 V circa sul pin 14 di U1;
- 2,7 V circa sul pin 1 di U1;
- 6 V circa sul pin 10 di U2.

Spegnete il circuito, attendete qualche istante che tutti i condensatori (in particolare quello da 2.200 μ F) si scarichino quindi posizionate il microcontrollore PIC16F88, il driver ULN2803 e il modem, riaccendete il circuito e procedete con la programmazione in-circuit del PIC.

Il codice MF839L.hex (gestisce le uscite 5÷8 in modo logico (ON-OFF) mentre il codice MF839P.hex le gestisce in modo proporzionale. Se intendete utilizzare queste uscite con dei carichi passivi (lampadine, led e relé) dovete caricare il codice MF839L.hex mentre se disponete di un servorelé, allora dovete caricare il codice MF839P.hex. Riavviate il dispositivo e verificate le seguenti segnalazioni luminose:

1. Rosso-verde-spegnito per circa 15 secondi: il programma è partito correttamente.
2. Lampeggio giallo veloce: sta per iniziare la configurazione del radiomodem.
3. Un lampeggio rosso: il radiomodem è stato allertato.
4. Un lampeggio rosso seguito da uno verde: il modem ha ricevuto il comando di configurazione (rosso) e l'ha processato (verde).
5. Un lampeggio rosso seguito da uno verde: il modem ha ricevuto il comando di salvataggio e uscita dalla fase di configurazione (rosso) e l'ha processato (verde).
6. Lampeggio rosso a cadenza di 1 secondo: richiesta della posizione di neutro per 120 sec.
7. Lampeggio rosso veloce: standby. Dopo 120 secondi circa in questo stato, il programma riparte dal punto 1 caricando la configurazione predefinita.

UTILIZZO IN PRATICA

Siamo finalmente giunti al momento più atteso: utilizzare il sistema in pratica. Alimentate entrambi i circuiti con le tensioni opportune, accendeteli contemporaneamente e verificate che entrambi diano le segnalazioni già viste durante il collaudo. Il messaggio di invio della posizione di neutro rimane visualizzato fino a che il ricevitore non manda la richiesta via radio: il trasmettitore invia i parametri e il ricevitore risponde confermando la ricezione. Ottenuta la conferma di ricezione del neutro da parte del ricevitore, il trasmettitore genera un tono sweepato crescente e si pone nella modalità operativa normale in cui il led lampeggia di rosso ad ogni invio di un pacchetto dati e il display mostra le posizioni dei joystick rispetto al neutro. Il led della ri-



L'antenna da noi usata per l'RX si collega con un cavetto al connettore siglato ANT.

Tabella 1 - Le segnalazioni dei led delle due unità (la x indica a quale ognuna si riferisce).

Lampeggio	Frequenza	Durata	TX	RX	Descrizione
Rosso-verde-speinto	1 Hz	15 sec		x	Il programma è partito correttamente.
Rosso-verde-rosso-verde-speinto	1 Hz	Infinita	x	x	Radiomodem non trovato.
Giallo	10 Hz	15sec	x	x	Il radiomodem sta per essere configurato
Rosso	Singolo lampeggio	Non applicabile	x	x	Il radiomodem ha ricevuto un comando o un dato sulla seriale.
Verde	Singolo lampeggio	Non applicabile	x	x	Il radiomodem ha elaborato il comando seriale e ha confermato l'elaborazione con una risposta via seriale
Rosso	1 Hz	0+120 sec		x	La ricevente sta chiedendo alla trasmittente la posizione di neutro. Se non viene ricevuto il neutro entro 120 secondi, il programma si pone in condizione di stand-by.
Rosso	0,5 Hz	120 sec		x	Stand-by; tutte le uscite sono in situazione di neutro (uscite proporzionali con impulso da 1,5 ms, uscite logiche a 1,2 ms oppure OFF). Dopo 120 secondi circa in questo stato, il programma riparte caricando la configurazione di default che prevede il canale RF 10 con potenza RF di -2 dBm.
Rosso	40 Hz	Legata al numero degli errori	x	x	Il trasmettitore sta inviando i comandi relativi agli 8 canali da controllare. Il ricevitore sta ricevendo i comandi relativi agli 8 canali da controllare, ma ha ricevuto alcuni dati errati e non li ha potuti processare.
Verde	40 Hz	Infinita		x	Il ricevitore sta ricevendo i comandi relativi agli 8 canali da controllare.

cevente lampeggia di verde ogni volta che riceve un pacchetto dati valido, e di rosso nel caso di dati non validi. Se avete collegato dei servocontrolli alle uscite da 1 a 4 e delle lampadine sulle uscite da 5 a 8, potete verificare che azionando i joystick e gli switch del TX, esse cambino di stato muovendo i servocontrolli e accendendo le lampadine. In questa situazione, il trasmettitore controlla il ricevitore esattamente come ci si aspetta, ma è necessario passare attraverso le varie fasi di configurazione. Vediamole una per una, facendo riferimento alla Fig. 1 che mostra i comandi associati ai vari switch. Notate che il pulsante del joystick di sinistra seleziona i vari menu di configurazione mentre quello del joystick di destra svolge la funzione di conferma o di selezione di un elemento all'interno di un menu. I deviatori associati ai canali logici, da sinistra a destra svolgono le funzioni di:

- toggle = cambio di stato di un parametro a due stati, come il reverse dei canali;
- exit = sale a schermata principale;
- up e down = incremento/decremento dei parametri che possono avere valori numerici differenti da 0 e 1.

Premendo ripetutamente il pulsante menu si passa nelle varie fasi di configurazione secondo l'ordine seguente:

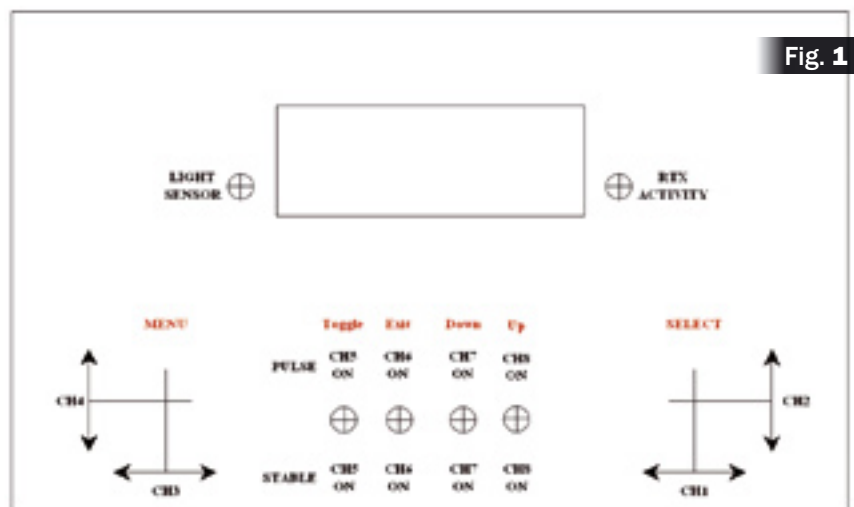
1. Trimming canali proporzionali,
 2. Gain canali proporzionali,
 3. Reverse di tutti i canali,
 4. Selezione del canale RF,
 5. Selezione della potenza RF,
 6. Verifica delle batterie del trasmettitore e del ricevitore,
 7. Impostazione del modo della retro-illuminazione del display,
 8. Abilitazione/disabilitazione dei toni di conferma,
 9. Invio della configurazione.
- Il comando exit permette di tornare alla schermata principale in qualsiasi momento senza salvare

in EEPROM i cambiamenti che rimangono comunque validi fino allo spegnimento.

