

Mensile di elettronica applicata, attualità scientifica, novità tecnologiche.

€ 5,00

Elettronica In

oltre l'elettronica

Anno XV - n. 140
Settembre 2009

www.elettronicaonline.it

MEMS
Tecnologia
e applicazioni



NOVITÀ ASSOLUTA

Un localizzatore senza GPS

- Transponder seriale
- Controllo motori DC 2 canali
- Gun Clock, zittiscilo sparando
- Tagliaerba solare, la messa a punto
- Corso FPGA

ESCLUSIVO!
Corso Wireless CPU

LA SVEGLIA? SE NON TACE LE SPARO!

Originale sveglia elettronica con display a LED: quando suona, l'unico modo per farla smettere è alzarsi e sparare contro il suo bersaglio con un'apposita pistola a laser.



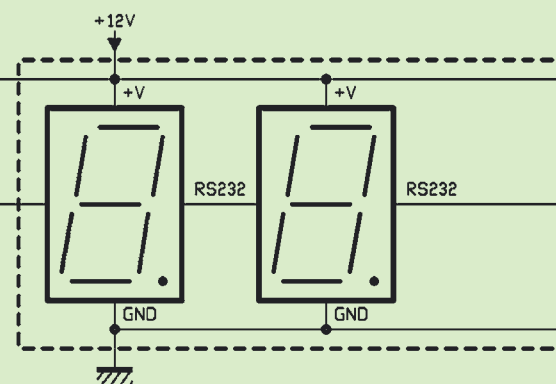
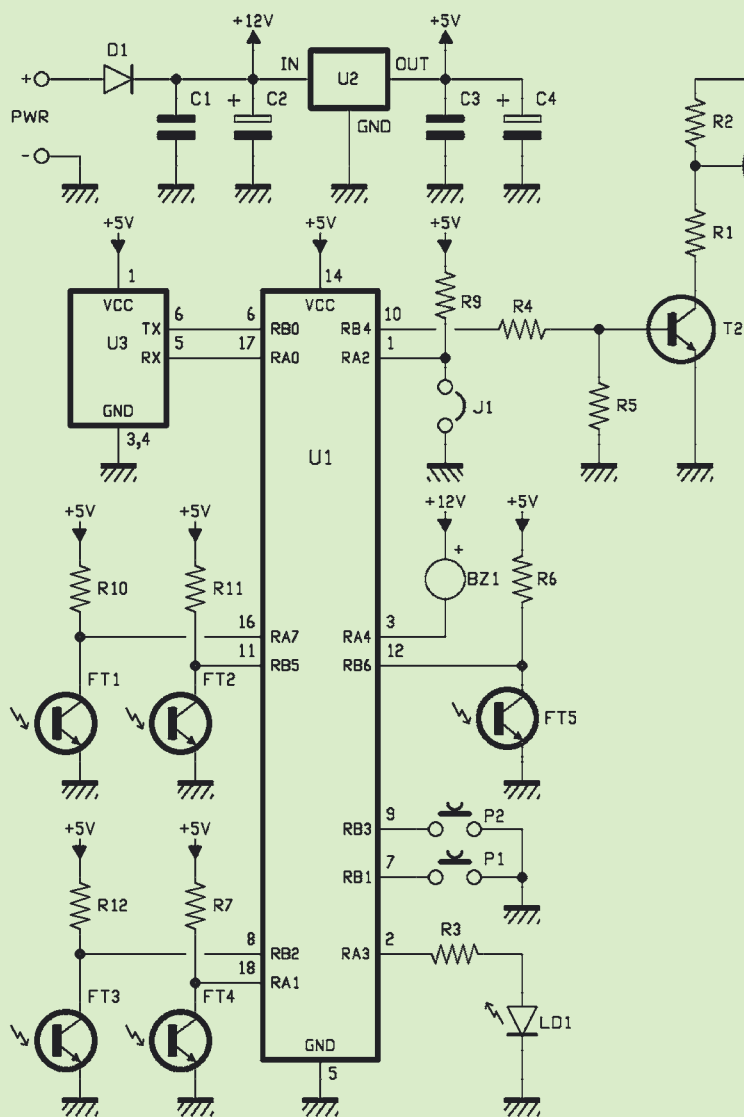
BEEP!
BEEP!
BEEP!
BEEP!
BEEP!
BEEP!
BEEP!



..... di RICCARDO LAZZATI

Quante volte la mattina, specie dopo una notte quasi insonne e di fronte alla prospettiva di una dura giornata di lavoro con le palpebre socchiuse, vi è venuta voglia di lanciare una ciabatta contro la sveglia o di scagliare la sveglia contro il muro? Non diteci che non avete mai avuto questa tentazione... Ebbene, se almeno una volta nella vita avete desiderato far smettere di suonare la sveglia senza mezze misure, vi diamo la possibilità

di vendicarvi per i tanti sogni interrotti bruscamente e per le troppe volte che il prezioso apparecchio vi ha ricordato, suo malgrado, che dovete alzarvi e correre al lavoro; in queste pagine vi proponiamo una sveglia alla quale potete anche sparare! Sì, avete capito bene! Naturalmente non con una pistola tradizionale, altrimenti sarebbero guai seri, dato che neppure *l'Ispettore Callaghan*, appena svegliato, avrebbe una mira infallibile!!!



cos'altro dire? Già, qualcos'altro ancora possiamo dirlo: oltre che un orologio con funzione di sveglia, il nostro dispositivo può anche essere un gioco elettronico; infatti, in un'opportuna modalità permette di realizzare un tiro a segno semplice e funzionale, che attribuisce un punteggio ogni volta che con la pistola si colpisce il bersaglio, assegnando 50 punti per il sensore centrale e 10 per uno di quelli esterni.

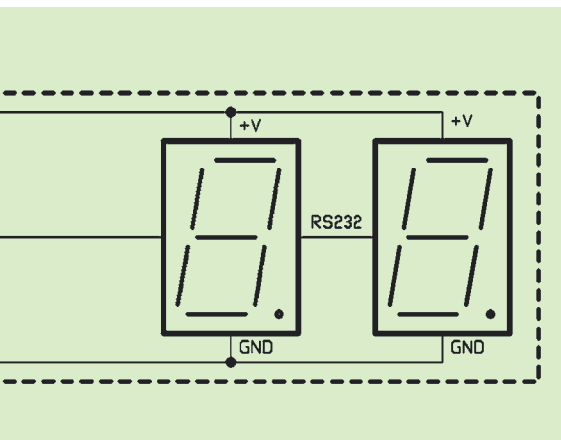
SCHEMA ELETTRICO

Il progetto qui descritto consta di due parti distinte, che sono la scheda base e il visualizzatore con interfaccia seriale, quest'ultimo formato da quattro moduli identici; la prima si occupa dei compiti più gravosi, ovvero mantiene l'ora (aiutata, in questo, da un modulo RTC on-board) permette le impostazioni di ora e sveglia, rileva lo sparo, ossia il fascio laser, e comanda il visualizzatore che mostra l'ora ed altro ancora. Il visualizzatore, come potete immaginare, non fa altro che mostrare i dati che l'unità base gli invia.

E allora, dedichiamo la nostra attenzione alla scheda base, iniziando col dire che rileva lo sparo, ossia il raggio del laser, mediante un bersaglio composto da cinque sensori composti da speciali fototransistor disposti a croce (quattro esterni ed uno

La nostra sveglia prevede un meccanismo di arresto della suoneria basato su un bersaglio elettronico, che dovete colpire al centro mediante una pistola giocattolo provvista di un piccolo puntatore laser rosso. Ma attenzione: la nostra sveglia è beffarda e mica si arrende facilmente! Per farla smettere di suonare bisogna colpire al centro del bersaglio e per farlo dovete essere piuttosto svegli; altrimenti continua ad emettere il suo avviso acustico. Se siete persone di spirito e volete alzarvi in un modo diverso dal solito (suvvia... non

ironizzate: è chiaro che tutti per *diverso dal solito* intendiamo avere una coniglietta di Playboy che ci porta il caffè a letto!!!) perché non prendete in considerazione l'idea di realizzare la sveglia qui proposta? Oltre all'indiscutibile originalità, presenta caratteristiche apprezzabili: innanzitutto visualizza l'ora mediante un display a led con quattro grandi digit, poi si imposta molto semplicemente mediante due soli pulsanti. Se poi pensate che i moduli dei digit si trovano in kit di montaggio e il circuito di controllo è semplicissimo... beh,



in centro); si tratta di L-932P3C della Kingbright, fototransistor che esternamente hanno due soli terminali e somigliano a led tondi da 3 mm in contenitore trasparente. La tensione ai capi di ciascun led viene letta da una linea di I/O del microcontrollore usato per gestire l'intera sveglia; per l'esattezza, le linee usate sono RA1, RA7, RB2, RB5, RB6 e ciascuna viene inizializzata come ingresso provvisto di resistore di pull-up. Quando il raggio del laser colpisce un fototransistor, il collettore di quest'ultimo si porta a livello logico basso. Quanto alle altre linee, RA2 legge il jumper che decide il modo di funzionamento del dispositivo, RB1 ed RB3 leggono lo stato dei pulsanti di impostazione; queste tre linee sono inizializzate come ingressi e le ultime due hanno impostato il pull-up interno. Ci

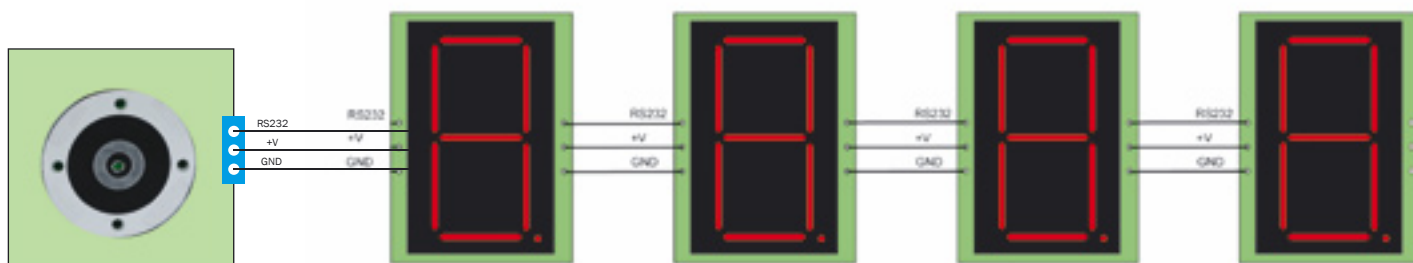
sono poi gli I/O inizializzati come uscite, che sono RA4, utilizzato per comandare (mediante un transistor NPN) il cicalino che produce le segnalazioni (avviso acustico della sveglia e avvisi di sensore colpito) ed RB4, impiegato come uscita per i dati diretti al visualizzatore. Il tutto si alimenta con una tensione continua di valore compreso tra 9 e 15 V, applicata al plug (+ e - PWR) di alimentazione; la tensione, passato il diodo di protezione D1, va ad un regolatore integrato U2, usato per ricavare 5 volt stabilizzati con cui far funzionare il microcontrollore e il modulo RTC. La stessa tensione a valle del D1 alimenta l'interfaccia seriale (T1 e T2) e i quattro moduli display che formano il visualizzatore. Il compito di visualizzare l'ora e assisterci nelle impostazioni è affidato a un display formato da quattro digit, ciascuno dei quali è un modulo a microcontrollore a sè stante e in grado di mostrare un carattere tra quelli previsti in un nutrito set, dietro comando impartito dalla scheda principale mediante l'interfaccia seriale di cui dispone. Ogni display rappresenta una singola cifra ed è il K8063 che abbiamo ampiamente descritto nel fascicolo n° 102 e utilizzato, ad esempio, nel progetto dell'indicatore di velocità

GPS per barche pubblicato nel fascicolo n° 129. Il modulo è progettato per essere usato insieme ad altri, in una struttura a bus dove la linea dati seriale è in comune fra tutti i K8063; non a caso i quattro moduli componenti il nostro visualizzatore sono collegati tra loro in parallelo, per quel che riguarda l'alimentazione e il canale seriale.

IL FUNZIONAMENTO

Dopo l'accensione il microcontrollore della scheda base inizializza gli I/O (i micro dei moduli display fanno altrettanto) e va subito a verificare lo stato del jumper collegato alla linea RA2 e quello dei pulsanti. Dando alimentazione mantenendo premuti entrambi i pulsanti fino a quando non compare la scritta *Shot*, invece che fare da orologio e sveglia il nostro circuito avvia il gioco di tiro a segno: ogni volta che il raggio laser della pistola colpisce uno dei sensori all'esterno il display incrementa di 10 punti, mentre quando ad essere colpito è il sensore centrale, il punteggio cresce di 50 punti. Per uscire dal gioco e usare il dispositivo come orologio e sveglia, bisogna spegnerlo e ridargli alimentazione accertandosi che entrambi i tasti siano a riposo. Nel normale utilizzo, il firmware che gira nel PIC della scheda base

Il nostro progetto



CIRCUITO BASE

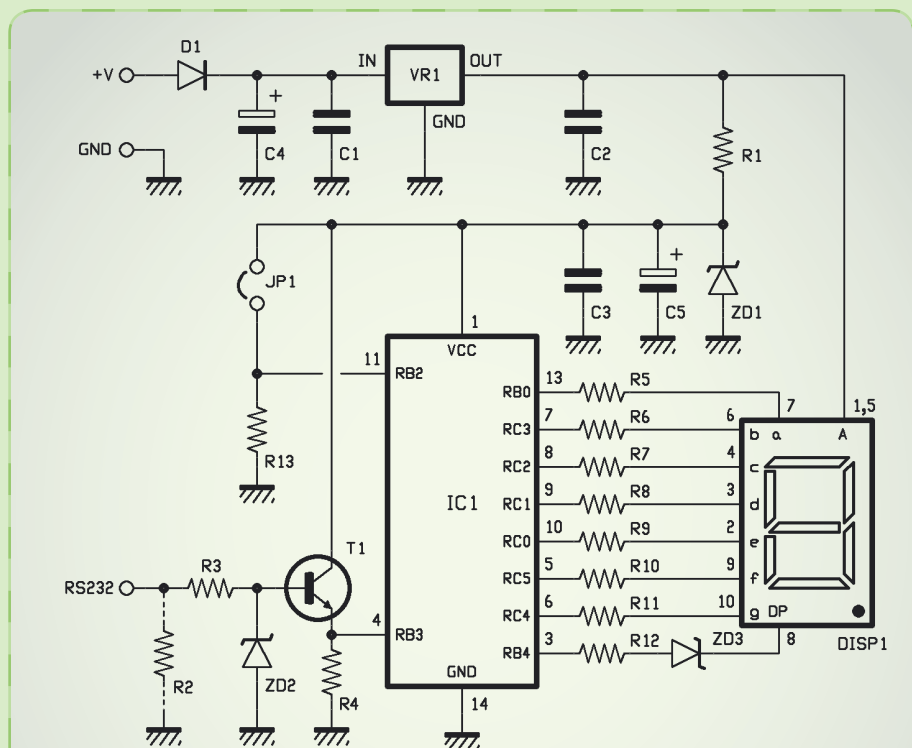
DISPLAY OROLOGIO/PUNTEGGIO

Il modulo display seriale

Ogni digit del display è un modulo Velleman K8063, il cui cuore è un PIC16F630 che legge e decodifica i dati in arrivo, pilota i sette segmenti del display a led e fa tutto quel che serve alle procedure di caratterizzazione. IC1 pilota direttamente i segmenti del display, in serie a ciascuno dei quali si trova solamente una resistenza per limitare la corrente; il comando avviene in modo "sink", nel senso che impone alle linee del microcontrollore impiegate a tale scopo di assorbire la corrente invece di erogarla. Tale modalità è, in realtà, obbligata, in quanto in erogazione le linee di I/O del PIC16F630 non potrebbero fornire la corrente che occorre ad accendere con la dovuta luminosità i segmenti di un display gigante.

Contornano il micro un alimentatore stabilizzato a 5 volt, realizzato con il diodo Zener ZD1, la relativa resistenza zavorra (R1) e i condensatori di filtro C3 e C5, il regolatore principale VR1 e un trasformatore di livelli logici RS232/TTL. Il regolatore principale serve ad ottenere i 9 volt che alimentano l'anodo comune del display a sette segmenti e quella che va poi a far funzionare il limitatore a 5 V (ZD1); proprio l'adozione dei 9 volt costringe a porre in serie ad RB4 il diodo Zener ZD3, perché le linee di I/O del microcontrollore possono lavorare al massimo con 5,5 o 6 volt e i 9 V presenti a riposo (uscita in stato open) potrebbero danneggiarle. Tale accorgimento si rende necessario per il solo punto decimale.

L'interfaccia di comunicazione attraverso cui riceve i dati sul carattere da visualizzare e sulle eventuali impostazioni è seriale, RS232-compatibile (perché il K8063 è stato costruito per essere comandato da un PC), e fa capo a un trasformatore d'ingresso necessario ad adattare i livelli di tensione



Elenco Componenti:

R1, R2: 560 ohm
R3: 10 kohm
R4: 1 kohm
R5÷R12: 100 ohm
R13: 100 kohm
C1÷C3: 100 nF multistrato

C4, C5: 100 µF 25V elettrolitico
IC1: PIC16F630 (VK8063)
VR1: 7809
ZD1, ZD2: Zener 5V1 400mW
ZD3: Zener 4V3 400mW
D1: 1N4007
T1: BC547

propria delle porte seriali dei computer alle esigenze degli I/O del PIC16F630, che accettano livelli TTL (0/5 V) e poco si addicono all'RS232, che prevede un livello alto pari a 10÷12 volt positivi e uno basso dell'ordine dei 10÷12 V negativi.

Nella nostra applicazione l'adattatore d'ingresso non servirebbe, tuttavia dato che il modulo è fatto così, per pilotarlo correttamente siamo stati costretti a implementare nell'unità di base un trasformatore di livello che trasformi l'1 logico TTL in una tensione di circa 12 V. Il trasformatore presenta in uscita 1 logico quando la linea seriale è a livello alto e zero quando la stessa

si trova al valore negativo. Notate che nel K8063 non serve alcun trasformatore TTL/RS232, perché il modulo display non deve trasmettere nulla, ma si limita a ricevere.

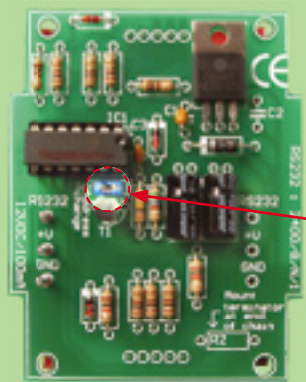
Per far visualizzare ad un display il carattere desiderato bisogna inviargli, mediante un PC o un microcontrollore, opportune istruzioni consistenti in stringhe composte da 5 byte ciascuna. Il significato dei byte e la loro sintassi sono i seguenti:

1°: è un header che inizia la comunicazione e avverte il display che stanno arrivando dei dati; vale 13 binario;
2°: indirizzo dell'unità display cui l'istruzione è diretta (0÷255);

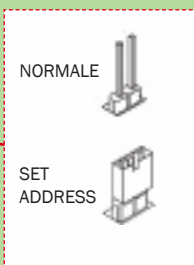
controlla costantemente lo stato dei pulsanti; se è partita la sequenza di sveglia e il cicalino sta suonando, un'apposita routine legge ciclicamente RA1, RA7, RB2, RB5 ed RB6; quando una di

esse si porta a livello logico alto, significa che è stato colpito uno dei sensori. Il microcontrollore tratta diversamente il sensore centrale da quelli esterni, nel senso che se a produrre l'1 logico

è il sensore centrale, il firmware va ad arrestare la routine di sveglia e quindi la segnalazione acustica corrispondente e ciò perché solo il segnale fornito da quello in centro può determinare



L'indirizzo si imposta alimentando il modulo digit con il jumper chiuso.



Il set dei comandi

COMANDO	DESCRIZIONE	PARAMETRO
'E'	Stop immediato	non serve
'D'	Visualizzazione indirizzo display	non serve
'B'	Segmenti da accendere	0÷255
'S'	Strobe	non serve
'C'	Cambio indirizzo del modulo	0÷255
'R'	Reset di tutti i moduli	non serve
'F'	Indirizzo forzato di tutti i moduli	non serve
'A'	Invio caratteri ASCII	32, 48...57 (spento, 0..9)
'P'	Punto decimale	0÷255 (OFF/ON)
'I'	Controllo di luminosità	0÷255 (luminoso/regolazione)

3°: comando, espresso sotto forma di equivalente ASCII della lettera che lo rappresenta (vedere tabella);

4°: parametro del comando (alcuni comandi, per essere eseguiti, richiedono sia definito un parametro);

5°: checksum; complemento di [256-(Byte1+Byte2+Byte3+Byte4)] per arrivare a 256.

Per essere certi che le istruzioni vadano a buon fine, conviene impartirle tre volte consecutive, onde evitare che uno dei moduli le perda (potrebbe accadere in visualizzatori con molti moduli).

Sempre per lo stesso motivo è consigliabile inviare la stringa di variazione

della luminosità per quattro volte.

Dopo aver impartito 'A', 'B', 'E' o 'P', bisogna lasciare trascorrere almeno 100 ms prima di inviare il comando di strobe ('S').

Dei cinque byte costituenti una stringa, il primo (che vale 13) è sempre un CR (Carriage Return) ed è espresso con l'equivalente binario del suo codice ASCII (13, appunto); in ASCII è rappresentato anche il comando, ovviamente definito da 8 bit che esprimono il suo codice. Gli altri tre sono, invece, numeri puri; per i comandi che non richiedono l'indirizzo, il secondo byte viene generato con un valore a caso, che verrà ignorato.

l'arresto della sveglia, mentre gli altri producono solo una segnalazione acustica. Nel gioco del tiro a segno, ogni sensore attivato determina l'incremento del numero visualizzato in base, ovviamente,

al sensore colpito. Quanto ai moduli display, dopo l'inizializzazione degli I/O conseguente all'accensione del circuito, ognuno esegue il self-test, ossia mostra tutte le cifre secondo la

sequenza .1234567890; fatto ciò visualizza l'attuale indirizzo, che, per impostazione predefinita, è 1. Il passaggio successivo è la verifica dello stato della linea RB2, impostata come ingresso: se la trova a zero logico significa che deve predisporre la routine di memorizzazione dell'indirizzo, altrimenti (jumper aperto) avvia il programma di visualizzazione, ossia si predispone a ricevere dalla seriale il carattere da visualizzare.

La memorizzazione dell'indirizzo consiste nell'assegnare a ciascun modulo un identificativo univoco in modo che, tra tutte le informazioni inviate sulla linea dati, ciascun modulo visualizzi solamente quella di sua pertinenza. Nel progetto qui descritto ci interessa la sola procedura manuale, che va condotta autonomamente per ciascun modulo. Quando la linea RB2 viene posta a zero (jumper chiuso) ha inizio la routine di impostazione manuale dell'address che prevede l'accensione in sequenza dei caratteri 5 (S) E, t, la quale conferma che si sta accendendo alla programmazione (in pratica il dispositivo visualizza, a suo modo, la scritta SET). Successivamente il modulo visualizza diciannove simboli: 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1., 2., 3., 4., 5., 6., 7., 8., 9. Ogni numero resta visibile per un paio di secondi; quando viene chiuso il jumper, il micro ferma la visualizzazione e scrive nella propria EEPROM il valore al momento visualizzato, che diviene il nuovo indirizzo del modulo.

Nel normale utilizzo ogni modulo mostra il carattere che il micro dell'unità di base gli ha inviato e lo fa fin quando la stessa scheda base non aggiorna i dati; per consentire ciò, ogni volta che sulla seriale arriva un carattere, il modulo destinatario lo salva nella propria EEPROM; se non è

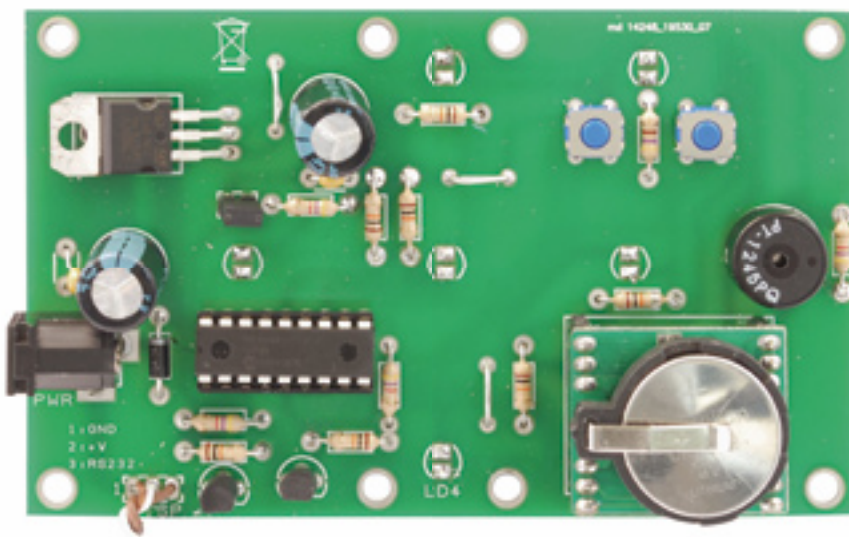
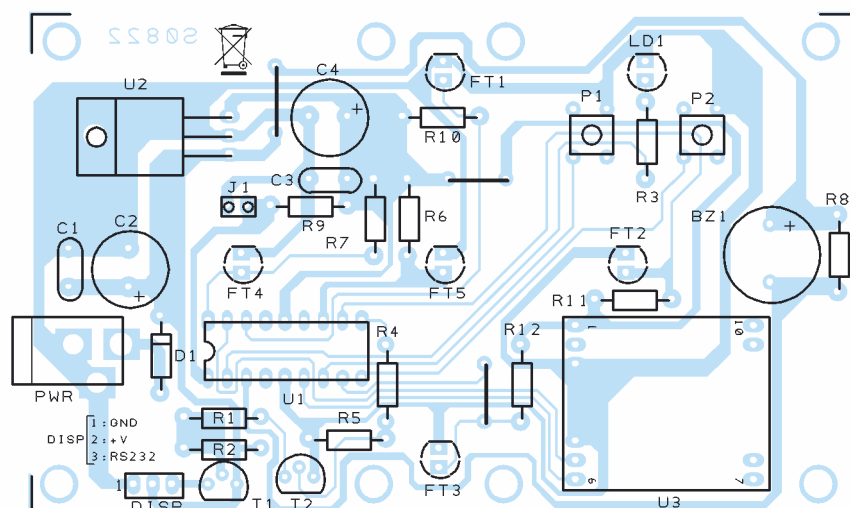
[piano di montaggio **SCHEDA BASE**]

Elenco Componenti:

R1, R4, R8, R9: 4,7 kohm
 R2, R5÷R7, R10÷R12: 10 kohm
 R3: 470 ohm
 C1, C3: 100 nF multistrato
 C2: 470 µF 35 VL elettrolitico
 C4: 470 µF 25 VL elettrolitico
 D1: 1N4007
 U1: PIC16F648A-I/PT (MF822)
 U2: 7805
 U3: Modulo RTC (FT674M)
 P1, P2: Microswitch
 FT1÷FT5: Fototransistor 3 mm (L-932P3C)
 LD1: Led 3 mm verde
 T1: BC557
 T2: BC547
 BZ1: Buzzer 12V con elettronica

Varie:

- Plug alimentazione
- Zoccolo 9+9
- Strip femmina 2 poli (3 pz.)
- Strip femmina 3 poli (2 pz.)
- Strip maschio 2 poli (1 pz.)
- Strip maschio 3 poli (1 pz.)
- Jumper
- Vite 10 mm 3 MA
- Dado 3 MA
- Scheda Display K8063 (4 unità display)
- Circuito stampato



stato memorizzato alcun carattere, il display resta spento. Per verificare i comandi in arrivo dalla seriale, il modulo display testa ciclicamente la linea RB3, verificando quando il livello di tensione passa da +12 V a zero volt, allorché il micro del modulo si predispone alla lettura dei dati. I comandi inviati dalla scheda base sono stringhe formate da cinque byte, che sono un header (13), l'indirizzo del modulo cui è diretto il comando, il comando, il parametro del comando e il checksum; quest'ultimo è usato

per verificare l'integrità della ricezione ed è la differenza tra 256 e la somma dei valori dei primi quattro byte.

Se il risultato è negativo, gli va sommato 256. Ad esempio, se il byte dell'indirizzo vale 1, quello del comando è C (83: è il valore decimale del carattere ASCII corrispondente) il parametro è 1, noto che l'header è sempre 13, otteniamo una somma pari a 158. Il micro dell'unità base esegue una routine periodica di aggiornamento dei display; dopo l'inizializzazione degli I/O e l'entrata

nella modalità normale, interroga ciclicamente il modulo RTC per chiedergli gli aggiornamenti sull'ora, quindi, ricevuti i dati prepara una sequenza di refresh. Ogni sequenza consiste nella generazione di una stringa per ogni modulo display, ciascuna delle quali è diretta a un determinato digit e contiene l'indirizzo corrispondente; le stringhe sono emesse sequenzialmente per gli indirizzi 1, 2, 3, 4. Siccome il nostro visualizzatore può mostrare ore e minuti, l'aggiornamento dei display e quindi l'invio periodico

delle stringhe nella modalità orologio avviene ogni minuto; per quanto riguarda le impostazioni, l'aggiornamento segue con breve ritardo la pressione dei singoli pulsanti. Il display delle decine delle ore deve avere indirizzo 1, quello delle unità ore 2, quello delle decine di minuti 3 e quello delle unità dei minuti 4: ogni sequenza di refresh inizierà aggiornando i dati nel display delle decine di ore.

COME SI USA

Bene, a questo punto diamo uno sguardo all'uso dell'orologio/sveglia: dopo l'accensione e la sequenza di avvio, possiamo iniziare a impostare l'ora e l'orario di sveglia; per impostare l'ora dell'orologio il jumper J1 deve restare aperto, mentre va chiuso quando si intende definire l'ora della sveglia.

Per regolare l'orario, a jumper aperto, premendo P2 si fanno avanzare le ore di un'unità per ogni pressione, mentre premendo P1 si regolano i minuti, che avanzano sempre di un'unità ad ogni pressione. Appena uno dei due pulsanti viene premuto compare la scritta *SetO*.

Se per almeno 5 secondi non viene premuto alcun tasto, si esce dalla procedura: l'ora viene memorizzata nell'RTC e compare la scritta *Set-*.

Per impostare l'ora di allarme (sveglia) bisogna chiudere il jumper; la cosa viene segnalata dall'illuminazione del punto decimale del display che visualizza le unità dei minuti; premendo P2 si regolano le ore (che avanzano di un'unità alla volta ad ogni pressione del tasto) mentre con P1 si regolano i minuti, che avanzano di un'unità ad ogni intervento sul pulsante.

Appena uno dei due pulsanti viene premuto compare la scritta

"SetA"; dopo circa 5 secondi, se non viene premuto alcun tasto si esce dalla procedura e l'ora viene memorizzata nell'RTC; ciò viene confermato dalla scritta *Set-*. All'ora impostata per la sveglia, quando la suoneria inizia a suonare è necessario sparare al centro del bersaglio; se si colpisce uno dei sensori esterni viene emessa una sequenza di note e la suoneria continua a suonare.

REALIZZAZIONE PRATICA

Per realizzare l'orologio sveglia bisogna preparare quattro

moduli visualizzatori K8063 e lo stampato della scheda base; per quanto riguarda i K8063, li trovate anche in scatola di montaggio presso Futura Elettronica (www.futurashop.it). Quanto, invece, all'unità base, lo stampato potete prepararlo per fotoincisione ricavando la pellicola dalla traccia lato rame scaricabile dal nostro sito web www.elettronica.in; incisa e forata la basetta, disponetevi i componenti a partire da quelli a più basso profilo e rispettando, per quelli polarizzati, l'orientamento indicato nel piano di mon-



Sonno e risveglio

Il sonno è discontinuo e interrotto da episodi di attività cerebrale simili alla veglia e variazioni vegetative (sonno REM, caratterizzato da movimenti ciclici degli occhi) ma anche da veri e propri risvegli con aumento del tono muscolare e della frequenza cardiaca e respiratoria, della pressione arteriosa e del consumo di ossigeno. Fin dai primi giorni di vita, il risveglio avviene prevalentemente in sonno REM, verso il 5° minuto dall'inizio di questa fase. Anche l'adulto si sveglia preferibilmente in sonno REM, mentre nell'anziano il risveglio avviene prevalentemente in fase 2 del sonno ortodosso anziché in sonno REM. I sogni si ricordano meglio dopo un brusco risveglio, che consente un accesso più facile ai contenuti del sogno immagazzinati nella memoria a breve termine. Il risveglio progressivo permette invece l'attivarsi dei meccanismi dell'oblio dovuti anche all'interferenza dei pensieri che sorgono durante la fase di veglia. Anche se i sogni sono presenti in tutte le fasi del sonno, sembra che la frequenza dei ricordi, la maggiore lunghezza e la bizzarria del sogno siano collegate al risveglio in fasi REM, perché qui la memoria del sogno si consolida più che nel sonno ortodosso.

La scheda base va racchiusa in un contenitore che frontalmente, da cinque fori, lasci esposti i lati sensibili dei fototransistor. I quattro display K8063 sono dotati di altrettanti coperchi di plexiglass rosso.



taggio. Notate che il led rosso e i fototransistor vanno montati dal lato saldature e rammentate che nei fototransistor il terminale più corto corrisponde al collettore. Quanto ai moduli digit, una volta montati bisogna provvedere alla programmazione dei loro indirizzi; ciò si effettua, per ciascun modulo, applicando l'alimentazione, mantenendo JP1 chiuso, allorché il display a sette segmenti comincia a visualizzare in rapida sequenza tutte le cifre che può mostrare, accendendo, nell'ordine, il punto decimale, poi 1, 2, 3, 4, 5, 6, 7, 8, 9, 0. Subito dopo mostra il numero corrispondente all'indirizzo attualmente impostato, che per impostazione predefinita è 1. Dopo l'indirizzo appare una breve sequenza di caratteri che compone la scritta **SEt**, a significare che ci troviamo nella procedura di impostazione, poi il display mostra una dopo l'altra le possibili impostazioni dell'address, ossia da 1 a 19. Disponendo di un solo digit, per rappresentare i numeri da 10 a 19 si avvale del punto decimale, che accende dietro ai numeri oltre il 9. In altre parole, il display per mostrare il 14 visualizza 4., mentre per visualizzare il quattro fa accendere 4. Per impostare

l'indirizzo voluto basta aprire JP1 quando il display mostra il numero corrispondente: ad esempio se si desidera assegnare l'address 12 bisogna attendere che venga visualizzato 2. e subito aprire il jumper JP1. Ricordate che il display delle decine delle ore deve avere indirizzo 1, quello delle unità ore indirizzo 2, quello delle decine di minuti 3 e quello delle unità minuti 4. Completato il montaggio, bisogna sistemare la scheda base dietro ad un coperchio rosso trasparente, ovvero dietro un pannello con cinque fori, uno per fototransistor; se volete, potete disegnare sul pannello il classico bersaglio. Adesso non vi resta che costruire la pistola con la quale spegnere la sveglia: basta una qualsiasi arma giocattolo acquistabile per pochi euro un po' dappertutto; dovrete modificarla inserendo nell'estremità anteriore della canna un puntatore laser quale ad esempio l'FR30 di Futura Elettronica (è un modulo a diodo laser con emissione di luce rossa a 670 nanometri) che è provvisto di un diodo laser a 675 nm, un collimatore con lenti in plastica e un alimentatore a corrente costante realizzato in SMD. Per l'alimentazione di questo puntatore bastano 3 V, che

potete ottenere inserendo nel manico due pile stilo collegate in serie; per accendere il laser utilizzate un microswitch collocato dietro al grilletto.



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono facilmente reperibili in commercio. Il master del circuito stampato ed il firmware sono scaricabili gratuitamente dal sito della rivista (www.elettronican.it). Il microcontrollore è anche disponibile già programmato al prezzo di 10,00 Euro (cod. MF822). Il modulo Real Time Clock FT674M è disponibile al prezzo di 18,00 Euro mentre ciascuna coppia di display giganti K8063 (in scatola di montaggio e con mascherina in plexiglass) costa 36,00 Euro. Ribadiamo che ciascun kit K8063 comprende due display giganti con la relativa circuiteria. Il modulo laser da 670 nm utilizzato nella pistola (cod. FR30) costa 22,00 Euro. Tutti i prezzi si intendono IVA compresa.

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

RASAERBA A ENERGIA SOLARE

dell' Ing. MIRCO SEGATELLO

È in grado di tagliare l'erba del vostro giardino seguendo un perimetro che potete definire a piacere ed evitando gli ostacoli che può incontrare; funziona con una batteria ricaricata mediante un pannello fotovoltaico e dispone di un sistema di sicurezza in grado di fermare la lama se si rovescia o viene sollevato da un bambino. Seconda e ultima puntata.

Sicuramente avete avuto modo di iniziare a leggere l'articolo sul tagliaerba a batteria con pannello solare proposto nel fascicolo di luglio/agosto scorsi; se avete apprezzato le prestazioni, l'ingegnerizzazione del prototipo e le funzionalità che offre, siamo certi che avrete già intrapreso la costruzione del robot e, dato che nella prima puntata ci siamo limitati a descrivere il progetto nell'insieme, la costruzione meccanica e il cablaggio dei circuiti elettronici, sicuramente state aspettando di conoscere le procedure di collaudo, taratura e messa in opera. Ebbene, in queste pagine concludiamo la descrizione del progetto occupandoci di tutti questi aspetti, oltre ad analizzare la struttura del firmware utilizzato per programmare il micro della scheda di controllo.



IL COLLAUDO

Per prima cosa dovete scollegare l'ESC, ovvero disconnetterne sia l'alimentazione che il comando; dopo sconnettete i sensori ad ultrasuoni e togliete il DSMC12 (in questa prima fase è importante siano collegati i soli pulsanti). Inserite nel portafusibile un fusibile da 0,5 A ed accendete il circuito; se tutto è a posto, sul display compare la scritta "Tagliaerba" e la versione del firmware e, dopo un secondo, la schermata di avvio. Spegnete tutto e programmate i sensori ad ultrasuoni: essendo collegati sullo stesso bus e disponendo ciascuno lo stesso indirizzo preimpostato in fabbrica, va da sé che non potranno funzionare contemporaneamente. Per il sensore destro lasceremo le impostazioni di fabbrica, che prevedono un indirizzo \$E0, mentre dovremmo cambiare l'indirizzo del sensore sinistro da \$E0 a \$F2. Per fare questo scollegate il sensore SRF destro e lasciate collegato solo l'SRF sinistro. Premete contemporaneamente il pulsante *PEn* e *PUp* ed accendete il robot, allorché sarà visualizzata la schermata seguente:

```
Set indirizzo  
SRF02 SX
```

Dopo 2 secondi apparirà questa schermata:

```
Collegare solo  
SRF02 SX
```

Dopo ulteriori 2 secondi vedrete questo messaggio:

```
PUp=set  
PEn=exit
```

Potete abortire la procedura premendo *PEn* o modificare l'indirizzo premendo *PUp*.

Nella malaugurata situazione in cui abbiate pasticciato un pochino è comunque possibile fare il passaggio inverso, ovvero passare dall'indirizzo \$F2 a \$E0 semplicemente abilitando la procedura di SETUP2, che si avvia premendo contemporaneamente i pulsanti *PEn* e *PDw* ed accendendo il robot. A questo punto inserite una alla volta le varie periferiche e verificate che sia tutto a posto; avrete così la possibilità di verificare eventuali anomalie ed il fusibile di piccolo valore eviterà danni importanti. Solo alla fine inserirete il fusibile definitivo, del valore di 2 ampere. Spegnete e riaccendete il rasaerba tenendo premuto il pulsante *PEn*: comparirà una schermata con la scritta "Test componenti" e subito dopo sarà avviata la procedura di collaudo dei singoli componenti. Tale procedura prevede la visualizzazione delle schermate che seguiranno.

```
Vbat 12,0V OFF→  
Ic 0,02A Pannel
```

Questa prima schermata riguarda la verifica dello stato dell'alimentazione, della tensione della batteria e della corrente generata dal pannello solare. Non

vi preoccupate se anche senza pannello viene letta una piccola corrente, perché ciò è dovuto alla tensione di offset dell'operazionale usato per la lettura. Adesso potete connettere il pannello fotovoltaico ed anche se non siete in pieno sole verificare che effettivamente tutto funzioni. Premete il pulsante *PUp* per abilitare la carica e *PDw* per disabilitarla; provate a mettervi al sole e verificate che la corrente aumenti (in pieno sole dovreste leggere circa 0,6 A). La freccetta in alto a destra del display vi indica che oltre alla schermata in esecuzione ce n'è un'altra, visualizzabile premendo *PEn*. Scollegate quindi il pannello fotovoltaico e premete *PEn*.

```
Mmotor 0 OFF  
Temperatura 22°C
```

Questa seconda schermata (Test temperatura) vi permette di verificare il funzionamento del sensore di temperatura (la NTC per intenderci). Non è importante che la misura sia accuratissima; anche un grado o due di differenza non creano problemi.



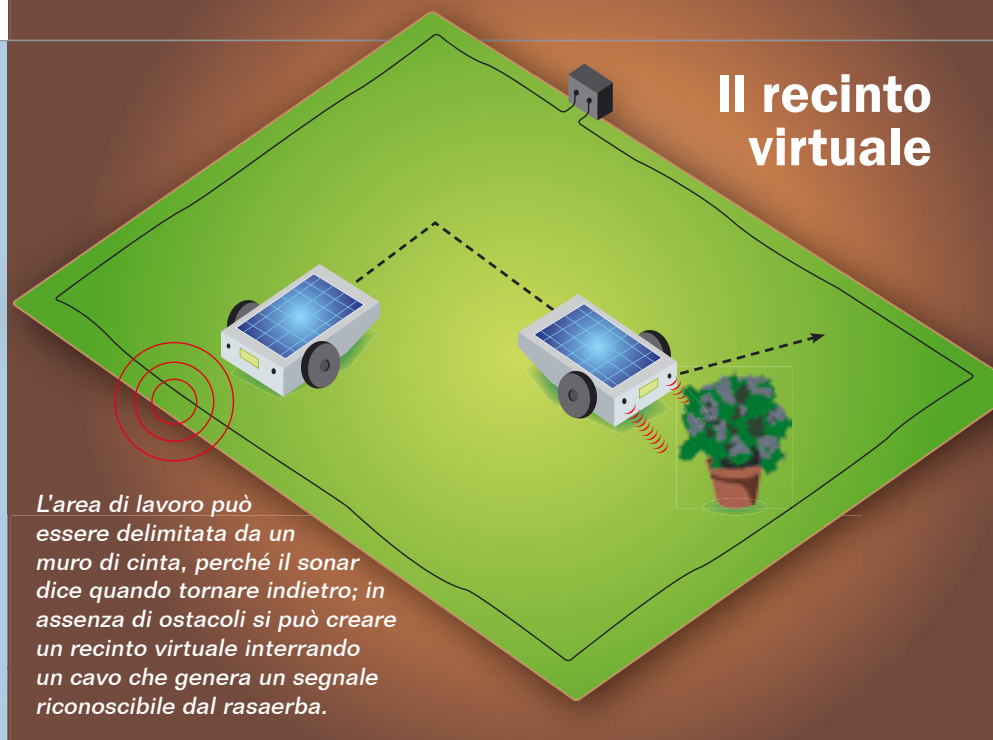
Toccate con le mani le due NTC per verificare che la temperatura salga. La variabile *Mmotor* mostra il livello (0÷1023) della tensione in ingresso al pin AN0: se avete collegato il sensore di movimento e la ruota pivotante sta girando, il valore deve mostrare un valore maggiore di zero. Questa funzione è disabilitata per impostazione predefinita, ma potete abilitarla premendo *PUp* (con *PDw* potete disabilitarla). Premete *PEn* per passare alla schermata successiva.

Test BWF →
DX=OFF SX=OFF

Questa schermata (Test BWF) vi permette di verificare il funzionamento del sistema di rilevamento del cavo interrato, cosa che potete fare solamente dopo aver alimentato il circuito generatore ed aver collegato alla sua uscita un filo di rame della lunghezza di qualche metro. Tenete il filo con le mani ed avvicinatelo nella parte bassa del frontale del rasaerba; le indicazioni sul display devono passare da OFF a ON. Se tutto va bene, premete *PEn* per passare alla schermata successiva, che è quella di *Test dei motori di trazione*.

Test Mot. Traz.
PUP=Mdx PDw=MSx

Questa quarta schermata serve a verificare il funzionamento del gruppo di propulsione. Premete *PUp* per abilitare il test del motore Destro: il motore deve accelerare lentamente, ruotare alla massima velocità per 2 secondi quindi decelerare e fermarsi; inoltre deve girare in modo da trascinare in avanti il rasaerba. Se così non è, basta invertire il filo + con il - sulla morsettiera. Ovviamente per facilitare il test



avrete l'accortezza di togliere le ruote. Provate sia il motore di destra che quello di sinistra, verificando che in entrambi i casi la rotazione avvenga nel verso di avanzamento del tagliaerba. Durante questa schermata, premendo contemporaneamente *PUp* e *PDw* si ottiene l'invio della configurazione del modulo DSMC12, impostando l'utilizzo di due motori con indirizzi 3 e 4. Queste sono anche le impostazioni di fabbrica riportate nel manuale fornito con il modulo. Se così non fosse, avviate questa procedura e spegnete e riaccendete il tutto per rendere operative le modifiche. Premete *PEn* per passare alla schermata successiva (*Test sensore ultrasuoni DX*).

Test SRF DX
Dist=56cm

Questa schermata permette il collaudo del sensore ad ultrasuoni destro: viene visualizzata sul display la distanza dall'oggetto rilevata. Utilizzate un oggetto rigido abbastanza grande e ponetelo a diverse distanze e verificate che coincida con le indicazioni a display.

Analoga è la schermata riguardante il test dell'altro sensore ad ultrasuoni (*Test sensore ultrasuoni SX*).

Test SRF SX
Dist=137cm

Questa schermata permette il collaudo del sensore ad ultrasuoni sinistro e per essa valgono le considerazioni fatte per quella precedente. Verificate che tutto sia a posto e poi proseguite fino a visualizzare la schermata seguente (*Test sensore ribaltamento*).

Test Sensore RB
Ribaltamento: NO

Questa schermata permette il collaudo del sensore di ribaltamento: quando il tagliaerba è in piano deve comparire la scritta NO, mentre se si va a ribaltare la scocca in qualsiasi direzione, la scritta deve diventare SI.

Warm-UP ESC
Attendere

Questa è la schermata successiva (*Warm-UP ESC*) e riguarda il test

I motori di trazione vengono gestiti da un doppio controller DSMC12, che si monta direttamente su un pin-strip della scheda base del rasaerba.



del motore per il taglio dell'erba. Se quando vi entrate, ancora non avete collegato l'ESC, spegnete tutto e connettetelo, quindi ritornate a questo punto della procedura.

Durante la fase di warm-up l'ESC si prepara all'avvio ed imposta il valore di comando relativo allo stop motore; in questa fase viene inviato un impulso della durata di 1,00 ms, valore che sul display LCD sarà rappresentato con il numero 0. Dopo 7 secondi (con il modello da noi usato) udirete un Beep, indicando che la procedura di calibrazione è terminata.

Dopo altri 3 secondi apparirà la schermata seguente:

Itaglio 0,12A →
M3_Vel=0

Anche in questo caso a motore spento il display può indicare un piccolo valore a causa dell'offset dell'operazionale usato per la lettura della corrente. Agite sui pulsanti *PUp* e *PDw* per aumentare o diminuire il valore dell'impulso di comando e di conseguenza della velocità di rotazione. Se state ancora collaudando il tagliaerba, riportate la velocità a 0, altrimenti potete lasciare il valore impostato e premere *PEn* per uscire dalla

procedura di test e ritornare alla schermata di avvio.

Il valore che imposterete qui sarà anche la velocità di rotazione durante il funzionamento regolare e dovrà essere sufficiente per un taglio regolare, senza eccedere per non incrementare troppo la corrente assorbita; nel nostro caso abbiamo usato il valore 18 su di un massimo impostabile di 30.

Fatto questo e risolti eventuali problemi che possono essere sorti potete fare un primo collaudo di funzionamento in interno, giusto per verificare che i motori e i sensori funzionino correttamente. Non è indispensabile l'utilizzo del pannello fotovoltaico. Premete *PUp* ed accendete il rasaerba per abilitare la modalità di test sul sistema; in questa modalità il tagliaerba avanza a mezza velocità rispondendo solo agli stimoli provenienti dai sensori sia di ostacoli che il filo interrato. In questa fase potete prendere dimestichezza con le funzioni svolte valutando anche la sensibilità di reazione dei sensori.

IMPOSTAZIONE DEI PARAMETRI

Molti dei parametri di funzionamento sono impostabili a piacere secondo le vostre esigenze, per cui potete accedere all'apposito menu che si abilita premendo il pulsante *PDw* durante la fase di accensione; si tratta del menu "Set parametri", che illustra tutti i parametri. Il primo di essi è *Tempo di retromarcia* e quella qui di seguito è la schermata corrispondente.

Tmp retromarcia→
1000msec

Quando il tagliaerba incontra un ostacolo, indietreggia e poi gira in modo da ricercare un percorso alternativo. Il tempo impiegato

per la retromarcia può essere impostato a piacere, mentre la velocità di esecuzione della manovra rimane fissata al 50 % della velocità massima.

Tmp virata →
1000msec

Questa seconda schermata (*Tempo virata*) mostra e permette di definire il tempo impiegato dal tagliaerba per girare dopo la fase di retromarcia conseguente alla presenza di un ostacolo.

