Mensile di progettazione elettronica, attualità scientifica, novità tecnologiche.

Elettronica In

oltre l'elettronica

www.elettronicain.it

Anno XVIII - n. 167
Giugno 2012



Robot Area

Rover Cingolato con braccio a pinza

- Combinatore vocale GSM
- Lettore RFID con SD card e bus I²C
- Programmatore HV modulare per MCU Atmel
- Comunicazioni audio a LED
- Minibus, un bus per l'automazione
- Corso Atmel OPEN SOURCE

Swinging Labyrinth





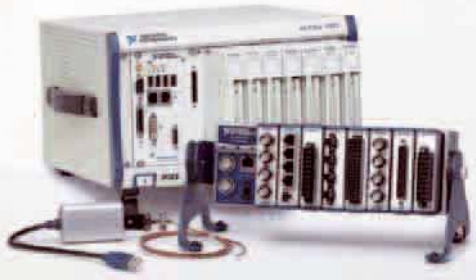
Speciale Maker Faire®

Tutorial openPICUS

Ad ogni misura
la sua soluzione



Per ogni tipo di applicazione, dalle più semplici alle più complesse, la piattaforma d'acquisizione dati di National Instruments è in grado di gestire ogni misura nel modo più efficace. Qualunque sia il tuo budget, il numero di canali richiesto e quali siano le tue specifiche esigenze a livello di prestazioni, la potenza degli I/O e il software in grado di incrementare la produttività dei tuoi progetti saranno sempre all'altezza della sfida da affrontare. Questa combinazione vincente permette di bilanciare prestazioni e investimenti fornendo la massima accuratezza di misure sempre scalabili e affidabili. **Progettata per prestazioni eccellenti.**



>> Scopri la piattaforma NI per l'acquisizione dati alla pagina web **ni.com/daq/i**

NATIONAL INSTRUMENTS

Elettronica In

www.elettronicain.it
Rivista mensile, anno XVIII n. 167
GIUGNO 2012

Direttore responsabile:
Arsenio Spadoni
(*Arsenio.Spadoni@elettronicain.it*)

Redazione:
Gabriele Daghetta, Paolo Gaspari, Boris Landoni, Marco Magagnin, Simone Majocchi, Davide Scullino, Alessandro Sottocornola
(*redazione@elettronicain.it*)

Grafica:
Alessia Sfulcini, Filippo Gori
(*Alessia.Sfulcini@elettronicain.it*)

Ufficio Pubblicità:
Monica Premoli (0331-752668)
(*Monica.Premoli@elettronicain.it*)

Ufficio Abbonamenti:
Elisa Guarniero (0331-752668)
(*Elisa.Guarniero@elettronicain.it*)

DIREZIONE, REDAZIONE, PUBBLICITA':
VISPA s.n.c. - via Adige 11 - 21013 Gallarate (VA)
Telefono 0331-752668 Fax 0331-778112

Abbonamenti:
Annuo 10 numeri Euro 45,00
Estero 10 numeri Euro 90,00
Le richieste di abbonamento vanno inviate a:

VISPA Edizioni s.n.c.
via Adige 11, 21013 Gallarate (VA)
tel. 0331-752668.

Distribuzione per l'Italia:
SO.DI.P. Angelo Patuzzi S.p.A.
via Bettola 18 - 20092 Cinisello Balsamo (MI)
Telefono 02-660301 Fax 02-66030320

Stampa:
ROTO3 Spa - Via Turbigo, 11/b
20022 CASTANO PRIMO (MI)

Elettronica In:
Rivista mensile registrata presso il Tribunale di Milano con il n. 245 il 3-05-1995.
Prezzo di copertina Euro 6,00. Gli arretrati nei formati cartaceo e digitale (pdf) sono acquistabili sul sito della rivista al prezzo di Euro 6,00.
Poste Italiane Spa - Spedizione in abbonamento Postale - D.L. 353/2003 (conv. in L. 27/02/2004) art.1 comma 1 - DCB Milano.
Vispa Edizioni è iscritta al Registro Operatori della Comunicazione n. 3754 del 27/11/2001.
Impaginazione ed immagini sono realizzati in DeskTop Publishing con programmi Adobe InDesign e Adobe Photoshop per Windows.
Tutti i contenuti della Rivista sono protetti da Copyright. Ne è vietata la riproduzione, anche parziale, la traduzione e più in generale la diffusione con qualsiasi mezzo senza l'autorizzazione scritta da parte dell'Editore. I circuiti, il firmware ed il software descritti sulla Rivista possono essere realizzati solo per uso personale, ne è proibito lo sfruttamento a carattere commerciale e industriale. Tutti possono collaborare con ElettronicaIn. L'invio di articoli, materiale redazionale, programmi, traduzioni, ecc. implica da parte del Collaboratore l'accettazione dei compensi e delle condizioni stabilite dall'Editore (*www.elettronicain.it/ase.pdf*). Manoscritti, disegni e foto non richiesti non verranno in alcun caso restituiti. L'utilizzo dei progetti e dei programmi pubblicati non comporta alcuna responsabilità da parte della Società Editrice.
© 1995÷2012 VISPA s.n.c.

Crisi, banda larga e agenda digitale

L'altro giorno, in macchina, mi è capitato di ascoltare un bollettino sul traffico che mi informava che, dopo pochi chilometri, avrei trovato un incidente con relativa coda. Giunto sul posto, l'incidente era già stato risolto e la coda smaltita.
Mi sono venute subito in mente le polemiche degli ultimi anni relative all'inadeguatezza del sistema informativo sulla viabilità e gli sforzi (costosi e spesso inutili) fatti da vari enti pubblici e privati per fornire un valido servizio di questo tipo. Ho rivisto le immagini delle tante sale di controllo della Polstrada, di Autostrade per l'Italia, del CCISS, ISO radio, eccetera, e ho pensato alle relative risorse di persone e mezzi impegnati in questa attività che, come nel mio caso, spesso non funziona nel modo sperato.



La cosa che però mi ha fatto maggiormente riflettere su questo episodio (abbastanza marginale, del resto) è stato il fatto che il mio smartphone, gratuitamente e senza costi per la collettività, mi ha fornito – tramite il servizio Google Maps Traffico – notizie in tempo reale più precise sulla viabilità, segnalandomi col colore verde che quel tratto di strada non presentava più problemi. Notizie ricavate, tra l'altro, automaticamente dagli smartphone di altri utenti che si trovavano in zona.
Quale morale ricavare da questo episodio? Semplicemente la conferma che l'innovazione in tutti i campi non può che passare dai servizi Internet e che quindi è sempre più urgente e necessario migliorare e rafforzare le reti per la trasmissione dati, sia la fibra ottica che i sistemi wireless. In sostanza dobbiamo al più presto creare quelle strade e autostrade digitali in grado di "reggere" tutti i servizi, nuovi e vecchi, che sono alle porte. E pigiare sull'acceleratore della digitalizzazione dei servizi pubblici, dalla sanità all'istruzione, dal fisco a tutte le pratiche di sportello.
In questo momento di grave crisi economica per l'Italia si può risparmiare su tutto ma non sulla creazione di un sistema di trasmissione dati che garantisca a tutti, a casa e in mobilità, perlomeno una velocità di 100 Mbsec; una infrastruttura ormai indispensabile non solo per la normale operatività ma soprattutto per migliorare l'efficienza del nostro sistema economico, tentando così di uscire dalla crisi. Dell'argomento si sta occupando proprio in questi giorni un team di esperti del ministro Passera: ci auguriamo che, come noi, tutti si convincano dell'urgenza di trovare le risorse per questa fondamentale infrastruttura per il Paese.

Arsenio Spadoni
(Arsenio.Spadoni@elettronicain.it)

Sommario

Giugno 2012
n. 167

ARTICOLI

22 GADGET
SWINGING LABYRINTH

Applicazione didattica o gioco da luna-park, questo sistema è indubbiamente un doppio esercizio di abilità, che vi metterà alla prova prima nel realizzarlo e poi nell'usarlo.



31 SICUREZZA
COMBINATORE VOCALE GSM

Collegato a una centrale di allarme, quando viene attivato almeno uno degli ingressi di cui dispone chiama i numeri telefonici memorizzati facendo ascoltare avvisi vocali personalizzati. Seconda e ultima puntata.



41 APPLICAZIONI
LETTORE RFID CON SD CARD E BUS I2C

Concludiamo la realizzazione del lettore di tessere RFID per controllo accessi con la descrizione del software e delle modalità di comunicazione e interfacciamento mediante il protocollo I2C. Ultima puntata.



50 EVENTI
MAKER FAIRE: LA FESTA DEL DIY

Un sole estivo ha accolto gli oltre 100 mila visitatori che hanno affollato nel week-end del 19-20 maggio la cittadina della Bay Area per la settima edizione della Maker Faire, l'evento dedicato a makers, innovatori e comunità di DIY.





IN COPERTINA:
Un gadget tecnologico e divertente adatto alla stagione estiva in arrivo, per mettere alla prova la vostra abilità, sia nel costruirlo che nell'utizzarlo.

ELENCO INSERZIONISTI

- Fiera di Novegro
- Futura Elettronica
- National Instruments
- Open Picus
- RM Elettronica
- Robotica
- SMAU
- www.mdsrl.it

Robot Area

82

ARRIVA IN ITALIA FIRST LEGO LEAGUE

ROVER CINGOLATO CON BRACCIO A PINZA



55 DIDATTICA

UN PROGRAMMATORE HV MODULARE PER MCU ATMEL

Dopo avere costruito il programmatore HV affrontiamo i concetti generali della programmazione e descriviamo il firmware della sezione Master. Terza parte.



67 AUDIO

COMUNICAZIONI AUDIO A LED

Utlilizziamo i LED per illuminare un ambiente e nel contempo sfruttiamo l'emissione luminosa per trasmettere un segnale audio in FM.



77 DIDATTICA

ALLA SCOPERTA DI OPEN PICUS

Dopo avere presentato i dispositivi hardware della serie FlyPort e l'ambiente di sviluppo gratuito del progetto OpenPicus, è tempo di iniziare ad approfondire le possibilità di utilizzo con qualche semplice programma. Terza Puntata.



La tiratura di questo numero è stata di 18.500 copie.

Mensile associato all'USPI, Unione Stampa Periodica Italiana

RUBRICHE

01 EDITORIALE
05 ELETTRONICA INSIEME
13 APPUNTAMENTI
17 COMPONENTI & SISTEMI
20 EVENTI
110 SCIENZA & TECNOLOGIA
125 RINNOVABILI
128 MERCATINO

99 DOMOTICA

MINIBUS, UN BUS PER L'AUTOMAZIONE

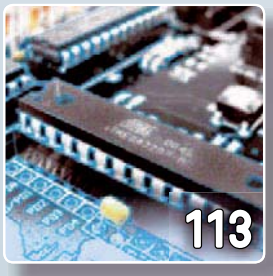
Proseguiamo la presentazione del sistema flessibile di automazione domestica con la descrizione del modulo di espansione e dei moduli periferici specializzati collegati al controllo centrale con il protocollo I2C a livello di segnale potenziato per poter coprire lunghe distanze. Seconda puntata



113 DIDATTICA

CORSO ATMEL OPEN SOURCE

Iniziamo l'esplorazione delle periferiche interne dell'ATMEGA16 analizzando quelle dedicate al conteggio di tempi ed eventi, per capire quali sono i loro compiti e come possano essere utilizzate all'interno di un programma. Sesta puntata.



Camera Car HD a colori con monitor integrato

- Riprese video in alta risoluzione (1280x960 pixel, M-Jpeg, 30 fps)
- Possibilità di scattare foto (Jpeg)
- Registrazione su SD Card (max 32 GB, non inclusa)
- Angolo di ripresa 120°
- Display 2,5" TFT
- Microfono e batteria ricaricabile integrati
- Uscita USB per collegamento a PC
- Dimensioni 100 x 22 x 61 mm
- Fissaggio tramite ventosa
- Alimentatore da auto e cavo USB inclusi



€ 68,00
cod. CAMCOLVC15

RIVIVI LE EMOZIONI PIÙ ESALTANTI!

Prezzi IVA inclusa.

ACTION CAMERA... RIPRESE SENZA CONFINI ! IDEALI PER REALIZZARE RIPRESE IN MOVIMENTO IN ATTIVITÀ SPORTIVE, AI FINI ASSICURATIVI O COME PURO DIVERTIMENTO. INSTALLABILI SUI CRUSCOTTI DI AUTOVEICOLI, CASCHI O SUL MANUBRIO DI MOTO E BICICLETTE.

Action camera HD con registratore

- Memorizzazione dei video (MPEG-4, 30fps) e delle immagini (JPEG) su micro SD (max. 32 GB, non inclusa)
- Dispone di 2 potenti LED per riprese al buio
- Telecamera CMOS 5 Mega pixel
- Microfono incorporato
- Dimensioni 105 x 45 x 37 mm
- Alimentazione tramite 2 batterie ricaricabili al Litio (incluse)



€ 189,00
cod. CAMCOLVC13



Action camera con registratore

- Memorizzazione dei video (AVI, 30fps) e delle immagini (JPEG) su memoria integrata da 64MB o su micro SD (max. 8 GB, non inclusa)
- Può funzionare anche come Webcam
- Telecamera CMOS 1,3 Mega pixel
- Microfono incorporato
- Dimensioni Ø31 x 101 mm
- Alimentazione tramite batteria ricaricabile al Litio (integrata)

€ 118,00
cod. CAMCOLVC14

FUTURA ELETTRONICA

Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775 • Fax. 0331/792287

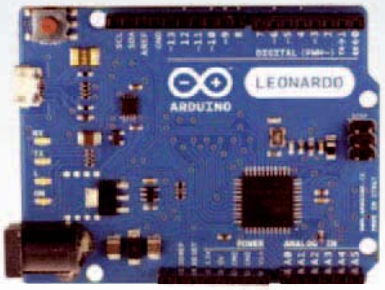
Maggiori informazioni su questi prodotti sono disponibili sul sito *www.futurashop.it* tramite il quale è anche possibile effettuare acquisti on-line.

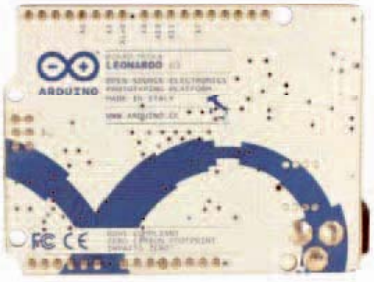
Queste pagine sono dedicate ai suggerimenti, alle richieste ed alle segnalazioni dei lettori. Raccomandiamo, per quanto possibile, di proporre argomenti di interesse generale. Contattateci numerosi all'indirizzo: *redazione@elettronicain.it*

Arduino Leonardo

Ho visto che esiste la disponibilità di una nuova scheda Arduino, per la precisione l'Arduino Leonardo. Dalle caratteristiche a me sembra che sia molto simile all'Arduino Uno. Potete evidenziarmi le particolarità di questa scheda e per quali progetti vale la pena adottarla?

Andrea Contino - Camerino (MC)





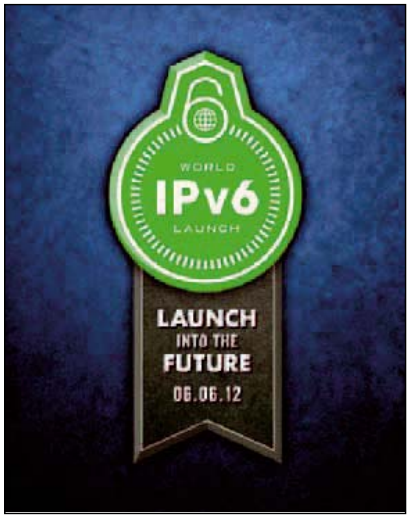
Effettivamente da poche settimane è disponibile Arduino Leonardo, l'ultimo nato della famiglia. Le novità sono importanti, Leonardo è basato sul microcontrollore ATmega32u4, che ha grosso modo le stesse caratteristiche di capacità, memorie RAM e flash dell'ATmega328, che equipaggia il fratello Uno. Possiede 20 ingressi/uscite digitali, dei quali sette utilizzabili come uscite PWM e 12 come ingressi analogici, un oscillatore a 16 MHz, un connettore micro USB, ICSP e power. Ma dove sta la novità? Sicuramente nell'impiego del micro ATmega32u4 che incorpora nel chip il modulo di comunicazione USB. Questo permette la riduzione dei componenti esterni, semplificando la scheda e la possibilità di gestire direttamente da software le funzionalità USB.
Se ti sembra poco pensa di poter programmare Leonardo in modo da farlo vedere al vostro PC come una tastiera ed un mouse, oppure come un joystick USB ed immagina quali applicazioni di interfaccia nel mondo reale possono essere sviluppate, come per esempio l'interfaccia tra un simulatore di volo e una serie di comandi come volantini, pedali, cloche e interruttori interpretati da Leonardo e veicolati al PC via USB come se li aspetta il software di simulazione. Nell'utilizzo di Leonardo bisogna tenere conto di alcuni comportamenti diversi rispetto ai fratelli della famiglia. Non possedendo un chip dedicato alla comunicazione seriale, la porta COM che viene attivata è una porta virtuale, così come quella dal lato del PC, ogni volta è necessario riconoscerla ed aggan-ciarla per stabilire la comunicazione; in compenso, Leonardo non si resetta quando si apre la comunicazione seriale da PC. Quando si sviluppa codice di emulazione di mouse e tastiera è bene prendere qualche precauzione durante il test, in quanto, una volta in esecuzione, mouse e tastiera vengono "animati" da Leonardo, e potrebbe essere complesso tornarne in possesso.
E' opportuno verificare se esistono funzionalità di sistema operativo atte ad abilitare e disabilitare periferiche esterne, oppure prevedere uno switch fisico, o ancora testare il programma mediante messaggi di debug via seriale, per accertarne il funzionamento corretto. Infine Leonardo è programmabile con le versioni di IDE a partire dalla 1.0.1.

Indirizzamento Internet IPv6

Nella stampa specializzata così come in alcune trasmissioni televisive si sente affermare che il sistema di indirizzamento di Internet si sta saturando, in quanto si sta raggiungendo il numero massimo di computer indirizzabili. Si parla anche di indirizzamento IPv6 ma sinceramente non so di che si tratta e quali impatti avrà sui comuni utilizzatori. Potete darmi delle delucidazioni sull'argomento?

Flavio Salaroli - San Gemini (TR)

L'argomento è di estrema attualità, anche se le notizie al riguardo non vengono diffuse come dovrebbero. Ad un anno di distanza dal World IPv6 Day è pianificato per il 6 giugno 2012, il World IPv6 Launch, con lo slogan "Questa volta è per davvero". Ma di che si tratta? In estrema sintesi si tratta della attivazione in contemporanea da parte dei principali fornitori di servizi Internet, del sistema di indirizzamento IP a 128 bit denominato IPv6. Cerchiamo di capire; fin dalle sue origini (la RFC. n° 1 di Internet risale al 1969) il protocollo di comunicazione ha utilizzato uno schema di indirizzi composto da quattro byte, per es. 192.168.1.125 che permetteva di indirizzare fino a 2^{32} diversi host e computer in genere, più di 4 miliardi. Una scelta assolutamente lungimirante per un protocollo destinato ad usi militari, che doveva prevedere di collegare agenzie militari, università, istituti di ricerca e fornitori del settore. Anche dopo il suo rilascio per usi commerciali, nel 1993, il sistema non ha mostrato segni di cedimento, fino all'avvento della comunincazione di massa con device di ogni genere che prevedono la accessibilità in rete; a parte i PC, stampanti, telefoni, smartphone, tablets, telecamere ed ora sempre più microcontrollori, elettrodomestici, navigatori, orologi, antifurti, sistemi di controllo industriale: l'Internet nelle cose. A questo punto il sistema di indirizzamento ha raggiunto il suo limite e la IANA (Internet Assigned Numbers Authority, la "custode" degli indirizzi IP) ha assegnato gli ultimi indirizzi disponibili nel Febbraio 2011.
Nel frattempo già nel 2004 l'ICANN (Internet Corporation for Assigned Names and Numbers), dopo lunghe battaglie tra diverse proposte



Logo del World IPv6 Launch

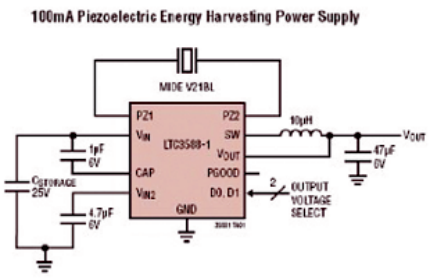
ha annunciato come soluzione per l'estensione degli indirizzi l'adozione dell'indirizzamento IPv6, come definito nelle RFC 2373 e RFC 2374. L'indirizzamento IPv6 prevede 128 bit di lunghezza, cioè 32 cifre esadecimali, per un totale di 2^{128} dispositivi indirizzabili. Non potendo scrivere la cifra per esteso, ricorriamo ad un esempio. Se prendiamo come riferimento la superficie terrestre, per ogni metro quadrato avremmo 655.570.793.348.866.943.898.599 indirizzi IPv6 unici disponibili (cioè 655 571 miliardi di miliardi), contro 0,000007 IPv4 (cioè solo 7 IPv4 ogni milione di metri quadrati). Dovremmo averne abbastanza per un pò. Ma come si scrive un indirizzo IPv6? Come 8 gruppi di 4 cifre esadecimali (ovvero 8 parole di 16 bit ciascuna), separati dai due punti. Ad esempio 2001:0db 8:85a3:0000:1319:8a2e:0370:7344 è un indirizzo IPv6 valido; inoltre sono previste varie modalità di contrazione per semplificarne la scrittura. Ed i nostri vecchi indirizzi IP? Niente paura, conviveranno almeno fino al 2025 ed i due sistemi di indirizzamento possono viaggiare in contemporanea sulla rete grazie al fatto di poter essere univocamente identificati dal campo "version", i primi quattro bit dell'header, genialmente previsto agli albori del protocollo, che vale 4 per i vecchi indirizzi e 6 per i nuovi (c'è posto fino a 15). Dopo quella data il nostro indirizzo di esempio iniziale diventerà semplicemente ::192.168.1.125. Ma a noi cosa cambierà? Per la maggior parte degli utenti con browser non proprio vecchi non dovrebbe cambiare nulla, il 6 giugno i maggiori fornitori Internet come Google, Facebook e simili, migreranno. Se vi potrete collegare senza problemi siete nel futuro. In caso contrario, basta aggiornare il browser.

Elettricità dalle pastiglie piezo

Ho nel cassetto alcune pastiglie piezoelettriche, anche di grandi dimensioni (tipo quelle usate nei trasduttori per le sirene degli impianti di allarme) e, sapendo che usate al contrario possono produrre elettricità, vorrei fare qualche esperimento; l'idea sarebbe realizzare un minitrasmettitore alimentato autonomamente recuperando qualcosa dalle vibrazioni sulla tubatura di una macchina. Ho però alcuni dubbi sulla fattibilità della cosa, perché se non ricordo male le pastiglie piezo sviluppano sì tensione, però non erogano granché di corrente. Che tipo di circuito potrei usare? E riesco ad ottenere qualche decina di milliampere??

Domenico Clerico - Bassano del Grappa (VI)

Come saprai, una delle soluzioni di Energy Harvesting, per recuperare elettricità da energia meccanica o termica altrimenti sprecata, consiste nell'applicare a tubazioni in pressione, macchinari o altri meccanismi, piastre piezoelettriche, sfruttando la loro caratteristica di generare una differenza di potenziale quando vengono sollecitate meccanicamente. Per ottenere una tensione abbastanza stabile e una corrente che possa caricare una batteria o alimentare un utilizzatore, bisogna interfacciare la pastiglia piezo con un circuito regolatore, che può essere un integrato specifico per queste applicazioni; ad esempio, Linear Technology produce l'LTC3588-1 (trovi il data-sheet su www.linear.com/product/LTC3588-1) che è un regolatore espressamente pensato per soluzioni di Energy Harvesting e più esattamente per alimentare dispositivi elettronici partendo dalla tensione ottenibile da una pastiglia o un trasduttore piezoelettrico. Il componente è un regolatore buck che accetta all'ingresso da 2,7 a 20 V, integra un raddrizzatore a doppia semionda ed eroga in uscita fino a 100 mA; la tensione di uscita può essere fissata a 1,8 V, 2,5 V, 3,3 V o 3,6 V. L'integrato dispone di una funzione detta UVLO che in caso di tensione d'ingresso sufficiente permette di immagazzinare l'energia prodotta in un condensatore fin quando il convertitore DC/DC non è in grado di regolare.
Il componente è l'ideale per alimentare dispositivi di vario genere, purché funzionanti con tensioni fino a 3,6 V, quindi anche sensori e microtrasmettitori da pochi mW.



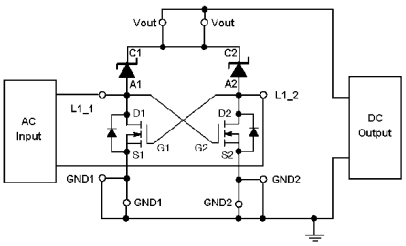
Caricare le batterie senza fili

Ho visto in commercio una torcia elettrica che si ricarica introducendola nella sua sede. E allora, direte voi, dov'è la novità? Beh, se ci ho visto bene, la ricarica avviene senza collegamento elettrico, dato che alla base della torcia non ci sono contatti. Direi che la cosa funziona in modo wireless. Vedere in funzione tale sistema mi ha fatto venire voglia di costruirne uno anch'io, solo che non so come dimensionarlo e come trasferire l'elettricità senza troppe perdite. Il principio lo so: basta alimentare una bobina di filo di rame con una tensione alternata e rilevare il campo elettromagnetico prodotto con un secondo avvolgimento, al capo del quale otterrò la tensione per alimentare il carico, da raddrizzare mediante diodi. Ma come devo scegliere le bobine? E qual è la frequenza adatta?

Mario Cavazzano - Castellalto (TE)

Mettersi qui a fare i calcoli ha poco senso, anche perché la frequenza, l'induttanza e la costituzione delle bobine, come altri fattori, dipendono strettamente dal tipo di apparato che vuoi ottenere, ossia dalla corrente che vuoi fornire al carico, dalla distanza tra la bobina irradiante e quella ricevente ecc. In linea di massima devi evitare frequenze superiori ai 100 kHz, sia per non disturbare le trasmissioni radio, sia per non avere troppe perdite nei circuiti driver.
Come ricevitore, possiamo consigliarti un recentissimo circuito integrato prodotto dalla ON Semiconductor fatto specificatamente per la ricarica wireless delle batterie di telefoni cellulari, PC tablet, videocamere, lettori digitali e

navigatori GPS; si tratta dell'NMLU1210, che contiene un raddrizzatore a doppia semionda da 20 volt, composto da MOSFET a canale N configurati a ponte semisincrono e due diodi Schottky.
Il circuito integrato è stato appositamente pensato per raddrizzare la tensione ricavata dalla bobina secondaria (cioè quella che riceve il campo magnetico e alimenta il carico) e fornire fino a 3,2 ampere di corrente. La frequenza di funzionamento è fino ad alcune decine di MHz, quindi non c'è alcun problema a lavorare a 100 kHz, per esempio.



Fuel Cell SOFC anche per le case

Ho seguito con molto interesse lo sviluppo delle pile a combustibile soprattutto per l'uso nelle automobili, dove potrebbero risolvere i problemi di autonomia di cui le attuali soffrono a causa dello sfavorevole rapporto energia/peso (densità di energia) e nella realizzazione di piccoli generatori per produrre elettricità nelle case partendo direttamente dal metano, disponibile in tutti i centri urbani in una rete molto capillare e attualmente usata per cucinare e per il riscaldamento. Purtroppo so anche dei costi e della scarsa disponibilità dei materiali necessari a realizzare su larga scala fuel-cell di tipo PEM, che peraltro funzionano solo con l'idrogeno puro, gas costoso da ottenere e che richiederebbe una rete di distribuzione a parte. Ma perché le altre fuel-cell, ad esempio MCFC e SOFC, sono disponibili solo per le centrali elettriche e non si valuta ancora l'impiego, ad esempio, nelle auto o nei piccoli generatori domestici?

Franco Tombolini - San Felice del Benaco (BS)

Le SOFC (Solid Oxide Fuel Cell) sono le pile a combustibile ideali per le centrali di media taglia (potenze tra 100 kW e 1 MW) perché possono funzionare con il metano ed effettuano direttamente il reforming da cui estraggono l'idrogeno (rilasciando CO_2 come scarto) grazie all'elevata temperatura de-

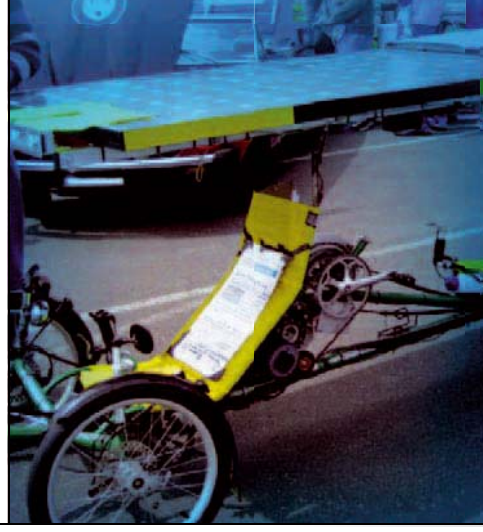
Vieni con noi a New York con OpenWheels

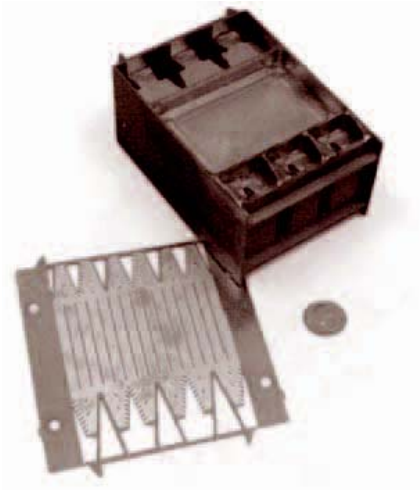


La Maker Faire di San Mateo è arrivato troppo in fretta; visti i tempi effettivamente ristretti, abbiamo deciso di prorogare il termine ultimo per presentare le candidature ad OpenWheels. Se vuoi partecipare, ti comunichiamo che il termine è stato spostato al 15 luglio 2012.
Il vincitore presenterà il proprio progetto al World Maker Faire di New York, in programma il 29 e 30 Settembre 2012. Il viaggio e il soggiorno li offriamo noi.

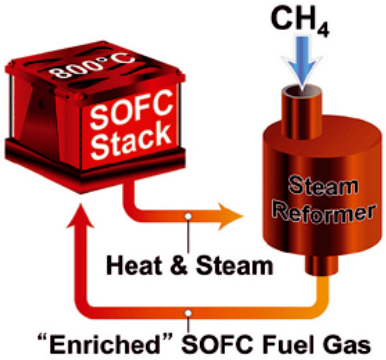
Il progetto OpenWheels

Ricordiamo che il concorso propone ai candidati di progettare l'elettronica di controllo di una sorta di Segway (un veicolo elettrico a due ruote capace di rimanere in equilibrio, avanzare o arretrare, girare e fermarsi) e immaginare applicazioni innovative ed originali di questo mezzo. Quello che vi chiediamo è di realizzare il sistema elettronico di controllo che sia open source e a basso costo; non dovete preoccuparvi della meccanica, perché l'abbiamo già realizzata noi.
Il veicolo deve essere alimentato a 24 volt e la trazione ed il controllo del movimento sono affidati a una coppia di motori da 24 V - 250 W. Il movimento richiesto viene deciso sia dal manubrio (che interviene su un potenziometro da 10 kohm che l'elettronica legge per rilevare le richieste) sia dalla pedana su cui poggia il conducente. Questi dati dovrebbero esservi sufficienti per progettare l'elettronica di controllo che dovrebbe comprendere come minimo: 1) un caricabatterie; 2) una piattaforma inerziale; 3) un sistema di controllo a microcontrollore con relativo firmware; 4) due driver di potenza per l'azionamento dei motori.
Se vuoi partecipare, scrivici all'indirizzo redazione@elettronicain.it dopo aver letto le condizioni di partecipazione nella pagina web *www.elettronicain.it/openwheels*.
Le candidature verranno presentate sul blog della rivista e sottoposte al giudizio dei nostri lettori; ai tre candidati che otterranno i maggiori consensi verranno forniti i componenti necessari alla realizzazione dell'elettronica di controllo della piattaforma OpenWheels. Al vincitore verrà fornita una piattaforma completa OpenWheels sulla quale montare l'elettronica di controllo per sperimentare le altre soluzioni innovative.
Sei un progettista e vuoi metterti alla prova? E allora ...forza! Potresti essere tu a girare sul tuo Segway per i padiglioni della World Maker Faire!





gli elettrodi; tuttavia per il costo e l'efficienza che hanno, sono tipicamente indicate per le installazioni fisse sopra una certa potenza. Non vanno bene, almeno in teoria, per le auto, anche a causa dell'elevata temperatura (circa 1.000 °C) a cui funzionano e dell'ingombro.
Questo, almeno fino a poco tempo fa, quando è stato dato l'annuncio che sono state realizzate SOFC capaci di raggiungere il 57% di efficienza (contro il 50% massimo raggiunto fino a prima) e idonee alla realizzazione di impianti di piccola taglia.
La nuova fuel-cell SOFC è stata sviluppata presso il Dipartimento di Energia del Pacific Northwest National Laboratory e ricava direttamente l'idrogeno per reforming del metano; può essere realizzata in tagli da 1 a 100 kW di potenza e rappresentano una valida alternativa alle PEM nella realizzazione di generatori locali, anche grazie al fatto che possono essere collegate direttamente alla rete di distribuzione del gas metano. In futuro, si potranno eliminare gli ultimi tratti delle linee elettriche per l'utenza civile e produrre l'elettricità direttamente all'interno degli edifici.



Se lo smartphone scalda troppo

Uso sovente il mio cellulare UMTS/HSDPA per la connessione ad Internet mobile ed ho notato in più occasioni che l'apparecchio si scalda molto di più di quando lo uso per telefonare.
Come mai succede questo?
Forse la potenza RF impiegata è maggiore di quella richiesta per le comunicazioni a voce?
Oppure devo preoccuparmi perché non funziona bene?
Armando Missora - Peschici (FG)

Quel che ti accade è abbastanza normale quando ti trovi in zone coperte dalla rete HSDPA e 3,5 G, perché l'elevata velocità di connessione porta ad un maggiore consumo di elettricità e quindi ad un maggiore riscaldamento del modulo dati dello smartphone; in pratica rispetto al funzionamento in UMTS, quello in HSDPA a 7,2 o 14,4 Mbps (o anche più, in base alla copertura offerta dalla rete radiomobile del gestore cui ti connetti) comporta maggiore dissipazione di potenza e perciò di calore, proprio in virtù dell'elevata velocità di elaborazione dei dati e delle perdite di commutazione dei dispositivi logici che ne derivano. Non è quindi un problema di potenza di trasmissione del telefono, che sostanzialmente resta la stessa, ma di potenza elettrica richiesta dai circuiti che trattano i dati, dato che, analogamente a quanto accade nelle CPU dei computer, più velocità significa maggior dissipazione di potenza e quindi più calore da smaltire.
Noterai il maggior consumo anche perché la batteria dello smartphone si scaricherà prima.



Web Forum e supporto tecnico

Hai un problema con uno dei circuiti pubblicati? Vorresti effettuare una modifica? Accedi al nostro Web Forum e Supporto Tecnico dove i nostri tecnici (ma anche gli altri lettori) ti aiuteranno a chiarire qualsiasi dubbio di natura tecnica.
Collegati a:

www.elettronicain.it/webforum

Touch screen

Vorrei utilizzare una interfaccia di controllo per una mia applicazione con Arduino, che sia diversa dai soliti pulsanti, anche perchè me ne servono diversi e non vorrei impegnare tutti gli ingressi digitali, o dover prevedere combinazioni di pulsanti troppo complicate. Avete una soluzione alla portata di hobbista?
Mattia Martini - Chiusdino (SI)

Per risponderti prendiamo spunto dal mondo dei videogiochi e in particolare dalla piattaforma Nintendo Ds, che utilizza uno schermo di tipo touch facilmente reperibile come ricambio. Con le sue dimensioni di circa 7 x 5,5 centimetri è l'ideale per la tua applicazione. In realtà non è un vero e proprio touch screen ma solo la parte trasparente, sensibile al tocco. Per questo è possibile creare una maschera in carta dei comandi come pulsanti, icone o menù sui quali puntare per attivare le funzioni del tuo codice. Una alternativa ancora più professionale potrebbe

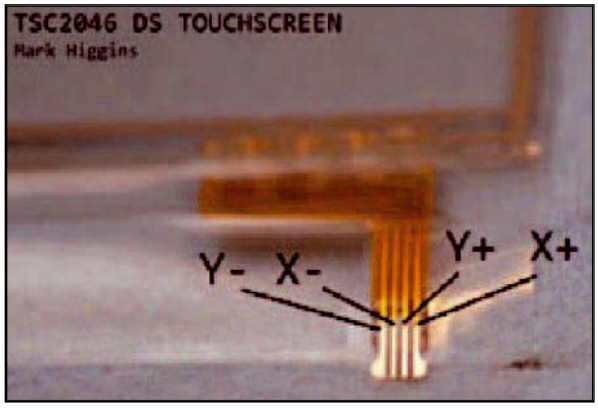


essere quella di posizionarlo sopra un piccolo monitor delle stesse dimensioni in modo da gestire schermate interattive, variabili a seconda del contesto ed al limite sovraimposte a immagini video. Lo schermo è composto da due film resistivi collegati ciascuno a due dei quattro terminali di uscita X+, X-, Y+ ed Y-. Ogni volta che si tocca lo schermo i due film resisteivi si cortocircuitano nel punto toccato, le coordinate del quale possono essere determinate leggendo i valori di resistenza ai capi dei terminali.
Per leggere il valore corrispondente alla coordinata X si devono collegare i terminali X+ a 5V e X- massa, si legge il valore in ADC dal terminale Y+. Viceversa per leggere il valore della coordinata Y si porta Y+ a 5V, Y- a massa e si legge il valore da X+.
Esiste anche una libreria apposita per il suo utilizzo con Arduino che è scaricabile liberamente da github.com/practicalarduino/TouchScreen
Un frammento di codice per darti l'idea di come reperire le coordinate X e Y in Arduino.

```
// Lettura del valore della coordinata X
  pinMode( A1, INPUT );     // Analog pin 1 (X-)
  pinMode( A3, INPUT );     // Analog pin 3 (X+)
  pinMode( A0, OUTPUT );    // Analog pin 0 (Y-)
  digitalWrite( A0, LOW );  //Analog pin 0  (Y-) a massa
  pinMode( A2, OUTPUT );    // Analog pin 2 (Y+)
  digitalWrite( A2, HIGH ); // Analog pin 2 (Y+) collegato a +5V
  xVal = analogRead( 1 );   // Lettura della coordinata X da X-

  // Lettura del valore della coordinata Y
  pinMode( A0, INPUT );     // Analog pin 0 (Y-)
  pinMode( A2, INPUT );     // Analog pin 2 (Y+)
  pinMode( A1, OUTPUT );    // Analog pin 1 (X-)
  digitalWrite( A1, LOW );  //Analog pin 1 (X-) a massa
  pinMode( A3, OUTPUT );    // Analog pin 3 (X+)
  digitalWrite( A3, HIGH ); // Analog pin 3 (X+) collegato a +5V
  yVal = analogRead( 0 );   // Lettura della coordinata X da Y-
```

Se vuoi realizzare la soluzione più completa con il monitor e le scritte variabili puoi utilizzare lo shield Overlay Video descritto nel numero 165 di Aprile 2012 della nostra rivista.



VOLUME SPECIALE
"Primi passi con Arduino"



Pensato per i principianti che si avvicinano all'elettronica e per gli hobbisti che non hanno mai affrontato la programmazione.

Acquistalo su
www.futurashop.it

FUTURA ELETTRONICA
Via Adige, 11 - 21013 Gallarate (VA)
Tel. 0331/799775 • Fax. 0331/792287

La Più Prestigiosa Rivista di Progettazione Elettronica

per un anno, direttamente a casa tua!

Campagna abbonamenti

Ei 2012



Elettronica In propone mensilmente - oltre a corsi, articoli didattici e notizie dal mondo dell'elettronica - tantissimi progetti tecnologicamente molto avanzati, sia dal punto di vista hardware che firmware, cercando di illustrare nella forma più chiara e comprensibile le modalità di funzionamento, le particolarità costruttive e le problematiche firmware dei circuiti proposti.

Solo per te ...

- L'abbonamento annuo (di 10 numeri) costa € 45,00 anzichè € 60,00 con un risparmio del 20% sul prezzo di copertina.
- È il massimo della comodità: ricevi la rivista direttamente al tuo domicilio, senza scomodarti a cercarla e senza preoccuparti se il numero fosse esaurito.
- Se il prezzo di copertina della rivista dovesse aumentare nel corso dell'abbonamento, non dovrai preoccuparti: il prezzo per te è bloccato!
- Hai a disposizione un servizio di consulenza: i nostri tecnici sono a tua disposizione per fornirti tutte le informazioni necessarie riguardanti i progetti pubblicati.

Abbonarsi è facilissimo...