Trimming dei canali proporzionali

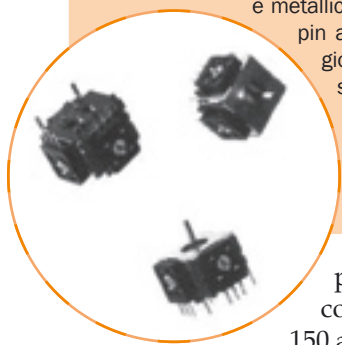
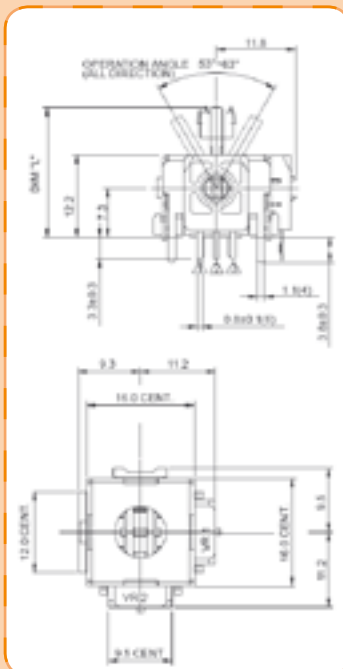
```
TC1<TC2 TC3 TC4
100 100 100 100
```

In questa impostazione, la schermata visibile qui sopra ricorda che tutti i canali proporzionali si trovano nella posizione centrale (valore 100). Premendo il joystick di destra (select) si sposta la freccia a fianco del canale proporzionale da regolare. I comandi up e down permettono di im-

**Fig. 1**

Quello utilizzato nel nostro progetto è prodotto da CTS Corporation (www.ctscorp.com) e fa parte della famiglia 252. Si tratta di un joystick a due assi, dotato di due potenziometri da 10 kohm (10% di tolleranza per il più preciso, 30% la tolleranza standard) ognuno, azionati da una leva che garantisce un angolo di rotazione di circa 60° in tutte le direzioni. Dispone di un pulsante normalmente aperto, azionabile premendo la leva di controllo. È garantito per un milione di cicli di rotazione, si presenta molto compatto (20x21x13 mm circa) e può funzionare da 0 a 70°C, con una tensione massima di 5 V in continua. La potenza massima dissipabile dagli elementi resistivi è di 0,0125 W a 50°C. Il corpo è metallico e dispone di quattro

Quello utilizzato nel nostro progetto è prodotto da CTS Corporation (www.ctscorp.com) e fa parte della famiglia 252. Si tratta di un joystick a due assi, dotato di due potenziometri da 10 kohm (10% di tolleranza per il più preciso, 30% la tolleranza standard) ognuno, azionati da una leva che garantisce un angolo di rotazione di circa 60° in tutte le direzioni. Dispone di un pulsante normalmente aperto, azionabile premendo la leva di controllo. È garantito per un milione di cicli di rotazione, si presenta molto compatto (20x21x13 mm circa) e può funzionare da 0 a 70°C, con una tensione massima di 5 V in continua. La potenza massima dissipabile dagli elementi resistivi è di 0,0125 W a 50°C. Il corpo è metallico e dispone di quattro



postare valori compresi tra 50 e 150 a step di un'unità, che corrispondono a $\pm 0,5$ ms rispetto all'impulso da 1,5 ms legato alla posizione teorica di neutro dei servocontrolli. Notate che ogni variazione del parametro di trimming genera un piccolo spostamento del relativo servocontrollo.

Gain dei canali proporzionali

```
%C1<%C2  %C3  %C4
100  100  100  100
```

Come vedete, la schermata è molto simile a quella della fase precedente: la sola differenza sta nel fatto che la T di trimming è stata sostituita dal simbolo %. Analogamente a quanto già visto, il joystick di destra seleziona il canale che si vuole modificare, mentre i comandi up e down permettono di variare il guadagno a passi di 5 unità, nel campo $20 \div 100$ %. Muovendo le leve dei joystick, potete verificare che lo spostamento dei servocontrolli è tanto minore quanto minore è il guadagno impostato.

Inversione dei canali

Nella schermata corrispondente, il canale da invertire va selezionato premendo ripetutamente il pulsante select. Ad ogni pressione, il contatore di canale (posto sotto la scritta INV) incrementa di un'unità (CH1, CH2..., CH8); raggiunto l'ottavo canale, una nuova pressione fa ripartire la sequenza da CH1.

Agendo sul comando toggle, si inverte il canale selezionato dal contatore. Quando il canale non è invertito, sotto il riferimento numerico relativo appare una D (Diretto); quando è invertito, appare una I (Invertito). Nel caso della figura qui sotto, è stato se-

```
INV=    12345678
CH1     II000000
```

lezionato il canale 1. Notate che i canali 1 e 2 sono invertiti, mentre quelli da 3 a 8 sono diretti.

Cambio canale e potenza RF

CANALE RF:10 >
434,57 MHz

La schermata del cambio canale è molto intuitiva: mostra il canale selezionato e la frequenza su cui



Ad Amendola (Puglia) stanno arrivando i nuovi aerei senza pilota *Predator B*, che l'Aeronautica militare userà per pattugliare il bacino

Radiomodelli ...da battaglia

mediterraneo. Il Predator B è lungo 11 m ed ha un'apertura alare di 20 m; pesa al decollo 4,5 tonnellate e supera i 440 km/h. Ha un'autonomia di volo di 24÷40 ore, ad una quota di 15.200 metri. Il suo armamento consiste in circa 1.360 kg di missili *Hellfire*, bombe a guida laser *Gbu-12 Paveway II* e *Gbu-38 Jdam* (Joint direct attack munition) a guida Gps. L'uso degli UAV sta però

creando qualche problema alla sicurezza dei voli: infatti questi aerei radiocomandati, seppure saranno fatti volare in rotte fuori dal traffico regolare (a 15 km d'altitudine) dovranno comunque decollare da scali militari alcuni dei quali sorgono in prossimità di grandi centri urbani e importanti aeroporti, col rischio che qualche aereo civile potrà trovarsi sulla propria rotta...

esso lavora. Il canale viene selezionato premendo ripetutamente il pulsante select; notate che nella parte in alto a destra del display appare una freccia destra seguita da una sinistra: la prima indica l'invio del canale al ricevitore e la seconda la conferma di ricezione da parte di quest'ultimo. La schermata di gestione della potenza di trasmissione è concettualmente identica alla precedente ed è illustrata di seguito.

```
POT. TRASM.: >
+10 dBm (MAX.)
```

La selezione viene sempre effettuata tramite il pulsante select e la freccia indica la trasmissione del nuovo valore seguita dalla conferma di ricezione.