Un valore troppo breve potrebbe causare un incremento del numero di manovre in zone ad alta densità di ostacoli; un buon valore è quello che permette al robot di eseguire una rotazione di 60° circa. Ovviamente la rotazione avviene nel verso opposto rispetto a quella della presenza dell'ostacolo. Non abbiamo previsto altri ulteriori metodi in quanto non avrebbero portato a sostanziali benefici.

Accelerazione →
2

La terza schermata del menu è *Accelerazione* e permette di visualizzare e modificare l'accelerazione; visti i tempi a disposizione per le manovre, questo non è un parametro critico. Essenzialmente questo valore definisce quanto il rasaerba oltrepassa il filo interrato una volta che l'ha rilevato.

Soglia distanza→
30cm

La quarta schermata (*Soglia di distanza ostacoli*) contiene un parametro molto importante, che indica a quale distanza dall'ostacolo il tagliaerba inizia la manovra per il cambio del percorso. Occorre fare in modo che a tutta

velocità e con l'inevitabile tempo di fermata, il robot non vada a sbattere contro l'ostacolo. Valori troppo elevati non consentirebbero il taglio dell'erba attorno agli ostacoli.

**Corrente M3 min→
0,60A**

La quinta voce di menu (*Corrente minima su M3*) corrisponde a questa schermata. Durante il funzionamento, il microcontrollore verifica in continuazione la corrente assorbita dal motore del taglio (M3 appunto) e la confronta con due parametri: *massima* e *minima*. Se la corrente è inferiore al valore minimo vuol dire che al motore non è richiesta particolare potenza, l'erba è bassa ed è possibile aumentare la velocità di avanzamento.

Se la corrente supera il valore massimo, molto probabilmente l'erba è alta e faticosa da tagliare quindi viene ridotta la velocità di avanzamento.

La schermata in questione e quella che segue (*Corrente massima su M3*) permettono di definire i valori minimo e massimo della corrente assorbita dal motore di azionamento delle lame.

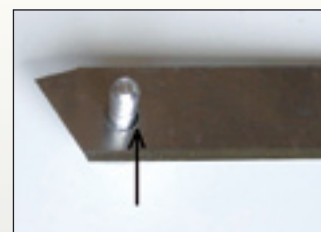
**Corrente M3 max→
1,00A**

Con il motore da noi utilizzato, le due soglie preimpostate risultano ottimali. Inoltre è previsto che queste due soglie definiscano anche un limite di allarme per il funzionamento del motore M3, cioè i valori per i quali si può ritenere che esso stia lavorando in condizioni ottimali e quelli che denotano un'eventuale anomalia.

STATO DEGLI ALLARMI

Abbiamo previsto, ovviamente, tutta una serie di allarmi per

Prepariamo la lama



Realizzare la lama del tagliaerba è molto facile: si parte da due lame per taglierino di misura standard (che hanno già il foro di aggancio al meccanismo di avanzamento del cutter); si infilano entrambe nel raccordo (mozzo) a vite già fissato all'albero del motore brushless, quindi si stringono mediante un bullone adatto, sotto il quale va inserita una rosetta per evitare che questo si sviti.

rendere il nostro tagliaerba il più sicuro possibile: in tutti i casi venga rilevata una anomalia, il robot viene immediatamente arrestato e messo in standby.

Il sensore di ribaltamento interviene se il robot viene rovesciato, caso ad esempio possibile in presenza di pozzetti aperti, laghetti non adeguatamente delimitati ecc., passando sopra i quali con una ruota è possibile che il rasaerba si inclini al punto da rovesciarsi su un fianco.

Se la corrente del motore M3 supera la soglia impostata, la velocità del tagliaerba viene diminuita, ma se ciò non dovesse bastare il rasaerba viene arrestato; ciò perché magari la lama si è impigliata in qualche cosa che si trova in giardino. Analogamente, se la corrente dovesse essere inferiore al minimo previsto e la velocità

di movimento del robot avesse già raggiunto il massimo, è probabile che il motore trifase stia girando a vuoto perché le lame si sono rotte oppure staccate; anche in questo caso il rasaerba verrebbe messo in standby.

Lo stesso accade se il sensore di temperatura rileva oltre 50°C. Quindi se il rasaerba è fermo in giardino, andate a leggere il display per verificare quale anomalia ne ha provocato l'arresto.

In ogni caso, **prima di sollevare o maneggiare il tagliaerba consigliamo di spegnerlo tramite l'interruttore generale**: solo questo vi garantirà che le ruote o le lame non gireranno.

Abbiamo volutamente lasciato per ultima la descrizione dell'assemblaggio delle lame per il taglio, proprio perché estremamente pericolose e quindi da usarsi

solo dopo aver già fatto tutti i controlli e la tarature necessarie. Già a maneggiarle ci si accorge di quanto taglino; immaginate quando sono in rotazione a 3.000 giri al minuto!

Quindi prima di toccarle raccomandiamo di scollegare anche la batteria.

Le lame sono facilmente recuperabili come ricambi per i cutter in qualsiasi supermercato; presentano già anche il foro per il fissaggio al cutter che nel nostro caso "cade proprio a fagiolo". Utilizzate un mozzo per alberi da 3,2 mm di quelli utilizzati in aeromodellismo per fissare le eliche e che ben si presta al nostro scopo. Quando assemblate le due lame dovete fare molta attenzione affinché risultino perfettamente allineate; inoltre, essendo il foro delle lame leggermente più grande del mozzo, dovrete fare in modo che la lama aderisca a quest'ultimo dalla parte più corta (non importa se rimane un piccolo spazio tra la lama ed il mozzo dalla parte del taglio)

L'intero gruppo (mozzo più lame) deve poi essere installato sull'albero del motore. A questo punto potete ricollegare la batteria e procedere alla fase di test componenti, verificando che la rotazione avvenga verso il filo delle lame. Se così non fosse invertite le lame oppure scambiate due dei tre cavi che alimentano il motore.

La verifica della corretta rotazione può essere fatta anche dopo alcuni tagli sul prato in tutta sicurezza e con rasaerba spento. Non si devono avvertire rumori e sibili particolari, né, tantomeno, vibrazioni. Per le prime prove impostate pure una bassa velocità di rotazione, corrispondente, nell'apposito menu, ad un valore di 15 o simile.

Collaudate con pazienza tutte le funzioni ed accertatevi che tutte le sicurezze funzionino a dovere.

IL FIRMWARE

Il firmware è abbastanza lungo e occupa l'intera memoria di programma del PIC16F876A; ciò perché abbiamo aggiunto parecchie funzioni di test che permettono di collaudare al meglio ogni singola funzione disponibile.

I più esperti di voi potranno sfruttare i codici sorgente per personalizzare le funzioni e magari aggiungerne di nuove.

La struttura del firmware è divisa in blocchi, precisamente:

- SETUP_1, per impostare l'indirizzo del sensore ad ultrasuoni sinistro;
- SETUP_2, per impostare (se necessario) l'indirizzo del sensore ad ultrasuoni destro;
- SETUP_PAR, per impostare i parametri di funzionamento;
- CHECK_1, per testare le periferiche;
- MAIN, che è il programma principale di taglio.

I vari blocchi possono essere richiamati tramite l'uso dei tre pulsanti a disposizione tramite le seguenti operazioni:

- Accensione+PEn
= collaudo singolo elemento del sistema
- Accensione+PDw
= impostazioni parametri di funzionamento
- Accensione+PEn+PUp
= imposta indirizzo SRFSX
- Accensione+PEn+PDw
= imposta indirizzo SRFDX
- Accensione
= schermata di avvio.

Nella fase di standby, evidenziata a display dalla scritta "Premi PEn per iniziare" è possibile accedere rapidamente ai menu di impostazione (PDw = impostazioni parametri di funzionamento).

Di seguito descriveremo solo la parte riguardante il programma principale, costituita essenzialmente da una serie di controlli dello stato dei sensori e delle periferiche, che sono:

- verifica della tensione della batteria;
- controllo distanza oggetti frontale;
- controllo presenza filo interrato;
- controllo temperatura interna;
- controllo corrente assorbita motore taglio;
- controllo pulsanti premuti;
- controllo ribaltamento;
- controllo di movimento.

Per quanto riguarda il controllo dello stato dei sensori, abbiamo usato un'accortezza: dato che i sensori ad ultrasuoni hanno un tempo morto di circa 100 ms dovuto al fatto che dopo aver dato il comando di misura (ossia dopo l'invio del segnale ad ultrasuoni) occorre attendere il segnale di eco prima di poter leggere il



risultato, approfittiamo di questo lasso di tempo per misurare il segnale proveniente dai sensori del filo interrato. Dovendo captare il segnale a 34 kHz generato dal filo, abbiamo utilizzato una funzione che possa misurare il numero di cambiamenti di stato del segnale di ingresso nell'arco di 100 ms. Confrontando questo valore con un valore di riferimento è possibile determinare se è stato rintracciato il segnale del filo interrato. Dovendo eseguire questa operazione per i sensori di entrambi i lati, in totale utilizziamo circa 200 ms. La lettura degli altri segnali impiega circa 100 millisecondi e quindi riusciamo a valutare la situazione circa 3 volte al secondo, frequenza più che sufficiente per un robot che viaggia a passo d'uomo.

L'intervento di uno dei sensori implica l'immediato avviso sul display e l'esecuzione in sequenza dei seguenti passi:

- decelerazione con stop;
- indietreggiamento;
- decelerazione con stop;
- rotazione (DX o SX a seconda del sensore coinvolto);
- decelerazione con stop;
- accelerazione e ripresa del taglio.

Se il pannello fotovoltaico fornisce più corrente di quanta ne viene consumata, automaticamente viene disconnesso dalla batteria; se invece la tensione della batteria scende al di sotto della soglia preimpostata, si avvia automaticamente la procedura di ricarica, che prevede lo spegnimento immediato del motore di taglio e lo stop del rasaerba in attesa della completa ricarica.

Questo punto è sicuramente il più critico di tutto il progetto, perché la logica imporrebbe la ricerca della zona a maggiore lu-

minosità per assicurare la ricarica della batteria nel minor tempo possibile. Tuttavia ciò è facile a dirsi, perché a noi basta uno sguardo per determinare la zona

del giardino a maggiore insolazione; ma il nostro rasaerba non ha un cervello: come fare, allora? All'inizio avevamo pensato di mappare lo stato della lumino-



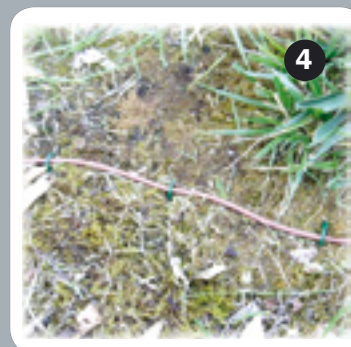
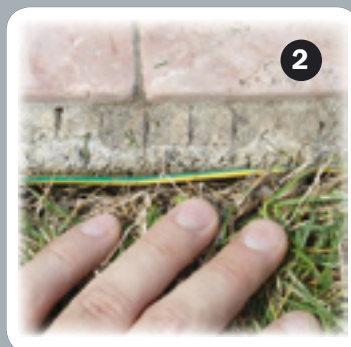
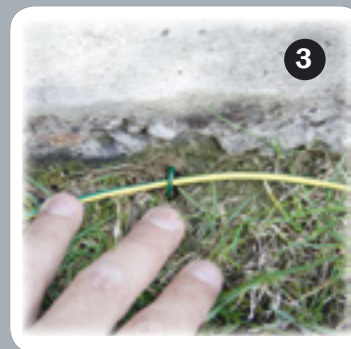
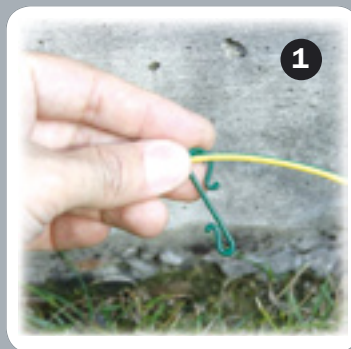
Un robot ... in erba!

Certe volte è vero che il progresso corre troppo forte: fino ad oggi credevamo che il massimo per un robot rasaerba fosse riuscire a muoversi da solo evitando animali, persone e ostacoli, e rimanendo in un perimetro delimitabile con un campo elettromagnetico. Insomma, un robot come il nostro. Adesso è arrivato il Nuovo Ambrogio L50 della Zucchetti, che può lavorare senza impianto perimetrale, in quanto è dotato di un sistema brevettato di riconoscimento dell'erba, che attraverso i 6 sensori di cui dispone riesce a capire quali sono le zone da tagliare e quali no. Ha davvero dell'incredibile!

Il Nuovo Robot Ambrogio L50 è dotato di batteria al litio che gli permette di lavorare in aree fino a 400 mq e di un motore di taglio con sistema di modulazione della lama che riconosce l'altezza dell'erba e ottimizza l'autonomia di lavoro e la qualità della rasatura. Il robot è stato concepito per muoversi senza alcun tipo di filo perimetrale e/o installazione; deve lavorare all'interno di un'area delimitata da muretti, cordoli, siepi o pavimentazione, in quanto i sensori lo arrestano quando incontra un ostacolo, una siepe, un muretto (alto almeno 10 cm) oppure quando rileva una pavimentazione (la distanza minima affinché il rasaerba riesca a capire che non c'è erba è di 25 cm). Per questa ragione va avanti finché c'è prato e torna indietro quando non c'è più nulla da tagliare. Più intelligente di così!

Preparazione dell'area di lavoro

Una volta messo a punto il tagliaerba potete passare a delimitare la sua zona di lavoro in giardino; per fare questo dovete procurarvi del filo di rame isolato di quello usato dagli elettricisti, del diametro di 1 o 1,5 mm. Si trova facilmente in matasse da 100 m in negozi specializzati oppure in negozi per il "fai da te". Scegliete un colore che si possa confondere con l'erba del vostro prato: ad esempio quello con guaina giallo-verde o marrone va più che bene. Procuratevi dei gancetti di plastica che serviranno per bloccare il filo al terreno: quelli usati per addobbare gli alberi di Natale o per fissare i drappaggi vanno benissimo (assolutamente da evitare quelli in metallo, per l'ovvia interferenza con il campo magnetico generato dal filo). Dovrete creare un anello tutto attorno all'area che vorrete delimitare, partendo dal circuito di alimentazione del cavo interrato, il quale sarà posizionato vicino a una presa di corrente. Il circuito di alimentazione del filo dovrà essere inserito in un contenitore stagno e posizionato, possibilmente, in una zona ombreggiata. Per la regolazione del trimmer della sintonia dovete utilizzare il tagliaerba abilitando la funzione di "test componenti"; la schermata corrispondente vi mostra lo stato di rilevazione dei sensori. Ruotate il trimmer in modo da ottenere la massima portata di ricezione; questa prova può essere



fatta anche al banco connettendo al circuito uno spezzone di filo di un paio di metri. Se notate parecchia differenza tra la portata del sensore destro e quello sinistro significa che i due circuiti sono accordati su due frequenze sensibilmente diverse, quindi rivedete i valori dei componenti C20, C23 e delle due

induttanze da 1 mH. Usate i gancetti per fissare il filo al terreno; l'erba in poco tempo lo ricoprirà e renderà invisibile. Se avete alberi molto esili difficilmente saranno individuati dai sensori ad ultrasuoni, perciò conviene delimitarli con il filo così come qualsiasi altro ostacolo non facilmente

sità nel tragitto percorso durante il taglio ed all'occorrenza raggiungere il punto migliore, però va detto che non è facile far raggiungere al nostro robot un punto ben preciso del giardino. E poi chi ci dice che quel punto sia ancora la migliore soluzione quando ci si ritorna? Basta che passi anche solo mezz'ora e una zona ben soleggiata può essere messa in ombra dalla fronda di un albero...

La ricerca della zona più luminosa potrebbe essere anche controproducente perché se si avvicina l'imbrunire il sole va

calando e più la si cerca, meno si trova. Le mutevoli condizioni atmosferiche, come ad esempio la presenza di nuvole, non ci permettono neppure di sapere cosa cercare, perché non possiamo sapere se già ci troviamo in una buona posizione oppure no. Come vedete, il problema non è dei più semplici; è per questo che abbiamo preferito sorvolare. Bene, chiuso questo discorso andiamo oltre e vediamo che, se è abilitato il sensore di movimento, il PIC testa continuamente la tensione ai capi del connettore corrispondente e se viene dato

il comando di rotazione ma non arriva segnale, significa che il tagliaerba è bloccato e si avvia la stessa procedura utilizzata nel caso di segnale da uno dei sensori.

Non è necessario valutare il movimento durante la manovra perché comunque essa dura non più di alcuni secondi. Se dopo tre tentativi consecutivi non arriva segnale di movimento si ferma tutto e viene generato l'avviso.

LE TARATURE

Regolate il tempo di retromarcia e di virata in modo che il

rilevabile dagli ultrasuoni. Se ponete vicino il filo di andata e quello di ritorno, i due campi magnetici si annullano a vicenda perché la corrente va in verso opposto, in questo modo potete delimitare anche piccole aree circoscritte. Avete la possibilità di regolare leggermente la distanza di rilevamento del filo da parte del tagliaerba variando la potenza del segnale trasmesso, cosa ottenibile modificando la tensione di alimentazione in un range tra 7 e 12 V. Fate eventualmente qualche prova per prendere confidenza con il funzionamento, modificando se necessario anche i parametri di funzionamento del microcontrollore. Abilitate il programma di test e verificate che il filo venga riconosciuto correttamente soprattutto nelle zone più critiche, quindi procedete al primo, vero collaudo. Scollegate l'ESC oppure impostate la velocità a zero (valore 0 sul menu) ed avviate il rasaerba anche senza pannello solare, tanto al momento vi interessa verificare i movimenti; sul display sono riportati la velocità di avanzamento impostata ed il livello della batteria. La pressione di un qualsiasi tasto provoca lo stop del tagliaerba. Nella versione definitiva, il pannello fotovoltaico viene fissato come coperchio e chiude il tutto. Per il fissaggio abbiamo utilizzato del semplice velcro autoadesivo interposto tra il pannello e la scocca; i pochi millimetri di distanziamento permettono una buona aerazione, necessaria perché durante il funzionamento in pieno sole la temperatura interna sale notevolmente.

tagliaerba riesca a muoversi con disinvoltura nel vostro giardino senza mostrare incertezze anche negli angoli più stretti. Durante il test del motore di taglio prendete nota della corrente assorbita (senza erba da tagliare): nel nostro prototipo eravamo sui 550 mA; per **IM3_min** dovete impostare un valore di poco superiore. Se durante il funzionamento la corrente assorbita è minore di questo valore, viene incrementata la velocità di avanzamento, ovviamente in modo graduale; se invece la corrente dovesse superare il valore

massimo impostato sarà gradualmente diminuita la velocità. Se dopo aver portato al minimo la velocità la corrente dovesse ancora superare il limite massimo, verrà generato un allarme ed il rasaerba sarà bloccato. Le tensioni di soglia per il controllo della batteria, per non appesantire con troppi parametri il settaggio, sono state preimpostate via firmware ad un valore standard. Se la tensione della batteria risulta minore di 10 volt, il rasaerba si mette in pausa sino al raggiungimento della piena carica, valore corrispondente a 13 volt. Durante il taglio del prato il pannello è sempre abilitato tranne il caso in cui la tensione della batteria supera i 13,5 volt; il pannello viene riconnesso quando la tensione della batteria scende sotto i 13 volt.

Durante il normale funzionamento il display mostrerà la tensione ai capi della batteria e la velocità di avanzamento:

```
VBat 12.0V
Velocit 70%
```

Ogni funzione svolta viene visualizzata a video per il tempo impiegato al tagliaerba per elaborarla, quindi ad esempio se viene incontrato il filo interrato sul sensore destro viene visualizzato:

```
BWF DX
```

Nel caso di intervento del sensore di movimento si avrà il seguente messaggio:

```
Anomalia RUN!
```

Se dovesse persistere per tre volte consecutive tutte le funzioni vengono interrotte e viene mostrato:

Bloccato!

Dopo ogni stop, premendo il tasto PEn si ritorna alla schermata di avvio dalla quale con il tasto PDw si possono modificare i parametri di funzionamento e, premendo nuovamente PEn, riprendere il taglio dell'erba. In ogni caso le funzioni sono molto intuitive e facili da comprendere e ben descritte dal display, perciò non dovrete incontrare grosse difficoltà a calibrare correttamente il vostro rasaerba. Bene, giunti a questo punto vi abbiamo detto tutto quel che avevamo da dirvi; l'erba ormai sarà già alta, quindi affrettatevi a finire il rasaerba e mettetelo subito all'opera! ■



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono facilmente reperibili in commercio. Il master del circuito stampato ed il firmware col quale programmare il microcontrollore sono scaricabili gratuitamente dal sito della rivista (www.elettronica.in) dal quale è anche possibile prelevare i disegni per realizzare lo chassis meccanico. Il microcontrollore PIC 18F876A-I/P già programmato (cod. MF827) è disponibile al prezzo di 15,00 Euro. I componenti elettromeccanici utilizzati nel progetto del tagliaerba solare sono tutti reperibili presso la ditta Futura Elettronica: Pannello solare STP10M Euro 82,00; Motoriduttore RH15812200 Euro 32,00 (2pz); Motore brushless BMA20-22L Euro 33,50; Driver motori DSMC12 Euro 28,00; Driver motore brushless A18A Euro 14,50; Sensore ultrasuoni SRF02 Euro 19,00 (2 pz); Interruttore a mercurio cod. 6710-978 Euro 2,70; Mozzo Pololu mod. HUB6MM Euro 10,50 (2 pz).

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

L'AUTO DAL VOLTO UMANO

a cura della REDAZIONE

L'auto del futuro di cui vi abbiamo parlato nel fascicolo 138 è già arrivata a casa dell'Ingegnere elettronico Mario Ravasi: si chiama RoadTunderStorm ed è una BMW dotata di computer e numerosi sensori, capace di interloquire con il conducente, esaudirne le richieste a voce ed assisterlo mediante sensori e apparati per rilevare le condizioni ambientali e di guida.

Chi non ha mai sognato un'automobile in grado di ascoltare, capire, parlare, avviarsi ed agire da sola? Il progetto R.T.S., acronimo di RoadThunderStorm, nasce per l'appunto da una vecchia idea di Mario Ravasi di realizzare un'auto per quanto possibile intelligente (ovvero dotata di un sistema elettronico computerizzato di controllo e di supporto alla guida e all'intrattenimento, con cui fosse possibile interagire tramite il linguaggio naturale come se al posto dell'auto si trovasse una persona fidata) accattivante dal punto di vista estetico, con un aspetto sportivo, aggressivo, ma

nei limiti dell'eleganza e in linea con le esigenze estetiche del momento. L'auto scelta per la realizzazione del progetto è una BMW serie 3 E36 del 1992. Gli interventi effettuati sull'auto sono tutti a norma di legge, come da regolare attestato di revisione periodica, nel rispetto delle codice della strada e della sicurezza nella guida: il monitor sullo specchietto retrovisore viene rimosso durante la guida e le luci strobo non vengono attivate se non da fermo e al di fuori della corsia di marcia. Il sistema computerizzato di controllo, denominato Bmw Unit Control System o BUC System, è



Il Cruscotto ed il quadro strumenti completamente accesi.

costituito da una piattaforma Windows ed è governato da Software proprietario realizzato dall'Ing. Ravasi, suddiviso in moduli per funzionalità, denominati BUC Modules, in grado di controllare l'auto nelle sue varie funzionalità.

L'interfacciamento con il sistema elettrico ed elettronico dell'autovettura avviene mediante l'utilizzo di schede e circuiti elettronici collegati al computer e comandate dai BUC Modules; in genere si tratta di schede a relé, di conversione A/D o D/A o provviste di INPUT/OUTPUT analogici e digitali, piuttosto che sensori veri e propri in grado di rilevare la luminosità esterna o la presenza di pioggia.

A tutti gli effetti il BUC System si può considerare un sistema aperto, nel senso che, anche in virtù del sistema operativo utilizzato, è in grado di implementare nuove eventuali funzionalità che potessero venire in mente senza particolari difficoltà di interfacciamento. L'auto coinvolta nel progetto si può così definire la prima ad essere totalmente "Windows Compliant".

L'interazione del guidatore con l'auto è consentita mediante l'utilizzo di un monitor touch-screen o tramite comandi vocali;

pertanto può avvenire in modalità:
- manuale, via touch-screen o mediante l'utilizzo della tastiera o del tastierino numerico posto sul tunnel centrale;

- vocale; cioè mediante comandi vocali impartiti tramite (ad esempio)

l'utilizzo di un microfono o di un auricolare Bluetooth, che permette di comandare l'auto anche dall'esterno. Il sistema è infatti dotato di un software di riconoscimento vocale in grado di interagire vocalmente col guidatore; per il momento il sistema di riconoscimento vocale non è indipendente da chi parla, ma deve essere addestrato da questi mediante una preventiva procedura di apprendimento.

I Modules sono dotati di interfaccia grafica (spesso con graficazione degli stati) in modo da fornire al guidatore le informazioni richieste ma anche di interfaccia vocale; quest'ultima, che il progettista ha chiamato Giulia, prevede anche un feedback vocale, in aggiunta a quello visivo e fornisce con voce femminile, i dettagli su quanto il sistema sta compiendo in relazione alla funzionalità attivata. È in grado, se necessario, di richiedere delucidazioni su quanto gli è stato detto di eseguire. La particolarità è che Giulia parla con tono dolce e sensuale ed è sempre pronta a prodigarsi in modo incondizionato per aiutare Mario nella

guida o nel recuperare le informazioni richieste; non a caso se interpellata, dice che "... è solo un piacere poter fare qualcosa per Mario ...".

A volte può mostrare un certo "caratterino" rispondendo per le rime, anche se esegue con assoluta abnegazione qualunque comando impartito, anche quello che le ordina di disattivarsi, perché anche in questi casi "... è stata programmata per ubbidire a Mario ...".

L'auto è dotata di un diffusore sonoro esterno attivabile anche vocalmente, per consentire di interagire e dialogare con Giulia e quindi di udire la sua voce anche all'esterno dell'auto, quando questa ha finestrini e porte chiusi.

Il sistema di governo dell'auto è altresì dotato di una telecamera esterna con illuminatore a infrarossi per monitorare l'ambiente esterno, attivabile tramite comando vocale o manuale; una telecamera interna che riprende il posto guida per monitorare chi sta guidando (attivata ad esempio dal Check Module). C'è inoltre una connessione ad Internet e un localizzatore satellitare.

Il sistema di controllo è in grado di pilotare una serie di funzionalità. In particolare è dotato di



una console grafica che rappresenta l'interfaccia principale con cui è possibile interagire con l'auto permettendo di accedere manualmente a tutti i comandi implementati, anche mediante una serie di menu. Tale interfaccia richiama persino un cruscotto virtuale in cui vengono evidenziate le informazioni principali quali la velocità, numero di giri, ecc., in sostituzione a quello in dotazione all'auto. Esiste anche la possibilità di far pronunciare vocalmente i valori del cruscotto virtuale e, ad esempio, farsi indicare da Giulia a voce la velocità istantanea. Non essendo dotata di standard OBDII, tali valori sono stati recuperati interfacciandosi direttamente ai sensori dell'auto, dopo aver recuperato tutti gli schemi elettrici.

Di seguito vediamo le principali funzionalità attivabili in maniera manuale o semplicemente richiedendone l'esecuzione a Giulia:

- accensione dell'auto; l'attivazione manuale richiede (per motivi di sicurezza) di inserire un codice mediante touchscreen o tastierino numerico dal momento che il sistema non è in grado di riconoscere, come abilitato alla guida, chi sta attivando l'auto, come avviene invece per il riconoscimento della voce;
- accensione del quadro strumenti e del Cruscotto; l'accensione avviene in modalità frazionata ovvero dispositivo per dispositivo per un maggior effetto visivo (prima il tunnel centrale, poi a seguire i restanti apparati elettronici);
- spegnimento del motore e del quadro strumenti;
- disattivazione del sistema; l'azione associata porta a disabilitare e spegnere tutti sistemi

- incluso il BUC System, togliendone la corrente; si tratta di un'azione a cui Giulia risponde in maniera non proattiva poiché in effetti porta al suo spegnimento fisico;
- accensione dei fari; il modulo permette l'accensione dei fari dell'auto;
 - apertura del tettuccio; l'azione permette di aprire il tettuccio elettrico;
 - attivazione dell'overboost; l'Overboost Module permette di interagire con l'ECM, Engine Control Module variandone alcuni parametri, al fine di permettere un incremento di prestazioni;
 - attivazione del climatizzatore; l'azione permette l'accensione del climatizzatore elettronico e la sua regolazione;
 - attivazione delle luci Strobo; l'azione è delegata allo Strobo Light System Module, e ha lo scopo di attivare le luci stroboscopiche, per un gioco ottico molto simpatico;
 - attivazione del BUC CheckControl e riconoscimento o espulsione del passeggero; il modulo è in grado di verificare lo stato dell'auto quali la presenza di luci accese e di porte aperte.

Mediante un sensore di presenza, il sistema è in grado di verificare la presenza di un passeggero, ne proietta l'immagine sul monitor e ne richiede il codice identificativo per capire se si tratta di un passeggero autorizzato o meno. Se il codice inserito dopo tre tentativi non corrisponde a quelli memorizzati, significa che la persona non è desiderata; in tal caso il Modulo attiva la procedura denominata



di espulsione consistente nel aprire il tettuccio e nello spegnere il PC, rendendo l'auto inutilizzabile. Infatti l'auto, in modalità automatica (impostabile) può venire accesa solo se il PC è acceso e si è autorizzati all'accesso o alla guida.

Oltre alle funzionalità descritte, esistono altri moduli di controllo che elenchiamo di seguito descrivendone il funzionamento

Attivazione di On Board Computer BMW

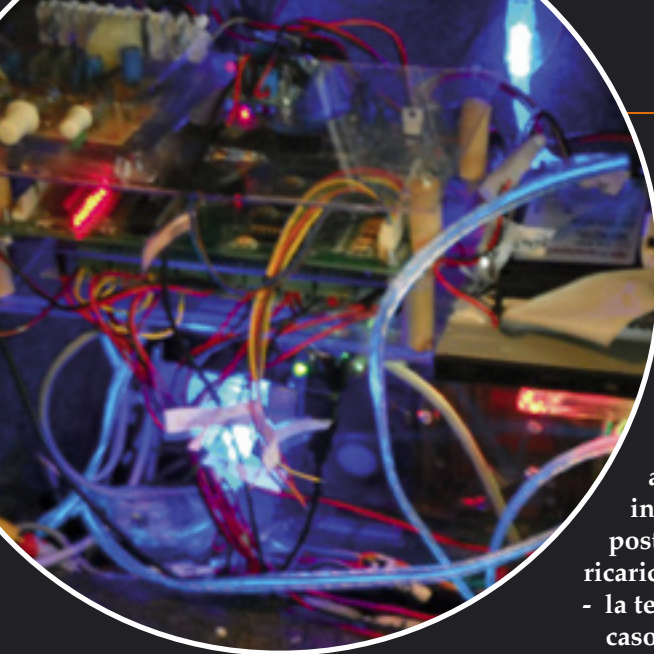
Il modulo è in grado di attivare il computer di bordo BMW, interfacciandosi direttamente con questo; l'OBC fornisce informazioni sulla macchina quali consumo medio, consumo istantaneo, velocità media, autonomia, luci bruciate, ecc.

Attivazione simulazione sbuffi NOS

È in grado di attivare la simulazione di sbuffi di protossido di azoto, mediante dispositivo elettronico comandato dal sistema che ha come effetto pratico la fuoriuscita di fumo dai parafranghi delle ruote anteriori.

Apertura e Chiusura delle Porte

È in grado di aprire o chiudere la sicura delle portiere e visualizzare nel contempo la



L'Unità Centrale di Controllo con alcune delle schede elettroniche di controllo collegate.

condizione assunta mediante diodi luminosi: l'accensione dei led verdi corrisponde a porte aperte, mentre i led rossi indicano porte chiuse.

Apertura e Chiusura Finestrini
Questo modulo è in grado di aprire o chiudere i finestrini.

Radar Anteriore
È in grado di rilevare mediante emettitori ad ultrasuoni la presenza di oggetti posti anteriormente l'auto. Nel caso si richieda l'accensione dell'auto il modulo di accensione verifica preventivamente che non vi sia alcun ostacolo entro una distanza impostabile. L'ideatore si è proposto di sviluppare l'interfacciamento con il Cruise Control, in modo da far decelerare l'auto agendo solo sull'acceleratore quando rileva ostacoli anteriormente.

Radar di Prossimità
Il radar è in grado di rilevare mediante un dispositivo ad onde radio (Bluetooth) l'avvicinamento di un individuo amico e programmare una serie di azioni da far intraprendere al sistema (apertura portiere, apertura finestrini, accensione auto, o altro).

Energy&Warm-Control Module
Il dispositivo, se attivato, per-

mette di monitorare i seguenti parametri:
- il livello della batteria: nel caso in cui la tensione dovesse scendere al di sotto di un valore predefinito è in grado di accendere il motore per un intervallo di tempo preimpostato in modo da poter ricaricare la batteria;
- la temperatura interna; nel caso in cui la temperatura interna arrivi al valore preimpostato è possibile programmare l'apertura del finestrino o del tettuccio (in maniera anche parziale) nonché l'accensione del climatizzatore, che avviene comunque 30 secondi dopo aver avviato il motore.

Quest'ultima funzione è utile specialmente d'estate, quando l'auto rimane per parecchio tempo esposta al sole cocente.

Verifica Parametri Principali
Il Parameters Module permette di effettuare una verifica dei parametri principali dell'auto, ovvero:

- temperatura interna;
- temperatura olio;
- tensione accumulatore principale;
- quantità residua di carburante;
- autonomia stimata.

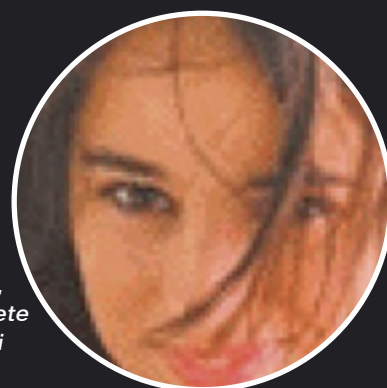
Il recupero dei dati avviene interfacciandosi direttamente coi sensori dell'auto e convertendo i relativi segnali in valori numerici; questo poiché l'auto non è dotata di sistema di diagnostica OBDII.

SCANSIONE DELL'AMBIENTE CIRCOSTANTE

Il sistema è in grado di effettuare una scansione dell'ambiente circostante recuperando le principali informazioni, quali la

temperatura esterna, la presenza di precipitazioni (tramite un sensore di pioggia) e il livello di luminosità (tramite un sensore crepuscolare). Mediante un ricevitore GPS e un software di navigazione satellitare, ricava anche latitudine, longitudine, indirizzo corrente, quindi provvede alla visualizzazione della cartina della zona corrispondente.

Dopo l'ampia sperimentazione della vettura, l'ideatore sta sviluppando alcune funzionalità aggiuntive quali la possibilità di accendere il sistema da remoto per poi collegarsi sempre in remoto (via connessione o desktop remoto) in modo da poterlo controllare tramite connessione GPRS. È in sviluppo anche la possibilità di inoltrare a intervalli periodici, ad un sito Internet, le principali informazioni che BUC System è in grado di rilevare (temperature, consumi, valori satellitari, immagini delle telecamere) in modo da renderle consultabili in real-time da parte di un qualsiasi utente autorizzato connesso via web. Se volete vedere in azione la supercar potete visitare la pagina web <http://www.marioravasi.it/RTS-Isolaccia.htm> e visionare il video dimostrativo; l'autore è a disposizione per tutti i chiarimenti del caso. ■



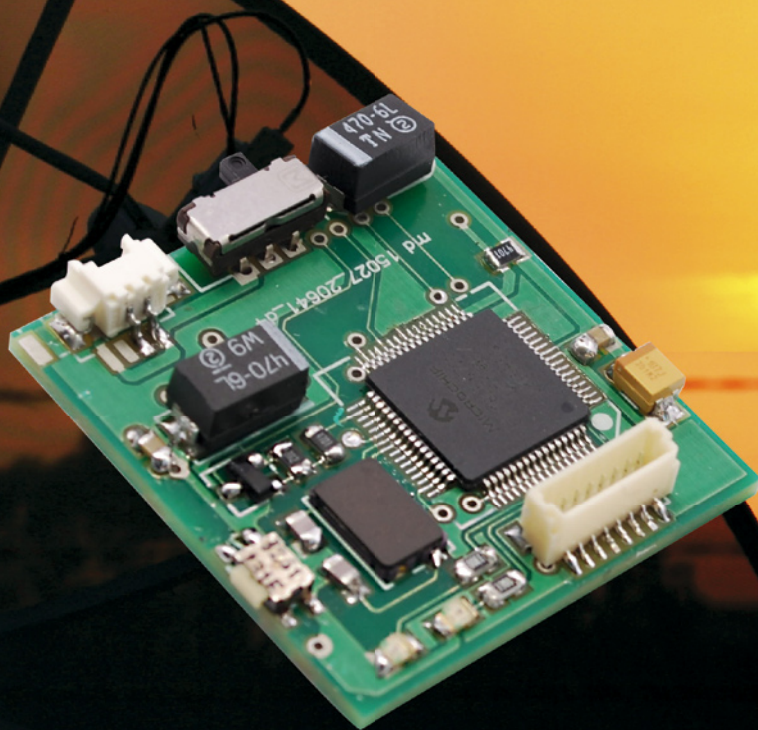
Il volto di Giulia, l'interprete vocale di BUC.

IL GPS? ... NON SERVE PIÙ!

LOCALIZZATORE TRAMITE RETE GSM

di BORIS LANDONI

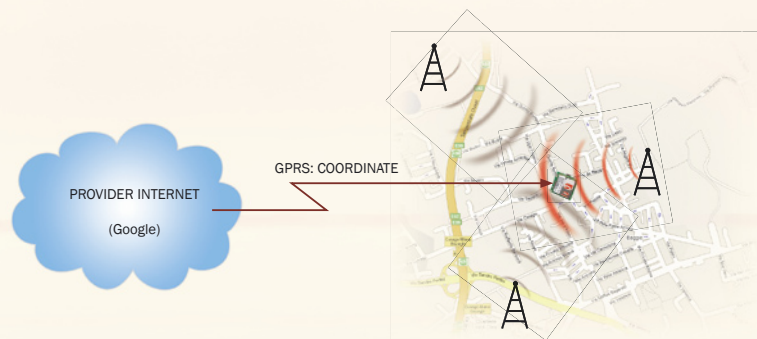
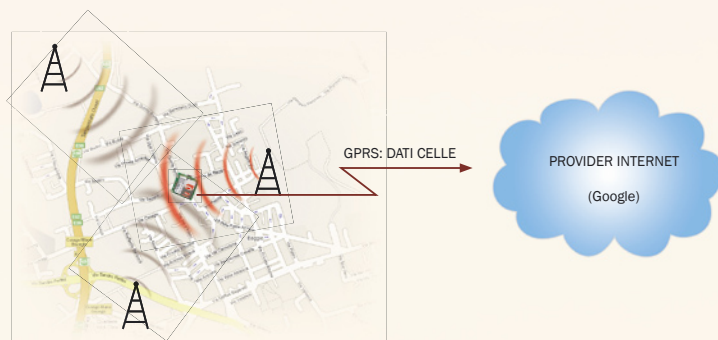
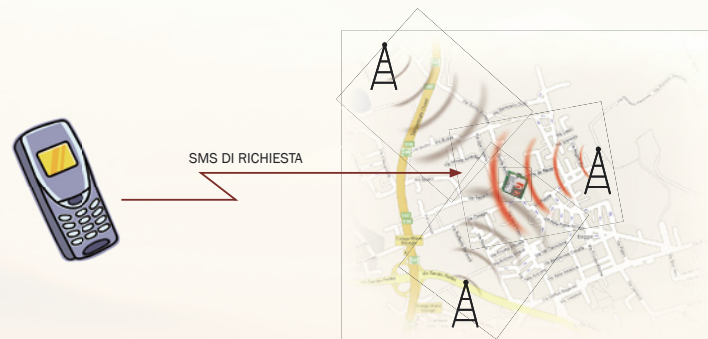
Pensate che per localizzare una persona o un veicolo ci voglia per forza il GPS? Preparatevi ad una sensazionale novità: sfruttando i dati della rete cellulare ed interrogando opportunamente Google, anche un semplice modulo GSM è in grado di determinare la propria posizione, con un margine di errore più che accettabile.



Se finora tutti i sistemi di localizzazione si sono basati sulla tecnologia GPS, oggi all'orizzonte si prospetta la possibilità di ricavare la posizione di un veicolo o di una persona

che porta con sé un dispositivo cellulare GSM, senza ricorrere ad alcun ricevitore satellitare e comunque con una discreta precisione. Miracolo? Naturalmente no! Semplicemente un

geniale connubio delle informazioni fornite dalla rete telefonica radiomobile e dei dati sulle coordinate delle celle ricavati da alcuni portali Internet tra cui anche Google Maps. Inten-

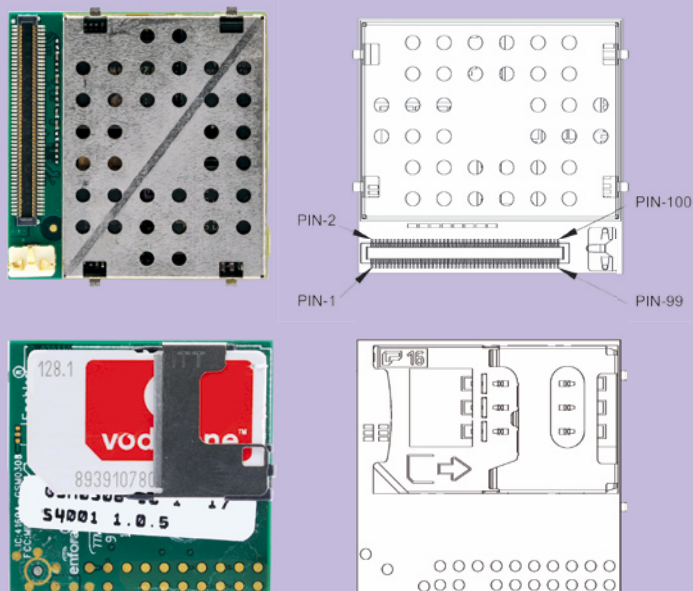


diamoci, in qualche modo il GPS c'entra, ma questo sistema consente di ricavare la posizione senza utilizzare direttamente questa tecnologia; semplicemente, sfruttando la disponibilità

Il modulo GSM/GPRS

Il cellulare utilizzato nel localizzatore è l'Enabler IIG GSM0308 della Enfora, uno dei più piccoli ma allo stesso tempo più performanti moduli del genere attualmente disponibili. Il dispositivo, che utilizza un chipset della Texas Instruments, misura appena 27 x 28 millimetri, è un Quadri-band in grado di operare sia in Europa che negli USA, e dispone di tutte le funzioni necessarie per realizzare dispositivi M2M, compreso lo stack TCP/IP. La documentazione tecnica di questo dispositivo è molto completa: dal sito del produttore (www.enfora.com) è possibile scaricare i manuali hardware, software nonché numerosi esempi di utilizzo. Per questo modulo è anche disponibile un economico sistema di sviluppo. Il modulo Enfora è un GSM/GPRS/Edge completo di tutto fuorché di

altoparlante, microfono e tastiera, che peraltro nella nostra applicazione non servono perché il cellulare viene sfruttato per gli SMS e la ritrasmissione di dati usando il sistema TCP/IP della rete GPRS. Anche l'antenna è esterna: nella nostra realizzazione abbiamo optato per portare la relativa connessione su una piazzola, dove si salderà il cavetto d'antenna. Il modulo ha un consumo ridottissimo: in trasmissione 230 mA in GSM 850/900 MHz e 175 in GSM 1.800/1.900 MHz, appena 10 mA in condizioni di riposo (scende a 100 µA in idle, ossia quando il modulo viene disabilitato mediante la linea PWRCTL). La potenza trasmessa in antenna è di 2 watt a 850/900 MHz e 1 W a 1.800/1.900 MHz; la sensibilità in ricezione è ottima: -106 dB.



di database con la posizione geografica delle celle, riusciamo a stabilire con sufficiente precisione la posizione. Ufficialmente, le coordinate delle celle non sono di pubblico dominio, e vengono gelosamente custodite dai gestori che le utilizzano per propri servizi a pagamento. E allora, dove troviamo questi dati? Chi e come ce li

può fornire? La risposta è molto semplice. Esistono numerosi portali Internet che hanno ricavato queste informazioni sfruttando i servizi messi a disposizione dei propri utenti. Ad esempio, Google col proprio servizio *Google Maps Mobile*, ha memorizzato miliardi di dati relativi alla posizione dei cellulari dei propri clienti, forniti dai telefoni

stessi: quando uno smartphone si connette al servizio di Google, invia sia l'identificativo della cella alla quale è connesso, che la posizione geografica ricavata dal proprio GPS. Ma ci sono anche dei siti che si ispirano alla filosofia dell'*open content* che utilizzano le informazioni inviate volontariamente dai propri utenti per realizzare database di questo tipo. Con questi sistemi vengono automaticamente ricavate e aggiornate le coordinate delle varie celle che sono alla base della localizzazione tramite rete GSM. Dobbiamo segnalare che, al contrario di quanto avviene in Italia, in alcuni paesi i gestori telefonici rendono disponibili le coordinate delle celle, la qual cosa rende molto più semplice la procedura di localizzazione.

L'ipotesi che Google abbia sfruttato il servizio Maps Mobile destinato ai possessori di cellulari con GPS per ricavare da esso, in modo collaterale, i dati che gli servivano, è la più probabile anche se non è la sola: qualcuno sostiene che le vetture usate per ricavare i filmati per la funzione Street View fossero munite anche di telefono cellulare per identificare e registrare la posizione delle celle in relazione alle coordinate fornite dal GPS; così, ogni volta che il telefono cambiava cella, un computer registrava le coordinate, costruendo il database. Anche se questa ipotesi è vera almeno in parte, è molto più probabile che l'archivio sia stato realizzato col primo metodo. Così facendo, in cambio di un servizio gratuito, Google ha usato i propri utenti per ottenere da loro informazioni che altrimenti sarebbero costate tempo (per andare in giro a fare i rilievi), e denaro per pagare personale e spostamenti o comperare le informazioni dai gestori telefonici,

ammesso che questi le fornissero. Intendiamoci: è una tecnica più che legittima, simile a quella usata dai grandi supermercati, che offrono ai clienti fidelizzati sconti in cambio dell'esibizione di una tessera nominativa grazie alla quale possono tracciare la spesa di ognuno e ricavare informazioni utili ad ottimizzare gli acquisti per area geografica e periodo dell'anno. Un favore in cambio di un altro favore...

Ma come funziona la localizzazione tramite GSM? La rete radiomobile è composta da tante celle radio adiacenti ognuna delle quali è caratterizzata da un identificativo, composto da quattro dati: un numero progressivo (Cell ID), un codice dell'area in cui si trova (LAC, ossia *Local Area Code*), il codice della rete nazionale di appartenenza (MCC, acronimo di *Mobile Country Code*) ed infine il codice del gestore (MNC, ossia *Mobile Network Code*) che, ovviamente, contraddistingue il gestore telefonico.

Per questa ragione, noti il nome della cella e le sue coordinate, considerata la distanza alla quale la cella permette la comunicazione con il prima che quest'ultimo ne agganci un'altra, si può sapere approssimativamente a quale distanza massima si trova il telefono. Ad esempio, se tale distanza ammonta a 1 km, il cellulare può trovarsi in una zona di 1 km di raggio.

Da ciò deriva che più numerose sono le celle in una zona, più precisa può essere la determinazione della posizione del telefono (fino a 50-100 metri).

La nostra idea di sfruttare unicamente un dispositivo GSM per realizzare un sistema di localizzazione remota è nata da una considerazione fatta a proposito di *Google Maps Mobile*: concepito per consentire agli smartphone

The screenshot shows the OpenCellID website interface. At the top, there's a navigation bar with links: About, Stats, Download, Api, Browse cells, Cells Map, and Raw data. The main heading is "Listing cells". Below it, a note says "Note: you can get the Kml view here:". The table lists cell data with columns: Lat, Lon, Mcc, Mnc, Lac, Cellid, Range, Nbsamples, Created at, Updated at, Needs computation, Country, and Operator. Three rows are visible, each with a "Show Map" link.