- **ON-LINE**
  compilando il modulo riportato nella pagina "Abbonamenti" del nostro sito www.elettronicain.it Se possiedi una carta di credito potrai effettuare il pagamento contestualmente alla richiesta.
  È anche possibile abbonarsi on-line richiedendo il pagamento tramite C/C postale.

- **COMPILANDO ED INVIANDO VIA POSTA O FAX IL MODULO DI ABBONAMENTO RIPORTATO A PIE' DI PAGINA**
  Riceverai direttamente a casa tua un bollettino personalizzato di C/C postale. L'abbonamento decorrerà dal primo numero raggiungibile.

  PER I RINNOVI ATTENDERE NOSTRO AVVISO

NEW

Scoprila anche su iPad



È disponibile per gli abbonati anche la versione digitale della rivista da sfogliare e leggere on-line.



ABBONAMENTO annuale SOLO € 45,00 + OMAGGIO

Sottoscrivendo l'abbonamento annuo alla rivista riceverai in omaggio la DISCOUNT CARD



che ti consentirà di usufruire di uno sconto del 10% su tutti i prodotti acquistati direttamente (oppure on-line) da Futura Elettronica.



Gli abbonati riceveranno anche in regalo ...

- I DVD con tutti i numeri della rivista dal n. 1 al n. 123.
- Un volume a scelta della collana "L'ELETTRONICA PER TUTTI".
- La possibilità di scaricare gratuitamente dal sito www.elettronicain.it i file sorgente dei firmware utilizzati in molti dei progetti pubblicati.



Sì desidero abbonarmi per un anno alla rivista *Elettronica In*. Riceverò i seguenti omaggi:

☒ **DISCOUNT CARD "Futura Elettronica"**

Uno, a scelta, tra questi volumi della collana "L'Elettronica per tutti".

- ☐ Programmiamo con i PIC
- ☐ Alla scoperta della CCTV
- ☐ Tecnologia e apparati GPS
- ☐ Energie alternative
- ☐ Primi passi in elettronica
- ☐ Top 100 Arduino Projects
- ☐ Primi passi con Arduino

Resto in attesa del primo numero e degli omaggi.

Nome: Cognome:

Via: N°:

CAP: Città: Provincia:

Telefono: e-mail:

Data: Firma:

Resto in attesa di vostre disposizioni per il pagamento.
Ai sensi dell'art. 13 del D.Lgs. 196/2003, in tema di protezione dei dati personali, Vispa snc informa che i dati liberamente forniti attraverso la compilazione del presente modulo saranno trattati e conservati presso Vispa snc in conformità a tutte le normative vigenti. I dati suddetti saranno trattati su supporto cartaceo e informatico, mediante sistemi di protezione atti alla tutela della riservatezza, per finalità connesse all'esecuzione dei rapporti contrattuali instaurati. Inviare in busta chiusa a:

VISPA Edizioni

VISPA snc
Via Adige 11
21013 Gallarate (VA)
Fax: 0331-778112

RADIANT

AND•SILICON

59° MERCATO DEL CONSUMER ELECTRONICS

22 - 23 SETTEMBRE 2012

sab. 9.00-18.00 - dom. 9.00-17.00



in contemporanea

NOVEGRO PHOTO VideoCINE 17ª Edizione Mostra-Mercato di apparecchi fotografici e cinematografici

PARCO ESPOSIZIONI NOVEGRO

MILANO/LINATE AEROPORTO ✈

Con il patrocinio di Città di Segrate Assessorato alla Cultura

www.parcoesposizioninovegro.it

11-13 LUGLIO 2012

MICROMACHINE/MEMS

Tokyo, Giappone

Exhibition Micromachine/MEMS

In questi ultimi anni, l'innovazione aperta e la fusione di diversi campi sono diventati concetti chiave per la ricerca all'avanguardia e lo sviluppo. Micromachine /MEMS è un evento fieristico dedicato a sviluppatori e produttori di tutto il mondo che si occupano di microelettronica e di applicazioni delle micro e nanotecnologie. I produttori impegnati nello sviluppo e nella produzione dei dispositivi, e gli OEM che utilizzano questi dispositivi, troveranno nell'evento il luogo ideale per uno scambio di idee con i partner commerciali e rappresentanti del mondo accademico.
Micromachine/MEMS 2012 coinvolgerà vari settori tra cui i sistemi informatici, elettronica consumer, elettrodomestici elettronici, strumenti di precisione, automobili, biotecnologie, robotica (nell'apposita sessione Robotech, un evento collaterale a Micromachine/MEMS), aerospaziale, energia. Gli espositori (348 lo scorso anno) misurandosi anche in incontri one-to-one con il pubblico, cercheranno di conoscere le ultime esigenze di mercato e le tendenze attuali per fornire il meglio ai consumatori finali. Giunta alla sua diciottesima edizione ed organizzata da Mesago Messe Frankfurt Corp. (*www.mesago-messe-frankfurt.com*) la fiera si terrà al Tokyo International Exhibition Center (Tokyo Big Sight) della capitale nipponica.

www.micromachine.jp/en

10-12 LUGLIO 2012

INTERSOLAR NORTH AMERICA

San Francisco – U.S.A.

inter solar NORTH AMERICA

Per il quarto anno consecutivo, innovatori dell'industria solare e uomini d'affari si riuniscono a San Francisco per discutere delle tendenze dell'industria e valutare le ultime novità della tecnologia solare. Eventi di networking, sia durante la mostra che nel corso del ciclo di conferenze abbinato (che si terrà dal 9 al 12 luglio) e forum tematici, proveranno a gettare le basi del fiorente business dell'industria solare. Nel corso dei tre giorni di esposizione, 19.170 visitatori registrati provenienti da circa 80 nazioni, saranno presenti al Moscone Center West di San Francisco (California, U.S.A.) dove, nelle North Halls, 834 espositori (tra cui il colosso tedesco Siemens) esibiranno il meglio dell'industria del solare fotovoltaico e termico e dove sarà possibile incontrare progettisti di sistemi per lo sfruttamento dell'energia solare. Nel ciclo di conferenze, 220 relatori di livello mondiale forniranno ai partecipanti informazioni e suggerimenti, confrontandosi con il pubblico. Quest'anno Intersolar North America indice il Premio per l'innovazione più prestigioso del settore dell'energia solare, ossia Intersolar AWARD, finalizzato a premiare i progetti più interessanti di sfruttamento dell'energia solare proposti dagli espositori a tutte le manifestazioni Intersolar che si tengono nelle varie regioni del mondo. I vincitori saranno scelti in base all'innovazione espressa dai loro progetti e all'eccellenza di servizi e prodotti. Il 5 ° anniversario della Intersolar AWARD vedrà assegnare dalla giuria tre premi per i vincitori delle altrettante categorie, che sono: Fotovoltaico, Tecnologie solari termiche e Tecnologie di produzione fotovoltaica. Con l'aggiunta della nuova categoria, "Progetti solari in Nord America", specificamente finalizzata all'innovazione del Nord America, l'evento punta a mettere in luce il talento e l'innovazione che escono dal mercato nordamericano. Dalla fondazione, Intersolar Award ha fatto conoscere al pubblico 301 innovazioni provenienti da 25 paesi.



www.intersolar.us

MOSTRE MERCATO

VALLEFIORITA (CZ)
FIERA DEL RADIOAMATORE
Quartiere fieristico - Vallefiorita (CZ)
Organizzazione: Comune di Vallefiorita
Tel: 3278876549
email: *comitato.fiera@libero.it*
07 e 08 Luglio 2012

LOCRI (RC)
23a MOSTRA MERCATO DEL ADIOAMATORE, DELL'ELETTRONICA E DEL COMPUTER
Istituto di I° Grado F. Maresca
Organizzazione: G.C.S.
Tel: 3382704739 - Fax: 096421152
email: *gallcosimo@tiscali.it*
web: *www.cisarlocri.net*
14 e 15 Luglio 2012

CECINA (LI)
FIERA DI ELETTRONICA E MODELLISMO
Area Espositiva "I Pini" (Ex Cecinella)
Organizzazione: Millennium Eventi
Tel: 0696840758
email: *info@millenniumfiere.it*
web: *www.millenniumfiere.it*
14 e 15 Luglio 2012

TARVISIO (UD)
5a ED. FIERA DELL'ELETTRONICA
Palazzetto dello sport
Via degli Atleti Azzurri
Organizzazione: Tarvisiofiera
Tel: 3492133698 Fax: 33967715O4
email: *info@tarvisiofiera.info*
web: *http://tarvisiofiera.info/*
18 e 19 Agosto 2012

CEREA (VR)
COMPUTERFEST & RADIOAMATORE
AreaExp – Cerea (VR)
Organizzazione: Compendio Fiere
Tel: 337676719 Fax: 0572498205
email: *info@compendiofiere.it*
web: *www.compendiofiere.it*
25 e 26 Agosto 2012

L'elenco aggiornato di tutte le Mostre Mercato del 2012 è disponibile sul sito *www.elettronicain.it* sul quale è anche possibile scrivere un commento sulle fiere visitate.

L'elenco completo dei più importanti eventi nazionali e internazionali di elettronica, sicurezza e fonti rinnovabili è disponibile sul nostro sito: www.elettronicain.it/MyMain/riviste/p_rivista.asp?pagina=fiereeventi

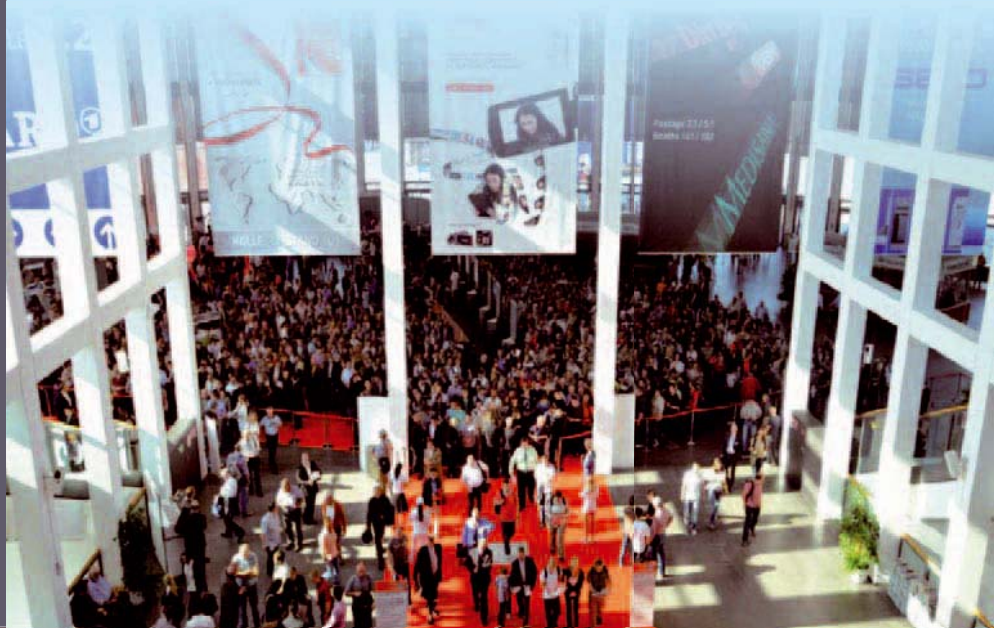
■ 31 AGOSTO - 5 SETTEMBRE 2012

IFA

Berlino, Germania

IFA Berlino è la più importante fiera europea dedicata all'elettronica consumer, agli elettrodomestici e alla tecnologia per migliorare la vita di tutti i giorni, specialmente nell'ambiente domestico. Home entertainment, sistemi di navigazione satellitare e le ultime innovazioni nella musica digitale, ma anche telecomunicazioni, saranno i leit motiv della manifestazione (giunta alla sua quinta edizione) che prenderà il via il 30 agosto prossimo presso i padiglioni della Fiera di Berlino.
Qui, aziende leader dell'elettronica di consumo e del settore degli elettrodomestici, offriranno al pubblico una visione globale del panorama della tecnologia rivolta alla casa ed al tempo libero. In mostra, oltre agli impianti per la diffusione di audiovisivi, della musica e per l'intrattenimento, ci saranno i moderni elettrodomestici che rendono la vita in casa più facile e vanno incontro alle tendenze attuali, come la cucina sana, il wellness, l'uso efficiente dell'energia, rispondendo con comandi semplici e tecnologie intelligenti che vedono gli elettrodomestici del futuro sempre più integrati in rete.
Non mancheranno i robot-elettrodomestici, sempre più diffusi e capaci di svolgere vari compiti, il più comune dei quali è indubbiamente pulire i pavimenti anche quando siamo fuori casa. Tema dominante sarà, anche quest'anno, l'efficienza energetica dei prodotti tecnologici: un confronto con i modelli prodotti circa 15 anni fa, dimostra che l'energia risparmiata dagli apparecchi moderni è tra il 50 e il 70 per cento. In termini di consumo di acqua lo stesso vale per lavatrici e lavastoviglie. Un alto grado di efficienza energetica è un must assoluto per i produttori che vogliono soddisfare la clientela di oggi, attenta al prezzo, alle funzioni, ma anche all'ecologia.
In questo contesto gli elettrodomestici moderni si integrano nelle Smart Grid, perché le reti elettriche intelligenti rendono possibile sfruttare tariffe elettriche flessibili, tramite la programmazione o l'attivazione remota di una lavatrice o lavastoviglie intelligente, dotata di connessione dati, quando l'elettricità è più conveniente. Un focus sull'integrazione degli apparecchi di casa e le Smart Grid farà il punto sullo stato della tecnologia, la situazione normativa (ossia la stesura di regole e protocolli universali per la comunicazione degli elettrodomestici con le Smart Grid) e le possibilità di business in questo nascente settore.

www.ifa-berlin.com



■ 25-26 AGOSTO 2012

JARL HAM FAIR 2012

Tokyo, Giappone



Come ogni anno ritorna, al Tokyo Big Sight, il più grande evento fieristico del mondo dedicato ai radioamatori, che proporrà le ultime novità di case storiche come Kenwood, ICOM, Yaesu, Tokyo Hi Power ed altre ad un pubblico che malgrado Internet e i social network crede ancora nell'utilità e nel piacere di comunicare via radio. JARL Ham Fair è organizzato da The Japan Amateur Radio League, un'associazione radioamatoriale nata nel lontano 1926 da un gruppo di appassionati di comunicazione radio, che attualmente vanta il maggior numero di stazioni radio nel mondo. L'edizione 2011 ha visto la presenza di circa 32.000 partecipanti nell'arco dei due giorni di manifestazione e quest'anno gli organizzatori contano di ripetere e incrementare questa presenza. Al JARL, i visitatori provenienti da tutto il mondo potranno incontrare i produttori e i rivenditori, ma anche i loro pari dei club radioamatoriali; inoltre troveranno, accanto al nuovo, anche surplus radio e tante notizie interessanti, tanto per l'hobbista, quanto per l'esperto radioamatore. Sulla scia dell'edizione precedente, anche quest'anno sono previste le premiazioni dei concorsi per il miglior apparecchio autocostruito e l'annuale contest.

www.jarl.or.jp/English

■ 10-12 LUGLIO 2012

SEMICON WEST 2012

San Francisco – U.S.A.

SEMICON West è l'evento di punta (inaugurato nel 1971) per l'industria globale della microelettronica e rappresenta un'interessante vetrina che mette in mostra nuovi prodotti e tecnologie per la progettazione e la produzione microelettronica, attraverso tutta la supply chain microelettronica: dall'automazione della progettazione elettronica, alla fabbricazione dei wafer da cui verranno ricavati i chip, alla produzione finale (assemblaggio, confezionamento e test). Più che per i semiconduttori, SEMICON West è anche una vetrina per i mercati e le tecnologie emergenti nate dall'industria microelettronica, tra cui quelle per i sistemi micro-elettromeccanici (MEMS), il fotovoltaico, i dispositivi elettronici e display flessibili, la nanoelettronica e l'illuminazione a LED.
Accanto all'area espositiva, SEMICON West 2012 offrirà più di 40 ore di sessioni tecniche e presentazioni attraverso uno spazio espositivo delle tecnologie diviso in tre aree tematiche (chiamate TechXPOT) incentrate su temi cruciali del settore come la progettazione e produzione di semiconduttori, LED ad alta luminosità, MEMS, elettronica stampata e flessibile, e altre tecnologie correlate. Le TechXPOTs sono il luogo di SEMICON West dove conoscere gli ultimi sviluppi tecnologici e le innovazioni nella produzione dei semiconduttori, ma anche per scoprire cosa c'è di nuovo e le prospettive nel campo della microelettronica.
SEMICON West 2012 si terrà in concomitanza con Intersolar North America 2012, sempre al Moscone Center, però nella zona South.
SEMICON West è patrocinato da SEMI (www.semi.org), l'associazione dell'industria globale al servizio della filiera di produzione della nano e microelettronica. Dal 1970, SEMI è stata impegnata ad aiutare i suoi associati a crescere in modo più redditizio, creare nuovi mercati e far fronte alle sfide comuni del settore. SEMI ha sedi ad Austin, Pechino, Bangalore, Berlino, Bruxelles, Grenoble, Hsinchu, Mosca, San Jose, Seoul, Shanghai, Singapore, Tokyo e Washington DC.

www.semiconwest.org



Sistemi di illuminazione a LED

Soluzioni per il risparmio energetico adatte ad ogni ambiente.



Pannello illuminante 112 LED



€ 115,00
cod. P3030-W

- Flusso luminoso 880 lm
- Consumo 12 W
- Dimensioni 29,5 x 29,5 cm

Disponibile anche nella versione a 416 LED. 3000 lm, 45 W, 59,5 x 59,5 cm.
cod. P6060-W € 239,00

Pannello da incasso 36 LED



€ 72,00
cod. P1717-WW

- Flusso luminoso 880 lm
- Consumo 16 W
- Dimensioni 17 x 17 cm

Tubo luminoso 56 LED



€ 48,00
cod. T8-HWSMD

- Flusso luminoso 840 lm
- Consumo 10 W

Faro da incasso 12 LED



€ 96,00
cod. D1202-WW

- Flusso luminoso 1410 lm
- Consumo 28 W
- Diametro 16 cm

Lampada a LED - E27



€ 39,00
cod. G60ME27-9W-WW

- Flusso luminoso 900 lm
- Consumo 9 W

Lampada 72 LED - E27



€ 27,00
cod. CORN72SMD-WW

- Flusso luminoso 900 lm
- Consumo 9,5 W

Prezzi IVA inclusa.

FUTURA ELETTRONICA

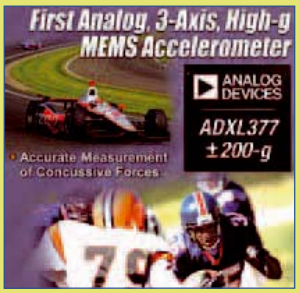
www.futurashop.it

Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775 - Fax. 0331/792287

Maggiori informazioni su questi prodotti e acquisti on-line su www.futurashop.it.

Da AD, il primo accelerometro analogico a 3-assi ad alto G

Progettato per misurare accuratamente le forze che sollecitano gli atleti in alcuni sport e quelle degli impatti nelle apparecchiature industriali, l'ADXL377 di Analog Devices è il primo accelerometro MEMS analogico a 3 assi, capace di rilevare accelerazioni fino a ±200 g senza saturare; questo range di misura, unito all'uscita analogica che fornisce in tempo reale i dati di accelerazione, fanno dell'ADXL377 il dispositivo ideale per rilevare, in alcuni sport, urti che possono causare traumi cerebrali. La larghezza di banda di 1.600 Hz rende l'ADXL377 ideale per la misura delle sollecitazioni in ambiente industriale. Inoltre l'accelerometro non richiede l'allineamento né la disposizione di sensori ortogonali, consentendo di ridurre fino a 5 volte lo spazio richiesto sul circuito stampato rispetto all'uso di soluzioni a singolo asse. Tra le applicazioni troviamo l'IZOD per le vetture della Formula Indy 2012, dove ausilierà il sistema di sicurezza della guida: l'accelerometro di Analog Devices equipaggerà un sistema collocato negli alloggiamenti del casco dove si trovano gli auricolari, che verrà usato per misurare le sollecitazioni cui la testa del pilota è sottoposta in caso di incidenti, ma anche quelle derivanti dagli scuotimenti in gara.



Caratteristiche dell'ADXL377 :
- Rilevamento su 3-assi.
- Range di misura (g): ±200.
- Consumo: 300 μA (tipico).
- Alimentazione: 1,8÷3,6 V.
- Resistenza agli urti: 10.000 g.
- Largezza di banda: 1.600 Hz.

www.analog.com

Controller a diodo ideale LTC4359 da Linear Technology

Un controller a diodo ideale e ad alta tensione rappresenta una semplice alternativa ai diodi Schottky nelle applicazioni nei settori automotive, avionico e dell'energia solare. L'LTC4359 è un valido esempio: funziona tra 4 V e 80 V, ma sopporta tensioni di ingresso da -40 a 100 V senza danni. La sua corrente di esercizio è di soli 150 μA, mentre un ingresso di controllo dello spegnimento gli consente di funzionare in modalità shutdown a bassa corrente, richiedendo così solo 13 μA; inoltre il componente di Linear Technology può funzionare tra -40 e 125 °C. Queste funzionalità consentono al dispositivo di proteggere i carichi negli ambienti automotive in condizioni difficili, ad esempio durante il load-dump, il cold-crank, l'avviamento a salto tra due batterie, e in presenza di connessioni con inversione della polarità delle batterie. Se combinato con il surge stopper LT4363, l'LTC4359 offre una solida protezione front-end da sovratensione, sovracorrente e polarità inversa della batteria. Anche i sistemi ad energia solare traggono vantaggio dalla bassa corrente d'esercizio di questo dispositivo, perché l'LTC4359 può essere utilizzato per fornire un percorso a bassa perdita che isola il pannello solare dal carico.
Nella tipica applicazione, l'LTC4359 controlla un MOSFET a canale N variandone la caduta drain-source per garantire l'erogazione della corrente senza oscillazione, anche in condizioni di carico leggero.

www.linear.com/idealdiodes



I nuovi IC PowIRaudio di IR semplificano la progettazione degli amplificatori in classe D

International Rectifier, presenta la famiglia IR43xxM PowIraudio™ di integrati ad alte performance per sistemi audio come home theatre e car audio; si tratta di amplificatori in classe D che integrano un nuovo controller PWM e due MOSFET di potenza costituenti lo stadio di uscita. Ciò permette di realizzare amplificatori audio in classe D risparmiando componenti e spazio sul circuito stampato. La combinazione di un avanzato controller audio e dello stadio d'uscita a MOSFET, ottimizza le performance in fatto di rendimento, distorsione armonica THD e generazione di spurie (EMI). Le soluzioni IR in classe D offrono prestazioni equivalenti a quelle degli amplificatori di Classe AB. Tra le caratteristiche fondamentali della famiglia, la protezione da sovracorrente, lo shutdown in caso di eccesso di temperatura, il controllo esterno di spegnimento e l'ingresso differenziale fluttuante; l'IR4302M ha in più il rilevamento del clipping.

www.irf.com



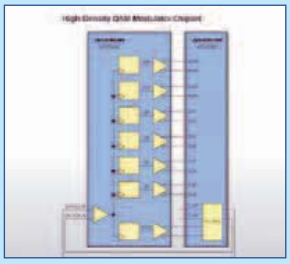
Nuovo driver per bobine di accensione da Fairchild

Per poter rispettare i rigidi requisiti in termini di emissioni e riduzione dei consumi nei motori a ciclo Otto, i progettisti del settore automobilistico hanno bisogno di driver per bobine di accensione dalle performance più elevate. Un aiuto in questo senso arriva dai nuovi IGBT Fairchild Semiconductor specifici per l'accensione elettronica, che vantano una dissipazione ridotta e dimensioni più contenute. I driver EcoSPARK 2, FGD3040G2 e FDG3440G2, garantiscono fino al 20 % di riduzione della tensione di saturazione (Vsat) senza significative perdite di energia SCIS (Self Clamping Inductive Switching); questo permette di ridurre la dissipazione di corrente, la necessità di dissipatori termici e la temperatura di giunzione operativa.
Il driver per bobine di accensione EcoSPARK 2 è disponibile anche in DPAK, che consente di sostituire componenti D2PAK riducendo gli ingombri a parità di Vsat. Entrambi i dispositivi supportano sistemi di accensione multi-spark ad alta tensione che permettono al motore di funzionare con miscele aria-carburante più magre.

www.fairchildsemi.com



Chipset QAM Modulator per la connettività domestica del futuro



Maxim ha annunciato un nuovo modulatore QAM scalabile ad alta densità e banda estesa, disponibile sotto forma di chipset formato dal Digital Up-Converter (DUC) MAX5880 e dal convertitore D/A RF (DAC RF) MAX5882. Una singola porta RF che utilizza il chipset MAX5880/MAX5882 può sostituire i numerosi box dei modulatori QAM necessari per erogare trasmissioni televisive, trasferimenti dati ad alta velocità e servizi VoIP triple-play, video on-demand, SVD e la futura IPTV interattiva.
Il DUC MAX5880 è in grado di interfacciarsi perfettamente con MAX5882; il modulatore QAM può sintetizzare fino a 128 canali QAM sull'intero spettro del cavo –da 50 MHz a 1GHz– in downstream.
www.maxim-ic.com

M95: Quectel, il modulo quadri-band più piccolo e facile da montare

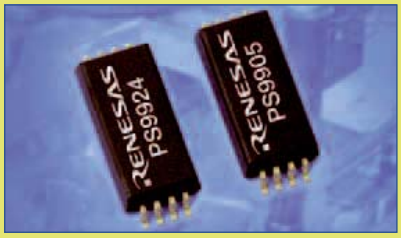
Recentemente Quectel ha lanciato sul mercato un nuovo modulo GSM/GPRS, il modello M95, che si distingue per le dimensioni (misura appena 19,9x23,6x2,65 mm) che ne fanno il più piccolo quadri-band GSM/GPRS al mondo, e per la facilità di montaggio, grazie al package scelto: l'LCC.
Oltre a tutte le funzioni che possono essere richieste ad un modulo GSM/GPRS, l'M95 garantisce un consumo di energia estremamente ridotto; il modulo è inoltre dotato di un amplificatore audio in classe AB per l'altoparlante, che fa



risparmiare spazio sul circuito stampato del cellulare o modem cui il dispositivo viene destinato.
Le applicazioni dell'M95 sono Vehicle Tracking System, Industrial PDA, Personal Tracking, Wireless POS, contatori intelligenti.
www.quectel.com

Da Renesas, la nuova famiglia PS99xx di fotoaccoppiatori ad alto isolamento

Renesas Electronics annuncia la disponibilità di nuovi fotoaccoppiatori a passo doppio per applicazioni industriali con tensioni fino a 690 V o impianti fotovoltaici con tensioni di lavoro fino a 1.000 V. La nuova famiglia PS99xx consta di due dispositivi: il PS9924 è un optoisolatore high-speed capace di spingersi fino a 10 Mbps e il PS9905 è un driver per IGBT accoppiato otticamente, capace di commutare 2,5 A ad alta velocità, presentando una bassa caduta di tensione, tipica dei MOSFET.
La caratteristica saliente dei dispositivi della famiglia PS99xx è il passo laterale dei case (entrambi a 8 pin) che è quasi doppio rispetto al DIP standard, ossia 14,5 mm; il nuovo package, detto Long-SDIP (LSDIP) ha un passo verticale di appena 1,27 mm, però per aumentare l'isolamento tra ingresso e uscita, tutti i pin di input sono da un lato e quelli di uscita dall'altro. La distanza di 14,5 mm tra i gruppi di piedini garantisce un'elevatissima tensione d'isolamento, pari a 7.500 Vrms in regime sinusoidale.
I nuovi dispositivi sono conformi alle norme internazionali di sicurezza, come UL, CSA e VDE. Per garantire un funzionamento sicuro nell'ambiente industriale, fortemente disturbato, il dispositivo Renesas incorpora una protezione che fornisce una reiezione di modo comune transitoria (CMR) di almeno 15 kV/ms (tipicamente 20 kV/ms).
Il PS9924 presenta un consumo energetico tra i migliori nel settore, che lo rende ideale per applicazioni che richiedono sia alta tensione di isolamento che basso consumo energetico.
Quanto al PS9905, è un optoisolatore con uscita a MOSFET progettato per controllare in modo efficace e sicuro degli IGBT; dispo-



ne di un LED d'ingresso (GaAlAs) mentre nello stadio di uscita ha un fotodiodo la cui tensione viene rilevata da un circuito di signal processing, il cui segnale viene poi amplificato in potenza da circuiti che permettono alte velocità di commutazione e l'erogazione di ben 2,5 A. Il massimo tempo di propagazione sia nella commutazione low/high, sia in high/low, è dell'ordine di 0,15µs entro tutto il range di temperatura da -40 a +110 °C. La reiezione dei transienti in modo comune è di 25 kV/µs, il che permette il trasferimento dei segnali di controllo senza errori.
www.renesas.eu

NXP lancia i suoi nuovi Digital Wideband VGA ad alta linearità e basso consumo

BGA7204 e BGA7210 sono due amplificatori digitali a guadagno variabile (VGA) che offrono alta linearità, basso rumore e consumi ridotti entro un ampio spettro di frequenze di funzionamento. Il BGA7210 è l'unico dispositivo nel suo genere capace di funzionare fino a 4 GHz senza compensazione esterna. Entrambi i VGA implementano la disposizione dei pin "smart routing" di NXP, che semplifica la progettazione del circuito stampato e permette di ridurre del 25% lo spazio richiesto.
Progettati principalmente per



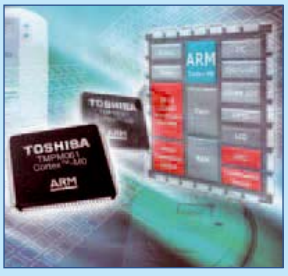
essere impiegati nelle infrastrutture wireless, i nuovi VGA di NXP garantiscono che il segnale entrante abbia sempre l'ampiezza ottimale per essere identificato anche in presenza di degradazioni causate dai cambiamenti di temperatura. Controllando il livello di amplificazione dell'amplificatore di potenza, il VGA aiuta anche a compensare le variazioni di carico delle celle del sistema wireless man mano che invecchiano. I BGA7204 e BGA7210 presentano il più basso consumo di energia rispetto ai dispositivi concorrenti, grazie a qualità come una modalità power-save proprietaria che può effettivamente ridurre fino al 45 % il consumo di corrente nei sistemi TDD. Il BGA7210, implementa una ripartizione variabile della corrente nei due amplificatori che lo compongono, in funzione dello stato di attenuazione, allo scopo di risparmiare energia.
www.nxp.com

IGBT Field Stop a 1.200 V alta efficienza da ON Semiconductor



ON Semiconductor presenta i suoi nuovi IGBT Field Stop per la conversione di potenza ad alta efficienza, ideali per sistemi di riscaldamento e cottura a induzione, controllo motori e inverter. I componenti, siglati NGTB15N120, NGTB20N120 e NGTB25N120, sopportano una tensione collettore-emettitore di 1.200 volt (V) e sono realizzati con la tecnologia Deep Trench utilizzando wafer molto sottili e altre tecniche costruttive che consentono di ottenere bassissime perdite di turn-off e di richiedere ridotte tensioni di gate in stato di ON; ciò permette di minimizzare le perdite e i tempi di commutazione.
I dispositivi sono incapsulati in contenitori TO-247, sono disponibili con correnti di ON di 15, 20 e 25 A e integrano diodi a bassissima caduta di tensione diretta e basso tempo di ripristino, per venire incontro alle esigenze tipiche di quelle applicazioni in cui ridurre i tempi di commutazione e con essi le perdite, è determinante, soprattutto oggi che in tutte le applicazioni, non ultime quelle elettriche ed elettroniche, risparmiare energia è di vitale importanza.
www.onsemi.com

TMPM061, un micro ARM Cortex™ - M0 per contatori elettronici



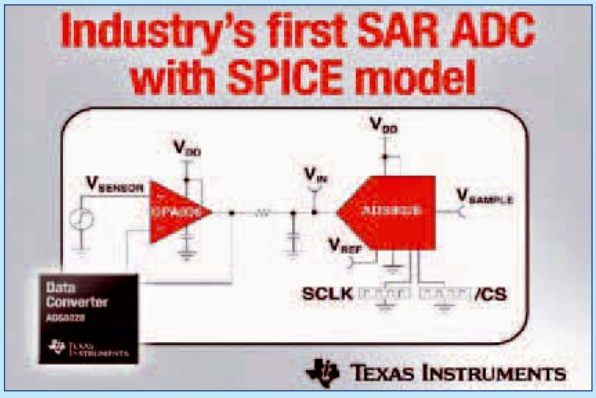
Toshiba presenta un microcontrollore RISC a 32 bit basato sul core del processore ARM Cortex™-M0: si chiama TMPM061 e diversamente dai dispositivi tradizionalmente impiegati nei contatori elettronici (che usano un chip per lo stadio d'ingresso analogico e un processore), è una soluzione single chip. Le ridottissime dimensioni, il basso consumo e la possibilità di programmarlo con poche righe di codice, rendono il core ARM, Cortex-M0 particolarmente adatto all'integrazione nel dispositivo di Toshiba, dove l'esecuzione dei calcoli relativi alla potenza elettrica sono delegati ad un motore di calcolo DSP integrato.
Quest'ultimo può determinare la potenza attiva, quella reattiva e il fattore di potenza, nonché tenere sotto controllo le variazioni di tensione e frequenza di rete. Oltre al motore per il calcolo della potenza elettrica, il circuito integrato TMPM061 offre diverse altre funzionalità che minimizzano il numero di componenti necessari per realizzare un contatore elettronico.
Tra le periferiche integrate vi sono un convertitore analogico digitale di precisione da 24 bit a tre canali di tipo Delta-Sigma ($\Delta\Sigma$), un convertitore digitale analogico da 10 bit e un RTC con compensazione della temperatura. Il TMPM061FWFG viene fornito in contenitore LQFP da 100 piedini da 14x14 mm e funziona con tensioni di alimentazione comprese tra 1,8 V e 3,6 V.
www.toshiba-components.com

Nuovi TFT da Sharp



Con i modelli LQ150X1LW12 e LQ150X1LG11, Sharp amplia la sua gamma di LCD TFT industriali con diagonale di 15 pollici (38 cm). Gli schermi hanno contrasto rispettivamente di 1.000:1 e 800:1 e supportano 16,8 milioni di colori. Il modello LQ150X1LW12 ha un angolo di visione di 170° in tutte le direzioni, che lo rende ideale nelle applicazioni in cui bisogna mostrare immagini a più persone. Il modello LQ150X1LG11, grazie al suo design e all'elevato livello di luminosità di 600 cd/m^2, può essere utilizzato in applicazioni quali sistemi di chioschi elettronici, macchine distributrici e ATM, ed interfacce HMI di macchinari e dispositivi esterni usati in aree molto luminose.
Entrambi i TFT hanno la retroilluminazione LED con un ciclo di vita di almeno 50.000 ore, pilotata da circuiti PWM che consentono di adattare la luce di fondo alle condizioni ambientali a tutto vantaggio del consumo di elettricità.
www.sharpsme.com

TI presenta il primo ADC SAR del settore con modello SPICE



Texas Instruments presenta un convertitore da analogico a digitale (ADC) successive-approximation-register (SAR) di cui fornisce il modello SPICE TINA-TI™ scaricabile da web, che consente per la prima volta ai progettisti di sistemi di simulare e caratterizzare col software l'intera catena del segnale analogico. Il modello è disponibile per il nuovo ADS8028 a 8 canali, 1 MSPS e 12 bit, un ADC SAR altamente integrato dotato di riferimento della tensione interna a bassa deriva, ampio intervallo della tensione analogica di riferimento esterna, ampi intervalli di alimentazione analogica e digitale e di un sensore termico interno.
In passato i progettisti potevano soltanto eseguire simulazioni, modellazioni e test fino all'ADC SAR; ora, con un modello SPICE TINA-TI, scaricabile da web, disponibile per tutti i nuovi ADC SAR, i progettisti potranno verificare integralmente la circuiteria di comando ADC, in modo da velocizzare lo sviluppo dei prodotti. Per l'ADS8028 TI mette a disposizione il modulo di valutazione (EVM) e lo schematico di riferimento TINA-TI, che includono l'amplificatore driver OPA836 per velocizzare la simulazione software e la valutazione hardware.
www.ti.com

NIWeek 2012



Torna anche quest'anno ad Austin (Texas, U.S.A.) NIWeek, la conferenza globale sulla progettazione grafica di sistemi, che per quattro giorni -dal 6 al 9 agosto- riunisce più di 3.000 progettisti, sviluppatori e personalità del mondo scientifico, affrontando temi come l'automotive, le telecomunicazioni, la robotica e l'energia. Un'occasione unica per imparare come si può accelerare la produttività e l'innovazione con le soluzioni di National Instruments.
Ad NIWeek si parlerà di nuove tecnologie la cui conoscenza e padronanza sono oggi fondamentali per misurarsi con la concorrenza nello sviluppo di sistemi e software destinati alla misura ed al controllo di processi. Attraverso sessioni interattive tenute dal settore Research and Development di National Instruments e da docenti ospiti, ma anche tramite incontri mirati con il settore dell'industria, workshop ed esposizioni, i partecipanti potranno conoscere e toccare con mano i più recenti progressi nella progettazione, la ricerca ed il test. Sessioni specifiche mostreranno come i nuovi strumenti di progettazione grafica di NI, integrati con le nuove tecnologie embedded hardware e software di NI, permettano di incrementare la produttività nella progettazione e realizzazione di sistemi embedded. Ad NIWeek si approfondiranno gli ultimi progressi nei prodotti embedded come NI CompactRIO, LabVIEW FPGA e LabVIEW Real-Time, come nelle nuove tecnologie per interfacce uomo macchina, controllo del movimento e network di sensori wireless.
Non meno importanti, le sessioni dedicate all'ottimizzazione delle tecniche di sviluppo e di test per produrre applicazioni di acquisizione dati di maggior qualità e affidabilità.
A sottolineare l'importanza dell'evento, il fatto che nelle passate edizioni hanno preso la parola personalità come il Dr. Neil Gershenfeld del MIT Media, il Dr. Michiu Kaku e Tim Samaras di Discovery Channel. Tra i partner delle precedenti edizioni, ingegneri, scienziati e dirigenti di realtà come Boeing, John Deere, CERN, ST Ericsson, UC Berkeley. Per quest'anno, l'intervento di chiusura sarà del professor Thomas R. Kurfess della Clemson University, che è stato all'Ufficio delle politiche scientifiche e tecnologiche della Casa Bianca ed ha condotto ricerche sulla progettazione e lo sviluppo di sistemi avanzati rivolti al settore automobilistico.
www.ni.com/niweek



Flash Memory Summit



A Santa Clara (U.S.A.) nella terra di Intel, dal 21 al 23 agosto prossimi si terrà Flash Memory Summit, la prima ed unica conferenza dedicata esclusivamente al settore delle flash memory ed alle loro applicazioni.
L'evento è rivolto a chi si occupa di applicazioni consumer, come telefoni cellulari, fotocamere digitali e lettori musicali, PDA, PC palmari e videogiochi, ed è anche utile per il settore del Personal Computing, ma anche per quello industriale, dei sistemi di comunicazione, delle applicazioni militari e dei sistemi di difesa, del controllo di processo, delle apparecchiature di prova e della strumentazione. Accanto alle conferenze, in programma per tutti e tre i giorni di durata dell'evento, nei soli giorni 22 e 23 agosto un'area espositiva mostrerà le ultime novità esposte dai principali produttori.
Nella giornata del 21 agosto si parlerà di metodi di progettazione, software e controller per flash memory, applicazioni consumer ed embedded, sistemi ibridi (disco o DRAM) sicurezza dei contenuti, test, dispositivi come Flash disk e Memory Card. Si prevede che il mercato delle flash memory, nel 2016, dovrebbe arrivare a movimentare qualcosa come 31 miliardi di dollari.



www.flashmemorysummit.com

ICQNM 2012



Conferenza Internazionale sulle tecnologie nano, micro e quantiche. Quest'anno la Sesta Conferenza ICQNM organizzata da IARIA, International Academy, Research and Industry Association, si terrà a Roma dal 19 al 26 Agosto. La conferenza è finalizzata a raccogliere e presentare risultati della ricerca scientifica ed applicazioni in ambito industriale nei diversi settori di sviluppo dell'immenso panorama delle tecnologie nano, micro e quantiche. Si parlerà di computer, sistemi di comunicazione, di sicurezza, di crittografia e di autenticazione in tecnologia quantica. Nel campo delle nanotecnologie si approfondiranno gli sviluppi ed applicazioni su materiali, strutture, dispositivi elettronici, robotica così come di metodi e strumenti di progettazione e sviluppo, di affidabilità dei sistemi in nanotecnologia e di standard. Non mancheranno contributi nel campo delle microtecnologie, micro sensori e dispositivi fabbricati in tecnologia micro, strumenti di fabbricazione e controllo. Completeranno il panorama delle presentazioni tutte le possibili applicazioni nel campo della bio-nanotecnologia, bio-nano robotica, nano-tubi e nano-particelle, sistemi di fabbricazione nano, autonomi e teleoperati e tutti i filoni emergenti nelle applicazione dei superconduttori, nano fluidi, sistemi nano-ottici, molecolari e cellulari.
www.iaria.org/conferences2012/ICQNM12.html

Microchip MASTERs WORLDWIDE 2012



Giunta alla sedicesima edizione, MASTERs, la manifestazione promossa da Microchip, è diventata la principale conferenza di formazione tecnica per i progettisti di sistemi di controllo embedded, che quest'anno si estende a sei Paesi e fornisce corsi di formazione a tutti i livelli tecnici.
I progettisti possono scegliere tra corsi sui prodotti Microchip o su argomenti generali, quali il controllo di motori, la progettazione della sezione di alimentazione, il controllo dell'illuminazione, i protocolli di comunicazione USB e TCP / IP, la programmazione in C, le tecnologie di visualizzazione grafica, i metodi sensoriali tattili e la progettazione di sistemi analogici. La maggior parte di questi argomenti verrà esposta ai partecipanti dagli stessi sviluppatori e progettisti che hanno creato i prodotti e i dispositivi di cui si parlerà.
Verranno presentate e utilizzate nelle aule anche le application-notes e le demoboard. MASTERs 2012 si svolgerà in sette nazioni in tutto il mondo: partendo da Phoenix (Arizona, U.S.A.), quindi approderà a Seoul, per concludersi in India. Gli argomenti di base sono gli stessi per tutti gli eventi, mentre da una località all'altra potranno cambiare le applicazioni didattiche. Le date comunicate da Microchip per la MASTERs WORLDWIDE Conference sono le seguenti.
- Phoenix, Arizona, 8-11 agosto.
- Seoul, Corea, 11-12 ottobre.
- Chengdu, Cina, 31 ottobre - 2 novembre.
- Shanghai, Cina, 7-9 novembre.
- Beijing, Cina, 28-30 novembre.
- Taipei, Taiwan, 21-23 novembre.
- Bangalore, India, 29-30 novembre.
- Pune, India, 3-4 dicembre.
- Nuova Delhi, India, 7-8 dicembre.