Verifica stato batterie

In questa fase non sono previsti comandi di alcun tipo: di fatto, vengono visualizzate le tensioni di batteria di entrambi i dispositivi e la freccia in alto a destra, come nei due casi precedenti, indica la comunicazione bidirezionale che avviene tra trasmettitore e ricevitore. La schermata corrispondente (qui sotto) consente anche di effettuare la regolazione fine della

```
BATTERIE: >
TX:6.20 RX_5.29
```

lettura della tensione di batteria del trasmettitore: a tal proposito misurate con un tester la tensione di batteria applicata ai morsetti e, se necessario, ritocate il trimmer R12 in modo da far apparire a display il valore di tensione misurato. Notate che la tensione di batteria del trasmettitore viene visualizzata a step di 0,05 V (6,20 – 6,25 – 8,50...) mentre quella del ricevitore viene approssimata tramite un calcolo matematico e

pertanto la lettura varia a step di circa 0,04 o 0,05 Volt, a seconda del resto di una divisione intera effettuata dal firmware.

Retroilluminazione display

In questa impostazione, la pressione del pulsante select permette di scegliere il tipo di illuminazione desiderato. 1, 2 e 3 sono rispettivamente il livello basso, medio e alto. AUT regola automaticamente la retro-illuminazione in funzione della luce ambientale captata dalla fotoresistenza e della tensione di batteria. La freccia indica sempre l'impostazione prescelta: la schermata qui di seguito mostra che è stata

```
LUCE DISPLAY
AUT< 1 2 3
```

impostata la modalità automatica.

Tono di avviso

```
TONO AVVISO
Attivato
```

In questa impostazione è possibile abilitare o disabilitare l'emissione dei toni premendo il pulsante select. La disabilitazione viene confermata con l'emissione di una scala decrescente; l'abilitazione, invece è segnalata con una scala crescente: la schermata qui sopra mostra che il tono è attualmente attivato.

Invio configurazione

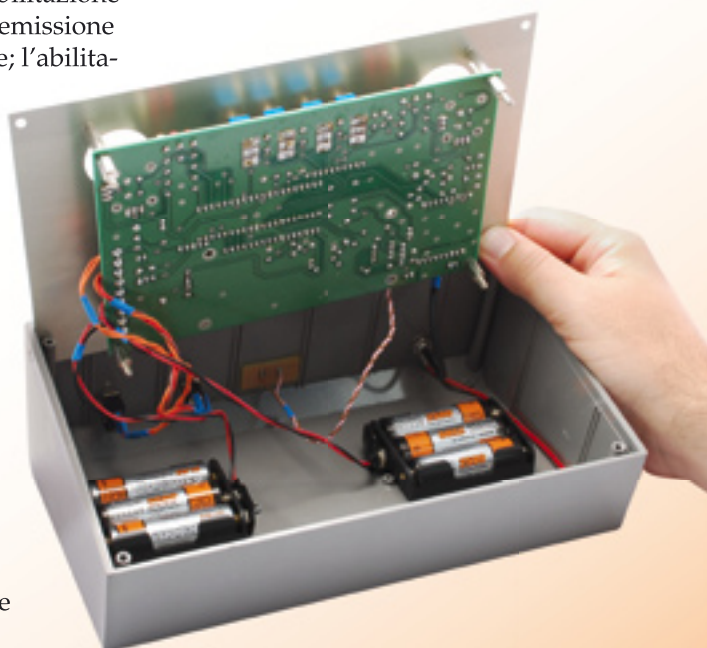
Dopo aver variato tutti i parametri, occorre memorizzarli sia nel ricevitore che nel trasmettitore. Una nuova pressione del pulsante menu fa apparire la schermata visibile qui di seguito, indicante che

```
INVIO CONFIG.
...ATTENDERE...
```

il trasmettitore sta salvando la nuova configurazione. Durante questa fase, esso interroga ripetutamente il ricevitore: quando questi risponde confermando di avere acquisito nuovi parametri e di avere effettuato il salvataggio, viene emesso il tono di conferma Big Beng, a conferma del buon esito del cambio configurazione. Ora sia il programma del trasmettitore che quello del ricevitore vengono riavviati caricando i nuovi parametri. In caso di mancata risposta da parte del ricevitore (cosa improbabile ma comunque possibile, ad esempio nel caso in cui le batterie si fossero scaricate in quel preciso istante) invece, appare la schermata visibile qui di seguito, che avverte

```
ERRORE! NESSUNA
RISPOSTA DA RX!
```

del fatto che qualcosa non è andato a buon fine nel ricevitore. Segue quindi la schermata illustrata qui sotto, con la quale il



CONFIGURAZIONE DEL SOLO TX

sistema comunica che la nuova configurazione è applicata solamente al trasmettitore.

SCHERMATE E SEGNALAZIONI ACUSTICHE RUNTIME

Esistono alcune schermate e segnalazioni acustiche non accessibili durante le normali fasi di utilizzo o configurazione: si tratta tipicamente di messaggi di errore, ricarica della configurazione predefinita o di allerta per situazioni potenzialmente a rischio, come ad esempio il radiomodem che non risponde più ai comandi, oppure la tensione di batteria troppo bassa.

Configurazione predefinita

Per ricaricare la configurazione predefinita nel trasmettitore, si deve accendere il trasmettitore mantenendo premuti i tasti di entrambi i joystick. In questo caso appare per cinque secondi la schermata visibile qui di seguito, a conferma del nuovo cambio di configurazione. Il program-

Conf. Predef. OK

ma, quindi, riparte secondo le modalità già viste. Per ricaricare la configurazione predefinita del ricevitore, accendete solo quest'ultimo (TX sempre spento) ed attendete per circa cinque minuti, verificando le sequenze di segnalazione viste durante la fase di collaudo.

Problemi al radiomodem

Durante la configurazione, abbiamo notato che in alcuni casi il radiomodem non rispondeva più ai comandi a causa di fluttuazioni sulla tensione di alimentazione in

condizioni di batteria quasi scarica; abbiamo pertanto realizzato una routine firmware che cerca di configurare il radiomodem per cinque volte.

Al sesto tentativo fallito, il display appare così:

ERRORE! RADIO
NON TROVATA!

La schermata è accompagnata dal lampeggio del led bicolore secondo il pattern di errore "rosso (100 ms) - verde (100 ms) - rosso (100 ms) - verde (100 ms) - spento (1000 ms)" che si ripete all'infinito: il programma, quindi, non può procedere. Lo stesso pattern viene riprodotto dal led del ricevitore qualora anch'esso non riesca a comunicare con il radiomodem in fase di configurazione all'avvio.

Segnalazioni di batteria

Durante l'avvio, il trasmettitore controlla lo stato di carica della batteria: se questa scende al di sotto del valore di 5,9 V, appare la schermata visibile qui sotto.

Batteria scarica
TX Bloccato!

Tale condizione impedisce al programma di procedere. Contemporaneamente viene emesso un tono a sirena con cadenza di due secondi.

Durante il normale utilizzo, invece, può capitare che la batteria sia inizialmente carica e si scarichi dopo un po' di tempo. In questo caso, appare una **B** nell'ultima casella di testo in alto a destra, a prescindere dalla schermata in cui ci si trova, accompagnata dall'emissione di un tono acutissimo di breve durata ogni trenta secondi circa. Le schermate seguenti mostrano due esempi di quel che appare sul display.