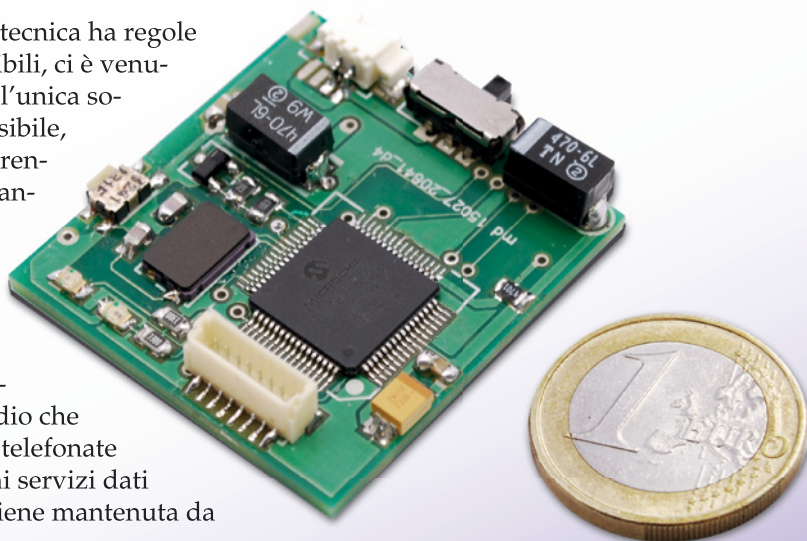
Lat	Lon	Mcc	Mnc	Lac	Cellid	Range	Nbsamples	Created at	Updated at	Needs computation	Country	Operator
47.2480344772339	39.7961282730103	250	20	26221	2632	0	2	Tue Aug 11 15:19:15	Tue Aug 11 15:22:38	false	Russia Russian Federation	Tele2 / ECC
53.34433	-6.26229	272	1	31	1411640		1	Tue Aug 11 15:07:04	Tue Aug 11 15:07:04	false	Republic of Ireland Ireland	Vodafone Ireland
53.3816614285714	-6.35114142857143	272	1	3014	3403440		2	Tue Aug 11 15:06:15	Tue Aug 11 15:06:36	false	Republic of Ireland Ireland	Vodafone Ireland

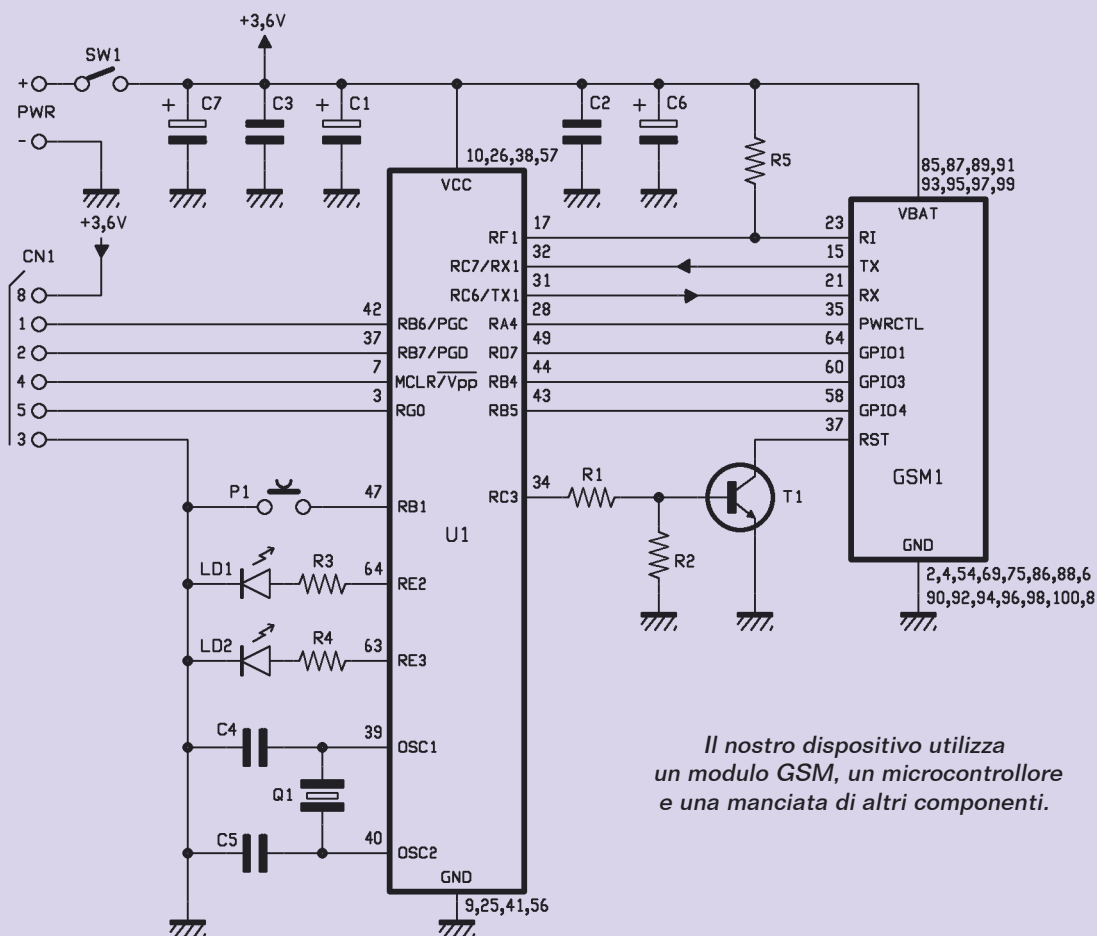
Pagina Browse Cell del sito www.opencellid.org.

dotati di ricevitore GPS di sfruttare Google per la navigazione satellitare, ad un certo punto il servizio è stato esteso ai cellulari comuni (purché capaci di supportare la modalità dati GPRS o UMTS) e a quel punto, a qualcuno particolarmente attento, noi compresi, si è "accesa la lampadina"... Tutti abbiamo cominciato a domandarci come facesse Google a dire a un possessore di un telefonino privo di ricevitore GPS dove si trovasse in quel momento.

Dato che la tecnica ha regole imprescindibili, ci è venuta in mente l'unica soluzione possibile, che si comprende considerando che nel sistema telefonico radiomobile GSM la connessione radio che permette le telefonate e l'accesso ai servizi dati e Internet viene mantenuta da

una rete di ponti radio, le celle di cui abbiamo parlato prima. Durante il normale funzionamento il cellulare si collega alla cella che gli garantisce il miglior collegamento possibile ma rileva anche i segnali delle celle vicine (*neighborhood cells*) in modo da essere pronto a commutare su una di esse quando quella in uso, ad esempio perché ci stiamo spostando, non permette la copertura necessaria a garantire una buona comunicazione.





Ogni cella trasmette, nel messaggio di comunicazione con il cellulare, il proprio identificativo, che, come abbiamo visto, è unico al mondo. Per questa ragione ogni cellulare, quando è connesso con la rete di un determinato gestore, conosce esattamente la cella cui si appoggia.

Sulla base di ciò abbiamo ritenuto

to che il sistema usato da *Google Maps Mobile* per localizzare la posizione del cellulare consistesse nell'abbinare il nome della cella ai dati sulla sua posizione; in pratica il cellulare trasmette via rete GPRS i dati della cella in quel momento "agganciata" e il server di *Google Maps Mobile* li correla con la posizione della cella stessa. Durante la messa a punto del nostro progetto siamo andati a tentativi, anche perché

Google non solo non mette a disposizione direttamente i suoi dati su Internet, ma si guarda bene dallo spiegare come sfruttarli. Durante le prove abbiamo connesso un cellulare GPRS con interfaccia USB (sul quale avevamo caricato *Google Maps Mobile*) a un computer a sua volta connesso a Internet via LAN: "sniffando" (ossia andando a verificare i dati che transitavano sulla rete durante la connessione) mediante programmi come *Microsoft Network Monitor* il dialogo tra cellulare e server di Google la nostra ipotesi si è fatta certezza: Google conosce la posizione (le coordinate) delle celle del sistema radiomobile e se gli forniamo l'ID riesce anche a dire dove una cella si trova. Nota la posizione della cella, visto che si conosce il raggio di copertura di ciascuna, è



facile valutare la zona circolare, di raggio pari al raggio di copertura, dove si trova il cellulare che inoltra la richiesta di determinazione della posizione.

Chiaramente questa metodica

ne precisa come quella offerta dai sistemi con ricevitore GPS, presenta numerosi vantaggi rispetto ad essi; innanzitutto è più leggero e meno ingombrante, dato che ha un pezzo in

mente con i dati della rete cellulare si tratta solo di attendere la risposta alla richiesta effettuata al server di Google. Senza contare che il cellulare "prende la linea" anche in molte gallerie autostra-

[piano di **MONTAGGIO**]

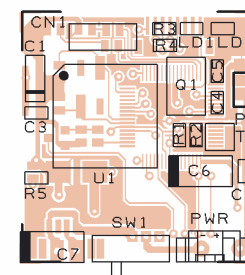
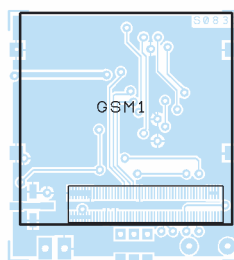
Elenco Componenti:

R1, R5: 4,7 kohm (0805)
R2: 10 kohm (0805)
R3, R4: 330 ohm (0805)
C1: 100 μ F 6,3 VL tantalio (CASE-B)
C2, C3: 100 nF multistrato (0805)
C4, C5: 10 pF ceramico (0805)
C6, C7: 470 μ F 6,3 VL tantalio (CASE-X)
U1: PIC18LF6722 (MF833)
LD1: led rosso (0805)

LD2: led verde (0805)
Q1: quarzo 20 MHz (12SMX)
T1: BC817
SW1: Deviatore a slitta GSM1: GSM0308
P1: Microswitch 90° SMD

Varie:

- Connettore da cs 100 pin per GSM
- Connettore 8 poli 90° passo 1 mm
- Circuito stampato



consente una localizzazione grossolana, la cui accuratezza dipende strettamente dai dati forniti dal server di Google circa la copertura di una data cella.

IL NOSTRO PROGETTO

A prescindere da come Google ha compilato il proprio database, una volta scoperto come avviene la richiesta da parte dei telefonini che usano *Google Maps Mobile*, ci è subito venuto in mente di sfruttare la funzionalità per realizzare un particolare localizzatore remoto, che, diversamente da quelli finora proposti, non monta un ricevitore GPS ma utilizza il solo modulo GSM/GPRS che garantisce la connessione radio. Detto...fatto: in queste pagine trovate descritto il nostro nuovo localizzatore, il quale, sebbene non garantisca una localizzazio-

meno. Inoltre ha un costo minore e, cosa non da poco, garantisce maggiore autonomia di esercizio, dovuta al fatto che siccome in un localizzatore quel che consuma di più è il ricevitore GPS, se non c'è, il problema è in buona parte risolto. Ciò significa che con una batteria a ioni di litio, ad esempio da 1 Ah, il nostro localizzatore può rimanere in funzione per parecchi giorni (tutto dipende dal numero di chiamate che deve fare) mentre con una batteria da 13Ah il dispositivo può restare in funzione per un paio di mesi. A ciò va aggiunto il fatto che un localizzatore basato sulla rete cellulare può essere più immediato nella risposta di uno convenzionale, dato che in particolari situazioni, il ricevitore GPS può richiedere alcuni minuti per determinare la propria posizione,

dali e in seminterrati dove un ricevitore GPS non riceve affatto. Prima di vedere com'è fatto il circuito del localizzatore è opportuno ricordare che l'opera dei numerosi interessati alla localizzazione da cellulare ha fatto sorgere uno dopo l'altro diversi siti Internet dove è possibile trovare molti dati sulle posizioni delle celle fruibili gratuitamente; ci sono poi servizi a pagamento, come quello di OctoTelematics (www.octotelematics.it) che custodisce gelosamente un database eccellente per completezza. Tra i siti gratuiti nei quali abbiamo sbirciato vi segnaliamo www.opencellid.org e www.cellidb.org; in entrambi potete trovare informazioni e verificare la posizione di una cella dopo averne introdotto l'identificativo (Cell ID). A tale riguardo segnaliamo che l'iden-

Chi ha orecchi per intendere, intenda ... la connessione a Google

Per poter associare la cella GSM alle coordinate geografiche relative, utilizziamo una funzione non documentata di Google Maps Mobile. Prima però dobbiamo sapere a quale cella è connesso il dispositivo remoto GSM, ovvero dobbiamo conoscere MCC, MNC, LAC e CID. Per ottenere questi dati il microcontrollore PIC che gestisce il dispositivo deve inviare i seguenti comandi al modulo Enfora:

AT+COPS? per ricavare MCC e MNC

AT+CREG? per ricavare LAC e CID

A questo proposito ricordiamo che nella fase di setup del modulo GSM è necessario impostare l'Enfora in modo che questo risponda correttamente a domande di questo tipo. Per fare ciò, dunque, precedentemente il PIC avrà provveduto ad inviare al modulo i seguenti comandi:

AT+COPS=0,2

AT+CREG=2

Una volta ricavati i dati di MCC, MNC, LAC e CID, l'unità remota potrà inviare la richiesta a Google (www.google.com) utilizzando la funzione HTTP POST. Come prima cosa, dunque, è necessario effettuare una connessione a Google e poi inviare una richiesta con la seguente procedura:

POST /glm/mmap HTTP/1.1

HOST: google.com

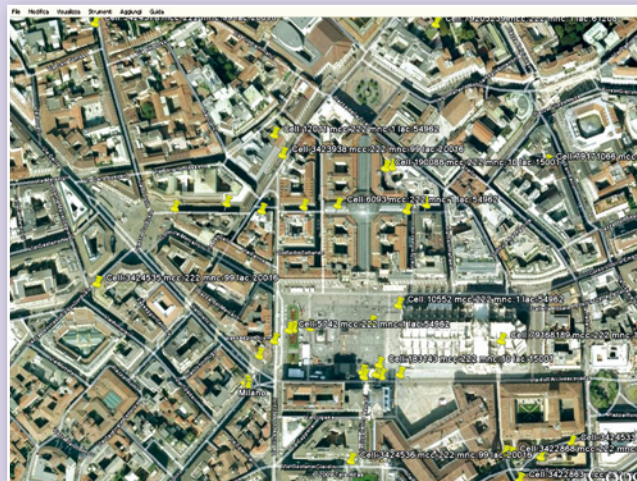
Content-Type: application/binary

Content-Length: 82

Connection: close

La stringa è composta da 82 caratteri che dovranno contenere anche i dati della cella:

21	2 byte	2
0	8 byte	8 10
2	2 byte	2 12
it	1 byte	2 14
13	2 byte	2 16
Boris Landoni	1 byte	13 29
7	2 byte	2 31
3.0.1.6	1 byte	7 38
3	2 byte	2 40
Web	1 byte	3 43
27	1 byte	1 44
0 0 3	4 byte	12 56
0	2 byte	2 58
cid lac mnc mcc 0 0	4 byte	24 82



E' molto importante rispettare la lunghezza dei dati, infatti nell'header Content-Length: 82 è necessario specificare quanti dati dovranno essere inviati: se la lunghezza della stringa dovesse essere differente, il server rifiuterà la richiesta. Dopo l'elaborazione, il server risponderà con una stringa contenente i dati geografici della cella (se ovviamente è presente nei database di Google) e l'approssimazione dell'unità remota, ovvero la massima distanza del dispositivo da quel punto:

2 byte	a
1 byte	b
4 byte	c
4 byte	latitude
4 byte	longitude
4 byte	approssimazione
4 byte	d
2 byte	e

Durante le nostre prove non siamo riusciti a stabilire con certezza a cosa corrispondono i dati a,b,c,d, anche perché, come già specificato, Google non documenta in alcun modo questa funzionalità, e a queste conclusioni siamo giunti solamente dopo un'opera di "sniffing" dei dati scambiati dall'applicativo per cellulari Google Maps Mobile e il server di Google. Se qualcuno volesse studiare più a fondo questo argomento ...

tificativo può essere ottenuto anche manualmente dal cellulare, installando programmi come *Nokia Net Monitor* o *WM Cell Catcher*, un'utility di Windows Mobile che abbiamo usato per verificare i dati "sniffati" nel collegamento tra il nostro cellulare e il server di Google. In particolare, *www.*

opencellid.org rende disponibili delle API, tra cui quelle per scaricare le informazioni sulle celle per ottenere la localizzazione, ma anche per dare al sito informazioni ricavate dagli utenti sulla posizione di nuove celle. Ad esempio, se abbiamo un cellulare con GPS e ci agganciamo a una

cella, possiamo desumere il codice identificativo e la posizione e trasmetterle al sito, che poi le verificherà e pubblicherà a favore di tutti quelli che lo visiteranno. Nello stesso sito troviamo un file che, sfruttando Google Earth, ci fa vedere zona per zona quante e quali celle ci sono, distinte per

MCC ed MNC. Proprio riguardo ai codici di gestore, sappiate che in Italia MNC 1 identifica TIM, 10 Vodafone, 88 Wind e 99 Tre (H3g). Il file si scarica dalla sezione Browse cells del sito www.opencellid.org, facendo clic sul simbolo di Google Earth. Perché tutto funzioni, bisogna prima aver scaricato ed installato Google Earth dal sito www.google.com; installato il plug-in, nella sezione *Luoghi* di Google Earth appare, sotto *Luoghi temporanei*, la voce *View Cells*.

Comunque, per sviluppare la nostra applicazione abbiamo preferito avvalerci dei dati di *Google Maps Mobile*, perché rispetto agli altri siti fornisce informazioni non solo sulla posizione presunta ma anche sull'approssimazione, ovvero, indicato un punto, completa l'informazione dicendo a quale distanza massima il cellulare può trovarsi da esso. Tra le informazioni viene indicato l'indirizzo della via più vicina. Va anche detto che il

cella sulla base dell'intensità del segnale non sempre fornisce indicazioni precise, in quanto ostacoli e condizioni atmosferiche, ma anche disturbi radioelettrici possono influenzare la ricezione.

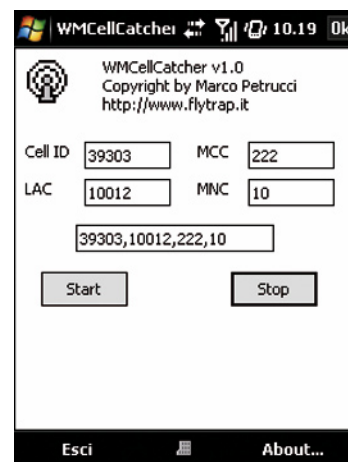
Il nostro localizzatore nasce per l'uso a batteria e quindi per essere portato addosso da persone che possono avere l'esigenza di chiedere aiuto o di essere rintracciate, ma anche per venire messo a bordo di autoveicoli (senza necessità di installazione) o semplicemente introdotto in merci viaggianti. Serve a fare un po' tutto quello che fa un localizzatore, a parte dirci se si trova da un lato o dall'altro della strada; ecco, questo certo non riesce a farlo, ma sicuramente può aiutarci a capire se un autista sta seguendo il percorso previsto o se ne sta andando in un altro quartiere in un paese vicino. Per evitare di scaricare inutilmente la batteria, abbiamo previsto che il localizzatore fornisca la propria posizione su richiesta inoltrata

Le segnalazioni dei led.

CONDIZIONE OPERATIVA	SEGNALAZIONE
Accensione	10 lampeggi veloci di entrambi i led
Setup gsm	Led verde acceso fisso
Attesa easy setup	Led rosso acceso fisso (lampeggia quando arriva la chiamata)
Connessione GPRS	Entrambi i led accesi fissi
Invio SMS	Led verde acceso fisso
Pressione pulsante SOS	Led rosso acceso fisso e poi sequenza di invio (GPRS -> Invio SMS)
Ricezione di una chiamata	Led rosso acceso e poi sequenza di invio (se il numero è tra quelli autorizzati)
Ricezione di un SMS	5 lampeggi veloci del led rosso
Stand by	Lampeggio continuo del led verde

sistema adottato da Google per ora non impiega la triangolazione, ovvero i dati delle celle vicine, che pure potrebbero permettere una localizzazione più precisa; e neppure verifica la potenza del segnale della cella, dato che potrebbe trarre in inganno. Infatti determinare la distanza dalla

via SMS, con una semplice telefonata o periodicamente; inoltre abbiamo implementato la funzione SOS: premendo il pulsante di cui è dotato invia un SMS di richiesta di aiuto al numero che è stato memorizzato, contenente le coordinate della posizione relativa. Più esattamente, permet-



WM Cell Catcher permette di conoscere i dati della cella cui il telefono è connesso.

te di memorizzare un massimo di otto numeri telefonici abilitati a inviare le richieste via SMS, o con una semplice chiamata, alla quale non risponde per non far spendere denaro inutilmente.

Quando viene interrogato, invia un SMS con l'esito della localizzazione. Nella modalità autoreport, ossia nell'invio periodico, manda SMS con i dati di localizzazione ad uno o più numeri di quelli in memoria, in base alle impostazioni fatte dall'utente. Il principio di funzionamento è illustrato nel disegno relativo: quando vogliamo conoscere la posizione del dispositivo remoto dobbiamo inviargli un SMS di richiesta. A questo punto l'unità verifica i dati della cella a cui è connesso ed invia una richiesta (via GPRS) al sito di Google; quest'ultimo risponde con le coordinate ed il dato relativo alla precisione. Non appena ricevute queste informazioni, l'unità remota ce le "gira" mediante un SMS. Il tutto avviene in pochi secondi.

SCHEMA ELETTRICO

Il circuito che realizza il localizzatore è formato semplicemente da un modulo GSM/GPRS quadri-band della Enfora, inter-



Camminare lungo la giusta via

Parenti e infermieri alle prese con malati di Alzheimer da oggi possono stare più tranquilli: negli Stati Uniti la GTX e la fabbrica di scarpe Aetrex Worldwide hanno realizzato speciali scarpe equipaggiate con un localizzatore GPS che permette di rintracciare i pazienti quando si allontanano dalle loro case e cominciano a vagare senza meta. Si tratta di un'importante novità, perché l'Alzheimer è una malattia che conduce ad una progressiva perdita di memoria, causando sovente uno stato confusionale e di disorientamento al punto che se escono per una passeggiata, i malati non sono più in grado di tornare a casa, né di cercare aiuto. Il 60% dei malati di Alzheimer si è perso almeno una volta. La

tecnologia consentirà di localizzare un individuo ovunque vada, con una precisione dell'ordine dei 9 metri; è stata sviluppata dalla GTX Corp. utilizzando micro-ricevitori GPS che vengono inseriti in una scarpa. I dispositivi vengono montati all'interno del tacco della scarpa invece che in braccialetti o ciondoli perché i malati di Alzheimer non amano indossare oggetti non familiari, che appena possono si tolgono di dosso. Negli Stati Uniti il prezzo delle scarpe al dettaglio è stimato tra i 200 e i 300 dollari, ai quali va aggiunto un canone mensile di circa 20 dollari per l'abbonamento al servizio di sorveglianza. I test dovrebbero essere completati entro l'anno, e le scarpe dovrebbero entrare in commercio nel 2010.

facciato ad un microcontrollore Microchip che, inizializzati gli I/O e l'UART di cui dispone, fa girare in loop il main-program attendendo un evento, che può essere l'arrivo di un SMS o la pressione del pulsante P1 (ovvero la transizione 1/0 logico della linea RB1, internamente provvista di resistore di pull-up). Nel caso di ricezione di un SMS, distingue le azioni a seconda che si tratti di un messaggio che imposta una configurazione o che richiede la posizione. Soffermiamoci un istante sulla procedura che consente di rica-

vare la posizione, che è praticamente uguale per la richiesta manuale e la pressione del P1 (allarme o S.O.S.): all'arrivo della richiesta, il PIC comanda il modulo cellulare affinché si connetta in modo dati (via GPRS, quindi su Internet) al server di Google Maps e invii la richiesta di posizione corredata dell'identificativo della cella cui il modulo Enfora è collegato; quindi si dispone a ricevere sul canale RX del proprio UART i dati in arrivo da Internet. Giunti i dati con la posizione presunta (latitudine e longitudine, via o piazza appros-

simativi) e l'accuratezza, prepara un SMS e lo trasmette al numero che ha fatto la richiesta da SMS o a quello memorizzato abbinato alla funzione di allarme.

È tutto qui: semplice e funzionale. Il cellulare viene gestito dal micro mediante le linee RF1 (con la quale legge l'RI, ossia il *Ring Indicator* che il modulo usa per segnalare l'arrivo di una chiamata) ed RB4, grazie alla quale rileva l'arrivo degli SMS. RC7/RX1 ed RC6/TX1 sono le linee, rispettivamente, di ricezione e trasmissione dell'UART, usato per la lettura e l'invio degli SMS, oltre che per gestire le funzioni del cellulare, eccezion fatta per il reset e l'alimentazione; quest'ultima viene controllata dalla linea RA4, che agisce su PWRCTL per comandare l'accensione e lo spegnimento del GSM1 e che viene usata essenzialmente per avviare il cellulare dopo l'inizializzazione, ovvero per spegnerlo e riaccenderlo se si blocca e non basta il reset a fargli riprendere il normale funzionamento. Il reset viene operato dal PIC tramite la linea RC3, che si avvale del transistor T1, qui impiegato come interruttore statico che fisicamente fa commutare l'RST del modulo. L'insieme, ossia il cellulare e il microcontrollore, sono alimentati mediante l'interruttore SW1 da una tensione continua di 3,6 volt, applicata ai punti + e - PWR, facilmente ottenibile da una batteria a tre elementi ricaricabili NiMH da 1,2 V oppure da una Li-Ion da 3,7 V. Notate i numerosi condensatori di filtro dislocati strategicamente lungo la linea di alimentazione, necessari a filtrare i disturbi prodotti dall'assorbimento del cellulare quando è in trasmissione, disturbi che altrimenti creerebbero fluttuazioni dell'alimentazione tali da bloccare il microcontrollore.

Funzioni e comandi

FUNZIONE	COMANDO SMS	PARAMETRI	VALORE DI DEFAULT
CAMBIARE PASSWORD	PWDxxxx;12345	xxxx=nuova password	12345
MEMORIZZARE UN NUMERO (massimo 8 numeri) (MAX 19 CIFRE PER NUMERO)	NUMx+39nnnnnnnnn;12345	x=posizione del numero nnnnnnn=numero da memorizzare	
CANCELLARE UN NUMERO	NUMx;12345	x=posizione del numero	
VERIFICARE I NUMERI MEMORIZZATI	NUM?;12345		
RESET COMPLETO	RES;12345		
IMPOSTARE I NUMERI A CUI VERRANNO INVIATI GLI SMS (PER AUTOREPORT / SOS / POLLING)	SMSxxxxxxx;z	xxxx=posizioni in cui si trovano i numeri da avisare z=può valere ON o OFF	tutti i numeri verranno avisati
RICHIEDERE IMEI	IMEI?		
IMPOSTARE IL NOME	NAME:xxxx	xxxx=nome del localizzatore (massimo 15 caratteri)	
RICHIEDERE IL NOME	NAME?		
MODALITA' DI INVIO NOTIFICA IN CASO DI RICHIESTA TRAMITE SQUILLO	ARI:x	x=M solo chiamante x=S tutti gli utenti abilitati	S
SELEZIONARE IL FORMATO COORDINATE PER L'INVIO TRAMITE SMS	FORS:x	Vedi testo	
RICHIEDERE FORMATO	FORS?		
AUTOREPORT CONTINUO	AUTOC:z	z=può valere ON o OFF	
INTERVALLO DI TEMPO PER L'INVIO DI UNA NOTIFICA (CONTINUO)	AUTOC:hh/mm	hh=ore mm=minuti	
RICHIESTA IMPOSTAZIONI AUTOREPORT	AUTO?		
IMPOSTAZIONE APN	GPRAPN:apn	apn=apn del provider (esempio web.omnitel.it)	
CANCELLAZIONE APN	GPRAPN		
IMPOSTAZIONE USERNAME GPRS	GPRSUSR:xxxxxxxx	xxxx=username (solitamente non necessario)	
CANCELLAZIONE USERNAME	GPRSUSR		
IMPOSTAZIONE PWD GPRS	GPRSPWD:xxxxxxxx	xxxx=password (solitamente non necessaria)	
CANCELLAZIONE PASSWORD	GPRSPWD		
VERIFICARE IMPOSTAZIONI GPRS	GPRS?		
DISABILITARE LA RISPOSTA PER QUEL MULTIMESSAGGIO	RISP		
RICHIESTA COORDINATE	COO		

LA REALIZZAZIONE

Il nostro localizzatore è decisamente semplice, tuttavia costruirlo richiede una certa attenzione, in quanto tutti gli elementi che lo compongono sono in versione SMD. Una volta ottenuta la basetta (è a doppia faccia e le due tracce possono essere scaricate dal nostro sito www.elettronica.in, in formato TIFF) armatevi di un saldatore da non più di 20 watt a punta molto sottile e di filo di lega saldante da 0,5 mm di diametro (o meno, se l'avete). Per prima cosa sistemate il microcontrollore, stagnando

un paio di piedini su due lati opposti per fermarlo, quindi procedendo col resto dei piedini; passate poi ai componenti passivi, al quarzo e al transistor. Per il modulo GSM/GPRS è previsto un piccolo connettore ad alta densità da stagnare sullo stampato, magari aiutandosi con una lente di ingrandimento per verificare che non ci siano "baffi" di stagno a cortocircuitare piazzole vicine; inserito il modulino, su di esso si fissa e si connette il porta-SIM. La connessione dell'antenna è, per il cellulare, un triplo contatto a molla dorato che,

spingendo il componente a fondo nel suo connettore, tocca una piazzola che abbiamo previsto su un angolo del c.s.; su di essa si può stagnare il cavetto dell'antenna o, a seconda della sagoma, l'antenna stessa. Per il nostro prototipo, come antenna è stato usato un semplice spezzone di filo lungo circa 8 cm, tuttavia, a seconda dello spazio disponibile, si può optare per un'antenna specifica da cellulare, magari anche di quelle realizzate su stampato. Completato il montaggio, attraverso il connettore CN1 (8 poli) si può provvedere alla

IMPOSTAZIONI E COMANDI

Prima di andare avanti va detto che alcuni comandi possono essere eseguiti solamente se corredati di password; per altri, basta che giungano da telefoni i cui numeri siano presenti nella lista degli otto abilitati al comando e alla richiesta delle coordinate. La password predefinita dal firmware (e reimpostata automaticamente dopo ogni reset totale del sistema) corrisponde a **12345**; è possibile sostituirla con una a piacimento (purché di cinque cifre) impartendo il comando del caso con un SMS contenente

Una volta salvati i numeri, bisogna definire quali sono quelli cui il localizzatore deve inviare gli SMS contenenti le coordinate della posizione; come accennato, a seguito di una chiamata

Vediamo adesso i comandi che definiscono gli identificativi del localizzatore: un SMS contenente il testo *IMEI?* dà in risposta a chi l'ha mandato il codice IMEI del telefono. Il comando *NAME: xxxx* permette di assegnare un nome all'apparecchio, cosa utile quando si devono gestire più localizzatori. *NAME?* interroga l'apparecchio per ottenere il suo nome identificativo. Tutti questi comandi possono essere inviati da numeri presenti in lista,



ovvero da qualsiasi altro purché corredati di password.

FORS:x stabilisce la rappresentazione degli SMS. I formati impostabili (1 o 2) danno i seguenti risultati:

1:

Probable position:

FT833

LAT: 45.643851

LON: 8.814346

Range: 1159 m

Gallarate VA, Italy

SOS

2:

<http://maps.google.it/maps?f=q&hl=it&q=45.643830,+8.814598>

Gallarate VA, Italy

SOS

Con il comando *REV:1* si visualizza anche l'informazione riguardante l'indirizzo presunto, mentre con *REV:0* l'indirizzo non viene visualizzato.

Negli esempi compare *SOS* in quanto l'invio è stato ottenuto premendo il pulsante *SOS* sul remoto, se invece la richiesta viene effettuata con uno squillo o con un SMS appare:

POLLING

oppure

AUTO_CONT

nel caso di auto report.

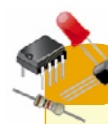
In ogni momento si può interrogare il sistema con il comando *FORS?*, per conoscere quale formato è attualmente impostato. Per quanto riguarda la modalità di autoreport, abbiamo previsto un apposito set di comandi: *AUTO?* permette di verificare a distanza lo stato delle impostazioni per l'autoreport. *AUTOC:xx* abilita o disabilita l'invio periodico delle coordinate

alla lista dei numeri memorizzati e abilitati alla ricezione di SMS; per abilitare l'autoreport scrivete *ON* al posto *xx*, mentre per disattivarlo dovete mettere *OFF*. Il comando *AUTOC:hh/mm* permette di definire la frequenza dell'invio periodico; in esso, al posto di *hh/mm* dovete scrivere ogni quante ore e minuti il localizzatore deve mandare i messaggi con la propria posizione ai numeri definiti nella lista con il solito comando *SMSxxxxxxx:ON*. Ad esempio, se volete un invio ogni 15 minuti dovete impartire il comando *AUTOC:00/15*.

Ora diamo uno sguardo al set di comandi riguardante le impostazioni Internet, indisponibili per accedere al server di Google durante la richiesta della posizione: l'APN si definisce con *GPRSAPN xxxxxxxxxx*, dove al posto delle *x* va inserito il parametro corrispondente al gestore del servizio Internet cellulare; ad esempio per Vodafone è *web.omnitel.it*. All'occorrenza, *GPRSAPN* cancella l'impostazione esistente. *GPRSUSR xxxxxx* serve a impostare il nome utente (username, da scrivere al posto delle *x*) e *GPRSUSR*, invece, cancella l'impostazione attuale. *GPRSPWD xxxxxxxxxx* definisce la password di accesso ad Internet e *GPRSPWD*, invece, cancella la password esistente. Infine, chi desidera verificare le impostazioni Internet può usare il comando *GPRS?*, che in risposta produce un SMS diretto al numero che l'ha richiesto, contenente i dati del caso. Resta inteso che il localizzatore risponde al comando se proveniente da un numero di quelli in lista o da uno qualsiasi a patto che l'SMS inviato contenga la password. In ogni momento, con il comando *COO* è possibile richiedere le coordinate dell'attuale posizione;

esso produce un SMS diretto al numero del telefono che l'ha impartito.

Il reset completo, che ripristina le impostazioni predefinite e cancella la lista dei numeri telefonici, si ottiene con il comando *RES;pwd*, dove *pwd* è la password attuale. Il sistema accetta messaggi multipli, ossia SMS contenenti più comandi ciascuno purché separati da una virgola; volendo, per risparmiarne denaro è possibile disattivare le risposte ad alcuni comandi inserendo, nell'SMS, il comando *RISP*, il quale impedisce la risposta ai comandi che prevedono la conferma tramite SMS. Il localizzatore prevede una procedura di memorizzazione automatica del primo numero telefonico della lista (Easy Setup): dopo l'accensione e per circa tre minuti attende una telefonata, al sopraggiungere della quale salva in memoria il numero del telefono chiamante ed entra nel normale esercizio, dove si porta comunque anche se il tempo passa senza che giunga alcuna chiamata. Per rientrare nell'Easy Setup occorre staccare l'alimentazione e riattaccarla. ■



per il MATERIALE

Tutti i componenti utilizzati in questo circuito sono facilmente reperibili in commercio. Il master del circuito stampato può essere scaricato dal sito della rivista (www.elettronica.in) così come il firmware per programmare il micro. Questo localizzatore è anche disponibile già montato e collaudato (cod. FT833M) al prezzo di 105,00 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 FPGA

Nate alcune decine d'anni fa, rappresentano un'ottima soluzione nelle applicazioni in cui serve svolgere funzioni logiche programmabili. Impariamo a conoscerle.

di
MARIANO SEVERI

FPGA è l'acronimo di *Field Programmable Gate Array*, ovvero, letteralmente, un insieme di porte logiche programmabili; niente è più adatto a dire che un chip FPGA è una vera e propria matrice di elementi più o meno complessi che possono essere configurati per realizzare le funzioni logiche desiderate. Le stesse interconnessioni tra gli elementi possono essere programmate per creare strutture a più alto livello e connettere tra loro i diversi circuiti. Da un certo punto di vista, quindi, le FPGA ereditano il con-

petto di riprogrammabilità proprio del mondo dei microcontrollori: sebbene l'architettura sia la stessa, le funzionalità implementate variano a seconda della particolare applicazione. D'altra parte, però, rifanno il verso agli ASIC nella misura in cui queste stesse funzionalità sono realizzate per via hardware mediante l'interconnessione di porte logiche e non eseguendo un qualche codice applicativo. In queste pagine inizia un corso che vi introdurrà alla conoscenza e all'uso del chip FPGA; in questa prima puntata iniziamo da una breve storia dell'evoluzio-



Fig. 1 - FPGA, programmabilità hardware.



Fig. 2 - Dispositivi FPLA nei primi calcolatori.

ne dei dispositivi logici programmabili, dalle prime PLD alle moderne FPGA; proseguiremo poi con una panoramica sulle principali tecnologie ed architetture oggi disponibili. Infine tratteremo il metodo di progetto che interessa questi dispositivi.

UN PO' DI STORIA

I componenti *FPGA* sono stati introdotti nella seconda metà degli anni '80 del secolo scorso come terza generazione di *PLD* (*Programmable Logic Device*); i *PLD* erano nati a loro volta nella seconda metà degli anni '60, con lo sviluppo presso i laboratori della *Harris Semiconductor* di una 'matrice di diodi configurabile'. Tale matrice consentiva di realizzare semplici funzioni logiche bruciando le connessioni tra i diodi, realizzate da piccolissimi fusibili che in fase di programmazione venivano interrotti facendovi scorrere una corrente ben più elevata di quella di normale funzionamento. Il problema di tali dispositivi è che la matrice aveva dimensioni limitate; inoltre non erano disponibili programmatori reperibili in commercio e quindi l'utente non poteva programmare da sé le *PLD*. La programmazione veniva fatta in fabbrica direttamente alla *Harris* sulla base delle maschere fornite dall'utente, quindi risultava possibile solo avere chip customizzati e non comperare *PLD* da customizzare.

In principio erano le ROM

Nel 1969 i ricercatori della *IBM* svilupparono la memoria *ROAM* (*Read Only Associative Memory*) che consisteva in una matrice di porte logiche, invece che di diodi, unite da interconnessioni programmabili. Nel 1970 la *Texas Instrument* realizzò il primo circuito integrato basato interamente su *ROAM*, coniando il termine *PLA* (*Programmable Logic Array*); il *TMS2000*, ad esempio, aveva 17 ingressi, 18 uscite ed 8 *flip-flop* di tipo *J-K*. Gli elevati

costi di produzione non ricorrenti che l'utente doveva affrontare non favorirono, tuttavia, la diffusione di questo tipo di dispositivi. Più o meno contemporaneamente, la *Harris* introdusse nel mercato le *PROM* (*Programmable Read Only Memory*); oltre che come memorie a sola lettura, questi dispositivi trovarono impiego nella realizzazione di semplici macchine a stati o decodificatori di indirizzo. Era ormai chiara e sempre più pressante l'esigenza dell'industria di avere dispositivi con componenti programmabili a costi contenuti, con i quali realizzare funzionalità logiche specifiche per le varie applicazioni.

Nel 1973 *National Semiconductor* produsse una versione semplificata di componenti *PLA*, denominata *DM7575/DM8575*, caratterizzata da soli 14 ingressi ed 8 uscite e senza elementi di memoria integrati. Questa costituì il prototipo dei dispositivi *FPLA* (*Field Programmable Logic Array*) introdotti in seguito dalla *Intersil* e dalla *Signetics*. Nel giugno del 1975 la *Intersil* annunciò la produzione degli *IM5200*, che introducevano una nuova tecnologia, definita *AIM* (*Avalanche Induced Migration*), per la realizzazione delle logiche programmabili; questa nuova tecnologia si basava su celle nelle quali come elemento programmabile veniva usato un transistor: inducendo una corrente molto elevata tra l'emettitore ed il collettore si produceva un cortocircuito della giunzione base-emettitore, che configura il transistor come un semplice diodo.

Sfortunatamente il nuovo dispositivo non si rivelò molto affidabile e non ebbe diffusione commerciale significativa; una sorte migliore toccò, invece, ai componenti *82S100 FPLA* prodotti dalla *Signetics*, grazie anche all'intensa campagna di promozione e sostegno attuata. I dispositivi *82S100* utilizzavano la vecchia tecnologia delle connessioni fusibili ed erano integrati in un package *DIP* (*Dual Inline Packa-*

ge) a 28 pin di dimensioni intorno ai 15 cm; tuttavia, come tutti i dispositivi *FPLA*, avevano prestazioni limitate per quel che riguardava la velocità di lavoro, erano ancora costosi e ritenuti non facilmente collaudabili. Inoltre le funzioni logiche che si intendeva realizzare in questi dispositivi dovevano essere convertite



Fig. 3 - H. T. Chua e J. Birkner realizzano nel 1978 i primi dispositivi PAL.

in uno standard tabulare definito *H&L*, poco familiare ai progettisti.

La nascita degli array programmabili

Una prima significativa svolta si ebbe nel 1978, con l'introduzione nel mercato dei primi componenti *PAL* (*Programmable Array Logic*) ad opera della *Monolithic Memories, Inc.* La *MMI* veniva da una lunga collaborazione con il *General Electric Research and Development Center* durante la quale aveva sviluppato a fini di ricerca dei dispositivi riprogrammabili denominati *PALA* (*Programmable Associative Logic Array*) basati sulla tecnologia delle *PROM* riscrivibili introdotta nel 1971 dalla Intel. Il successo delle *PAL* fu determinato anche dalla definizione di un linguaggio di programmazione, denominato *PALASM* (*PALA ASseMbler*), simile al *FORTRAN*, che consentiva di descrivere funzioni logiche complesse. Il linguaggio, per la sua semplicità, fu adottato da diverse compagnie produttrici di programmatori di dispositivi *PAL*. A testimonianza della grande diffusione delle *PAL* basti ricordare come, anche dieci anni dopo l'introduzione dei primi componenti, questi rappresentavano ancora la soluzione principale adottata dai progettisti, sebbene già allora fossero commercialmente disponibili oltre

200 *PLD*, ognuna dalle proprie caratteristiche. Tuttavia la *MMI* non fu in grado di adeguare, almeno inizialmente, la produzione di *PAL* alla enorme richiesta. La *Digital Equipment Corporation*, che impiegava questi dispositivi nella realizzazione dei calcolatori *VAX730*, suggerì alla *MMI* lo sviluppo di una versione programmabile a maschera di questi componenti, denominata in seguito *HAL* (*Hard Array Logic*). I componenti *HAL* ebbero un enorme successo che consentì alla *MMI* una rapida espansione. In seguito, gli sforzi della società furono tesi a sviluppare *PAL* dalle prestazioni più elevate, in concorrenza, inizialmente, con la *AMD* (*Advanced Micro Device*). Furono prodotti ad esempio i dispositivi *MegaPAL 64R32* ad 84 pin, che si rivelarono però un insuccesso, in quanto presentavano soltanto piccoli vantaggi rispetto all'utilizzo di un numero maggiore di *PLD* dalle prestazioni inferiori.

L'introduzione della tecnologia CMOS

A quei tempi la tecnologia dominante nello sviluppo dei dispositivi *PAL* era la bipolare, con cui erano state realizzate le prime *PROM*. Si dovette aspettare lo sviluppo degli *EP300* da parte dell'*Altera*, nella primavera del 1984, per avere i primi componenti programmabili realizzati in tecnologia *CMOS*. La nuova tecnologia consentiva di produrre dispositivi a densità di integrazione più elevata, caratterizzati da maggiore complessità e migliori prestazioni, anche se va detto che i nuovi componenti richiedevano lo sviluppo di *tool* di supporto più sofisticati.

Alcuni mesi dopo l'introduzione degli *EP300*, l'*Altera* presentò il sistema denominato *A + PLUS*, per personal computer *IBM*. L'ambiente di sviluppo comprendeva, inoltre, un semplice programmatore che si inseriva in una delle schede di espansione del calcolatore. La strada aperta dall'*Altera* fu percorsa in seguito anche da altre compagnie, quali la *Cypress Semiconductor* in collaborazione con la *MMI*, o la *Lattice Semiconductor* e la *ICT* (*International CMOS Technology*).

Due anni più tardi, usando la tecnologia *CMOS*, *Xilinx* - fondata solo un anno prima da Ross Freeman, Bernie Vonderschmitt e Jim Barnett - irruppe sul mercato con i dispositivi *LCA* (*Logic Cell Array*) *XC2064* che a tutti gli effetti possono essere considerati il primo

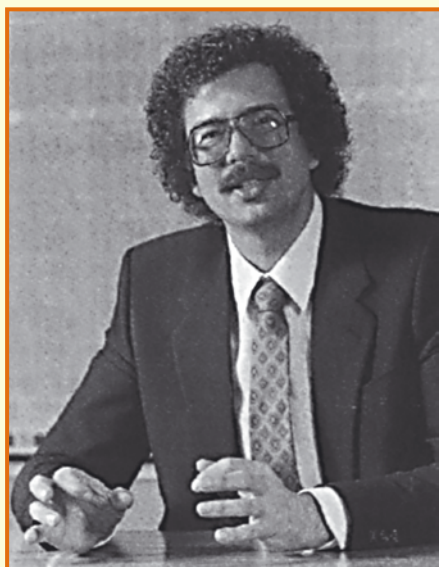


Fig. 4
Nel 1985
Ross Freeman
inventa gli FPGA.

esempio di FPGA. Nel febbraio 2009, le sue invenzioni sono valse a Ross Freeman l'entrata nella *National Inventors Hall of Fame*.

Diversamente dai dispositivi PLD, che presentano un'unica struttura programmabile, le LCA XC2064 si basavano su una matrice di celle configurabili denominate CLB (*Configurable Logic Block*) in grado di realizzare qualunque funzione logica formata da un numero limitato di variabili e su una rete programmabile di interconnessioni tra queste.

Dopo la realizzazione dell'XC2064 l'industria delle FPGA ha avuta una forte espansione, grazie anche ai continui progressi tecnologici nel settore dei semiconduttori, raggiungendo nel 2005 una quota di mercato vicina ai 2 miliardi di dollari. Oggi sono disponibili sul mercato dispositivi di capacità logica equivalente a milioni di gate, che integrano in un singolo chip funzionalità eterogenee quali *core* di processori o transceiver seriali; la tecnologia ha raggiunto i 40 nm, riducendo consumi di potenza e ingombri, incrementando prestazioni ed affidabilità.

Di seguito riportiamo una panoramica delle più significative soluzioni e di alcune delle principali famiglie di dispositivi FPGA oggi presenti sul mercato.

UN UNICO MONDO, DIVERSE TECNOLOGIE

Attualmente per la realizzazione dei dispositivi FPGA vengono utilizzate principalmente tre tecnologie:

- antifuse
- flash
- RAM statica.

Funzionalità ed architetture possono essere in certi casi anche profondamente differenti. Qui di seguito esaminiamo una ad una le possibili tecnologie.

La tecnologia antifuse

I dispositivi *antifuse* sono caratterizzati dalla presenza da tantissimi moduli logici piuttosto semplici, con funzionalità predefinite, che sono connessi tra loro mediante antifusibili. Questi elementi, al contrario dei fusibili, sono normalmente aperti ma diventano dei cortocircuiti quando vi viene fatta scorrere una corrente maggiore di un certo valore di soglia. In questo modo sono abilitate le interconnessioni tra i moduli. I dispositivi *antifuse* sono quindi non volatili, nel senso che la configurazione del dispositivo resta memorizzata anche quando la tensione di alimentazione viene rimossa; però non sono riprogrammabili. Tra i vantaggi principali delle FPGA *antifuse* vi sono una minore dissipazione di potenza, l'operatività immediata all'accensione, minori ritardi di propagazione delle interconnessioni, una riduzione dell'area di silicio occupata; i dispositivi *antifuse* sono inoltre tipicamente più affidabili in applicazioni critiche ed

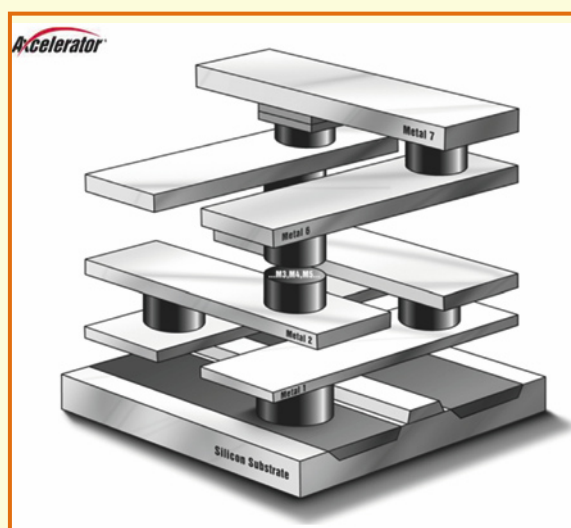


Fig. 5 - Tecnologia antifuse proprietaria per i dispositivi Actel Axcelerator.

assicurano migliore protezione del progetto da tentativi di *hacking* o *cloning*, in quanto non prevedono l'impiego di memorie esterne per la configurazione del dispositivo.

I principali svantaggi della tecnica *antifuse* sono invece l'impiego di un processo di fabbricazione non proprio standard, che ha determinato negli anni una crescita delle prestazioni più lenta rispetto a quanto osservato invece per i dispositivi riprogrammabili in tecnologia RAM statica, una minore capacità logica, la non riprogrammabilità del componente e, di contro, la necessità di programmare il dispositivo prima del montaggio su scheda utilizzando, per di più, programmatori dedicati.

Il principale produttore al mondo di logiche programmabili in tecnologia *antifuse* è la Actel; uno dei suoi prodotti è la serie *Axcelerator*, composta da FPGA di capacità logica fino a 4 milioni di gate, disponibili in package di tipo tradizionale (CSP, PQFP e CQFP) o *grid-array* (BGA, FBGA, CCGA). I moduli logici disponibili sono di tipo R-Cell, che implementano flip-flop con abilitazione ed ingressi asincroni di preset e clear, e C-Cell, che includono, invece, un multiplexer a quattro ingressi e logica di riporto dedicata. Programmando le interconnessioni tra C-Cell è possibile realizzare una qualunque funzione logica di n variabili. I diversi moduli, come mostrato nella Fig. 5, sono diffusi negli strati di silicio, mentre la matrice di interconnessioni è realizzata impiegando i piani di metallo più esterni del "die" (il chip...) con una tecnologia antifusibile proprietaria che consente di ridurre occupazione di area e ritardi di propagazione.