Tutti i dettagli del ciclo di conferenze si trovano alla pagina web del sito di Microchip dedicata all'evento.
http://techtrain.microchip.com

Sensors & Process Instrumentation 2012

SPI SENSORS & PROCESS INSTRUMENTATION 1ª edizione

Martedì, 26 giugno 2012, presso il Centro Congressi Cosmo Hotel Palace a Cinisello Balsamo (MI), Fiera Milano Media presenta un evento unico e innovativo dedicato alla sensoristica e alla strumentazione di processo.
La mostra-convegno proporrà un'esclusiva vetrina di prodotti, sistemi e soluzioni che trovano applicazione negli impianti e macchine per industria manifatturiera e di processo: sensori, trasduttori, trasmettitori, indicatori, valvole di regolazione, registratori, regolatori, controllori, attuatori, strumenti e sistemi completi per la misura di tutti i fondamentali parametri di processo: temperatura, pressione, livello, flusso, portata, umidità, forza, peso ecc.
Accanto alle soluzioni per le misure di processo, uno spazio speciale sarà dedicato alla strumentazione per l'analisi e il laboratorio: apparecchiature e strumenti per determinare il pH, la torbidità, la conducibilità, la presenza di ossigeno, analizzatori di TOC, titolatori, gascromatografi, spettrofotometri, densimetri e viscosimetri, campionatori.
www.mostreconvegno.it/sepi/home.asp

MCU Solution Summit 2012

SILICA An Avnet Company

Silica, società del gruppo Avnet, ha annunciato la più importante e vasta serie di eventi di formazione in Europa mai realizzata, con training specifici in 20 città Europee; un vero e proprio tour che vedrà la partecipazione di esperti dei principali player del settore, ossia i più importanti produttori di Microcontrollori e di Microprocessori quali Cypress, Freescale, Infineon, Intel, Microchip, NXP, Renesas, STMicroelectronics, Texas Instruments, ma anche di ARM. In programma da giugno ad ottobre, MCU Solution Summit 2012 è un'iniziativa chiave per la formazione nell'ambito della strategia Core 'n More di Silica, che mira ad aiutare i clienti a realizzare applicazioni in modo semplice ed efficace, attraverso la scelta più ampia possibile di microcontrollori presenti sul mercato e con le migliori risorse tecniche di supporto.
Diversamente dai seminari incentrati sull'architettura ARM e sulle architetture proprietarie MCU e basate su processore, l'MCU Solution Summit" sarà orientato alle applicazioni piuttosto che essere una pura descrizione delle caratteristiche dei prodotti.
Ogni seminario dell'MCU Solution Summit, della durata di un giorno, prevede fino a sedici corsi di formazione della durata di settantacinque minuti. I partecipanti potranno scegliere quattro corsi componendo così la propria giornata ideale, in base alle applicazioni ed alle architetture che più li interessano.
I principali argomenti trattati nel corso del Summit saranno Embedded connectivity, Motion Control, Home Automation/HMI, Safety, Medical, Video/Audio Broadcast, Wireless/Sensors Lighting, Energy/Metering. Grazie ad un'area espositiva, i partecipanti avranno inoltre accesso ad esposizioni dimostrative dei partner, dei fornitori di MCU e dei numerosi fornitori di tool e di software (che possono variare da una sede all'altra), i quali collaboreranno con Silica durante l'evento "MCU Solution Summit".
I lavori del summit avranno inizio a Milano il 19 Giugno, seguiti da eventi in Germania, Italia, Regno Unito, Francia, Spagna, Benelux e Paesi Nordici; il programma completo e le informazioni sulla registrazione sono disponibili alla pagina web www.silica.com/mcu-summit.
Di seguito, le quattro date italiane:
- 19/06/2012 a Milano;
- 26/06/2012 a Torino;
- 03/07/2012 a Bologna;
- 10/07/2012 a Padova.

www.silica.com/mcu-summit



Sapete spostare una pallina da tennis o da golf in un labirinto circolare inclinando opportunamente la base che lo sostiene? Ma soprattutto, sapreste farlo controllando i movimenti del labirinto con il vostro corpo, ondeggiando analogamente in piedi su una pedana mobile? Ciò è quanto vi proponiamo di fare con questa nuova versione del labirinto con cui giocavamo da piccoli, solo che mentre quello si teneva in mano e inclinandolo facevamo in modo da far scivolare la pallina fino al centro, con questo progetto dobbiamo muovere il nostro corpo per indicare al labirinto (nel nostro caso circolare, di 1 metro di diametro) come muoversi. Una prova di abilità ed anche un gioco divertente: dipende da come lo interpretate.

Ma come funziona in pratica questo particolare gadget? Semplice: la pedana su cui ci muoveremo è un sensore, mentre la piastra oscillante che muoverà di conseguenza il labirinto è l'attuatore; sotto la pedana, sulla sua circonferenza, abbiamo posizionato un accelerometro a tre assi, in modo da rilevare i movimenti in almeno due direzioni. Il segnale ricavato dall'accelerometro viene inviato ad una board Arduino tramite uno shield che abbiamo realizzato allo scopo e che è lo stesso con cui faremo il controllo dei motori dell'azionamento del labirinto.

Quest'ultimo viene mosso da una meccanica formata da un basamento a croce su cui sono montati due stepper-motor disposti ortogonalmente tra loro, ognuno dei quali, mediante un cavo di acciaio e due carrucole, tira da un lato e libera da quello opposto la base del labirinto; i due motori vengono comandati dallo shield in base ai segnali che Arduino fornisce, i quali dipendono dall'inclinazione della pedana su cui ci muoviamo.

Ma chi assicura che l'inclinazione del labirinto corrisponderà sempre a quella della pedana? Per garantire ciò abbiamo previsto una retroazione che indichi ad Arduino di quanto è inclinato il labirinto e confronti l'angolo corrispondente con l'inclinazione rilevata dall'accelerometro posto sulla pedana, correggendo

SWINGING LABYRINTH

di BORIS LANDONI

Applicazione didattica o gioco da luna-park, questo sistema è indubbiamente un doppio esercizio di abilità, che vi metterà alla prova prima nel realizzarlo e poi nell'usarlo.

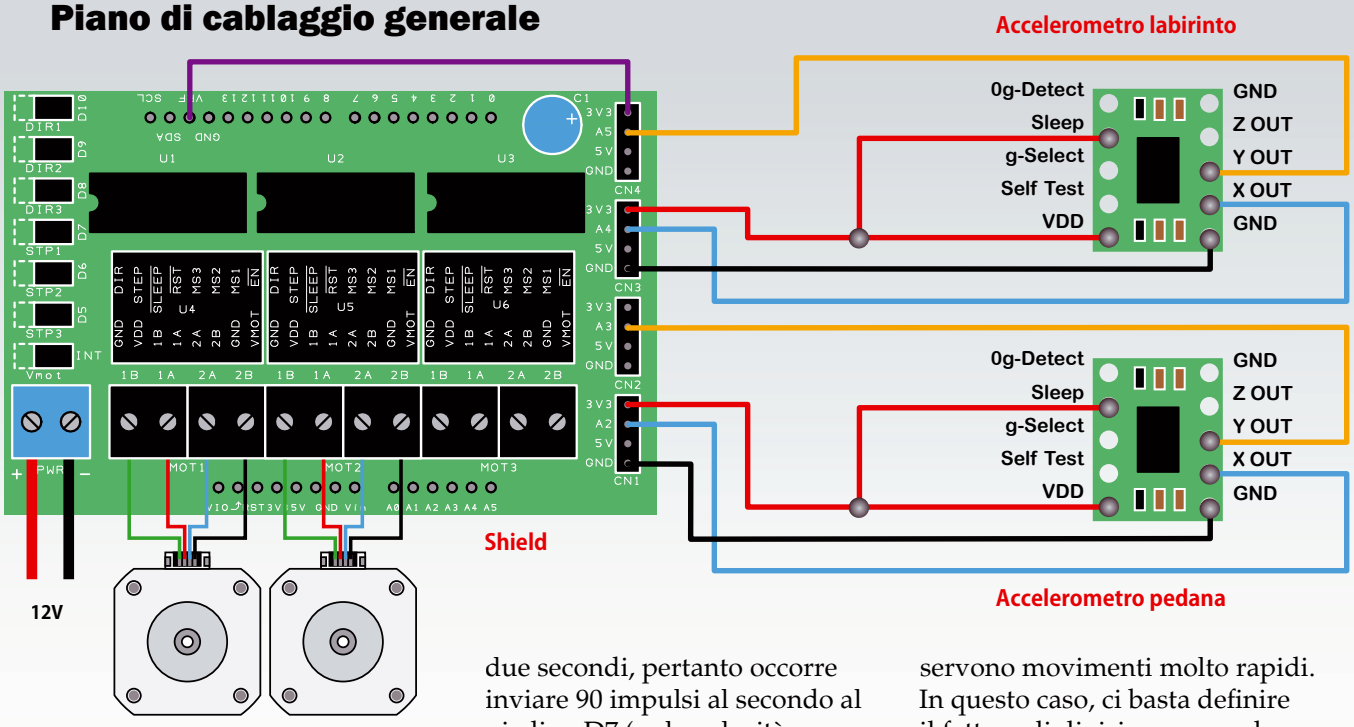


l'angolazione del labirinto fino a pareggiarla con quella della pedana. La retroazione è ottenuta con un secondo accelerometro posto sotto il labirinto, uguale al primo, posto sotto la pedana e orientato nello stesso modo, rispetto alla direzione attesa. Ecco, più o meno questo è il sistema. Abbiamo quindi un Arduino, uno shield, due accelerometri, altrettanti motori passo-passo e infine un firmware da caricare in Arduino per gestire il tutto.
Partiamo dunque dalla descrizione dello shield, del quale trovate lo schema elettrico nelle pagine seguenti.

LO SHIELD

Si tratta dell'interfaccia mediante cui la scheda Arduino (lo shield è compatibile con UNO e Mega) acquisisce lo stato dei due accelerometri e comanda i due motori passo-passo che muovono il labirinto.
Il controllo dei motori si realizza tramite un driver per ciascuno, anche se in realtà il nostro shield ha 3 driver, dei quali uno è per futuri sviluppi e non va usato. I driver, prodotti dalla Pololu, nello schema elettrico sono siglati U4, U5 e U6. La particolarità di questi moduli, ognuno dei quali contiene un doppio ponte ad H a MOSFET, è che possono essere impostati per gestire sia la direzione, sia il numero di gradi che il rotore del motore deve compiere alla ricezione di ogni comando; in altre parole, possiamo decidere se quando lo comandiamo, il modulo deve far ruotare l'albero di uno step alla volta, oppure di 1/2, 1/4, 1/8 o 1/16, in base all'accuratezza che si desidera ottenere.
In virtù di ciò, il comando del movimento può essere ottenuto in

Piano di cablaggio generale



due modi: nel primo, Arduino gestisce, tramite due linee digitali dedicate, la direzione e lo step del motore agendo direttamente sui pin DIR e STEP del driver Pololu; nella seconda Arduino si appoggia agli shift-register per impostare il verso di rotazione e inviare impulsi che determinano rotazioni di uno step o frazione di esso, a seconda dell'impostazione fatta all'inizializzazione e conservata da opportuni shift-register, che sono i componenti siglati U1, U2 e U3 nello schema elettrico.

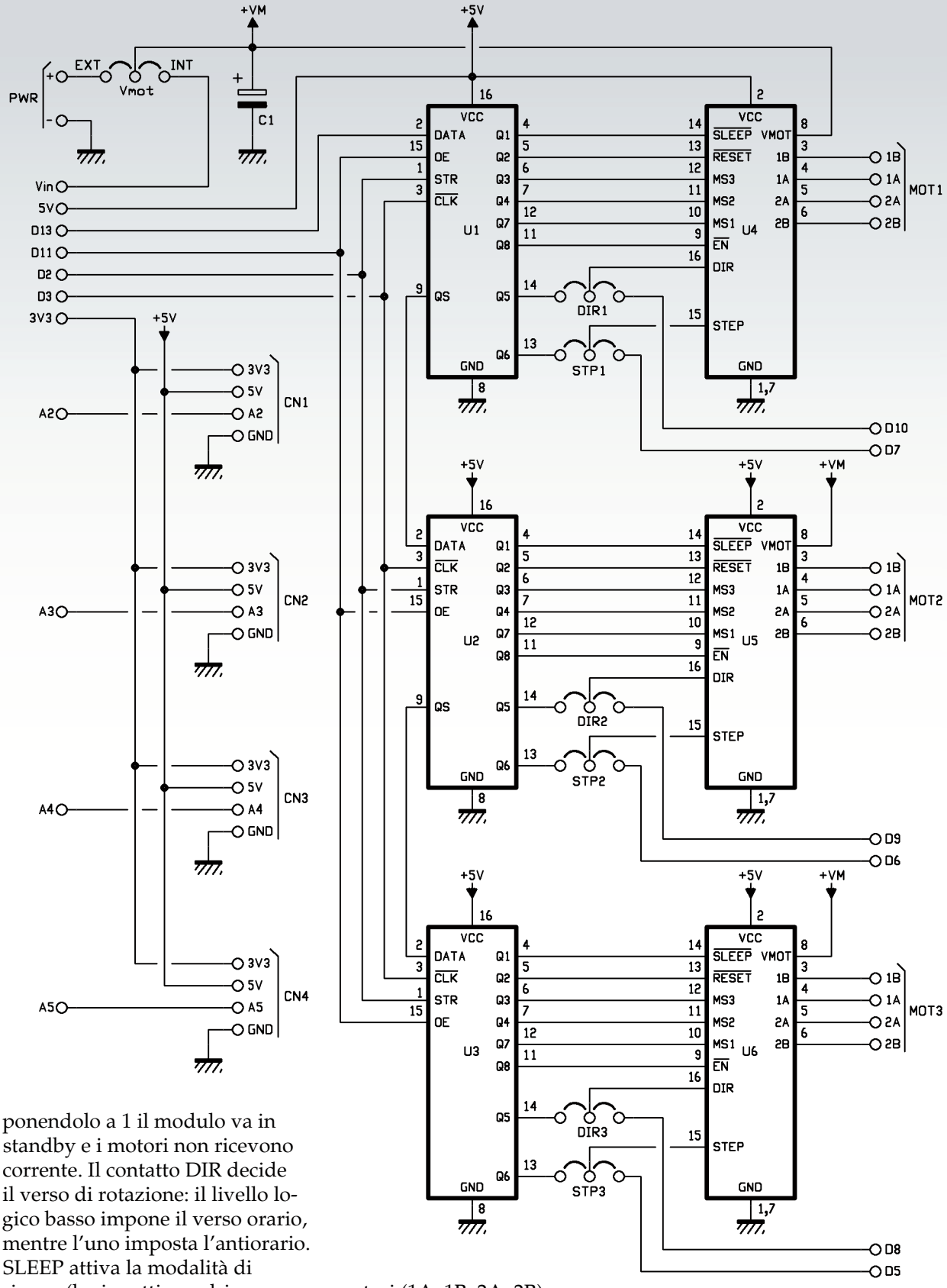
La differenza tra le due modalità è che nella prima Arduino aggiorna direttamente i pin DIR e STEP dei driver dei motori passo-passo. Questo modo di funzionamento si ottiene spostando i jumper DIR ed STP (quindi DIR1/STP1, DIR2/STP2, DIR3/STP3) verso le linee di Arduino; ipotizzando che il motore collegato all'uscita MOT1, sia un 360 passi, il suo albero deve ruotare di 180° in due secondi, pertanto occorre inviare 90 impulsi al secondo al piedino D7 (se la velocità angolare deve aumentare, ad esempio per far rispondere il labirinto più velocemente, la frequenza degli impulsi di step deve crescere di conseguenza). Prima, però, Arduino deve impostare anche la direzione con l'opportuno livello logico sul piedino D10, mediante il jumper DIR.

Nella seconda modalità, ci avvaliamo degli shift-register per gestire anche la direzione e lo step del driver Pololu; questa modalità si ottiene spostando i jumper DIR e STEP verso i pin Q5 e Q6 dei rispettivi shift register, ma è da tenere in considerazione che i tempi di risposta sono più lenti rispetto a quelli della prima modalità e quindi non idonei al nostro scopo, poichè servono movimenti molto rapidi. In questo caso, ci basta definire il fattore di divisione una sola volta e poi, per variare la velocità di risposta, giochiamo sulla frequenza degli impulsi generati da Arduino.

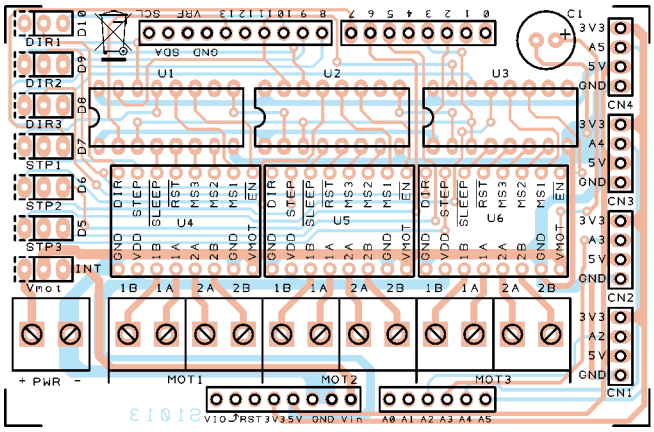


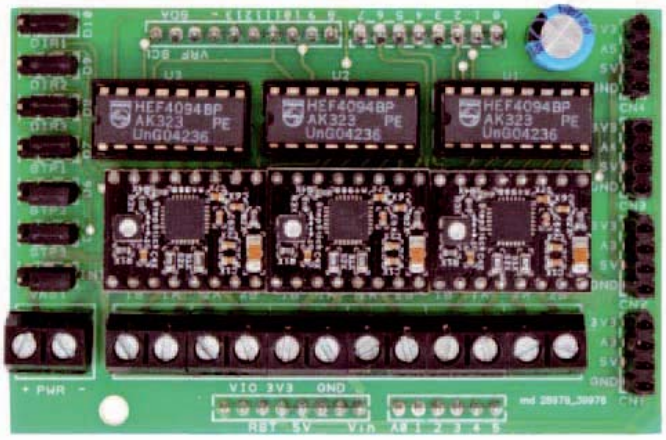
Analizziamo dunque questa modalità, per comprendere la quale dobbiamo spendere qualche altra parola sul modulo controller per stepper-motor prodotto dalla Pololu, ovvero sulla sua interfaccia di comando. Il modulo, siglato MD09B, consta di un doppio ponte ad H governato da un'elettronica che permette di impostare il verso di rotazione del campo elettromagnetico e quindi dell'albero dello stepper-motor, oltre a frazionare gli step. Ogni volta che sul piedino STEP giunge un impulso (la durata minima ammissibile è 1 µs) a livello alto, a meno di diversa impostazione, le uscite 1A, 1B, 2A e 2B forniscono gli impulsi per comandare lo spostamento del rotore del motore di uno step. Ciò vale se gli ingressi MS1, MS2, MS3 sono tutti a livello basso; in caso contrario, si ottiene la rotazione di 1/2, 1/4 ecc.

Affinché il controller funzioni correttamente, bisogna che il contatto EN sia a livello basso;



ponendolo a 1 il modulo va in standby e i motori non ricevono corrente. Il contatto DIR decide il verso di rotazione: il livello logico basso impone il verso orario, mentre l'uno imposta l'antiorario. SLEEP attiva la modalità di riposo (logica attiva e driver spenti) mentre RESET azzera il controllore che governa i driver e pone le uscite di comando dei motori (1A, 1B, 2A, 2B) a zero logico anche se STEP continua a ricevere impulsi. La logica è TTL-compatibile, comunque il modulo accetta sugli ingressi di controllo tensioni fino a 0,7 volte quella applicata tra il contatto 2





Elenco Componenti Shield:

C1: 220 µF 25 VL elettrolitico
U1÷U3: CD4094
U4÷U6: Driver motori (Pololu MD09B)

Varie:
- Morsetto 2 poli (7 pz.)
- Zoccolo 8+8 (3 pz.)
- Strip maschio 3 poli (7 pz.)
- Strip maschio 4 poli (4 pz.)
- Strip maschio 6 poli
- Strip maschio 8 poli (2 pz.)
- Strip maschio 10 poli
- Jumper (7 pz.)
- Circuito stampato

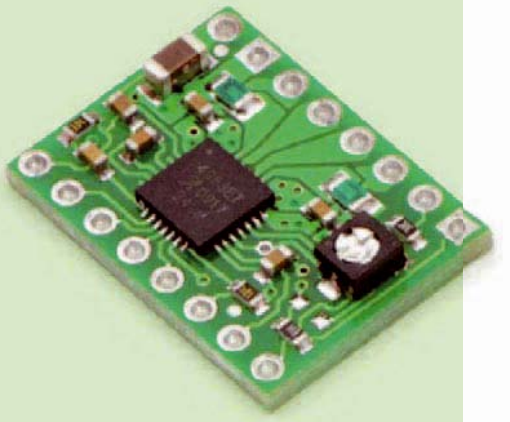
(Vcc) e massa (GND); il livello basso non deve superare 0,3xVcc. Detto ciò, vediamo che per impostare il fattore di divisione degli step, Arduino deve definire la combinazione delle linee MS1, MS2, MS3 di ogni modulo; per farlo distintamente per ciascun modulo e liberarsi del problema, nello shield sono previsti tre shift-register che caricano l'impostazione di ciascun controller individualmente e sequenzialmente. Una volta configurati i controller, le impostazioni possono rimanere memorizzate fin quando non serve modificarle e comunque non bisogna aggiornarle ad ogni impulso di comando di STEP; è questo sgrava Arduino di parte del lavoro. L'aggiornamento delle impostazioni dei controller si effettua caricando sequenzialmente i dati tramite la linea D13 di Arduino, che finisce in parallelo sui pin 2 (DATA) dei registri a scorrimento; ogni volta che è stata inviata la configurazione di tutti e tre i controller, Arduino attiva la linea di STROBE (piedini 1 di U1, U2, U3, ovvero I/O D2 della Arduino) e trasferisce sugli MD09B i rispettivi stati, quindi riporta D2 a zero logico.
Insieme alla configurazione degli MS1, MS2, MS3 (che può essere diversa da U1 a U3) i dati inviati da Arduino riguardano lo stato di SLEEP e RESET, che sono impostati uguali per tutti i moduli e valgono 1 logico fisso.
Ogni controller alimenta direttamente il motore e viene a sua volta alimentato da una linea comune che, tramite il jumper siglato Vmot, può prendere tensione da Arduino o da un alimentatore esterno mediante l'apposita morsettiera PWR; nel nostro caso conviene optare per quest'ultima soluzione, in quanto i motori assorbono più corrente di quella che Arduino potrebbe erogare. Diciamo che un alimentatore da 12 V e 2 A dovrebbe bastare.

GLI ACCELEROMETRI

Per collegare gli accelerometri sono stati previsti appositi connettori (CN1, CN2, CN3, CN4) che portano alle linee A2, A3, A4 e A5; per come è stato scritto il firmware, ai primi due bisogna collegare le uscite degli assi rispettivamente X e Y dell'accelerometro posto sotto la pedana e ad A4 e A5 le uscite degli assi rispettivamente X ed Y dell'accelerometro montato sotto al labirinto. Ogni connettore veicola il segnale analogico al rispettivo input analogico di Arduino e porta l'alimentazione a 5 V e a 3,3 V (prelevate dalle linee 5V e 3V3 di Arduino). Notate che l'accelerometro qui utilizzato funziona a 3,3 V, quindi nella nostra applicazione non colle-

ghiamo alcun filo al contatto 5V dei CN1÷CN4. Le linee di I/O usate sono analogiche perché gli accelerometri forniscono in uscita un segnale analogico per ciascun asse. Arduino elabora i segnali e ne trae le informazioni sull'inclinazione assunta dalla pedana e dal labirinto oscillante, comandando di conseguenza i motori passo-passo se le due differiscono tra loro.
Per quanto il sistema sia ben retroazionato, il software veloce e la risposta istantanea, in qualche condizione limite il movimento del labirinto potrebbe non seguire con la stessa rapidità quello della pedana; ciò è dovuto alla coppia dei motori, che in certi casi (ad esempio l'inversione del movimento in forte accelerazione) non basta. A ciò si può rimediare alleggerendo il labirinto, scegliendo motori più potenti o, se questi sono sufficienti, incrementando la corrente erogata dai driver MD09B.

Il driver per motori **MD09B**



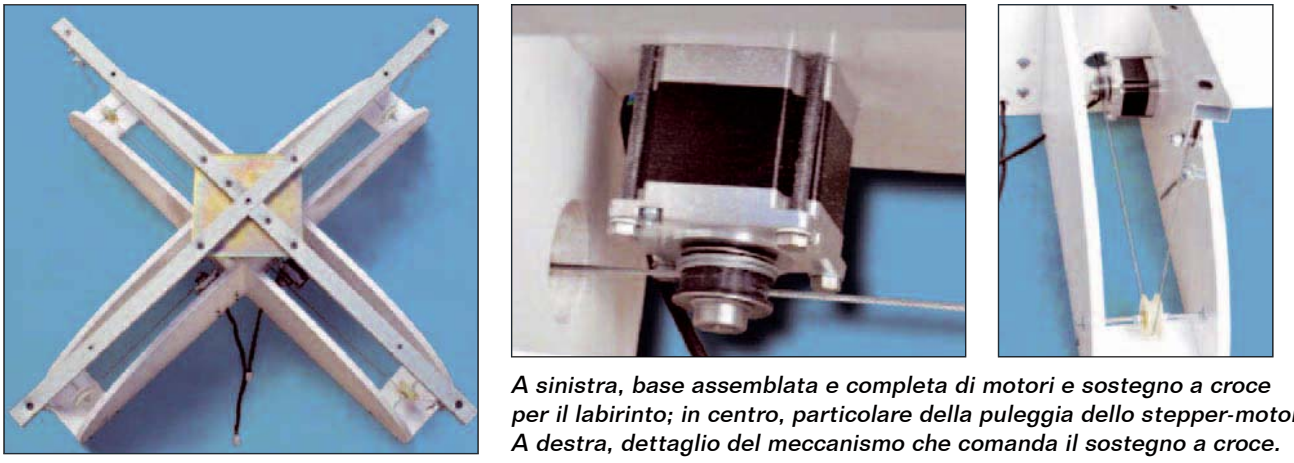
Il controllo dei motori passo-passo è operato da moduli Pololu (*www.pololu.com*) basati sull'integrato A4983 della Allegro e dotati ciascuno di un doppio driver a ponte H governato da una logica che controlla l'accensione e permette di far compiere al motore passi ridotti rispetto a quelli previsti dalla sua parte elettromeccanica; ad esempio, per ogni impulso di comando inviato a STEP (il contatto che riceve gli impulsi di azionamento del motore) lo stepper-motor può compiere un passo intero oppure frazioni da 1/2 a 1/6, a seconda della combinazione logica presentata ai piedini MS1, MS2, MS3, secondo la tabella riportata in questo riquadro. La direzione del movimento conseguente ad ogni impulso ricevuto sul contatto STEP dipende dal livello logico applicato all'ingresso DIR (1 logico forza il verso antiorario e 0 quello orario). Ogni ponte può erogare una corrente di 2 A, mentre il modulo si alimenta in continua con tensioni fino a 35 V; un piccolo trimmer posto sulla basetta e connesso al pin REF (17) dell'A4983, permette di regolare la corrente erogabile dai ponti.

MS1	MS2	MS3	Microstep Resolution	Excitation Mode
L	L	L	Full Step	2 Phase
H	L	L	Half Step	1-2 Phase
L	H	L	Quarter Step	W1-2 Phase
H	H	L	Eighth Step	2W1-2 Phase
H	H	H	Sixteenth Step	4W1-2 Phase

Impostazione degli step compiuti dal motore rispetto agli impulsi di comando.

LA COSTRUZIONE MECCANICA

Per realizzare il nostro labirinto oscillante occorre costruire una parte meccanica, che può essere assemblata a piacimento o seguendo le foto visibili in queste pagine e nel nostro blog; la meccanica consta di una base in cui si trova il fulcro del sostegno (è una croce di alluminio) del labirinto e che alloggia i motori, ognuno dei quali traina il sostegno mediante una corda d'acciaio che gira su due carrucole e si fissa alle estremità della croce. Il sostegno è imperniato sulla base mediante un raccordo a sfera e il labirinto si fissa su di essa mediante spine o bulloni. Per costruire la base e il labirinto seguite i disegni in queste pagine; ricordate che usando i nostri motori (STEPMOT02 della Futura Elettronica, *www.futurashop.it* oppure motori bipolari equivalenti



A sinistra, base assemblata e completa di motori e sostegno a croce per il labirinto; in centro, particolare della puleggia dello stepper-motor. A destra, dettaglio del meccanismo che comanda il sostegno a croce.

L'accelerometro **MMA7361L**

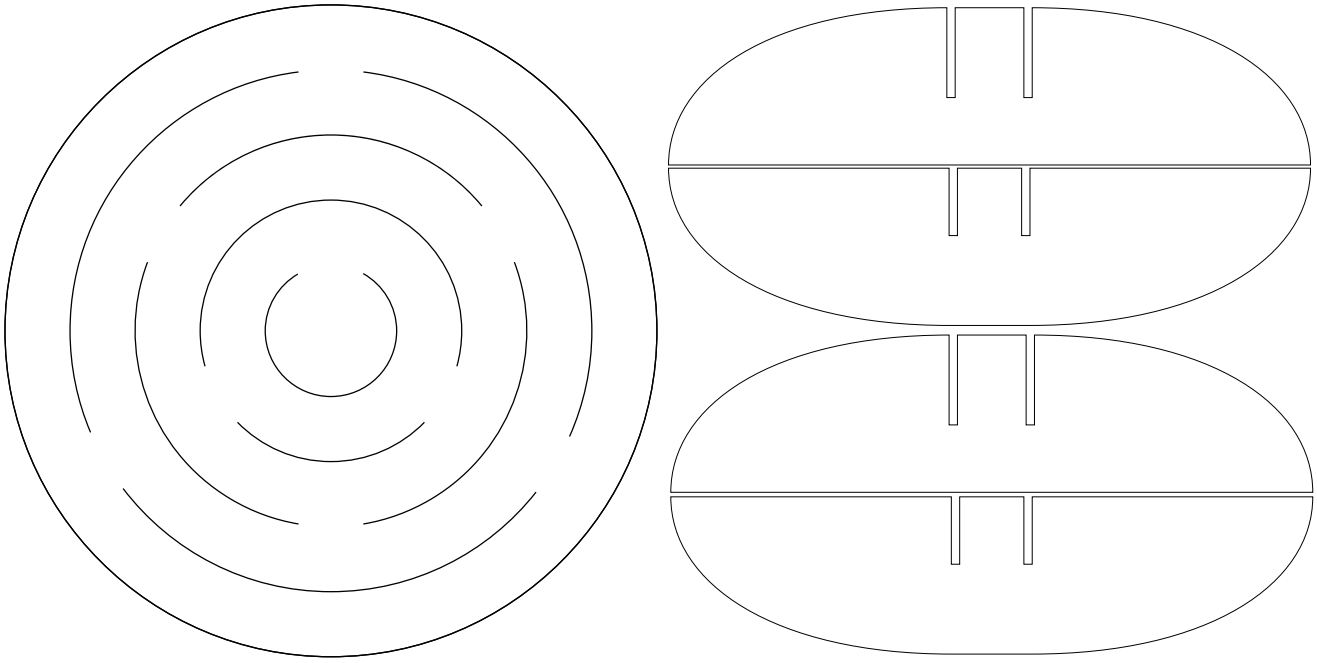


Il modulo accelerometro è un 3 assi della Pololu, basato sull'integrato MMA7361LC della Freescale; si tratta di un dispositivo MEMS in grado di fornire in uscita una tensione analogica corrispondente all'accelerazione rilevata su ognuno dei tre assi. La logica prevede un pin di SLEEP che, se posto a zero logico, porta il modulo al minimo assorbimento (<3 µA); nel normale utilizzo dovete collegare tale contatto alla linea di alimentazione (il filo che arriva da 3V3 di Arduino) la quale deve fornire da 2,2 a 3,6 V (il componente assorbe 0,5 mA). Ciascuna delle uscite analogiche dell'MMA7631L ha il riferimento a metà dell'alimentazione; la tensione sale se l'accelerazione rilevata sul relativo asse è positiva e scende se è negativa. La sensibilità dell'accelerometro può essere impostata mediante il contatto **g-Select**, che internamente è posto a zero logico tramite un resistore; questa condizione imposta ±1.5g (800 mV/g) di sensibilità. Forzando ad 1 logico tale contatto, si sale a ±6g (206 mV/g). Il contatto Self Test è posto internamente a zero logico e nella nostra applicazione non va utilizzato (deve restare aperto).

da 200 passi, 12 V e 2,5A), il labirinto deve essere fatto con la base in compensato spesso 10 mm o in forex (PVC semiespanso) da 13 mm e deve avere un diametro non superiore ad 1 metro. La sua faccia superiore va lavorata per ricavare le scanalature nelle quali incastrare e incollare le pareti esterne e interne, alte circa 8 cm. Dal lato opposto, in prossimità del bordo fissate l'accelerometro che collegherete ai connettori CN3 e CN4: usate un cavo a quattro fili, prendendo l'alimentazione (GND e 3V3) dal solo CN3 e portando l'uscita dell'asse X ad A4 sul CN3 e quella dell'asse Y all'A5 su CN4. L'altro accelerometro sistematelo sulla pedana, collegando l'alimentazione (sempre GND e 3V3) al solo CN1 e le uscite ad A2 del CN1 (asse X) e A3 del CN2 (asse Y). Le uscite asse Z di entrambi gli accelerometri non vanno collegate. Ricordate che gli accelerometri della pedana e del labirinto devono essere rivolti allo stesso modo, altrimenti i movimenti dei due non saranno coordinati. A proposito di pedana: può avere qualsiasi forma e può essere una tavola in abete spessa 20 mm imperniata (ad esempio con un raccordo a sfera particolarmente robusto) ad una base in legno o metallo, oppure semplicemente appoggiata sopra un cuscino o un pezzo di gomma morbida posto centralmente, che consente a chi ci sale di ondeggiare.

REALIZZAZIONE SHIELD E CABLAGGIO

Bene, passiamo al circuito dello shield: la relativa basetta è a doppia faccia può essere ottenuta a partire dalle tracce lato rame scaricabili dal nostro sito *www.elettronicain.it*. Il montaggio dei componenti non è critico: basta iniziare da quelli a basso profilo e rispettare la polarità dei componenti attivi



Traccia per realizzare il labirinto (il diametro è di circa un metro) e le quattro parti che compongono la base. I disegni meccanici dettagliati sono scaricabili dal nostro sito www.elettronicain.it.

Listato 1

```
int average(int num, byte axis){

  int val;
    switch (axis){
      case 0:
        val=analogRead(ACCPED_X);
        totpx += val;
        val=totpx/num;
        totpx=val*(num-1);
        break;
      case 1:
        val=analogRead(ACCPED_Y);
        totpy += val;
        val=totpy/num;
        totpy=val*(num-1);
        break;
      case 10:
        val=analogRead(ACCFLI_X);
        totfx += val;
        val=totfx/num;
        totfx=val*(num-1);
        break;
      case 11:
        val=analogRead(ACCFLI_Y);
        totfy += val;
        val=totfy/num;
        totfy=val*(num-1);
        break;
  }
  return(val);

}

void motorX(){
      takeSingleStepX();
      delayMicroseconds(timeBetweenStepsInMicroSec);
  }

void motorY(){
      takeSingleStepY();
      delayMicroseconds(timeBetweenStepsInMicroSec);
  }


void SetDrive(void) {
 shiftOut(DataPin, ClockPin, MSBFIRST, Drive1);
 shiftOut(DataPin, ClockPin, MSBFIRST, Drive2);
 shiftOut(DataPin, ClockPin, MSBFIRST, Drive3);
 digitalWrite(StrobePin,HIGH);
 delay(2);
 digitalWrite(StrobePin,LOW);
 }
```

e dell'elettrolitico. I moduli MD09B vanno montati su file di contatti femmina a passo 2,54 mm dopo averli dotati di pin-strip a passo 2,54 mm.
Per completare la parte elettronica fissate lo shield su Arduino, quindi collegate gli accelerometri usando i connettori SIL e dei cavetti semplici (basta che il collegamento non superi i 2÷3 metri di lunghezza); chiudete i jumper DIR ed STEP verso l'interno della basetta, in modo che sia Arduino a comandare direttamente gli MD09B con impulsi di STEP e direzione. Quanto a Vmot, questo jumper è meglio chiuderlo verso il lato della basetta e prevedere l'alimentazione esterna alla morsettiera.
Fatto ciò, collegate i motori passo-passo con le due bobine ai contatti 1A/1B e 2A/2B delle morsettiere MOT1 e MOT2; se necessario, potrete invertire in un secondo momento sia la polarità (ossia scambiare per ogni bobina degli stepper-motor 1A e 1B) sia i motori, in base a come il labirinto risponderà al movimento della pedana: per esempio, se inclinando quest'ultima in avanti il labirinto va indietro, dovete scambiare 1A e 1B con 2A e 2B, mentre se l'inclinazione longitudinale ne determina una trasversale, dovete connettere a MOT1 il motore prima collegato a MOT2 e viceversa.
Un'ultima impostazione riguarda la linea VRF di Arduino: questa determina il riferimento di tensione dell'A/D converter dell'ATmega e normalmente è fluttuante, il che corrisponde ad avere 5 V come riferimento; per avere la massima risoluzione di lettura degli accelerometri consigliamo di portare VRF allo stesso potenziale che alimenta questi componenti, ovvero a 3V3 di Arduino.

IL FIRMWARE

Per ragioni di spazio non possiamo riportare il firmware per intero; trovate comunque lo sketch completo sul nostro sito. Nel **Listato 1** riportiamo solo le routine *average*, che opera la lettura degli ingressi analogici (con media) e la *motor*, per la gestione dei motori. Notate come venga chiamata la routine *takeSingleStep* per fare ruotare l'albero del motore di uno step e come poi sia inserito un ritardo in microsecondi che determina la frequenza e quindi la velocità di rotazione del motore. Nel listato trovate anche la routine *SetDrive*, che permette di inviare agli shift register le variabili "Drive" con le impostazioni del caso.



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono facilmente reperibili in commercio. Il master del circuito stampato e lo sketch per Arduino sono scaricabili gratuitamente dal sito della rivista (www.elettronicain.it). Lo shield utilizzato in questo progetto è disponibilie anche in scatola di montaggio (cod. 8190-STEP_SHIELD) al prezzo di 53,00 Euro. Il kit comprende tutti i componenti ed i moduli di potenza per pilotare 3 motori; eventuali driver supplementari (cod. 8218-MD09B) costano 15,00 Euro cadauno. Ciascun accelerometro (cod. 8218-MMA7361L) costa 11,00 Euro mentre ciascun motore da 2,5A (cod. STEPMOT02) costa 44,00 Euro. Per funzionare il dispositivo necessita di una board Arduino Uno (Euro 24,50). 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 - www.futurashop.it

Dai LUCE al tuo tempo libero!



Lampada a LED DC/AC ricaricabile



Efficiente e a basso consumo, può essere utilizzata come normalissima lampada o come torcia. Alimentazione 85-265Vac; 32 LED; 220 lumen; potenza 3W; Ø70mm. Non dimmerabile.