JC1 JC2 JC3 JC4B
I+ _ I+ _ I+ _ I+ _

LUCE DISPLAY B
AUT< 1 2 3

Bene, a questo punto avete una visione completa del sistema e del modo in cui utilizzarlo. Avrete, inoltre, certamente notato che, contrariamente a quanto facciamo di solito, non abbiamo parlato del firmware dei due dispositivi. La ragione è che si tratta di programmi molto complessi e un'analisi dettagliata necessiterebbe di troppo spazio. Coloro che vorranno realizzare ed utilizzare questo sistema hanno a disposizione in queste pagine un buon pacchetto d'informazioni; a chi volesse approfondire l'argomento firmware, diciamo che dal nostro sito www.elettronica.in.it è possibile scaricare il listato col quale programmare i microcontrollori utilizzati nei due dispositivi. ■



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono di facile reperibilità. I master dei due circuiti stampati possono essere scaricati gratuitamente dal sito della rivista così come il firmware per programmare i microcontrollori utilizzati nel trasmettitore e nel ricevitore. Il transceiver multicanale Aurel XTR-7020A-4 operante a 433 MHz è disponibile al prezzo di 38,00 Euro, IVA compresa.

Il materiale va richiesto a:

Futura Elettronica, Via Adige 11,
21013 Gallarate (VA)

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

Corso Wireless CPU



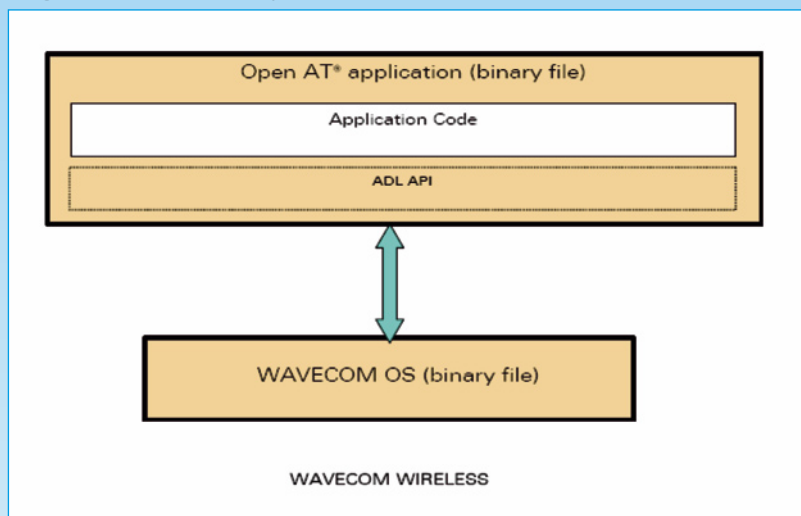
Introduzione alle librerie API di Open AT.

Entriamo nel vivo della programmazione imparando ad usare le librerie ad alto livello ADL, a gestire le funzioni e a dichiarare le variabili, mediante un esempio pratico.

dell'Ing.
NICOLA FRISO

In questa puntata del corso introduciamo le librerie API (Application Programming Interface) per lo sviluppo di applicazione con il sistema operativo Open AT di Wavecom. Per chi si fosse perso le puntate precedenti, ricordiamo che Open AT è un sistema operativo che comprende un set di librerie scritte in C che permettono di interfacciarsi ad alto livello con le periferiche hardware

delle Wireless CPU di Wavecom. Le funzioni messe a disposizione dalle librerie permettono altresì di gestire connessioni dati GSM e GPRS, invio e ricezione di SMS e molto altro ancora. L'insieme di librerie ad alto livello che Wavecom mette gratuitamente a disposizione dello sviluppatore prende il nome di ADL (Application Developer Layer); questo "strato" di software

Fig. 1 - Livelli software Open AT.

fornisce allo sviluppatore una serie di servizi per l'accesso a tutte le capacità e periferiche delle Wireless CPU. La **Fig. 1** tratta dall' "ADL User Guide for Open AT" di Wavecom, mostra chiaramente come il set di API ADL funzioni da interfaccia verso i servizi messi a disposizione dal sistema operativo sottostante, che gestisce a basso livello la WCPU. La nostra applicazione sarà dunque formata da codice C che richiama queste librerie ADL. Durante lo sviluppo del codice è fondamentale tener conto del fatto che abbiamo a che fare con un sistema operativo vero e proprio, per cui il flusso logico nello sviluppo del programma non è lineare e non è quello tipico a cui molti sono abituati quando si programma con i microcontrollori; sul nostro codice non troverete una vera e propria funzione main con un ciclo infinito tipo **while(1)** o **for(;;)** al cui interno si richiamano le diverse funzioni. C'è invece un unico punto di ingresso del programma (entry-point) chiamato "adl_main", dove eseguire le eventuali inizializzazioni, il quale viene eseguito una volta sola all'avvio della applicazione. Tutto il resto del codice è gestito ad eventi ed a "task", i quali, nella terminologia dei sistemi operativi, sono pezzi di codice che svolgono una specifica funzione e che vengono eseguiti come se fossero in parallelo, alternandosi uno all'altro. Alcuni task più prioritari possono interromperne altri meno prioritari, ma per l'utente è come se il tutto fosse eseguito allo stesso momento. Fortunatamente con le librerie ADL, sebbene

sia possibile, non è necessario gestire a basso livello i task e le loro priorità; tutto è così molto più semplice di quanto possa sembrare.

Ogni funzione o periferica che si vuole abilitare (ad esempio Timer, ingressi/uscite, porta seriale, interrupt ecc...) deve essere prima attivata, nel gergo Open AT, si dice che deve essere "sottoscritta"; ecco perché per ogni interfaccia che si vuole abilitare c'è la sua funzione di "subscribe".

Dopo essere stata utilizzata, ogni funzione può essere disattivata, richiamando, allo scopo, la corrispondente funzione di de-sottoscrizione "unsubscribe". Ad ogni sottoscrizione il sistema assegna anche un identificativo ID dell'evento, che è un numero intero assegnato dal sistema nel momento della sottoscrizione e che identifica l'evento quando si scatena. Nella sottoscrizione dell'evento occorre anche passare come parametro anche il puntatore alla funzione che gestirà l'evento, quando si scatterà.

In pratica, una volta che si è eseguita la sottoscrizione, quando si verifica l'evento il flusso del programma salta automaticamente dentro la funzione di gestione dell'evento, al cui interno lo sviluppatore avrà inserito il proprio codice di gestione. Una volta entrati in questa filosofia di programmazione, diventa molto semplice e veloce sviluppare le nostre appli-



cazioni.

Un esempio di questo concetto di programmazione lo abbiamo già visto nella seconda puntata del corso, con il programma "Hello World". In questo esempio abbiamo attivato un Timer con la funzione **adl_tmrSubscribe** richiamata all'interno dell'entry-point **adl_main**, che deve essere sempre presente

all'interno delle nostre applicazioni (M2M Studio lo crea automaticamente quando si avvia un nuovo progetto). A riguardo guardate il **Listato 1** che trovate qui accanto.