La tecnologia flash

Simile alla *antifuse*, almeno da un punto di vista di principio, è l'architettura delle FPGA *flash*: in esse gli elementi di interconnessioni programmabili si basano su semplici switch realizzati, appunto, in tecnologia *flash*. *ProASIC3*, ad esempio, è la nuova serie di logiche programmabili realizzata dalla Actel con questa tecnologia; elemento fondamentale è, in questo caso, il *VersaTile*, cioè un insieme di semplici elementi logici di base come multiplexer e porte NOT (le cui connessioni possono essere programmate per realizzare una qualunque funzione logica di tre variabili) un latch con ingressi di set e clear o un

flip-flop di tipo D con ingressi di clock_enable e set/reset. Ulteriori switch consentono la connessione delle linee di ingresso/uscita dei *VersaTile* alla rete di distribuzione dei segnali lungo tutta l'FPGA. Ogni switch è realizzato mediante un transistor con *floating-gate*, ovvero materiale conduttivo, rivestito di isolante e disposto tra gate e canale, tipicamente utilizzato per memorizzare la carica; un secondo transistor è invece utilizzato per il read-back e la verifica della configurazione dello switch. Quando è abilitato, lo switch assicura una connessione a bassa resistenza.

La struttura descritta consente una minore occupazione di area rispetto alle soluzioni adottate, invece, nel caso di dispositivi basati su tecnologia SRAM: in questo caso sono tipicamente utilizzati da quattro a sei transistor per ogni elemento programmabile; per questo motivo, le FPGA *ProASIC3/E* possono essere realizzate impiegando un processo costruttivo a minore miniaturizzazione, che risulta quindi più conveniente. L'impiego della tecnologia *flash*, inoltre, assicura l'immunità del dispositivo ai *firm-error*, ovvero ai difetti di funzionamento conseguenza di un cambiamento di stato dell'elemento di configurazione, cambiamento dovuto all'impatto di neutroni ad elevata energia prodotti nelle zone alte dell'atmosfera terrestre. Anche le FPGA in tecnologia *flash* sono non volatili ma a differenza delle *antifuse* possono essere riprogrammate, sebbene la programmazione richieda tensioni accessorie a quelle necessarie per il normale funzionamento; i dispositivi *ProASIC3*, ad esempio, supportano fino a 500 cicli di cancellazione e programmazione e possono essere



Fig. 6 - ProASIC3, FPGA in tecnologia flash: soluzioni single-chip low-power.

programmate in-circuit dopo il montaggio sul circuito stampato. Per proteggere il dispositivo da tentativi di clonazione possono essere usati chiavi di cifratura del bitstream di configurazione; nei casi più critici si può completamente inibire l'accesso alla matrice di configurazione, ma in tale evenienza si perde la possibilità di riconfigurare l'FPGA. La famiglia *ProAsic3* include dispositivi di capacità logica fino a tre milioni di gate, anch'essi in package tradizionali (QN64, VQ100, PQ208) oppure di tipo *fine-pitch grid-array* fino a 896 pin, destinati ad applicazioni che richiedano un elevato numero di segnali di ingresso/uscita. Estremamente interessanti, inoltre, sono le versioni *ARM-enabled*, che vengono distribuite con una sezione del dispositivo già programmata per implementare un processore ARM di tipo *CoreMP7* o *Cortex-M1*; il progettista può utilizzare la restante parte di risorse per realizzare la propria logica utente che diventa quindi una periferica accessibile da parte del processore.

La tecnologia SRAM

Piuttosto diversa è, invece, la struttura dei dispositivi logici programmabili basati su tecnologia SRAM; infatti in questo caso le funzioni logiche non vengono realizzate connettendo tra di loro delle *primitive* ma impiegando LUT (*Look-Up Table*). Una LUT è una memoria statica a n bit di indirizzamento; data una funzione logica di n variabili, e nota la sua tabella della verità, il valore della funzione per una qualsiasi combinazione degli ingressi viene programmato nella LUT all'indirizzo corrispondente a quella stessa combinazione di ingressi. Ad esempio nel caso di una funzione a 4 ingressi che restituisce "1" quando i suoi ingressi valgono "1001", la LUT sarà programmata con il valore "1" all'indirizzo $0b1001 = 0d9$. In questo modo, una LUT ad n ingressi può realizzare una qualsiasi funzione logica di n variabili. Diversamente, nel caso dei dispositivi *antifuse* e *flash*, il numero di *primitive* connesse e le loro modalità di connessione dipendono dal tipo di funzione che si intende realizzare. Quindi nelle FPGA in tecnologia SRAM,



Fig. 7 - Xilinx Virtex-5: le prime FPGA in tecnologia triplo-ossido a 65 nm con processore PowerPC440, end-point PCIe ed Ethernet MAC embedded.

per creare le LUT vengono utilizzate celle di memoria riprogrammabili. Una fitta rete di linee di distribuzione di segnali consente poi di connettere le LUT tra loro; ulteriori celle statiche vengono utilizzate per collegare gli ingressi/uscite delle LUT alla rete di interconnessioni. Le FPGA basate su RAM statica sono realizzate con una tecnologia CMOS standard che consente di integrare piuttosto facilmente anche elementi diversi; ad esempio i moderni dispositivi incorporano estesi blocchi di memoria RAM embedded, circuiti di riporto veloci per la realizzazione di addizionatori, blocchi DSP ed accumulatori per applicazioni di elaborazione digitale dei dati, interfacce SerDes ad elevato data-rate (dell'ordine dei gigabit/s), controller Ethernet e *core* di processori embedded. I vantaggi principali dei dispositivi SRAM sono l'elevata capacità logica, l'elevato livello di integrazione, le migliori prestazioni in termini di frequenza di funzionamento nella maggior parte delle applicazioni, i costi minori a parità di funzionalità e risorse. Non meno importante è la possibilità di riprogrammare i dispositivi, che in fase di debug consente di fissare velocemente errori di progettazione e che in applicazioni su campo permette di modificare le

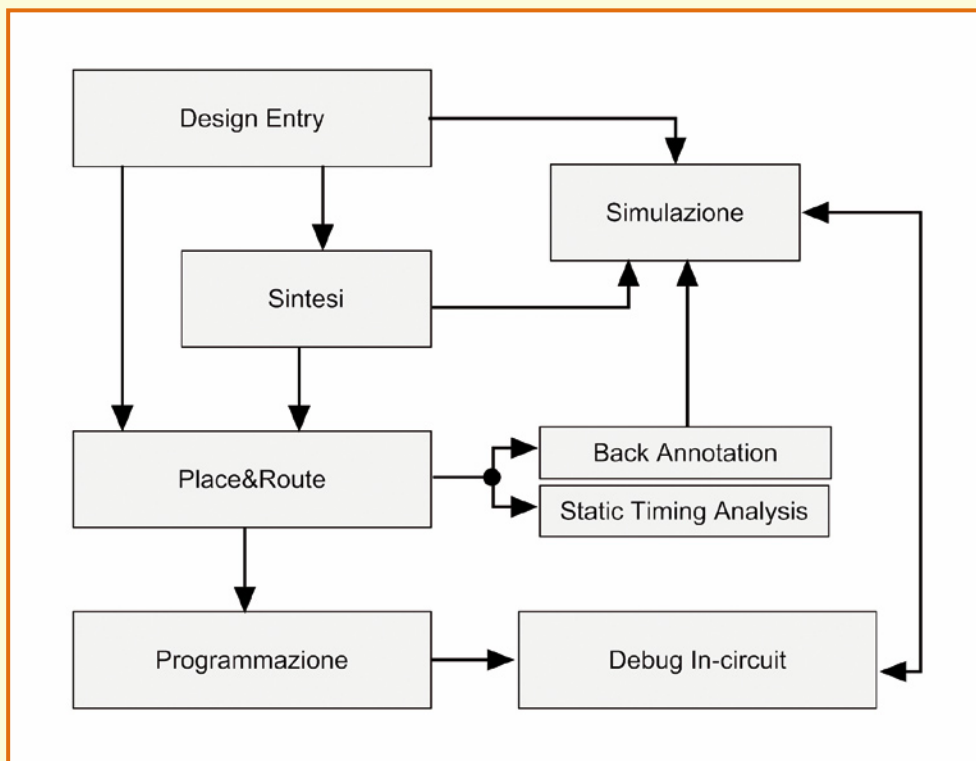


Fig. 8 - Il flusso di progetto con FPGA.

funzionalità del dispositivo in base ai diversi scenari operativi. Le FPGA di tipo SRAM sono dispositivi volatili e quindi per poter essere usate, ad ogni accensione richiedono il caricamento della configurazione da una memoria esterna; anche in questo caso sono previsti meccanismi di cifratura del bit-stream di configurazione per evitare clonazioni del dispositivo nelle applicazioni critiche. In funzione della complessità e del tipo di dispositivo, i tempi di programmazione possono variare da pochi millisecondi fino ad oltre un secondo. Il dispositivo quindi non è *live* al *power-up*, il che potrebbe non essere accettabile in alcune applicazioni critiche. Minore è poi la tolleranza ai *firm-error* e maggiore la dissipazione di potenza. I principali produttori al mondo di logiche programmabili in tecnologia SRAM sono *Xilinx* e *Altera*. I nuovi dispositivi *Virtex-5* di *Xilinx*, in particolare, utilizzano LUT a sei ingressi ed integrano fino a 11 Mbit di RAM, transceiver *RocketIO GTP* fino a 3,75 Gbps che supportano, ad esempio, i principali standard di comunicazione seriale per applicazioni telecom (OC-48, OBSAI), video (HD-SDI) o computing (*PCIexpress*, *SRIO*, *Infiniband*); integrano inoltre fino a 4 *Endpoint PCI-Express* e 8 *Ethernet MAC*, monitor di

sistema e convertitore analogico-digitale per misure di parametri vitali come temperatura del chip, tensioni e correnti assorbite. Incorporano anche fino a due processori *IBM PowerPC 440*. I dispositivi della serie *Spartan-3*, invece, rappresentano la soluzione per i sistemi low-cost e le produzioni di massa nella maggior parte delle applicazioni consumer. *Stratix-IV* e *Cyclone-III*, infine, sono i dispositivi concorrenti distribuiti da *Altera*.

IL FLUSSO DI PROGETTO

Il flusso di progetto riguardante la programmazione di circuiti contenenti FPGA segue uno schema di principio che tipicamente è quello rappresentato nella **Fig. 8**. Le operazioni del caso sono divise nelle fasi che esamineremo qui di seguito.

Design Entry

La prima fase è quella di *Design Entry*, che consiste nella descrizione delle funzioni logiche che il dispositivo dovrà implementare. Esistono per questo diverse modalità che vanno dal disegno dello schema circuitale in ambienti CAD appositi ed utilizzando *primitive* di base come flip-flop e porte logiche, a descrizioni di più alto livello basate su diagrammi a

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity counter is
  generic ( c_width : integer := 8 );
  port
  (
    arst : in std_logic;
    clk  : in std_logic;
    ce   : in std_logic;
    q    : out unsigned(c_width-1 downto 0)
  );
end entity counter;

architecture rtl of counter is
  signal q_i : unsigned(c_width-1 downto 0);
begin

  Cnt : process (arst, clk)
  begin
    if arst='1' then
      q_i <= (others=>'0');
    elsif clk'event and clk='1' then
      if ce='1' then q_i <= q_i+1; end if;
    end if;
  end process Cnt;

  q <= q_i;