€29,00
cod. LED-ADC-02


Attacco E27


Impugnatura telescopica

Lampada ricaricabile magnetica a 27 LED



- Impermeabile
- Base magnetica
- Gancio estraibile
- Con caricabatterie da auto e da rete

€38,00
cod. ZLLWL2

Lampada multiuso magnetica a 72 LED



- 72 LED ad alta luminosità
- Base magnetica
- Gancio estraibile

€13,00
cod. EWL27

Lampada a LED con timer



- 18 LED bianchi
- Timer 15-30-60 minuti
- Funzione luce notturna
- Luminosità regolabile
- Gancio estraibile

€11,00
cod. EFLC3

Lampada multifunzione da campeggio



- Lanterna 6 LED bianchi
- Lanterna LED giallo
- Torcia 3 LED bianchi

€11,00
cod. ZLLC6

Mini lanterna a 5 LED alta intensità



- Lanterna 5 LED bianchi
- Torcia 1 LED bianco

€7,50
cod. ZLLC4

Lanterna a LED con radio FM



- 6 LED bianchi e 6 LED rossi lampeggianti
- Caricabatteria a dinamo
- Radio FM e sirena

€24,00
cod. EDL4

Prezzi IVA inclusa.

FUTURA ELETTRONICA

www.futurashop.it

Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775 • Fax. 0331/792287

Maggiori informazioni su questi prodotti e acquisti on-line su *www.futurashop.it*.



COMBINATORE VOCALE GSM

di BORIS LANDONI

Collegato a una centrale di allarme, quando viene attivato almeno uno degli ingressi di cui dispone chiama i numeri telefonici memorizzati facendo ascoltare avvisi vocali personalizzati. Seconda e ultima puntata.

Il combinatore telefonico che abbiamo iniziato a descrivere nel fascicolo del mese scorso è l'ideale complemento per ogni genere di impianto di allarme ed è esso stesso una forma di teleallarme: infatti dispone di due ingressi configurabili a piacimento che se attivati determinano l'effettuazione di telefonate, ad uno o più dei numeri che avremo avuto cura di memorizzare nell'apposita lista, e la conseguente riproduzione di messaggi a voce, anche questi preventivamente registrati.
La scelta di far riprodurre messaggi vocali nasce dalla considerazione che per chi riceve le telefonate è certamente più immediato ascoltare un avviso che interpretare segnalazioni formate da qualsivoglia nota acustica o da combinazioni di suoni, specie se si tratta di persone poco avvezze alla tecnologia e comunque non di tecnici.
Nel fascicolo n° 166 abbiamo iniziato la descrizione del combinatore analizzando l'hardware, formato da tre schede da assemblare in un preciso ordine; ci siamo anche soffermati sulle funzionalità, spiegando che il combinatore ha, oltre a due ingressi di attivazione, anche due relé in funzione di uscita, il cui stato può essere controllato a distanza via DTMF (quindi con la tastiera di un cellulare o di un moderno apparecchio telefonico per rete fissa) durante ogni chiamata. Con i relé possiamo ad esempio far suonare una sirena sul posto, attivare un lampeggiante, sbloccare un'elettroserratura per rendere utilizzabile un'uscita di sicurezza aggiuntiva, far chiudere un cancello o una porta tagliafuoco ed altro ancora, ovvero resettare la centrale di allarme collegata al nostro



***Tabella 1** - Gli SMS di comando e configurazione.*

FUNZIONE	COMANDO SMS	VALORE PREDEFINITO	PWD NECESSARIA
Cambio password	PWDxxxxx;12345	12345	SI
Memorizzare 1 numero (massimo 8 numeri, massimo 19 caratteri per numero); x è la posizione	NUMx+393355760937;12345	-	solo se la posizione x non è libera
Cancellare un numero	NUMx;12345	-	SI
Verificare i numeri memorizzati	NUM?;12345	-	SI
Reset completo	RES	-	SI
Impostare i numeri cui verranno inviati gli SMS	SMSxxxxxxxx:ON	Tutti	SI
Impostare i numeri cui verrà fatta la chiamata vocale	VOCxxxxxxxx:OFF	Tutti	SI
Livello logico di allarme alto (presenza tensione in ingresso); x può valere 1 o 2	LIVx:A	Attivo in presenza di tensione	-
Livello logico di allarme basso (assenza tensione in ingresso); x può valere 1 o 2	LIVx:B	Attivo in presenza di tensione	-
Livello logico di allarme variazione	LIVx:V	Attivo in presenza di tensione	-
Richiesta livello	LIV?	-	-
Tempo inibizione ingresso1 (da 00 a 59 minuti)	INI1:mm	5 minuti	-
Tempo inibizione ingresso2 (da 00 a 59 minuti)	INI2:mm	5 minuti	-
Interroga il tempo di inibizione	INI?	-	-
Azzera tempo di inibizione se l'ingresso1 torna a riposo X=0: non azzera x=1: azzera	TIZ1x	0	-
Azzera tempo di inibizione se l'ingresso2 torna a riposo X=0: non azzera x=1: azzera	TIZ2x	0	-
Tempo di osservazione ingresso1 (00÷59 s)	OSS1:ss	1	-
Tempo di osservazione ingresso2 (00÷59 s)	OSS2:ss	1	-
Richiesta tempo osservazione	OSS?	-	-
Attivazione relé in modalità bistabile	OUTx:ON	-	-
Disattivazione relé in modalità bistabile	OUTx:OFF	-	-
Cambio stato relé monostabile (tempo in secondi da 01÷59)	OUTx:ss	-	-
Ripristino relé: x vale 1 per avere il ripristino e 0 per non averlo	RIPx	1	-
Interrogazione ripristino	RIP?	-	-
Richiesta stato ingressi e uscite	STA?	-	-
Invio sms all'accensione: x vale 1 per attivarlo e 0 per disattivarlo	AVVx	0	-
Testo allarme ingresso1 quando è presente tensione (max 100 caratteri)	TIN1A:xxxxxxxxxx	Allarme 1 alto	Multimessaggi non permessi
Testo allarme ingresso1 quando è assente tensione (max 100 caratteri)	TIN1B:xxxxxxxxxx	Allarme 1 basso	Multimessaggi non permessi
Testo allarme ingresso2 quando è presente tensione (max 100 caratteri)	TIN2A:xxxxxxxxxx	Allarme 2 alto	Multimessaggi non permessi
Testo allarme ingresso2 quando è assente tensione (max 100 caratteri)	TIN2B:xxxxxxxxxx	Allarme 2 basso	Multimessaggi non permessi
Testo messaggio accensione (max 100 caratteri; invio solo al primo numero in memoria)	TSU:xxxxxxxxxx	System startup	Multimessaggi non permessi
Tempo di attivazione relé 1 su comando DTMF (00÷59 s)	TIM1:xx	5 secondi	-
Tempo di attivazione relé 2 su comando DTMF (00÷59 s)	TIM2:xx	5 secondi	-
Richiesta tempo attivazione relé	TIM?	-	-
Richiesta pwd DTMF X=0: richiesta disabilitata x=1: richiesta abilitata	SICx	0	-
Filtro chiamate per gestione DTMF X=0: filtro disabilitato x=1: filtro bilitato	FILx	1	-
Disabilita la risposta per quel multimessaggio	RISP	-	-

combinatore, ad esempio perché abbiamo accertato che la chiamata è stata dovuta ad un falso allarme.

Ancora, possiamo attivare delle telecamere per videosorveglianza.

Dopo aver descritto l'hardware e la sua realizzazione non ci resta che vedere quali sono e come sfruttare le funzionalità di cui il

combinatore dispone, spiegando altresì la sintassi dei comandi corrispondenti; concluderemo con un accenno al software che abbiamo realizzato per consentire la configurazione da PC tramite la porta USB. Tale configurazione è di indubbia utilità, perché sebbene il combinatore possa essere configurato da cellulare mediante l'invio di appositi SMS (che, per economizzare sui costi della bolletta possono anche contenere più comandi) ci rendiamo conto che la relativa procedura è lunga e tediosa ed ha senso solamente se siamo lontani dal combinatore.

***Tabella 2** - Comandi DTMF (funzionamento normale).*

FUNZIONE	COMANDO	RISPOSTA
Cambia stato del relé 1 Modalità bistabile	1	Relé attivato Relé disattivato
Cambia stato del relé 2 Modalità bistabile	2	Relé attivato Relé disattivato
Cambia stato del relé 1 Modalità monostabile	3	Relé attivato dopo la commutazione Relé disattivato dopo la commutazione
Cambia stato del relé 2 Modalità monostabile	4	Relé attivato dopo la commutazione Relé disattivato dopo la commutazione
Richiesta stato relé 1	5	Relé attivato Relé disattivato
Richiesta stato relé 2	6	Relé attivato Relé disattivato
Richiesta ingresso 1	7	Tensione presente in ingresso Tensione non presente in ingresso
Richiesta ingresso 2	8	Tensione presente in ingresso Tensione non presente in ingresso
Chiudi comunicazione	0	-

FUNZIONI E COMANDI

Il dispositivo qui descritto svolge due funzioni:
1) avverte con messaggi vocali del verificarsi di una certa condizione elettrica ad uno o entrambi i suoi ingressi optoisolati;
2) comanda a distanza utilizzatori elettrici, elettronici o gestibili elettricamente.

In realtà, il nostro sistema prevede la notifica delle condizioni di allarme agli ingressi anche via SMS: si tratta di una funzione utile ad esempio ai non udenti (che non possono certo ascoltare i messaggi a voce).
Per la configurazione e il controllo a distanza è previsto un nutrito set di comandi da inviare tramite SMS, secondo la sintassi esposta nella **Tabella 1**. C'è poi un secondo set di comandi, impartibili da DTMF, limitato al controllo dei relé e alla lettura dello stato degli ingressi (**Tabella 2**).
I comandi da SMS sono di due tipi: quelli per definire come il sistema debba funzionare (configurazione vera e propria) e quelli operativi, che servono ad attivare e disattivare ognuna delle uscite presenti, in modo sia impulsivo (definendo, entro i limiti previsti, il tempo di attivazione) sia bistabile, nonché ad interrogare l'unità per conoscere la condizione di una delle uscite presenti.
Quest'ultima funzione è indispensabile quando si sia dimenticato l'ultimo comando inviato, ovvero per accertarsi che un SMS di comando abbia sortito l'effetto desiderato; nel caso del comando tramite bitoni, la conferma arriva immediatamente tramite un messaggio vocale udibile nella cornetta del telefono da cui si chiama.
Vediamo brevemente i comandi di configurazione, meglio descritti nella **Tabella 1**: sono LIVx:A per impostare l'allarme quando l'ingresso corrispondente (*x*) riceve tensione (oltre i 3 volt), LIVx:B per determinare allarme in caso di assenza di tensione e LIVx:V per far sì che il circuito rilevi allarme quando all'ingresso indicato si verifica un cambiamento di stato. Ogni impostazione determina la risposta del telecontrollo, sotto forma di SMS contenente il testo "Allarme!! Ingresso 1 alto", qualora sia presente una tensione sull'ingresso 1, oppure "Allarme!! Ingresso 1 basso", in assenza di tensione sull'ingresso 1.
Lo stesso tipo di SMS, ovviamente, viene invito in caso di variazione del livello sull'ingresso 2.
Onde evitare continui invii di segnalazioni di allarme, abbiamo previsto un tempo di inibizione della lettura dopo la segnalazione del primo allarme; tale tempo (l'impostazione predefinita è 5 minuti) si definisce mediante l'invio del comando INI1 per l'input 1 e INI2 per l'ingresso 2; INI? consente di interrogare il combinatore per sapere se e per quali ingressi è stato impostato un tempo di inibizione e a quanto corrisponde.
Scaduto l'intervallo di inibizione, un nuovo cambiamento dello stato logico determina l'invio del messaggio di allarme.
Quando viene rilevato l'allarme agli ingressi, il sistema può inviare messaggi di testo diversi a seconda di qual è il livello riscontrato; i messaggi del caso si impostano con i comandi da SMS TIN1A (per la presenza di tensione) e TIN1B (assenza di tensione) per l'input 1 e TIN2A, TIN2B per l'ingresso 2.
Vediamo adesso i messaggi di comando delle uscite, che sono OUTx:ON (*x* è il numero del relé) per attivare e OUTx:OFF per disattivare; impartendo OUTx:ss l'uscita *x* cambia il proprio stato rispetto all'attuale per un tempo, in secondi, pari ad *ss*.
Con il comando STA? possiamo richiedere al circuito di dirci in che condizione sono ingressi ed

uscite.
Il nostro apparato dispone della funzione di ripristino dello stato dei relé dopo una perdita di tensione; essa serve ad evitare che, in caso di spegnimento accidentale, alla riaccensione i relé perdano l'impostazione data dall'utente mediante i comandi remoti. L'attivazione del ripristino si effettua con RIP1 e la disattivazione con RIP0. Altra funzione è l'invio di un SMS quando il circuito riceve alimentazione; si inserisce con AVV1 e si disattiva con AVV0. Il comando che permette di definire il testo del messaggio è TSU:xxxxxxx, dove al posto delle *x* va scritto il testo.

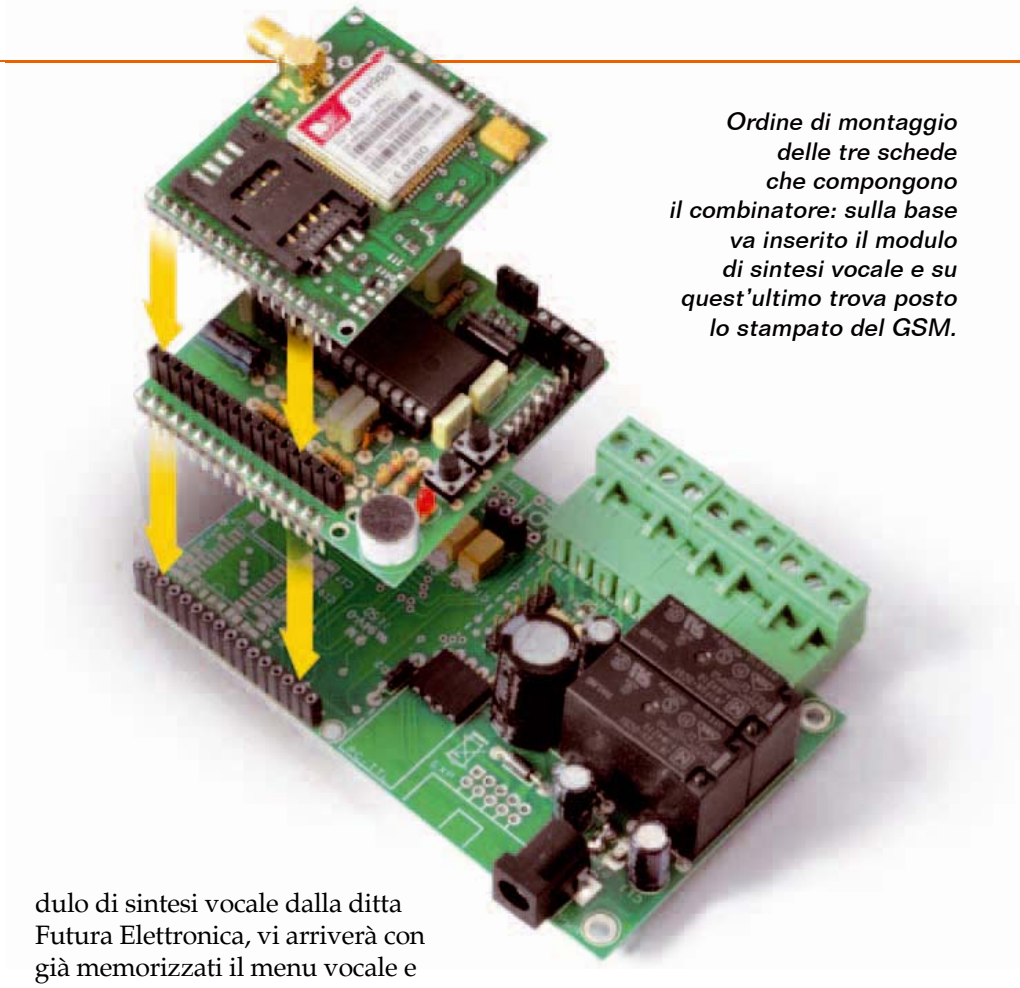
IL COMANDO DA DTMF

Veniamo adesso al comando diretto, che si effettua nel corso di una connessione telefonica: quando chiamiamo il sistema, esso alla risposta presenta un menu vocale, proponendo, con messaggi preventivamente registrati, l'elenco dei possibili comandi; ognuno di essi si impartisce premendo un tasto della tastiera del telefono (**Tabella 2**). Non è necessario attendere che la voce elenchi tutte le opzioni: potete impartire un comando prima, premendo il tasto corrispondente, allorché l'elenco viene interrotto ed il comando è eseguito.
Il menu e alcune frasi relative alla password non possono essere modificati; invece potete modificare a piacimento le risposte che il sistema fornisce sullo stato degli ingressi e delle uscite, nel senso che avete la facoltà di registrare otto nuovi messaggi: due per ingresso ed altrettanti per uscita. Gli avvisi vocali sono due per ovvi motivi: nel caso dello stato degli ingressi, uno avverte della presenza di tensione e l'altro dell'assenza; ognuno può far corrispondere tali condizioni a specifici messaggi. Ad esempio, se un ingresso è collegato all'uscita di allarme di un antifurto che è attiva quando si apre (è la soluzione più sicura perché attiva il teleallarme anche se i ladri tagliano i cavi che collegano l'allarme al nostro circuito) in condizioni di apertura possiamo far riprodurre qualcosa del tipo: "tentativo di furto a casa". Interrogando lo stato dello stesso ingresso quando non è allarmato, potremo avere un messaggio del tipo: "situazione normale".
I messaggi ottenuti durante l'interrogazione telefonica sono gli stessi riprodotti dal combinatore quando chiama a seguito del verificarsi di un allarme.
Lo stesso discorso si applica alla condizione delle uscite, nel senso che abbiamo previsto di registrare un messaggio che indica quando il rispettivo relé è a riposo ed uno che segnala l'attivazione dello stesso; ad esempio, se il relé comanda un'elettroserratura che blocca una porta possiamo far riprodurre "serratura aperta" quando è a riposo e "serratura chiusa" quando è attivato o viceversa. Se usiamo un relé per comandare una sirena locale supplementare possiamo far riprodurre messaggi del tipo: "sirena supplementare azionata" o "sirena supplementare spenta".
Per registrare i messaggi è necessario premere uno dei due pulsanti e poi dare alimentazione; così facendo si entra in programma-



La Pen Drive riconosce la mia voce

La protezione dei dati memorizzati nelle Pen Drive è oggi affidata a chiavi di crittografia, password ed anche all'identificazione da parte del PC tramite sensori biometrici. Se questi sistemi di sicurezza non vi bastano, da oggi avete un alleato in più: Voicelok (*www.voicelok.com*), la prima USB Pen (8 GB di capacità per un costo di circa 38 euro) protetta da un sistema di riconoscimento vocale: basta registrare la propria password vocale usando il software preinstallato sul dispositivo, quindi pronunciarla davanti al microfono del computer ogni volta che si inserisce la chiavetta nella presa USB. Se avete perso la voce a causa di un forte mal di gola, o avete scordato la parola magica, non disperate: è anche possibile inserire la classica password testuale alla richiesta di accesso. Un LED sulla Pen Drive ci dice se la protezione è inserita (luce rossa) o no (luce verde).



Ordine di montaggio delle tre schede che compongono il combinatore: sulla base va inserito il modulo di sintesi vocale e su quest'ultimo trova posto lo stampato del GSM.

zione (5 lampeggi dei LED gialli lo segnalano). Non è possibile registrare i messaggi quando il circuito è già in funzione; tale protezione evita che accidentalmente si possano cancellare i messaggi registrati o che un malintenzionato alteri la programmazione.
Una volta entrati nella programmazione audio, dopo che il pulsante è stato rilasciato viene riprodotto il primo messaggio; premendo il pulsante P1 (NEXT) si passa al messaggio successivo che viene riprodotto e così via.
Per registrare un nuovo messaggio al posto di quello esistente, bisogna premere e tenere premuto P2 (REC) allorché il led LD1 si accende indicando la registrazione (il tempo massimo a disposizione per registrare un messaggio è 5 secondi).
Per fare un esempio, immaginiamo di voler registrare il messaggio che viene riprodotto per secondo: avviata la programmazione audio, quando sentite il primo dei messaggi premete una volta P1 ed udrete il secondo, cioè quello che volete cambiare. Ebbene, prima che questo finisca, premete P2 e parlate a 30÷50 cm dalla capsula microfonica; il LED vi segnala -accendendosi- quando il circuito sta registrando e si spegne non appena termina il tempo a vostra disposizione.
Se volete, potete registrare anche usando l'ingresso di linea, nel qual caso dovete cortocircuitare i capi della capsula microfonica ed applicare il segnale audio ai punti S e – LINE; ciò vi permetterà di registrare messaggi anche con il sonoro. Nel registrare dalla linea state attenti al livello sonoro della fonte BF usata, perché potrebbe far saturare lo stadio d'ingresso dell'ISD1790, sebbene l'AGC sia abbastanza efficace nel prevenire tale situazione.
Notate che se comperate il modulo di sintesi vocale dalla ditta Futura Elettronica, vi arriverà con già memorizzati il menu vocale e i messaggi predefiniti sullo stato degli input e dei relé. Notate altresì che non è possibile cancellare il menu vocale in quanto nelle operazioni di programmazione audio il microcontrollore parte dagli indirizzi successivi a quello dell'ultima locazione dello spazio di memoria riservato al menu stesso.
Bene, torniamo ai comandi e diciamo che sono divisi in quattro categorie: gestione uscite, gestione ingressi, gestione comunicazione e programmazione; sono dieci e vengono impartiti ciascuno con un proprio bitono, ovvero con un tasto della tastiera del telefono.
Premendo il tasto 1, si cambia lo stato del relé 1, intendendo con ciò che tale relé si eccita se stava a riposo e viene rilasciato se, prima del comando, risultava eccitato; con il 2 si fa lo stesso, ma su RL2.
Questi due comandi riguardano la modalità bistabile. Ad ogni istruzione, il modulo risponde con un avviso vocale dicendo qual è lo stato attuale.
Ma i relé possono anche essere configurati per funzionare in maniera impulsiva, intendendo con ciò che gli si può impartire l'ordine di invertire provvisoriamente la propria condizione. Nel comando monostabile è possibile cambiare lo stato di un relé solo per un certo intervallo, definibile via SMS.
Per cambio di stato si intende che se un relé si trova in una certa condizione inverte la sua situazione per il tempo definito con il comando TIMx:tt.
Riguardo a questa funzione, il bitono corrispondente al tasto 3 comanda l'inversione temporanea su RL1 e quello del 4 quella su RL2. Anche in questo caso il telecontrollo conferma le operazioni con i messaggi vocali previsti per il comando bistabile.
Da tastiera è anche possibile conoscere lo stato attuale delle uscite: 5 richiede lo stato del RL1 e 6 quello

del RL2; i messaggi vocali sono sempre gli stessi già descritti.
Quanto ai comandi inerenti agli ingressi, 7 richiede lo stato dell'IN1 e 8 quello dell'IN2; ad entrambi il modulo risponde con un messaggio che comunica lo stato.
In ogni momento, ritenuto di aver terminato il comando, si può ordinare al telecontrollo di chiudere la comunicazione: basta premere il tasto 0.
Bene, prima di parlare dell'installazione è il caso di analizzare i comandi da SMS che impostano il controllo da DTMF: il primo di questi attiva o disattiva il filtro sulle chiamate entranti; si impartisce con un messaggio contenente il testo FILx, dove al posto della x va 1 se si intende impedire l'accesso (la risposta del telecontrollo alle chiamate entranti) ai numeri che non siano quelli in lista, ovvero 0 se si desidera permettere che il telecontrollo risponda alle chiamate provenienti da qualsiasi numero telefonico.
Il secondo è quello che permette di impostare il sistema affinché, dopo aver risposto alle chiamate entranti, richieda la password di accesso al comando DTMF: si impartisce inviando un SMS con il testo SICx, dove al posto di x si scrive 0 se si desidera che l'accesso al comando sia libero ed 1 se, invece, si vuole che, una volta ottenuta la risposta dal telecontrollo, si esige che il comando DTMF possa essere svolto solo dalle persone in possesso della password.
E veniamo agli ultimi comandi, che riguardano la definizione dei tempi di attivazione monostabile, ma da SMS; è stata prevista questa opportunità per poter estendere gli intervalli di inversione dello stato nel comando impulsivo oltre i nove secondi impostabili mediante la tastiera del telefono. Sono previsti due messaggi (uno per relé) aventi in comune la sintassi; per impartirli, bisogna inviare al telecontrollo un SMS (da un telefono il cui numero sia in lista) che contenga il testo: TIM1:xx, dove al posto di xx va il tempo in secondi, espresso nel formato 00÷59; ciò vale per il canale 1, mentre per il due il testo da inserire nell'SMS è TIM1:xx.
Sempre in tema di uscite, esiste un comando che permette di conoscerne il tempo attualmente impostato per la modalità monostabile, senza dover chiamare il telecontrollo e interrogarlo da DTMF: consta di un SMS contenente il testo TIM?, al quale il sistema risponde inviando, al numero di telefono che l'ha mandato, un unico messaggio che riassume gli intervalli definiti per i relé 1 e 2.
Tutti i comandi di configurazione della modalità di telecontrollo e di quella di teleallarme, vengono accettati dal dispositivo ad una condizione: devono provenire da uno degli otto possibili numeri telefonici memorizzati nell'apposita lista. Naturalmente, è prevista una soluzione per operare sul sistema quando lo si avvia per la prima volta, condizione nella quale, evidentemente, non avendo numeri memorizzati non riconoscerebbe alcun utente: i messaggi inerenti alla memorizzazione di indicativi telefonici nella lista vengono riconosciuti ed i corrispondenti comandi eseguiti, anche se provenienti da numeri di telefono sconosciuti, a condizione che contengano la password.
Alla prima accensione e dopo un reset totale, il circuito per tre minuti si dispone a memorizzare il numero del telefono che lo chiamerà (funzione Easy Setup) e lo salverà nella prima posizione della lista.
Passato tale tempo, per aggiungere un numero, cancellarne uno esistente, verificare i numeri che sono già stati memorizzati, bisogna utilizzare gli appositi comandi SMS. A riguardo, si noti che taluni di essi devono essere completati dalla password, anche se provenienti da numeri già memorizzati nella lista: si tratta di quelli riguardanti la richiesta dei numeri, la cancellazione di posizioni contenenti numeri e la sostituzione di un numero esistente (cioè quando nella posizione in cui si vuole scrivere è già presente un numero).
La password è un numero di cinque cifre; dopo la prima accensione o a seguito di un reset del sistema, viene predefinita 12345. È possibile cambiarla, in ogni momento, con un apposito comando via SMS, ovviamente corredato di password (che deve essere quella corrente).



Il combinatore vocale completo, racchiuso in un contenitore lavorato in modo da rendere accessibili la presa d'antenna e l'alimentazione, forato per far uscire il suono dell'altoparlante.

Completa il set di comandi il reset: inviato via SMS da uno dei telefoni in lista, riporta il telecontrollo nelle condizioni iniziali, nel senso che cancella la memoria dei numeri e le modifiche alle impostazioni riguardanti la modalità di funzionamento degli ingressi e delle uscite. Essendo un comando di notevole importanza, il telecontrollo lo prende in considerazione solo se corredato della password.

L'INSTALLAZIONE

Il nostro combinatore vocale si applica a qualsiasi impianto di allarme dotato di un'uscita che si attiva (in caso d'allarme o anomalia) fornendo tensione, ovvero togliendola: nel primo caso occorre impostare l'ingresso corrispondente (IN1 o IN2) in modo che sia sensibile alla presenza di tensione (comando SMS LIVx:A), mentre nel secondo si dovrà configurare l'input affinché risponda quando la tensione viene a mancare (comando da SMS LIVx:B).
Ogni ingresso può essere comandato con un livello di tensione o con un relé che fornisca o tolga alimentazione; per realizzare una buona funzione antisabotaggio conviene optare per la linea a caduta di positivo che poi è quella tipicamente usata per il controllo delle sirene: infatti in questo caso se vengono tagliati i cavi che dall'uscita dell'antifurto portano alla sirena, questa suona lo stesso perché rileva la mancanza dell'alimentazione, che fino a poco prima ne caricava la batteria. La **Fig. 1** mostra alcuni tipi di connessione del genere realizzati con una fonte di tensione esterna o sfruttando un'uscita a relè normalmente chiuso.
Naturalmente in tutti i casi il combinatore deve essere alimentato mediante un'alimentazione sotto batteria, quindi con una propria batteria in tampone: va bene una da 12 volt collegata tra massa ed il punto +V mediante un diodo 1N4007 (con l'anodo sul positivo dell'accumulatore) e una resistenza da 10 ohm in parallelo, ma nulla vieta di prendere tensione dalla centralina cui il combinatore è abbinato, a patto, ovviamente, che i cavi di alimentazione non siano eccessivamente esposti e a portata di mano dei malintenzionati.

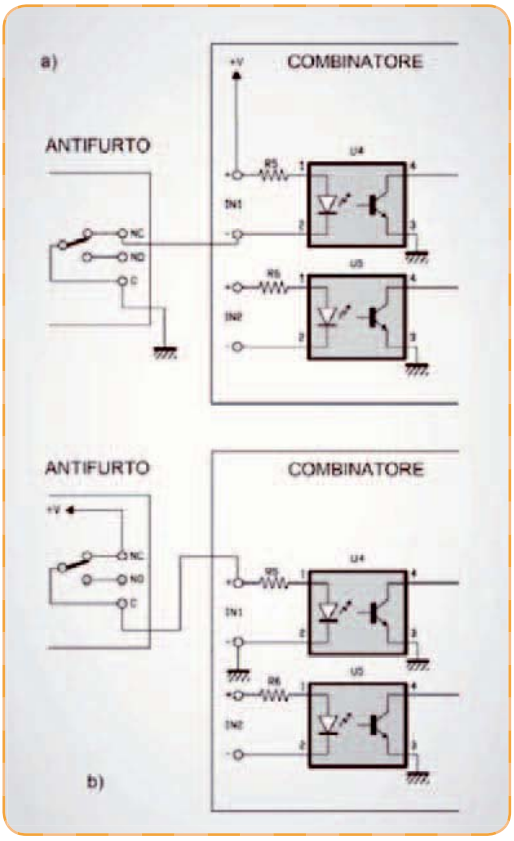


***Fig. 1** - Esempi di comando a caduta di tensione da parte di una centrale dotata di uscite a relè: in a) si alimenta il fotoaccoppiatore del combinatore mediante l'alimentazione principale e si chiude a massa il punto - mediante l'uscita della centrale, mentre in b) è quest'ultima a fornire alimentazione propria.*

IL SOFTWARE DI CONFIGURAZIONE

Il programma che permette la configurazione da PC funziona in ambiente Windows XP/2000/Vista/7 - versione a 32 bit o 64 bit.
Il file corrispondente, in versione compattata, si scarica dal nostro sito *www.elettronicain.it*; fatta l'installazione, nel menu *Programmi* trovate il gruppo *Futura Elettronica*, all'interno del quale vedrete la voce di menu *Configuratore TDG*.
Il programma vale, oltre che per il combinatore telefonico, per i nostri telecontrolli TDG133, TDG134, TDG139, TDG140; basta collegare via USB l'apparato e il computer lo riconoscerà automaticamente.
In base al dispositivo connesso, il software adatta la propria interfaccia grafica, mostrando nella finestra di dialogo principale le sole schede attinenti.
Nella scheda *Informazioni* troviamo riepilogate le caratteristiche del telecontrollo in fase di configurazione, ossia il modello (nel nostro caso TDG135), la versione del firmware e l'IMEI del modulo SIM900.
La sezione *Numeri Master* riepiloga i numeri della lista (8 al massimo) abilitati ad effettuare la configurazione ed impartire comandi via SMS.
Le caselle *Porta COM* e *Baud Rate* riepilogano le informazioni sulla COM virtuale aperta per la connessione USB.
Nella barra dei menu troviamo due comandi: *Esci* permette di abbandonare il programma, mentre *Database* apre la finestra *Rubrica*, dove sono esposti i numeri di telefono memorizzati; in questa finestra i pulsanti *Pulisci Lista* e *Aggiorna Lista* consentono rispettivamente di cancellare la rubrica o inviare al circuito le modifiche alla rubrica.
Va notato che nella sezione in basso della finestra principale, appare l'indicazione di "attesa chiamata da parte del Numero Master" nel caso si colleghi un combinatore in cui non sia stato memorizzato almeno un numero, ovvero nel quale non sia stato eseguito l'Easy Setup.
La finestra principale (**Fig. 2**)

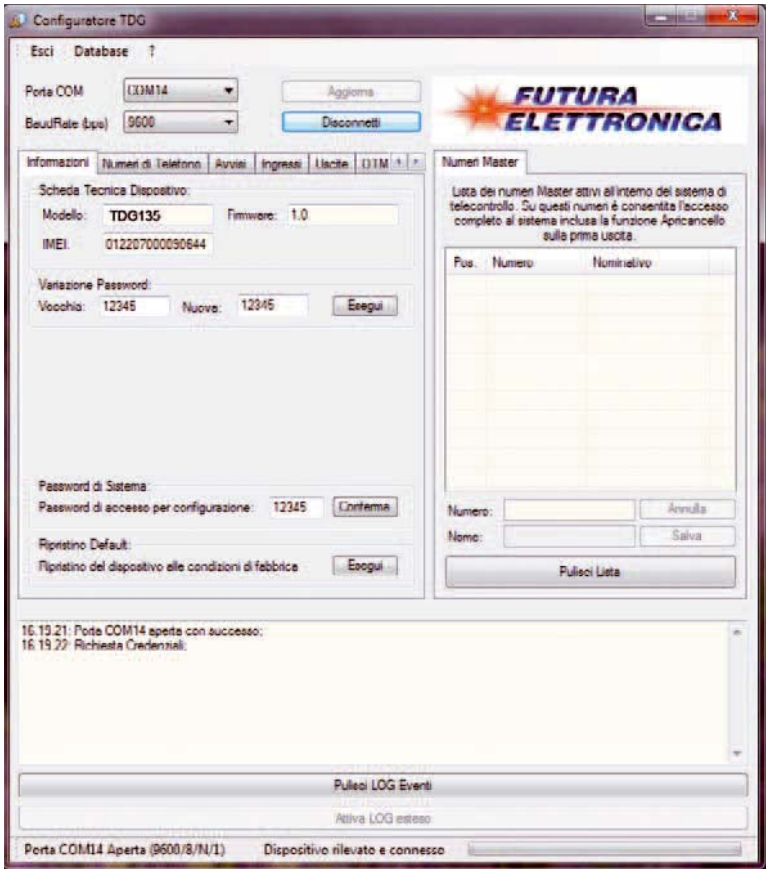


***Fig. 2** Finestra principale del software di configurazione.*

mostra la memorizzazione di un numero di quelli abilitati al comando; notate che si può definire (mediante la casella *Posizione*) in quale posizione salvare il numero, oltre che il nome della persona che possiede il telefono di cui memorizziamo il numero.
Con la scheda *Avvisi* è possibile specificare, per ciascun numero di telefono inserito nella lista dei numeri Master, se in caso di allarme deve ricevere l'avviso mediante SMS o con una telefonata.
La selezione si effettua apponendo il segno di spunta sulle caselle corrispondenti, fermo restando che il combinatore funziona anche senza attivare tale funzione.
La scheda *Ingressi* permette di verificare o modificare la logica degli input (cioè ogni ingresso deve considerarsi in allarme quando è alimentato o privato dell'alimentazione, oppure solo in presenza di una variazione di stato) il tempo di inibizione tra un livello che determina allarme e il successivo, oltre al tempo di osservazione, ovvero l'intervallo temporale per cui la condizione di allarme deve permanere affinché venga considerata tale.
La scheda *Uscite* consente invece l'impostazione del modo di funzionamento dei relé; dall'apposita sezione (*Controllo Manuale*) possiamo attivare manualmente i relé, in modo sia bistabile che impulsivo (in tal caso nel menu a tendina *Tempo* possiamo scegliere l'intervallo di attivazione) per verificare il corretto funzionamento loro e dell'intero sistema.
Nella stessa scheda è presente la sezione *Ripristino all'avvio*, da cui possiamo decidere se il telecontrollo deve ripristinare lo stato delle uscite dopo un black-out, oppure lasciarle tutte a riposo.
E veniamo ora alla scheda forse più importante: DTMF; da qui possiamo impostare gli intervalli di attivazione dei relé di uscita e decidere se il combinatore deve eseguire i comandi DTMF incondizionatamente o solamente se il chiamante digita prima la password di accesso.
Una terza sezione della scheda *DTMF* è quella che permette di attivare o disattivare il filtro sulle chiamate (*Filtro Chiamate DTMF*).
Bene, vediamo adesso la scheda che riguarda gli SMS inviati dal telecontrollo in risposta ai vari comandi o quelli trasmessi a seguito di allarme: si chiama *Testo SMS* e contiene due sezioni; dalla prima (*SMS Startup Modulo)* si può definire il testo del messaggio inviato dal sistema ogni volta che viene alimentato e attivare o disattivare la funzione corrispondente.
Dalla sezione *Personalizza SMS di Allarme si* può, invece, definire ingresso per ingresso sia il testo dei messaggi inviati, sia il livello o la condizione (stato basso, alto, cambio di stato logico) che ne determina l'invio.



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono facilmente reperibili in commercio mentre i master dei CS utilizzati sono scaricabili gratuitamente dal sito della rivista (www.elettronicain.it). Il progetto del combinatore (cod. TDG135) è anche disponibile, comprensivo di antenna, montato e collaudato al prezzo di 128,00 Euro. Sono disponibili separatamente l'apposito contenitore plastico già forato (BOXTDG135, Euro 16,90) e l'interfaccia USB per la programmazione tramite PC (FT782M, Euro 15,00). Non è compreso l'alimentatore. 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

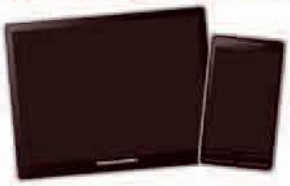
smau

Save the date:
mercoledì 17
giovedì 18
venerdì 19 Ottobre
fieramilanocity

smau business

Vieni in fiera per decidere come innovare la tua impresa!

I principali player del mondo ICT pronti a illustrare le ultime novità nell'ambito delle tecnologie digitali **a supporto del tuo business.**



NOVITÀ 2012 Tablet e cloud per il business

App? Tablet? Cloud? A Smau tutto quello che la tua azienda non può non sapere sui dispositivi, le tecnologie e i servizi che rivoluzioneranno per sempre il nostro modo di lavorare.

Ti aspettano | i migliori fornitori, le School of Management più prestigiose e tutte le anteprime di prodotto.

Smau Mob APP Awards | Le migliori APP in ambito business e consumer basate sui dispositivi mobili di nuova generazione (new tablet, pad e smartphone).

Scopri anche... Aggiornamento, informazione e formazione a cura di:

- ✓**Cloud Computing** e ICT as a Service
- ✓**Fatturazione** Elettronica e **Gestione Documentale**
- ✓**Unified Communication** & Collaboration
- ✓**Progettazione** e Product Lifecycle Management
- ✓**Mobile** business

SDA Bocconi School of Management

Workshop esclusivi di **aggiornamento per la formazione manageriale**. Partecipa e riceverai un attestato SDA Bocconi

arena SOCIAL MEDIA

Tutto quello che non puoi non sapere su **facebook**, **twitter**, **google adwords**, advertising online, ...

smau trade

3ª edizione

Giunto alla sua 3ª edizione, **Smau Trade** si è affermato come il più importante appuntamento dell'ICT Community **per fare il punto sulle nuove sfide offerte dall'industria mondiale delle tecnologie digitali**: dal cloud computing, alle unified communication & collaboration, alla virtualizzazione, e molto altro ancora.

Una piattaforma unica di incontro e networking tra i maggiori **Vendor, i distributori e il mondo delle terze parti** - rivenditori, VAR, system integrator, software house - dove scoprire i programmi di recruiting e le politiche di fidelizzazione dei grandi Player ICT e prendere parte alle sessioni di training e certificazione dedicati ai professionisti del comparto.

smart city ROADSHOW

2ª edizione

un'iniziativa di: smau

La via italiana alle città intelligenti

Dopo il successo della 1ª edizione del 2011, un nuovo appuntamento interamente dedicato al tema delle Smart City e alle più interessanti **progettualità evolute** in atto sul territorio italiano. Dedicato a **Sindaci** e **Assessori** dei Comuni italiani per capire come le moderne tecnologie e i principali fornitori possano supportarli concretamente nel cammino verso lo sviluppo sostenibile della loro città.

! Ingresso riservato a professionisti e manager, dirigenti e operatori della Pubblica Amministrazione e al mondo del Canale ICT previa registrazione al sito **www.smau.it**

contatti: **tel.** 049 8808444 - **email** contact@smau.it - **www.smau.it**

✓ AUTO
✓ CAMPER
✓ MOTO

Fornisci la giusta carica alle tue batterie!

Utilizza questi caricabatterie automatici, di ultima generazione, per ricaricare in modo estremamente semplice e sicuro la batteria dell'auto, della moto o del camper. Estremamente efficienti e affidabili, sono protetti contro errori di utilizzo ed in grado di eseguire in totale autonomia la diagnosi, la rigenerazione, la carica ed il mantenimento.