Il prototipo della funzione **adl_tmrSubscribe** è il seguente:

adl_tmr_t *adl_tmrSubscribe(bool bCyclic, u32 TimerValue, adl_tmrType_e TimerType, adl_tmrHandler_t Timerhdl)

Qui di seguito trovate la descrizione dei parametri della funzione.

bool bCyclic = valore booleano;

TRUE corrisponde ad un timer ciclico, mentre **FALSE** attiva un timer di tipo "one shot" in cui allo scadere del tempo impostato il timer si ferma e termina automaticamente.

u32 TimerValue = valore temporale impostato (in multipli di 100 ms oppure in unità di "tick").

adl_tmrType_e TimerType = tipo di timer, i valori ammessi sono due:

- **ADL_TMR_TYPE_100MS** ; base del tempo del timer pari a 100ms;
- **ADL_TMR_TYPE_TICK**; base del tempo del timer pari a un tick (18,5 ms).

Notate che il tick permette di avere temporizzazioni più precise.

Esistono anche delle funzioni di conversione da ms a tick e viceversa (vedere, sulla "ADL UserGuide for Open AT" di Wavecom, le macro-funzioni) che sono, rispettivamente:

ADL_TMR_MS_TO_TICK(MsT), ADL_TMR_100MS_TO_TICK(MsT)

Per la conversione da millisecondi a tick e: **ADL_TMR_S_TO_TICK(SecT), ADL_TMR_MN_TO_TICK(MnT)**

Per la conversione da tick a millisecondi. L'ultimo parametro della funzione da analizzare è **adl_tmrHandler_t Timerhdl**, che è il puntatore alla funzione che gestisce il timer.

Quindi alla funzione **adl_tmrSubscribe** viene passato il primo parametro **TRUE** per impostare un timer ciclico che si attiva ogni 10x100

Listato 1 - Codice dell'applicazione "Hello World".

```

/*****
/* File      : Hello_World.c
/*-----
/* Object    : Customer application
/*-----
/* contents  : Customer main procedures
/*-----
/* Change    :
/*-----
*****/

#include "adl_global.h"

/*****
/* Mandatory variables
/*-----
/* wm_apmCustomStackSize
/*-----
*****/
const u16 wm_apmCustomStackSize = 1024*3;

/*****
/* Function   : HelloWorld_TimerHandler
/*-----
/* Object     : Timer handler
/*-----
/* Variable Name |IN |OUT|GLB| Utilisation
/*-----
/* ID            |  |  |  | Timer ID
/*-----
/* Context       |  |  |  | Context
/*-----
*****/
void HelloWorld_TimerHandler ( u8 ID, void * Context )
{
    /* Hello World */
    TRACE (( 1, "Embedded : Hello World" ));
    adl_atSendResponse ( ADL_AT_UNSP, "\r\nHello World from Open-AT\r\n" );
}

/*****
/* Function   : adl_main
/*-----
/* Object     : Customer application initialisation
/*-----
/* Variable Name |IN |OUT|GLB| Utilisation
/*-----
/* InitType     |  |  |  | Application start mode reason
/*-----
*****/
void adl_main ( adl_InitType_e InitType )
{
    TRACE (( 1, "Embedded : Appli Init" ));

    /* Set 1s cyclic timer */
    adl_tmrSubscribe ( TRUE, 10, ADL_TMR_TYPE_100MS, HelloWorld_TimerHandler );
}

```

ms = 10s e che ad ogni attivazione "salta" alla funzione **HelloWorld_TimerHandler**.

Vediamo ora un esempio un po' più complesso: salviamo il nostro progetto "Hello_World" con un nuovo nome "Allarme" e andiamo a realizzare un'applicazione che alla pressione di un tasto genera un allarme. Utilizziamo un pin della tastiera "keypad" per rilevare la

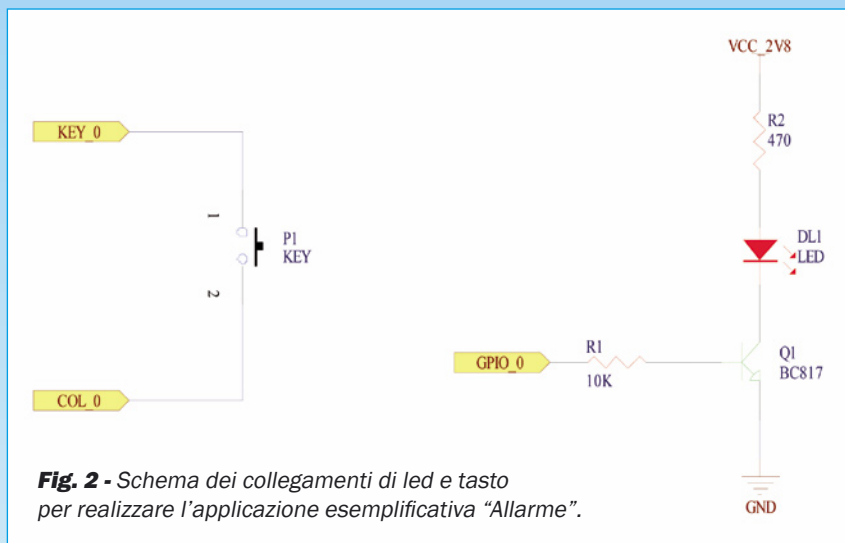


Fig. 2 - Schema dei collegamenti di led e tasto per realizzare l'applicazione esemplificativa "Allarme".

pressione del tasto e un pin di GPIO, configurato come uscita, per pilotare il led, il quale deve lampeggiare per 10 secondi e poi smettere; inoltre alla pressione del tasto generiamo un SMS di allarme verso un numero preimpostato.

Tutto questo ci permetterà di analizzare nel dettaglio, oltre ai timer anche le funzioni che gestiscono la tastiera, i pin di GPIO e quelle relative all'invio degli SMS.

Per realizzare questo esperimento possiamo utilizzare la demoboard EVKAUGUSTO per il modulo Augusto di Shitek Technology; in alternativa, se non disponiamo dell'EVKAUGUSTO possiamo collegare il modulo Augusto ai pin GPIO0, KEY0, COL0 secondo lo schema riportato nella Fig. 2.

Analizziamo riga per riga il codice dell'applicazione: nella prima parte troviamo l'includa "adl_global.h" che deve essere sempre presente per caricare tutte le definizioni, le costanti predefinite e i prototipi della libreria ADL. A seguire troviamo una variabile di sistema che alloca la memoria per lo stack delle chiamate a funzione. Da questo punto in avanti inseriamo il codice della nostra applicazione "Allarme".

Definiamo prima di tutto quanti pin di GPIO intendiamo usare come uscita; in questo caso si tratta di un solo pin: GPIO0. Per fare ciò dichiariamo il tipo di struttura predefinita `adl_ioDefs_t`, tramite la quale è possibile definire la direzione di tutti i pin di GPIO e il loro stato iniziale.

In questo caso abbiamo definito un solo pin di uscita GPIO0 (`ADL_IO_GPIO | 0`); la

direzione di uscita è fissata dalla costante predefinita `ADL_IO_DIR_OUT` e lo stato iniziale è basso (`ADL_IO_LEV_LOW`). Così facendo, nella struttura `Gpio_Config` di tipo `adl_ioDefs_t` abbiamo inserito una maschera di 16 bit, mettendo in OR le varie costanti predefinite; la maschera, in questo caso, dice al sistema operativo che al momento della sottoscrizione dei GPIO deve settare il GPIO0 come uscita a livello basso.