end rtl;

```

Fig. 9 - Descrizione di un semplice contatore in VHDL.

bolle o linguaggi di descrizione hardware. Tra questi ultimi, i due più diffusi sono il *Verilog* ed il *VHDL*. Nella prossima puntata vedremo con un semplice esempio come descrivere un circuito elettrico in *VHDL*; a titolo di semplice curiosità, nella riquadro illustrato nella **Fig. 9** è riportata la descrizione di un contatore di tipo *up* con ingresso di abilitazione e reset asincrono. Mentre agli inizi il disegno degli schemi elettrici era la metodologia principale, oggi la progettazione di FPGA è praticamente interamente basata sui linguaggi di descrizione hardware. In effetti, la crescente complessità dei dispositivi logici programmabili e dei sistemi in essi implementati, unitamente

alla necessità di ridurre i tempi di sviluppo, sta spingendo sempre più all'utilizzo di metodi e linguaggi di alto livello. Da questo punto di vista, ad esempio, esistono librerie e supporti per la descrizione di circuiti logici in *Matlab/Simulink* con generazione automatica di codice *HDL*. Più o meno allo stesso modo, *Catapult*, un tool di *Mentor Graphics*, consente di generare automaticamente una descrizione *HDL* a partire da una in linguaggio C++; tali soluzioni sono principalmente utilizzabili per algoritmi di elaborazione dati.

Il *SystemC*, invece, è una libreria per C++ che introduce in questo linguaggio i concetti di tempo, parallelismo e concorrenza e permette quindi di creare modelli di circuiti logici; un sottoinsieme dello standard che risulti traducibile automaticamente in porte logiche e registri è in fase di standardizzazione.

Simulazione

Come mostrato nella **Fig. 8**, alla fase di *design entry* seguono le attività di simulazione e verifica funzionale del circuito; obiettivo è verificare che la descrizione che se ne è fatta corrisponda effettivamente al comportamento atteso per il circuito. In maniera piuttosto semplicistica, si tratta di stimolare gli ingressi del nostro circuito e verificare che i segnali di uscita seguano l'andamento atteso. Esistono diversi tool che consentono di interpretare le descrizioni dei circuiti in linguaggio *HDL* per simularne il comportamento. Due dei più diffusi sono *ModelSim* di *Mentor Graphics* e *ActiveHDL* di *Aldec*. La **Fig. 10** mostra ad esempio l'ambiente di lavoro di *ModelSim* con le diverse finestre disponibili durante una simulazione; sono previste, ad esempio, una finestra di editor delle forme d'onda dei segnali che consente di visualizzarne l'andamento ed una di debug che consente di eseguire invece passo passo il codice che rappresenta il circuito. Descrizioni in linguaggio *SystemC* possono invece essere simulate all'interno di un tradizionale catena di debugger C++ come quella *GNU/GCC* o *Microsoft Visual Studio*.

Sintesi

La fase successiva nel flusso di progetto con FPGA prevede la traduzione della descrizione del circuito in un formato che i tool proprietari della case produttrici di FPGA siano in grado

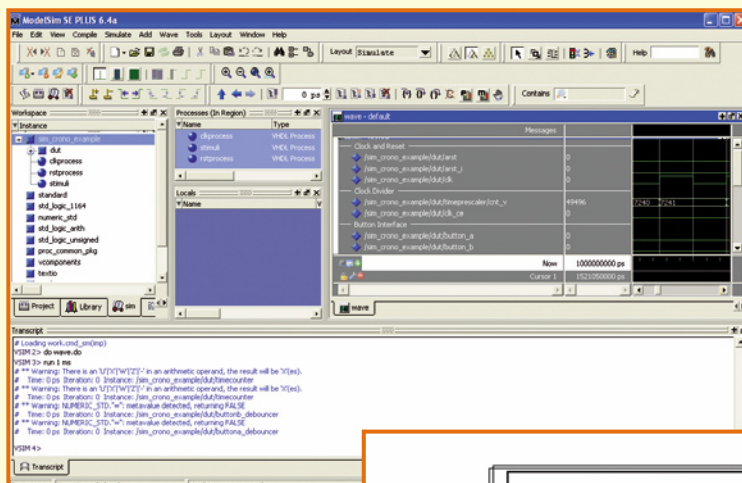
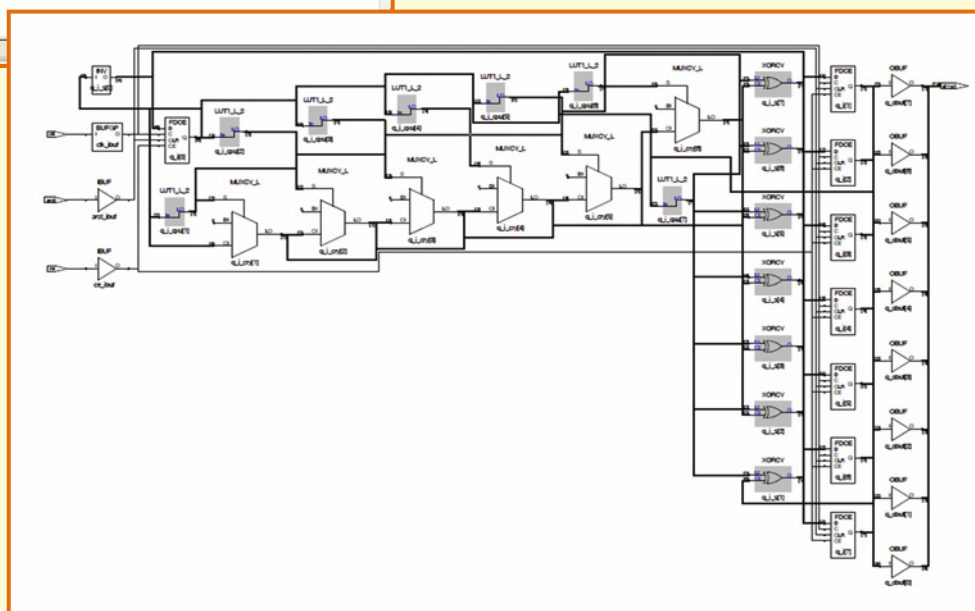


Fig. 10 - Simulazione VHDL in ModelSim.

Fig. 11 - Sintesi di un contatore con Synplify Synplify.



di interpretare per poter generare alla fine, il bitstream di programmazione. Nel caso si abbia a che fare con descrizioni fatte in linguaggio HDL tale fase viene altrimenti indicata come *sintesi* del circuito. Esistono diversi tool di *sintesi*, tra cui Leonardo Spectrum e Precision RTL di Mentor Graphics o Synplify di Synplicity; a questi si affiancano tool proprietari dei produttori di FPGA come ad esempio XST di Xilinx o Quartus di Altera. Il processo di sintesi astrae, in definitiva, le funzionalità logiche del circuito dalla descrizione e le traduce in porte logiche, flip-flop e connessioni tra questi; in **figura 11** è mostrato ad esempio il circuito sintetizzato con Synplify per il contatore la cui descrizione è stata presentata in precedenza (**figura 9**). La descrizione in termini di *primitive* logiche viene indicata con il termine di *netlist*; esistono diversi formati, di cui alcuni sono standard, altri proprietari. Tra i più diffusi vi è il formato EDIF (Electronic Design Interchange Format). Oltre a generare la *netlist*, il processo di *sintesi* fornisce una indicazione preliminare sulle risorse occupate nel dispositivo target e della frequenza di funzionamento massima che si potrà ottenere; permette inoltre di generare automaticamente una nuova descrizione

in linguaggio HDL del circuito che si basi direttamente su porte logiche e segnali di interconnessione. Ove lo si ritenga opportuno, è possibile simulare tale descrizione per verificare che il circuito corrispondente sia corretto. In realtà, i moderni tool sono diventati piuttosto affidabili e, supposto che non vi siano errori nella descrizione di partenza, solo in casi rari (legati principalmente a particolari stili di codifica del codice HDL di partenza da parte del progettista) i risultati della *sintesi* non sono corretti.

Place&Route

La fase successiva a quella di *sintesi* è denominata *Place&Route* e serve ad associare le porte e le funzioni logiche presenti nella *netlist* alle risorse del dispositivo, definendo poi i *path* di connessione tra queste; è possibile fornire dei *constraint* sulla disposizione delle risorse

Fig. 12 - Implementazione di un contatore in FPGA (in azzurro le risorse occupate).

utilizzate all'interno della FPGA, sulla associazione dei segnali di ingresso/uscita ai pin del dispositivo o, ancora, sulla massima frequenza di funzionamento che si intende raggiungere. I tool di *place&route* usano algoritmi di natura casuale per il piazzamento ed il routing delle *primitive*; il processo quindi non è mai ripetibile.

Anche se la *netlist* è la stessa, è del tutto improbabile che il layout della FPGA al termine del processo di *place&route* appaia lo stesso in due diverse implementazioni.

Successivamente alla fase di *place&route*, viene creato il database con i ritardi dei singoli segnali. Tale database può essere estratto in un file di formato standard SDF (*Standard Delay Format*) unitamente ad una descrizione del circuito in linguaggio HDL basata direttamente su modelli delle *primitive* del dispositivo cui possono, appunto, essere associati i ritardi di propagazione stimati dei segnali.

Il layout della FPGA può essere analizzato e verificato rispetto ai parametri temporali imposti (massima frequenza di funzionamento, rispetto dei *constraint* di *setup/hold* e *clock-to-output*) ed ai requisiti in termini di dissipazione di potenza.

Programmazione e debug-in-circuit

L'ultima fase del processo prevede la generazione del file di programmazione della FPGA. Come abbiamo visto in precedenza, per i dispositivi *antifuse* dovrà essere usato un apposito programmatore per programmare il dispositivo.

Nel caso invece dei dispositivi *flash*, il file di programmazione viene convertito in altri formati e solitamente mediante cavo parallelo connesso al PC scaricato nella memoria esterna di configurazione del dispositivo. Una volta programmato il dispositivo, non resta che verificarne nella realtà il corretto funzionamento.

Spesso, a seguito della crescente complessità dei sistemi implementati in logiche programmabili, si è determinata negli ultimi anni la necessità di utilizzare metodologie di

debug in circuit per risolvere bug di progettazione. Il problema, in effetti, è che proprio a causa di questa complessità è praticamente impossibile coprire tutti i casi di utilizzo del dispositivo in simulazione.

Con il debug-in-circuit, allora, si ha la possibilità di vedere l'andamento nel tempo dei segnali della FPGA durante il reale funzionamento di questa.

Analizzando questi è possibile scoprire l'origine del malfunzionamento e replicarlo in simulazione per risolverlo; in fondo è come avere la possibilità di simulare nella realtà il comportamento del circuito.

Esistono diversi tool di debug in circuit tra i quali *ChipScope* di Xilinx oppure *Identify* di Synplicity.

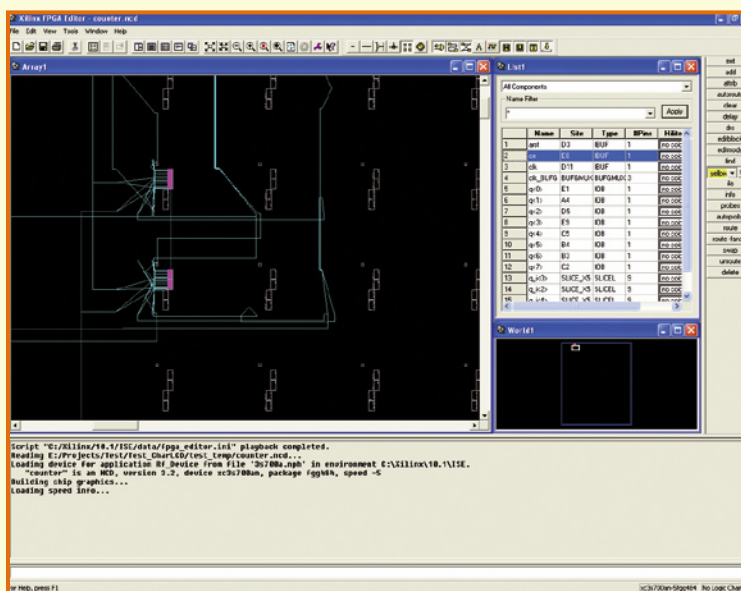
CONCLUSIONI

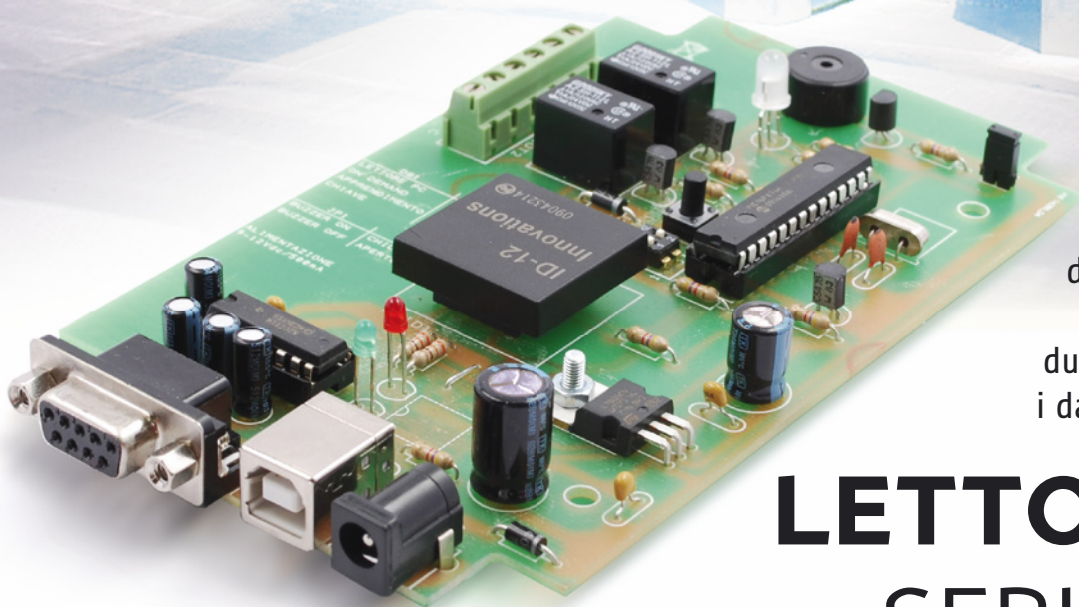
Come abbiamo visto, le FPGA rappresentano una delle principali innovazioni che hanno coinvolto il mercato dei semiconduttori negli ultimi vent'anni.

L'introduzione delle logiche programmabili ha definitivamente modificato il modo di progettare i sistemi digitali.

L'articolo ha presentato una introduzione al mondo delle FPGA, alle modalità di utilizzo ed al flusso di progettazione che viene seguito per il loro impiego.

Nella prossima puntata approfondiremo l'architettura delle FPGA Xilinx *Spartan-3AN* e presenteremo un kit di sviluppo che può essere impiegato per iniziare a progettare con questi dispositivi. ■





Al riconoscimento di uno dei 120 transponder memorizzabili comanda due uscite a relé e trasmette i dati via porta seriale o USB.

LETTORE RFID SERIALE/USB

..... di ALESSANDRO SOTTOCORNOLA

Come hanno dimostrato i numerosi progetti da noi pubblicati, non ultimo quello nel fascicolo di giugno scorso, sovente un controllo accessi può essere realizzato con un semplice lettore di transponder stand-alone, in grado di gestire autonomamente un'elettroserratura o un segnalatore; esistono però casi in cui non basta semplicemente dare o negare il consenso al passaggio di una persona ma occorre identificare, tramite il tag che

porta con sé, di chi si tratta. In questo caso ci vuole un lettore in grado sia di gestire autonomamente l'accesso, sia di comunicare a un operatore o registrare in un PC un log dei passaggi. Il progetto qui descritto nasce proprio con il duplice intento sia di regolamentare gli accessi e gestire direttamente un tornello o un cancello elettrico, sia di informare un operatore e/o registrare i dati inerenti ai passaggi; a tal proposito dispone sia di relé attivabili automati-

camente al riconoscimento di uno dei tag preventivamente memorizzati, sia di una connessione seriale ed USB per l'interfacciamento con un qualsiasi Personal Computer, dove un apposito software permetterà di elaborare i dati, identificare chi passa ed eventualmente subordinare la gestione dei relé e delle elettroserrature o dei cancelli e tornelli

del sistema in modalità PC Link *On Demand* e *Lettore*.

SCHEMA ELETTRICO

Prima di illustrare le funzioni previste, diamo subito uno sguardo al circuito che le realizza; il relativo schema elettrico è decisamente semplice, grazie all'adozione di un modulo che da solo gestisce la lettura del

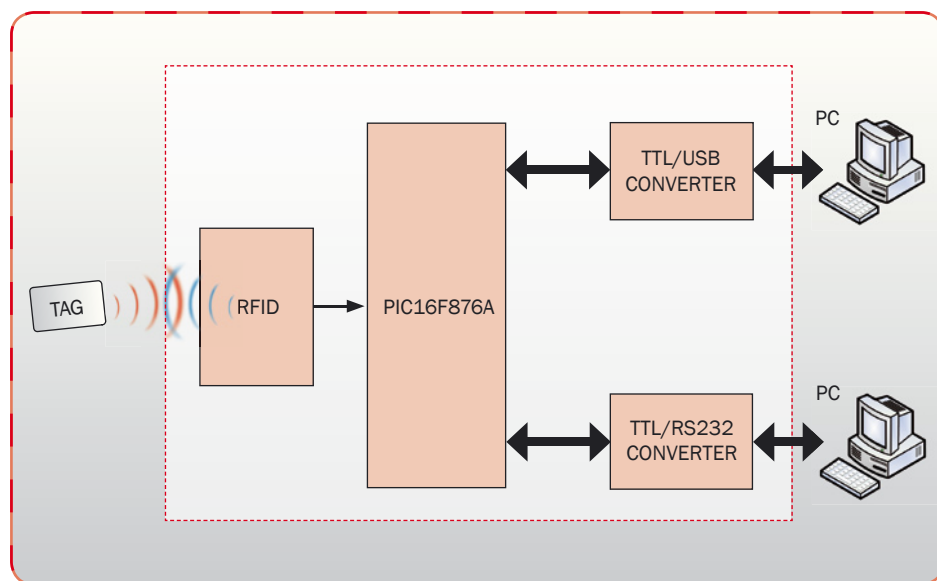
Basta alimentarlo e, quando gli si avvicina un tag, fornisce in uscita i dati rilevati. Per realizzare una serratura elettronica o altro dispositivo di identificazione e discriminazione di transponder per RFID, basta aggiungere un microcontrollore che, nel caso di dispositivi stand-alone, provveda alla memorizzazione e all'identificazione e per i sistemi basati su computer, trasformi i dati letti nel formato riconoscibile dai PC (ad esempio converta le stringhe di dati in caratteri ASCII).

Nel nostro circuito l'ID-12 è interfacciato con un microcontrollore che provvede, nel funzionamento autonomo, all'apprendimento dei tag e al loro riconoscimento, quindi alle azioni correlate all'identificazione di uno dei transponder presenti in memoria; nella gestione da computer, legge i dati e li prepara per l'invio al PC e si occupa di identificare i comandi in arrivo da quest'ultimo, provvedendo alla modifica della lista dei tag appresi e all'impostazione dei relé di uscita.

Il micro utilizzato è un PIC16F876A che si occupa di elaborare i dati in arrivo dall'ID-12 (siglato, nello schema elettrico, RFID1) e di inviarli via seriale (mediante il convertitore TTL/RS232 U2) e su USB (tramite U4) ad un computer; oltre a ciò, il micro gestisce i due relé di uscita e un cicalino per segnalare acusticamente, all'occorrenza, il riconoscimento di un tag.

All'inizializzazione, il firmware del PIC imposta RC4 ed RC5 come uscite dedicate al comando dei relé (mediante i transistor T1 e T2, usati come interruttori statici), RC3 ancora come uscita, ma per il comando del cicalino, inseribile mediante JP1; anche le linee RC1 ed RC2 sono settate come uscite e vengono usate per far illuminare i led contenuti

Schema a blocchi

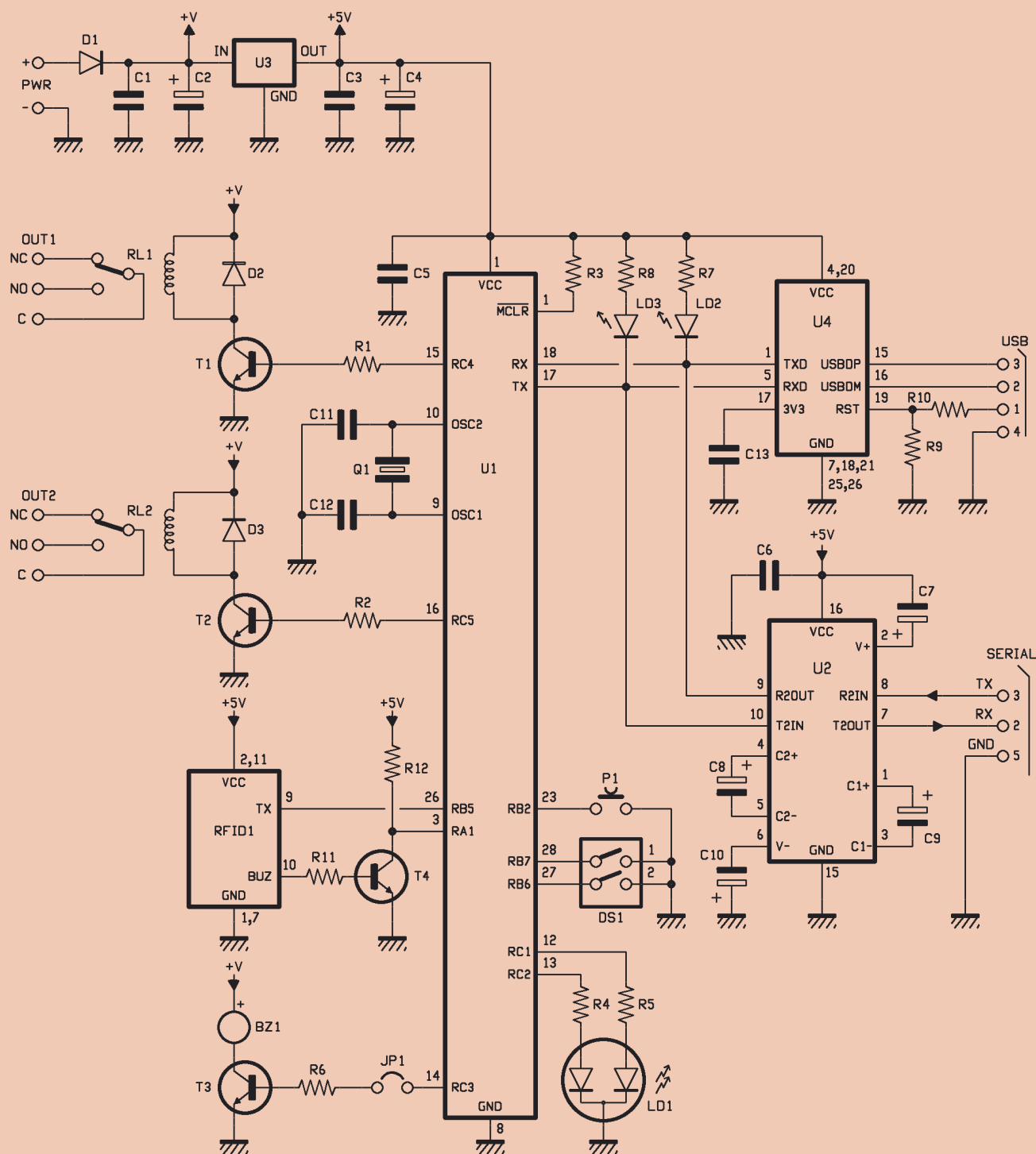


CARATTERISTICHE TECNICHE

- 3 modalità di funzionamento
- 120 codici memorizzabili in EEPROM
- segnalazione lettura tessera mediante buzzer disinseribile
- distanza rilevamento tag: 6÷8 cm
- 2 relé con funzione differente a seconda della modalità operativa
- interfacciamento Seriale o USB
- tensione di alimentazione 9÷12 Vcc, 300 mA max
- software di gestione per la modalità "Lettore" e "On Demand"

motorizzati al consenso da parte di un operatore addetto alla vigilanza o di un programma. Il software a corredo (scaricabile dal nostro sito web www.elettronica.it) che descriveremo tra breve, permette una facile gestione

transponder: si tratta dell'ID-12 della Innovations Technology, un componente contenente il circuito che genera il campo magnetico eccitatore, la bobina irradiante, il ricevitore, l'amplificatore e il discriminatore dei dati.

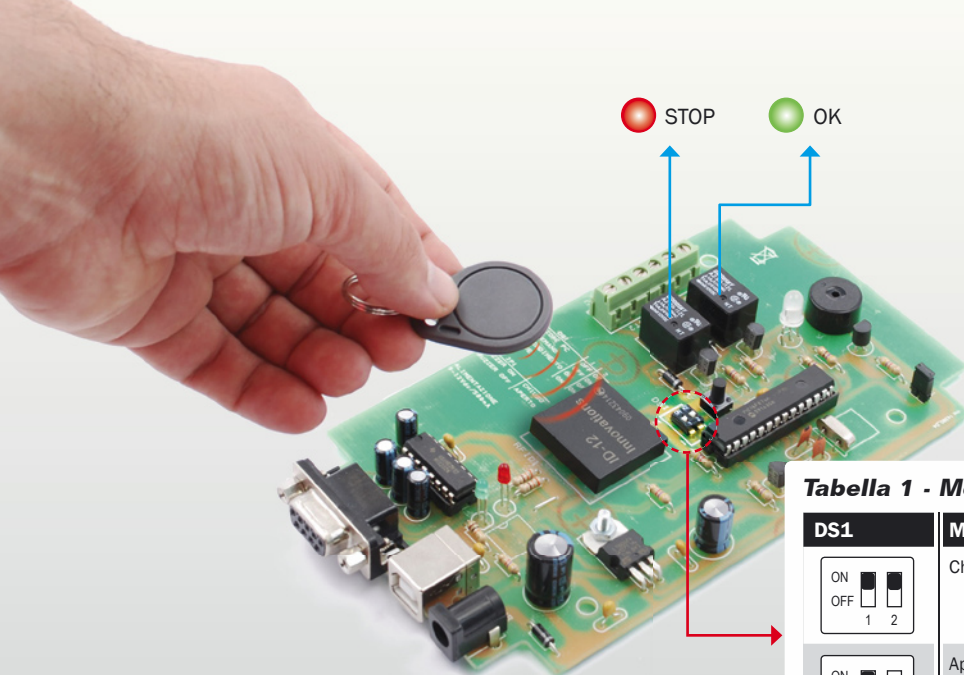


nel diodo bicolore LD1. Sempre all'inizializzazione, l'UART interno al microcontrollore viene assegnato al dialogo con l'USB ed è interfacciato con il canonico FT232RL, un convertitore TTL/USB comprendente anche il gene-

ratore di clock e la logica di conversione dei segnali; l'FT232RL è un adattatore da RS232 (o TTL con dati in formato RS232) a USB 2.0 e viceversa, funzionante a 5 volt. L'alimentazione a 5 volt viene qui prelevata non dal-

l'USB, che pure potrebbe fornirla (ricordiamo che il bus consta di quattro linee, ossia D+, D-, +5 V e massa) ma dal regolatore del circuito che fa funzionare anche il microcontrollore. L'integrato della FTDI dispone

Impostiamo la modalità di funzionamento



L'unità è molto versatile e può funzionare da semplice serratura indipendente (stand-alone) con un relé di apertura ed uno di allarme (segnalazione tag sconosciuto) oppure da lettore asservito ad un computer; ancora, può fare da semplice interfaccia RFI per un PC dove si registrano log dei tag rilevati.

Tabella 1 - Modalità di funzionamento.

DS1	MODALITÀ	DESCRIZIONE
ON OFF 1 2	Chiave stand-alone	Attiva/Disattiva il relé 1 o 2 a seconda che il codice transponder è presente in memoria oppure no.
ON OFF 1 2	Apprendimento/Cancellazione	Permette di abilitare/disabilitare la tessera desiderata per fare sì che attivi o meno il relé 1 o 2 in modalità chiave stand alone.
ON OFF 1 2	PC Link On Demand	È il PC a gestire interamente, il lettore. Il codice transponder letto viene memorizzato in EEPROM del PIC.
ON OFF 1 2	PC Link Diretto	Il dispositivo funziona solo come lettore transponder, il codice viene inviato al PC.

internamente un regolatore di tensione che ricava i 3,3 V necessari alla propria logica; lo stesso potenziale viene reso disponibile per altri circuiti, dal piedino 17, al quale, nel nostro caso, è collegato il condensatore di filtro C13, utile a filtrare la tensione interna. I pin CBUS0 e CBUS1 vengono programmati in fabbrica dal costruttore per gestire i due led di stato, ma nel nostro circuito non vengono utilizzati. Lo stesso UART del microcontrollore dialoga parallelamente con la RS232, mediante il converter MAX232, perché le linee RX e TX sono collegate in parallelo ai rispettivi canali di comunicazione dell'U2 e dell'U4. Tuttavia ciò non significa che il nostro circuito possa comunicare simultaneamente su una porta USB e su una RS232, nel senso che se è attiva una, l'altra non può essere usata; ciò perché l'unità non può dialogare contemporaneamente con due computer diversi, che

imporrebbero temporizzazioni differenti. Quindi se la comunicazione avviene sull'USB non è permesso il dialogo su RS232 e viceversa.

Il MAX232 lavora nella tipica configurazione, con i condensatori elettrolitici C7, C8, C9 e C10 che servono ai circuiti interni a pompa capacitiva per ricavare, partendo dai 5 volt, le tensioni di ± 12 V con cui viene fatta funzionare la sezione d'uscita RS232. Bene, finora abbiamo esaminato gli I/O inizializzati come uscite; all'accensione, però, il PIC programma alcune linee come input: si tratta della RB5, impiegata per leggere i dati in arrivo dall'ID-12, della RB2, provvista di pull-up interno e usata per leggere lo stato del pulsante P1 e delle RB6 ed RB7, qui sfruttate per la lettura dei dip-switch contenuti in DS e anch'esse settate con il pull-up interno.

Quanto alla RA1, è anche questa un input e viene sfruttata dal mi-

cro per sapere quando attivare il canale dati e comunicare con l'ID-12; infatti sarebbe stato un po' problematico far girare in loop il main-program attendendo i dati dal modulo perché ciò avrebbe significato impegnare il micro costantemente in una routine. Invece abbiamo fatto in modo che il programma del PIC vada a leggere RB5 solamente quando riceve il trigger, nel nostro caso ottenuto sfruttando quell'uscita dell'ID-12 normalmente prevista per il comando del segnalatore acustico. Infatti il modulo della Innovations Technology prevede un'uscita per pilotare un cicalino senza oscillatore (una comune pastiglia piezoelettrica) o un led quando viene riconosciuto il codice di un transponder codificato secondo gli standard che esso supporta (attenzione, che non è l'ID-12 a riconoscere i tag memorizzati: l'uscita si attiva solo se il tag è standard, ovvero se il codice che fornisce rientra in

uno degli standard gestibili dal modulo...); sfruttando il segnale di tale pin (il 10) informiamo il micro che deve andare a leggere i dati dalla linea TX del modulo RFID1. Notate che l'uscita BUZ dell'ID-12 è normalmente a livello basso e quando viene identificato un transponder pulsa rapidamente per un secondo generando un segnale ad audiofrequenza; il micro filtra tale segnale e ricava l'informazione voluta. Altra cosa è il cicalino BZ1, che verrà attivato, sempre che il JP1 sia stato chiuso) quando il microcontrollore lo deciderà, ovvero in apprendimento per confermare la lettura e l'acquisizione di un tag e nel normale utilizzo quando un tag viene riconosciuto come uno di quelli memorizzati. Come accennato, l'ID-12 è un

prestante lettore RFiD in grado di funzionare autonomamente e di identificare le sequenze di codici secondo gli standard usati a livello internazionale; dispone di una porta seriale TTL-compatibile (a livello 0/5 V) attraverso la quale (dal piedino TX) invia, sequenzialmente, i caratteri presenti nei codici letti, affinché un dispositivo a microprocessore o un computer possa leggerli ed elaborarli. Il riconoscitore di transponder funziona nella sua tipica configurazione, dove l'uscita dati (TX) viene connessa all'ingresso del convertitore TTL/USB o TTL/RS232; l'uscita BUZ viene usata per triggerare il microcontrollore. Si noti che il contatto 10 dell'ID-12 genera non un semplice livello logico alto, ma una serie di impulsi: normal-

mente a zero, pulsa alla frequenza di 3 kHz quando viene letto e identificato un transponder.

COME FUNZIONA

Per rendere il dispositivo il più versatile possibile abbiamo previsto diverse modalità di funzionamento e, in particolare, la capacità di gestirsi in modo autonomo o di affidarsi al computer; più esattamente, sono possibili tre modalità di esercizio: *chiave stand-alone*, *PC Link On Demand* e *PC Link Diretto*.

Nella prima modalità il circuito funziona del tutto autonomamente e da lettore: quando si avvicina un tag al modulo lettore ID-12, se è del tipo riconosciuto, il transistor T4 commuta rapidamente per un

Il lettore ID-12

Il componente impiegato nel nostro lettore RFiD rappresenta una piccola rivoluzione nel settore della progettazione di dispositivi basati su transponder: si tratta infatti di un integrato ibrido che contiene in un unico case non solo il sistema eccitatore e rilevatore (la bobina, insomma...) della risposta dei transponder, ma anche un microprocessore in grado di decifrare i dati rilevati, analizzarli e verificarne la compatibilità con gli standard in uso a livello internazionale nell'RFiD.

È quindi un lettore single chip, capace di sbrigare da sé tutto quel che serve e mandare lungo una porta seriale TTL solamente le stringhe corrispondenti alle letture valide, ossia alle risposte di transponder passivi codificati secondo gli standard ASCII, Wiegand26 e Magnetic ABA Track2. Proprio perché provvede da solo al riconoscimento dei transponder, l'ID12 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 affrancando il

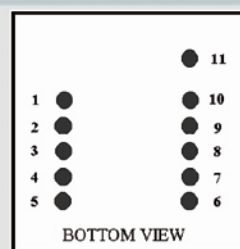
computer o microcontrollore che deve leggerne i dati dai compiti di decifrazione ed eliminazione: il componente legge i transponder e scarta i dati di quelli fuori formato o letti male, passando al computer solo le stringhe integre.

Il nostro componente si presenta come un parallelepipedo con 11 piedini sfalsati a passo dip doppio; internamente ha, oltre al microprocessore e all'oscillatore a 125 kHz, una bobina che irradia il campo magnetico e rileva il carico determinato dai transponder. Proprio perché contiene la bobina, il componente deve essere montato lontano da pareti metalliche; inoltre, nel disegnare lo stampato bisogna evitare che abbia sotto di sé un grande piano di massa (le piste devono essere solo quelle che servono e avere percorsi che siano molto corti). Gli undici piedini hanno

ciascuno il significato descritto nel prospetto qui sotto; in particolare, il pin 7 è quello che definisce il formato dei dati da riconoscere e va disposto a zero logico se si desidera che l'ID12 legga transponder ASCII, collegato al piedino 10 per identificare dispositivi Magnet ABA Track2, ovvero alla linea di alimentazione per leggere transponder Wiegand26.



ID2 / ID12 / ID20 PIN-OUT



1. GND
2. RES (Reset Bar)
3. ANT (Antenna)
4. ANT (Antenna)
5. CP
6. Future
7. +/- (Format Selector)
8. D1 (Data Pin 1)
9. D0 (Data Pin 0)
10. LED (LED / Beeper)
11. +5V

secondo e il micro si predispone alla lettura dei dati che RFID1 gli passa lungo il canale TX; letti i dati, il microcontrollore estrae il codice e lo confronta con quelli in memoria. A questo punto possono accadere due cose: se il codice non è di quelli dei transponder appresi, il micro attiva l'uscita di allarme, corrispondente al RL2, per un secondo; dunque, questo relé pulsa e torna a riposo dopo un secondo. Ciò permette di segnalare l'allarme: ad esempio, in un controllo accessi può avvertire localmente il custode o la guardia alla porta con una segnalazione acustica o visiva o con entrambe, del fatto che qualcuno ha tentato di passare con un badge scaduto o non idoneo. Se invece il transponder è uno di quelli già appresi, 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.

E passiamo alla modalità *PC Link*

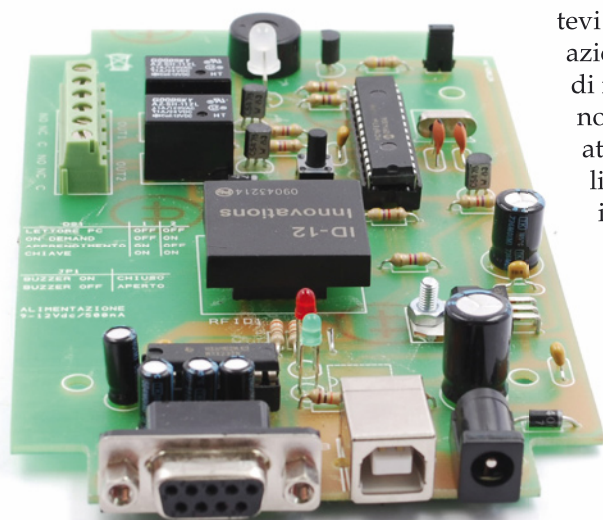
On Demand, ossia al funzionamento demandato al computer: in questo caso il lettore può compiere numerose operazioni e diventa a tutti gli effetti un terminale del PC; le sue funzioni vengono gestite dall'apposito software che descriveremo tra breve, quindi per ora accontenta-

caso, ovvero trasmette al circuito il comandi di attivazione del RL2 se il transponder non è tra quelli memorizzati nel database, mentre manda il comando di attivazione del RL1 se il transponder è, invece, uno di quelli presenti nello stesso database.

Ora, vediamo l'ultima modali-

Tabella 2 - Segnalazioni dei LED.

LED	MODO	DESCRIZIONE
LED 1	Verde lampeggiante ogni secondo	- Dispositivo in funzione.
	Verde fisso	- Indica l'attivazione del relé 1 in modalità "On Demand" e "Chiave" (Tessera Autorizzata).
	Rosso fisso	- Indica l'attivazione del relé 2 in modalità "On Demand" e "Chiave" (Tessera non Autorizzata); - Indica in modalità "Apprendimento", durante la fase di memorizzazione, che il codice è già presente in memoria.
	Rosso fisso all'avvio del dispositivo	- Cancellazione totale della memoria.
	10 lampeggi rossi	- Indica la memoria piena in fase di inserimento nuovo codice in memoria.
	Arancione lampeggiante	- In fase di accensione indica l'inizializzazione del sistema.
	Arancione fisso per 1 secondo	- Cancellazione o memorizzazione tessera avvenuta in modalità "Apprendimento".
LED 2	Acceso	- Dati trasmessi dal PC al lettore.
LED 3	Acceso	- Dati trasmessi dal lettore al PC.



Connessioni DB-9, USB e di alimentazione della scheda; le porte seriale e USB devono essere usate una in alternativa all'altra.

tevi di sapere che il PC decide le azioni da intraprendere in caso di riconoscimento o non riconoscimento del tag, i tempi di attivazione dei relé e le modalità corrispondenti, ecc. Inoltre in questa modalità è possibile apprendere e memorizzare in un database salvato sul PC un numero praticamente illimitato di transponder; anzi, bisogna far apprendere da qui i transponder che si desidera gestire, in quanto il computer ignora la memoria del micro e quindi i tag memorizzati nel PIC con la procedura stand-alone. In questa modalità quando un tag viene avvicinato all'ID-12 il microcontrollore dell'unità invia i dati al computer, il cui software li elabora e decide le azioni del

te operativa: *PC Link diretto*; in questo caso il circuito viene usato come puro e semplice lettore di tag per RFID. Ogni volta che un tag viene passato vicino al modulo ID-12, il microcontrollore invia i dati seriali corrispondenti via seriale e USB e il software caricato nel computer non fa altro che rilevare i codici e visualizzarli, ovvero, se sono stati abbinati a dei nominativi, visualizza i nominativi corrispondenti.

MODI D'USO E APPRENDIMENTO

Tramite il Dip-Switch DS1 è possibile impostare in qualsiasi momento, anche durante il normale funzionamento, la modalità operativa desiderata; la **Tabella 1** elenca le impostazioni del caso. Come vedete nella tabella, è pos-

Tabella 3 - Protocollo di comunicazione.

COMANDO	FUNZIONE	NOTE	RISPOSTA
* / L c	Accende/Spigne il LED	c: Indica il colore del LED e può assumere valore "R" (Rosso), "A" (Arancione), "V" (Verde), "S" (Spento).	
* / R n t1 t2	Attivazione dei Relè	n: Indica il relè che si desidera comandare. t1, t2: Indicano il tempo di attivazione del relè. Se entrambi sono posti a 0 il relè funziona in bistabile.	
* / B t	Attivazione Buzzer	t: Tempo di attivazione del buzzer. Il valore indicato viene moltiplicato per 500ms, quindi se si sceglie t=2 allora il buzzer suonerà per 1 secondo.	
* / C	Cancellazione EEPROM	Esegue la cancellazione totale della memoria.	CANCELLAZIONE MEMORIA
* / N	Numero codici Memorizzati	Numero di tessere presenti in EEPROM.	NUMERO TESSERE MEMORIZZATE TESSERE MEMORIZZATE x
* / P d1 d2	Scarica singolo Codice	Il lettore comunica al PC il numero della tessera che è memorizzata della posizione d1 e d2.	SCARICA POSIZIONE SONO PRESENTI IN MEMORIA x TESSERE SCARICO LA POSIZIONE d1 d2 CODICE LETTO ->1234<-
* / T	Scarica tutti i codici	Download completo dei codici presenti in memoria.	SCARICA TUTTO CODICE LETTO ->1234<-
* / V	Versione Firmware	Restituisce la versione firmware	VERSIONE FIRMWARE x.x

I caratteri che compongono la parola di comando devono essere scritti obbligatoriamente con caratteri maiuscoli. Ogni volta che un comando viene inserito in modo errato viene mostrata la seguente scritta: "COMANDO NON VALIDO".

Si consiglia di inviare ogni carattere che compone la parola di comando, ad una distanza di 1-2 ms dal carattere successivo, per evitare che il sistema perda qualche carattere.

sibile distinguere tra due modalità principali: la prima, con dip1 ad ON, riguarda il funzionamento stand alone, mentre se dip1 è OFF la scheda viene controllata dal Personal Computer. Vediamo nei dettagli le due fun-

zionalità possibili in stand-alone.

Chiave stand-alone

Il circuito legge i transponder RFiD, e se il loro codice è presente in memoria attiva il relé 1 per un certo periodo di tempo,

mentre in caso di tag non riconosciuto viene attivato il relé 2. In entrambi i casi si avrà una segnalazione visiva anche mediante led (led verde: tag riconosciuto; led rosso: tessera non riconosciuta). I tempi di attivazione dei due relé sono, per impostazione predefinita, pari ad 1 secondo, ma possono venire modificati mediante collegamento al PC, impostando temporaneamente la modalità *PC Link On Demand*.

Memorizzazione/Cancellazione

Tale modalità permette di decidere quali tag abilitare o meno in modalità "chiave stand-alone".

Per memorizzare un tag è sufficiente passarlo sul lettore (in prossimità dell'ID12) quindi se il codice identificativo non è in memoria, verrà memorizzato e il led bicolore si colorerà di arancione per un secondo. Subito dopo il circuito si porrà nuovamente in attesa di un nuovo tag.

Nel caso il codice sia già presente, il led si accenderà di rosso per un secondo ad indicare l'impossibilità di memorizzare la tessera, mentre se la memoria è piena, in fase di apprendimento il led rosso emetterà 10 lampeggi.

Per procedere alla cancellazione di un determinato tag è sufficiente tenere premuto il pulsante P1



Mostra Mercato Mondo Elettronica.net

4ª EDIZIONE

FIERA DELL'ELETTRONICA

INFORMATICA E RADIANTISMO

ORARIO 9,00/18,30 - PARCHEGGIO GRATUITO

BOLOGNA

PALANORD - PARCO NORD

Uscita Tangenziale n. 7 bis

31 OTTOBRE - 1 NOVEMBRE '09

Tel. e Fax 0545 27548 - Cell. 348 9030832 - www.mondoelettronica.net

VALE COME RIDOTTO

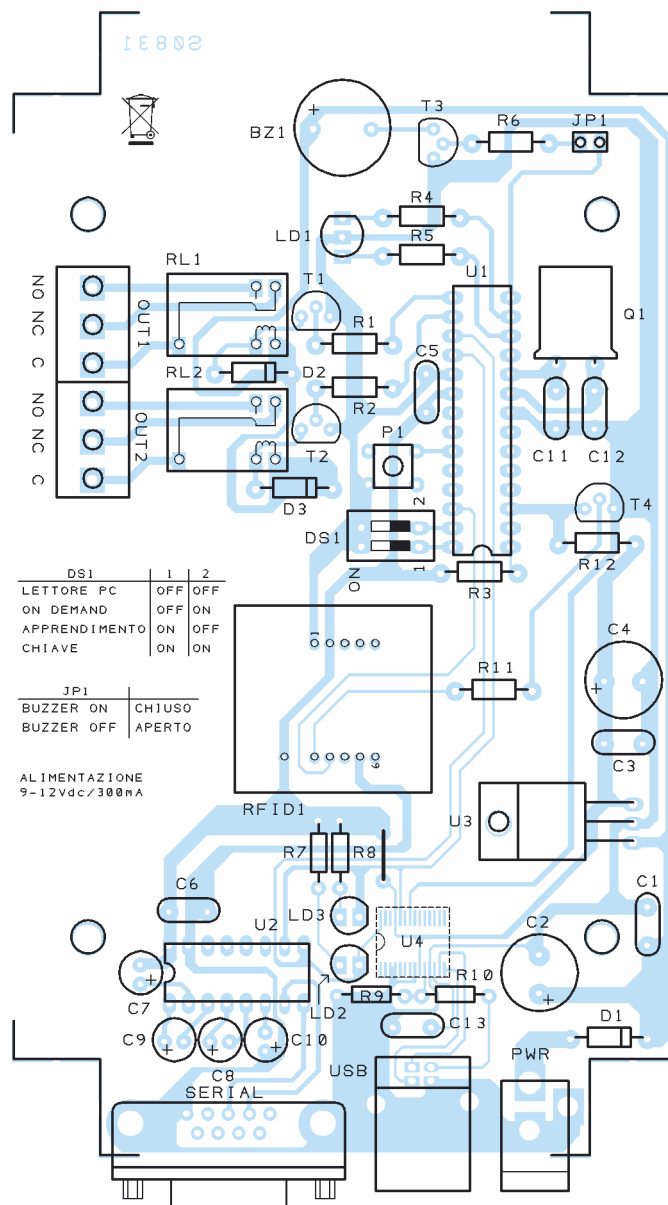
ANCHE IN FOTOCOPIA

Elenco Componenti:

R1÷R3, R6, R10÷R12: 4,7 kohm
 R5: 470 ohm
 R7, R8: 330 ohm
 R9: 10 kohm
 C1, C3, C5, C6, C13: 100 nF multistrato
 C2: 470 µF 25 VL elettrolitico
 C4: 470 µF 16 VL elettrolitico
 C7÷C10: 1 µF 63 VL elettrolitico
 C11, C12: 15 pF ceramico
 D1÷D3: 1N4007
 Q1: Quarzo 20 MHz
 U1: PIC16F876A (MF831)
 U2: MAX232
 U3: 7805
 U4: FT232RL
 DS1: Dip-switch 2 vie
 P1: Microswitch
 T1÷T4: BC547
 LD1: Led bicolore 5 mm rosso/verde
 LD2: Led 3 mm verde
 LD3: Led 3 mm rosso
 RL1, RL2: Rele 12V 1 scambio
 BZ1: Buzzer 12V con elettronica
 RFID1: Modulo ID12

Varie:

- Zoccolo 8+8
- Zoccolo 14+14
- Vite 10 mm 3 MA
- Dado 3 MA
- Morsettiera 3 poli (2 pz.)
- Plug alimentazione
- Connettore USB-B
- Connettore DB9 femmina
- Strip maschio 2 poli passo 2,54 mm
- Strip femmina 5 poli passo 2 mm
- Strip femmina 7 poli passo 2 mm
- Jumper
- Circuito stampato



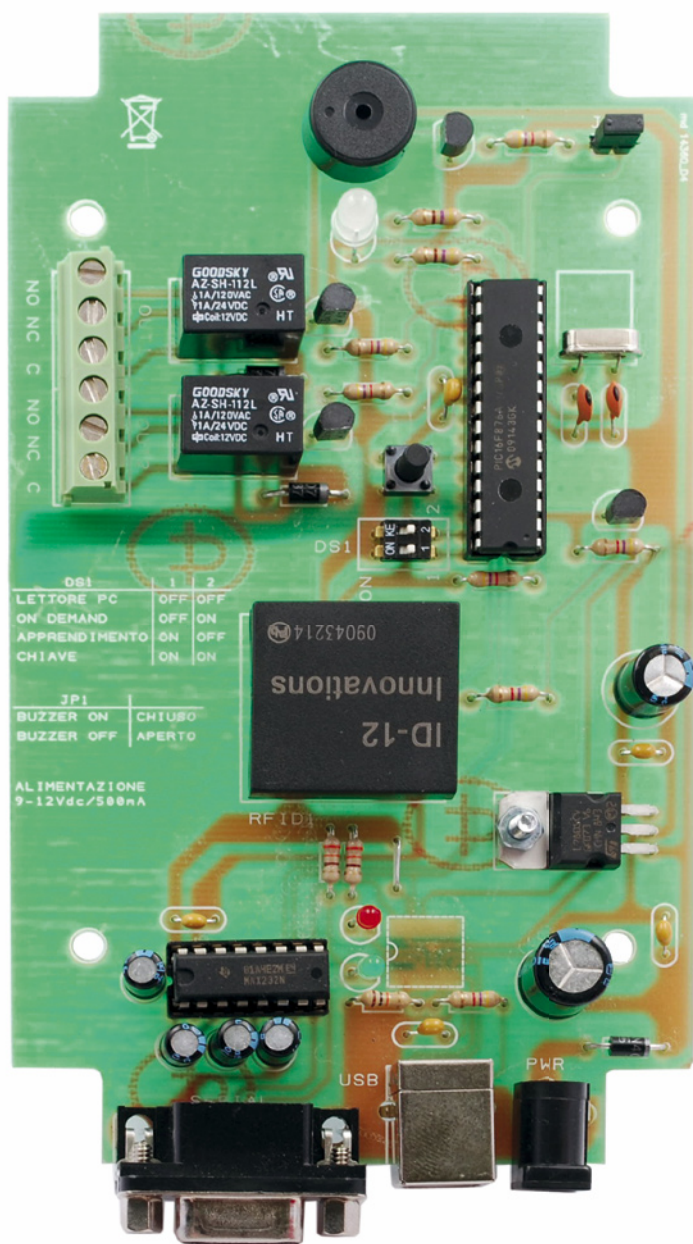
e mantenerlo premuto mentre si passa il tag sopra l'ID-12; se il led rimane acceso per un secondo di colore arancione, significa che il transponder è stato cancellato. Se il tag non era presente in memoria, il led rosso si accenderà per un secondo. Nel caso il DIP1 e DIP2 siano impostati rispettivamente su ON ed OFF durante la fase di accensione, verrà azzerata totalmente la memoria codici,

cosa segnalata dal led rosso acceso durante la fase di inizializzazione del sistema.

PC Link On Demand

Con questa modalità è possibile memorizzare nella EEPROM del PIC fino a 120 diversi codici (il tag viene memorizzato solo se non è già in memoria) e scaricarli successivamente sul PC mediante apposito software.

In questa modalità il circuito viene comandato esclusivamente via software, pertanto durante la normale esecuzione il dispositivo invierà al PC ogni 500 ms la stringa "INSERISCI COMANDO" ed attenderà quindi per un pari intervallo che il PC invii una stringa di comando. Trovate l'elenco completo dei comandi nella **Tabella 3** (Protocollo di comunicazione).



PC Link Diretto

In quest'ultima modalità il nostro dispositivo funziona come *Letto-re transponder* puro e semplice, quindi ogni volta che si passa una tessera RFiD nelle vicinanze del sensore, il corrispondente valore viene inviato direttamente al computer; dopo l'invio del codice, il dispositivo attenderà l'avvicinamento di un nuovo transponder.

REALIZZAZIONE PRATICA

Bene, giunti a questo punto possiamo vedere come si costruisce in pratica il lettore di transponder, per il quale abbiamo previsto un apposito 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 la basetta, fate i necessari fori e iniziate a disporre i componenti partendo da resistenze, diodi al silicio e zoccoli, quindi proseguendo con il dip-switch (il dip 1 di questo dovrà essere vicino al modulo ID-12) e i condensatori, dando la precedenza a quelli non polarizzati e rispettando la 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 transistor e i led, rammentando che per i diodi rosso e verde il catodo è il terminale che si trova dal lato smussato del contenitore (se usate i led da 3 mm è probabile che lo smusso non ci sia, quindi ricordate che il catodo è l'elettrodo più corto) mentre per il bicolore lo smusso corrisponde al terminale di anodo della giunzione rossa. Procedete collocando il quarzo, i due relé miniatura, il cicalino, le morsettiere a passo 5 mm per le connessioni degli scambi dei relé e il regolatore 7805, che potete

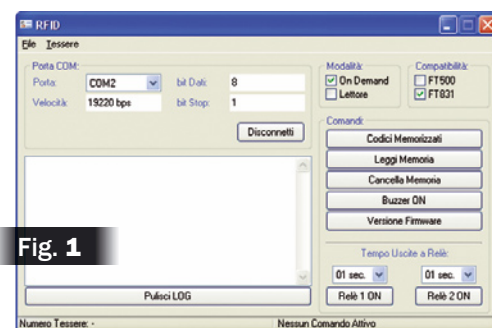


Fig. 1

montare adagiato e fissato con un piccolo bullone 3MA più dado dopo averne piegato i terminali ad angolo retto. Fate attenzione all'FTF232RL, che va disposto dal lato delle saldature; siccome è un SMD, dovete fissarlo con un saldatore a punta finissima utilizzando filo di lega saldante da 0,5 mm di diametro. A questo punto inserite e saldate i connettori DB-9 femmina o



RFiD contro la distrazione

A chi non è successo, per distrazione o per fretta, di dimenticare qualcosa sul tetto della macchina per poi accorgersene quando ormai era troppo tardi? Probabilmente a tutti. Per rimediare, la Ford ha sviluppato una soluzione molto semplice: il proprietario dell'auto applica dei tag (sensori UHF RFID) agli oggetti che più spesso sposta su e giù dalla macchina, e il reader RFID integrato nel bagagliaio si occupa di rilevarli e verificare che tutto sia in ordine.

In questo modo, se il conducente dimentica il telefonino sul tettuccio o non si ricorda di mettere a posto il tappo del serbatoio della benzina, il lettore a radio frequenza capta l'anomalia e la segnala mostrando un avviso sul display della vettura.

Il sistema RFID progettato da Ford in partnership con DeWalt e ThingMagic (società specializzate in soluzioni RFID) è pensato anche per rimediare a quelle distrazioni che potrebbero influire sulla sicurezza dei bambini, quando per esempio ci si dimentica di fissare il seggiolino al sedile o quando le cinture non sono assicurate in maniera adeguata. Naturalmente, non si tratta del primo caso in cui la tecnologia di identificazione a radio frequenza incontra il mondo dei motori e in particolare quello delle quattro ruote; infatti esistono già applicazioni RFID per il car sharing, per l'apertura delle portiere e perfino per il controllo dei pneumatici.

file di cinque contatti ciascuna da uno zoccolo per integrati dip. 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 montaggio 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 9÷12 volt e una corrente continua dell'ordine di circa 300 milliampere.

IL SOFTWARE DI GESTIONE

Il programma permette di gestire il lettore in entrambe le modalità legate al PC (*On Demand* e *Lettore*). E' possibile utilizzare il dispositivo sia con collegamento seriale che USB. Ad esempio, all'interfaccia seriale è possibile collegare un convertitore Ethernet o RS485 per portare

l'USB B, a seconda che vogliate connettere il circuito al computer mediante, rispettivamente, seriale o USB, ovvero disponeteli entrambi se pensate che possano esservi utili per spostare la gestione dell'unità da un computer con la COM ad un altro con la sola USB; per l'alimentazione disponete sullo stampato una presa plug da c.s. A questo punto non vi resta che sistemare il modulo ID-12, che troverà alloggiamento su uno zoccolo artigianale che potrete costruire ritagliando due

Tabella 4 - Descrizione comandi.

Pulsante	Descrizione
"Codici Memorizzati"	Permette di richiedere al lettore il numero di tessere presenti in memoria, a tal proposito si ricorda che il valore massimo ammissibile, per limiti di spazio nel lettore, è di 120 tessere. Il valore restituito viene mostrato nella ToolBar in basso a sinistra.
"Leggi Memoria"	Viene effettuata la lettura della memoria del lettore. Tutte le tessere presenti vengono salvate in un database presente nella directory software il cui nome è "Tessere.mdb".
"Cancella Memoria"	Cancella la memoria tessere. Solitamente utilizzata dopo il download.
"Buzzer ON"	Attiva il buzzer per 500 ms.
"Versione Firmware"	Restituisce la versione Firmware del lettore.
"Relè x ON"	Attiva il relè 1 o 2 per il tempo impostato mediante apposita ComboBox presente sopra il relativo pulsante. Mediante questo comando è anche possibile impostare le tempistiche dei relè in funzionamento "Chiave stand alone".

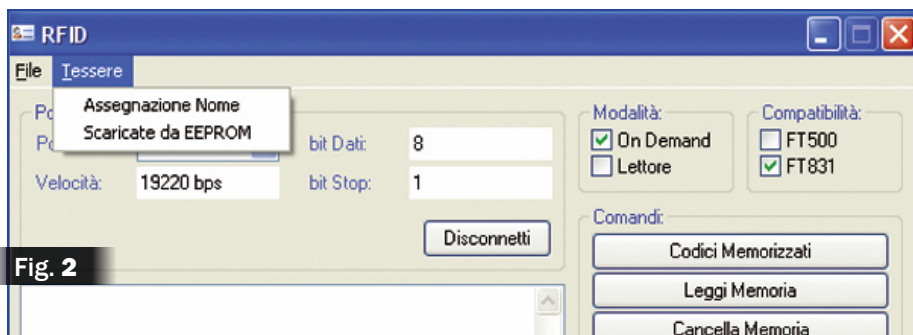


Fig. 2

il segnale in remoto, mentre in locale si può comodamente utilizzare la porta USB. Per evitare problemi di conflitto, non è possibile ovviamente utilizzare in contemporanea da entrambe le postazioni il software di gestione. In Fig. 1 è possibile vedere come si presenta la schermata principale del software. Al primo avvio la modalità impostata è "On Demand" ma in qualsiasi momento è possibile impostare la modalità "Lettore". La modalità scelta viene salvata in un file di configurazione per evitare che alla successiva apertura la si debba impostare nuovamente. In merito alla prima modalità (*On Demand*) va detto che tutti pulsanti disponibili sulla schermata principale sono attivi in quanto è possibile gestire interamente il lettore, mentre nella seconda modalità, trattandosi di un lettore, sarà solo possibile visualizzare il codice della tessera passata sul

lettore. Impostando la "Compatibilità" si seleziona invece il tipo di dispositivo che si sta utilizzando. La Fig. 2 raffigura il menu principale del software. Mediante i menu è possibile assegnare un nome ad una tessera (Assegnazione Nome), oppure visionare le tessere scaricate dalla memoria EEPROM del lettore in modalità "On Demand". L'assegnazione di un nome ad una tessera nella modalità "On Demand" è fattibile in due modi:

- dopo avere scaricato le tessere presenti in EEPROM;
- direttamente in fase di lettura.

In merito alla prima modalità, precisiamo che, per poter associare un nome, occorre che in memoria siano già presenti dei codici di tag RfId. Come prima cosa dovete procedere ad effettuare la "Lettura Memoria", dopo di che accedete al menu "Tessere > Scaricate da EEPROM". Si presenterà la finestra di dialo-

go illustrata nella Fig. 3.

Nell'elenco corrispondente è possibile vedere i codici di tutti i tag appresi dal lettore.

Bene, visto ciò selezionate, (facendo un Check) tutti i nominativi di cui si desidera associare un nome; inutile dire che tale procedura vale per le tessere prive di nome. Selezionate quelle desiderate, fate click sul pulsante "Nominativo". Così facendo, comparirà una schermata che permetterà di associare il nome desiderato (Fig. 4). La schermata comparirà per ogni transponder (tessera) selezionato. Il campo "Tessera" è di sola lettura e contiene il codice tessera a cui associare il nome. Invece nel campo "Nominativo" si dovrà inserire il nome dell'utente che utilizzerà il tag. Il campo "Tempo Relé" indica per quanto tempo deve essere attivato il relé 1 nel caso il tag venga passato sul lettore, (nell'immagine esempio viene scelto 01 s). Confermate con "OK". Una volta terminato il salvataggio, la schermata precedente verrà aggiornata e compariranno i codici dei tag con i relativi nomi, associati come si può vedere nella finestra di dialogo nella Fig. 5. Relativamente agli altri pulsanti, "Elimina" permette di cancellare tutti i tag memorizzati nel

Fig. 3

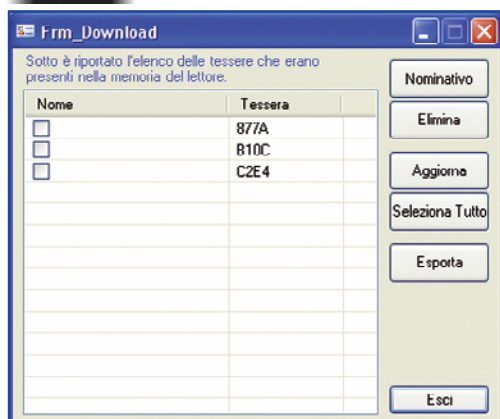


Fig. 4

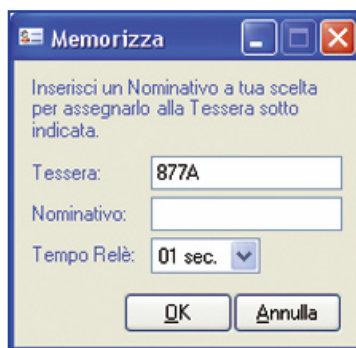


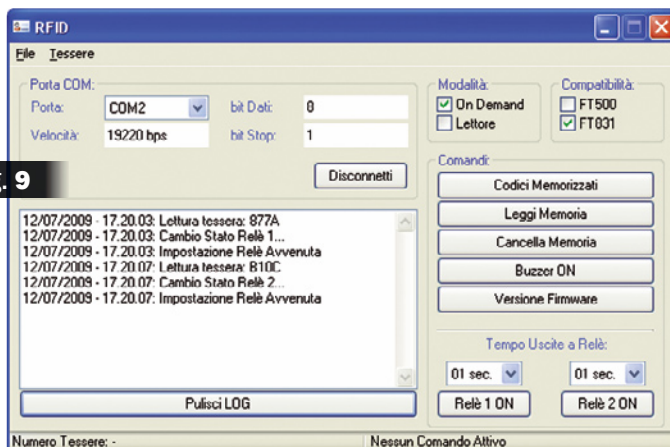
Fig. 5



Fig. 8



Fig. 9



database. Il pulsante "Aggiorna" effettua solo un refresh della schermata, il tasto "Seleziona Tutto" non fa altro che selezionare/deselezionare tutto il contenuto della lista, utile soprattutto quando si deve eliminare tutto, magari dopo un'esportazione del database in un file di testo, opera-

zione fattibile tramite il pulsante "Esporta". In quest'ultimo caso viene richiesto il nome del file di testo (.txt) con cui esportare tutto il contenuto appena scaricato. Analizziamo ora la seconda modalità di assegnazione nominativo, più semplice e veloce. Tale metodo di assegnazione è

valido sia per la modalità "On Demand" che per quella "Lettore". Come prima cosa bisogna accedere al menu "Tessere > Assegnazione Nome". Per associare un nome e memorizzare un tag dovete premere il pulsante "Nuovo"; un messaggio informerà di passare la tessera sul lettore (Fig. 7).

Appena avrete passato il tag, comparirà la schermata (Fig. 4) che permetterà di associare un nome ad esso. Notate che è molto importante impostare il tempo di attivazione del relé, perché è appunto grazie a questo che nel caso si passerà un tag sul lettore, se riconosciuto come valido, sarà attivato il relé 1 per il tempo scelto. Nel caso il tag non rientri fra quelli memorizzati nel database, verrà attivato il relé 2 per 500 ms. Ora confermate con "OK" e il tag verrà memorizzato (Fig. 8). Tornando, invece, alla schermata principale del software, notate la presen-

za di una finestra di LOG che permette di mostrare tutto ciò che avviene (se è stato passato un tag sul lettore, se il relé è stato attivato o meno, se il buzzer ha suonato, ecc.); insomma, ogni singolo comando eseguito verrà mostrato nella finestra di LOG. Ovviamente trattandosi di un LOG, ogni informazione è accompagnata da Data-Ora (come illustrato in Fig. 9).

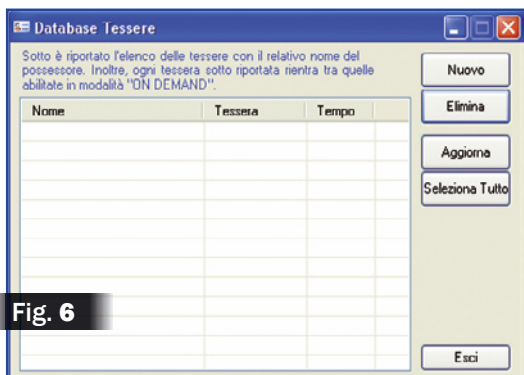
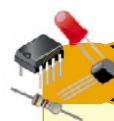
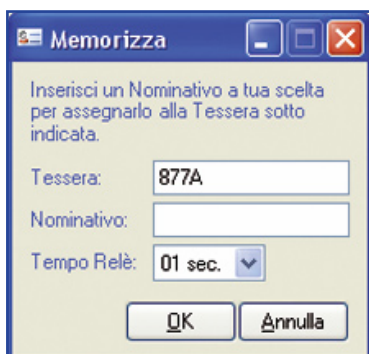


Fig. 6

Passare sul lettore la Tessera Trasponder che si desidera inserire nel database.

Annulla

Fig. 7



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono facilmente reperibili in commercio. Dal sito della rivista è possibile scaricare gratuitamente gli unici elementi non disponibili sul mercato, ovvero il master del circuito stampato, il firmware col quale programmare il microcontrollore ed il software di gestione per PC da noi sviluppato. Per quanto riguarda il microcontrollore, questo elemento è anche disponibile già programmato al prezzo di 12,00 Euro (codice MF831). Per quanto riguarda gli altri componenti, il modulo RFID a 125 kHz ID12 costa 33,00 Euro, mentre i transponder - disponibili sia in versione carta di credito (TAG2) che portachiavi (TAG1) - costano 3,00 Euro cadauno. 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>

& MICROELETTRONICA MEMS

di ARSENIO SPADONI

Dirigere un'orchestra sinfonica virtuale con una bacchetta wireless, trasformare un noioso esercizio di fisioterapia in una divertente esperienza interattiva o semplicemente navigare nel menu di un telefono cellulare inclinando l'apparecchio, invece di premerne i tasti: tutte attività rese possibili dalla tecnologia di rilevazione del movimento basata sui dispositivi MEMS (Micro-Electro-Mechanical System, sistema micro-elettromeccanico), tecnologia che sta trasformando il mondo dell'elettronica di consumo e dei dispositivi portatili ma che ha anche importanti ricadute in campo medico e industriale.

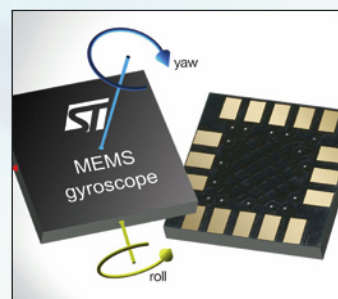
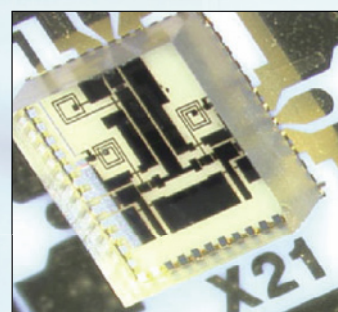




Un MEMS è un sistema con una miniaturizzazione spinta che integra dispositivi di diversa natura, meccanici, elettrici ed elettronici, su uno stesso circuito, solitamente di silicio, svolgendo funzioni di rilevazione, elaborazione o attuazione. Questi dispositivi sono il risultato dell'integrazione dei chip con altre tecnologie innovative e rappresentano una naturale evoluzione della microelettronica. Le parti elettroniche sono infatti realizzate usando

i processi standard dei circuiti integrati; analoghi processi abbinati a particolari tecniche di microlavorazione vengono usati per fabbricare i componenti meccanici o di altra natura che fanno parte dei MEMS. Nei sistemi più complessi l'integrazione dei vari elementi meccanici ed elettronici in uno stesso substrato di silicio consente di realizzare un oggetto che ha molteplici funzioni e possibilità applicative. I circuiti elettronici del chip pos-

sono essere visti come il cervello del sistema, cui gli altri dispositivi contenuti nello stesso chip aggiungono sensi e braccia. Si hanno quindi sistemi in grado di captare informazioni dall'ambiente, di prendere decisioni e, infine, di attuare le azioni opportune. I sensori possono misurare fenomeni di varia natura: meccanica, termica, ottica, chimica, magnetica biologica, ecc; i circuiti elettronici del chip elaborano i dati raccolti e indicano agli attuatori le azioni



Videoproiettori & MEMS

Sfruttando le ricerche nel settore dei MEMS, molti anni fa Texas Instruments ha realizzato un piccolissimo chip completamente ricoperto da migliaia di microspecchi: il DMD (Digital Micromirror Device) che è alla base della tecnologia DLP (Digital Light Processor), utilizzata in proiettori frontali per impieghi professionali ma anche casalinghi e, recentemente, portatili. Il DMD è costituito da una matrice di specchi (anche di milioni di elementi), disposti su una superficie di pochi millimetri quadrati, ciascuno dei quali può assumere due posizioni (ON e OFF) e ruotare di $\pm 10^\circ$ o $\pm 12^\circ$ (esistono due varianti di chip) in funzione dell'attrazione elettrostatica tra la

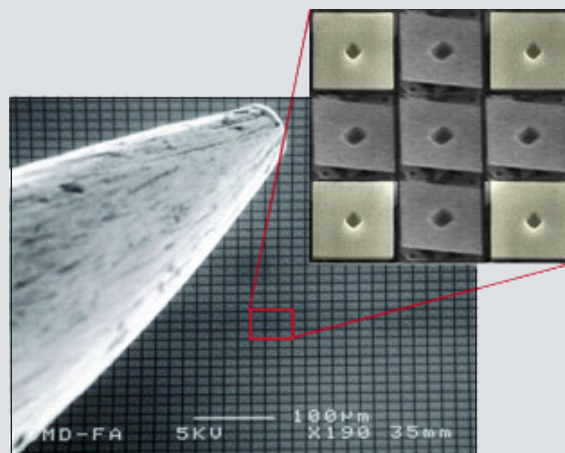
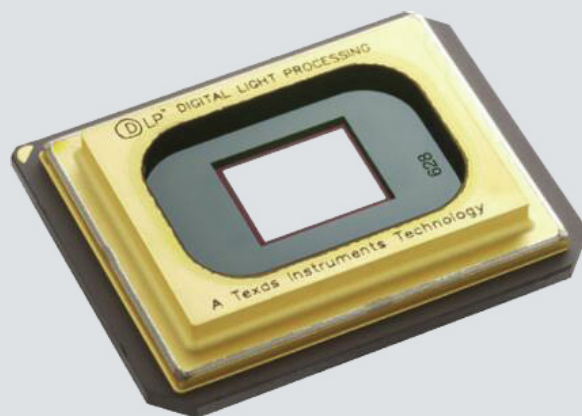


struttura dello specchio e gli elettrodi connessi alla sottostante cella di memoria. I perni, che permettono agli specchi di ruotare migliaia di volte al secondo, non si piegano e non si rompono perché le leggi della meccanica hanno effetti diversi quando si applicano a oggetti miniaturizzati. Un materiale, sottoposto a stress, cioè piegato ripetutamente, si rompe a causa dello spostamento, all'interno della struttura cristallina, dei singoli cristalli che lo compongono. Nel caso del perno del microspecchio, per effetto delle dimensioni microscopiche, tutti i cristalli si trovano sulla superficie della struttura: non esiste una struttura di cristalli interni e quindi il fenomeno di stress dovuto allo spostamento non ha conseguenze. Gli specchi non rimangono "incollati" perché la struttura del dispositivo preve-

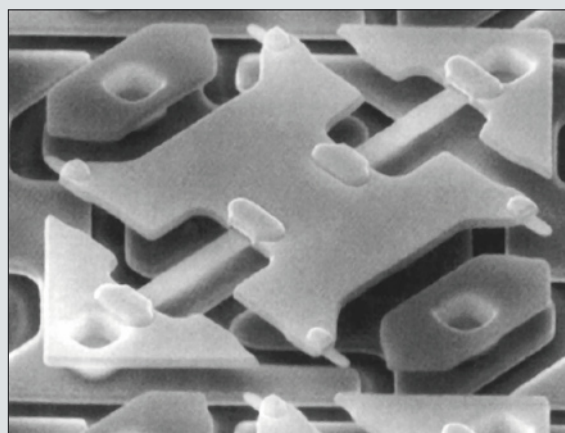
de delle minuscole molle che spingono lo specchio lontano dalla superficie di atterraggio, quando cessa la forza elettrostatica; sono stati ridotti al minimo gli effetti delle forze di van der Waals che causano l'attrazione dei materiali a livello molecolare; anche gli effetti dovuti alla possibile condensazione dell'umidità sono ridotti poiché il dispositivo è sigillato in ambiente secco in un package ermetico che assicura l'assenza di umidità per l'intera vita del DMD.

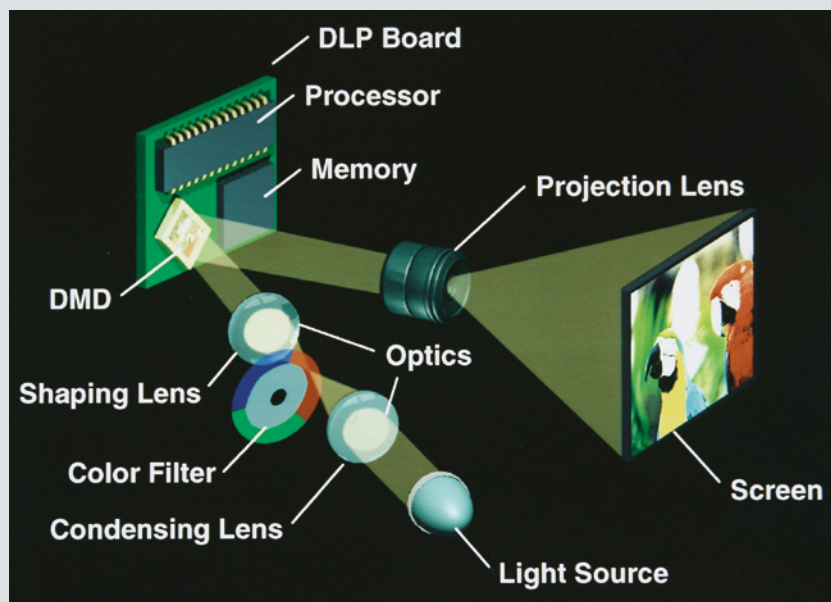
I microspecchi non sono per nulla fragili. Normalmente, infatti, la rottura di un vetro è causata dalle vibrazioni; nel caso di strutture DMD, invece, le vibrazioni sono caratterizzate da frequenze di almeno due ordini di grandezza superiori a quelle generate nell'uso normale (la frequenza di risonanza più bassa è 100 kHz, le successive armoniche sono dell'ordine dei MHz). In pratica le sorgenti normali di vibrazione e di shock, ad esempio la caduta sul pavimento, non hanno particolari conseguenze sui dispositivi a microspecchi. Anche la probabilità che un microspecchio sia difettoso è molto bassa: i processi di fabbricazione dei DMD sono gli stessi utilizzati per la produzione dei circuiti integrati (clean-room per la fabbricazione dei wafer) e l'attuale resa è così elevata da consentire di soddisfare la richiesta di dispositivi con un livello di affidabilità del 100%.

Nel disegno della pagina di destra viene schematizzato lo schema di funzionamento della configurazione che utilizza un solo DMD: la luce monocromatica passa attraverso tre filtri (rosso, verde, blu) che costituiscono le tre porzioni di una ruota: quindi l'informazione di cromaticità, così come quella di luminanza, viene ottenuta grazie alla tecnica di modulazione d'impulso della luce e alla elevata velocità di commutazione dei microspecchi. Per migliorare l'efficienza è stato sviluppato un sistema che permette di minimizzare la luce non trasmessa attraverso



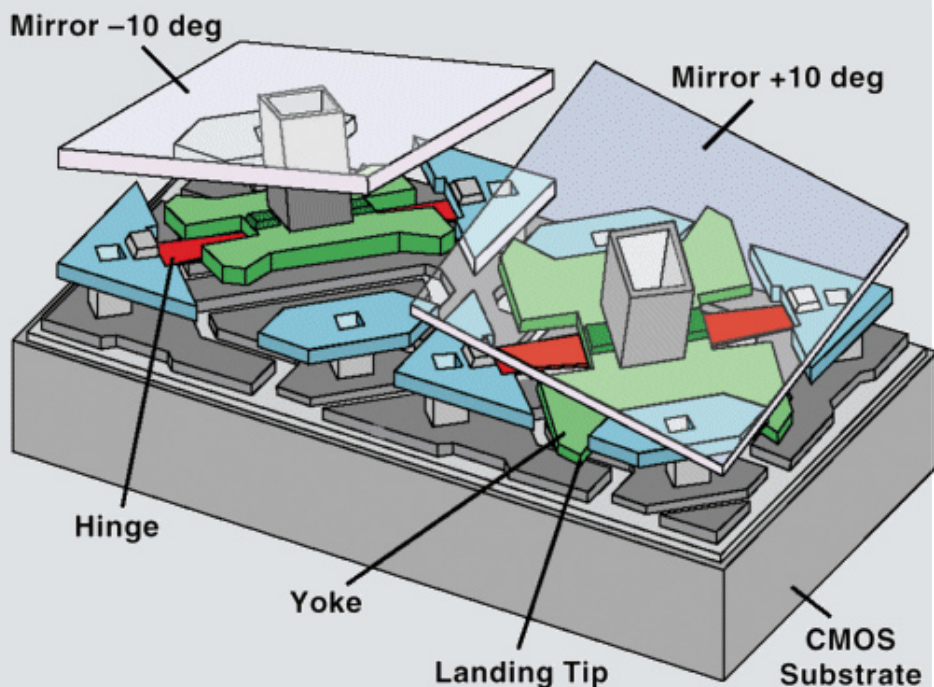
Un una zampa di formica sopra una matrice di specchi di un DMD.





la ruota, ma assorbita all'interno del proiettore; con queste nuove tecniche è possibile raggiungere con un solo modulatore l'efficienza che caratterizzava precedentemente i sistemi basati su tre DMD. La scala dei grigi viene ottenuta utilizzando una tecnica di modulazione di ampiezza d'impulso (PWM, pulse width modulation): il singolo microspecchio viene rapidamente commutato (tempo di ommutazione meccanica ~15 ms e di commutazione ottica ~2 ms) e il rapporto tra il tempo in cui è ON

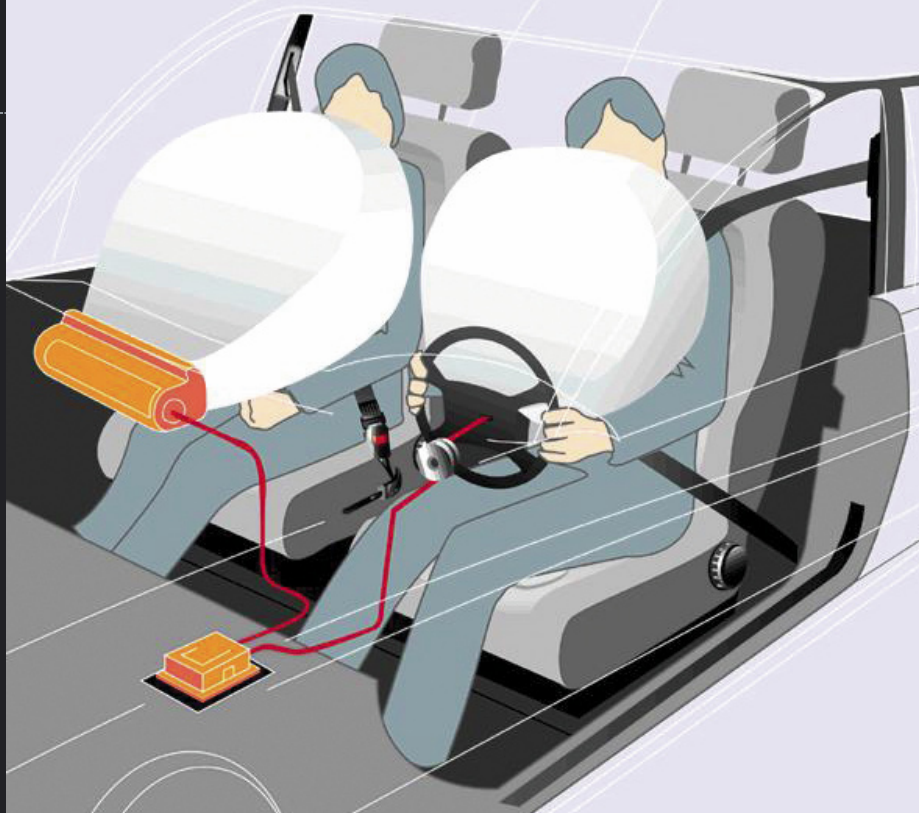
e il tempo in cui è OFF è direttamente proporzionale al livello di luminosità del punto corrispondente sullo schermo. È possibile distinguere tra sistemi DLP: a un chip, oppure a tre chip. Il secondo e più costoso sistema prevede l'utilizzo di 3 chip DMD diversi, dedicati ciascuno ad una delle componenti cromatiche RGB. I display DLP a un chip invece, i più diffusi, utilizzano un solo elemento DMD che genera la scala di grigio alternativamente per ciascun canale RGB.



da effettuare: muovere, posizionare, regolare, pompare, filtrare, eccetera.

La tecnologia MEMS deriva dalla tecnologia dei circuiti integrati a semiconduttore, ma se ne differenzia per alcuni aspetti e per il materiale di supporto che non è più solo il silicio ma anche vetro, polimeri, plastica. L'interesse per i MEMS deriva dal fatto che possono essere impiegati in svariati settori, necessitano di pochissimo materiale, sono economici e, pur avendo dimensioni dell'ordine dei micron o dei millimetri, sono in grado di eseguire le stesse funzioni di rilevazione, elaborazione e di attuazione di oggetti molto più ingombranti e costosi.

Dopo gli importanti sviluppi degli anni passati che hanno portato alla grande diffusione di questi dispositivi soprattutto nel campo dei sistemi di controllo degli airbag, delle testine per stampanti ink-jet e dei videoproiettori, negli ultimi 4-5 anni abbiamo assistito alla nascita di una nuova generazione di sensori destinati al mercato consumer, essenzialmente accelerometri, sensori di pressione e giroscopi utilizzati nei telefonini, nelle console per videogiochi e negli elettrodomestici. La società che è alla guida di questa rivoluzione è la ST Microelectronics, primo fornitore mondiale di questi nuovi entusiasmanti dispositivi. "Prima del 2006" commenta Benedetto Vigna a capo della divisione MEMS di ST "i sensori MEMS erano utilizzati in poche applicazioni, come i rilevatori di collisione per airbag, ma erano troppo grandi, consumavano molto ed erano troppo costosi per le applicazioni di elettronica di consumo". La ST ha affrontato i primi due problemi con tecniche ingegneristiche innovative, sviluppando e brevettando nuove strutture MEMS



più piccole, più robuste e che richiedono minore energia elettrica per funzionare, ma sempre con prestazioni molto elevate. Per risolvere il terzo problema, la società nel 2006 fece un passo coraggioso, investendo più di 40 milioni di dollari per realizzare la linea più avanzata al mondo per la produzione di MEMS. L'impianto per la diffusione di fette di silicio da 200 mm di Agrate Brianza (Milano) venne inaugurato nel novembre 2006 ed è completamente dedicato alla produzione di dispositivi MEMS come accelerometri, giroscopi e sensori di pressione. In seguito all'enorme successo della propria linea di accelerometri MEMS per la misura del movimento lineare, la ST ha ampliato il portafoglio con l'aggiunta di piccolissimi giroscopi MEMS che rilevano e misurano l'inclinazione e la velocità angolare; sono dispositivi ideali per controller di videogiochi ancora più sofisticati, trasduttori per applicazioni di realtà virtuale, controllo del movimento, dispositivi di puntamento e navigatori per implementare funzioni come il "dead-reckoning"

(il calcolo del percorso effettuato anche nel caso di temporanea assenza della ricezione del segnale satellitare) e il "map-matching" (la verifica di congruità tra il percorso effettuato e la mappa memorizzata nel sistema).

I MEMS possono essere classificati secondo quattro macro categorie:

- Sensori e attuatori;
- MEMS RF;
- MEMS ottici ed optoelettronici;
- MEMS microfluidici e bioMEMS.

I sensori rappresentano una classe di MEMS progettata per rilevare i cambiamenti e interagire con l'ambiente circostante; possono essere chimici, di movimento, inerziali, termici e ottici.

Gli attuatori costituiscono un gruppo di dispositivi progettati per fornire energia o stimoli ad altri componenti o ad altri MEMS e sono tipicamente guidati per mezzo di stimoli termici o elettrostatici.

I MEMS RF rappresentano un gruppo di dispositivi utilizzati per trasmettere segnali radio ad

elevata frequenza. Tipicamente si impiegano interruttori a contatto metallico, condensatori regolabili, antenne ecc.

I MEMS ottici sono dispositivi progettati per dirigere, riflettere, filtrare e/o amplificare la luce; possono essere interruttori ottici oppure riflettori. Si sta lavorando nella direzione di una loro sempre maggiore integrazione all'interno di dispositivi optoelettronici.

I MEMS microfluidici sono dispositivi progettati per interagire con ambienti in cui siano presenti dei fluidi: pompe e valvole progettate per muovere, espellere e mescolare volumi di liquido particolarmente ridotti. I MEMS microfluidici, progettati specificatamente per interagire con campioni biologici quali proteine, cellule, reagenti ecc. sono detti bioMEMS e possono essere impiegati per il drug-delivery, per analisi mediche e per tantissime altre applicazioni.

SENSORI E ATTUATORI

I sensori basati sulla tecnologia MEMS presentano una gamma di applicazioni molto ampia, dai sensori inerziali e di movimento a quelli biologici e risultano particolarmente adatti in tutti quegli ambiti dove le tecnologie tradizionali tendono a fallire a causa delle caratteristiche dell'ambiente circostante, delle limitazioni a livello di sistema o per problemi di progettazione o dimensioni. I principali sensori basati sui MEMS sono i seguenti:

- sensori di pressione;
- sensori di accelerazione;
- sensori di velocità;
- sensori di forza.

I sensori di pressione sono stati i primi dispositivi basati sul silicio introdotti sul mercato nei lontani anni '60 per applicazioni militari e aerospaziali. Nel 2003 il volume di sensori di pressione prodotti ha superato i 200 milioni di unità, con i mercati dell'automobile (oltre 10 sensori per veicolo sulle vetture di fascia più alta) e medicale (sensori per apparecchiature usa e getta) a fare la parte del leone.

I MEMS per misurare l'accelerazione, i giroscopi e i sensori di velocità, dopo la massiccia diffusione nell'ambito degli airbag per autovetture hanno visto aprirsi un mercato altrettanto importante: quello consumer, dai sensori di stabilità per macchine fotografiche e videocamere, ai sensori per videogiochi, dai sistemi GPS ai palmari. Per quanto riguarda i sensori di forza (peso), il principale mercato è quello delle bilance casalinghe.

Tra le applicazioni emergenti dei sensori basati sui MEMS possiamo citare quelle a infrarosso, quelle di flusso, quelle chimiche e quelle biologiche. Le principali tecnologie di rilevazione che possono essere applicate mediante dispositivi MEMS sono le seguenti:

- Piezoresistiva
- Capacitiva
- Elettromagnetica

L'effetto piezoresistivo produce nel silicio un cambiamento di

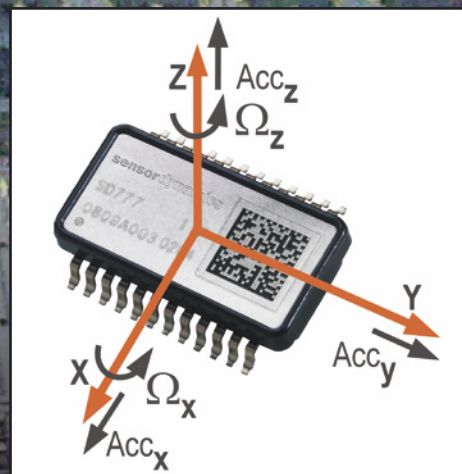
resistenza maggiore di circa 2 ordini di grandezza rispetto a quello che si otterrebbe con un cambiamento nelle dimensioni in condizioni di stress (tipico dei metalli). Ad esempio, se si allungasse per trazione un metallo dello 0,1%, il tipico allungamento di un metallo per resistori sarebbe dello 0,2%; la resistenza del silicio, invece, cambierebbe di circa il 10%. I sensori piezoresistivi si utilizzano per misurare la pressione, l'accelerazione e la forza (il peso): tipicamente se ne collegano fra loro più di uno per ridurre gli errori.

I sensori capacitivi presentano almeno un elettrodo capace di muoversi al variare di un valore in ingresso, tipicamente la pressione, l'accelerazione o la velocità. La configurazione più semplice è quella composta da due capacitori a elettrodi piatti, ma la più diffusa perché più efficace è quella "a pettine", che garantisce una maggiore sensibilità.

Tra i sensori elettromagnetici ricordiamo quelli a risonanza e quelli termoelettrici.

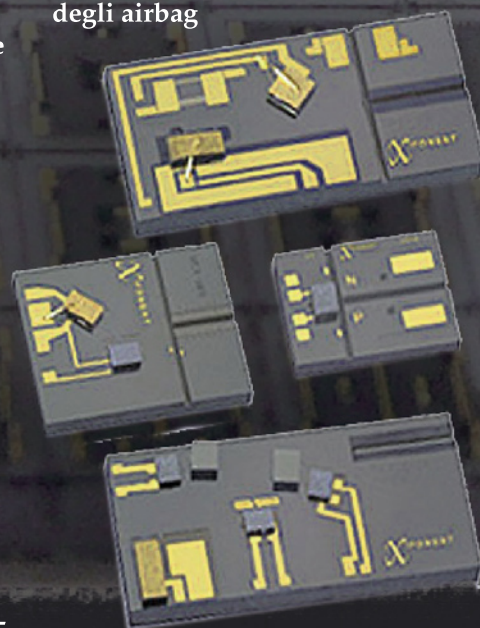
I sensori a risonanza sono basati su raggi o diaframmi risonanti.

Applicando sollecitazioni differenti è possibile variare la frequenza di risonanza (un po' come accade per le corde di una chitarra) consentendo la misurazione di variabili in input come la pressione, l'accelerazione, la velocità e la tempe-



ratura. I sensori termoelettrici basati sui MEMS utilizzano un certo numero di piccole termocoppie collegate in serie e vengono impiegati soprattutto per misurare l'intensità della radiazione infrarossa. Un'estremità della termocoppia è posizionata al di sopra di un'area isolata (dal punto di vista termico) del chip, l'altra sul conduttore in silicio. In risposta alla radiazione l'estremità isolata tende a scaldarsi, consentendo la misurazione dell'intensità della radiazione.

Una delle applicazioni di maggiore successo dei sensori MEMS riguarda il loro utilizzo per i sistemi di attivazione degli airbag



MEMS & elettronica di consumo

Anche quanti non avevano un particolare feeling con i videogiochi, dopo aver visto in azione e provato la Wii di Nintendo si sono dovuti ricredere. E alla fine è stato un boom di vendite: tutto grazie alla interattività dei nuovi telecomandi e accessori. Giocare a tennis col telecomando Wii (magari in coppia,

o con un utente remoto), dà sensazioni molto simile al tennis reale; affrontare uno slalom con la Wii Balance Board è come scendere da una vera pista innevata. Tutto merito dei progettisti di Nintendo ma anche di chi, principalmente ST Microelectronics, ha progettato e prodotto i sensori MEMS presenti all'interno di questi attrezzi. Il teleco-