CARICABATTERIE PER BATTERIE AL PIOMBO DA 6 E 12V

- Adatto per batterie piombo-acido ricaricabili con capacità da 1,2 a 12Ah (6V) e da 1,2 a 60Ah (12V);
- Corrente di carica ad impulsi 1A - 0,6A - 0,2A - 0,1A;
- Carica automatica in 5 fasi;
- Funzione per la carica della batteria nei periodi invernali;
- Protezione contro cortocircuito, inversione di polarità e surriscaldamento.

€ 46,00
cod. BX-3



€ 58,00
cod. BX-1

CARICABATTERIE PER BATTERIE AL PIOMBO DA 12V

- Adatto per batterie piombo-acido ricaricabili con capacità da 1,2 a 120Ah;
- Corrente di carica ad impulsi 3,8A - 0,8A;
- Carica automatica in 5 fasi;
- Funzione per la carica della batteria nei periodi invernali;
- Protezione contro cortocircuito, inversione di polarità e surriscaldamento.

Prezzi IVA inclusa.

FUTURA ELETTRONICA
www.futurashop.it

Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775 • Fax. 0331/792287

Maggiori informazioni su questi prodotti sono disponibili su *www.futurashop.it* tramite il quale è anche possibile effettuare acquisti on-line.



LETTORE RFID CON SD CARD E BUS I²C

Concludiamo la realizzazione del lettore di tessere RFID per controllo accessi con la descrizione del software e delle modalità di comunicazione e interfacciamento mediante il protocollo I²C. Ultima puntata.

Il mese scorso abbiamo iniziato la presentazione di questo nuovo progetto basato su un microcontrollore PIC ed un lettore di tessere RFID, questo mese vogliamo concludere l'illustrazione della scheda ponendo attenzione al software scritto appositamente ed ai meccanismi d'interfacciamento della board verso i sistemi utilizzatori.
Il mezzo di comunicazione fisico, per convogliare i segnali, sarà una coppia di cavi che, attenendosi alle specifiche del protocollo *Inter Integrate Circuit* in sigla I²C, consentirà comunicazioni bidirezionali, quindi da e verso i dispositivi, impiegando un indirizzamento software delle periferiche.
Fin ora vi abbiamo presentato il regolatore di carica per fotovoltaico, il regolatore di carica da parete con ingresso a 220V e questo progetto; in cantiere abbiamo ancora molte idee, la prossima riguarderà un sistema di ricezione radio FM e lettore MP3. Per il momento concentriamoci sull'illustrazione di questo progetto, continuando la descrizione con la presentazione delle strategie adottate per l'implementazione del firmware, della modalità di interconnessione con dispositivi appartenenti alla stessa famiglia ed infine con il metodo di riconoscimento che fornisce il comando di consenso o di dissenso.

IL FIRMWARE

Il firmware descritto in questo articolo (**Fig. 2**) si basa sugli esempi forniti dalla *Microchip tramite le Microchip Application Libraries*. Le *MAL*, sono un'insieme di utility e di librerie software fornite gratuitamente dalla *Microchip* con l'obiettivo di promuovere e facilitare la realizzazione dei firmware più sofisticati. Il loro contenuto informativo spazia in ambiti di applicazione molto diversi, facilitando così l'integrazione di discipline talvolta



dell'ing. LUCA PASCARELLA

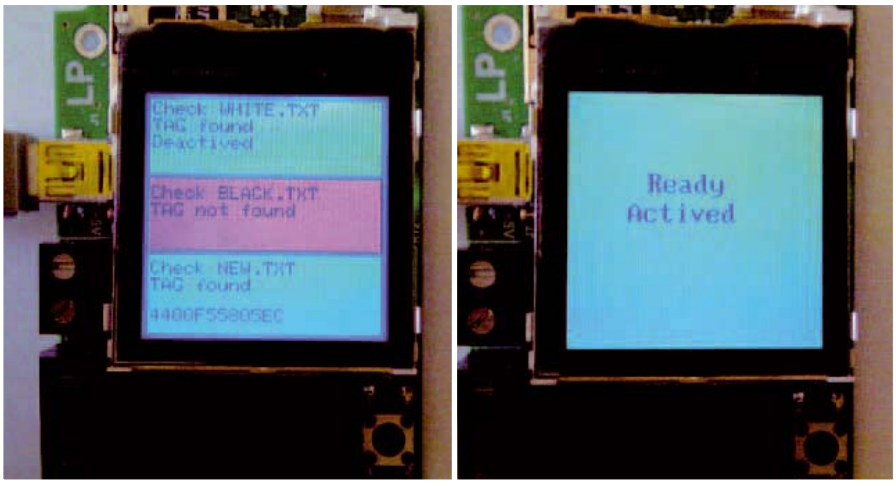


***Fig. 1** - La prima immagine mostra il risultato della lettura di una tessera il cui ID è presente nel file* WHITE.TXT, *la seconda immagine mostra la pulizia dello schermo ed il nuovo stato assunto allo scadere del timeout di 10 secondi dalla lettura della tessera RFID.*

complesse. L'ultima versione del software attualmente disponibile, installabile in ambiente Windows, è scaricabile gratuitamente dall'indirizzo web: http://ww1.microchip.com/downloads/en/softwarelibrary/microchip-application-libraries-v2012-02-15-windows-installer.exe ed è comprensiva: del *framework USB* capace di fornire facile accesso all'impiego delle periferiche *USB* come *Client*, come *Host* oppure come *OnTheGo*; delle librerie grafiche che consentono l'impiego di alcuni modelli molto diffusi di controllori grafici, ma anche per i *PIC24* dotati di apposita uscita grafica o per i *PIC32* capaci di ricreare un *frame buffer* tramite l'impiego della *DMA (Direct Memory Access)*; ancora, le *MAL* appena scaricate, possono fornire facile accesso al *Mass Storage* basato su *File Systems* di tipo: *FAT12, FAT16* e *FAT32* con diversi supporti di memorizzazione, ad esempio le chiavette *USB*, le memorie *SD (Secure Digital)* e le memorie *CF (Compact Flash)*; continuando si possono trovare le librerie che implementano il modello *ISO/OSI* tramite lo *stack TCP/IP* fornendo anche delle

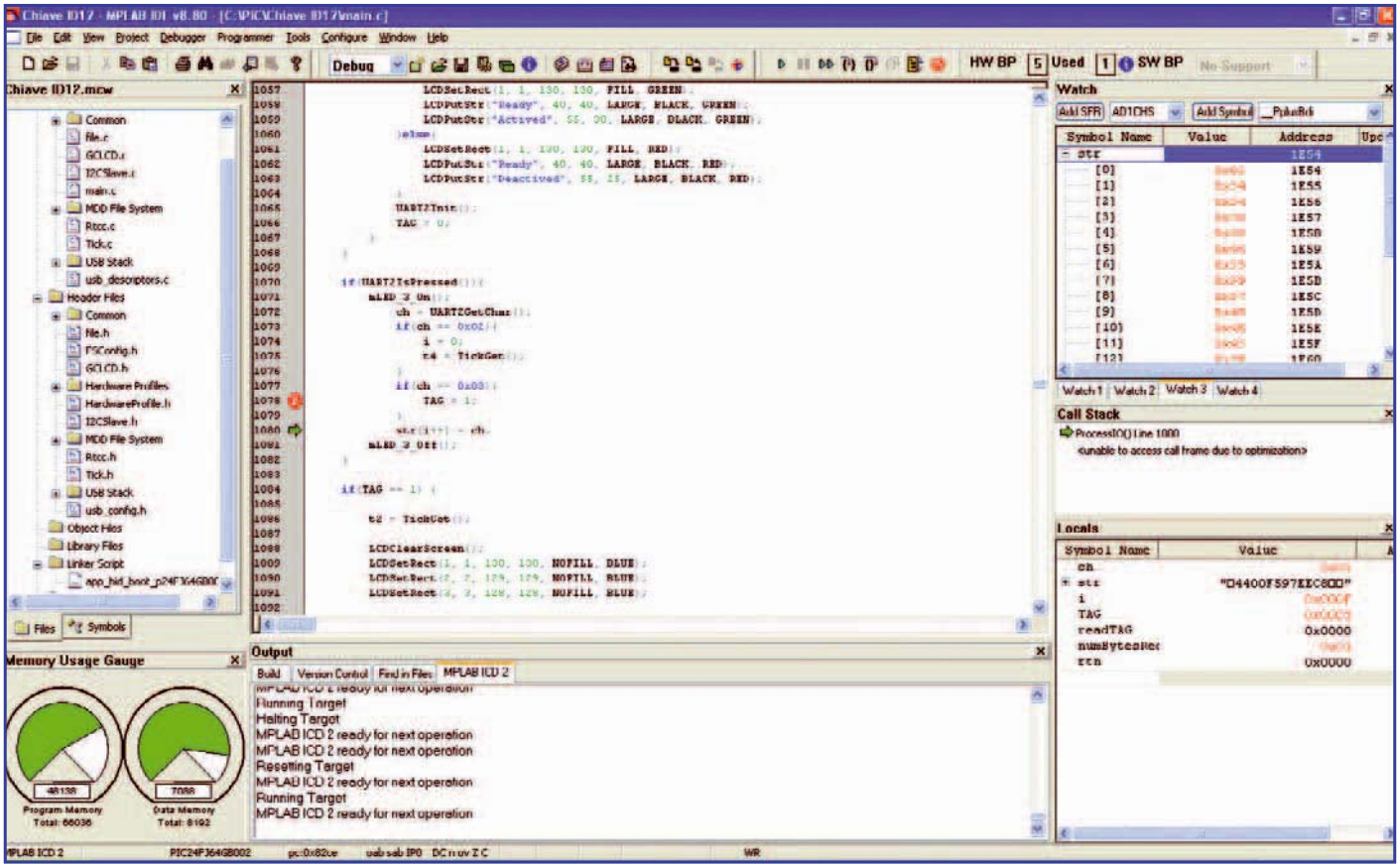


***Fig. 2** - Screenshot dell'ambiente di sviluppo MPLAB IDE versione 8.80 aperto in modalità debug sul progetto lettore di tessere RFID.*

funzioni per alcuni servizi molto utili, come ad esempio: l'*SMTP*, l'*SNMP*, il *DHCP*, il web server, e molto altro; inoltre le *MAL*, forniscono il supporto per il *touch capacitivo* di cui ormai molti PIC sono dotati; le *smart card library* per poter avere facile accesso ai dispositivi *Smart Card* ed infine le librerie per la gestione delle reti wireless *P2P* e *MESH* tramite uno stack di funzioni proprietarie che prendono il nome *MiWi* e si basano su 3 diversi *transceiver* che lavorano a 3 diverse frequenze (*433MHz, 868MHz e 2,4GHz*).
Sfruttando il codice di esempio, quindi testato e debuggato quotidianamente da migliaia di utenti, diventa molto semplice realizzare progetti decisamente complessi, oppure, diventa altrettanto semplice impiegare periferiche particolari, come i display LCD, la connessione ethernet o, nel nostro caso, la comunicazione *Universal Serial Bus* in sigla *USB*.
È proprio in riferimento a quest'ultima che nasce il firmware scritto per questo circuito.
Le *Microchip Application Libraries* sono strutturate in modo da fornire degli esempi pratici, pronti all'uso divisi per categoria di applicazione in cartelle e sottocartelle. L'esempio di partenza scelto per questo progetto è il "*Device - Composite - MSD + CDC*", quest'ultimo è stato adattato per il microcontrollore PIC24FJ64GB002 ed è servito come punto di partenza per realizzare il collegamento *USB client* verso un normale computer.
Essendo l'esempio strutturato e commentato già dalla fonte non ci resta che descrivere la parte di codice aggiunta per controllare il lettore di tessere RFID ed il display LCD.
Proseguendo con ordine, si può vedere che tutto il codice di controllo, al di fuori delle inizializza-

Listato 1

```
void ProcessIO(void)
{
 char ch;
 static char str[20];
 static int i = 0, TAG = 0, readTAG = 0;
    BYTE numBytesRead;
 int rtn;

 if(loading && TickGet() - t3 >= TICK_SECOND * 10ul){
        LCDClearScreen();
        LCDPutStr("Ready", 55, 40, LARGE, BLACK, WHITE);
        loading = 0;
 }

    // User Application USB tasks
     //Blink the LEDs according to the USB device status
    if(blinkStatusValid)
    {
        BlinkUSBStatus();
    }

 // Blink LED0 (right most one) every second.
 if (TickGet() - t >= TICK_SECOND / 2ul) {
     t = TickGet();
     mLED_3_Toggle()
 }

 if (TickGet() - t2 >= TICK_SECOND * 10ul) {
     t2 = TickGet();
        if(TAG == 3){
              //LCDClearScreen();
              if(active){
                    LCDSetRect(1, 1, 130, 130, FILL, GREEN);
                    LCDPutStr("Ready", 40, 40, LARGE, BLACK, GREEN);
                    LCDPutStr("Actived", 55, 30, LARGE, BLACK, GREEN);
              }else{
                    LCDSetRect(1, 1, 130, 130, FILL, RED);
                    LCDPutStr("Ready", 40, 40, LARGE, BLACK, RED);
                    LCDPutStr("Deactived", 55, 25, LARGE, BLACK, RED);
              }
              UART2Init();
              TAG = 0;
        }
 }

 if(UART2IsPressed()){
        mLED_3_On();
              ch = UART2GetChar();
              if(ch == 0x02){
                    i = 0;
                    t4 = TickGet();
              }
              if(ch == 0x03){
                    TAG = 1;
              }
              str[i++] = ch;
        mLED_3_Off();
 }

 if(TAG == 1) {

        t2 = TickGet();

        LCDClearScreen();
        LCDSetRect(1, 1, 130, 130, NOFILL, BLUE);
        LCDSetRect(2, 2, 129, 129, NOFILL, BLUE);
        LCDSetRect(3, 3, 128, 128, NOFILL, BLUE);

        rtn = checkWhiteList(&str[1]);
        switch(rtn){
              case 1:
                    LCDSetRect(4, 4, 42, 127, FILL, GREEN);
                    LCDPutStr("Check WHITE.TXT", 5, 5, SMALL, BLACK, GREEN);
                    LCDPutStr("TAG found", 14, 5, SMALL, BLACK, GREEN);
                    if(active){
                          active = 0;
```

(Continua)

Listato 1 - segue

```
                        LCDPutStr("Deactived", 23, 5, SMALL, BLACK, GREEN);
                }else{
                        active = 1;
                        LCDPutStr("Actived", 23, 5, SMALL, BLACK, GREEN);
                }
        break;
        case 2:
                LCDSetRect(4, 4, 42, 127, FILL, RED);
                LCDPutStr("Check WHITE.TXT", 5, 5, SMALL, BLACK, RED);
                LCDPutStr("TAG not found", 14, 5, SMALL, BLACK, RED);
        break;
        case 3:
                LCDSetRect(4, 4, 42, 127, FILL, BLACK);
                LCDPutStr("Check WHITE.TXT", 5, 5, SMALL, RED, BLACK);
                LCDPutStr("Error in open", 14, 5, SMALL, RED, BLACK);
        break;
        default:
                LCDPutStr(&str[1], 5, 5, SMALL, BLACK, WHITE);
        break;
}

LCDSetLine(43, 3, 43, 127, BLUE);
LCDSetLine(44, 3, 44, 127, BLUE);
LCDSetLine(45, 3, 45, 127, BLUE);

rtn = checkBlackList(&str[1]);
switch(rtn){
        case 1:
                LCDSetRect(46, 4, 84, 127, FILL, GREEN);
                LCDPutStr("Check BLACK.TXT", 50, 5, SMALL, BLACK, GREEN);
                LCDPutStr("TAG found", 59, 5, SMALL, BLACK, GREEN);
        break;
        case 2:
                LCDSetRect(46, 4, 84, 127, FILL, RED);
                LCDPutStr("Check BLACK.TXT", 50, 5, SMALL, BLACK, RED);
                LCDPutStr("TAG not found", 59, 5, SMALL, BLACK, RED);
        break;
        case 3:
                LCDSetRect(46, 4, 84, 127, FILL, BLACK);
                LCDPutStr("Check BLACK.TXT", 50, 5, SMALL, RED, BLACK);
                LCDPutStr("Error in open", 59, 5, SMALL, RED, BLACK);
        break;
        default:
                LCDPutStr(&str[1], 5, 5, SMALL, BLACK, WHITE);
        break;
}

LCDSetLine(84, 3, 84, 127, BLUE);
LCDSetLine(85, 3, 85, 127, BLUE);
LCDSetLine(86, 3, 86, 127, BLUE);

rtn = checkNewList(&str[1]);
switch(rtn){
        case 0:
                LCDSetRect(87, 4, 127, 127, FILL, BLACK);
                LCDPutStr("Check NEW.TXT", 90, 5, SMALL, RED, BLACK);
                LCDPutStr("Error in open", 99, 5, SMALL, RED, BLACK);
                LCDPutStr(&str[1], 115, 5, SMALL, RED, BLACK);
        break;
        case 1:
                LCDSetRect(87, 4, 127, 127, FILL, GREEN);
                LCDPutStr("Check NEW.TXT", 90, 5, SMALL, BLACK, GREEN);
                LCDPutStr("TAG found", 99, 5, SMALL, BLACK, GREEN);
                LCDPutStr(&str[1], 115, 5, SMALL, BLACK, GREEN);
        break;
        case 2:
                LCDSetRect(87, 4, 127, 127, FILL, YELLOW);
                LCDPutStr("Check NEW.TXT", 90, 5, SMALL, BLACK, YELLOW);
                LCDPutStr("TAG added", 99, 5, SMALL, BLACK, YELLOW);
                LCDPutStr(&str[1], 115, 5, SMALL, BLACK, YELLOW);
        break;
        default:
                LCDPutStr(&str[1], 5, 5, SMALL, BLACK, WHITE);
        break;
}
```

zioni varie, avviene in un'unica funzione, ovvero *ProcessIO()*, questa procedura, riportata anche nel **Listato 1** per maggiore chiarezza, è invocata periodicamente dal core del microcontrollore in quanto inserita in un loop infinito *while(1){…}* insieme alle funzioni necessarie a gestire la comunicazione USB. Le prime operazioni eseguite all'interno della funzione riguardano il lampeggio coordinato dei LED D5 e D6 che assumono diverse combinazioni di luminosità in relazione allo stato di attivazione della connessione USB con la periferica *host*. Proseguendo si può trovare l'istruzione condizionale *if (TickGet() - t2 >= TICK_SECOND * 10ul){…}* che ha l'obiettivo di ripulire lo schermo dopo che sono trascorsi 10 secondi dall'ultima tessera accostata al lettore RFID. All'interno di questa istruzione condizionale il fondo dello schermo viene colorato di verde o di rosso in relazione allo stato che dovrà assumere l'uscita: attiva o disattiva (rispettivamente verde o rosso). Un esempio dell'azione di pulizia dello schermo si può vedere nella **Fig. 1**.
Alla pulizia del display segue l'istruzione *if(UART2IsPressed())* che ha l'obiettivo di controllare il buffer in ricezione dalla chiave ID12 e copiare nell'array statico *str* la stringa ricevuta. Le librerie che gestiscono la porta seriale risiedono nel file **uart2.c**, queste ancora una volta sono librerie standard impiegate in molti progetti e quindi ampiamente documentate e descritte anche in altri nostri articoli. Ritornando alla descrizione dell'ultima istruzione condizionale citata, essendo la ricezione dei byte molto lenta, data dai soli 9600 bps della seriale, accade che durante la fase di copia della stringa solo parte dell'informazione viene tra-

smessa e quindi ricevuta dal PIC, per cui è necessario attendere la completa ricezione ed iterare più volte il prelievo dei singoli byte dal buffer seriale. Per far ciò avremmo potuto impiegare un loop ed attendere la ricezione del carattere di chiusura che corrisponde al valore esadecimale 0x03, così facendo però si sarebbe ottenuto uno stallo del software che avrebbe bloccato la gestione della comunicazione USB verso l'Host con la perdita della connessione. Per sostituire il ciclo while è stata preferita ed adottata una logica a stadi per cui il codice può proseguire con la gestione dell'USB fin quando la variabile booleana TAG non verrà settata.
Nel caso in cui si riceve il valore esadecimale *0x03* di chiusura della trasmissione e la variabile TAG è settata, il codice potrà proseguire con il contenuto di questa nuova istruzione condizionale *if(TAG == 1){…}*. L'obiettivo dell'ultima istruzione citata è quello di controllare la presenza della stringa ricevuta all'interno dei file: *WHITE.TXT, BLACK.TXT e NEW.TXT*. Anche in quest'ultimo caso il codice è molto semplice ed è sufficiente spiegare solo uno dei tre casi. In particolare nel **Listato 2** è possibile vedere riportata l'implementazione della funzione *int checkNewList(char *str);* quest'ultima è strutturata in modo da ricevere come parametro un puntatore a *char* che nel contesto impiegato consentirà di leggere i byte dall'array statico str dichiarato nel **Listato 1**.

Listato 1 - segue

```
        TAG = 3;
        if(mUSBUSARTIsTxTrfReady())
        {
              putrsUSBUSART(&str[1]);
              TAG = 3;
        }
}

    // User Application USB tasks
    if((USBDeviceState < CONFIGURED_STATE)||(USBSuspendControl==1)) return;

    if(buttonPressed)
    {
        if(stringPrinted == FALSE)
        {
            if(mUSBUSARTIsTxTrfReady())
            {
                putrsUSBUSART("Button Pressed -- \r\n");
                stringPrinted = TRUE;
            }
        }
    }
    else
    {
        stringPrinted = FALSE;
    }

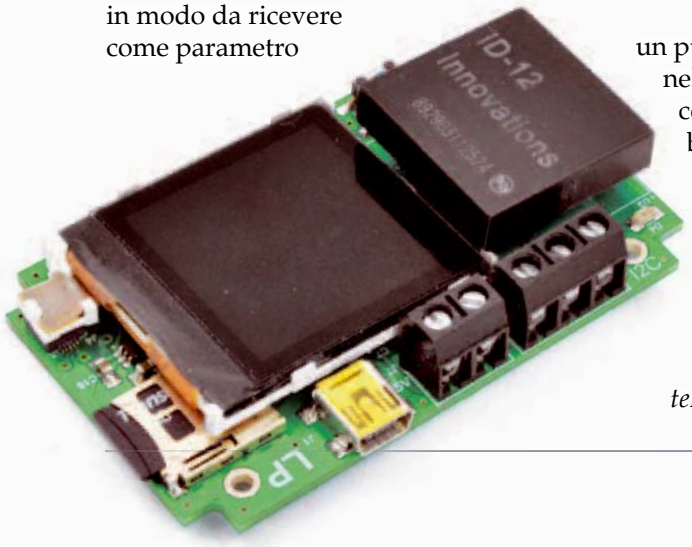
    if(mUSBUSARTIsTxTrfReady())
    {
         numBytesRead = getsUSBUSART(USB_Out_Buffer,64);
         if(numBytesRead != 0)
         {
               BYTE i;

               for(i=0;i<numBytesRead;i++)
               {
                     switch(USB_Out_Buffer[i])
                     {
                              case 0x0A:
                              case 0x0D:
                                   USB_In_Buffer[i] = USB_Out_Buffer[i];
                                   break;
                              default:
                                   USB_In_Buffer[i] = USB_Out_Buffer[i] + 1;
                                   break;
                     }

               }

               putUSBUSART(USB_In_Buffer,numBytesRead);
         }
}

    CDCTxService();
    MSDTasks();
}//end ProcessIO
```



La funzione *checkNewList* differisce solo per un particolare, rispetto alle funzioni *checkWhiteList* e *checkBlackList* ovvero nel primo caso, oltre a controllare la presenza della stringa ricevuta (aprendo il file in lettura) nel caso in cui il confronto fallisce, l'ID viene aggiunto in coda al file aggiornando il suo contenuto (aprendolo in modalità *append*). La comparazione delle stringhe avviene in maniera semplice, infatti il codice si limita ad estrarre una riga (che corrisponde ad un unico ID) ed

Listato 2

```
int checkNewList(char *str){

 FSFILE *fp;
 char buf[256];
 char *ps;
 int found = 0;
 CETYPE err;
 int len;

 if((fp = FSfopen("NEW.TXT", "r"))){
        while(1){
              if(!(ps = _fgets(buf, sizeof(buf), fp))){
                    found = 0;
                    break;
              }

              if(strstr(str, ps)){
                    found = 1;
                    break;
              }
        }
        FSfclose(fp);
 }

 if(found == 0){
        if(!(fp = FSfopen("NEW.TXT", "a"))){
              err = FSerror();
              Nop();
              return found;
        }
        //len = strlen(str);
        len = 12;
        FSfwrite(str, sizeof(char), len, fp);
        FSfwrite("\r\n", sizeof(char), 2, fp);
        FSfclose(fp);
        found = 2;
 }

 return found;
}
```

a compararlo con la funzione *strstr(str, ps)*, ovvero si ricercherà la presenza della stringa ricevuta (*str*) dal lettore nella stringa (*ps*) ottenuta tramite la funzione *_fgets(buf, sizeof(buf), fp)*. Il controllo da esito positivo nel caso in cui tutta la stringa ricevuta è contenuta all'interno della riga del file. Ovviamente il software potrebbe essere migliorato introducendo un controllo di *buffer overflow* sulle stringhe ricevute dal lettore e sulle righe lette dalla *micro SD*. A questo punto il software non farà altro che presentare i risultati sul disply LCD modificando il colore dello sfondo in funzione del risultato ottenuto dai check.

***Tabella 1** - Elenco dei comandi e delle risorse disponibile in lettura tramite interfaccia I²C.*

Comando	Direzione	Descrizione	Risultato
0x00	Lettura	Legge l'anno	Da 0 a 99. 2000 sottointeso
0x01	Lettura	Legge il mese	Da 1 a 12
0x02	Lettura	Legge il giorno	Da 1 a 31
0x03	Lettura	Legge il giorno della settimana	Da 0 a 6
0x04	Lettura	Legge le ore	Da 0 a 23
0x05	Lettura	Legge i minuti	Da 0 a 59
0x06	Lettura	Legge i secondi	Da 0 a 59
0x07	Lettura	Legge lo stato	0 (disabilitato) o 1 (abilitato)
0x00	Scrittura	Imposta l'anno	Da 0 a 99. 2000 sottointeso
0x01	Scrittura	Imposta il mese	Da 1 a 12
0x02	Scrittura	Imposta il giorno	Da 1 a 31
0x03	Scrittura	Imposta il giorno della settimana	Da 0 a 6
0x04	Scrittura	Imposta le ore	Da 0 a 23
0x05	Scrittura	Imposta i minuti	Da 0 a 59
0x06	Scrittura	Imposta i secondi	Da 0 a 59

L'*USB* acronimo di *Universal Serial Bus* è uno standard di comunicazione seriale adottato largamente in molte tipologie di circuiti elettronici capace di collegare diverse famiglie di periferiche ad un'unità *Host* (in genere un personal computer). L'interfaccia USB è stata ideata e progettata per consentire la connessione simultanea di più periferiche anche funzionalmente differenti le une dalle altre. La standardizzazione sia del connettore (in **Fig. 3** è possibile vedere i diversi formati attualmente disponibili) che dei messaggi software ha consentito di creare un sistema *plug-and-play* capace di collegare o scollegare qualsiasi dispositivo senza dover riavviare il computer (funzionamento noto anche come *hot swap*).
Le periferiche *USB* possono appartenere a due grandi gruppi: i *device* e gli *host*. Al primo gruppo appartengono tutti quei dispositivi che offrono servizi, ad esempio i mouse, le chiavette di archiviazione di massa e così via. Il gruppo degli *host*, invece, comprende comunemente i personal computer, gli oggetti *embedded*, come particolari *Router ADSL*, alcune tipologie di *tablet* e simili. Quindi con il termine *host* ci si riferisce a quell'insieme di dispositivi che usufruiscono dei servizi forniti dai *client*. In realtà l'evoluzione dello standard USB ha portato alla creazione di

ATTIVAZIONE REMOTA

Dal principio di funzionamento e dalla descrizione del firmware sarà facilmente emerso come questo progetto di chiave elettronica non sia dotato né di relè né di Mosfet direttamente commutabili, ciò induce a pensare che il siste-

circuiti in grado di commutare al volo la loro funzionalità, questi dispositivi sono noti come *On-The-Go* ed è proprio il caso della periferica *USB* selezionata sul microcontrollore scelto in questo progetto ad appartenere a questa categoria. Essa, infatti, è dotata di tutto l'hardware necessario per funzionare sia come periferica *client* che come periferica host, ovviamente l'impiego nell'una o nell'altra modalità dipende dal software sviluppato, nel nostro caso ci limiteremo ad usarla solo come periferica *client*. Tra il software fornito dalle *Microchip Application Libraries* si possono trovare anche alcuni esempi che illustrano il funzionamento della periferica USB in modalità *OnTheGo*.
Continuando a descrivere lo standard *USB*, possiamo riassumervi alcuni dei passaggi fondamentali che hanno portato all'evoluzione ed al miglioramento di quest'ultimo. Si è iniziati con lo standard *1.0* dell'USB nel gennaio 1996 capace di supportare una velocità massima teorica nei collegamenti di solo *1,5 Mbit/s*, (velocità adeguata per mouse, tastiere ed in genere per dispositivi lenti), le limitazioni di banda della prima versione hanno fatto si che venisse migliorato e definito un nuovo standard capace di incrementare le performance; ciò ha portato alla definizione della versione *1.1* del settembre 1998 capace di aggiungere la modalità *full speed*, che innalza la massima velocità teorica a *12 Mbit/s*.
Il grande cambiamento si è ottenuto con la definizione dello standard 2.0 dell'aprile 2000 che ha portato ad un incremento di banda sostanziale, fino al valore massimo teorico di *480 Mbit/s*, consentendo così alla periferica USB di essere competitiva con altri standard veloci che si sono affermati nel frattempo, come ad esempio lo standard *Firewire 400 (IEEE 1394)* che ha una velocità di trasferimento di *400 Mbits/s*.
Solo negli ultimi anni si parla della versione *3.0* che porta un significativo incremento di banda innalzandola a ben *4,8 Gbit/s*. Questo standard, sta avendo una diffusione sempre maggiore, in quanto viene ormai integrato nei nuovi personal computer. Lo standard *3.0* oltre a mantenere la compatibilità con la versione precedente dimostra degli ottimi vantaggi soprattutto nel settore delle memorie di massa dove a beneficiare saranno i trasferimenti di dati che potranno avvenire in tempi più ragionevoli.
Per quanto riguarda le caratteristiche fisiche delle connessioni *USB* possiamo dire che esso è sostanzialmente composto da 4 connessioni che servono ad alimentare i client ed a scambiare i dati, quindi avremo: GND e +5V e due segnali differenziali i cui valori in tensione sono simmetrici e riflessi, questa caratteristica tipica di molti bus moderni consente di avere un'elevata immunità ai disturbi elettromagnetici sulle lunghe distanze.
Solo nelle ultime versioni di connettori (*mini* e *micro* di **Fig. 3**) i collegamenti sono 5 e non più 4, la presenza di un ulteriore collegamento serve per consentire alla CPU funzionante in modalità *OnTheGo* di discriminare il tipo di periferica connessa ovvero tra *client* e *host*, questa particolarità si può già apprezzare in molti *smartphone* e *tablet* moderni.

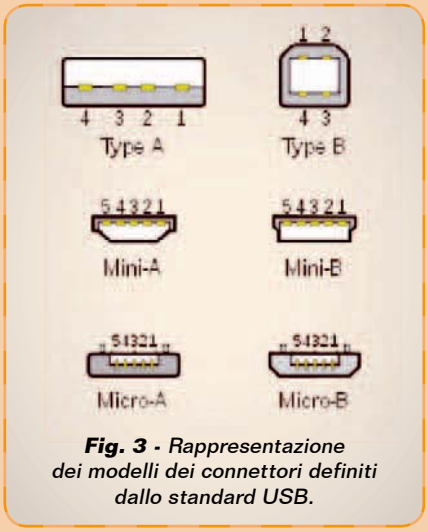


***Fig. 3** - Rappresentazione dei modelli dei connettori definiti dallo standard USB.*

ma non possa essere impiegato in un contesto di tipo stand alone ma lo si possa impiegare solo insieme ad altre periferiche. La deduzione da un certo punto di vista non è corretta. Se è pur vero che il segnale di controllo è solo un'informazione binaria residente nella memoria del dispositivo e che l'attuazione (trasformazione dell'informazione binaria in circuito aperto o chiuso) bisogna ottenerla impiegando un'altra scheda che interrogherà periodicamente il lettore e stabilirà se è il caso di chiudere o aprire il circuito, non è detto che non si possa realizzare una controparte capace di interrogare il lettore e riportare la risposta commutando lo stato di un Relè.
Tornando al nostro caso concreto della **Tabella 1** si possono ricavare quali sono i codici da inviare al circuito con chiave ID12 per poter conoscerne lo stato e gestire le impostazioni di sistema.
Sostanzialmente le interrogazioni minime sono molto semplici, ad esempio, inviando il comando esadecimale *0x07* si può ottenere una risposta positiva rappresentata da un byte con tutti i bit posti ad uno logico oppure si può ottenere una risposta negativa rappresentata da un byte con tutti i bit posti a zero logico. Volendo estendere le funzionalità si potrà aggiornare il sistema comunicando anche la data e l'ora della centralina così che la chiave possa funzionare anche da temporizzatore. Come possibili espansioni si potrebbe inserire all'interno della *micro SD* un file nominato ad esempio *Pink list* nel quale sono riportati i codici delle chiavi per le quali sono previsti servizi specifici, oppure sono consentiti utilizzi con limitazioni delle risorse protette dal sistema ID2.

FILE DI CONTROLLO

Come anticipato in altri punti dell'articolo all'interno della *micro SD* sono sempre presenti tre file: *WHITE.TXT, BLACK.TXT* e *NEW.TXT*, visibili anche nella **Fig. 4**. Il software è strutturato in modo tale che se all'accensione uno di essi risulta assente o danneggiato esso viene sempre rigenerato con uno nuovo. Il file *NEW.TXT* è l'unico che gode di vita propria, ovvero, verrà sempre generato e popolato dal firm-

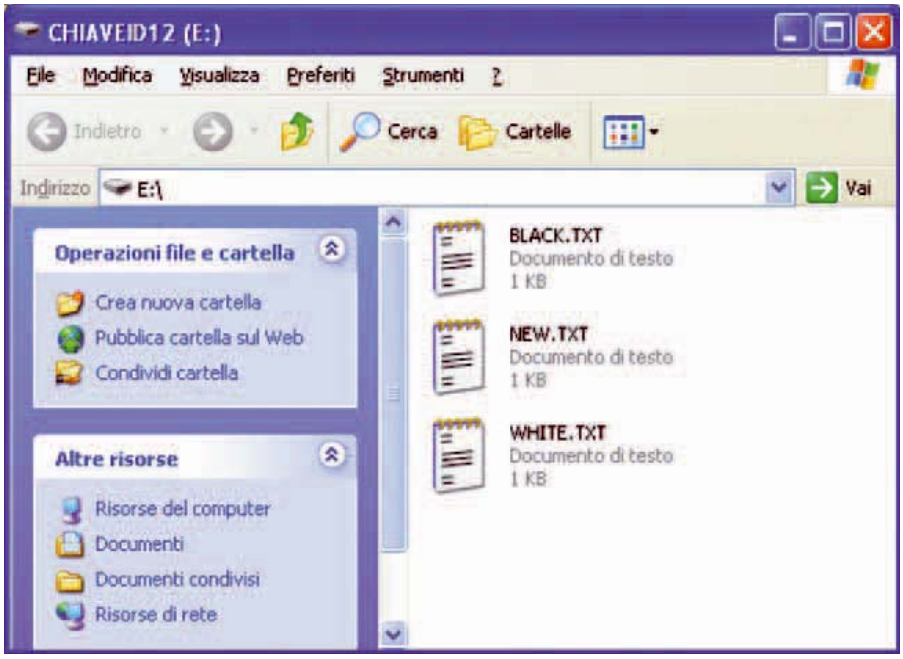


***Fig. 4** - In questa figura è possibile vedere i 3 file che saranno sempre presenti all'interno della micro SD contenuta nel lettore.*



***Fig. 5** - La prima figura mostra il file* WHITE.TXT *appena generato dal firmware del microcontrollore PIC, la seconda figura mostra come apparirà il contenuto del file dopo l'inserimento manuale delle stringhe rappresentative del codice ID presente all'interno delle tessere abilitate all'accesso. Nella seconda parte della figura, si può notare che ogni ID è separato da un carattere di* new line *che identifica il fine riga.*

ware presente all'interno del PIC, ed al suo interno saranno riportate tutte le stringhe corrispondenti ai TAG che sono stati riconosciuti dal lettore fino a quel momento. Viceversa nei rimanenti due file, è necessario sempre copiare manualmente le stringhe dei file che si vogliono abilitare o meno all'accesso al sistema controllato. In **Fig. 5** è possibile vedere lo stato che assumerà il file *WHITE.TXT* prima e dopo l'inserimento delle stringhe. Per popolare correttamente i file WHITE.TXT e BLACK.TXT è sufficiente passare una nuova tessera sul lettore, il quale provvederà a memorizzare l'ID in fondo al file NEW.TXT. A questo punto l'utente può collegare il circuito ad un PC ed aprire il file NEW.TXT per copiare l'ultima riga in WHITE.TXT o BLACK.TXT.

CONCLUSIONE

La nostra prassi prevede di estendere l'invito ad ogni lettore della rivista nel cimentarsi con l'assemblaggio del circuito e la programmazione del firmware base. Allo stesso tempo, la possibilità di scaricare anche i sorgenti originali oltre al firmware precompilato, apre la strada alla sperimentazione ed allo studio casalingo. Il software fornito a corredo all'articolo può essere preso come trampolino di lancio dal quale partire ed aggiungere le vostre personalizzazioni, in modo da ottenere un sistema di protezione elettronico comodo e versatile adatto ad ogni esigenza. A tal fine vi ricordiamo che il circuito, per come è stato pensato e disegnato, è adatto ad essere impiegato anche in progetti semplici nei quali il circuito da commutare (da aperto a chiuso o viceversa) sia posto ad una certa distanza dal lettore RFID (anche ad alcuni metri). Facendo viaggiare delle informazioni digitali piuttosto che un semplice segnale ON OFF il sistema sarà evidentemente più robusto alle manomissioni. L'ultima nota riguarda l'ambiente di sviluppo; tutto il software è stato sviluppato in linguaggio C sfruttando l'ambiente gratuito *MPLAB-IDE* della *Microchip* ed il compilatore *C30 Lite* che può anche esso essere scaricato gratuitamente direttamente dal sito della Microchip e privo di limitazioni per due mesi.



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono di facile reperibilità. Il master del circuito stampato può essere scaricato dal sito della rivista così come il firmware utilizzato per programmare il microcontrollore. Il lettore RFiD a 125 kHz ID12 costa 33,00 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

IMPARA A PROGRAMMARE ARDUINO con i nostri strumenti didattici

Libri in italiano con esercitazioni didattiche e tutti i materiali necessari per imparare a programmare facilmente con la piattaforma Arduino. Particolarmente indicati per corsi scolastici e per quanti affrontano per la prima volta la programmazione di microcontrollori.



Libro "Primi passi con Arduino"

Pensato per i principianti che si avvicinano all'elettronica e per gli hobbisti che non hanno mai affrontato la programmazione. Con dieci esercitazioni didattiche.

cod. ARDUBOOK1
€ 9,90

Libro "Primi passi con ARDUINO " + scheda Arduino UNO R3



cod. ARDUKITBOOK1
€ 29,90



Libro "Primi passi con Arduino" + Starter Kit con Arduino UNO Rev3



cod. ARDUKITBOOK2
€ 56,90



Set composto dal libro "Primi passi con Arduino" e dallo starter kit cod. ARDUKITBOOK comprendente la scheda Arduino UNO Rev3 e tutto il materiale necessario per realizzare gli esperimenti didattici escritti nel libro.