Utilizzando la stessa struttura potevamo definire come uscite o ingressi anche altri pin: se, ad esempio, volevamo definire anche il pin GPIO1 come uscita a livello alto e il GPIO2 come ingresso, bastava aggiungere alla struttura le maschere di bit relative a questi pin, ad esempio:

```
#define NUM_GPIO 3 // number of GPIOs sets

static adl_ioDefs_t Gpio_Config[NUM_GPIO] =
{
    {ADL_IO_GPIO | 0 | ADL_IO_DIR_OUT | ADL_IO_LEV_LOW}, // set GPIO0 as output low
    {ADL_IO_GPIO | 1 | ADL_IO_DIR_OUT | ADL_IO_LEV_HIGH}, // set GPIO1 as output high
    {ADL_IO_GPIO | 2 | ADL_IO_DIR_IN} // set GPIO2 as input
};
```

La riga successiva, ossia "s32 myGpioOut_Handle" dichiara un intero a 32 bit con segno (vedere il riquadro **Tipi di variabili**) che sarà il numero identificativo (detto "handle") il quale servirà alle funzioni che gestiranno questa uscita. In modo analogo, nelle due righe successive definiamo un handle associato agli SMS ed uno associato al timer che farà lampeggiare il led.

"tim_count" è una variabile globale di tipo intero a 8 bit senza segno, che utilizzeremo come contatore di numero di volte che viene richiamata la funzione che gestisce il timer.

Le ultime due righe della parte di dichiarazione delle variabili definiscono due stringhe ASCII contenenti il numero di telefono cui inviare l'SMS di allarme e il testo dell'SMS.

Stabilite quali sono le variabili e strutture globali utilizzate, andiamo a vedere il contenuto della funzione di ingresso **adl_main**: qui troviamo l'inizializzazione della variabile contatore **tim_count** e, a seguire, le chiamate alle funzioni utili per sottoscrivere e attivare il pin di uscita GPIO0, per attivare la tastiera e per inizializzare la SIM del modulo. Vediamole nel dettaglio.

La funzione **adl_ioSubscribe** restituisce un intero maggiore-uguale a zero se la sottoscrizione dei pin di GPIO passati come parametro riesce. Se ciò non avviene, può restituire vari tipi di codice di errore sul cui significato per ora possiamo soprassedere, rimandando alla guida ADL di Wavecom. I parametri passati sono il numero di pin di GPIO da configurare, la struttura di definizione dei pin che abbiamo visto sopra, più altri tre parametri, impostati a zero, che servono qualora si fossero settati dei GPIO come ingressi e si volesse attivare un polling automatico per interrogarne lo stato. Vedremo più avanti in un capitolo del corso dedicato agli I/O anche questa comoda funzione. Per ora lasciamo questi parametri a zero.

La riga successiva è una chiamata alla funzione **adl_atCmdCreate** che serve per richiamare dall'interno della nostra applicazione qualsiasi comando AT. In questo caso abbiamo inoltrato il comando "AT+CMER=,1", il quale abilita la generazione di un evento ogni volta che si preme un tasto sulla tastiera. L'evento viene segnalato dalla generazione sulla seriale di una stringa del tipo "+CKEV:

Listato 2 - Codice dell'applicazione "Allarme".

```

/*****
/* File      : Allarme.c
/*-----
/* Object    : Customer application
/*-----
/* contents  : Customer main procedures
/*-----
/* Change   :
*****/

*
* Date      | Author | Revision | Description
*-----+-----+-----+-----
* 29.09.09 | NicolaF | 0.1      | * Tested version
*-----+-----+-----+-----
*/

#include "adl_global.h"

/*****
/* Mandatory variables
/*-----
/* wm_apmCustomStackSize
/*-----
*****/
const u16 wm_apmCustomStackSize = 1024*3*3;

#define NUM_GPIO 1          // number of GPIOs sets

static adl_ioDefs_t Gpio_Config[NUM_GPIO] = {

    {ADL_IO_GPIO | 0 | ADL_IO_DIR_OUT | ADL_IO_LEV_LOW}    // set GPIO0 as output low

};

static s32 myGpioOut_Handle;          // handle to GPIO0 out
static s8 sms_handle;                 // handle for sms event
static adl_tmr_t* tim_handle;         // timer handle for led
static u8 tim_count;                  // timer tim_counter

static ascii phone_number[] = "+393289531658";           // phone number to send sms
static ascii sms_text[] = "Allarme";                     // text of sms

/*****
/* Function  : Led_TimerHandler
/*-----
/* Object    : Timer handler
/*-----
/* Variable Name | IN | OUT | GLB | Utilisation
/*-----+-----+-----+-----+-----
/* ID            |   |   |   | Timer ID
/*-----+-----+-----+-----+-----
/* Context       |   |   |   | Context
/*-----+-----+-----+-----+-----
*****/
void Led_TimerHandler ( u8 ID, void * Context )
{
    adl_ioDefs_t Gpio_to_write = ADL_IO_GPIO | 0 ;

    tim_count++;

    // Read output and toggle led
    if(adl_ioReadSingle( myGpioOut_Handle,&Gpio_to_write))
        // Reset output
        adl_ioWriteSingle ( myGpioOut_Handle, &Gpio_to_write, FALSE );
    else
        // Set output
        adl_ioWriteSingle ( myGpioOut_Handle, &Gpio_to_write , TRUE );

    if(tim_count > 20)    // Stop timer after 10s
    {
        // Reset output
        adl_ioWriteSingle ( myGpioOut_Handle, &Gpio_to_write , FALSE );
        // init timer counter
        tim_count = 0;
        // Stop timer led
        adl_tmrUnSubscribe ( tim_handle, Led_TimerHandler, ADL_TMR_TYPE_100MS);
    }
}

/*****
/* Function  : CKEV_Handler
/*-----
/* Object    : CKEV_Handler
/*-----
/* Variable Name | IN | OUT | GLB | Utilisation
/*-----+-----+-----+-----+-----
/* gpio_handle   |   |   |   |
/*-----+-----+-----+-----+-----
*****/