il sensore, la cosiddetta forza G. Nel momento in cui il MEMS subisce un'accelerazione, la massa si sposta e la molla controbilancia la forza esercitata misurandone lo spostamento. Fino a pochi anni fa, gli accelerometri erano destinati a usi scientifici e militari mentre oggi sono presenti praticamente ovunque. Nei telefonini di ultima generazione funzionano come inclinometri e permettono di cambiare la visualizzazione dello schermo ruotandolo in orizzontale o in verticale. Nelle fotocamere evitano che le immagini vengano mosse mentre possono servire per rilevare le vibrazioni di automobili,

mando Wii è infatti basato su un accelerometro MEMS creato proprio dalla ST Microelectronics che consente di individuare la posizione del controller nello spazio tridimensionale. Alla base del funzionamento di un accelerometro c'è il principio di equivalenza di Einstein secondo il quale gravità e accelerazione sono indistinguibili. La parte meccanica del sensore è una massa solidale con una "molla" microscopica. Il dispositivo misura l'accelerazione e la gravità a cui è sottoposto



delle automobili.

Gli attuatori sono una famiglia di componenti MEMS progettati per generare movimento oppure per produrre forza motrice per muovere altri componenti MEMS. Esistono fondamentalmente due tipi di attuatori, quelli elettrostatici e quelli termici, di solito alimentati tramite un segnale di tipo elettronico in forma di onda. Gli attuatori elettrostatici basano il loro funzionamento sull'applicazione di un campo elettrico tra strutture fisse e strutture mobili. Una struttura è agganciata al substrato e viene spostata produ-

cendo un campo elettrico ai margini della struttura stessa mentre i componenti fissi rappresentano le sole strutture statiche del dispositivo. Il risultato è una struttura "a pettine" che si intreccia con la struttura mobile, le cui estremità possono essere spostate proprio grazie al campo elettrico che percorre i denti "fissi". Grazie al medesimo principio le strutture possono poi essere fatte ritornare nella posizione originale. Gli attuatori termici sono dispositivi che operano utilizzando la corrente elettrica in una struttura allungata e costruita leggermente fuori asse. Tale accorgimento (del fuori asse) determina la direzione in cui l'attuatore termico si muoverà. Infatti, quando la corrente passa tra i bracci orizzontali dell'attua-

tore, i bracci si scaldano e si espandono, causando lo spostamento nella direzione voluta. Dispositivi di questo tipo, attivati per via elettrostatica, sono già disponibili sul mercato e oggi molti MEMS (microfluidici, ottici, RF ecc.) operano utilizzando una qualche forma di attivazione elettrostatica.

MEMS A RADIOFREQUENZA

I MEMS a radio frequenza si stanno diffondendo quali eccellenti alternative alle attuali tecnologie RF allo stato solido e rappresentano una tecnologia estremamente promettente, che si pensa destinata a rivoluzionare le tecnologie a radiofrequenza, il mondo della produzione industriale e il mercato del largo consumo. Numerosi dispositivi

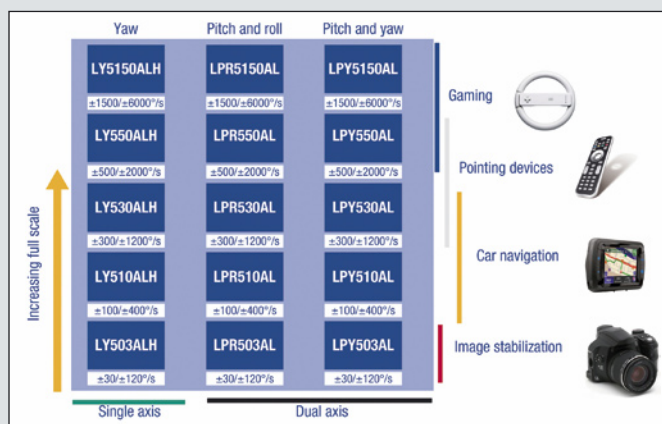
macchinari e edifici, per misurare l'attività sismica e l'inclinazione a cui viene sottoposto, ad esempio, il veicolo in curva, in modo da aumentarne la stabilità.

Sempre nelle automobili, vengono utilizzati per l'attivazione

ne degli airbag. Anche i navigatori GPS montano un accelerometro per ovviare alla mancanza di segnale del satellite. Questo microchip può essere inoltre montato su laptop e valigette come dispositivo antifurto e negli hard-disk per bloccarne il funzionamento in caso di caduta. Le vendite di MEMS di ST sono lievitare nel 2008 del 118% a più di 209 milioni di dollari, collocando il gruppo italo-francese al primo posto al mondo in questo settore, dal quarto che ricopriva l'anno precedente. Dietro la vertiginosa crescita di questo tipo di dispositivi c'è, nella maggioranza dei casi, il successo

dei MEMS destinati alla rilevazione del movimento.

Si stima che ST arriverà a produrre entro il 2010 un miliardo di questi sensori per un valo-



re di circa 800/900 milioni di dollari annui. Sensori sempre più sofisticati, piccoli ed economici. Ma c'è qualcuno - nel campo dell'Home Entertainment - che va oltre, prefigurando la scomparsa dei controller: è la Microsoft col suo Project Natal per Xbox 360. E' questo il nome in codice di un nuovo, rivoluzionario modo di giocare e di godere del divertimento che assicura l'Xbox 360. In breve: giocare senza controller. Le nostre azioni e le nostre reazioni saranno riconosciute dall'Xbox 360 grazie ad una teleca-

mera RGB, un sensore di profondità, un microfono a matrice multipla e un processore appositamente studiato, il tutto coordinato da software proprietario. Le telecamere e i controller esistenti rilevano il movimento solo in due dimensioni, mentre Project Natal rileva tutti i movimenti del corpo in 3D e risponde a comandi, indicazioni e persino alle sfumature emotive della voce. Sarà sufficiente porsi di fronte al sensore Project Natal per interagire con qualsiasi videogioco, anche solo sfruttando l'espressione del viso.

MEMS RF sono già presenti, senza che questo sia enfatizzato, all'interno di dispositivi di piccole dimensioni per utilizzo sia civile che militare, ma soltanto recentemente hanno raggiunto volumi di produzione su larga scala. I mercati di maggiore interesse per questo tipo di dispositivi sono quello della telefonia cellulare, della telefonia mobile per uso domestico, delle reti wireless, delle radio e dei ricevitori GPS.

Oltre al potenziale basso costo, la tecnologia RF MEMS promette oggi di soddisfare le nuove priorità dei moderni sistemi di comunicazione e dei sistemi senza fili multifunzionali: bassi consumi di potenza, compattezza, riconfigurabilità e alto SNR (Signal/Noise Ratio) dei

dispositivi. La massima parte dei sistemi di telecomunicazione e per applicazioni radar sono stati sviluppati negli ultimi decenni impiegando dispositivi a stato solido come diodi pin o transistor ad effetto di campo (FET), i quali sono però causa di elevate perdite front-end (tra l'inizio e la fine del processo) e di un basso SNR dovuto alle loro non linearità intrinseche. La funzionalità principale che deve svolgere un dispositivo RF è quella di mantenere l'integrità del segnale: trasmetterlo cioè con perdita e riflessione minime, senza interferenze esterne o rumore, e senza generare altri segnali non desiderati. Tutto ciò, alle frequenze più elevate, non è del tutto ovvio. La tecnologia MEMS, grazie alle ridotte dimensioni

(µm-nm), ai bassi consumi di potenza (µW), alle basse perdite di inserzione (da 0,02 a 0,2 dB) e all'eccellente linearità, rappresenta una validissima prospettiva tecnologica per ovviare agli inconvenienti propri dei dispositivi a stato solido. Allo stesso tempo essa consente di soddisfare i più importanti requisiti dei sistemi di TLC nel campo delle microonde e delle onde millimetriche.

La possibilità di modificare la configurazione, e quindi i parametri caratteristici, di un circuito, risulta di grande interesse applicativo in quanto consente di adattare le prestazioni di un sistema alle mutevoli condizioni esterne formando così le basi per la realizzazione di sistemi e micro-sistemi intelligenti. Mediante

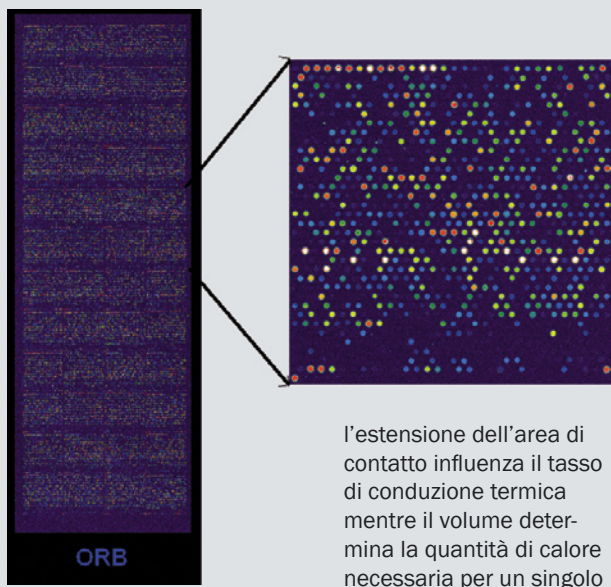
MEMS & biosensori

La tecnologia MEMS ha avuto e avrà sempre di più importanti ricadute anche nel campo della medicina, con una serie di applicazioni pratiche di grande diffusione ma anche con applicazioni d'avanguardia, specie nel campo della genomica. Ad esempio, la tecnica della PCR (Poly-

merase Chain Reaction), la più comune tecnica per l'amplificazione del DNA, può essere miniaturizzata su un singolo chip, con numerosi vantaggi. Le piccole dimensioni delle camere di reazione presentano un elevato rapporto superficie/volume. Dal momento che

ciclo, un elevato rapporto superficie/volume consente una maggiore rapidità nei cicli di PCR. Inoltre, il volume ridotto della camera di reazione fa sì che siano necessari campioni più piccoli e quantità più limitate di reagenti. Infine, se integrato con uno schema di rilevazione come quello per la separazione elettroforetica o con un sistema di marcatura intelligente, l'intero processo si semplifica, si velocizza, diventa più economico e più ripetibile. Nel caso dell'elettroforesi on-a-chip le prime dimostrazioni risalgono al 1992, presso i laboratori svizzeri della Ciba-Geigy, mentre Woolley e Mathies dell'Università di Berkeley furono i primi a dimostrare il sequenziamento del DNA su chip di vetro per mezzo di elettroforesi capillare. Nonostante questo importante risultato, molto rimane ancora da fare prima di riuscire ad avere sul

mercato strumenti portatili per il sequenziamento del DNA. Un sistema completo deve integrare PCR ed elettroforesi (o altri metodi di detection del DNA) e disporre di un sistema per la preparazione e la gestione dei campioni. Tutto ciò richiede l'impiego di molteplici tecnologie abilitanti, tra le quali le apparecchiature MEMS. La parte di elettroforesi per il processo di sequenziamento del DNA è già stata realizzata e commercializzata, ad esempio, da Caliper Life Sciences con i prodotti venduti nella serie LabChip da Agilent Technologies. In questo caso il chip contenente il campione presenta sì dimensioni di pochi centimetri ma, per essere analizzato richiede il sistema Bioanalyzer. Il sistema LabChip è realizzato mediante strati sovrapposti di vetro, lavorati tramite fotolitografia, etching o ablazione laser dopodiché



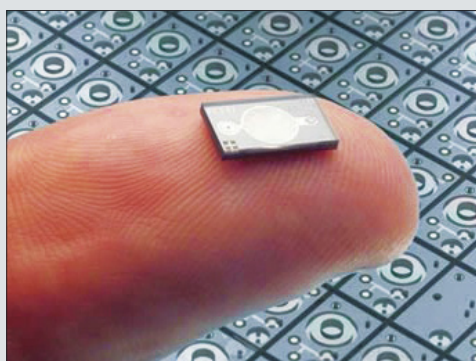
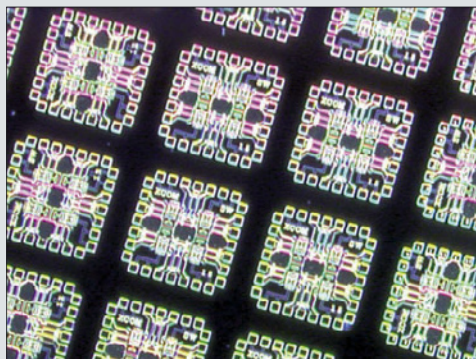
l'uso di commutatori MEMS è possibile realizzare una varietà di dispositivi riconfigurabili, quali sfasatori, divisori e ricombinatori, filtri accordabili, eccetera, così come elementi radianti e antenne riconfigurabili; in tutti questi casi l'utilizzo dei MEMS risulta molto promettente in virtù delle loro caratteristiche di bassa potenza assorbita, linearità ed elevati livelli di integrazione. Tra i settori applicativi emergenti per i quali le strutture MEMS appaiono particolarmente promettenti vi è certamente quello delle antenne riconfigurabili e delle antenne intelligenti. A livello industriale queste tecnologie potranno essere impiegate nelle comunicazioni satellitari e militari, nei sistemi di navigazione, nei sensori e nell'avionica. Attualmente uno dei principali

ambiti di ricerca è quello che riguarda il settore dei segnali di switch RF. Fino ad ora sono stati sviluppati due tipi di interruttori per questa soluzione: il primo è in grado di gestire sia corrente alternata che corrente continua attraverso una serie di contatti metallici, il secondo invece è un deviatore che consente lo switching di corrente alternata attraverso un dielettrico sottile, consentendo di eliminare i problemi causati dall'alternanza caldo/freddo sul contatto.

MEMS OTTICI

I MEMS ottici rappresentano un'applicazione molto particolare, nella quale i dispositivi sono utilizzati per dirigere, guidare, filtrare e, in alcuni casi, amplificare la luce. Gli utilizzi più noti dei MEMS ottici sono costituiti dallo

switching ottico e dalla proiezione digitale (DLP). Nel primo caso si utilizzano micro specchi per deviare la luce proveniente da un input in fibra ottica ad un output, sempre in fibra ottica. Sono stati fabbricati numerosi tipi di microspecchi MEMS, le cui funzioni possono essere classificate a seconda delle configurazioni. Nel secondo caso, quello della proiezione digitale, si utilizzano array di micro-specchi: questa tecnologia è stata ideata da Texas Instrument qualche anno fa e ultimamente ha ottenuto lusinghieri risultati commerciali. Si tratta sempre di un dispositivo MEMS, costituito da un enorme array di micro-specchi mobili, in grado di cambiare angolo di configurazione anche 15.000 volte al secondo. Esistono vari tipi di Digital Light Processor (DLP): si è partiti



sovrapposti e saldati tra loro mediante un processo termico. L'impiego del vetro rende il dispositivo economico nonostante sia monouso, una caratteristica necessaria per evitare contaminazioni da campioni precedenti, possibili intasamenti e usura dei materiali.

da configurazioni minime 3x3 per arrivare fino a 1280x1024, ossia 1,3 milioni di specchi comandabili singolarmente. Nell'apposito riquadro ci occupiamo in maniera approfondita di questa tecnologia. Nel settore dell'optoelettronica, la rilevazione delle informazioni e la loro elaborazione sono due ambiti strettamente correlati ma le corrispondenti tecnologie sviluppate fino ad oggi viaggiano quasi su due binari paralleli. L'universo dell'elaborazione è totalmente dominato dall'elettronica e dal silicio, quello dell'acquisizione delle informazioni è dominato dalla fotonica (soprattutto perchè la generazione, il controllo e la gestione della luce giocano un ruolo fondamentale nel rilevamento) e si basa su un'ampia varietà di materiali:

uno degli obiettivi su cui lavorare in futuro sarà senz'altro quello della convergenza tra questi due universi che richiedono tecnologie e materiali differenti, in grado di svolgere diverse funzioni con livelli di performance differenti: l'obiettivo è quello di riuscire ad integrarle in un unico dispositivo sempre più intelligente.

MEMS microfluidici e bioMEMS

I MEMS microfluidici sono dispositivi che trasportano, erogano, combinano e/o separano fluidi a livello microscopico. Le industrie hanno da poco incominciato a cogliere il potenziale dei sistemi di spruzzamento, trasporto e misurazione basati sui MEMS. L'applicazione di tali dispositivi consente di ridurre le quantità di liquidi impiegati, minimizza gli scarti e consente di operare in serie o in parallelo su dispositivi multipli.

Applicazioni tipiche dei MEMS microfluidici comprendono valvole, pompe e sistemi per getto d'inchiostro: in questo caso i MEMS sono utilizzati per migliorare la risoluzione della stampa (maggiore numero di DPI grazie ad una maggiore densità degli ugelli) e ridurre il consumo di inchiostro, con un incremento della vita media delle cartucce. Anche in Italia si lavora con queste tecnologie. Uno dei più avanzati centri di ricerca, sviluppo e produzione di MEMS in Europa si trova in Valle d'Aosta, ad Arnad, nel polo tecnologico del Gruppo Olivetti Tecnost (Telecom Italia).

Altre applicazioni riguardano il campionamento e l'analisi delle acque, i test non invasivi, eccetera. In questi casi numerosi dispositivi possono essere collocati uno vicino all'altro in punti strategici delle reti idriche per

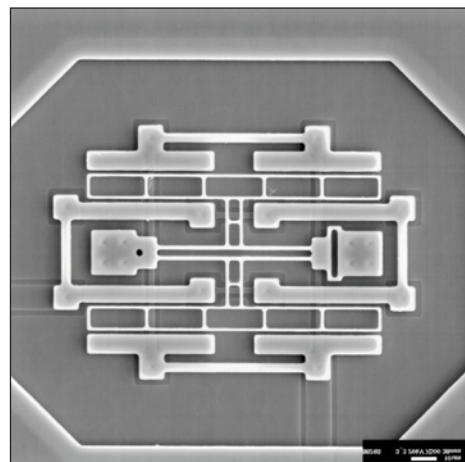
misurare la qualità dell'acqua e l'eventuale presenza di sostanze tossiche.

Così come i MEMS microfluidici, i bioMEMS sono progettati per trasportare, mescolare e/o separare liquidi a livello microscopico. Una differenza rilevante tra i due tipi di dispositivi è costituita dal fatto che i bioMEMS sono in grado di trattare sia liquidi contenenti materiale biologico che fluidi biologici (es. sangue).

I bioMEMS costituiscono un gruppo molto vasto di dispositivi che comprende apparecchiature integrate e miniaturizzate da utilizzare per scopi biologici/biochimici nella ricerca e sviluppo, nella diagnostica, nella terapia e nel monitoraggio.

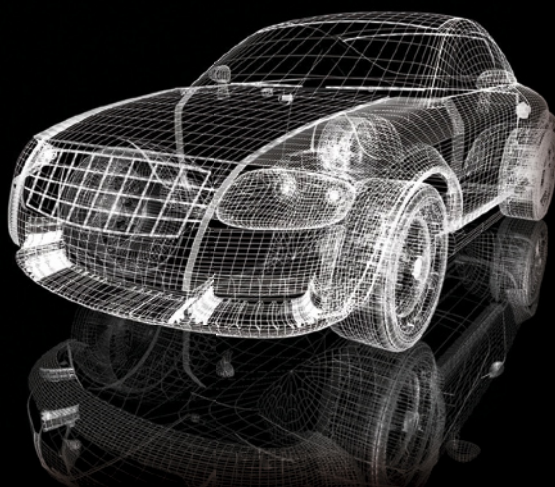
Tali dispositivi vengono chiamati, a volte con eccessiva variabilità, biochip, bioMEMS, microarray, chip a DNA, lab-on-a-chip, cell-chip, micro-impianti, drug delivery systems (DDS) oppure ancora μ TAS (micro Total Analysis Systems), spesso senza una idea chiara delle funzioni e delle caratteristiche di ciascuno.

In ogni caso, al di là delle etichette, le applicazioni possibili sono numerose: la microdialisi, i biosensori e le analisi dei cosiddetti Lab-on-a-chip sono aree tuttora in fase di grande sviluppo. Queste e molte altre saranno





Grazie alla tecnologia MEMS, le dimensioni e le prestazioni degli accelerometri hanno raggiunto livelli impensabili fino a pochi anni.



senz'altro implementate in applicazioni su larga scala. Ciò che non risulta del tutto chiaro è l'orizzonte temporale: ad oggi infatti, un po' per la vastità del settore e degli interessi in gioco, un po' per il fatto di operare in assoluta avanguardia tecnologica, un po' per l'intrinseca rapidità dei cambiamenti, non è possibile individuare un preciso e definito "stato dell'arte".

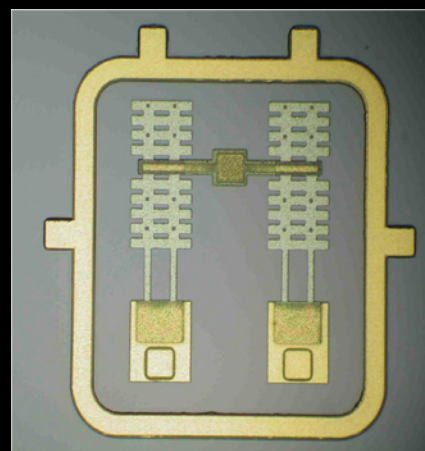
I bioMEMS e la microfluidica sono legati fra loro in maniera inscindibile, dal momento che la maggior parte dei dispositivi per analisi biologiche e mediche operano con campioni in forma liquida. Al di fuori delle applicazioni di analisi biologica, la microfluidica trova applicazione nell'analisi chimica, nella sintesi dei farmaci, nel drug delivery e nella sintesi on-site di sostanze chimiche pericolose. Per quanto riguarda il micro-pompaggio e la micro-miscelazione (mixing), i canali di trasporto utilizzati nelle apparecchiature per microfluidica hanno tipicamente sezione rettangolare, sono dotati di capillari, fissati ad un substrato con una copertura superiore e possono essere inframezzati da strati di gel. Le dimensioni della loro sezione vanno da 10 a 100 μm , con lunghezze che

vanno dall'ordine delle decine di micrometri ai centimetri. Per le analisi biologiche le tecniche di trasporto e di pompaggio dei liquidi comprendono il differenziale di pressione, la capillarità, l'elettroforesi, l'elettrosmosi, la forza elettrodinamica e la forza magnetoidrodinamica. Le prime quattro sono piuttosto comuni anche nel mondo macro, le altre due sono tipiche della scala micro. A livello micro le correnti di flusso all'interno dei canali di trasporto sono ovviamente molto più deboli che a livello macro. Questa considerazione, apparentemente banale, ha importanti implicazioni per la miscelazione dei campioni. A livello macro, la semplice unione di due canali di trasporto fra loro consente il mescolarsi dei due flussi; a livello micro due "correnti" provenienti da due canali differenti continueranno a scorrere una a fianco all'altra e potranno mescolarsi soltanto per diffusione.

Per quanto riguarda l'analisi del DNA, la tecnica della PCR (Polymerase Chain Reaction), la più comune tecnica per l'amplificazione del DNA può essere miniaturizzata su un singolo chip, con numerosi vantaggi. Le piccole dimensioni delle camere di reazione presentano

un elevato rapporto superficie/volume. Dal momento che l'estensione dell'area contatto influenza il tasso di conduzione termica e il volume determina la quantità di calore necessaria per un singolo ciclo, un elevato rapporto superficie/volume consente una maggiore rapidità nei cicli di PCR. Una volta che i campioni di DNA da analizzare sono stati amplificati (ad es. mediante PCR), possono essere "letti" tramite array per l'ibridazione, insieme preordinati di sequenze differenti di nucleotidi agganciate ad un substrato. Le sezioni di DNA da identificare (con lunghezze nell'ordine di centinaia o migliaia di basi), marcate ad un'estremità con un colorante fluorescente, vengono poste in una soluzione nella quale sono libere di muoversi e di andare ad agganciarsi alle sequenze loro complementari. Una volta lavato e illuminato a fluorescenza, il DNA rivela quali sequenze hanno subito il processo di ibridazione mentre risulta possibile identificare le sequenze del DNA sconosciuto.

Questo approccio risulta particolarmente utile nell'individuazione di specifiche mutazioni a livello genico e nella ricerca di patogeni noti. ■



Le più recenti normative in materia di efficienza energetica nei caricabatterie mettono a dura prova i progettisti; un modo per rispettarle consiste nello sfruttare la tecnica PSR (Primary Side Regulation). Vediamo di che si tratta.

PROGETTARE CARICABATTERIE CON LA TECNICA PSR

..... a cura della REDAZIONE

Nonostante la maggior parte degli elettrodomestici e delle apparecchiature per l'ufficio sia collegata alle prese a muro e quindi alimentata da corrente alternata (AC, Alternating Current) ad alta tensione, praticamente tutti i relativi circuiti interni richiedono invece corrente continua (DC, Direct Current) a bassa tensione. Ne consegue la necessità di alimentatori in grado di convertire la corrente da alternata a continua. Secondo uno studio condotto da Ecos Consulting, solamente negli Stati Uniti sono presenti quasi 3 miliardi di alimentatori AC/DC e in tutto il mondo quasi 10 miliardi. Con la sempre crescente diffusione di questi dispositivi, il loro impatto sull'ambiente in termini di efficienza elettrica ha suscitato la crescente attenzione della comunità internazionale. La *California Energy Commission* (CEC) ha proposto standard di efficienza obbligatori per gli alimentatori esterni, e numerosi altri Paesi che attualmente si affidano a programmi di regolamentazione volontari stanno valutando

l'adozione di standard obbligatori per promuovere l'efficienza degli alimentatori. Oltre la metà degli alimentatori esterni è utilizzata per dispositivi elettronici portatili quali laptop, cellulari e lettori MP3 e, pertanto, richiede capacità di regolazione della tensione e della corrente di uscita per il caricabatteria ottenibili da circuiti come quello illustrato nella **Fig. 1**. Quando le applicazioni richiedono una regolazione precisa della corrente erogata, è sempre necessario un circuito di misurazione, il quale, sfortunatamente, si trova in serie all'uscita e comporta un'ulteriore perdita di corrente, almeno nelle architetture SSR (*Secondary Side Regulation*). Per i progettisti di alimentatori chiamati a rispettare normative sempre più severe, la misurazione della corrente di uscita rappresenta da sempre una sfida. L'applicazione della tecnica PSR (*Primary Side Regulation*) può rappresentare la soluzione ottimale per alleviare l'aggravio associato al rispetto della normativa CEC nella progettazione degli alimen-

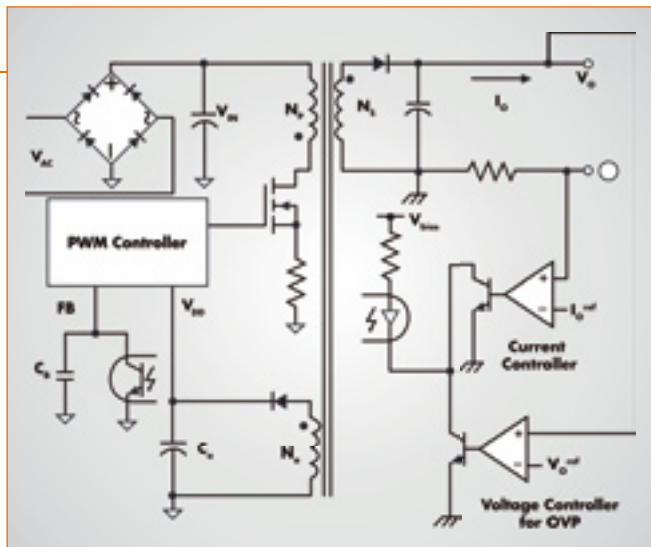


Fig. 1 - Convertitore flyback SSR
(Secondary Side Regulation) convenzionale.

tatori. La tecnica PSR permette di misurare con precisione la corrente e la tensione di uscita unicamente attraverso le informazioni disponibili dal lato primario dell'alimentatore, eliminando non soltanto le perdite associate alla misurazione della corrente di uscita, ma anche tutta la circuiteria per il feedback dal secondario. Ciò facilita la progettazione di circuiti di alimentazione più efficienti senza incorrere in costi proibitivi. In queste pagine descriviamo i principi alla base del funzionamento della tecnica PSR e presentiamo un controller PWM PSR altamente integrato che offre diversi vantaggi rispetto ai metodi SSR.

PRINCIPI BASE DELLA TECNICA PSR

Nella Fig. 2 sono illustrati lo schema elettrico del circuito base di un convertitore flyback PSR e le sue forme d'onda caratteristiche. In linea generale, nella tecnica PSR è preferibile utilizzare la modalità DCM (Discontinuous

Conduction Mode) in considerazione della migliore regolazione ottenibile in uscita. Il fondamento della tecnica PSR consiste nell'ottenere le informazioni relative alla tensione e alla corrente di uscita senza doverle misurare direttamente. Una volta ottenuti tali valori, la regolazione può essere effettuata tramite il normale controllo PI. Nella fase di funzionamento del MOSFET (TON), la tensione di entrata (V_{IN}) fluisce all'interno dell'induttore del

lato primario (LM). La corrente all'interno del MOSFET (I_{ds}) cresce quindi linearmente da zero al valore di picco (I_{pk}). In questa fase l'energia viene assorbita dall'ingresso e immagazzinata nell'induttore. Quando il MOSFET si spegne, l'energia immagazzinata nell'induttore provoca la conduzione del diodo raddrizzatore (D). Nella fase di funzionamento del diodo (T_D) la tensione di uscita (V_o) fluisce all'interno dell'induttore del lato secondario ($L_m \times N_s^2 / N_p^2$) e la corrente all'interno del diodo (I_D) decresce linearmente dal valore di picco ($I_{pk} \times N_p / N_s$) fino allo zero. Al termine della fase di funzionamento del diodo, tutta l'energia immagazzinata all'interno dell'induttore è stata rilasciata all'uscita. In questa fase la som-

ma della tensione di uscita e della caduta di tensione nel diodo si riflette nel lato dell'avvolgimento ausiliario nella formula $(V_o + V_F) \times N_a / N_s$. Poiché la caduta di tensione nel diodo decresce con il diminuire della corrente, la tensione nell'avvolgimento ausiliario rispecchia più efficacemente la tensione di uscita al termine del periodo di conduzione del diodo, quando la corrente si riduce a zero. In questo modo è sufficiente misurare la tensione dell'avvolgimento al termine del periodo di conduzione del diodo per ottenere le informazioni relative alla corrente di uscita. Il tempo di conduzione del diodo può essere rilevato monitorando la tensione dell'avvolgimento ausiliario. La stima della corren-

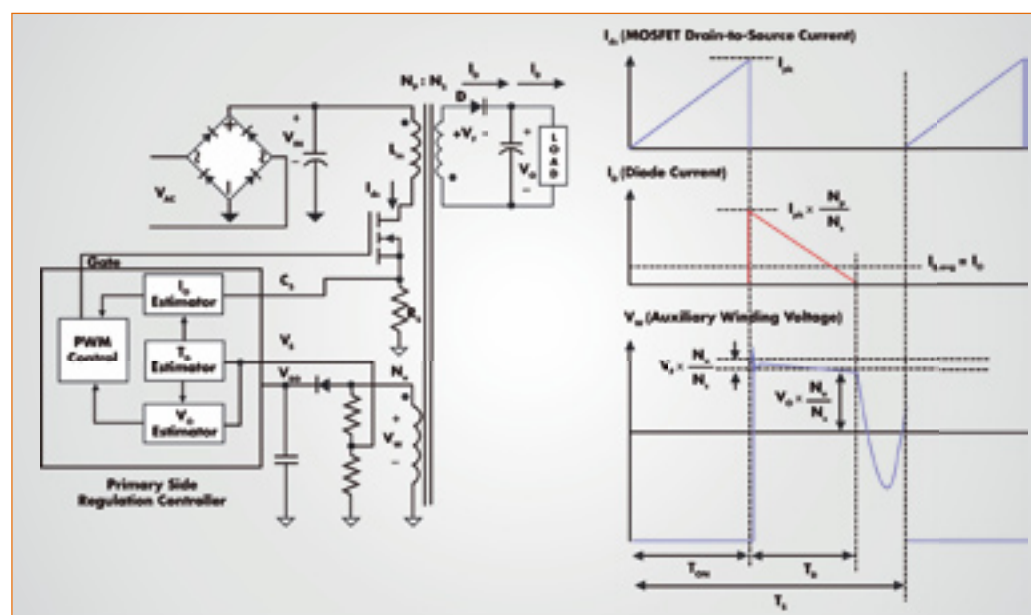


Fig. 2 - Schema di un convertitore flyback di tipo PSR e sue forme d'onda.

te di uscita richiede però lo svolgimento di alcune moltiplicazioni. Ipotizzando che la corrente di uscita sia uguale al valore medio della corrente all'interno del diodo in regime stazionario, la corrente di uscita può essere calcolata attraverso la formula:

$$I_o = I_{pk} \times (N_p / N_s) \times (T_D / 2T_s)$$

L'estimatore della corrente di uscita ricava il valore di picco della corrente assorbita mediante un apposito circuito e calcola la corrente di uscita effettiva basandosi sul tempo di conduzione del diodo (T_D).

CONTROLLER PRIMARY SIDE REGULATION INTEGRATO

Una tecnologia che risponde alle esigenze dei progetti di alimentatori PSR è quella dei controller PWM PSR come il FAN102 Fairchild Semiconductor, il quale semplifica considerevolmente l'adeguamento ai requisiti più severi in termini di efficienza energetica, eliminando contemporaneamente componenti esterni che aggiungono problemi di costo e affidabilità. Con il dispositivo FAN102 i progettisti di alimentatori hanno l'opportunità di rispettare le normative esistenti e future, come lo standard *Energy Star* e le specifiche introdotte dalla *California Energy*

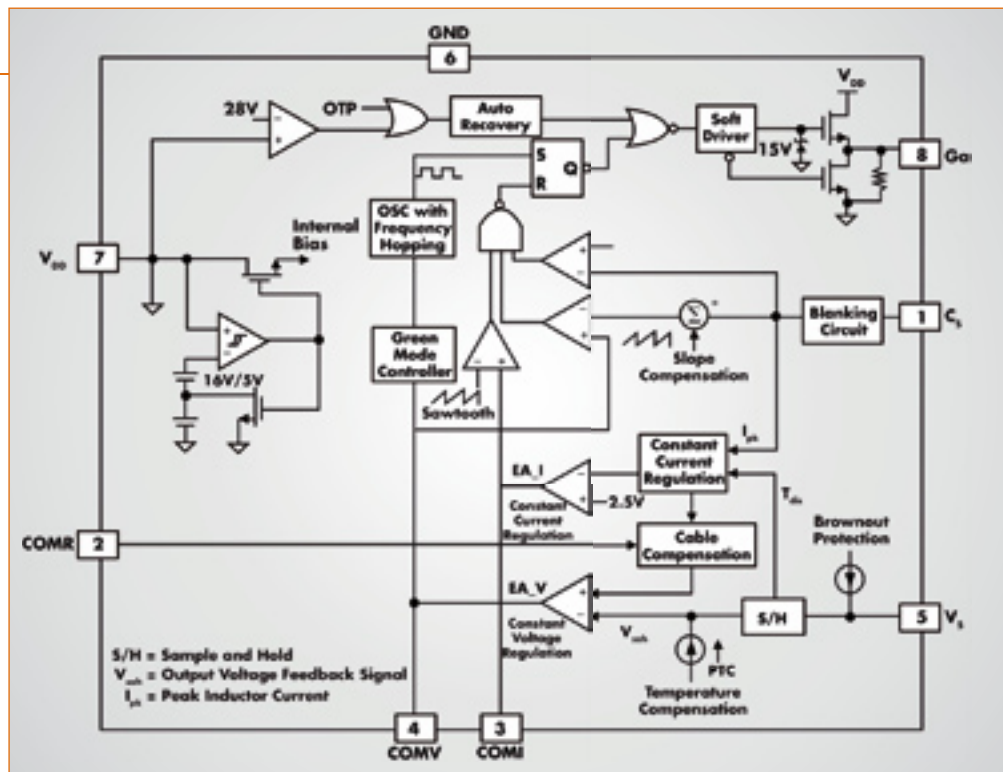


Fig. 3 - Schema a blocchi dell'interno di uno switching di potenza integrato (FSCQ-Series).

Commission. Questo dispositivo incorpora inoltre una modalità operativa "green mode" per il funzionamento in standby e soddisfa i criteri stabiliti dalla *1 Watt Initiative* della IEA (International Energy Agency) finalizzata a contenere il consumo di corrente in standby in non più di un watt. La Fig. 3 mostra la struttura del dispositivo FAN102: si vede che esso incorpora un circuito di compensazione della caduta di tensione nel cavo di uscita e della variazione di temperatura dei componenti esterni, che assicura un alto grado di accuratezza persino al terminale del cavo di alimentazione. L'oscillatore interno utilizza la tecnica del salto di frequenza per ridurre le emissioni EMI. Un'altra importante caratte-

ristica del FAN102 è l'ampia escursione della tensione V_{DD}, tra 5 e 28 V. Quando l'alimentatore opera in modalità di corrente di uscita costante, come nel caso dell'alimentazione per i circuiti integrati di controllo, la V_{DD} varia in accordo con la tensione di uscita. Pertanto è l'escursione della V_{DD} a determinare la gamma di controllo della corrente costante e il dispositivo FAN102 permette di ottenere una regolazione stabile anche con tensioni di uscita inferiori a un quarto del suo valore nominale.

IN CONCLUSIONE

Pressati dalle normative vigenti ed emergenti in materia di efficienza energetica da un lato e dalle esigenze del mercato dall'altro, i progettisti di alimentatori sono

impegnati nella ricerca di superiori livelli di efficienza, robustezza e prestazioni senza incorrere in costi aggiuntivi. Il controller PSR di Fairchild promette di migliorare drasticamente i costi e le performance complessive dei futuri convertitori di corrente AC/DC e DC/DC. Una combinazione proprietaria di tecniche di campionamento e stima del valore di uscita fornisce un'implementazione efficiente e rigorosa delle normative che permette di realizzare alimentatori AC adatti a qualunque tipo di applicazione, dai cellulari ai telefoni cordless fino ai lettori MP3. Tali alimentatori ora possono essere più compatti, meno costosi e più efficienti grazie a tecnologie a semiconduttore convenienti ed efficaci.

CONTROLLO MOTORI DC A DUE CANALI

Comandato tramite il segnale impulsivo proporzionale fornito da qualsiasi ricevitore a due canali per radiomodelli, può azionare uno o due motori in corrente continua in modo sia proporzionale che differenziale. Dispone di un pulsante per memorizzare la posizione del neutro dei joystick della trasmettente.

..... di CRISTIANO RUGGERI

Gli appassionati di modellismo di oggi sono molto avvantaggiati rispetto a quelli di alcuni anni fa, perché le tecnologie elettroniche si sono evolute decisamente e continuamente in tutti i possibili campi di applicazione, compresi quelli prettamente ludici quali giochi e modellismo dinamico. Se, infatti, le vecchie guardie di modellisti potevano disporre, soldi (e tanti...) permettendo, delle attrezzature per realizzare macchinine e navi radio-

comandate dotate di movimenti semplici (avanti, indietro, destra, sinistra e poco altro ancora), le nuove generazioni possono avvalersi di sistemi ben più complessi e meno costosi, che permettono di realizzare movimentazioni più veritiere, complicate e fantasiose. A conferma di ciò, è sufficiente recarsi presso un'esposizione di modellismo (come ad esempio l'Hobby Model

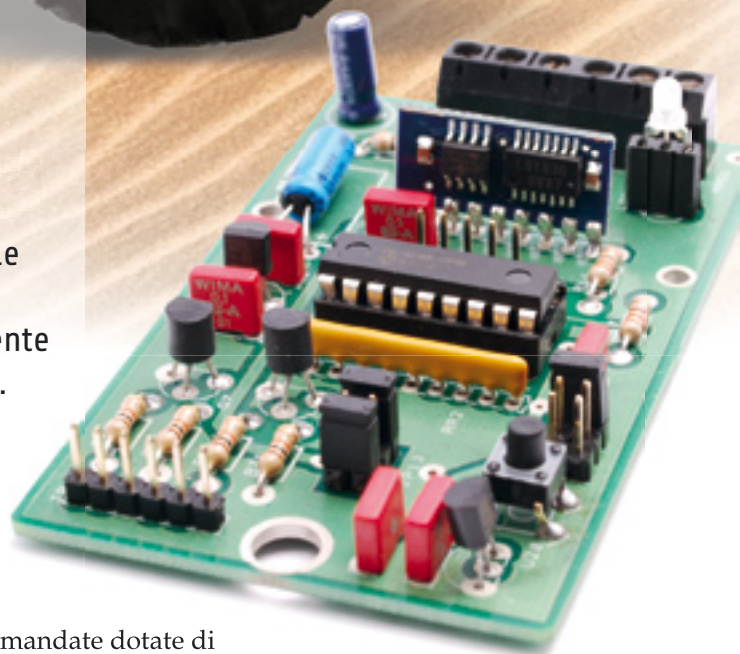
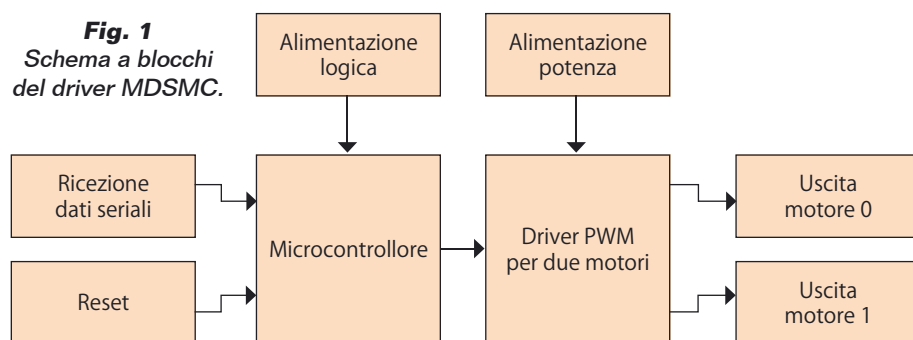


Fig. 1
Schema a blocchi
del driver MDSMC.



Expo, che si tiene a Novegno negli stessi padiglioni dove si tiene il Radiant, esposizione nota agli appassionati di elettronica) per poter ammirare elicotteri e aerei dinamici (volanti!) dalle dimensioni che vanno da pochi centimetri a oltre un metro, ruspe e scavatori a comando idraulico, navi da guerra in cui si muovono radar, ponti di volo, pompe di sentina e ancore, carri armati. In particolare, alcuni modelli bellici, oltre a muoversi in tutte le direzioni, grazie ad alcuni sensori e dispositivi di sbilanciamento, possono anche simulare un combattimento tra loro, con tanto di rinculo dovuto allo sparo del cannone sulla torretta e sbandamento dipendente dal colpo subito. In questi ultimi anni, pure la robotica dilettantistica ha vissuto e sta vivendo un periodo di grande prosperità: è molto facile trovare, sia presso negozi specializzati che semplici cartolerie o negozi di giocattoli e gadget, robot combattenti, bipedi deambu-

lanti simili a quello che abbiamo pubblicato nel fascicolo numero 133, robot antropomorfi in grado di riprodurre i movimenti delle principali articolazioni umane (come quello pubblicato in edicola da DeAgostini), e molto altro ancora.

Insomma per chi ha ambizioni modellistiche, oggi, c'è solo l'imbarazzo della scelta. Vi interessate ai modelli di auto radiocomandate? Kyosho produce, oltre ai modelli, anche i radiocomandi, le riceventi e i servocomandi adatti. Preferite aerei, elicotteri, imbarcazioni e sommergibili? Le elettroniche Futaba e Graupner sono sempre una scelta vincente. Se, invece la robotica è la vostra passione, allora dovete conoscere obbligatoriamente Pololu (www.pololu.com), azienda americana che si è specializzata nella realizzazione di sistemi elettronici, elettromeccanici e meccanici per realizzare il robot "su misura". E proprio su un dispositivo Pololu, l'MDSMC (Micro Dual

Serial Motor Controller) abbiamo basato il progetto che stiamo per presentarvi in queste pagine: si tratta di un controllo motori per applicazioni modellistiche estremamente versatile, utilizzabile dovunque sia necessario gestire uno o due motori e comandabile tramite due canali di un qualsiasi radiocontrollo proporzionale. Vediamo ora nel dettaglio sia il dispositivo MDSMC che la sua applicazione in pratica.

IL DRIVER MDSMC

Come dice il nome stesso, questo dispositivo dalle dimensioni molto compatte (circa 23 x 12 mm, con 9 pin SIL a passo 2,54 mm) svolge la funzione di driver per due motori in corrente continua di piccole dimensioni (che chiameremo motore 0 ed 1) e può essere schematizzato con il disegno nella Fig. 1.

Alimentazione

Il dispositivo è dotato di due alimentazioni distinte, una per la logica ed una per la sezione di potenza. La prima viene applicata sul pin 3 e può avere valore compreso tra 2,5 e 5,5 Vcc ben stabilizzati ed è normalmente la stessa tensione utilizzata per alimentare il microprocessore della scheda che ospita il nostro MDSMC. La seconda, invece (pin 1) fornisce energia al driver

RM ELETTRONICA

Forniture per hobbisti, scuole e industria

Distributore autorizzato:

FUTURA ELETTRONICA

ELSE Kit

ELETTRONICA

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

Il tuo punto
di riferimento
per l'elettronica
a ROMA

PWM che pilota i motori; tale tensione può avere un'escursione più ampia ($1,8 \div 9,0$ Vcc) di quella applicabile al pin 3, ma nulla vieta di collegare entrambe le linee ad un'unica sorgente, a patto che questa soddisfi sia i requisiti della tensione per la logica ($2,5 \div 5,5$ Vcc) che quelli della corrente richiesta dalla sezione di potenza (massimo 2 A). La massa del dispositivo (pin 2) comune sia alla logica che al driver PWM, coincide con la massa della scheda logica base.

Linea di reset

Localizzata sul pin 5, permette di riportare il driver nella condizione nota di "motori fermi e attesa comando"; durante il normale funzionamento deve rimanere a livello logico alto. Un livello logico basso della durata di almeno due microsecondi forza nel dispositivo la condizione di reset.

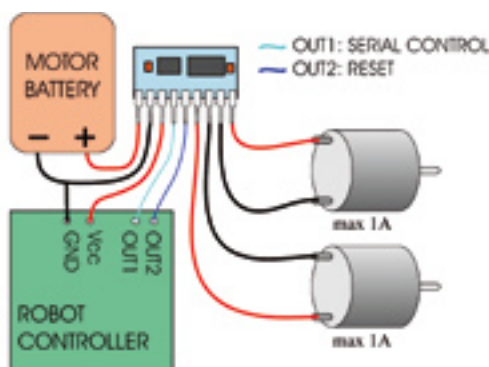


Fig. 2

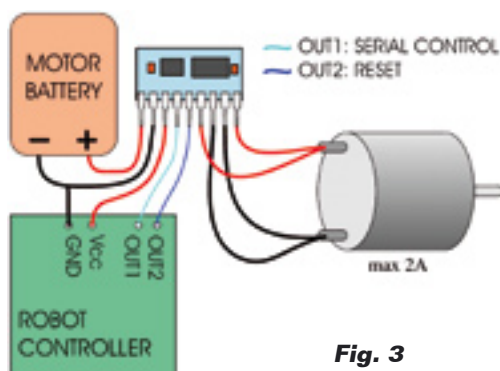


Fig. 3

La linea può essere collegata ad un'uscita del microprocessore preposto al controllo del driver, oppure messa a livello logico alto tramite una resistenza di pull-up del valore di 4,7 kohm o porta-

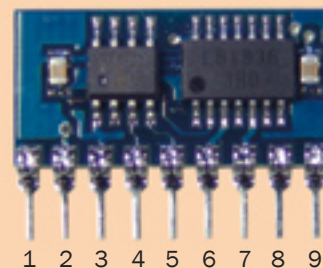
L'ingresso della connessione seriale si trova sul pin 4 e riceve i comandi inviati dal microcontrollore; il formato è asincrono con velocità compresa tra 1.200 e 19.200 baud, 8 bit dati,

Il modulo MDSMC

Il driver MDSMC è un modulo molto compatto che può essere montato a inserzione sia su un connettore a passo 2,54 mm, sia direttamente sullo stampato. È realizzato in tecnologia SMD

e integra a bordo un processore PIC12F623 e un doppio driver con ponte ad H modello LB1836. Sul lato posteriore è possibile incollare un piccolo dissipatore di calore.

Pin	Funzione
1	Alimentazione motori (1,8÷9Vdc)
2	Massa
3	Alimentazione logica (2,5÷5,5Vdc)
4	Ingresso seriale
5	Ingresso di reset
6	Uscita positiva Motore 1
7	Uscita negativa Motore 1
8	Uscita negativa Motore 0
9	Uscita positiva Motore 0

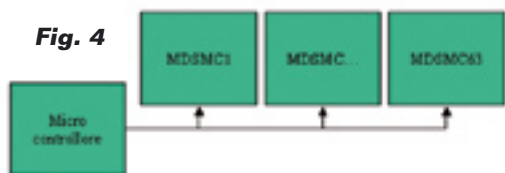


- Dimensioni: 23 x 11,5 mm circa
- Numero canali motore: 2, con uscita doppia (ponte ad H)
- Regolazione velocità: 127 livelli, sia in avanti che indietro
- Corrente massima ai motori: in modalità a due motori 1 A per motore; in modalità a un motore, 2 A.
- Alimentazione motori: 1,8÷9 Vcc, anche non stabilizzati
- Alimentazione logica: 2,5÷5,5 Vcc stabilizzati, 20 mA tipici
- Frequenza di PWM: 600 Hz, nella modalità a due motori; 750 Hz nella modalità a un motore.
- Velocità di comunicazione: 1.200÷19.200 baud con riconoscimento automatico al primo comando ricevuto.
- Protezioni: limitazione corrente in uscita per sovratemperatura.

ta a massa con un pulsante. La linea di reset è consigliabile sia collegata al micro che gestisce il modulo, così da avere un controllo diretto sul driver sia hardware che firmare (linea comandi). Come spiegato più avanti, nel nostro progetto è stata collegata ad un I/O del microcontrollore di gestione.

Ingresso seriale

1 bit di stop e parità N (nessuna parità). La velocità viene riconosciuta automaticamente al primo comando ricevuto dopo un reset e non può essere modificata fino al reset successivo. Tale ingresso può essere collegato direttamente alla linea di trasmissione seriale di un microcontrollore, secondo lo standard RS232 TTL che prevede la logica invertita, ossia il livello logico 1 corrispondente



Start byte = 0x80	Byte di identificazione	Data byte 1	Data byte 2
-------------------	-------------------------	-------------	-------------

Start byte = 0x80	Cambio configurazione = 0x02	Nuove impostazioni = 0x0+0x7F
-------------------	------------------------------	-------------------------------

a 0 V e il livello logico 0 pari al valore della tensione di alimentazione. È comunque possibile gestire il dispositivo tramite la seriale di un PC attraverso un adattatore di livello, quale l'integrato MAX232 della Maxim.

Funzionamento

a uno o due motori

Come visibile dalla pin-out del modulo, le uscite positiva e negativa del motore 1 fanno capo ai pin 6 e 7, mentre quelle del motore 0 si trovano sui pin 9 e 8. Esse possono alimentare sia due motori differenti secondo lo schema visibile nella Fig. 2, ovvero un singolo motore, come illustrato nella Fig. 3.

Il modo di funzionamento viene impostato tramite un comando seriale specifico, che verrà descritto nel paragrafo relativo alla struttura dei comandi.

La struttura dei comandi

A collegamento effettuato, non rimane che inviare nel modo corretto i comandi necessari tanto per controllare i motori quanto per configurare il circuito, secondo un protocollo compatibile con altri dispositivi Pololu. La compatibilità del protocollo seriale rende possibile comandare più dispositivi attraverso una sola linea seriale, realizzando così dei sistemi elettronici di automazione tanto complessi quanto immediati da gestire (vedere Fig. 4). Il protocollo, schematizzato nella

Fig. 5, richiede uno start-byte seguito da un byte di identificazione di dispositivo/comando e da alcuni Data-byte che dipendono dal dispositivo/comando identificato con il secondo byte. Lo start-byte viene individuato dal bit 7 (MSb) fisso a 1 logico e deve valere sempre 0x80 (128 in ASCII): ciò significa che tutti gli altri byte devono avere il bit 7 a zero e, pertanto, possono valere da 0x00 (0) a 0x7F (127). Tutte le volte che un byte viene trasmesso, ogni dispositivo controlla se il bit 7 di tale byte è a 1 (start byte): nel caso lo sia, ogni dispositivo posto sulla linea seriale analizza il byte successivo allo start byte (byte di identificazione) per verificare se i data byte sono indirizzati ad esso oppure ad un altro dispositivo.

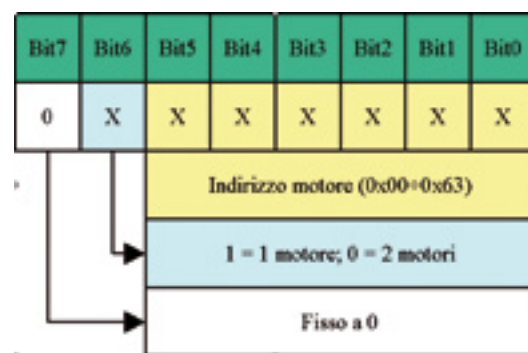
Configurazione dei parametri

L'MDSMC consente di configurare sia la modalità di funzionamento a uno o due motori sia l'indirizzo base dei motori. Questa seconda opzione è necessaria in quanto, volendo utilizzare più driver sulla stessa linea seriale, occorrerebbe assegnare un indirizzo ad ogni motore, ad esempio motori 2 e 3 sulla scheda MDSMC1, motori 4 e 5 sulla scheda MDSMC2, motore 6 sulla scheda MDSMC3 e così via...

Nota molto importante: se sulla linea seriale sono presenti più dispositivi compatibili con il protocollo dell'MDSMC, durante la fase di configurazione è necessario collegare un solo dispositivo alla volta. Se non si facesse così, tutti i dispositivi verrebbero configurati allo stesso modo dall'ultimo comando ricevuto, vanificando la possibilità dell'indirizzamento.

Esaurite le necessarie premesse, la configurazione predefinita prevede la modalità a due motori,

Fig. 7



con indirizzo base 2 per motore 0 e 3 per motore 1. Il cambio di configurazione viene effettuato inviando tre byte in sequenza, secondo la sintassi visibile nella Fig. 6. Notate che quando il byte successivo allo start byte (byte di identificazione) assume valore 0x02, esso viene "identificato" dall'MDSMC come comando di cambio configurazione, comando che pone il dispositivo in attesa di un terzo byte, contenente le nuove impostazioni.

Il byte "Nuove impostazioni" è diviso in tre parti, come visibile nella Fig. 7.

Il bit 7 rimane fisso a zero, condizione che permette al Pololu di comprendere che non si tratti di uno start byte. Il bit 6 imposta il modo a 1 motore (1 logico) o a 2 motori (0 logico) mentre i bit 5÷0 specificano gli indirizzi motore cui il modulo risponderà durante il normale funzionamento. In modalità ad un motore, l'MDSMC risponde solo all'indirizzo definito con i bit 5÷0; in modalità a 2 motori, invece, il driver parte dall'indirizzo ricevuto per associare ai motori due indirizzi differenti secondo la logica "motore dispari indirizzo dispari e viceversa".

Facciamo alcuni esempi:

- 01000111 (0x47, decimale 71)

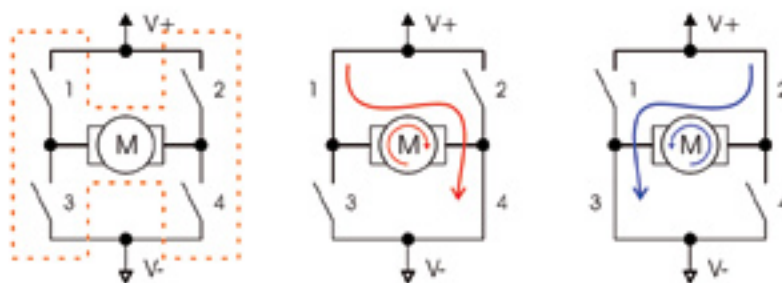
Fig. 8

Start byte = 0x80	Byte di identificazione = 0x00	Motore e direzione	Velocità motore
-------------------	--------------------------------	--------------------	-----------------

Fig. 9



Fig. 10



imposta la modalità a 1 motore con indirizzo base 7;

- 00001111 (0x0F, decimale 15) imposta la modalità a 2 motori con indirizzo base 14 per il motore 0 e 15 per il motore 1;
- 00001110 (0x0E, decimale 14) imposta la modalità a 2 motori con indirizzo base 14 per il motore 0 e 15 per il motore 1;
- 00000010 (0x02, decimale 2) imposta la modalità a 2 motori con indirizzo base 2 per il motore 0 e 3 per il motore 1.

Sappiate che tutti i driver rispondono sempre all'indirizzo 0 (bit 5÷0 pari a 000000) per il motore 0, e 1 (000001) per il motore 1, a prescindere dall'indirizzo impostato. A conti fatti, esclusi gli indirizzi 0 e 1, rimangono a disposizione ben 62 indirizzi differenti (2÷63) che permettono di utilizzare fino a 62 MDSMC in modalità a 1 motore e fino a 31

in modalità a due motori. Sono possibili anche combinazioni miste di differenti moduli Pololu, purché non si superi il massimo di 62 indirizzi differenti.

Una volta ricevuto il comando di cambio configurazione, il driver conferma di avere accettato i nuovi parametri facendo pulsare il pin 8 nel modo a un motore e il 9 nel modo a due motori. Il numero di pulsazioni dipende dall'indirizzo assegnato e vale sempre "indirizzo motore + 1". Facciamo ancora alcuni esempi per chiarire il concetto:

- modalità un motore, indirizzo base 5; il pin 8 pulsa 6 volte;
- modalità due motori, indirizzi base 12 e 13; il pin 9 pulsa 14 volte;
- modalità due motori, indirizzi base 2 e 3; il pin 9 pulsa 4 volte.

A questo punto, per poter utilizzare la nuova configurazione

è necessario effettuare un ciclo di reset, azione che può essere eseguita sia movendo la linea di reset che spegnendo e riaccendendo il driver.

Uso in pratica

Modificata la configurazione, è possibile iniziare ad utilizzare il driver: per fare ciò, si invia una stringa di quattro byte secondo la sintassi esposta nella Fig. 8. Come sempre, il primo è lo start byte, fisso a 0x80, seguito dal byte di identificazione; il valore 0x00 viene interpretato dall'MD-SMC come comando di cambio di velocità e direzione di un motore. Questo comando è specifico per i controller di motori a due canali e viene ignorato da tutti gli altri dispositivi. Il terzo byte contiene l'indirizzo del motore e la direzione secondo la struttura visibile nella Fig. 9.

Come in precedenza, il bit 7 è



Mostra Mercato
Elettronica.net

3ª EDIZIONE

RAVENNA

FIERA DELL'ELETTRONICA
INFORMATICA E RADIANTISMO

ORARIO 9,00/18,30 - PARCHEGGIO GRATUITO

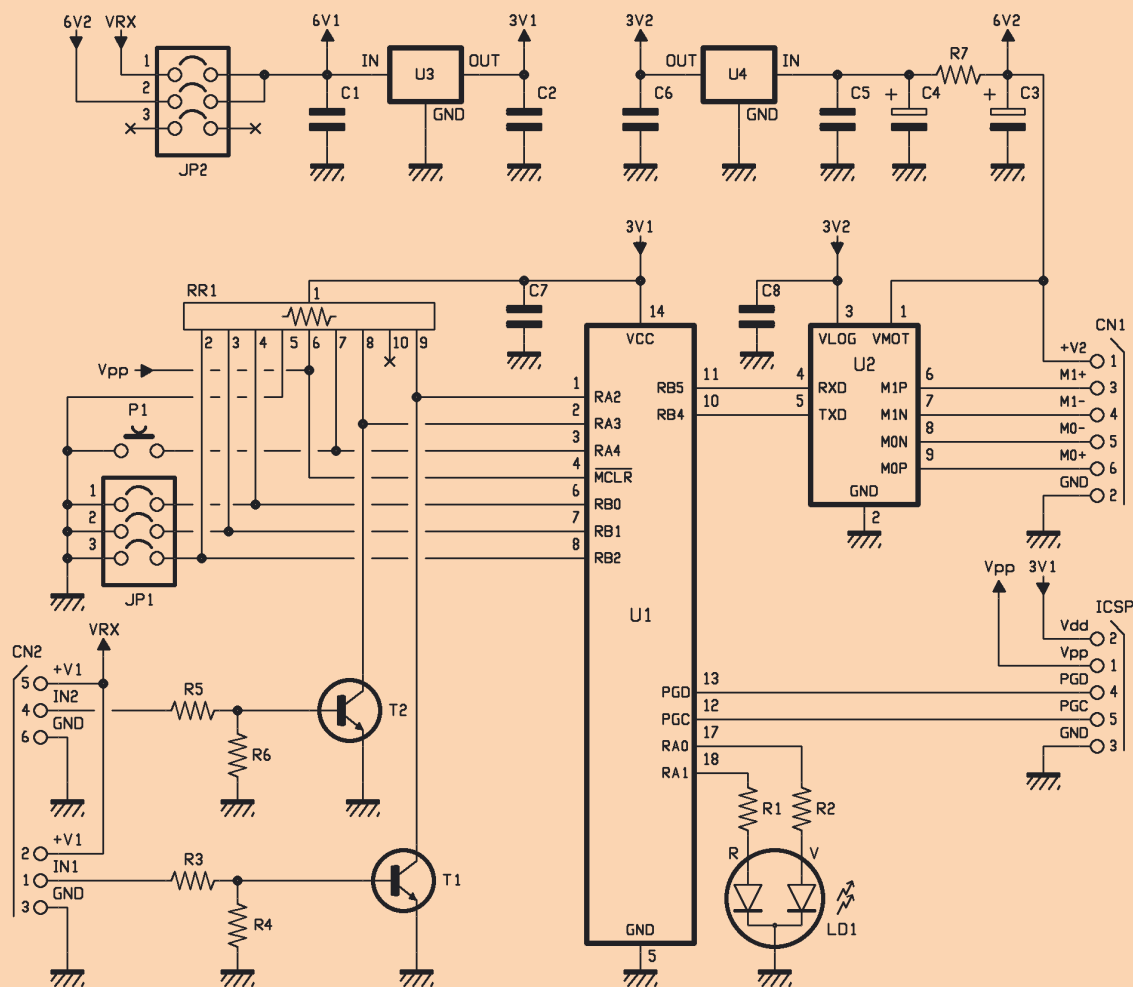
Pala De Andrè
Viale Europa 1

26-27 SETTEMBRE 2009

Tel. e Fax 0545 27548 - Cell. 348 9030832 - www.mondoelettronica.net

VALE COME RIDOTTO

ANCHE IN FOTOCOPIA



Rimane il quarto byte, che può valere da 0 a 127 (0x00÷0x7F) ed imposta la velocità di rotazione. Ancora una volta qualche esempio può essere di aiuto:

- 0x80-0x00-0x07-0x7F fa ruotare il motore all'indirizzo 3 in avanti al massimo della velocità;
- 0x80-0x00-0x09-0x3F fa ruotare il motore all'indirizzo 4 in avanti a metà regime circa;
- 0x80-0x00-0x1A-0x1F fa ruotare il motore all'indirizzo 13 all'indietro a circa 1/4 di regime.

Giunti qui, mancano solamente alcune note sullo stadio che pilota i motori: entrambe le uscite realizzano un ponte di alimentazione ad H, schematizzato nella **Fig. 10**. Quando gli switch (dei mosfet) da 1 a 4 sono tutti aperti, non circola corrente nel motore che, pertanto, rimane fermo. Chiudendo gli switch 1 e 4, il motore ruota in un senso mentre chiudendo gli switch 2 e 3, il motore ruota in senso opposto. Qualche che sia la direzione di rotazione, quando gli switch rimangono costantemente chiusi, il motore ruota alla massima velocità. Per

variare la velocità di rotazione è sufficiente comandare entrambi gli switch (1-4 oppure 2-3) con un segnale ad onda quadra di frequenza costante (circa 600/750 Hz nel caso dell'MDSMC) e duty cycle (rapporto tra la durata dell'1 logico e il periodo) variabile secondo la velocità di rotazione che si vuole ottenere. Un duty cycle del 50% fa ruotare il motore a circa metà regime, mentre un D.C. del 25 % limita la rotazione a 1/4 di regime, eccetera. Questo sistema per pilotare i carichi è noto come PWM, acronimo delle parole "Pulse Width Modulation", ossia modulazione della larghezza

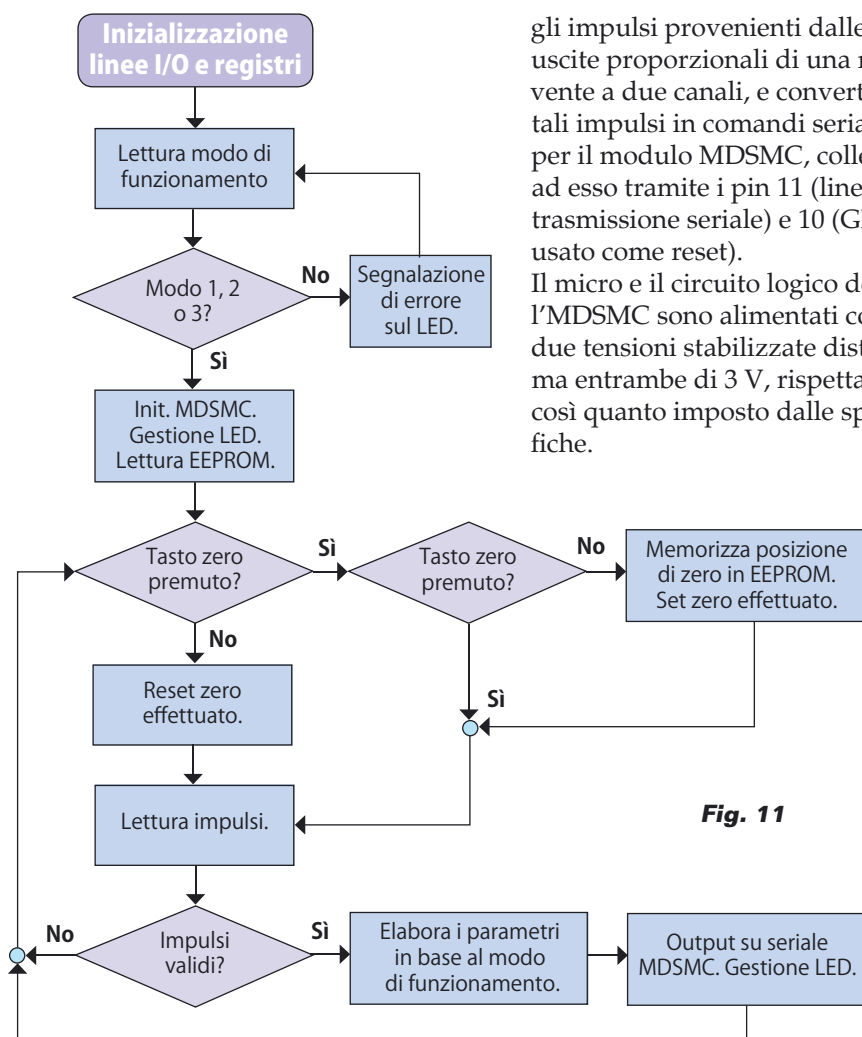


Fig. 11

gli impulsi provenienti dalle uscite proporzionali di una ricevente a due canali, e convertire tali impulsi in comandi seriali per il modulo MDSMC, collegato ad esso tramite i pin 11 (linea di trasmissione seriale) e 10 (GPIO usato come reset). Il micro e il circuito logico dell'MDSMC sono alimentati con due tensioni stabilizzate distinte, ma entrambe di 3 V, rispettando così quanto imposto dalle specifiche.

cuito talmente versatile da poter essere utilizzato con tutti i tipi di riceventi radio proporzionali, abbiamo inserito sia un adattatore di livello degli impulsi di controllo sia un doppio stadio di alimentazione. L'adattatore di livello è realizzato con i due transistor T1 e T2, pilotati tramite i partitori resistivi R3-R4 e R5-R6 (tutte resistenze da 10 kohm) che permettono di applicare all'ingresso del microcontrollore impulsi di ampiezza 2÷12 Vcc circa.

Tali impulsi vengono prelevati dai collettori di T1 e T2 invertiti e stabilizzati al livello di 3 V (necessari al PIC per operare correttamente) per essere poi applicati ai pin 1 e 2 del micro.

Il doppio circuito di alimentazione permette di scegliere il modo in cui si desidera alimentare il dispositivo; sono previste le seguenti possibilità.

1. Sezione logica del circuito alimentata dalla radioricevente e sezione di potenza alimentata da una batteria aggiuntiva.
2. Sezione logica e sezione di potenza alimentate entrambe direttamente dalla ricevente, tensione di alimentazione prelevabile dai morsetti 1 e 2.
3. Sezione logica e sezione di potenza alimentate da una batteria specifica mentre la ricevente fornisce solo gli impulsi di controllo.

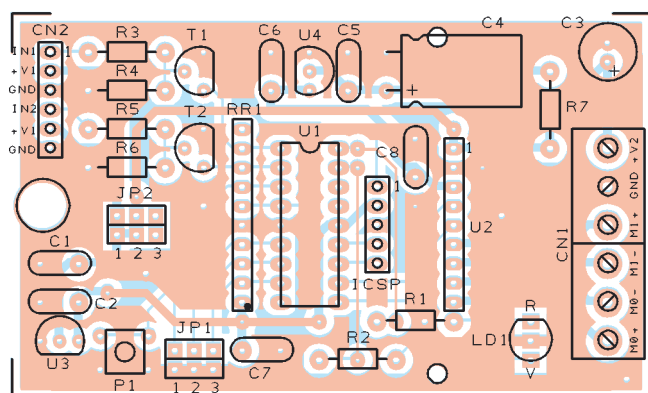
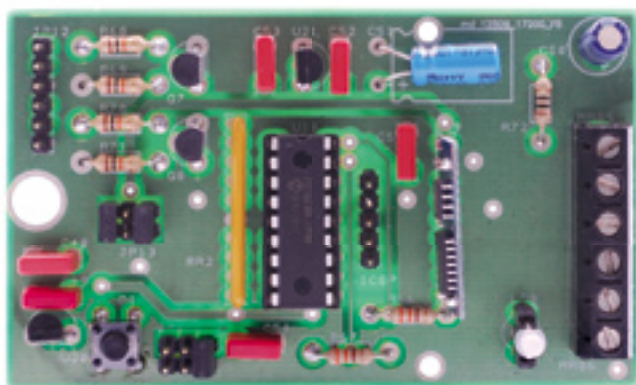
Le tre modalità di alimentazione vengono definite tramite il jumper JP2, che deve essere configurato come esposto nel riquadro dedicato alle impostazioni. Il microcontrollore e la sezione logica del driver MDSMC vengono alimentati tramite due linee a 3 V distinte, ottenute utilizzando due regolatori low drop-out mai utilizzati prima: si tratta precisamente degli MCP1702 di casa Microchip, nella versione a 3 V,

dell'impulso. Qualora desideriate saperne di più, potete rivedere l'articolo relativo all'alimentatore PWM per trenini, presentato nel fascicolo 127.

SCHEMA ELETTRICO

Date le sue caratteristiche, l'MDSMC si presta benissimo per le applicazioni modellistiche dove è necessario regolare in modo proporzionale la velocità di rotazione di piccoli motori DC, oltre che comandare dei servocontrolli. Facendo riferimento allo schema elettrico, il punto centrale del circuito è rappresentato dal microcontrollore, un PIC16F88 programmato per poter leggere

Relativamente agli impulsi di controllo provenienti dalla ricevente del radiocomando ed applicati sul connettore JP2 (sul quale vengono anche prelevate la tensione di alimentazione della ricevente e la massa), questi si ripetono 50 volte al secondo ed hanno una durata compresa tra 1 ms (servo tutto a sinistra o motore indietro a massimo regime) e 2 ms (servo tutto a destra o motore avanti a massimo regime). 1,5 ms definiscono la posizione di servo al centro o di motore fermo. La loro ampiezza, infine, dipende dalla tensione di alimentazione della ricevente. Per poter rendere il nostro cir-



Elenco Componenti:

R1, R2: 330 ohm
 R3÷R6: 10 kohm
 R7: 10 ohm
 RR1: Rete resistiva 9x10 kohm + C
 C1÷C3, C5÷C8: 100 nF 63 VL
 poliestere
 C4: 100 µF 16 VL elettrolitico
 LD1: Led 5 mm rosso/verde
 P1: Microswitch
 T1, T2: BC547
 U1: PIC16F88-I/P (MF838)
 U2: Driver motori (MDSMC)

U3, U4: MCP1702-3

Varie:

- Morsettiera 3 poli passo 5,08 mm (2 pz.)
- Zoccolo 9+9
- Strip maschio 3 poli (4 pz.)
- Strip maschio 5 poli
- Strip maschio 6 poli
- Jumper (3 pz.)
- Contenitore plastico SC-700
- Circuito stampato

che sono tra i pochi a garantire una corrente di funzionamento nominale di 250 mA in un piccolo case TO-92. Come potete notare dallo schema, il modo in cui essi sono impiegati è tipo il classico 7805. Notate il filtro CRC (C3-R7-C4) posto all'ingresso dello stabilizzatore della sezione di potenza, che limita i disturbi generati dai picchi di assorbimento causati dallo spunto dei motori durante l'avvio. Alimentare tanto

il microcontrollore quanto la logica del Pololu con la tensione di 3 V proveniente dai due low dropout differenti consente di evitare anomalie di funzionamento sui circuiti logici fino ad una tensione di batteria nell'intorno dei 3,5 V (pile completamente scariche). Il jumper JP1 agisce direttamente su tre linee di input al micro, e seleziona uno dei seguenti modi di funzionamento:

1. **modo 1:** monocanale propor-

zionale;

2. **modo 2:** bicanale proporzionale;

3. **modo 3:** bicanale differenziale proporzionale.

Nel primo caso, il dispositivo gestisce un solo motore e gli impulsi di controllo vengono applicati sul solo ingresso 0 (pin 1 di JP2); nel secondo caso, i motori vengono comandati individualmente dai due segnali di controllo applicati su JP2 (pin 1 e 4): il motore 0 viene comandato dal canale 0 (pin 1) e il motore 1 dal canale 1 (pin 4). Nell'ultimo caso, invece, la cosa si complica leggermente, dato che un canale gestisce il movimento avanti/indietro e l'altro quello destra/sinistra. Naturalmente, trattandosi di un controllo per motori, questo modo di funzionamento genera il cambio di direzione di un eventuale dispositivo in movimento facendo ruotare in modo differente il motore di destra rispetto a quello di sinistra, ma questo sarà spiegato con maggiori dettagli nel capitolo dedicato all'utilizzo in pratica.

Il pulsante P1, collegato alla porta RA4, serve per impostare la posizione di zero (neutro) dei joystick della trasmettente, posizione che viene salvata nella memoria non volatile del micro. I motori vanno collegati ai morsetti 3-4 e 5-6 della morsettiera siglata CN1. Un'eventuale batteria esterna va, invece, connessa ai morsetti 1 e 2 della morsettiera, morsetti dai quali è anche possibile prelevare alimentazione quando il circuito viene alimentato completamente dalla ricevente. Il led bicolore LD1, gestito dalle linee RA1 e RA0, scandisce le fasi operative del circuito, emettendo lampeggi di durata e colore variabile, descritti nella **Tabella 2**. Completa il tutto

il connettore ICSP, utilizzato per programmare il PIC direttamente sul circuito stampato.

IL PROGRAMMA DI GESTIONE

Il firmware che controlla tutto il sistema (diagramma di flusso nella Fig. 11) non è particolarmente complesso: è, infatti, basato su una serie di loop che si ripetono o meno al verificarsi di condizioni particolari, ma vediamo le varie fasi. Il programma inizializza le linee di I/O del micro nonché i valori di default per la posizione di neutro (verificati sperimentalmente) in EEPROM quindi va a leggere il jumper dei modi di funzionamento. In caso di errore (nessun jumper collegato) viene generato un segnale con il led bicolore. In caso contrario, invece (è stato selezionato uno dei tre modi) vengono inizializzati il driver Pololu (il led segnala anche questa fase) ed i registri utilizzati dalle numerose routine matematiche, calibrate sui valori di zero predefiniti, prelevati dalla EEPROM. A questo punto inizia il loop vero e proprio che controlla sempre se il tasto di zero è premuto. Qualora lo sia, verifica che non lo fosse anche in precedenza e, nel caso, legge gli impulsi in ingresso e li associa al valore di zero (motori spenti) quindi li salva in EEPROM e pone a 1 logico il flag di "Zero Effettuato". In questo modo anche se il tasto rimane premuto, la memorizzazione in EEPROM viene effettuata una sola volta. Quando il pulsante viene rilasciato, il programma azzerava il flag. Sia a pulsante premuto che rilasciato, il loop principale legge gli impulsi di controllo in ingresso e quindi li elabora, ma solo se hanno una durata compresa tra 0,5 e 2,5 millisecondi. Viene quindi eseguito un output sulla seriale del modulo MDSMC, ac-

Tabella 1 - Segnalazioni luminose.

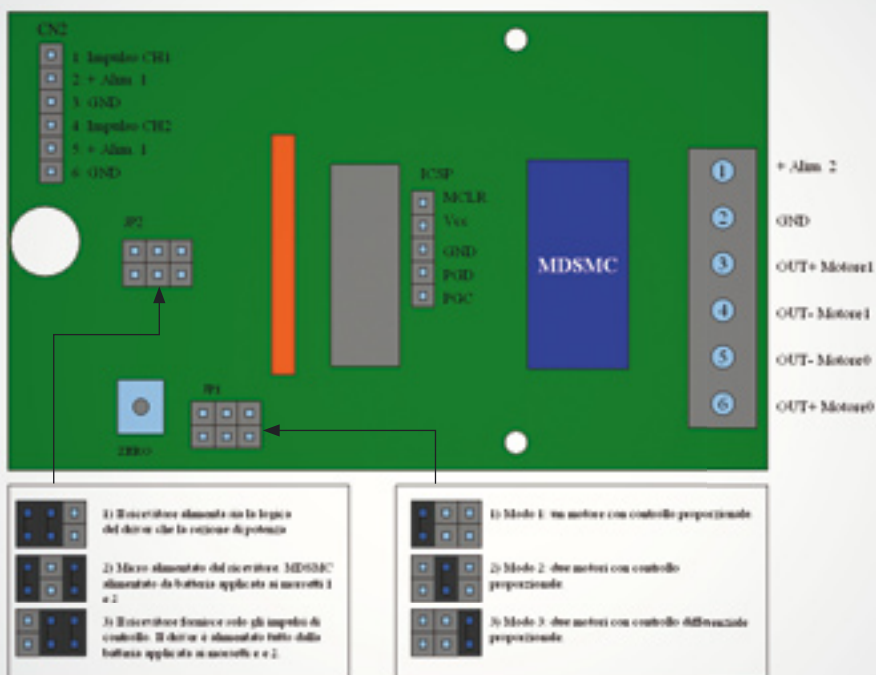
Segnalazione	Significato
Giallo ON 2 s., OFF 2s	Firmware partito correttamente
Giallo 1/2/3 lampeggi	Modo 1, 2 o 3.
Giallo ON 1 s., OFF 1 s.	Reset al Pololu.
Verde 1 lampeggio	Invio stringa configurazione Pololu
Lampeggio Rosso/Verde veloce	Impulsi in ingresso corretti.
Lampeggio Rosso/Verde lento	Impulsi in ingresso non corretti.
Lampeggio Rosso/Verde uno lento uno veloce	Uno dei due ingressi non ha gli impulsi corretti.
Lampeggio Rosso/Verde seguito da 1 secondo di OFF	Nessun modo selezionato. Controllare Jumper.

compagnato da una segnalazione di attività sui led.

È prevista una funzione (non riportata nel flow-chart) chiamata "Learning Mode", che viene abilitata direttamente da codice sorgente ponendo la costante `MODE_LEARN` a 1 (0 per disattivarla) e, se abilitata, ogni volta che il driver MDSMC riceve dei comandi seriali, gli stessi comandi vengono inviati alla linea di I/O RB7, che corrisponde alla linea dati ICSP. In tale modo è possibile utilizzare l'emulatore di terminale fornito con il program-

matore pickit2 (che usa come linea di ricezione proprio la porta RB7) per vedere in modo più "user friendly" (ossia più comprensibile) i comandi e i parametri ricevuti dall'MDSMC. Abbiamo voluto dare la possibilità di attivare il Learning Mode solo via firmware per due ragioni: rallenta di parecchio l'esecuzione del programma (a causa delle numerose e continue scritture sulla seconda seriale); serve solo quando ci si avvicina per la prima volta il protocollo seriale Pololu.

Alimentare il circuito



Riepilogo delle impostazioni dei due jumper (JP1 riguarda i modi di funzionamento e JP2 l'alimentazione) e collegamenti del nostro circuito.

REALIZZAZIONE PRATICA

Per realizzare il circuito per prima cosa dovete scaricare dal nostro sito web (www.elettronica-in.it) il master a doppia faccia in scala 1:1, che utilizzerete per ricavare lo stampato mediante fotoincisione. Praticate i fori necessari ed iniziate il montaggio saldando le vie (i ponticelli che collegano due piazzole poste sui due lati dello stampato). Saldate quindi tutti i componenti a profilo basso (resistenze, rete resistiva, zoccolo 9+9 pin DIP e condensatori in poliestere) e poi i condensatori elettrolitici (C4 va montato sdraiato, come visibile nel piano di montaggio). Montate infine tutti i jumper coi relativi cavalieri, la morsettiera a 6 morsetti, il pulsante, il led bicolore e il pin strip a cinque pin che funge da ICSP. Il modulo MDSMC è l'ultimo da inserire e va posizionato come visibile sul piano di montaggio. Per programmare il PIC attraverso il connettore ICSP, dovete impostare il jumper di selezione JP2 in modo da alimentare tutto il circuito con una sola batteria esterna: come vi avevamo già anticipato, potete seguire le indicazioni riportate nell'appo-

sito riquadro qui accanto, dedicato alla pin-out ed ai vari collegamenti. Fornite ora una tensione di valore compreso tra 3,5 e 9 V (i canonici 5 V andranno benissimo) sui morsetti 1 e 2 della morsettiera, rispettando la polarità, pena la distruzione del modulo Pololu. Il circuito stampato è stato progettato per l'inserimento in un contenitore SC-700 Futura Elettronica (www.futurashop.it).

UTILIZZO IN PRATICA

Giunti qui, avete a disposizione il driver bicanale programmato e pronto per l'uso; non vi rimane, quindi, che preparare il banco di prova (sempre conveniente provare al banco prima che su strada...) e verificare che tutte le varie opzioni di funzionamento siano a punto. Per le prove vi occorrono una trasmettente ed una ricevente proporzionali, eventualmente un paio di servocontrolli (con i quali verificare che i canali proporzionali funzionino) e due motorini elettrici di piccole dimensioni (ad

esempio, 6 V 500 mA). Essi devono poter funzionare alla tensione con cui alimentate la sezione di potenza dell'MDSMC: se questi è alimentato a 6 V, anche i motori devono poter funzionare con tale tensione.

Nel nostro schema applicativo abbiamo ipotizzato l'utilizzo con un sistema radio a quattro canali, ma nulla vieta di sceglierne uno a due o uno a otto. Se lo preferite, potete realizzare il piccolo generatore di impulsi che abbiamo realizzato noi: simula egregiamente i due canali proporzionali di una ricevente, richiede pochissimi componenti (per cui può essere realizzato molto facilmente su una scheda millefori) e rende superfluo l'utilizzo del sistema radio completo durante i test al banco di prova.

Quale che sia la scelta, il driver

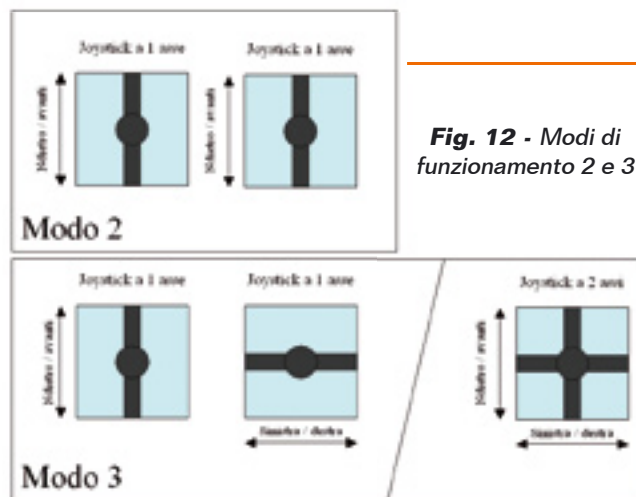


Fig. 12 - Modi di funzionamento 2 e 3



Ad Amburgo si trova *Miniatur Wunderland*, il plastico ferroviario più grande al mondo, ricavato, in un palazzo abbandonato sulla riva del fiume Elba, dai fratelli

Modellismo su larga scala

Braun, il cui lavoro certosino è iniziato nel 2000 e terminerà nel 2014. Su una superficie che per ora è limitata (si fa per dire...) a 1.150 metri quadrati (diventeranno 1.800) si trovano 12 km di binari e 14.100 vagoni. Nel plastico non solo treni, ma anche automobili, camion e navi. Sull'immenso territorio in miniatura sono fedelmente

riprodotte alcune zone di Stati Uniti, Svizzera, Scandinavia, Germania e le Alpi austriache. Presto verranno aggiunti scenari dedicati a Francia, Italia e Regno Unito. Il tutto ha richiesto 4.000 kg d'acciaio e 700 kg d'erba artificiale, oltre a 250.000 luci, sensori e microprocessori per simulare il giorno e la notte.