Starter kit con Arduino UNO R3



Comprende la scheda Arduino UNO R3 e tutto il materiale necessario per realizzare gli esperimenti didattici descritti nel volume "Primi passi con ARDUINO".

cod. ARDUKITBOOK
€ 52,00



Arduino UNO Rev3 con ATMEGA328

Arduino UNO R3 e la più recente versione della scheda di sperimentazione basata su ATmega328.

cod. ARDUINOUNOREV3
€ 24,50

NOVITÀ

Dopo il libro "base" ecco la versione "advanced"!



solo **€ 22,00**

"ARDUINO UNO Programmazione avanzata e Librerie di sistema"

Gli approfondimenti e le informazioni destinate a chi vuole utilizzare Arduino per progetti evoluti in un libro di facile consultazione e ricco di spunti. Con dieci esercitazioni didattiche avanzate.

cod. ARDUADVANCED

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

Caratteristiche tecniche di questi prodotti e acquisti on-line su *www.futurashop.it*

Prezzi IVA inclusa

Maker Faire

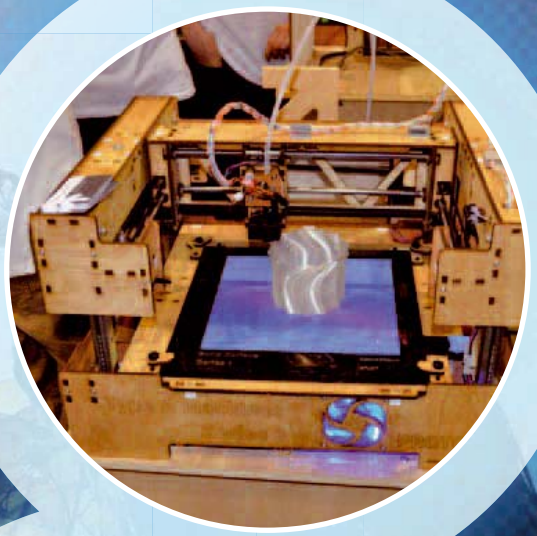


La festa del DIY

Un sole estivo ha accolto gli oltre 100 mila visitatori che hanno affollato nel week-end del 19-20 maggio la cittadina della Bay Area per la settima edizione della Maker Faire, l'evento dedicato a makers, innovatori e comunità di DIY.

di ARSENIO SPADONI

Protagoniste indiscusse dell'edizione di quest'anno della Maker Faire di San Mateo sono state le stampanti 3D che dovrebbero rappresentare il fulcro di quella rivoluzione industriale che secondo molti è alle porte e che cambierà radicalmente il modo di produrre le merci: non più pochi mercati con milioni di consumatori ma un'infinità di piccoli produttori, delocalizzati che, grazie anche ad internet, saranno in grado di







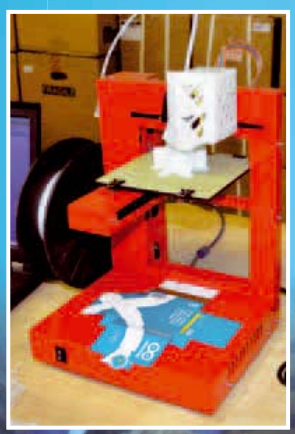


soddisfare al meglio le proprie esigenze e quelle dei propri clienti. Una rivoluzione basata anche sulla condivisione, che vede migliaia di appassionati non solo scambiarsi informazioni sugli argomenti più vari, ma anche e soprattutto sviluppare progetti in comune che, grazie all'esperienza ed alla competenza di ciascuno, anche se piccola, non possono che produrre risultati spesso entusiasmanti. Al centro di questo nuovo interesse c'è principalmente l'elettronica, la robotica e la meccatronica, ma anche attività meno



tecnologiche come il bricolage domestico o il modellismo statico. Sono attività e interessi che, in ogni caso, coinvolgono valori più profondi, dall'impiego più consapevole delle risorse, all'utilizzo più intelligente delle tecnologie. Attività che spesso prevedono una componente manuale di un certo rilievo, da qui il nome di Maker Faire, ovvero la fiera di coloro che "fanno" e che spesso sono in grado anche di innovare in maniera significativa. Non a caso la Maker Faire è nata in quell'area di poche miglia che va da San Francisco a San Jose, dove sono sorte e si sono sviluppate la maggior parte di quelle società di elettronica e informatica le cui tecnologie stanno cambiando il nostro modo di vivere e lavorare, da Google a Intel, da eBay ad HP, da Facebook ad Apple. E come nel caso di queste aziende, anche la formula della Maker Faire ha avuto in pochi anni un successo clamoroso con edizioni che sono sorte in numerose altre località degli Stati Uniti e del resto del mondo, Africa compresa. Anche nel nostro paese, dove qualcuno sta cercando di realizzare delle Mini Maker Faire, sembra che ci sarà, nel marzo del 2013, la prima edizione europea (ed italiana) della Maker Faire, probabilmente nei padiglioni della Fiera di Roma, col supporto di alcuni enti pubblici locali e nazionali e di alcune realtà che di questo movimento sono in un certo senso i "paladini" (Arduino e altri).

Tornando all'edizione di quest'anno di San Mateo, si sono moltiplicate le presentazioni di prodotti ed iniziative di vari genere, specie nel campo dell'educational, con la















partecipazione di personaggi del calibro di Eben Upton (Raspberry Pi), Massimo Banzi (Arduino), Chris Anderson (DIY Drones) e Dale Daugherty, il fondatore della Maker Faire. Durante il suo speech Massimo Banzi ha presentato i nuovi prodotti, dalla scheda low cost Leonardo (già disponibile) alla Wi-Fi Shield (disponibile tra qualche settimana) ad Arduino Due, di cui sono state distribuite un centinaio di board di test tra i membri più attivi della community. Nel complesso un evento che, nonostante le nuove "attrazioni" fossero pochine, ha sicuramente pienamente soddisfatto gli oltre 100 mila visitatori che hanno affollato l'area espositiva di San Mateo, in un week-end caratterizzato da un sole già estivo, specie nella giornata di domenica.

Venerdì sera, invece, la festa nella festa, col party sponsorizzato da Intel per i makers presenti e con le prove generali dei draghi "sparafuoco" al ritmo del gruppo Rock The Bike. Un tipo di attività (quello delle attrazioni di fuoco) difficilmente importabile in Italia, non solo per motivi di sicurezza, ma anche perché tipicamente americano e molto "lontano" dai gusti del vecchio continente.

Così come tipicamente americani erano molti personaggi presenti con le proprie attrazioni - più o meno tecnologiche - ma sempre in grado di suscitare l'interesse del pubblico.

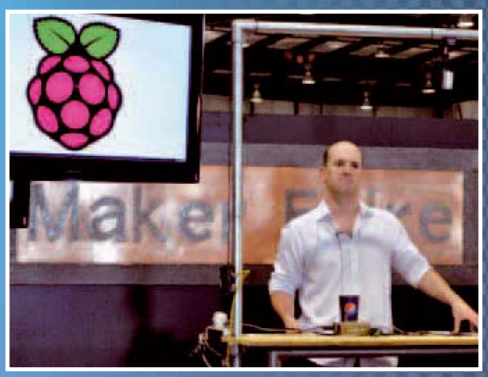


Una Maker Faire anche in Italia



Una Maker Faire anche in Italia? Sembra proprio di sì. Non c'è ancora la certezza ma sembra che sarà proprio Roma nel marzo del 2013 ad ospitare la prima edizione italiana ed europea della manifestazione dedicata ai makers nata sette anni fa a San Mateo, una cittadina a sud di San Francisco su iniziativa di Dale Daugherty di Make.





Per te mai più segreti!

Nuovi dispositivi professionali per misure e analisi precise.



Analizzatore di componenti DCA55

Indispensabile in laboratorio, consente di analizzare una vasta gamma di componenti attivi, identificando la pinout e fornendo informazioni relative ai principali parametri.

€75,00
cod. DCA55



Componenti analizzabili:





Multimetro digitale con USB - 6600 conteggi

€124,00
cod. DVM4200

Strumento professionale con interfaccia USB e display LCD retroilluminato collegabile al PC per registrare, elaborare e stampare i dati misurati. Dispone di selezione manuale/automatica delle portate, misura corrente (10A) e tensione (1000V), resistenza (66Mohm), capacità (66mF), frequenza (66MHz), temperatura (1000°C) ed esegue test su diodi e di continuità elettrica. Completo di software "PC-LINK", custodia e vari accessori.

Disponibile anche nelle versioni:

Multifunzioni 6000 conteggi



Dotato di porta USB, misura tensione (1000V), corrente (10A), resistenza (60Mohm), capacità (300µF), frequenza (999,9kHz) e temperatura (1000°C).

€89,00
cod. DVM4100

Stesse caratteristiche del DVM4100 ma senza porta USB.

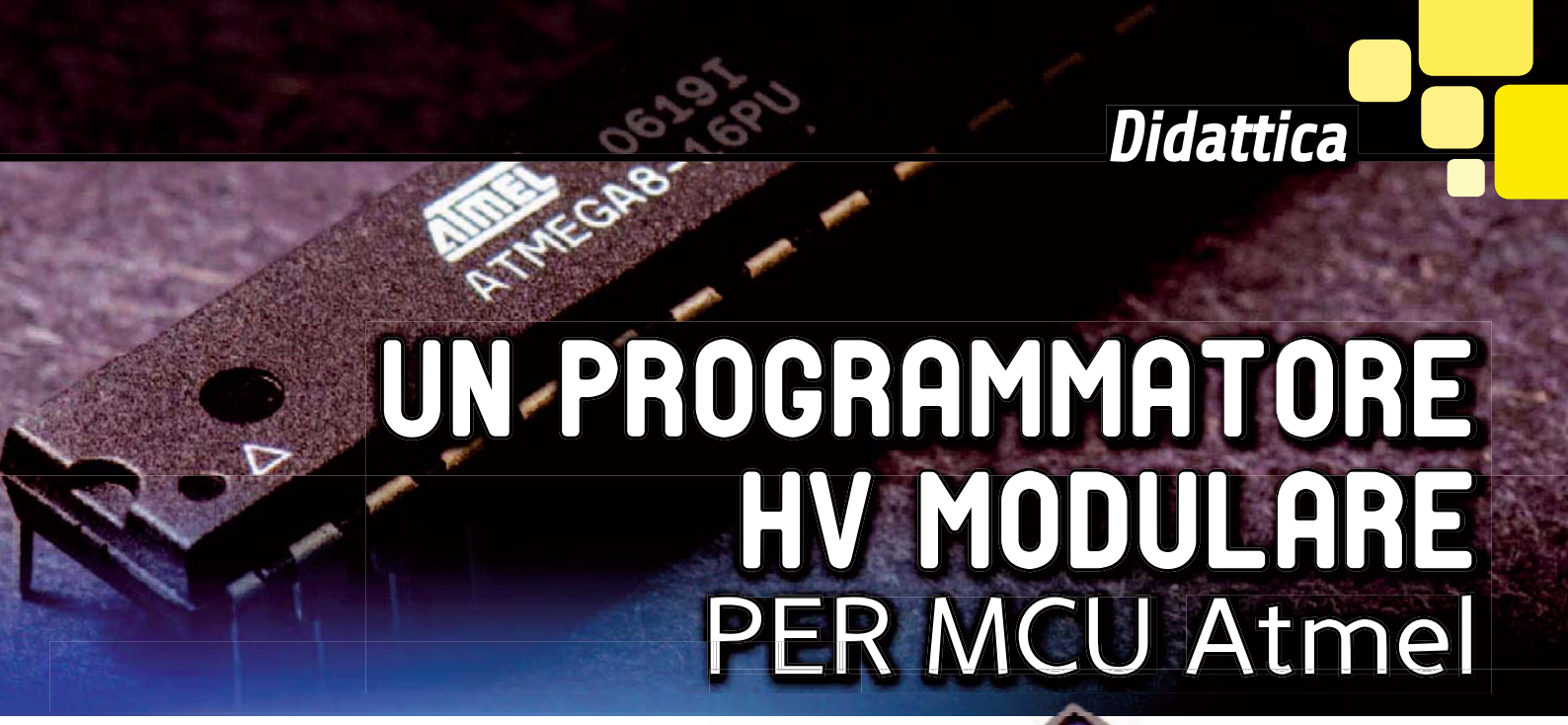
€67,00
cod. DVM4000

FUTURA ELETTRONICA
www.futurashop.it

Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775 • Fax. 0331/792287

Caratteristiche tecniche di questi prodotti e acquisti on-line su *www.futurashop.it*

Prezzi IVA inclusa



UN PROGRAMMATORE HV MODULARE PER MCU Atmel

Dopo avere costruito il programmatore HV affrontiamo i concetti generali della programmazione e descriviamo il firmware della sezione Master. Terza parte.

del prof. MICHELE MENNITI



Nella puntata precedente (numero 166 di Maggio) vi abbiamo presentato la terza ed ultima sezione hardware del nostro Programmatore HV (*Keypad & Display* e relativo firmware), ed il montaggio in un mobile idoneo a contenere lo strumento in modalità stand-alone completa.
In questo numero tratteremo tutta la problematica relativa agli elementi coinvolti nella Programmazione HV, quindi vi presenteremo il firmware per la sezione Master, mostrandovi come esso sia derivato dallo studio approfondito di svariati data-sheet.
Iniziamo col riepilogare le caratteristiche tecniche del nostro Programmatore in entrambe le modalità operative: stand-alone e PC (vedi box **Caratteristiche tecniche del Programmatore HV**).

ELEMENTI TECNICI DELLE MCU ATMEL

Prima di spiegare le tecniche e le fasi della programmazione, è necessario un approfondimento sugli elementi coinvolti: la *signature*, i *fuse bits* ed il *lock bit byte*.

La ***signature*** è costituita da tre byte scritti in un'area protetta della memoria Flash ed è accessibile esclusivamente in lettura mediante specifici comandi di programmazione; rappresenta un identificativo univoco del modello di MCU. Per esempio la signature dell'ATmega328P è <1E 95 0F>. Il byte più significativo è sempre "1E", un valore identico per tutte le signature dei microcontrollori ATMEL. Il secondo byte indica la quantità di memoria Flash: 90=1kB, 91=2kB, 92=4kB, 93=8kB, 94=16kB, 95=32kB, 96=64kB, 97=128kB, 98=256kB. La combinazione tra il secondo byte e quello meno

Caratteristiche tecniche del Programmatore HV

- **Alimentazione: 5V**
  Consumo max: 300mA
  Visualizzazione dati
- **stand-alone: Display LCD 4x20 retroilluminato**
- **PC: su Monitor mediante programma di emulazione terminale seriale**
  Inserimento dati:
- **stand-alone: Keypad esadecimale a matrice**
- **PC: tastiera standard**
  MCU programmabili: famiglie ATMEL in case PDIP con pin 40W, 28, 20, 14, 8.
  Tipi di programmazione: HVPP (con variante per tiny) e HVSP
  Operazioni di lettura: Lock bits, Fuse bits, Signature
  Operazioni di scrittura: Lock bits, Fuse bits (valori di default o personalizzati), Chip Erase
  Inserimento dei valori di scrittura direttamente in formato esadecimale.
  Indicazione automatica modello di MCU sotto test e selezione automatica del tipo di programmazione specifica

significativo invece rappresenta univocamente l'MCU. Per esempio 950F rappresenta l'ATmega328P; infatti una variante di questo microcontrollore, ATmega328 (senza la P finale) è rappresentata da 9514. La signature è una delle informazioni delle MCU memorizzate nel file *avrdude.conf*. Quando si programma un micro mediante la tecnica ISP, il programmatore legge la signature del micro e la confronta con quella memorizzata nel file avrdude.conf, solo se combaciano procede con la programmazione.

I ***fuse bits*** sono dei valori che possono essere programmati per variare una serie di impostazioni del microcontrollore. Tutti i microcontrollori ATMEL hanno i fuse bits organizzati in due byte: *Low Fuse Byte* e *High Fuse Byte*, molti hanno anche un terzo elemento, l'*Extended Fuse Byte*. Da questo momento in poi faremo sempre riferimento ai Fuse Bytes del microcontrollore ATmega328P, il cuore degli Arduino Duemilanove e UNO; quindi rispetto a quanto leggerete potreste riscontrare qualche differenza nei data-sheet di altri modelli, ma queste informazioni possono essere ritenute standard. Il Low Fuse Byte serve fondamentalmente per il settaggio del clock operativo del micro. l'High Fuse Byte serve per una serie di importanti parametri quali la gestione del reset esterno, la programmazione SPI, la protezione dell'EEprom, l'area di memoria Flash da riservare al bootloader, il vettore di reset, e altri. L'Extended Fuse Byte in questo caso non è usato completamente, sono interessati solo i primi tre bit, e serve per impostare il livello di trigger del Brown-out Detector.

In **Fig. 1** riportiamo la descrizione ed i valori di default dei bit dei tre Fuse Byte dell'ATmega328P; il valore "1" si intende come "non programmato", mentre il valore "0" si intende come "programmato", quindi per attivare un determinato bit bisogna settarlo a "0".

Iniziamo col comprendere come si calcolano i valori dei byte in modo da poterli poi inserire al momento di usare il nostro Programmatore HV. Per comodità usiamo i valori di default, che sono proprio quelli riportati in **Fig. 1** (in rosso le combinazioni di default dei bit). Scrivendo i suddetti valori in sequen-

Low Fuse Byte	N° Bit	Descrizione	Valore Default 0=abilitato 1=disabilitato
CKDIV8	7	Divide il clock x8	0
CKOUT	6	Riporta il clock sul pin 14	1
SUT1	5	Impostano un ritardo all'avvio della CPU in seguito al rientro dallo sleep o ad un RESET: 00=6ck+14ck - 01=6ck+14ck+4.1ms - 10=6ck+14ck+65ms – 11=reserved	1
SUT0	4		0
CKSEL3	3	Impostano la modalità di clock della CPU, p.es: 0000=clock esterno (quarzo sui pin 9 e 10) 0010=oscillatore interno a 8MHz (default) 0011=oscillatore interno a 128KHz 1010=quarzo esterno da 1 a 3MHz 1100=quarzo esterno da 3 a 8MHz 1110=quarzo esterno a 8MHz	0
CKSEL2	2		0
CKSEL1	1		1
CKSEL0	0		0
High Fuse Byte	**N° Bit**	**Descrizione**	**Valore Default 0=abilitato 1=disabilitato**
RSTDISBL	7	Disabilitazione reset esterno	1
DWEN	6	Abilitazione debugWIRE	1
SPIEN	5	Abilitazione programmazione seriale (SPI)	0
WDTON	4	Timer WatchDog sempre attivo	1
EESAVE	3	Protezione EEprom da Chip Erase	1
BOOTSZ1	2	Impostazione dello spazio per il bootloader: 11=256 words=512Byte 10=512 words=1024Byte 01=1024 words=2048Byte 00=2048 words=4096Byte (default)	0
BOOTSZ0	1		0
BOOTRST	0	Selezione del vettore di reset: 1= Sezione Application (0x0000) 0= Sezione Bootloader	1
Extended Fuse Byte	**N° Bit**	**Descrizione**	**Valore Default 0=abilitato 1=disabilitato**
-	7	-	-
-	6	-	-
-	5	-	-
-	4	-	-
-	3	-	-
BODLEVEL2	2	Livello di trigger del Brown-out: 111= BOD Disabled 110= 1,8V 101= 2,7V 100= 4,3V 0XX= reserved	1
BODLEVEL1	1		1
BODLEVEL0	0		1

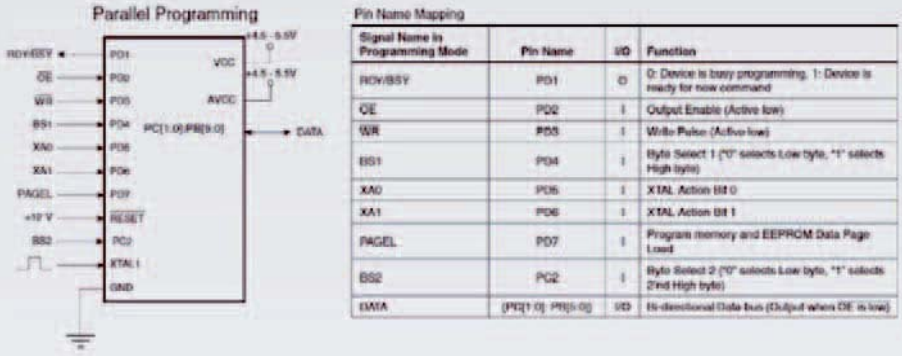
Fig. 1 *- I Fuse Byte dell'ATmega328P*

Lock Bit Byte	N° Bit	Descrizione	Valore Default 0=abilitato 1=disabilitato
-	7	-	-
-	6	-	-
BLB12	5	Boot Lock Bits per la protezione della Sezione Bootloader della memoria Flash: 11= protezione disabilitata 10= protezione in scrittura 01= protezione della lettura da parte dell'applicazione 00= protezione in scrittura e della lettura del bootloader da parte dell'applicazione	1
BLB11	4		1
BLB02	3	Boot Lock Bits per la protezione della Sezione Application della memoria Flash: 11= protezione disabilitata 10= protezione in scrittura 01= protezione della lettura da parte del bootloader 00= protezione in scrittura e della lettura dell'applicazione da parte del bootloader	1
BLB01	2		1
LB1	1	Lock Bits per la protezione della memoria Flash e EEPROM: 11= protezione disabilitata 10= protezione in scrittura e blocco dei fuse bits 01= reserved 00= protezione in scrittura e lettura e blocco dei fuse bits e dei boot lock bits	1
LB0	0		1

Fig. 2 *- Il Lock Bit Byte dell'ATmega328P*

za (a partire dal bit 7) si ottiene che il Low Fuse Byte è 01100010 che in esadecimale diventa "62"; l'High Fuse Byte è 11011001 che in esadecimale diventa "D9"; infine l'Extended Fuse Byte è 11111111 (i bit non usati si mettono a 1) che in esadecimale diventa "FF". Quindi i fuse di default del micro ATmega328P sono: 62, D9, FF.
Vediamo a cosa corrispondono i settaggi di default:
Il ***Low Fuse Byte*** ci dice che il clock è settato per l'oscillatore interno a 8MHz (CKSEL3÷0 su 0010), ma tale valore è diviso per 8 (CKDIV8 a 0), quindi il micro è settato di fabbrica per lavorare con l'oscillatore interno a 1MHz. Inoltre i bit SUT1-0, settati su 10, ad ogni impulso di reset fanno seguire 6+14 cicli di clock ed un ritardo di 65ms.
L'***High Fuse Byte*** abilita la programmazione seriale (SPI enabled) e riserva al bootloader uno spazio di 2048 Word, corrispondenti a 4kB (BOOTSZ1-0 su 00) a patto di settare a 0 il fuse BOOTRST. Degni di una nota particolare sono due bit dell'High Fuse Byte, settati di default a 1, quindi disabilitati; si tratta del DWEN (DebugWire Enabled) e del WDTON (WatchDogTimer ON): il primo serve per attivare un potente sistema di Debug che sfrutta una sola linea (Wire), quella del pin RESET del micro; il secondo invece abilita il WatchDog, un sistema di sicurezza che, se attivato, riesce a resettare il micro e a far ripartire il suo programma, in caso di crash dovuti a cause esterne, molto spesso disturbi sull'alimentazione o di natura elettromagnetica.
L'***Extended Fuse Byte*** non ha alcun bit settato a 0, ciò significa che il Brown-out Detector è disattivo. In pratica, con il Brown-out Detector è possibile impostare una soglia di tensione di alimentazione, al di sotto della quale il micro viene bloccato nello stato di reset; la soglia viene fissata in base alla combinazione dei tre bit BODLEVEL2÷0 (v. valori in **Fig. 1**) oppure il BOD può essere disabilitato, in tal modo non avviene alcun controllo sulla tensione di alimentazione del micro. Il ***Lock Bit Byte***, opportunamente settato, serve per proteggere da scrittura o lettura/scrittura le due sezioni (Application e Bootloader) della memoria Flash e la memoria EEPROM interna al micro, ed è così organizzato: i primi due bit (0 e 1) si chiamano ***lock bits***, i successivi quattro bit (2÷5) si chiamano ***boot lock bits***. I lock bits agiscono sull'intera memoria Flash e sulla EEPROM mentre i boot lock bits agiscono in modo separato sulle due sezioni Application e Bootloader della memoria Flash. Inoltre i lock bits si usano per bloccare (e quindi impedirne la modifica dello stato) sia i boot lock bits che i fuse bits.
In **Fig. 2** riportiamo l'intera tabella del Lock Bit Byte e le configurazioni di protezione dei lock bits.
Dopo aver letto attentamente le descrizioni possiamo provare a riepilogare le funzioni delle tre coppie di bits:

- BLB12-11: agiscono esclusivamente sulla sezione Bootloader della memoria Flash, e la loro azione è di tipo "interna", cioè la protezione riguarda il tentativo di operazioni eseguite dal firmware. Per cui il settaggio "10" significa che se il firmware caricato nel micro tentasse di scrivere nell'area riservata al Bootloader, l'operazione verrebbe bloccata da questa coppia di boot lock bits. Il settaggio "01" invece blocca i tentativi di lettura dell'area Bootloader da parte del software presente nell'area Application;
- BLB02-01: agiscono esclusivamente sulla sezione Application della memoria Flash, ed anche in questo caso abbiamo una protezione interna, quindi riferita al firmware.Il settaggio "10" significa che se il firmware caricato nel micro tentasse di scrivere nell'area Application, l'operazione verrebbe bloccata da questa coppia di boot lock bits. Il settaggio "01" invece blocca i tentativi di lettura dell'area Application da parte del software presente nell'area Bootloader;
- LB1-0: agiscono sull'intera memoria Flash e sulla memoria EEPROM, e la protezione riguarda gli accessi dall'esterno, quindi tentativi di scrittura e/o lettura fatti tramite un programmatore o Arduino impostato come tale; da notare che la protezione è efficace sia in modalità HVPP che in modalità HVSP. Contestualmente alla protezione della memoria viene impostata la protezione sui boot lock bits e sui fuse bits. Così il settaggio "10" blocca i tentativi di scrittura sulle memorie Flash e EEPROM e blocca anche possibili modifiche dei tre Fuse Byte, mentre il settaggio "00" blocca, oltre ai tentativi di scrittura, anche quelli di lettura, inoltre blocca sia i tre Fuse Byte che i quattro boot lock bits.



Pin Name Mapping

Signal Name in Programming Mode	Pin Name	I/O	Function
RDY/BSY	PD1	O	0: Device is busy programming, 1: Device is ready for new command
OE	PD2	I	Output Enable (Active low)
WR	PD3	I	Write Pulse (Active low)
BS1	PD4	I	Byte Select 1 ("0" selects Low byte, "1" selects High byte)
XA0	PD5	I	XTAL Action Bit 0
XA1	PD6	I	XTAL Action Bit 1
PAGEL	PD7	I	Program memory and EEPROM Data Page Load
BS2	PC2	I	Byte Select 2 ("0" selects Low byte, "1" selects 2nd High byte)
DATA	[PB[5:0], PC[1:0]]	I/O	Bi-directional Data bus (Output when OE is low)

***Fig. 3** - I segnali HVPP dell'ATmega328P*

A questo punto è assolutamente normale chiedersi se ci sia un modo per sbloccare una MCU a cui sia stata applicata la massima protezione, e cioè LB1-0 su "00". La risposta è affermativa: la programmazione dei microcontrollori ATMEL prevede un comando, che si chiama "Chip Erase" che, una volta eseguito, cancella irrimediabilmente il contenuto delle memorie Flash e EEPROM e resetta il Lock Bit Byte, impostando tutti i suoi bit a "1"; non vengono invece resettati i Fuse Byte che, però, a quel punto sono facilmente modificabili. Quando vi illustreremo l'uso dello strumento vi spiegheremo come settare il Lock Bit Byte ed i Fuse Byte per ottenere la protezione totale del micro, sia in lettura che in scrittura, utili soprattutto se vorrete proteggere il vostro firmware, sia da scritture indesiderate che da tentativi di copia. Infatti è vero che il Chip Erase (che peraltro è una delle funzioni presenti nel nostro Programmatore HV) è in grado di sbloccare il Lock Bit Byte, ma è altrettanto vero che contestualmente cancella la memoria del micro che sblocca, quindi la protezione è garantita.
Il Lock Bit Byte si programma naturalmente come i Fuse Byte, quindi una volta settate le tre coppie di bit, esse vanno prese nel loro insieme trasformando il byte da valore binario a valore esadecimale. Ad esempio, se decidete di impostare la massima protezione esterna, senza agire su quella interna, i sei bit diventano: 111100, poiché i bit 7-6 non sono usati vengono settati a 1 (non programmati), quindi il Lock Bit Byte diventa: 11111100 che, trasformato in esadecimale, vale "FC"; quindi il valore da inserire nel Programmatore HV sarà appunto FC.

LA PROGRAMMAZIONE HV

Abbiamo già dato la definizione generale di Programmazione HV, spiegando che la sigla sta per High Voltage, in riferimento alla tensione di 12V, molto più elevata rispetto all'alimentazione standard di 5V, che viene applicata durante le varie operazioni al pin di RESET del microcontrollore target.
La Programmazione HV si può eseguire in base a due differenti modalità, in funzione del modello di MCU da programmare: la modalità HVPP (High Voltage Parallel Programming), con la quale il dialogo tra lo strumento Programmatore ed il micro target avviene mediante trasferimento di un byte per volta, usando le 8 linee del bus "DATA"; e la modalità HVSP (High Voltage Serial Programming), in cui il dialogo tra lo strumento Programmatore ed il micro target avviene mediante trasferimento di un bit per volta, tramite le linee seriali SDI, SII, SDO. Più in generale la modalità HVSP viene adottata per programmare MCU con package PDIP (nel caso del nostro Programmatore HV) a 8 e 14 pin, mentre la modalità HVPP si adotta per le MCU a 20, 28 e 40W pin. Nell'ambito della programmazione HVPP si riscontrano delle piccole differenze tra le MCU a 20 e quelle a 28 e 40W pin, in realtà riguardano solo un segnale, per gestire il quale abbiamo dovuto scrivere delle istruzioni specifiche che abbiamo deciso di identificare con la sigla HVP13, che non esiste come acronimo ma che per noi rappresenterà, d'ora in poi, la variante dell'HVPP per i modelli di MCU a 20 pin.
Nelle operazioni della Programmazione HV sono coinvolti tutta una serie di segnali che devono essere perfettamente sincronizzati affinché tutto vada a buon fine. La parte forse più importante del nostro lavoro è stata proprio quella di implementare questi segnali che, peraltro, sono molto differenti per le due diverse modalità di programmazione. Spiegare ogni singolo segnale ovviamente sarebbe tanto dispersivo quanto poco utile, quindi ci limiteremo a fornivi le fonti documentali a cui abbiamo fatto riferimento nella fase di studio, mentre nel firmware troverete abbondati spiegazioni effettuate tramite numerose linee di commento.

LA PROGRAMMAZIONE HVPP

I data-sheet a cui abbiamo fatto riferimento per la programmazione HVPP (v. BOX "Riferimenti Bibliografici") sono quelli dell'ATmega328P (micro a 28 pin), dell'ATtiny2313 e ATtiny861 (micro a 20 pin) e dell'ATmega1284P (micro a 40W pin). In **Fig. 3** è possibile vedere tutte le sigle dei segnali coinvolti nella programmazione HVPP dell'ATmega328P, la loro corrispondenza sui pin fisici del micro ed il loro significato e direzione operativa (Input, Output o bidi-

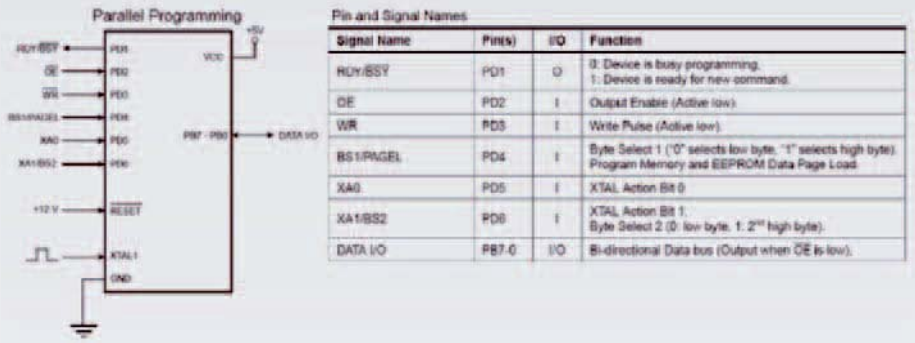


***Fig. 4** - I segnali HVPP dell'ATtiny2313A*

rezionale I/O).
Per entrare nel modo di programmazione HVPP bisogna effettuare una serie di passaggi:

1. Settare i pin PAGEL, XA1, XA0 e BS1 a 0 mantenendo i pin RESET e VCC a 0V;
2. Applicare 5V tra VCC e GND;
3. Dopo circa 20 - 60µs applicare una tensione tra 11,5V e 12,5V al pin RESET;
4. Mantenere a 0 i pin del punto 1 per almeno 10µs dal momento in cui vengono applicati i 12V al pin RESET;
5. Attendere almeno 300µs prima di inviare i comandi di programmazione;
6. Per uscire dal modo di programmazione HVPP è sufficiente riportare il pin RESET a 0V e quindi togliere l'alimentazione al micro.

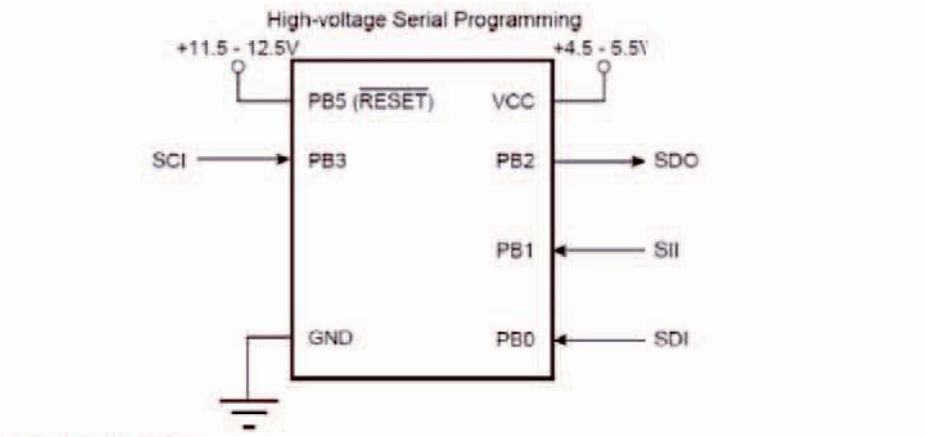
Da notare che nel firmware troverete delle variazioni riguardo i tempi di attesa; questo è dovuto al fatto che ogni micro ha i propri dati specifici, ma noi abbiamo dovuto impostare il firmware tenendo conto dei tempi massimi; p.es. la pausa tra l'applicazione dei 5V a VCC e dei 12V al RESET è di 200µs, diversamente alcuni modelli non si potrebbero programmare.
La coppia di segnali XA1-XA0 (inattiva se settata a 11) serve per gestire le varie fasi della programmazione, in funzione di un impulso che arriva sul pin XTAL1; in pratica quando al pin XTAL1 arriva un impulso positivo di 1ms, in base alla combinazione XA1-XA0 il micro si predispone per:
00 : caricare l'indirizzo della memoria Flash o EEPROM;
01 : caricare i dati sul bus DATA;
10 : caricare il comando.
Il segnale BS1 serve per settare il byte High o Low sia nel caso di caricamento dell'indirizzo che nel caso di caricamento dei dati.
Vediamo l'elenco dei comandi nel box: **Le combinazioni del byte per i comandi della programmazione HV**.
Tutte queste informazioni vi aiuteranno a comprendere perfettamente il funzionamento del firmware, nel momento in cui lo descriveremo. Ribadiamo che le informazioni sono relative all'ATmega328P, ma valgono praticamente per tutti i modelli di MCU a 28 e 40W pin, salvo piccole variazioni che comunque sono correttamente gestite via software. Inoltre esse valgono anche per le MCU a 20 pin, solo che in questo caso il segnale PAGEL viene ad essere sostituito dal BS1 ed il BS2 viene sostituito da XA1, ciò a causa della minore disponibilità di pin; anche questo aspetto verrà approfondito nella spiegazione del firmware; per il momento sarà sufficiente confrontare la **Fig. 4** con la **Fig. 3**, in modo da comprendere come sono distribuiti i segnali nell'ATtiny2313A (20 pin) rispetto all'ATmega328P (28 pin).

LA PROGRAMMAZIONE HVSP

Affrontiamo ora la questione HVSP, molto differente dalla HVPP, in quanto il fatto che i dati vengano trasmessi ad un bit per volta complica notevolmente questa tecnica, come vedremo. In questo caso il nostro riferimento diventa il data-sheet ATMEL dell'ATtiny85 (per le versioni a 8 pin) e dell'ATtiny84 (per le versioni a 14pin). Come è possibile vedere in **Fig. 5**, i segnali coinvolti sono quattro: SDI (Input Dati), SDO (Output Dati), SII (Input Istruzioni) e SCI (Clock Input), quindi indirizzi, dati e comandi dovranno viaggiare usando esclusivamente le tre linee dedicate.
Per entrare nel modo di programmazione HVSP bisogna

Le combinazioni del byte per i comandi della programmazione HV

10000000 : Chip Erase (cancellazione totale della memoria Flash e EEPROM e reset del Lock Bit Byte)
01000000 : Write Fuse bits (scrittura dei bit dei tre Fuse Byte)
00100000 : Write Lock bits (scrittura dei bit del Lock Bit Byte)
00010000 : Write Flash (scrittura sulla memoria Flash)
00010001 : Write EEPROM (scrittura sulla memoria EEPROM)
00001000 : Read Signature Bytes and Calibration byte (lettura dei tre byte della signature e del Byte della Calibrazione dell'oscillatore)
00000100 : Read Fuse and Lock bits (lettura dei tre Fuse Byte e del Lock Bit)
00000010 : Read Flash (lettura della memoria Flash)
00000011 : Read EEPROM (lettura della memoria EEPROM)



Pin Name Mapping

Signal Name in High-voltage Serial Programming Mode	Pin Name	I/O	Function
SDI	PB0	I	Serial Data nput
SII	PB1	I	Serial Instruction Input
SDO	PB2	O	Serial Data Output
SCI	PB3	I	Serial Clock input (min. 220ns period)

Fig. 5 *- I segnali HVSP dell'ATtiny85*

effettuare questi passaggi:

1. Settare i pin SDI, SDO e SII a 0, mantenendo i pin RESET e VCC a 0V;
2. Applicare 5V tra VCC e GND;
3. Dopo circa 20 - 60µs applicare una tensione tra 11,5V e 12,5V al pin RESET;
4. Mantenere a 0 i pin del punto 1 per almeno 10µs dal momento in cui vengono applicati i 12V al pin RESET;
5. Per evitare conflitti subito dopo settare SDO come Input;
6. Attendere almeno 300µs prima di inviare le istruzioni seriali su SDI/SII;
7. Per uscire dal modo di programmazione HVSP è sufficiente riportare il pin RESET a 0V e quindi togliere l'alimentazione al micro.

In **Fig. 6** abbiamo riportato una selezione dei comandi di programmazione HVSP, chi volesse vederli tutti dovrà fare riferimento alla Tabella originale presente sul data-sheet dell'ATtiny85; questi sono praticamente tutti quelli integrati nel nostro firmware. A prima vista sembra qualcosa di assolutamente incomprensibile ed in effetti la sua interpretazione è possibile solo se si studia attentamente tutto il testo del data-sheet che riguarda la programmazione HVSP. Comunque, nella fase di spiegazione del firmware imparerete rapidamente a tradurre tutti quei byte in comandi, indirizzi e dati, per ora limitiamoci a comprenderne l'uso in base alla posizione. Come si può notare ogni cella di questa Tabella possiede tre righe; seguendo le intestazioni di colonna in alto e quelle di riga laterali è possibile capire in che momento si devono usare questi valori ed a quali segnali vanno associati. Prendiamo ad esempio la riga "Read Signature Bytes", che useremo per leggere i tre byte della signature del micro. Nella colonna "Instr.1/5" troviamo:

- SDI = 0_0000_1000_00, se prendiamo gli 8 valori interni otteniamo il byte 00001000 che, come descritto nel box: **Le combinazioni del byte per i comandi della programmazione HV**, rappresenta appunto il comando di lettura della signature, che viene inviato, tramite la linea SDI al micro target;
- SII = 0_0100_1100_00, il byte 01001100 rappresenta la prima istruzione da fornire al micro target mediante la linea SII;
- SDO = x_xxxx_xxxx_xx, il significato di un byte rappresentato con xxxxxxxx è che tale segnale non è coinvolto in questa fase.

Nella colonna "Instr.2/6" troviamo:

- SDI = 0_0000_00bb_00, la coppia "bb" significa che per ogni ripetizione del ciclo di istruzioni (i passaggi vanno ripetuti tutti tre volte, una per byte) al posto di queste due lettere si dovrà inserire una coppia di bit che rappresenterà l'indirizzo del byte da leggere, quindi, in questo caso, tramite la linea SDI, viene inviato un "address" al micro target;
- SII = 0_0000_1100_00, il byte 00001100 rappresenta la seconda istruzione da fornire al micro target mediante la linea SII;
- SDO = x_xxxx_xxxx_xx, vale quanto spiegato prima.

Nella colonna "Instr.3" troviamo:

- SDI = 0_0000_0000_00, in questo caso la linea SDI porta al micro target un byte 00000000;
- SII = 0_0110_1000_00, il byte 01101000 rappresenta la terza istruzione da fornire al micro target mediante la linea SII;
- SDO = x_xxxx_xxxx_xx, vale quanto spiegato prima.