```

```

/*-----+-----+-----+-----*/
/* gpio_result | | | |
*/
/*-----+-----+-----+-----*/
/*****
bool CKEV_Handler(adl_atUnsolicited_t *paras)
{
    ascii key_nb[3];    // key number
    ascii key_st[3];    // key status (ON=1 OFF=0)

    u8 kn,ks;

    TRACE (( 1, "CKEV" ));

    wm_memset(&key_nb,0x00,sizeof(key_nb));
    wm_memset(&key_st,0x00,sizeof(key_st));

    wm_strGetParameterString ( key_nb, paras->StrData, 1 );
    kn = wm_atoi(key_nb); // key number
    TRACE (( 1, "KEY_NB=%d",kn));

    wm_strGetParameterString ( key_st, paras->StrData, 2 );
    ks = wm_atoi(key_st); // key status
    TRACE (( 1, "KEY_ST=%d",ks));

    if(!tim_count)      // active alarm only if previus alarm is terminated
    {
        //if(!wm_strcmp(res,"0"))
        if(kn==0 && ks==0) // key number 0 release
        {
            // Send sms
            TRACE (( 1, "Alarm activated...send sms" ));
            adl_smsSend (sms_handle, phone_number, sms_text, ADL_SMS_MODE_TEXT);

            // Set 500ms cyclic led timer
            TRACE (( 1, "Start led timer" ));
            tim_handle = adl_tmrSubscribe ( TRUE, 5, ADL_TMR_TYPE_100MS, Led_TimerHandler );
        }
    }

    return TRUE;
}
/*****
/* Function : SMS_Control_f
*/
/*-----+-----+-----+-----*/
/* Object : SMS_Control_f
*/
/*-----+-----+-----+-----*/
/* Variable Name |IN |OUT|GLB| Utilisation
*/
/* Event | | | |
*/
/* Nb | | | |
*/
/*-----+-----+-----+-----*/
/*****
void SMS_Control_f(u8 Event,u16 Nb)
{
    if(Event == ADL_SMS_EVENT_SENDING_OK)
    {
        TRACE((1,"SMS SEND OK"));
        adl_atSendResponse (ADL_AT_RSP, "\r\nSMS SEND OK\r\n");
    }
    if(Event == ADL_SMS_EVENT_SENDING_ERROR) // can be here if no credit
    {
        TRACE((1,"SMS SENDING ERROR"));
        adl_atSendResponse (ADL_AT_RSP, "\r\nSMS SENDING ERROR\r\n");
    }
}
/*****
/* Function : SMS_Handler_f
*/
/*-----+-----+-----+-----*/
/* Object : SMS_Handler_f
*/
/*-----+-----+-----+-----*/
/* Variable Name |IN |OUT|GLB| Utilisation
*/
/* SmsTel | | X | |
*/
/* SmsTimeOrLength | | X | |
*/
/* SmsText | | | X | |
*/
/*-----+-----+-----+-----*/
/*****
bool SMS_Handler_f(ascii *SmsTel, ascii *SmsTimeOrLength, ascii *SmsText)
{
    TRACE((1,"SMS ricevuto:"));
    TRACE((1,SmsText));
    TRACE((1,SmsTel));
    return TRUE;
}
/*****

```

<key>,<press>" in cui <key> è il codice del tasto premuto e <press> è lo stato del tasto (1 premuto, 0 rilasciato). Occorre quindi intercettare questa stringa per capire quando il tasto viene premuto. La chiamata:

`adl_atUnSoSubscribe("+CKEV:",(adl_atUnSoHandler_t)CKEV_Handler);` fa proprio questo. Per l'esattezza, sottoscrive un evento di tipo "Unsolicited" che riguarda la tastiera: ogni volta che viene generata una stringa "+CKEV:" il flusso del programma salta alla funzione (gestita dall'utente) "CKEV_Handler" che deve andare a leggere i successivi due parametri <key> e <press> per capire quale tasto è stato premuto. Normalmente tutti gli eventi "Unsolicited" (ossia indesiderati...) per impostazione predefinita, sono disattivati. L'ultima riga del main sottoscrive i servizi relativi alla SIM che comprendono l'invio e la ricezione degli SMS, ma anche la gestione del PIN; in questo caso non abbiamo volutamente gestito il PIN per non complicare il codice. Il programma quindi funziona solo se si disabilita il PIN della SIM prima di inserire quest'ultima nel modulo Augusto. La funzione **appSimHandler** restituisce delle informazioni circa lo stato della SIM.

Se è inizializzata e pronta all'uso, restituisce `ADL_SIM_STATE_FULL_INIT`, mentre se è in attesa di PIN ritorna `ADL_SIM_STATE_PIN_WAIT`; se il pin è errato, la funzione restituisce `ADL_SIM_STATE_PIN_ERROR`, e infine, se la SIM è rimossa o inserita, restituisce rispettivamente `ADL_SIM_EVENT_REMOVED` e `ADL_SIM_EVENT_INSERTED`.

Quando la SIM è inizializzata, sottoscriviamo il servizio SMS in modo da essere in grado in seguito di ricevere e trasmettere SMS.

La riga di codice seguente:

`sms_handle = adl_smsSubscribe(SMS_Handler_f,SMS_Control_f,ADL_SMS_MODE_TEXT);`

sottoscrive gli SMS. Durante la sottoscrizione vengono passati tre parametri, che sono:

SMS_Handler_f = puntatore ad una

funzione gestita dall'utente in cui l'applicazione salta quando riceve un SMS;

SMS_Control_f = puntatore ad una funzione gestita dall'utente in cui l'applicazione salta quando ha inviato un SMS; **ADL_SMS_MODE_TEXT** = indica il modo in cui gli sms devono essere ricevuti o inviati.

In quest'ultimo caso nel parametro è stato impostato il modo testo; è anche possibile scegliere il modo **ADL_SMS_MODE_PDU**, ricordando che PDU è un protocollo binario nativo del sistema GSM ed è con questo protocollo che gli SMS vengono scambiati nella rete. Per semplicità utilizzeremo sempre il modo testo. Tornando al flusso del programma, dopo aver eseguito il main, la SIM si è inizializzata e l'applicazione è in attesa di un tasto premuto. Alla pressione del tasto si esegue la funzione "CKEV_Handler".

In questa funzione si analizzano i parametri <key> e <press> con la funzione di libreria **wm_strGetParameterString** che copia il primo (<key>) e secondo (<press>) parametro nell'array "key_nb" e "key_st" rispettivamente. Il primo parametro è il numero del tasto premuto mentre il secondo è lo stato zero o uno, del tasto. Prima di generare l'allarme verifichiamo lo stato del timer di allarme. Se il contatore è uguale a zero, cioè non ci sono altri allarmi precedenti in corso si genera l'sms di allarme e si fa partire il timer per il lampeggio del led. L'allarme viene generato quando lo stato "key_st" del tasto è zero, che significa far partire l'allarme quando il tasto viene rilasciato. Quindi si genera l'SMS di allarme e si fa partire il timer per il lampeggio del led. La funzione **adl_sms_Send** invia l'SMS; quando il messaggio è stato realmente inoltrato alla rete GSM, il flusso del programma salta alla funzione **SMS_Control_f** che avevamo definito in fase di sottoscrizione degli SMS. Se tutto è andato a buon fine si verifica l'evento **ADL_SMS_EVENT_SENDING_OK** e il programma genera sulla porta seriale la stringa "SMS SEND OK".