```

*****
* VARIABILI, COSTANTI E FLAG
*****
SPEEDO      VAR WORD      ' CONTIENE LA VELOCITÀ ASSOLUTA CANALE 0.
SPEED1      VAR WORD      ' CONTIENE LA VELOCITÀ ASSOLUTA CANALE 1.
' NOTA: LE ROUTINE MATEMATICHE PER IL CALCOLO DELLA VELOCITÀ NECESSITANO DI SPEEDO E 1 COME WORD
' MA RESTITUISCONO SEMPRE UN VALORE DI DIMENSIONE BYTE, CHE PUÒ ESSERE CONTENUTO IN SPEED.
DIRO        VAR BIT       ' FLAG DIREZIONE CANALE 0.
DIR1        VAR BIT       ' FLAG DIREZIONE CANALE 1.
START_BYTE  CON $80       ' START BYTE PER COMANDO ALL'MSMC.
CHANGE_CONFIG CON $2      ' COMANDO CAMBIO CONFIGURAZIONE.
DEVICE_NMB  CON $0        ' IDENTIFICATIVO DEL DRIVER IN MODO REGOLAZIONE VELOCITÀ.
MOT_ADD     CON 2         ' INDIRIZZO BASE MOTORI (QUI È FISSO A 2).
MOT_NMB     VAR BYTE      ' VARIABILE SELEZIONE INDIRIZZO MOTORE.
MOT_NMB = MOT_ADD      ' INIZIALIZZAZIONE VARIABILE SELEZIONE MOTORE..
SYMBOL      ONE_MOTOR=MOT_NMB.6 ' FLAG NUMERO MOTORI: 1 = 1 MOTORE. 0 = DUE MOTORI.
                                ' E' LOCALIZZATO SUL BIT 6 DELLA VARIABILE SELEZIONE MOTORE.
MOT_DIR     VAR BYTE      ' CONTIENE NUMERO MOTORE E DIREZIONE
SYMBOL      DIR=MOT_DIR.0 ' FLAG DIREZIONE (0=INDIETRO, 1=AVANTI).
SPEED       VAR BYTE      ' CONTIENE IL VALORE DELLA VELOCITÀ (0=127).
*****
* INIZIALIZZAZIONE DELL'INDIRIZZO.
*****
HSEROUT [START_BYTE,CHANGE_CONFIG,MOT_NMB] ' INVIA COMANDO AL DRIVER.
' HSEROUT [$80,$02,$02] ' COME SOPRA, CON VALORI NUMERICI.
*****
* COMANDO AL MOTORE 0.
*****
OUT_MOTOR0: ' INIZIO ROUTINE.
MOT_DIR = MOT_ADD ' IMPOSTA MOTORE A INDIRIZZO 2.
MOT_DIR = MOT_DIR << 1 ' MOLTIPLICA PER 2 (SHIFT DI 1 SX).
DIR = DIR0 ' IMPOSTA FLAG DIREZIONE (IL FLAG DIRO È STATO
            ' IMPOSTATO IN UN ALTRO PUNTO DEL PROGRAMMA).
SPEED = SPEEDO ' IMPOSTA LA VELOCITÀ (SPEEDO CALCOLATA
                ' IN UN ALTRO PUNTO DEL PROGRAMMA).
HSEROUT [START_BYTE] ' INVIA START BYTE.
HSEROUT [DEVICE_NMB] ' INVIA IDENTIFICATORE DI DISPOSITIVO.
HSEROUT [MOT_DIR] ' INVIA INDIRIZZO MOTORE E FLAG DIREZIONE.
HSEROUT [SPEED] ' INVIA VELOCITÀ.
IF MODE_LEARN = 1 THEN ' MODO APPRENDIMENTO ATTIVO
CALL OUT_MONITOR0 ' SCRITTURA SU PORTA DI MONITOR.
ENDIF
PAUSE 10 ' PAUSA DI 10 MSEC.
RETURN ' ESCE.
*****
* COMANDO AL MOTORE 1.
*****
OUT_MOTOR1: ' INIZIO ROUTINE.
MOT_DIR = MOT_ADD + 1 ' IMPOSTA MOTORE A INDIRIZZO 3.
MOT_DIR = MOT_DIR << 1 ' MOLTIPLICA PER 2 (SHIFT DI 1 SX).
DIR = DIR1 ' IMPOSTA FLAG DIREZIONE (IL FLAG DIR1 È STATO
            ' IMPOSTATO IN UN ALTRO PUNTO DEL PROGRAMMA).
SPEED = SPEED1 ' IMPOSTA LA VELOCITÀ (SPEED1 CALCOLATA
                ' IN UN ALTRO PUNTO DEL PROGRAMMA).
HSEROUT [START_BYTE] ' INVIA START BYTE.
HSEROUT [DEVICE_NMB] ' INVIA IDENTIFICATORE DI DISPOSITIVO.
HSEROUT [MOT_DIR] ' INVIA INDIRIZZO MOTORE E FLAG DIREZIONE.
HSEROUT [SPEED] ' INVIA VELOCITÀ.
IF MODE_LEARN = 1 THEN ' MODO APPRENDIMENTO ATTIVO
CALL OUT_MONITOR1 ' SCRITTURA SU PORTA DI MONITOR.
ENDIF
PAUSE 10 ' PAUSA DI 10 MSEC.
RETURN ' ESCE.

```

Commento al LISTATO

Riportiamo la porzione di codice utilizzata per inizializzare l'indirizzo base del driver MDSMC e per inviare i comandi relativi al controllo della velocità. Come linea guida, abbiamo ritenuto conveniente definire degli mnemonici costanti per tutti i comandi base (start byte, cambio configurazione, selezione dispositivo) e delle variabili di servizio utilizzate per il trasferimento dei parametri associati ai comandi. Per esempio, la variabile SPEED viene utilizzata dal comando HSEROUT come vettore per trasferire al motore 0 la velocità SPEEDO e al motore 1 la velocità SPEED1.

deve essere collegato tramite due cavetti specifici che potreste comprare a costi piuttosto elevati, o realizzare voi stessi con dei conduttori da 0,25 mm² e dei pin strip femmina a passo 2,54 mm.

Scegliere l'alimentazione.

Per alimentare il modulo non esistono regole precise, bensì situazioni in cui è più conveniente scegliere in un modo piuttosto che in un altro; noi abbiamo pensato a tre possibili casistiche di utilizzo, che sono anche le più tipiche:

1. modello elettrico di dimensioni molto compatte;
2. modello elettrico di dimensioni meno restrittive;
3. modello in cui, per cause di natura meccanica, ricevente e driver motori possono essere distanti tra loro e, quindi, prendere l'alimentazione dalla ricevente potrebbe causare grosse cadute di tensione sul cavo di collegamento.

Nel primo caso, difficilmente si riesce a collocare sul modello una seconda batteria per alimentare

la sezione di potenza ai motori e pertanto il driver deve prelevare l'energia richiesta direttamente dalla ricevente, esattamente come fanno i servocontrolli: si sceglie, quindi, l'opzione di alimentazione 1 (vedere l'apposito riquadro). È lecito credere che, per via delle dimensioni compatte, i motori non consumino molto, quindi non ci si aspettano grossi problemi per quanto riguarda l'autonomia. Nel secondo caso, invece, le dimensioni maggiori determinano masse e pesi maggiori, quindi diventa quasi obbli-

Simulatore di due canali proporzionali

gatorio l'utilizzo di una seconda fonte di alimentazione per la sola sezione motori (opzione 2). Nel terzo caso, invece, si è in presenza di un modello stretto e lungo, come un sommergibile: in questo frangente, il bilanciamento dei carichi e delle masse è un fattore chiave per la corretta navigazione subacquea e pertanto le varie parti devono essere equamente distribuite. Le batterie possono essere messe centralmente oppure una a prua e una a poppa; in ogni caso, qui può essere conveniente alimentare il driver motori con una batteria propria e prelevare dalla ricevente i soli impulsi di controllo (opzione 3).

Il modo di funzionamento

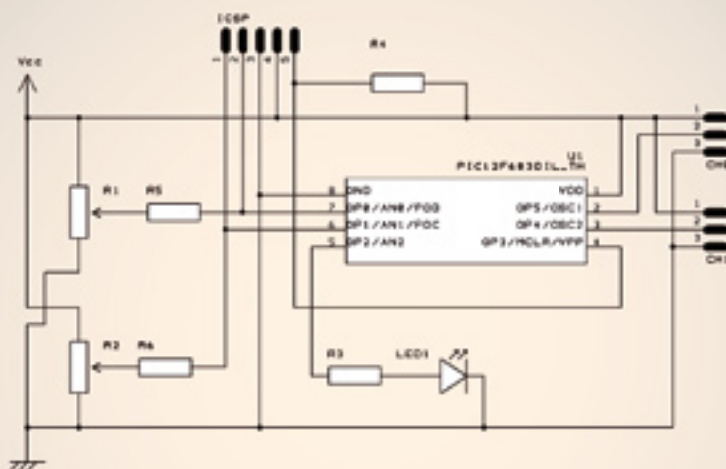
Il modo 1 (un canale proporzionale) va scelto quando si deve pilotare un solo motore secondo lo schema già visto nella Fig. 3, oppure due motori uguali allo stesso regime e verso di rotazione, secondo lo schema di Fig. 2; gli impulsi di controllo vengono sempre applicati all'ingresso impulsi 0, pin 1 del CN2.

I modi 2 e 3 si scelgono sempre per co-

Per collaudare il driver a due canali potete utilizzare il sistema completo, costituito da trasmettitore e ricevente, oppure il piccolo circuito riportato qui, che è talmente semplice da poter essere facilmente realizzato su uno spezzone di basetta millefori. Il circuito va collegato direttamente al connettore JP2 del driver, al quale fornisce gli impulsi di controllo e dal quale preleva la tensione di alimentazione necessaria (massimo 5,5 Vcc). Ruotando i due trimmer è possibile generare impulsi alla frequenza di 50

Hz e di durata compresa tra 0,9 e 2,1 millisecondi circa, parametri questi che si adattano non solo al nostro driver bensì a tutti i driver e servocontrolli reperibili in commercio. Utilizzando il circuito di test è necessario impostare il modo di alimentazione 1 (il ricevitore alimenta sia la sezione logica che quella di potenza) e applicare 5 V stabilizzati ai morsetti 1 e 2 della morsettiera.

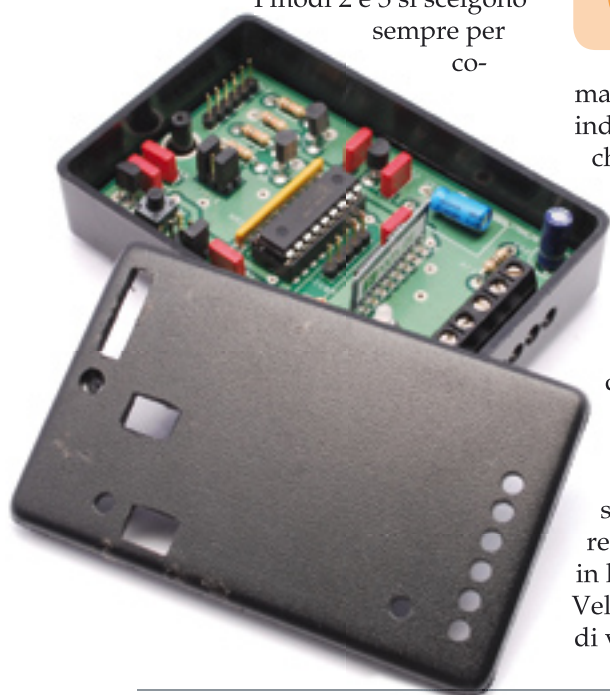
Il listato del programma è ben commentato pertanto non ci dilungheremo in ulteriori spiegazioni.



Elenco Componenti:

R1, R2: 10 kohm trimmer
R3, R5, R6: 1 kohm 1/4W
R4: 10 kohm 1/4W
LED1: led rosso 5 mm

U1: PIC16F683 DIL
ICSP: pin strip maschio 5 poli
PL2: pin strip maschio 3 poli
PL3: pin strip maschio 3 poli



mandare due motori in modo indipendente secondo una logica che dipende da come volete utilizzare tanto il radiocomando quanto il modello, che per ragioni di semplicità di spiegazione ipotizziamo essere un carro armato. Come ben noto, per variare la direzione di spostamento (per "sterzare") questo veicolo varia il regime di rotazione dei cingoli: se essi ruotano alla stessa velocità nella stessa direzione, il carro armato procede in linea retta avanti o indietro. Velocità di rotazione uguali ma di verso opposto determinano la

rotazione sul proprio asse ferma sul posto. La rotazione dei cingoli nella stessa direzione ma con velocità differenti, infine, determina la curvatura dalla parte del cingolo che gira più lentamente: ciò significa che se vogliamo, ad esempio, che il nostro carro armato svolti a sinistra durante la marcia in avanti, dobbiamo ridurre la velocità di rotazione del cingolo sinistro oppure aumentare quella del cingolo destro. Il raggio di curvatura è tanto più stretto quanto più alta è la differenza di velocità tra i cingoli. Compreso come sterza il carro armato, vediamo come scegliere