Nella colonna "Instr.4" troviamo:

- SDI = 0_0000_0000_00, anche in questo caso la linea SDI porta al micro target un byte 00000000;
- SII = 0_0110_1100_00, il byte 01101100 rappresenta la quarta istruzione da fornire al micro target mediante la linea SII;

Listato 1 – L'organizzazione del firmware

```
Tabella dei microcontrollori programmabili
Assegnazione dei pin per le modalità di programmazione HVPP e HVSP
Definizioni per lettura/scrittura FUSE in HVSP
Definizioni per lettura SIGNATURE in HVSP
Definizioni per CHIP ERASE in HVSP
Definizioni per lettura/scrittura LOCK BITS BYTE in HVSP
Variabili globali

SETUP

Impostazione dei pin usati durante la programmazione

LOOP

Avvio programmazione
Lettura signature in HVSP
Estrapolazione e visualizzazione del tipo di micro e dei suoi parametri
Main Menu
Chiusura programmazione

FUNCTIONS & VOIDS
void begin_programming (inizio programmazione)
void end_programming (fine programmazione)
➢ LETTURA HVSP MODE (routine e funzioni di lettura in HVSP mode)
  o byte signature_read_HVSP (lettura signature)
  o void fuses_read_HVSP (lettura fuses)
  o void lockbits_read_HVSP (lettura lock bits byte)
  o byte read_HVSP_command (invio comandi di lettura)
➢ LETTURA HVPP MODE (routine e funzioni di lettura in HVPP mode)
  o byte signature_read_HVPP (lettura signature)
  o byte S_read_HVPP_command (invio comandi di lettura signature)
  o void fuses_read_HVPP (lettura fuses)
  o void lockbits_read_HVPP (lettura lock bits byte)
  o byte FL_read_HVPP_command (invio comandi di lettura fuses e lock bits byte)
➢ CONFRONTO E VERIFICA
  o void signature_check (controllo di validità della signature del micro target)
  o unsigned char cfr_signature (confronto di 2°-3° byte della signature con i valori in tabella)
  o byte flags_assign (estrapolazione dei flags BURN_EFUSE e mode)
➢ VISUALIZZAZIONE DATI
  o void signature_display (visualizzazione modello e signature micro target)
  o void def_fuses_display (visualizzazione fuse default micro target)
  o void current_fuses_display (visualizzazione fuse correnti micro target)
  o void current_lockbits_display (visualizzazione lock bits byte corrente micro target)
➢ SCRITTURA HVSP MODE (routine di scrittura in HVSP mode)
  o void fuses_write_HVSP (scrittura fuses)
  o void lockbits_write_HVSP (scrittura lock bits byte)
  o void chip_erase_HVSP (chip erase)
  o void write_HVSP_command (abilitazione della modalità di programmazione)
➢ SCRITTURA HVPP MODE (routine di scrittura in HVPP mode)
  o void fuses_write_HVPP (scrittura fuses)
  o void lockbits_write_HVPP (scrittura lock bits byte)
  o void chip_erase_HVPP (chip erase)
  o void F_write_HVPP_command (abilitazione della modalità di programmazione di scrittura fuses)
  o void L_write_HVPP_command (abilitazione della modalità di programmazione lock bits byte)
➢ UTILITIES & ERRORS
  o void send_cmd_HVPP (invio byte di comando)
  o void send_address_HVPP (invio byte di indirizzo)
  o byte hexfb_ask (inserimento manuale di un byte esadecimale)
  o void sclk (impulso di clock in HVSP)
  o void strobe_xtal (impulso XTAL in HVPP)
  o int hex2dec (conversione di un carattere HEX in un numero)
  o void signature_error (errori nella lettura della signature)
```

- SDO = q_qqqq_qqqx_xx, questa volta le lettere "q" significano che la linea SDO conterrà un byte; in effetti al termine della quarta istruzione il micro target restituirà, tramite la linea SDO, il valore di uno dei byte della signature.

Poiché la signature è costituita da tre byte, queste quattro istruzioni dovranno essere ripetute per ognuno dei byte (in realtà come vedremo, comando e prima istruzione saranno inviati una volta sola), cambiando ogni volta l'address e ricevendo contestualmente il relativo byte.

Alcune particolari operazioni (come la scrittura della EEPROM interna del micro target, richiedono ben sei istruzioni, in questi casi l'istruzione 1 viene ripetuta come quinta, e l'istruzione 2 viene ripetuta come sesta, sempre variando alcuni valori; questo è il motivo delle diciture "Instr.1/5"

High-voltage Serial Programming Instruction Set for ATtiny25/45/85

Instruction		Instruction Format				Operation Remarks
		Instr.1/5	Instr.2/6	Instr.3	Instr.4	
Chip Erase	SDI SII SDO	0_1000_0000_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_0100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 x_xxxx_xxxx_xx		Wait after Instr.3 until SDO goes high for the Chip Erase cycle to finish.
Write Fuse Low Bits	SDI SII SDO	0_0100_0000_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_A987_6543_00 0_0010_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_0100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 x_xxxx_xxxx_xx	Wait after Instr. 4 until SDO goes high. Write A - 3 = "0" to program the Fuse bit.
Write Fuse High Bits	SDI SII SDO	0_0100_0000_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_IHGF_EDCB_00 0_0010_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_0100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_1100_00 x_xxxx_xxxx_xx	Wait after Instr. 4 until SDO goes high. Write I - B = "0" to program the Fuse bit.
Write Fuse Extended Bits	SDI SII SDO	0_0100_0000_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_000J_00 0_0010_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_0110_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1110_00 x_xxxx_xxxx_xx	Wait after Instr. 4 until SDO goes high. Write J = "0" to program the Fuse bit.
Write Lock Bits	SDI SII SDO	0_0010_0000_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_0021_00 0_0010_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_0100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 x_xxxx_xxxx_xx	Wait after Instr. 4 until SDO goes high. Write 2 - 1 = "0" to program the Lock bit.
Read Fuse Low Bits	SDI SII SDO	0_0000_0100_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1000_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 A_9876_543x_xx		Reading A - 3 = "0" means the Fuse bit is programmed.
Read Fuse High Bits	SDI SII SDO	0_0000_0100_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_1010_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_1110_00 I_HGFE_DCBx_xx		Reading I - B = "0" means the Fuse bit is programmed.
Read Fuse Extended Bits	SDI SII SDO	0_0000_0100_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1010_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1110_00 x_xxxx_xxJx_xx		Reading J = "0" means the Fuse bit is programmed.
Read Lock Bits	SDI SII SDO	0_0000_0100_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_1000_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_1100_00 x_xxxx_x21x_xx		Reading 2, 1 = "0" means the Lock bit is programmed.
Read Signature Bytes	SDI SII SDO	0_0000_1000_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_00bb_00 0_0000_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1000_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 q_qqqq_qqqx_xx	Repeats Instr 2 4 for each signature byte address.
Read Calibration Byte	SDI SII SDO	0_0000_1000_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0000_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_1000_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_1100_00 p_pppp_pppx_xx	
Load "No Operation" Command	SDI SII SDO	0_0000_0000_00 0_0100_1100_00 x_xxxx_xxxx_xx				

***Fig. 6** - Set di istruzioni per la programmazione HVSP (parziale)*

e "Instr.2/6" presenti nelle intestazioni delle prime due colonne.

IL FIRMWARE DEL PROGRAMMATORE HV

Ed ora passiamo a descrivere la componente fondamentale del nostro Programmatore HV, il firmware. Vi diciamo subito che abbiamo previsto due versioni: una per chi gestirà il programmatore tramite PC e l'altra per chi ha deciso di realizzare la versione stand-alone. I nomi dei file sono rispettivamente: HV_Master_for_PC e HV_Master_for LCD; entrambi sono scaricabili gratuitamente dalla sezione download del sito *www.elettronicain.it*.
Come abbiamo anticipato nei numeri precedenti, l'unica differenza tra i due consiste nella diversa "impaginazione" del menu e dei messaggi video, necessaria in quanto, mentre un emulatore di terminale seriale (PC) ha uno spazio praticamente illimitato, l'LCD da noi usato ha solo quattro righe da venti caratteri ciascuna. Del protocollo di comunicazione Master-LCD abbiamo già abbondantemente parlato sul numero scorso, quindi ora non ci resta che spiegarvi il funzionamento del programma principale. Non dimentichiamo infatti che lo scopo di questi articoli della serie "Didattica" della nostra Rivista è fondamentalmente quello di aumentare le conoscenze dei nostri lettori, ecco perché non ci limitiamo a spiegare il funzionamento del nostro Programmatore HV, bensì approfondiamo ogni aspetto software, così come nelle prime due puntate abbiamo fatto con l'hardware. Per ovvie ragioni di spazio non ci è possibile pubblicare l'intero listato del firmware, trattandosi di circa 1200 righe di codice, quindi se volete seguire in modo efficace le prossime spiegazioni, conviene aprire il firmware con l'IDE oppure stamparlo. Il firmware è stato organizzato in modo estremamente ordinato; per reperire facilmente le varie routine e funzioni citate durante la spiegazione del firmware, potete far riferimento al box **Listato 1**, nel quale riportiamo appunto l'organizzazione del firmware, con la collocazione sequenziale delle varie sezioni, routine e funzioni. Iniziamo dalle definizioni: poiché l'apertura e chiusura della comunicazione seriale viene effettuata svariate volte durante l'uso, abbiamo pensato di prevedere una costante per la velocità così, nel caso si voglia cambiare tale parametro, sarà sufficiente farlo solo sulla riga *#define BAUD 9600*.
La costante TOT deve contenere il numero di microcontrollori "tabellati" nel firmware, aumentato di 1; nel nostro caso abbiamo inserito trenta modelli di MCU, quindi la riga è *#define TOT 31*.
Abbiamo lasciato queste due righe in testa al firmware proprio perché potrebbero essere facilmente soggette a variazioni.
Le tre istruzioni successive rappresentano la tabellazione delle MCU, sono tre array di dati che contengono rispettivamente:

- ATMEL : le descrizioni delle MCU, limitate a tredici caratteri;
- SIGNATURE : i corrispondenti tre byte della Signature di ogni MCU;
- FUSES : i corrispondenti tre FUSE BYTE di ogni MCU, oltre ad un quarto byte con funzione di flag; questo byte ci tornerà utile, dopo aver identificato il modello di micro, per estrapolare le necessarie informazioni di programmazione: presenza o meno dell'Extended Fuse byte e modo di programmazione richiesto. Infatti, come già detto in precedenza, non tutti i modelli di MCU hanno tre Fuse Byte, alcuni hanno solo il Low e l'High Fuse Byte; in questi casi in tabella è stato

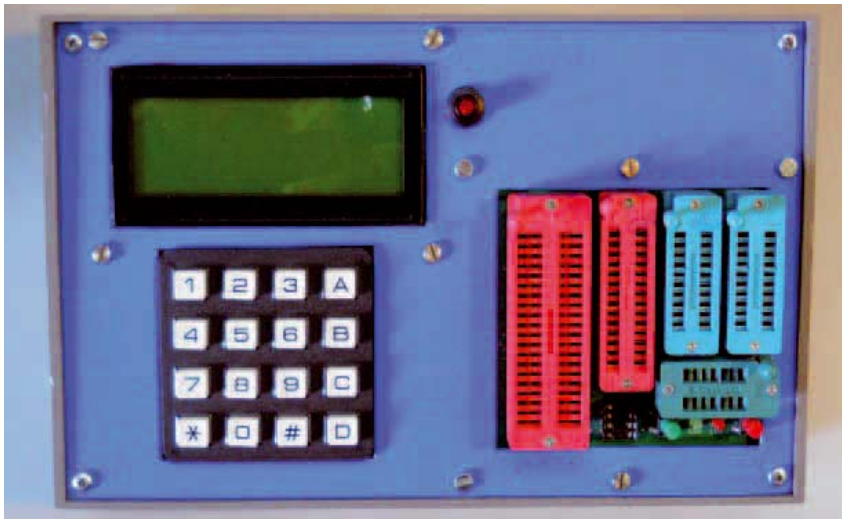
inserito il valore 0xFF per il Fuse mancante, ma poiché tale valore spesso rappresenta realmente quello dell'Extended Fuse, quando questo è presente, abbiamo dovuto distinguere quelli veri da quelli "finti".
Riguardo i modi di programmazione, ormai ben sapete che sono soltanto due: HVPP e HVSP, però abbiamo anche accennato al fatto che i modelli a 20 pin necessitano di una variante software, in quanto non dispongono di alcuni segnali che, come vedrete più avanti, vengono riassegnati su altri pin; questa variante l'abbiamo definita HVP13, un termine che abbiamo coniato per l'occasione, usando le prime tre lettere di HVPP e aggiungendo le due cifre finali dell'ATtiny2313, uno dei modelli usati per sviluppare il firmware.

In pratica questo quarto byte fornisce, in base al suo valore, le seguenti informazioni:
00 = NoEFuse e HVSP;
01 = NoEFuse e HVPP;
02 = NoEfuse e HVP13;
03 = EFuse e HVSP;
04 = EFuse e HVPP;
05 = EFuse e HVP13.

La sigla "NoEFuse" indica appunto la mancanza dell'Extended Fuse Byte nella relativa MCU, il resto è facilmente intuibile. È molto importante comprendere la logica usata in questi array, perché se deciderete di aggiungere qualche modello, solo inserendo tutte le corrette informazioni avrete certezza di poterlo programmare senza problemi.
Le successive definizioni sono quelle che assegnano i vari segnali della programmazione (distinti in HVPP/HVP13 e HVSP) ai corrispondenti pin dell'ATmega328P usato nella sezione Master, quello che conterrà questo firmware. Nella prima puntata (v. numero 165) vi abbiamo mostrato gli schemi elettrici delle sezioni Master e Socket, in essi troverete le corrispondenze dei vari segnali, così come li abbiamo definiti nel firmware, seguendo puntigliosamente le indicazioni dei vari datasheet che abbiamo consultato. In queste definizioni sono presenti due sigle che non avete incontrato finora: si tratta del controllo di avvio della programmazione, tramite uno dei due pulsanti presenti sul PCB Socket; GO è il pin che controlla lo stato del pulsante e *V12EN* genera la tensione necessaria per attivare lo switch elettronico che porta i 12V sul pin RESET del micro target. A questo punto inizia l'elenco delle definizioni per assegnare un nome ad ognuno dei valori di byte che serviranno per la programmazione HVSP, sia in lettura che in scrittura; tali costanti hanno un nome mnemonico e sono organizzate per elemento:

- Lettura/Scrittura del Low Fuse Byte;
- Lettura/Scrittura dell'High Fuse Byte;
- Lettura/Scrittura dell'Extended Fuse Byte;
- Lettura dei tre byte della Signature;
- Chip Erase;
- Lettura/Scrittura del Lock Bit Byte.

Come vedrete la programmazione HVSP richiede prima di tutto l'invio al micro target di un byte di comando, per comunicare qual è l'operazione da fare; questo byte lo abbiamo definito "DATA". Quindi seguono un certo numero di byte di istruzioni (tre o quattro, secondo i casi), che abbiamo definito "INSTR1…4". Nel caso specifico della Signature, trattandosi di tre byte per un'unica informazione, è necessario specificare anche gli indirizzi di ogni singolo byte; a tale scopo abbiamo inserito i byte definiti "ADDR1…3".
È facile notare come i byte "DATA" della lettura dei tre Fuse Byte e del Lock Bit Byte siano identici tra loro, così come identici sono i tre byte della scrittura dei tre Fuse Byte, mentre è diverso quello della scrittura del Lock Bit Byte. Quindi esiste un unico comando per avviare un'operazione di lettura di un Fuse Byte o del Lock Bit Byte, ed è costituito dal byte "00000100", poi sono i successivi byte di istruzioni che stabiliscono quale informazione prelevare dal micro target. Lo stesso discorso vale per la scrit-



Riferimenti bibliografici

Rev. 8271D–AVR–05/11 data-sheet ATmega48A/PA/88A/PA/168A/PA/328/P:
http://www.atmel.com/Images/doc8271.pdf
Rev. 8246B–AVR–09/11 data-sheet ATtiny2313A/4313:
http://www.atmel.com/Images/doc8246.pdf
Rev. 8197C–AVR–05/11 data-sheet ATtiny261A/461A/861A:
http://www.atmel.com/Images/doc8197.pdf
Rev. 8272C–AVR–06/11 data-sheet ATmega164A/PA/324A/PA/644A/PA/1284/P:
http://www.atmel.com/Images/doc8272.pdf
Rev. 2586N–AVR–04/11 data-sheet ATtiny25/45/85:
http://www.atmel.com/Images/doc2586.pdf
Rev. 8183E–AVR–01/12 data-sheet ATtiny24A/44A/84A:
http://www.atmel.com/Images/doc8183.pdf

tura dei tre Fuse Byte, mentre per la scrittura del Lock Bit Byte il comando è specifico.
In pratica la sequenza DATA→INSTR1…INSTRX serve per fornire al micro target tutte le istruzioni adatte a chiedere un valore che esso ha memorizzato oppure a memorizzare un valore fornito dall'utente del Programmatore HV, ma questi passaggi li approfondiremo nella prossima puntata.
Le successive righe del firmware servono per attribuire un valore numerico progressivo ad alcune costanti e per dichiarare una serie di variabili globali, che saranno usate in varie parti del programma. A questo punto termina la sezione delle definizioni ed inizia quella del Setup del firmware; qui semplicemente vengono resettati ed impostati come ingressi tutti i pin del micro della sezione *Master*, delegati a funzione di bus "DATA"; quindi vengono settati i vari pin di segnale come INPUT o OUTPUT, in base alla loro funzione; infine vengono spenti i due canali di alimentazione: l'abilitazione dei 12V ed il VCC del micro target.
Con la sezione Loop del firmware si entra nella parte operativa del programma, quella che vi consentirà di eseguire tutte le opzioni previste dal nostro Programmatore HV.
Questa sezione inizia con l'apertura della comunicazione seriale verso il programma che deciderete di usare; il comando "*Serial.print*("\033[2J");" è specifico per programmi come l'HyperTerminal di Windows e serve per dare un clear, quindi per svuotare la schermata; sul Serial Monitor non ha alcun effetto quindi se decidete di usare il programma in dotazione all'Ide di Arduino potete eliminare questa riga.
Il primo messaggio operativo chiede l'inserimento del micro target sul relativo zoccolo e la pressione del pulsante GO (posto sulla scheda *Socket*). Per evitare comportamenti strani durante la pressione del pulsante abbiamo inserito nel firmware un semplice controllo di debounce, cioè un ritardo che serve per ignorare i rimbalzi tipici delle commutazioni meccaniche di qualsiasi natura.
Il comando "begin_programming();" fa eseguire il contenuto della relativa routine, che è la prima che trovate nella sezione del firmware denominata "FUNCTIONS & VOIDS"; tale routine predispone il dialogo, ai fini della programmazione, tra l'ATmega328P della sezione *Master* ed il micro target; come si può notare vi sono delle istruzioni comuni e delle istruzioni specifiche per le tre modalità di programmazione (HVPP, HVP13, HVSP), che vengono eseguite o meno, in base al valore assegnato alla variabile *mode*. All'avvio della procedura, *mode* è impostata su HVSP, infatti questa modalità ci permette di leggere la signature di tutti i tipi di micro programmabili in HVSP, mentre restituisce il valore "FF" se il micro target richiede la modalità HVPP o HVP13. Questo ci dà l'indubbio vantaggio di poter analizzare automaticamente il tipo di micro target, senza doverne prima specificare le caratteristiche. Per compiere questa prima operazione di analisi viene quindi eseguita la funzione "byte signature_read_HVSP", che si trova nella sezione denominata "LETTURA HVSP MODE", il cui risultato sarà una signature valida (in caso di micro programmabili in HVSP) oppure "FF" (negli altri casi), o ancora "00" nei casi di micro con signature danneggiata. La validità della signature, che in pratica consiste nel verificare che il primo byte abbia valore "1E", viene effettuata dalla routine "signature_check": nel caso di valore "FF" la procedura cambia modo di programmazione in HVPP e riesegue la lettura mediante la routine "byte signature_read_HVPP". Accertata la validità della signature (primo byte uguale a "1E"), la procedura confronta gli altri due byte con quelli dell'array "signature", mediante la funzione "unsigned char cfr_signature". In caso di corrispondenza, mediante il valore di indice ricavato, vengono estrapolati dai due array "atmel" e "fuses" tutti i parametri relativi al micro target: modello, numero e valori di default dei fuse, modo di programmazione. A questo punto sul monitor vengono mostrati tutti i valori appena letti, che sono quelli di default del micro; subito dopo si passa alla lettura dei valori attuali degli elementi programmabili e cioè i fuse ed i lock bits. Nel caso del modo di programmazione HVSP vengono eseguite le due routine

"fuses_read_HVSP", per la lettura dei fuse, e "lockbits_read_HVSP", per la lettura dei lock bits; invece, se il modo di programmazione impostato è HVPP o HVP13 vengono eseguite le routine "fuses_read_HVPP" e "lockbits_read_HVPP". Da notare che fino a questo momento la routine di inizializzazione "begin_programming" è stata eseguita in modalità HVSP e che l'eventuale passaggio alla modalità HVPP o HVP13 è stato dovuto solo all'eventuale lettura della signature.
Ma quando si devono leggere valori come i fuse o i lock bits, è necessario che l'inizializzazione venga fatta per la corretta modalità di programmazione.
Ecco perché, in questa fase, nel caso di micro che richiedano la modalità HVPP o HVP13, è assolutamente necessario chiudere la programmazione attuale e riaprirla tenendo conto della situazione attuale. A lettura conclusa sul monitor vengono visualizzati anche i valori correnti del micro, per quanto riguarda fuse e lock bits.
Questa fase è molto meno banale di quanto non si possa pensare. Infatti mentre la signature è un valore invariabile, i fuse ed i lock bits sono soggetti a programmazione, quindi poter controllare in un istante il settaggio di un micro "usato" non è cosa da poco. Inoltre una delle problematiche ricorrenti quando si manipola molte volte un chip, è la cancellazione della sigla presente sul corpo; in questo caso è estremamente difficoltoso poter risalire al modello di microcontrollore se non si ha un qualsiasi indizio, ecco che questa operazione di lettura automatica del micro target diventa importante e fondamentale, ai fini del suo successivo utilizzo. Bene, ora che disponete di tutte le informazioni relative al micro target che avevate in precedenza inserito nel rispettivo zoccolo, potete passare alla fase di programmazione diretta, ecco perché vedrete apparire sul monitor il Menu Principale, le cui voci sono:

0: FINE
1: Ripristino fuse default
2: Scrittura fuse personali
3: Scrittura Lock bits
4: Erase micro

L'opzione 0 chiude la fase di programmazione eseguendo la routine "end_programming", che in pratica spegne le due alimentazioni del micro target e pone su LOW tutti i pin di comando.
L'opzione 1 permette di scrivere nel micro target i valori di default dei fuse, prelevati in precedenza dalla tabella; l'operazione viene eseguita mediante la routine "fuses_write_HVSP" o "fuses_write_HVPP", in base al modo di programmazione del micro target, ed è seguita da una rilettura dei fuse mediante, rispettivamente, la routine "fuses_read_HVSP" o "fuses_read_HVPP", al fine di verificare il buon fine dell'operazione. L'opzione 2 richiede l'inserimento manuale dei valori dei fuse, da parte dell'utente; si usa nei casi di personalizzazione di tali valori; vengono richiesti nell'ordine: Low Fuse e High Fuse seguito, se previsto nel micro target, dall'Extended Fuse; a questo punto seguono le operazioni di scrittura e rilettura viste nel punto precedente, ovviamente la procedura è identica, a prescindere dal valore di fuse che si va a scrivere nel micro.
L'opzione 3 permette la programmazione del Lock bit Byte, mediante inserimento del valore desiderato; per consentire all'utente di scrivere i valori direttamente in esadecimale (diversamente occorrerebbe tradurre tutto in byte) sia l'inserimento dei fuse che quello dei lock bits viene effettuato mediante la funzione "hexfb_ask" (e relativa funzione di conversione "hex2dec"). Subito dopo, come per i fuse, viene effettuata prima la scrittura (routine "lockbits_write_HVSP" o "lockbits_write_HVPP") e poi la rilettura per verifica (routine "lockbits_read_HVSP" o "lockbits_read_HVPP"). L'opzione 4 effettua la cancellazione della memoria flash e EEPROM del micro target e resetta il valore del Lock Bit Byte, ponendolo a "FF"; viene effettuata mediante la routine "chip_erase_HVSP" o "chip_erase_HVPP".
Se avete dato un'occhiata al firmware vi sarete resi conto di come praticamente ogni linea sia commentata ai fini della comprensione del suo funzionamento. Tuttavia alcune procedure sono piuttosto complesse, per chi non ha dimestichezza con la programmazione HV, quindi nella prossima puntata, a titolo di esempio, vi spiegheremo come vengono effettuate alcune delle operazioni previste dal Programmatore HV, descrivendole in modo molto minuzioso, sia per la modalità HVSP che per la modalità HVPP.



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono di facile reperibilità. Il master del circuito stampato ed il software per programmare il microcontrollore possono essere scaricati dal sito della rivista.
Il microcontrollore vergine ATMega328 (cod. ATMEGA328P-PU) è disponibile al prezzo di 7,00 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

made in Italy

Conosci bene ARDUINO?

Vuoi realizzare un telecontrollo GSM per gestire e monitorare a distanza impianti di riscaldamento/condizionamento o apparecchiature elettriche di vario genere?

La soluzione è...

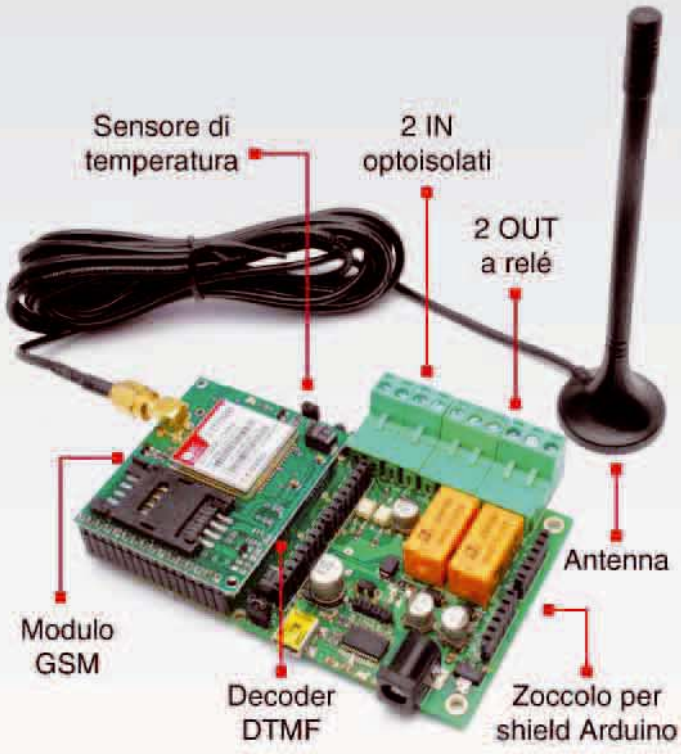
TiDiGino

la piattaforma hardware open-source, flessibile, semplice da utilizzare e alla portata di tutti.

Sviluppa il tuo controllo remoto basato su ARDUINO!

Con TiDiGino potrai ad esempio monitorare e impostare da remoto la temperatura ambiente di casa tua, gestendo l'impianto di riscaldamento/condizionamento, controllare pompe ed impianti di irrigazione, antifurti per immobili civili ed industriali, l'apertura di porte o cancelli, e molto altro ancora.

Lo zoccolo disponibile sulla scheda TiDiGino ti permette di utilizzare shield per Arduino progettati per svolgere specifiche funzioni utili ad ampliare le risorse hardware e le potenzialità applicative dell'unità.



Il telecontrollo GSM dispone di:

- ✓ Modulo GSM con SIM900 completo di antenna.
- ✓ Decoder DTMF.
- ✓ 2 uscite a relé.
- ✓ 2 ingressi optoisolati.
- ✓ Sensore di temperatura.
- ✓ Zoccolo per il montaggio di shield ARDUINO.
- ✓ Porta USB per programmazione (IDE ARDUINO).

Grazie alla disponibilità di librerie per la gestione della sezione GSM, ed alcuni esempi per testare le funzionalità hardware, potrai realizzare il tuo progetto ancora più facilmente.

Il telecontrollo GSM TiDiGino (cod. FT924M) viene fornito programmato con bootloader e sketch in grado di far svolgere al dispositivo tutte le funzioni disponibili. Informazioni tecniche dettagliate disponibili su ***www.futurashop.it****.*

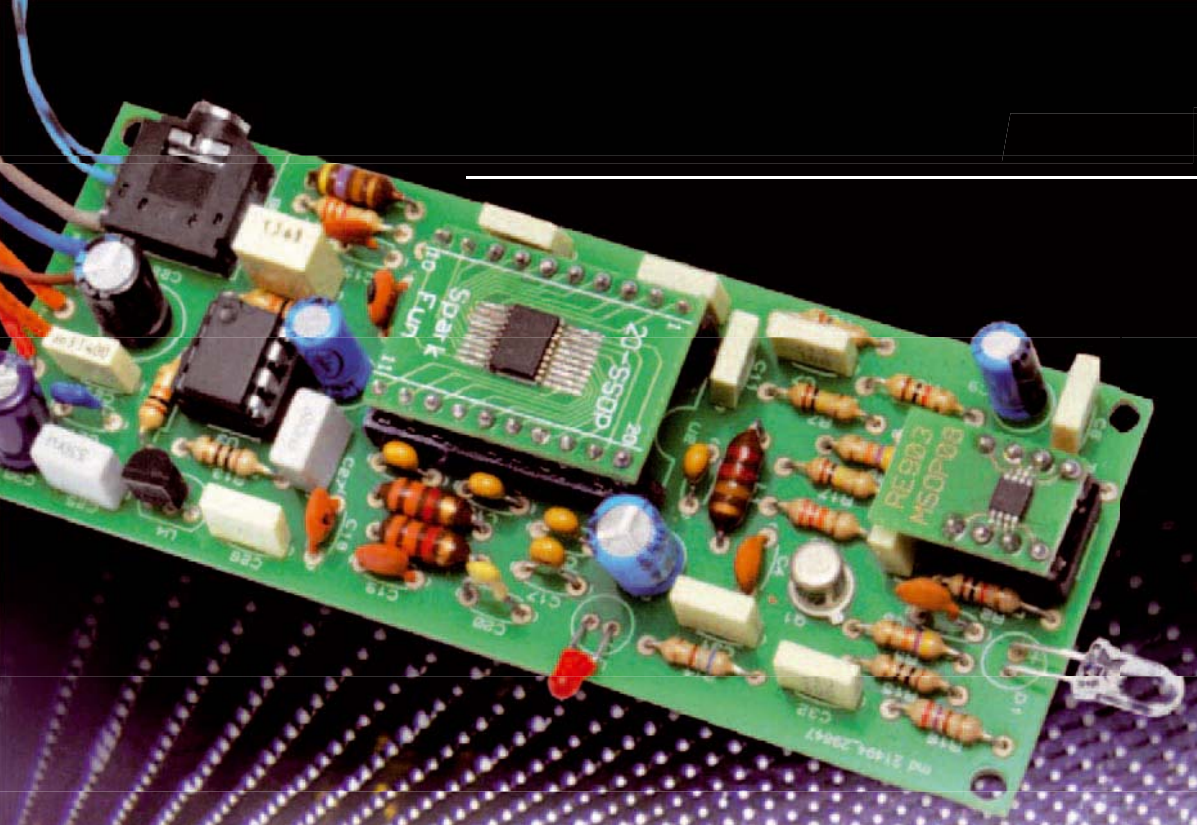
€ 138,00 *cod.* **FT924M**

FUTURA ELETTRONICA

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

Prezzi IVA inclusa.

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



Utilizziamo i LED per illuminare un ambiente e nel contempo sfruttiamo l'emissione luminosa per per trasmettere un segnale audio in FM.

COMUNICAZIONI AUDIO A LED

di FULVIO DE SANTIS



Inviare e ricevere informazioni mediante raggi luminosi è senz'altro una alternativa valida anche se futuristica al tradizionale sistema di comunicazioni wireless a radio frequenza. Conosciamo infatti quali siano i rischi dell'utilizzo di sorgenti a radio frequenza, specialmente nell'ambiente domestico, ma in generale ovunque esse agiscano. In un futuro, credo abbastanza prossimo, potremo avere a disposizione nelle nostre abitazioni, nei luoghi di lavoro e in tutti gli ambienti ove sia necessario avere punti informativi, sorgenti di luce emittenti segnali sottoforma di variazioni lumi-

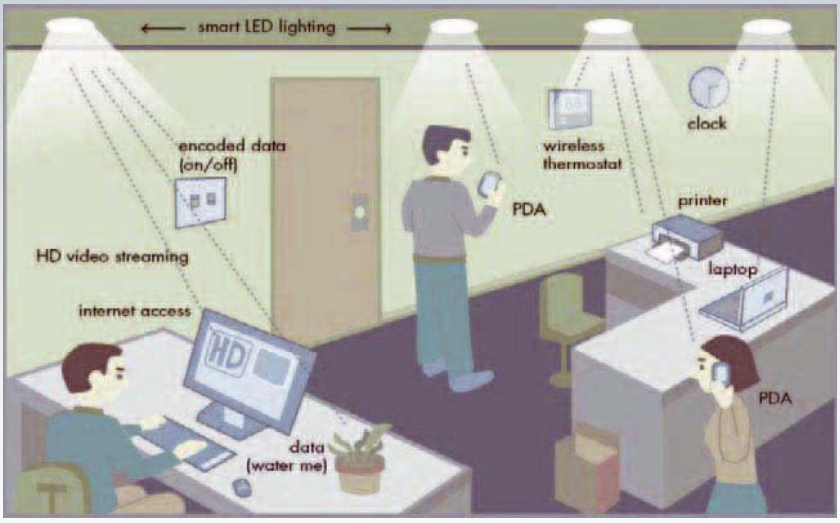


***Fig. 1** - Soluzioni di utilizzo dei sistemi LED wireless.*

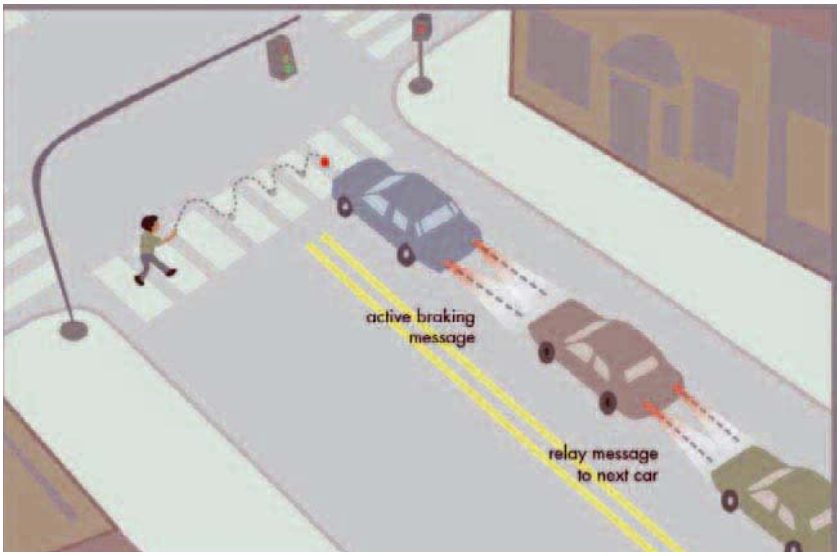


***Fig. 2** - Sistema di sicurezza attiva stradale a LED,*

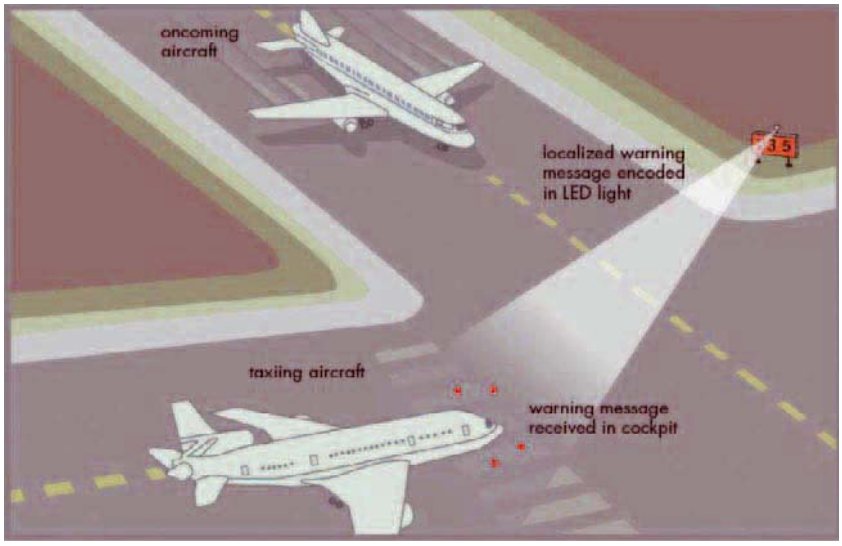


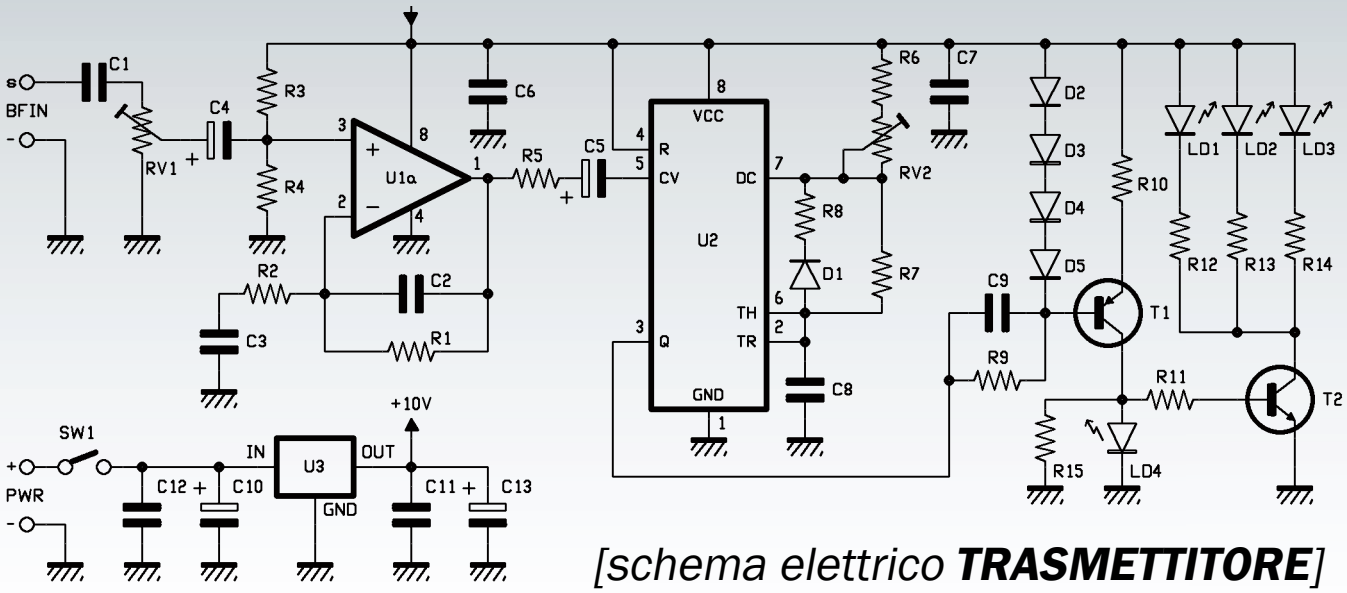
***Fig. 3** - Sistema di sicurezza attiva aereoportuale a LED,*

nose, o luce modulata, che consentano di inviare informazioni e dati a postazioni riceventi fisse e mobili. Immaginiamo dunque la possibilità di navigare in internet semplicemente accendendo una lampadina a LED ricevendone i relativi raggi luminosi mediante una chiavetta USB dotata di ricevitore a fotodiodi collegata al nostro computer, iPHONE, Tablet.

È possibile allora arrivare alla creazione di una rete LAN basata su LED, identificando le postazioni remote attraverso IP dedicati dei rispettivi corpi illuminanti a LED (**Fig. 1**). I LED luminosi di potenza sono già utilizzati nei fari delle auto prodotte dalle case automobilistiche più importanti; anche in questo caso pilotando i LED con moduli appositi potrebbe essere possibile realizzare sistemi di sicurezza intelligenti per la segnalazione di arresto imprevisto di veicoli, mediante comunicazione di messaggi fra i LED delle luci dei fari anteriori di un veicolo e i LED delle luci di arresto posteriori del veicolo che precede (**Fig. 2**).

Nel campo dell'aviazione il sistema TX-RX LED potrebbe essere utilizzato come faro LED per la sicurezza attiva durante le fasi di decollo e atterraggio mediante l'invio di messaggi codificati a bordo degli aerei per la segnalazione di traffico imprevisto di aeromobili e veicoli di servizio non segnalati dalla torre di controllo e ridurre così possibili collisioni in pista (**Fig. 3**).

Certo questi scenari possono sembrare troppo avveniristici ma probabilmente, come in altri campi in evoluzione "è solo questione di tempo". In tempi di crisi e di alti costi dell'energia elettrica, sarebbe auspicabile un impiego sempre più diffuso di lampade a LED intelligenti in luo-



*[schema elettrico **TRASMETTITORE**]*

go delle lampade tradizionali, ivi comprese quelle cosiddette a basso consumo di fascia energetica A, lampade a LED che dovrebbero essere ancora più diffuse grazie alle innovative caratteristiche di multifunzionalità.
Attualmente alcune Università internazionali in collaborazione con le industrie del settore stanno sperimentando i primi prototipi ma, anche se i risultati sono incoraggianti, ancora non sono disponibili previsioni di commercializzazione a livello industriale per la diffusione al pubblico a causa degli alti costi di produzione e di limiti tecnologici.