Listato 2 ... continuazione

```

/* Function : appSimHandler */
/* Object : Handler to receive SIM related events */
/* Return : */
/* */
/*-----*/
/* Variable Name |IN |OUT|GLB| Utilisation */
/*-----*/
/* | | | | */
/*-----*/
/*****
void appSimHandler (u8 Event)
{
    TRACE((1,"APP SIM EVENT HANDLER"));
    switch (Event)
    {
        case ADL_SIM_STATE_FULL_INIT:

            TRACE (( 1, "ADL_SIM_STATE_FULL_INIT"));

            // SMS subscribe
            sms_handle = adl_smsSubscribe(SMS_Handler_f,SMS_Control_f,ADL_SMS_MODE_
TEXT);
            TRACE((1,"sms_handle= %d",sms_handle));
            break;

        case ADL_SIM_STATE_PIN_WAIT:
            TRACE((1,"\\r\\nADL_SIM_STATE_PIN_WAIT\\r\\n"));
            adl_atSendResponse (ADL_AT_RSP, "\\r\\nADL_SIM_STATE_PIN_WAIT\\r\\n");
            break;

        case ADL_SIM_STATE_PIN_ERROR:
            TRACE((1, "\\r\\nADL_SIM_STATE_PIN_ERROR\\r\\n"));
            adl_atSendResponse (ADL_AT_RSP, "\\r\\nADL_SIM_STATE_PIN_ERROR\\r\\n");
            break;

        default:
            TRACE (( 1, "SIM Event = %d", Event ));
            break;
    }
    return;
}
/*****
/* Function : adl_at_CMER_response_handler */
/*-----*/
/* Object : adl_at_CMER_response_handler */
/*-----*/
/* Variable Name |IN |OUT|GLB| Utilisation */
/*-----*/
/* gpio_handle | | | | */
/*-----*/
/* gpio_result | | | | */
/*-----*/
/*****
void adl_at_CMER_response_handler(adl_atResponse_t *at_str_pars)
{
    // nothing to do
    TRACE (( 1, "CMER" ));
}
/*****
/* Function : adl_main */
/*-----*/
/* Object : Customer application initialisation */
/*-----*/
/* Variable Name |IN |OUT|GLB| Utilisation */
/*-----*/
/* InitType | | | | Application start mode reason */
/*-----*/
/*****
void adl_main ( adl_InitType_e InitType )
{
    s8 res;

    //adl_ioDefs_t Gpio_to_write1 = ADL_IO_GPIO | 0 ;

    TRACE (( 1, "Embedded : Appli Init" ));

    tim_count = 0; // init timer counter

    // Subscribe GPIO0 output
    myGpioOut_Handle = adl_ioSubscribe(NUM_GPIO,Gpio_Config,0,0,0);
    TRACE (( 1, "handler returns %d", myGpioOut_Handle ));

    // Subscribe keypad
    adl_atCmdCreate("AT+CMER=,1",FALSE,(adl_atRspHandler_t)adl_at_CMER_response_
handler,"*",NULL);

    // Subscribe the '+CKEV: unsolicited response
    adl_atUnSoSubscribe("+CKEV: ",(adl_atUnSoHandler_t)CKEV_Handler);

    // Subscribe to SIM service to send and receive sms
    res = adl_simSubscribe ( (adl_simHdlr_f) appSimHandler, NULL);
}

```

Tipi di variabili

I tipi base sono definiti nel file di sistema **wm_types.h**

u8	8 bit senza segno (corrisponde al tipo "unsigned char" del C)
s8	8 bit con segno (corrisponde al tipo "signed char" del C)
u16	16 bit senza segno (corrisponde al tipo "unsigned short" del C)
s16	16 bit con segno (corrisponde al tipo "signed short" del C)
u32	32 bit senza segno (corrisponde al tipo "unsigned int" del C)
s32	32 bit con segno (corrisponde al tipo "signed int" del C)
u64	64 bit senza segno (corrisponde al tipo "unsigned long" del C)
s64	64 bit con segno (corrisponde al tipo "signed long" del C)
bool	booleano true o false
ascii	carattere 8 bit (corrisponde al tipo "char" del C)

Nel frattempo il timer per il lampeggio del led gestito dalla funzione **Led_TimerHandler** è partito e, ad ogni richiamo della funzione, commuta lo stato del pin di uscita GPIO_0. Per farlo commutare prima, leggiamo lo stato del pin con la funzione **adl_ioReadSingle**; se è ad uno lo mettiamo a zero mediante la funzione **adl_ioWriteSingle**, viceversa se lo stato era basso. Si noti che alle funzioni di Read e Write dobbiamo passare l'handle "myGpioOut_Handle" che avevamo ottenuto

sul main (in fase di sottoscrizione del pin di GPIO) e il puntatore alla struttura "adl_ioDefs_t Gpio_to_write", cui abbiamo assegnato il valore "ADL_IO_GPIO | 0", che corrisponde al GPIO_0. Il timer incrementa il contatore **tim_count** ad ogni chiamata (ogni 500 ms); quando il contatore arriva a 20 (pari a 10 s) il timer viene disattivato con la funzione **adl_tmrUnSubscribe** e il pin di GPIO_0 viene portato a

zero logico. Si noti come anche per la funzione di de-sottoscrizione del timer **adl_tmrUnSubscribe** abbiamo dovuto passare ad essa l'handle ricavato in fase di sottoscrizione. Bene, con questo abbiamo finito; ora non vi resta che compilare il progetto e attivare il debug sull'M2M Studio (vedere la seconda puntata del corso) per testare il codice. Nel nostro sito www.elettronica.in trovate il file "allarme.zip" con il progetto per M2M Studio e il codice sorgente dell'applicazione.

Modulo GSM-GPRS con CPU integrata

Tutti i prezzi si intendono IVA inclusa.



SWMAUGUSTO
€ 89,00

SWMAUGUSTO è un modulo embedded GSM/GPRS che integra una CPU Wireless Wavecom serie WMP con a bordo 11 GPIO, 1 ADC, 2 ingressi di interrupt, ingressi keyboard, 2 UART, 1 SPI, 1 USB, RTC. Permette di eseguire un codice proprietario all'interno della CPU serie WMP. Può essere utilizzato senza microcontrollore esterno, risparmiando

anche sui costi legati allo sviluppo di librerie software per la gestione dei comandi AT e di tutte le periferiche a bordo del microcontrollore le quali risultano essere già integrate nella CPU serie WMP. L'ambiente di sviluppo OPEN-AT Software Suite 2.0 di Wavecom (gratuito) mette a disposizione un sistema operativo multitasking completo di tutte le librerie ed esempi già pronti per gestire connessioni dati GSM, GPRS, SMS, voce, vivavoce con cancellatore d'eco integrato, input/output, ADC, keyboard e tutte le periferiche della CPU serie WMP.

Principali caratteristiche:
CPU: Wavecom WMP50, microcontrollore ARM 9, 32 bit, 26 MHz; memoria: 4 MB Flash, 2 MB RAM; interfacce: 11 GPIO uso generico, 2 interrupt configurabili, 1 ADC 10 bit, 5x5 keypad con antirimbato integrato, 2 canali audio separati con DTMF integrato; interfacce seriali: 2 UART, connettore programmazione/debug (con Uart 1), 1 USB SLAVE, 1 SPI/I2C, JTAG; aliment.: 3,4 ÷ 4,2 Vdc.

DEMOBOARD per modulo GSM-GPRS AUGUSTO

Permette di programmare facilmente il modulo SWMAUGUSTO e di testarne tutte le periferiche. Dispone di porte COM, USB, RJ45, ingresso audio e circuito di ricarica batteria. Sono presenti inoltre trimmer per testare gli ingressi analogici, led per testare gli I/O, due pulsanti di interrupt e una

tastiera a matrice 3x2 (costituita da 6 pulsanti). Tutti i pin del modulo AUGUSTO sono riportati all'esterno su comodi pin strip passo 2,54mm. La confezione NON comprende il modulo SWMAUGUSTO.

DEMOBOARD SWM

€ 295,00



FUTURA ELETTRONICA

Via Adige, 11 - 21013 GALLARATE (VA)
Tel. 0331/799775 - Fax 0331/778112
www.futurashop.it

Maggiori informazioni e caratteristiche tecniche sono disponibili sul sito www.futurashop.it tramite il quale è anche possibile effettuare acquisti on-line.