```

* ESTRATTI DA FILE C:\PBP\INC\M12F683.INC
@ DEVICE INTOSC10 * OSCILLATORE INTERNO.
@ DEVICE WDT_OFF * WATCH DOG DISABILITATO.
@ DEVICE PWRT_OFF * POWER UP TIMER DISABILITATO.
@ DEVICE BOD_OFF * BROWN OUT RESET DISABILITATO.
@ DEVICE MCLR_ON * MASTER CLEAR ABILITATO.
*****
* CONFIGURAZIONE DEI REGISTRI
*****
OSCON = %01110000 * IMPOSTA CLOCK INTERNO A 8 MHZ.
OPTION_REG = %00000000 * IMPOSTAZIONI VARIE
ADCON0 = %10000000 * DISABILITA ADC.
ANSEL = %00000011 * GPO E GPI ANALOGICI.
CCPICON = %00000000 * CCP1 DISABILITATO.
VRCON = %00000000 * IMPOSTAZIONI VARIE.
CMCON = %00000111 * IMPOSTAZIONI VARIE.
DEFINE OSC 8 * SPECIFICA CLOCK A 8 MHZ.
DEFINE ADC_BITS 10 * RISULTATO ADC A 8 BIT.
DEFINE ADC_CLOCK 0 * CLOCK ADC (0=CK/2, 1=CK/8, 2=CK/32, 3=RC).
DEFINE ADC_SAMPLEUS 50 * TEMPO DI CAMPIONAMENTO IN MICROSECONDI.
INPUT GPIO_0 * INGRESSO.
INPUT GPIO_1 * INGRESSO.
OUTPUT GPIO_2 * USCITA.
OUTPUT GPIO_3 * USCITA.
OUTPUT GPIO_4 * USCITA.
OUTPUT GPIO_5 * USCITA.
SYMBOL POT_CHO=GPIO_0 * INGRESSO TENSIONE CANALE 1.
SYMBOL POT_CH1=GPIO_1 * INGRESSO TENSIONE CANALE 2.
SYMBOL ACT=GPIO_2 * LED DI ATTIVITA'.
SYMBOL OUT0=GPIO_5 * USCITA IMPULSI CANALE 1.
SYMBOL OUT1=GPIO_4 * USCITA IMPULSI CANALE 1.
WORD * CONTIENE LETTURA POTENZIOMETRO CANALE 1.
POT_VAL0 VAR WORD * CONTIENE LETTURA POTENZIOMETRO CANALE 2.
POT_VAL1 VAR WORD * ADC CANALE 0.
CHO CON 0 * ADC CANALE 1.
CHI CON 1 * VALORE MASSIMO DI CICLO DI PULSAZIONE
PULSE_TIME VAR WORD * OFFSET GENERAZIONE IMPULSO 989 USEC
OFFSET CON 989 * DURATA DEL PERIODO DI IMPULSO (20000 USEC, PARI A 50 HZ).
PERIOD CON 20000 * VARIABILE DI SERVIZIO.
TMP VAR BYTE * AZZERA VARIABILE DI SERVIZIO.
TMP = 0
*****
HIGH_ACT * ACCENDE LED.
LOW_OUT0 * AZZERA USCITA 0.
LOW_OUT1 * AZZERA USCITA 1.
MAIN: * LOOP PRINCIPALE.
CALL POT_READ * LEGGE I DUE POTENZIOMETRI.
CALL PULSE_OUT * GENERA GLI IMPULSI IN USCITA.
CALL PAUSE_CALC * GENERA PAUSA CALCOLATA.
GOTO MAIN * RIPEETE LOOP PRINCIPALE
*****
POT_READ: * LETTURA DEI POTENZIOMETRI.
ADCON0 = 0 * LEGGE PRIMO POTENZIOMETRO.
ADCON1 = 0 * LEGGE SECONDO POTENZIOMETRO.
RETURN * FINE LETTURA POTENZIOMETRI.
*****
PULSE_OUT: * GENERAZIONE IMPULSI.
PULSE_TIME = OFFSET + POT_VAL0 * IMPULSO0 = LETTURA ADC1 + OFFSET.
HIGH_OUT0 * ATTIVA USCITA 0.
PAUSEUS PULSE_TIME * PAUSA CALCOLATA.
LOW_OUT0 * DISATTIVA USCITA 0.
PULSE_TIME = OFFSET + POT_VAL1 * IMPULSO1 = LETTURA ADC2 + OFFSET.
HIGH_OUT1 * ATTIVA USCITA 1.
PAUSEUS PULSE_TIME * PAUSA CALCOLATA.
LOW_OUT1 * DISATTIVA USCITA1.
RETURN * FINE GENERAZIONE IMPULSO.
*****
PAUSE_CALC: * PAUSA TRA DUE IMPULSI.
PAUSEUS (PERIOD - (2*OFFSET) - POT_VAL0 - POT_VAL1) * CALCOLATA PER AVERE 50 HZ.
TMP = TMP+1 * CICLO DI CONTEGGIO
IF TMP = 10 THEN * CHE SERVE PER FAR LAMPEGGIARE
TOGGLE ACT * IL LED IN MODO VISIBILE.
TMP = 0
ENDIF
RETURN * FINE GENERAZIONE PAUSA.
*****
END * FINE PROGRAMMA.

```

tra il modo 2 e il modo 3, scelta legata (come già detto) sia alla trasmittente sia a come preferite comandare il vostro modello. Per comandare lo spostamento tramite due joystick con il solo movimento verticale della leva (Fig. 12) è necessario impostare il modo 2: così facendo ogni leva controlla velocità e direzione di un solo cingolo; il motore 0 viene comandato dagli impulsi applicati al pin 1 del CN2, mentre il motore 1 è gestito da quelli applicati al pin 4. Questa scelta di utilizzo potrebbe essere determinata da due cause: gli unici canali proporzionali che avete ancora

liberi sono associati agli assi y dei joystick, oppure preferite gestire il carro armato con le due leve avanti/indietro, come avveniva nei carri veri di tanti anni fa. Se, invece, disponete di due canali proporzionali posti su assi differenti (sia sullo stesso joystick che su joystick differenti), dovete scegliere il modo 3 (differenziale proporzionale), che calcola velocità e verso di rotazione dei motori automaticamente, controllando la posizione della leva avanti/indietro e confrontandola con quella del destra/sinistra. In pratica, la leva dell'asse y (verticale) varia la velocità di rotazione

dei motori da 0 a +/-85% mentre quella dell'asse x (orizzontale) corregge la differenza di velocità di rotazione tra i due motori da 0 a +/- 30% (+/- 15% su entrambi i motori). In pratica, con il joystick al centro i motori sono fermi. Quando la leva dell'asse y è tutta avanti o tutta indietro e quella dell'asse x è al centro, entrambi i motori girano all'85% del regime. Spostando il joystick tutto avanti tutto a destra, il motore sinistro è al 100% del regime e quello destro al 70%. Analogamente, tutto avanti tutto a sinistra pone il motore di sinistra al 70% e quello di destra al 100%.

Notate che quanto appena spiegato è valido se i canali del trasmettitore sono impostati in modalità normale. In modalità invertita (reverse) il ragionamento è analogo ma ribaltato di 180°. Non sarà comunque un problema identificare il giusto verso di spostamento sul telecomando ed eventualmente invertire la polarità di alimentazione ai motori.

Le segnalazioni

Applicata la tensione di alimentazione, il led si illumina di giallo per due secondi e si spegne per altri due secondi, segnalando che il programma è partito correttamente. La segnalazione successiva è quella del modo di funzionamento: nel caso il jumper JP1 sia aperto (nessun modo operativo selezionato) il led lampeggia velocemente alternando il colore rosso al verde, per comunicare un errore. Quando invece, il jumper è inserito su una coppia di pin di JP1 secondo quanto riportato al riquadro delle impostazioni, il led lampeggia di giallo un numero di volte pari al modo di funzionamento impostato e precisamente una volta per il modo 1, due per il modo 2 e tre per il 3. Segue un breve



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono facilmente reperibili in commercio. Il master del circuito stampato ed il firmware necessario per programmare il microcontrollore sono scaricabili gratuitamente dal sito della rivista (www.elettronica.in.it). Il driver per motori della Pololu utilizzato in questo progetto è disponibile nel nostro paese presso Futura Elettronica srl (cod. MDSMC) al prezzo di 36,00 Euro, IVA compresa.

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

lampeggio verde che comunica l'avvenuta programmazione del driver Pololu, il quale risponde facendo pulsare l'uscita negativa 0 (pin 5 della morsettiera) per il modo 1 o l'uscita positiva zero (pin 6 della morsettiera) per i modi 2 e 3. Il numero di pulsazioni è sempre pari all'indirizzo assegnato all'MDSMC, incrementato di un'unità e, nel nostro caso corrisponde sempre a tre lampeggi. A questo punto il led

si illumina brevemente di giallo, indicando che è in corso il reset del driver Pololu, azione questa che rende effettiva la programmazione. Ora il driver inizia a controllare gli impulsi in ingresso. Lampeggi veloci che alternano il rosso al verde indicano che sugli ingressi sono presenti degli impulsi di durata corretta. Lampeggi lenti indicano che nessun canale sta ricevendo impulsi di durata valida. Lampeggi veloci

alternati a lenti indicano che uno dei due canali non sta ricevendo impulsi della durata corretta. Muovendo i due joystick (oppure ruotando i trimmer del simulatore, qualora lo abbiate realizzato), i motori iniziano a girare ad una velocità che dipende dalla posizione degli stessi joystick. È possibile che i motori inizino già a girare lentamente nonostante i joystick/trimmer si trovino nella posizione di neutro: premete il pulsante di zero per memorizzare appunto la posizione di neutro. Il led emette un breve lampeggio ed i motori si fermano. A questo punto il driver è calibrato e pronto per l'utilizzo su di un modello dinamico: vi basterà accendere trasmettente ricevente e driver per poter iniziare a pilotare il vostro modellino, qualsiasi esso sia. ■

FUTURA ELETTRONICA

I nostri rivenditori in Emilia Romagna

Tutti i negozi dell'Emilia Romagna dove puoi acquistare direttamente i prodotti Futura Elettronica.

CITTA'	RIVENDITORE	TEL./FAX
ALBINEA (RE)	MONTELETTIC SNC VIA XIV APRILE, 3/V	0522566337/ 0522567248
BOLOGNA	TOMMESANI SNC VIA CASARINI, 58-E	051550761/ 051550591
CASALECCHIO DI RENO (BO)	XTREM TEK SRL VIA PORRETTANA, 361/2	051573263/ 05118890396
CREVALCORE (BO)	ZSAR NET COMPUTER SRL VIALE AMENDOLA, 290	0515900098/ 0515900881
FERRARA	EDI ELETTRONICA VIA PLATTIS, 12	053264881/ 053262505
FORMIGINE (MO)	F.B.E. COMMERCIALE SRL VIA R. MORO, 35	059570460/ 059564373
IMOLA (BO)	MAZZINI SPAZIO DIGITALE VIA CESAÑA, 12	0542780844/ 0542780843
MODENA	BRICO INFORMATICA SRL VIA CORASSORI, 24	059239685
MODENA	OGEL COMPONENTI ELETTRONICI VIA CESARI, 7	059622733/ 0593364703
MONGHEDORO (BO)	MARCHIONI ELETTRODOMESTICI SAS VIA DEGLI OLIVETI, 23	059552063/ 059554424
PARMA	F&S SYSTEMS SRL VIA GIULITI, 7	0521292516/ 0521981690
PRACENZA	ELETTRONICA MILM VIA RAFFAELLO SANZIO, 14	0523695212
REGGIO EMILIA	R.U.C. ELETTRONICA SNC VIA GIACOMO DA SANODIA, 26/A/B	052272469/ 0522730258
RIMINI	C.G.E. ELETRON SNC VIA SAN GOTTARDO, 6	054159948/ 054159929
RIVERGARO (PC)	ACIDA SRL VIA DELLA FORNACE	0523652330/ 0523651852
SCANDIANO (RE)	F.M. ELETTRONICA SNC VIA MAZZINI, 16/A	0522684134
VIGNOLA (MO)	ELETTRONICA INN VIA R. PISSA, 11/B	059702662

Corso Wireless CPU



Utilizzo dell'M2M Studio di Wavecom

Entriamo nel vivo dell'ambiente operativo M2M Studio di Wavecom e vediamo come usarlo per iniziare a programmare le CPU Wireless.

dell'Ing.
NICOLA FRISO

Nella precedente puntata vi abbiamo fatto conoscere le CPU Wireless di Wavecom/Sierra Wireless elencando le varie famiglie disponibili in commercio distinte per prestazioni e funzionalità, ma anche iniziando a descrivere il sistema operativo Open AT e l'ambiente M2M Studio realizzati per esse. In particolare, ci siamo lasciati dopo aver spiegato come ottenere l'M2M Studio dal sito www.wavecom.com, in che modo inizializzare la Wireless CPU affinché sia pronta

a interagire con esso e come accedere alle opzioni dello stesso M2M Studio. Abbiamo poi iniziato a descrivere l'uso del wizard. Ora è il momento di procedere riprendendo da dove ci siamo fermati, ossia dai *tutorial* del *wizard* accessibile dalla schermata Welcome dell'M2M Studio: una volta visualizzata la finestra di dialogo corrispondente, facciamo clic su *tutorials* e, nella finestra che si apre, clicchiamo sul link che consente di aprire i *cheat sheets* (che possiamo tradurre con "pagine

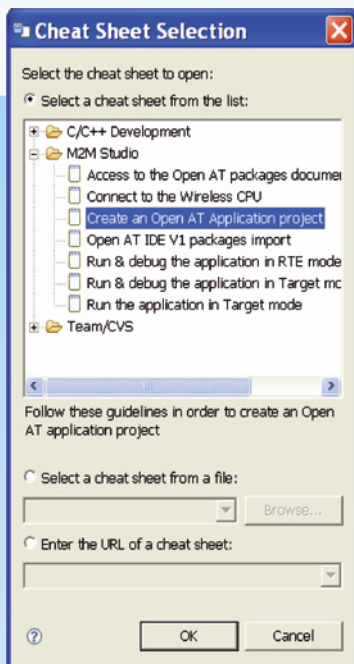
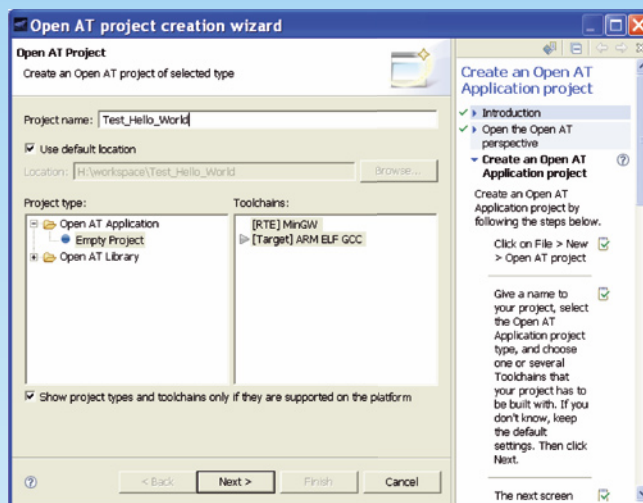


Fig. 1
Cheat sheet
selection.

Fig. 2 - La finestra di dialogo Project name del Wizard.



furbe"...) i quali saranno la guida che ci permetterà, passo-passo, di creare il nostro primo progetto.

Si apre la finestra visibile nella Fig. 1. In essa selezioniamo *Create an Open AT Application project* e, a seguire, clicchiamo su OK. Seguiamo le istruzioni che compaiono sulla cheat page e clicchiamo su *Begin*.

A questo punto, seguendo i link proposti arriviamo alla schermata visibile nella Fig. 2. Diamo un nome al nostro primo progetto: ad esempio *Test_Hello_World*.

Nella finestra seguente (Fig. 3) lasciamo spuntate tutte le opzioni di configurazione. L'RTE (*Real Terminal Emulator*) è l'emulatore su PC sul quale è possibile testare l'applicazione in fase di debug attraverso il tool *miniGW_debug*. In alternativa si può eseguire il debug sul

target (direttamente sulla wireless CPU) attraverso il tool *ARM_ELF_GCC_Debug* o programmare la WCPU attraverso il tool *ARM_ELF_GCC_release*.

Passiamo dunque alla finestra successiva, *Binary project creation*, e selezioniamo l'esempio già pronto *Hello_World* (vedere Fig. 4). Clicchiamo su *Finish*: in pochi secondi il progetto viene creato.

Il *cheat sheet* ci indica che tutti i necessari passi sono stati compiuti (Fig. 5); ora clicchiamo sull'icona per ridurre la finestra di *cheat* (vedere freccia nella Fig. 5) e ci ritroviamo nell'ambiente di sviluppo *M2M Studio* con il codice del progetto caricato e pronto ad essere debuggato (Fig. 6). Sulla colonna di destra troviamo sempre i *cheat sheet* che ci fanno da guida, qualsiasi operazione vogliamo compie-

Fig. 3 - La finestra
Configurations del Wizard.

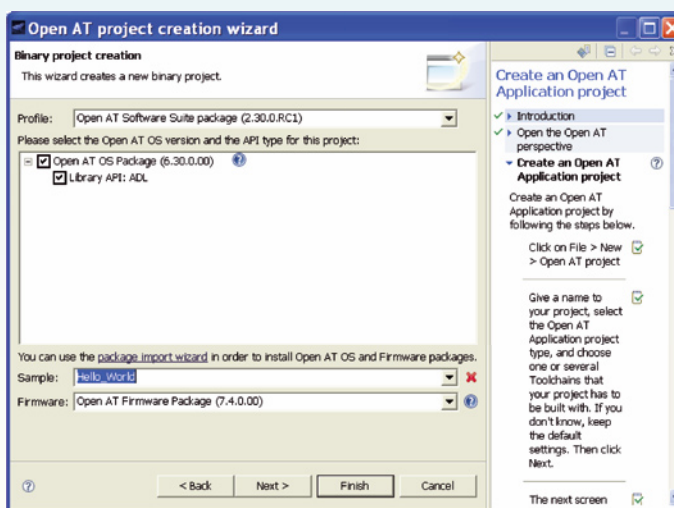
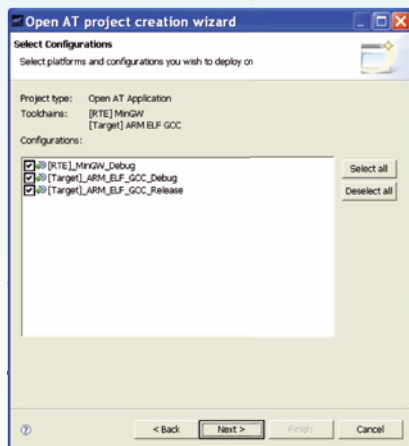


Fig. 4
Wizard:
la finestra
Binary
project
creation.

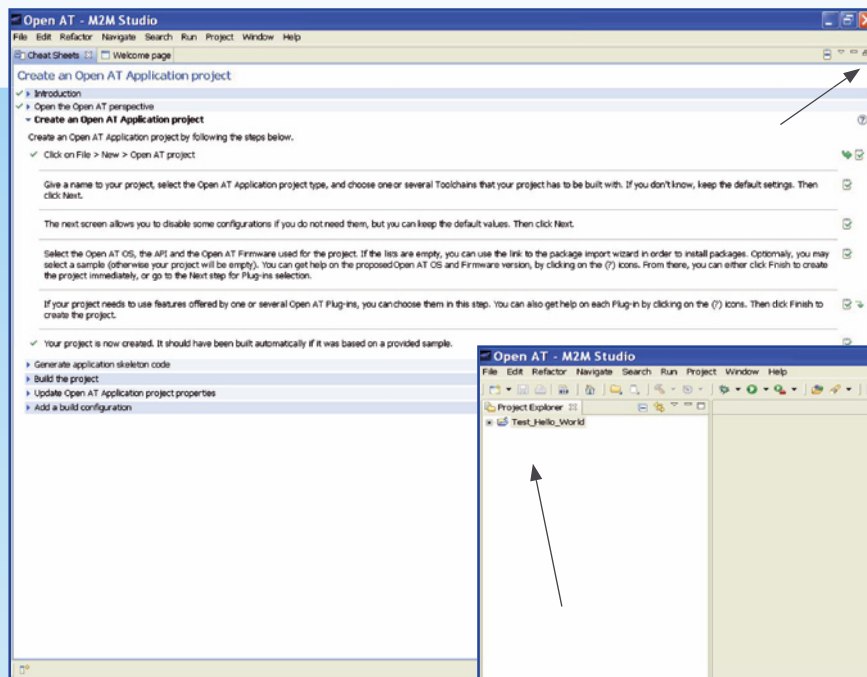
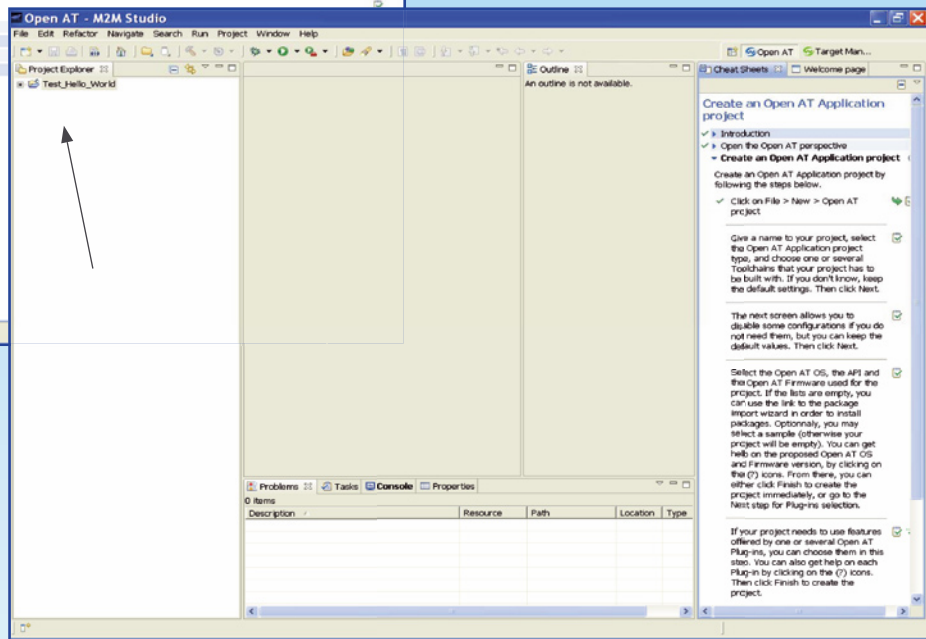


Fig. 5 - I risultati della finestra Cheet Sheet.

Fig. 6 - Ambiente Open AT-M2M Studio.



re. Se clicchiamo su *Test_Hello_World* possiamo esplorare il codice dell'applicazione *Hello World* (Fig. 7).

L'applicazione è composta da una direttiva *include adl_global.h* che deve essere sempre presente all'interno delle applicazioni Open AT e che serve per includere tutte le principali API della suite Open AT. A seguire viene riservata un'area di memoria di stack per le chiamate

a funzione. L'applicazione è costituita da due funzioni, la funzione *HelloWorld_TimerHandler* e la funzione principale *adl_main* la quale deve

essere sempre presente in ogni applicazione. La prima riga della funzione *adl_main* è un richiamo alla macro *TRACE*:

TRACE ((1, "Embedded : Hello World"));

L'effetto di questa macro è stampare la stringa nella finestra di shell del debugger. Il primo parametro è il livello di TRACE; sono possibili fino a 32 livelli di trace. Ogni livello di TRACE può essere attivato/disattivato a run time attraverso la finestra di *Remote Tasks Monitor* (Fig. 8) che si apre automaticamente quando lanciamo il debug. Questo è uno strumento molto

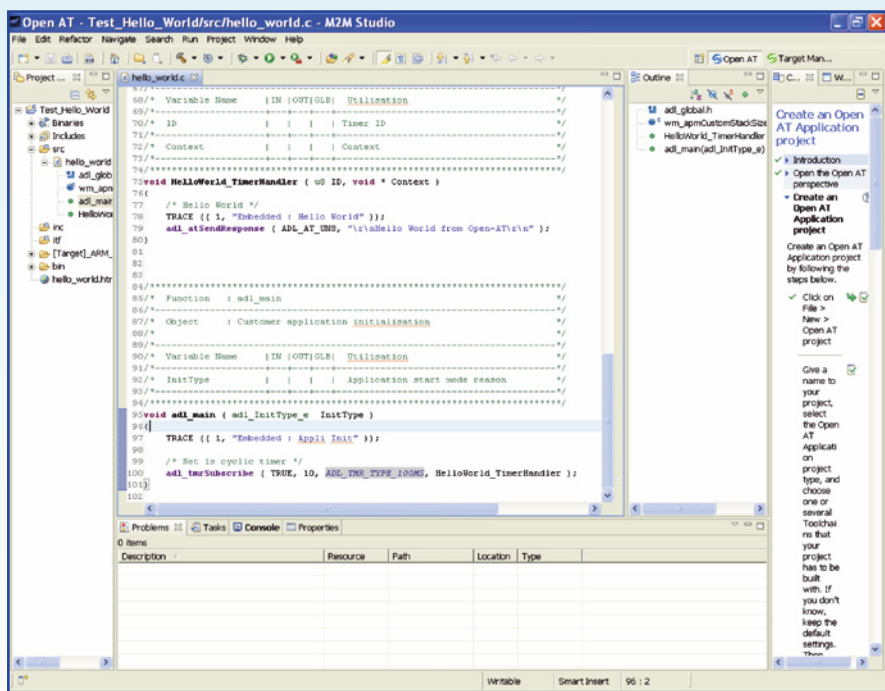


Fig. 7 - Ambiente M2M Studio: il codice dell'applicazione.



potente ed indispensabile per tener traccia del flusso del programma. Il secondo parametro della TRACE è la stringa da stampare. Sono consentiti, alla funzione C `printf`, anche parametri analoghi all'interno della stringa; ad esempio:

```
u16 var_int;
// dichiarazione di un
intero 16b bit
```

```
TRACE (( 1, "Stampa
un intero:%d",var_int ));
// stampa del contenuto della variabile var_int.
```

La seconda riga di codice all'interno della funzione `adl_main` inizializza un timer. Con questa funzione si attiva un evento timer, che scatta ogni secondo ed è associato alla funzione `HelloWorld_TimerHandler`. Il risultato è che ogni secondo il flusso di programma salta all'interno della funzione `HelloWorld_TimerHandler`. Il codice di esecuzione del timer stampa la stringa di TRACE `Embedded: Hello World` nella finestra di shell, la stessa stringa con la riga di codice:

```
adl_atSendResponse ( ADL_
AT_UN, "\r\nHello World
from Open-AT\r\n");
```

viene inviata sulla seriale di default UART 1 del modulo e quindi visualizzata sulla finestra di terminale. La differenza fra TRACE e

Fig. 8
La finestra Remote Tasks Monitor.

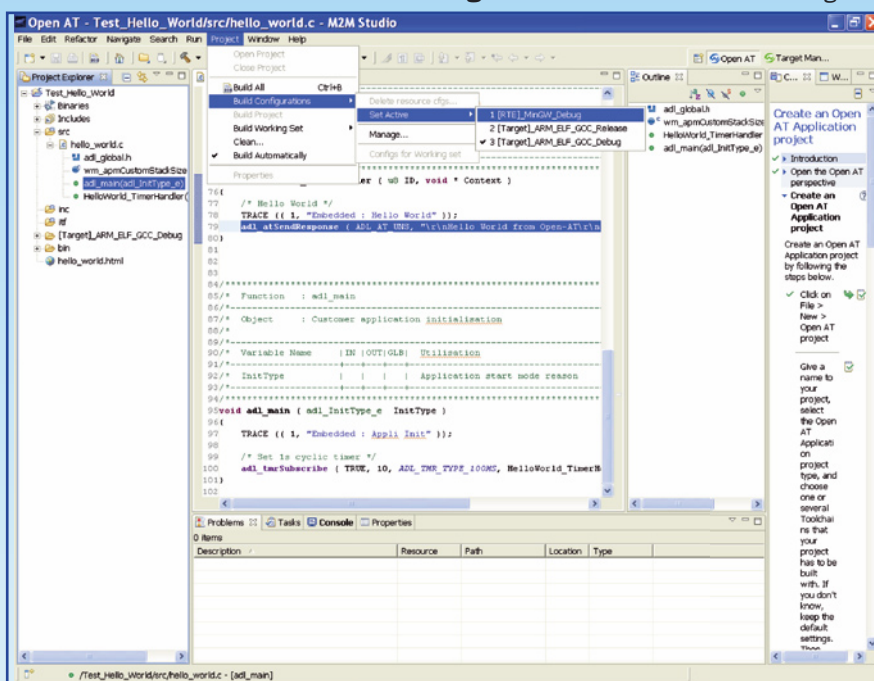


Fig. 9 - M2M Studio - Set RTE Debug.

`adl_atSendResponse` sta nel fatto che mentre con la macro TRACE la stringa è visualizzata solo sulla shell di debug, con la funzione `adl_atSendResponse` viene stampata anche sulla seriale e quindi la stessa stringa viene visualizzata anche in fase di run-time quando

Fig. 10 - M2M Studio Debug RTE.

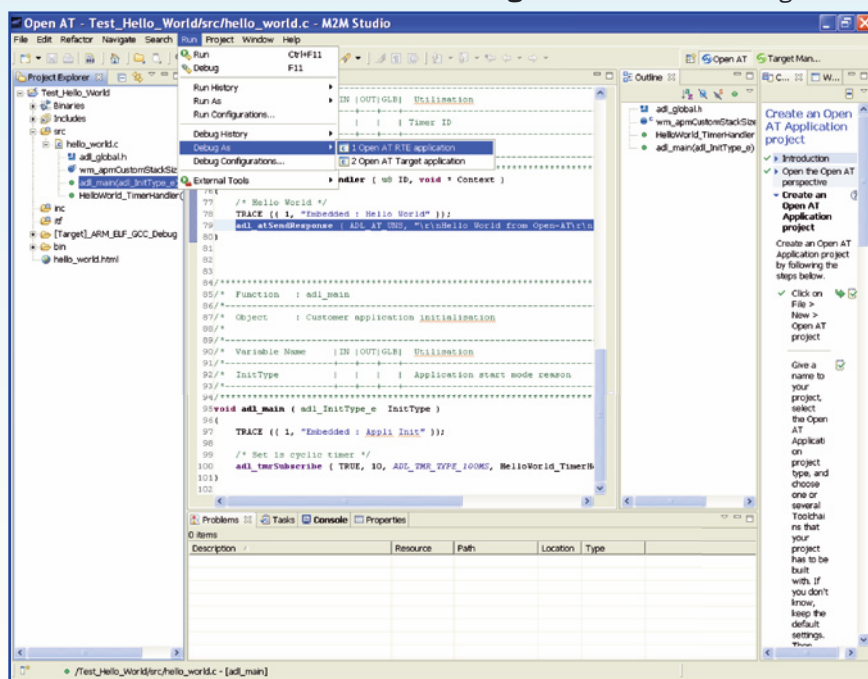




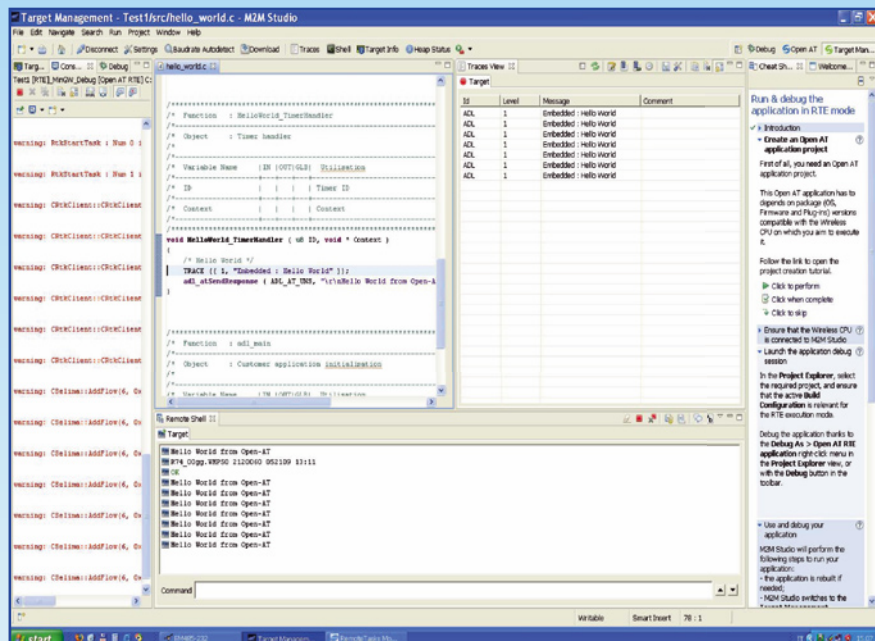
Fig. 11
Serial link Manager

si avvia un applicativo tipo *hyper terminal* connesso alla UART 1 del modulo GSM. Ora va detto che fin dall'inizio di questo corso abbiamo sottolineato un concetto importante: la funzione `adl_main` viene eseguita solo all'avvio della applicazione; non c'è alcun loop infinito tipo `while(1)` o `for(;;)` come succede nelle tipiche applicazioni per microcontrollori. Dovete infatti ricordare che stiamo lavorando all'interno di un sistema operativo. Tutte le risorse che si vogliono utilizzare, come timer, GPIO, interrupt ecc. vanno inizializzate (sottoscritte, nel gergo Open AT); ad ogni risorsa viene automaticamente assegnato un identificativo (ID) e viene richiesto di definire una funzione di aggancio (Handler) che viene messa in esecuzione dal S.O. nel momento opportuno. Nel caso del timer dell'esempio, tale funzione viene eseguita ogni secondo. Bisogna quindi pensare e sviluppare le applicazioni secondo questa filosofia. Si capisce quindi quanto sia importante, per noi, inserire le TRACE nel nostro programma, allo scopo di riuscire a seguirne il flusso in ogni momento. Ciò sarà più evidente nel seguito di questo corso, quando avremo modo di approfondire il concetto e di imparare a gestire nel modo corretto programmi più complessi che fanno uso di un elevato numero di eventi sottoscritti.

Bene, arrivati a questo punto proviamo a lanciare l'applicazione in modalità di Debug RTE; per farlo clicchiamo su *Project-> Build Configurations-> Set Active-> [RTE]_MinGW_Debug* per attivare la modalità di debug *Real Terminal Emulator* (Fig. 9).

Ora facciamo clic sul pulsante di compilazione oppure, dal menu, impartiamo il comando *Project-> Build All*; in basso, nella finestra *Console*, otteniamo il risultato della compilazione.

Fig. 12- La finestra di dialogo Target Management



Nella finestra *Problem* troviamo gli eventuali errori e/o warning.

Adesso lanciamo il debug cliccando su *Run-> Debug As-> Open AT RTE Application* (vedere la Fig. 10).

Se la *target board* (Demoboard oppure il modulo "Augusto") è collegata correttamente al PC, si apre la finestra di *Remote Tasks Monitor* della Fig. 8. Se la UART cui è collegata la target board non è corretta, possiamo cambiarla cliccando sul *Serial Link manager* che si trova nella barra delle applicazioni sul desktop in basso a destra (vedere la Fig. 11).

Una volta aperta la COM corretta, avviamo il debugger cliccando sul pulsante *Start* della finestra *Remote Tasks Monitor*. Con questa finestra è possibile attivare/disattivare le TRACE a run-time oppure fermare l'applicazione cliccando su STOP. Selezioniamo *ADL** e la *Trace 1*. Il risultato di tale operazione, dopo aver impartito lo *Start*, è riportato in Fig. 12. Sulla finestra di *Shell* e sulla finestra di *trace* vediamo comparire ogni secondo la stringa di *Hello World*.

Complimenti: avete appena fatto funzionare in debug la vostra prima applicazione sviluppata con Open AT! Adesso blocchiamo il flusso dell'applicazione cliccando su *Stop*. Poi ritorniamo all'editor premendo il pulsante

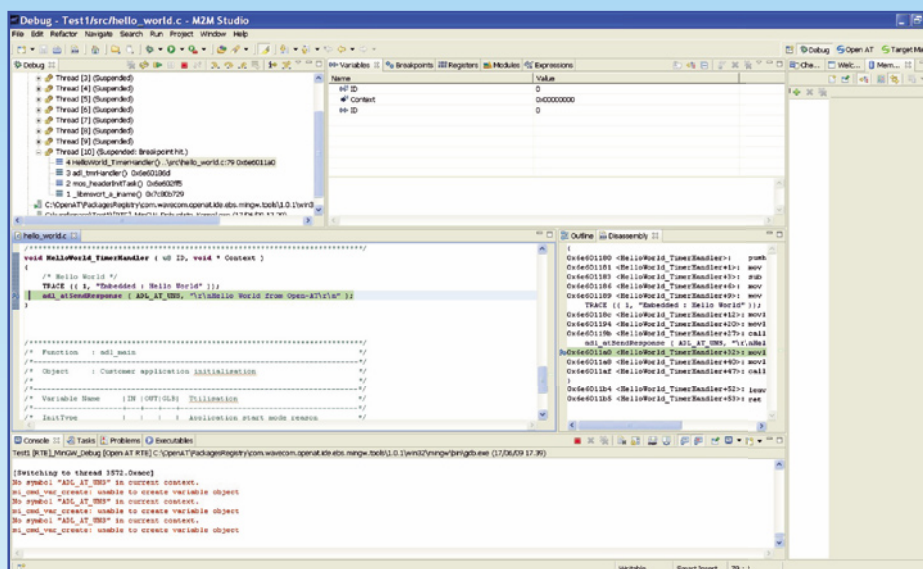


Fig. 13 - Svolgimento del debug.

“Open AT” in alto a destra e clicchiamo due volte sul bordo sinistro a fianco della riga di codice seguente allo scopo di inserire un breakpoint:

adl_send_response (ADL_AT_UNS, “\\r\\nHello World from Open-AT\\r\\n ”);

Adesso facciamo clic sul pulsante di *Debug* (quello con l’insetto...) e lanciamo di nuovo il programma in modalità debug RTE. Il flusso del programma ora si interrompe ogni secondo, fermandosi sulla riga di codice indicata dal breakpoint (**Fig. 13**). Con il tasto funzione F6 seguiamo passo-passo, debuggando riga per riga senza, quando viene incontrata una chiamata, entrare nelle funzioni. Con F5, invece, otteniamo lo stesso effetto entrando anche nelle chiamate a funzione. Con il tasto F8 riprendiamo il flusso normale. All’interno del debugger possiamo vedere anche il contenuto delle variabili, stringhe, classi o strutture che sono state dichiarate nel nostro codice. Finora abbiamo lavorato sempre in modalità *DEBUG RTE*, ma va detto che possiamo ripetere tutti i passi fin qui visti semplicemente selezionando la modalità debug *TARGET* (cliccare su *Project-> Build Configurations-> Set Active-> Target_ARM_EL3_GCC_DEBUG*). In questo caso il debug viene fatto direttamente sulla WCPU. La differenza tra le due modalità di debug sta principalmente nell’uso che viene fatto della memoria: nella modalità *RTE*

la memoria programma e dati è emulata dal PC; sul target vengono inviati tutti i comandi necessari per muovere i GPIO e tutte le altre periferiche, ma il programma vero e proprio sta girando in modalità real time sul PC. Si ottiene così un debug molto rapido nell’esecuzione dei breakpoint e nella analisi dei contenuti di memoria. Invece nella modalità *TARGET* l’applicazione è realmente in esecuzione nella WCPU; quando si avvia questa modalità, infatti, il programma viene automaticamente caricato nella WCPU. In modalità *TARGET* si deve utilizzare il connettore JTAG invece della porta seriale; l’uso del JTAG è limitato ad applicazioni professionali e richiede il permesso esplicito della Wavecom e l’attivazione della porta JTAG, sempre da parte della Wavecom. Il modulo “Augusto” da noi utilizzato in questo corso è predisposto con un connettore JTAG e la WCPU è configurata per poter lavorare anche con il JTAG. Nelle prossime puntate vedremo come eseguire il debug delle nostre applicazioni utilizzando anche questo potente strumento. Finora abbiamo utilizzato il solo debugger; facciamo un ultimo sforzo e proviamo la nostra applicazione in modalità *RELEASE*, andando a scaricare ed eseguire l’applicazione direttamente sulla WCPU. Dalla finestra *Target Management* clicchiamo su “Download” e attendiamo che il programma venga scaricato. La CPU viene quindi automaticamente resettata e sulla shell viene inviato automaticamente il comando:

AT+WOPEN=1

che serve per mandare in esecuzione la predetta applicazione. Al massimo dopo 8 secondi (questo tempo è necessario al sistema operativo affinché carichi l'applicazione) dovrebbe comparire la scritta *Hello World from Open-AT* sulla *shell*, periodicamente ogni secondo. Il comando AT sopra indicato occorre digitarlo solo dopo aver scaricato una nuova applicazione nella target board. A questo punto possiamo anche spegnere e riaccendere la target board, perché il programma partirà comunque senza dover digitare alcun comando AT. Se chiudiamo l'M2M Studio e apriamo *Hyperterminal*, vedremo la scritta *Hello World from Open-AT* sulla finestra dello stesso *Hyper Terminal*, segno che la nostra applicazione sta girando sul target e non c'è più nulla che la legghi al PC, tranne il fatto che usa la seriale per inviare la stringa.

Se volete caricare altri esempi o altre applicazioni, occorre prima che cancelliate questa applicazione (ossia quella che sta girando); potete fare ciò con i seguenti comandi AT:

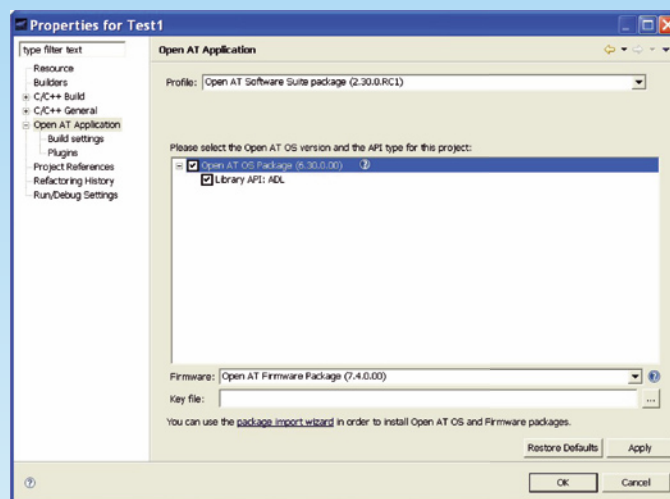
AT+WOPEN=3 // cancella la memoria flash object della applicazione

AT+WOPEN=4 // cancella la memoria flash A&D della applicazione

Alla trattazione della memoria sarà dedicata un'apposita puntata di questo corso; per ora accontentatevi di sapere che la "A&D" è la memoria Flash dove risiede l'applicazione e che la "object" è la memoria Flash, eventualmente utilizzata dalla nostra applicazione, destinata a salvare i dati.

Con la procedura descritta in questo articolo introduttivo potete testare altri esempi tra i molti disponibili in fase di creazione del progetto. Per lo studio del codice degli esempi conviene sempre avere sottomano la documentazione delle API ADL, che è disponibile on-line cliccando sul tasto destro nella finestra del *Project Explorer* di *M2M Studio*, quindi selezionando *Properties* e facendo clic sull'icona con il punto di domanda. Ciò permette di aprire la pagina con i collegamenti a tutte le guide Open AT (Fig. 14). La guida più importante è la *Open AT Development Guide*. La stessa guida la si può trovare anche in locale

Fig. 14 - La finestra di dialogo *Project Properties*.



nella cartella di installazione dell'Open AT:
 C:\OpenAT\PackagesRegistry\com.wavecom.
 openat.ide.spm.lib.os.model\6.30.0.00\doc

Il file è siglato *ADL_User_Guide.pdf*.

Per ogni applicazione di esempio invece, è disponibile un file html a corredo, nel quale vengono spiegati l'utilizzo e le funzioni di libreria utilizzate. I progetti di esempio con la relativa documentazione si possono trovare in locale nella seguente cartella:

C:\OpenAT\PackagesRegistry\com.wavecom.
 openat.ide.spm.lib.os.model\6.30.0.00\ADL\
 samples.

Nella prossima puntata analizzeremo più nel dettaglio la pin-out e l'hardware della WCPU WMP50, andando a costruire una demoboard per essa, che utilizzeremo per tutti i nostri test. Chi volesse maggiori chiarimenti sulla materia, potrà consultare il file *ADL_User_Guide.pdf* e la Guida on-line *M2M Studio 1.0*, entrambi scaricabili dal sito Internet di Wavecom. Nel rimandarvi alla prossima puntata del corso, vi auguriamo buon lavoro e buon divertimento con Open AT!