DESCRIZIONE DEL SISTEMA TX-RX AUDIO FM LED

Il presente progetto RX-TX AUDIO FM ad illuminazione a LED BIANCHI segue il progetto presentato nel numero 153 di Febbraio 2011 di Elettronica In. In quel caso la trasmissione avveniva in AM ed era soggetta ad alcune limitazioni di distanza, superate nel progetto attuale in FM, che dimostra come sia possibile illuminare un ambiente e nel contempo utilizzare il flusso luminoso per trasmettere un segnale audio da una sorgente esterna ad una postazione ricevente remota. È quindi possibile realizzare un collegamento wireless affidabile in quanto immune alle interferenze elettromagnetiche, caratteristica delle comunicazioni basate su fasci luminosi, ad esempio collegando un'apparecchiatura audio al trasmettitore e una cuffia stereo o un altoparlante al ricevitore remoto.
Il sistema LED wireless è in grado di coprire una distanza di oltre 10 metri, ovviamente senza ostacoli fra TX ed RX. Un'altra idea di utilizzo può essere il creare una sorta di luce ambiente di cortesia per una più riposante visione della TV ponendo il TX in prossimità del televisore e l'RX collegato alla cuffia o altoparlante del telespettatore. Inoltre possiamo ipotizzare di utilizzare il sistema, dopo opportuna eventuale post-amplificazione in ricezione, come ripetitore wireless del segnale audio di un sistema Home Theatre verso gli altoparlanti posteriori eliminando così l'impiego di cavi lungo la stanza. Il progetto si compone di due parti: la scheda TX trasmettitore e la scheda RX ricevitore.

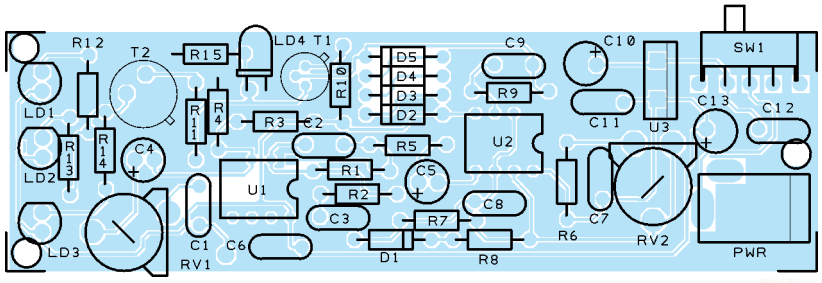
DESCRIZIONE DEL TRASMETTITORE

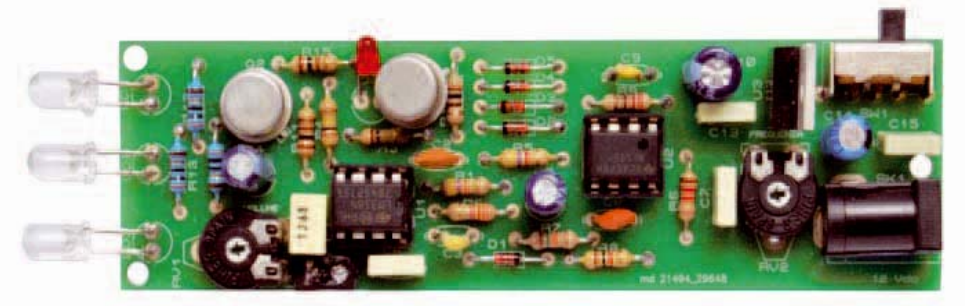
Il TX si compone di quattro sezioni: il preamplificatore audio, il VCO, il driver dei tre LED bianchi luminosi e la sezione di alimentazione. Analizziamo il funzionamento delle varie sezioni.

[piano di montaggio **TRASMETTITORE**]

Elenco Componenti:

R1: 47 kohm
R2: 3,3 kohm
R3, R4: 100 kohm
R5: 4,7 kohm
R6: 1,2 kohm
R7: 15 kohm
R8: 1,2 kohm
R9: 220 ohm
R10: 10 ohm
R11: 100 ohm
R12÷R14: 30 ohm 1%
R15: 100 ohm
RV1: Trimmer 47 kohm MO
RV1: Trimmer 10 kohm MO
C1: 1 μF poliestere
C2: 68 pF ceramico
C3: 820 pF ceramico
C4, C5: 4,7 μF 63 VL elettrolitico
C6, C7: 100 nF poliestere
C8: 220 pF ceramico
C9: 820 pF ceramico
C10: 100 μF 25 VL elettrolitico
C11, C12: 100 nF poliestere
C13: 10 μF 63 VL elettrolitico
U1: LM358





U2: NE555
U3: 7810
D1÷D5: 1N4148
LD1÷LD3: LED 5 mm bianco
LD4: LED 3 mm rosso
T1: 2N3505
T2: BC301

SW1: Deviatore a slitta 90°

Varie:
- Zoccolo 4+4 (2 pz.)
- Plug alimentazione
- Strip maschio 2 poli
- Circuito stampato

IL PREAMPLIFICATORE AUDIO

È costituito dal circuito integrato LM358 al cui interno vi sono due amplificatori operazionali ad alto guadagno che possono essere alimentati con singola o duale alimentazione. In questo progetto utilizziamo solo uno dei due amplificatori (U1/A) che viene alimentato in alimentazione singola.
Il segnale audio proveniente da apparecchiature esterne viene applicato al connettore BFIN e regolato in ampiezza dal trimmer RV1 verso l'operazionale U1 in configurazione non invertente. Il funzionamento dell'operazionale in singola alimentazione è garantito dalle due resistenze R3-R4 che riportano sul piedino 3 non invertente metà della tensione di alimentazione. La rete di controreazione R1-C2 e il circuito RC serie R2-C3 modellano la risposta in frequenza per ottenere una ottimale resa sonora alle alte frequenze audio.

IL VCO

La sezione dell'oscillatore controllato in tensione viene realizzata utilizzando il noto NE555, U2 nello schema elettrico.
Dal preamplificatore il segnale audio viene applicato all'ingresso del VCO attraverso il filtro RC serie R5-C5 e da questo al piedino 5 di controllo la cui variazione di ampiezza và a modulare la frequenza del segnale impulsivo generato dall'integrato NE555 U2. L'oscillatore è progettato per fornire un segnale impulsivo con duty-cycle del 10% per una migliore risposta dei LED. La frequenza dell'oscillatore ad impulsi, che quindi è la frequenza portante del trasmettitore, in assenza di modulazione viene regolata a 200KHz dal trimmer RV2. Il segnale così modulato è quindi un segnale FM che dopo il filtro RC parallelo R9-C9 viene trasferito al driver dei LED bianchi.

IL DRIVER DEI LED

Il driver dei LED è a sua volta composto da due sezioni: la sezione pilota del driver costituita dai diodi D2-D3-D4-D5 e dal transistor T1 che ha la funzione di rendere quanto più ripidi possibile i fronti degli impulsi modulati provenienti dal VCO; la sezione driver dei LED è realizzata mediante il transistor di media potenza T2 che alimenta i LED con una corrente che è funzione degli impulsi modulati. Una descrizione a parte per i LED è opportuna.
I LED bianchi luminosi impiegati in questo progetto sono i LL-1516HCWW1-151 della LEDman Optoelectronics, ma è possibile sostituirli con altri di caratteristi-

che simili che di seguito andiamo ad analizzare.
Le specifiche di valori assoluti sono riportate nella **Tabella 1**.
Le caratteristiche di funzionamento principali sono riportate in **Tabella 2**.
Come vediamo nella **Tabella 2** i LED dovranno essere polarizzati con una tensione diretta VF = 3.2 V e con una corrente diretta di 20 mA si avrà una intensità luminosa tipicamente di 18 candele per un angolo di irradiazione di 0°, mentre per un angolo di radiazione di 15° si avrà il 50% della potenza luminosa, ciò significa che l'intensità del fascio luminoso che forma un angolo di 15° con l'asse di puntamento del LED varrà 9 candele. Nel grafico di **Fig. 4** osserviamo quanto detto.
Vediamo inoltre nella **Fig. 5** come l'intensità luminosa dipenda dalla corrente diretta di polarizzazione del LED.
La lunghezza d'onda di emissione dei LED è di 470 nm.

***Tabella 1** - Valori massimi applicabili ai LED LL1516HCWW1-151*

Valori massimi assoluti (T_a=25°C)
Corrente diretta DC IF 30 mA
Corrente di picco* I_{FP} 100mA
Tensione inversa V_R 5V
Dissipazione P_D 108 mW
Temperatura Operativa T_{opr} -20 ~ +75°C

***Tabella 2** - Specifiche di funzionamento dei LED LL1516HCWW1-151*

Parametri	Valori
Tensione diretta	V_F = 3.2 V (I_F = 20mA)
Intensità luminosa I_v	18.000 mcd (IF=20mA)
Angolo di irradiazione	15° (al 50% dell'intensità luminosa)
Lunghezza d'onda	470 nm (100% intensità luminosa)

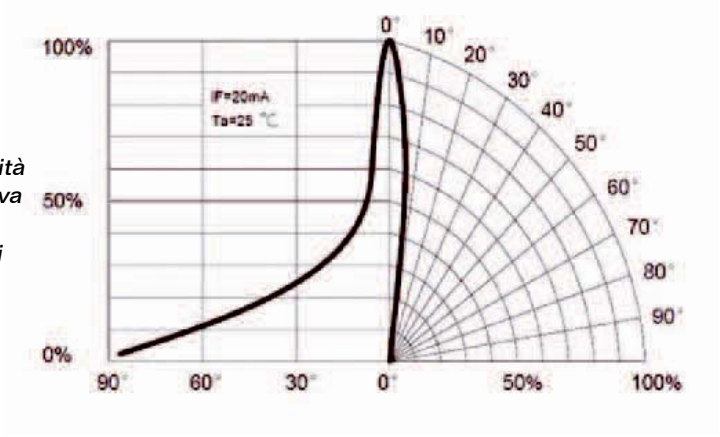


***Fig. 4** - Intensità luminosa relativa in funzione dell'angolo di radiazione.*

L'ALIMENTATORE

L'alimentatore è molto semplice: realizzato con l'integrato regolatore L7810 e relativi condensatori di filtraggio eroga in uscita una tensione stabilizzata di +10V.
Deve essere alimentato con un alimentatore esterno da 12 Vdc.

DESCRIZIONE DEL RICEVITORE

Il ricevitore si compone di cinque parti: la sezione di ricezione di luce modulata composta dal fotodiodo e l'amplificatore a transimpedenza; l'amplificatore selettivo, il demodulatore FM, l'amplificatore di potenza audio e l'alimentatore.
La sezione di ricezione ha in ingresso il fotodiodo SFH203 che alimentato con una tensione inversa di 5V svolge la funzione di captare il flusso luminoso emesso dai LED bianchi del trasmettitore in corrente elettrica.
Nella Tabella 3 sono riportate le principali specifiche di funzionamento del fotodiodo SFH203.
Nella **Fig. 6** osservate la variazione lineare della corrente Ip del fotodiodo in funzione dell'il-

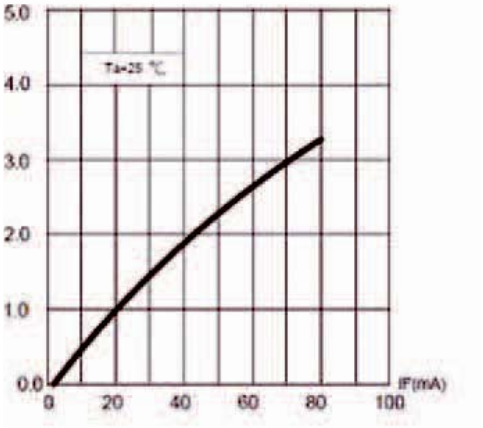


***Fig. 5** - Intensità luminosa relativa in funzione della corrente diretta.*

***Tabella 3** - Specifiche di funzionamento del fotodiodo SFH203*

Parametri	Valori
Tensione inversa	V_F = 5V
Fotocorrente	80 nA/I_x (V_R = 5V)
Angolo radiante	+/- 20° (50% Intensità luminosa)
Range Spettrale Sensibilità	400-1100 nm
Area radiante sensibile	1 mm^2
Sensibilità spettrale (a 850 nm)	0,62 A/W
Dark current (V_R = 20V)	< 5nA (E_L = 0)
Tempo di salita e discesa impulso della foto-corrente (λ 850 nm)	Switching Time St = 5 ns)



Nuova illuminazione a LED telecontrollata per il comune di Sulbiate

L'iniziativa nasce nell'ambito del Distretto Green High Tech Monza e Brianza, che consorzia le realtà produttive e le amministrazioni pubbliche del territorio di riferimento, su iniziativa del Comune di Sulbiate, che ha scelto come partner energetico e di sistema Edison... e come partner tecnologico Philips. L'area di intervento individuata è prossima ad una zona di parcheggio con annessa pista ciclabile a ridosso di una importante strada provinciale. Gli obiettivi del progetto hanno previsto la massima attenzione ai livelli di sicurezza per veicoli e pedoni grazie alla qualità della luce, l'attenzione al risparmio energetico e alla riduzione dei costi di gestione e manutenzione e la possibilità di telecontrollo del sistema attraverso la piattaforma di gestione integrata CityTouch, una soluzione di cloud computing che permette di attuare un controllo online intelligente, dinamico e flessibile su scala urbana dell'illuminazione realizzata interamente a LED.

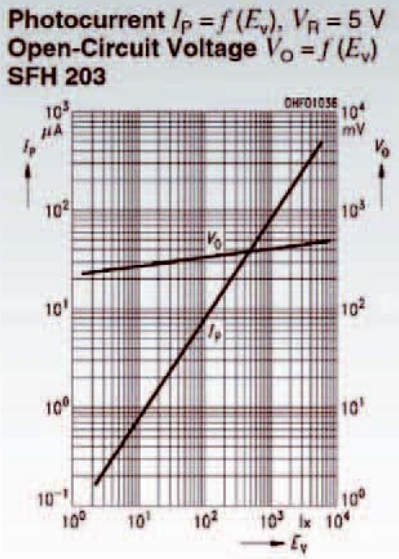


***Fig. 6** - Fotocorrente funzione dell'illuminamento*

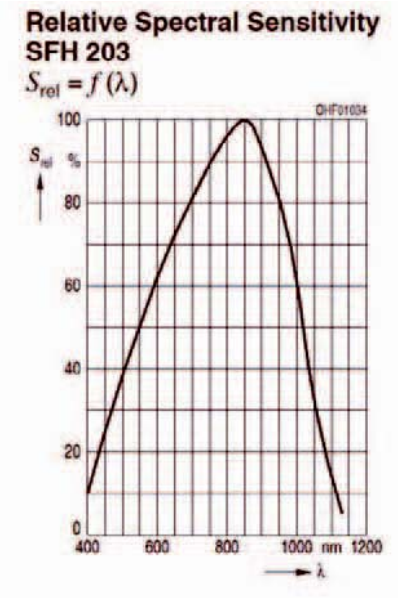


***Fig. 7** - Sensibilità in funzione della lunghezza d'onda del fotodiodo SFH203.*

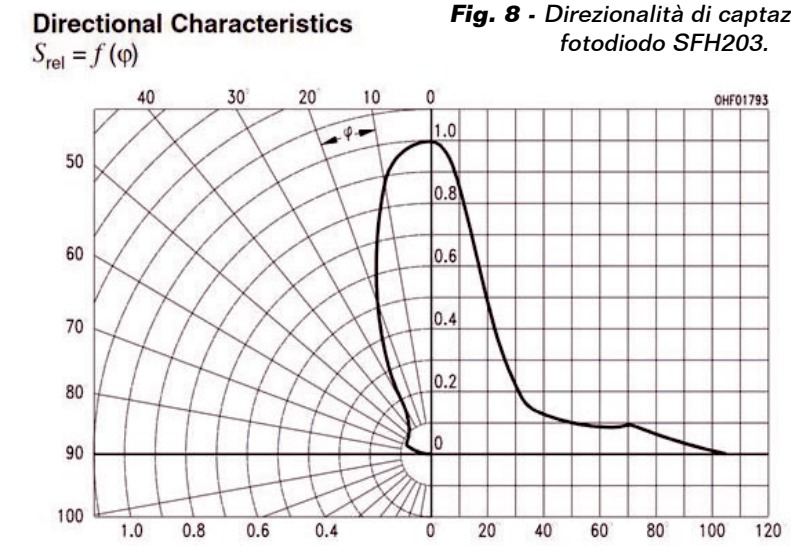


***Fig. 8** - Direzionalità di captazione del fotodiodo SFH203.*

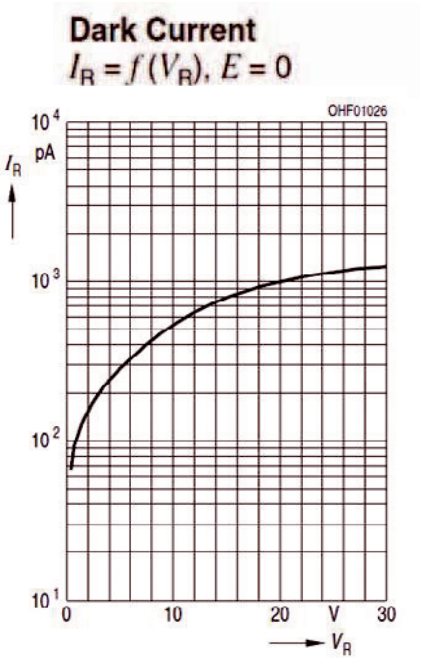
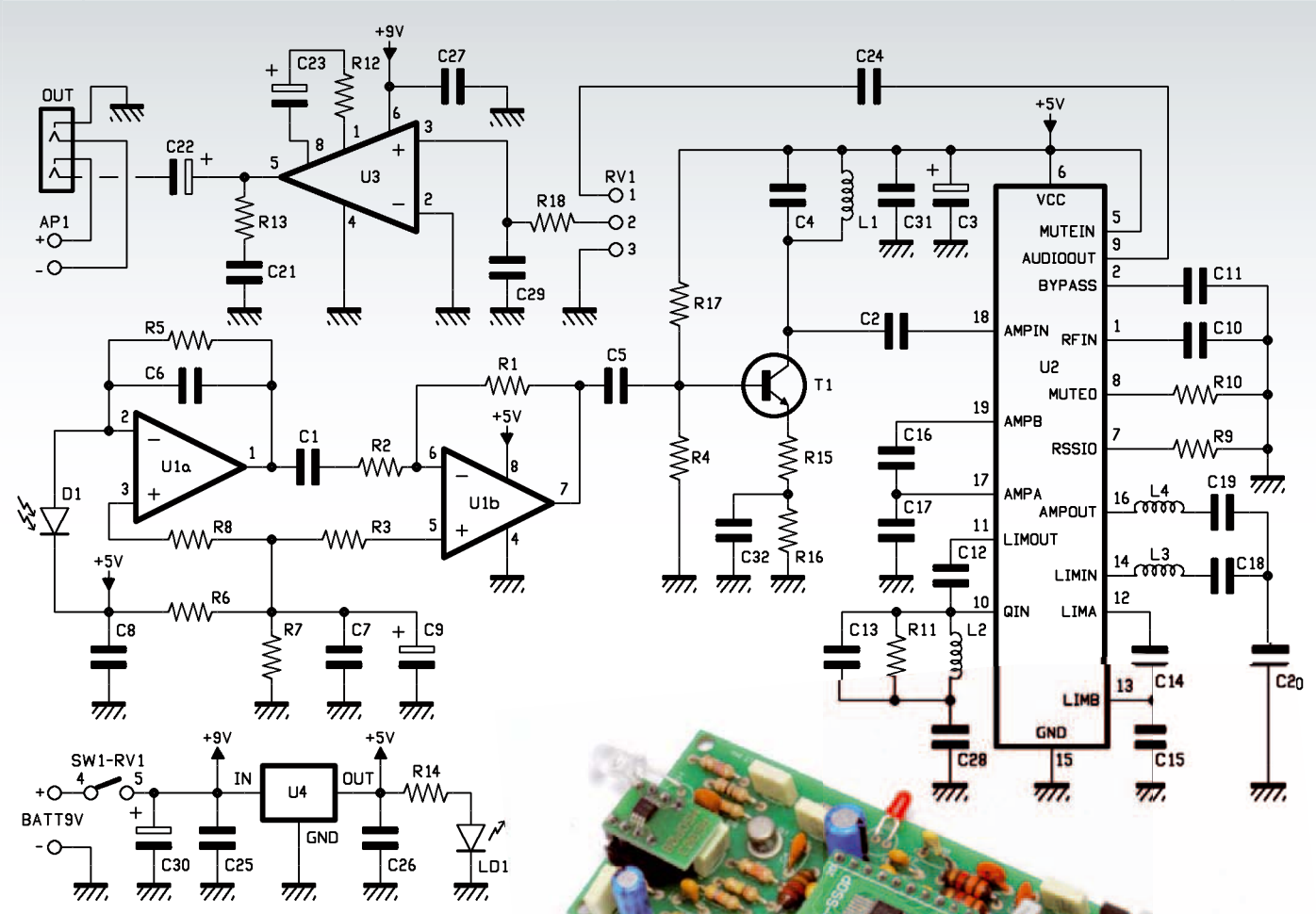
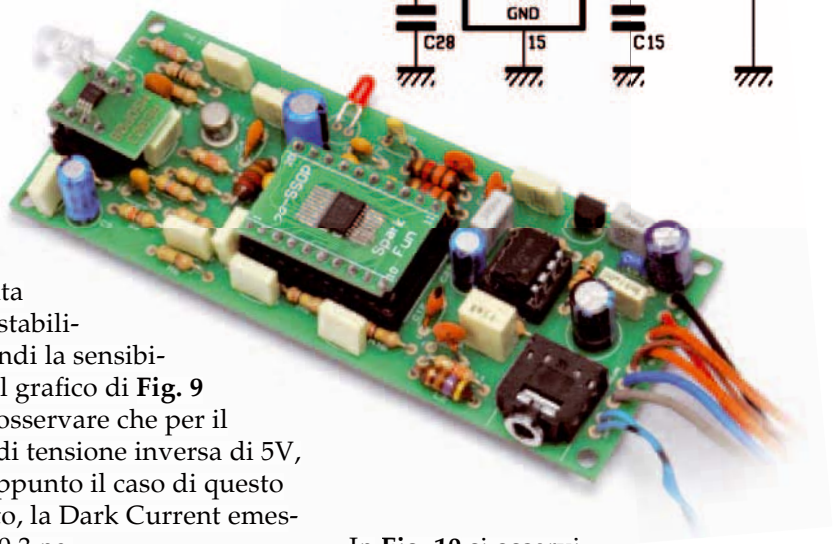
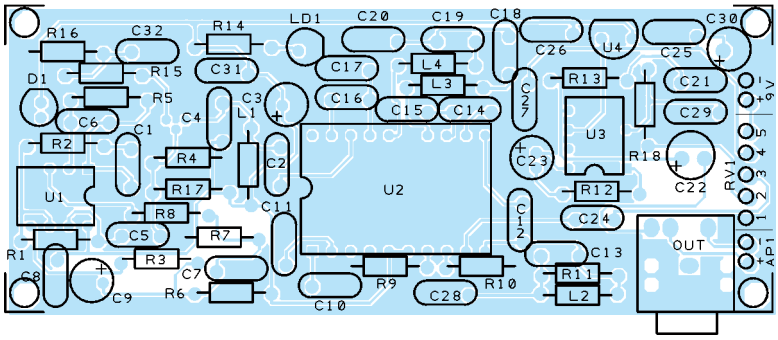


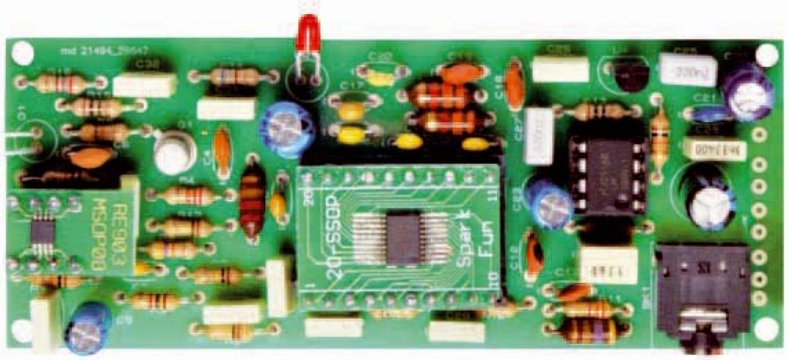
***Fig. 9** - Dark Current del fotodiodo SFH203.*





luminamento Ev.
Nel grafico della **Fig. 7** viene invece mostrato l'andamento della sensibilità del fotodiodo in funzione della lunghezza d'onda del flusso di luce captato.
Altro elemento importante da considerare per la sensibilità del fotodiodo è l'angolo di captazione che ne definisce la direzionalità. Nella **Fig. 8** potete osservare l'angolo di radiazione misurato al 50% dell'intensità luminosa ricevuta, corrispondente ad un angolo di captazione di +20°/-20°.
La soglia minima di captazione del fotodiodo SFH203 è determinata dalla Dark Current, o corrente di buio, il cui valore è determinato dal valore della tensione di polarizzazione inversa applicata che ne stabilisce quindi la sensibilità. Dal grafico di **Fig. 9** potete osservare che per il valore di tensione inversa di 5V, che è appunto il caso di questo progetto, la Dark Current emessa è di 0,3 na.
La corrente emessa dal fotodiodo quando è investito dal flusso luminoso viene applicata all'ingresso della sezione U1:A dello speciale doppio operazionale U1 LMV797 della National configurata come amplificatore a transimpedenza caratterizzata da altissima impedenza, alto guadagno e basso rumore che trasforma in tensione la corrente; la sezione U1:B amplifica ulteriormente il segnale modulato.
In **Fig. 10** si osservi la risposta in frequenza dell'amplificatore a transimpedenza in cui notiamo una banda passante di oltre 300 KHz.
Lo stadio successivo è l'amplificatore selettivo costituito dal transistor T1 2N2222 con circuito risonante L1-C4 la cui banda stretta centrata a 200 KHz consente un alto guadagno e sensibilità al ricevitore.
In **Fig. 11** è riportata la risposta in frequenza del circuito rice-





Elenco Componenti:

R1: 10 kohm
R2, R3: 1 kohm
R4: 22 kohm
R5, R88: 47 kohm
R6, R7: 100 kohm
R8: 47 kohm
R9, R10: 100 kohm
R11: 3,3 kohm
R12, R13: 10 kohm
R14: 680 ohm
R15: 10 ohm
R16: 270 ohm
R17: 100 kohm
R18: 10 kohm
RV1: Potenziometro 10 kohm con interruttore
C1: 100 nF poliestere
C2: 100 nF ceramico
C3: 10 µF 63 VL elettrolitico
C4: 4,7 nF poliestere
C5: 100 nF ceramico
C6: 5,6 pF ceramico
C7, C8: 100 nF poliestere
C9: 10 µF 63 VL elettrolitico
C10, C11: 100 nF poliestere
C12: 47 pF ceramico
C13: 1,5 nF poliestere
C14, C17: 100 nF ceramico
C18, C19: 220 pF ceramico
C20: 820 pF ceramico
C21: 47 nF ceramico
C22: 220 µF 16 VL elettrolitico
C23: 10 µF 63 VL elettrolitico
C24: 100 nF ceramico
C25: 330 nF poliestere
C26: 100 nF poliestere
C27: 330 nF poliestere
C28: 330 nF poliestere
C29: 3,3 nF poliestere
C30: 100 µF 16 VL elettrolitico
C31: 100 nF poliestere
C32: 100 nF poliestere
U1: LMV797
U2: SA615D
U3: LM386
U4: 78L05

D1: SFH203
L1: Induttanza 150 µH
L2: Induttanza 470 µH
L3: Induttanza 3,3 mH
L4: Induttanza 3,3 mH
LD1: LED 3 mm rosso
T1: 2N2222

Varie:
- Zoccolo 4+4
- Clip per batteria 9V
- Presa 3,5 mm stereo da CS
- Altoparlante1W/8ohm
- Adattatore 10+10 SSOP/DIP
- Adattatore 4+4 MSOP/DIP
- Circuito stampato

vente dal fotodiodo all'uscita dello stadio selettivo. Come si nota l'amplificazione totale è di circa 120 db.
Dall'amplificatore selettivo il segnale FM passa all'ingresso delle stadio amplificatore-limitatore demodulatore FM realizzato dallo speciale integrato SA615.
Nella **Fig. 12** è riportato lo schema a blocchi del circuito integrato SA615. Potete osservare che è composto da mixer RF, oscillatore RF, amplificatore/limitatore IF, rivelatore in quadratura e muting.
Analizziamo ora lo stadio con il chip SA615SD.
Come vedete dal circuito, non viene utilizzato il mixer del SA615, quindi il segnale FM viene inviato all'ingresso dello stadio amplificatore IF per una prima amplificazione (pin 18); dall'uscita dell'amplificatore IF (pin 16) il segnale è applicato ad un filtro passa-banda a T centrato a 200 KHz e da questo all'ingresso dello stadio limitatore (pin 14) impiegato per evitare il rischio di saturazione degli stadi successivi; dal limitatore il segnale viene inviato al rivelatore in quadratura, la cui uscita demodulata (pin 10), risultante del solo segnale audio originario, è applicata attraverso il condensatore C24 all'amplificatore BF di potenza audio costituito dal circuito integrato U3 LM386.
La configurazione utilizzata del LM386 è ottimizzata per un guadagno in tensione di circa 25 db, come osservabile nel grafico di **Fig. 13** relativo alla simulazione della risposta in frequenza dell'amplificatore BF, con distorsione contenuta entro 1% nella banda 20 Hz-10 KHz, fino a circa 1,4% da 10 KHz a 20 KHz.
Il livello di uscita viene regolato dal potenziometro RV1; il filtro RC R18-C29 è indispensabile per l'eliminazione del fruscio di fondo causato dal rumore di demodulazione. In uscita dall'amplificatore è possibile collegare cuffie o altoparlanti con impedenza di 8 ohm.
Infine la sezione di alimenta-

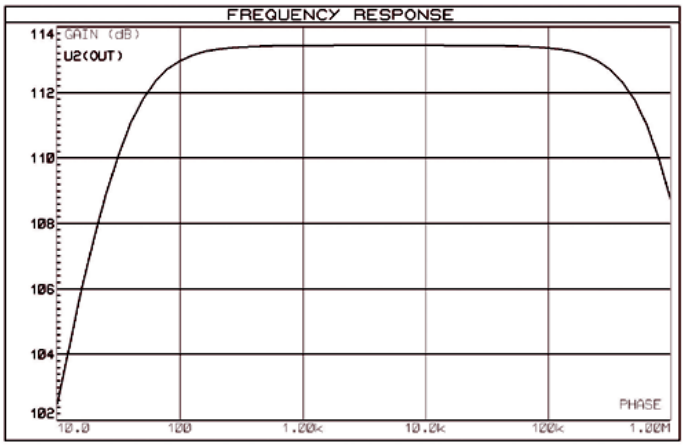


Fig. 10 - *Risposta in frequenza dell'amplificatore a transimpedenza.*

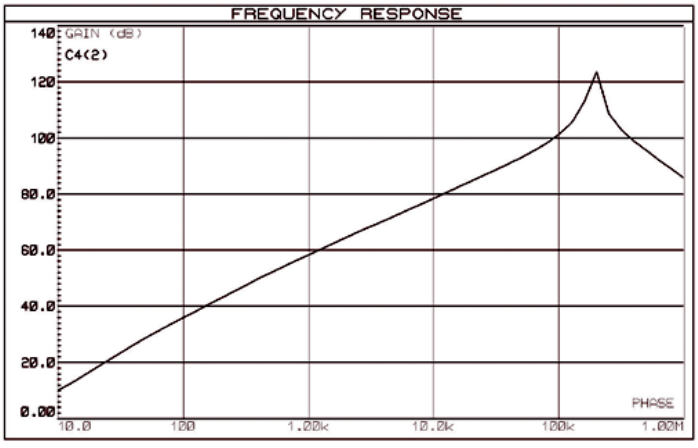


Fig. 11 - *Risposta in frequenza della sezione ricevente.*

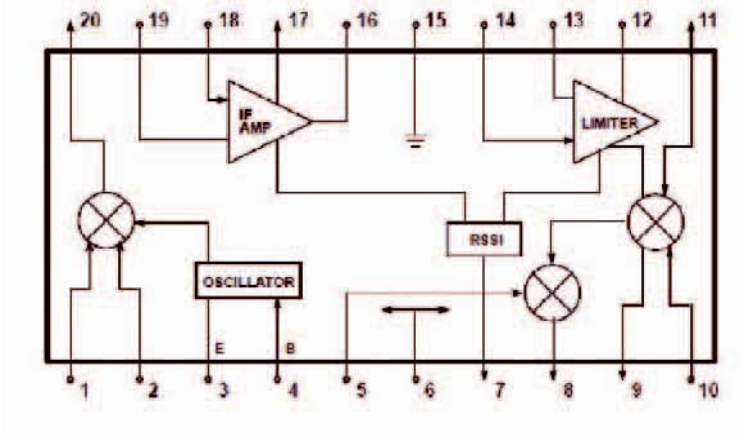


Fig. 12 - *Schema a blocchi SA615.*

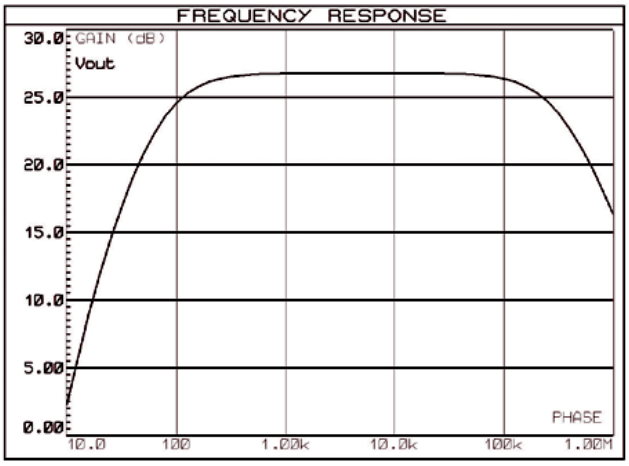


Fig. 13 - *Risposta in frequenza dell'amplificatore BF.*

zione è costituita dal circuito integrato regolatore 78L05 che stabilizza una tensione di +5V proveniente da alimentazione esterna, in questo caso è previsto il collegamento di una batteria da 9V.

REALIZZAZIONE PRATICA

Per il montaggio dei circuiti (gli stampati potete realizzarli scaricando le tracce lato rame disponibili nel nostro sito *www. elettronicain. it*) è consigliabile procurarsi un buon saldatore da 10÷15 W o una stazione saldante con regolazione della temperatura, dotati di punte sottili. Il montaggio si inizia con i componenti a basso profilo, quindi prima di tutto un'attenzione particolare per l'integrato U2 in SMD SA615-SMT nella scheda del ricevitore; prima di iniziare le operazioni di saldatura di questo integrato sarà bene pulire con dell'alcool per togliere eventuali depositi di grasso e sporcizia varia; si consiglia poi di passare con un pennellino dosatore del flussante liquido sul rame e bloccare il corpo dell'integrato sul circuito stampato con le punte delle pinzette facendo corrispondere i pins in allineamento con le relative piste. Saldate prima i pins incrociati estremi e poi i restanti pin.
Completata la saldatura del SA615 procedete con le resistenze, diodi (attenzione al corretto collegamento di anodo e catodo), circuiti integrati, trimmer a montaggio orizzontale, condensatori (facendo attenzione alla polarità degli elettrolitici), poi i transistor, procedete con i LED, sempre rispettando i collegamenti anodo-catodo considerando che il catodo è il terminale più corto; infine zoccoli, morsettiere, connettori.
Per applicare il segnale audio all'ingresso mono BF della scheda del trasmettitore occorre realizzare un cavetto audio per consentire il collegamento di sorgenti audio indifferentemente mono o stereo; se la fonte



per il MATERIALE

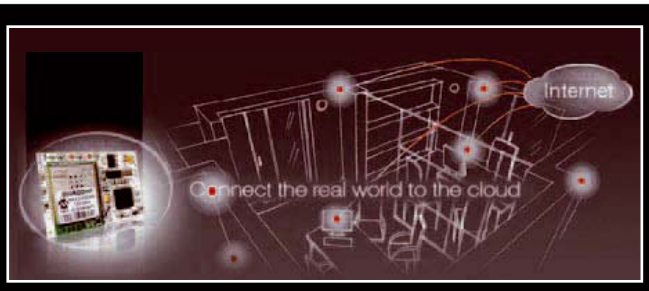
Tutti i componenti utilizzati nel progetto del sistema di comunivazione audio a LED sono facilmente reperibili in commercio. I master dei circuiti stampati sono scaricabili gratuitamente dal sito della rivista (www.elettronicain.it) e realizzabili con uno dei tanti metodi disponibili.

sonora è mono, va bene un cavetto coassiale terminante con un jack da 3,5 mm mono dal lato del TX e con uno spinotto RCA o un jack da quello della fonte.
Se invece si deve collegare al circuito un apparecchio stereofonico, bisogna partire da un cavo stereo, ma tagliarne il jack sostituendolo con uno mono, quindi connettere i due conduttori interni del cavo corrispondenti ai segnali dei canali L ed R (sinistro e destro) ciascuno con una resistenza da 1 kohm 1/8 W al contatto interno dello spinotto; le calze di schermo vanno unite e collegate insieme al contatto esterno (quello che stringe il cavo). Per evitare cortocircuiti conviene isolare le resistenze dal contatto esterno mediante un pezzo di termorestringente.
A completamento del montaggio occorre collegare i componenti esterni della scheda RX, quindi il potenziometro RV1, l'interruttore SW1, l'altoparlante ed infine il supporto della batteria a 9V.

TARATURA E COLLAUDO

Terminato il montaggio dei componenti delle schede TX e RX procedete con la taratura e messa in funzione del sistema. Saldate il cavetto che avete precedentemente realizzato alle piazzole relative all'ingresso audio del TX, denominate BFIN; mantenendo aperto l'interruttore SW1 collegate l'alimentatore esterno a 12V al connettore PWR del TX. Collegate l'altra estremità libera del cavetto audio ad una sorgente audio esterna utilizzandone l'uscita line-out o i connettori RCA dell'apparecchio. Regolate a metà corsa il potenziometro RV1 del volume del TX. Ponete le due schede ad una distanza di circa un metro. Collegate ora la batteria da 9V al supporto batteria della scheda ricevitore e accendete la scheda mediante l'interruttore SW1 e regolate a metà corsa il potenziometro del volume RV1, dovreste udire un certo fruscio dall'altoparlante o dalle cuffie.
Accendete la sorgente esterna audio e il trasmettitore mediante SW1 regolando il volume per una regolazione media. Se tutto funziona vedrete i tre LED accendersi con intensa luminosità. Udrete un fruscio dall'altoparlante del ricevitore a significare che non sono sintonizzate le due schede; a questo punto agite sul trimmer RV2 del trasmettitore regolando la frequenza portante fino ad udire con buona qualità il segnale audio della sorgente esterna dall'altoparlante del ricevitore.
Ora che il sistema è sintonizzato potete testare la portata del sistema RX-TX ponendo le schede ad una distanza fra di loro di almeno dieci metri ed affinate la taratura sempre agendo sul trimmer RV2 del trasmettitore fino ad ottenere la migliore qualità del suono sia con altoparlante che con le cuffie mono o stereo che avete.
Verificate il sistema possibilmente in condizioni di oscurità per valutarne così le caratteristiche di luminosità dei LED.









Alla scoperta di openPicus

Dopo avere presentato i dispositivi hardware della serie FlyPort e l'ambiente di sviluppo gratuito del progetto OpenPicus, è tempo di iniziare ad approfondire le possibilità di utilizzo con qualche semplice programma. Terza Puntata.

3

di Simone Marra

Nelle precedenti puntate abbiamo preso confidenza con l'hardware e con l'IDE di sviluppo per cui ora possiamo esplorare le funzionalità di base del FlyPort cominciando a scrivere dei semplici programmi di esempio, per poi analizzare più nel dettaglio la gestione delle funzionalità legate ai parametri della rete Wi-Fi.

CONFIGURAZIONE DEI PIN

Prima di affrontare il funzionamento delle varie periferiche del FlyPort, riportiamo in **Tabella 1** il pinout di default del dispositivo che, come vedremo tra breve, può essere cambiato in qualunque momento a runtime (con le dovute accortezze per quanto riguarda i collegamenti circuitali) .
Attraverso la funzione `IOInit ()`, sarà possibile modificare il comportamento dei vari pin. Ad esempio, per modificare il pin 2 da ingresso digitale a uscita digitale (e quindi per controllare ad esempio un LED) sarà sufficiente usare il comando `IOInit (p2, out)`.
Non solo si può selezionare se un pin debba essere un pin di ingresso (con o senza resistenze di pullup/pulldown intergrate) o di uscita, ma sarà anche possibile assegnare funzionalità

Tabella 1

PIN	FUNZIONE	RIMAPPABILE?
p1	I²C - linea di clock	x
p2	digital in 5	v
p3	I²C - linea dati	x
p4	digital out 1	v
p5	digital in 1	v
p6	digital out 2	v
p7	digital in 1	v (ingresso)
p8	SPI - SCLK	v
p9	digital in 3	v
p10	SPI - SDO	v
p11	digital in 4	v
p12	SPI - SDI	v
p13	UART1RX	v
p14	SPI - CS	v
p15	UART1TX	v
p16	+5V	x
p17	digital out 3	v
p18	Analog input 4	x
p19	digital out 4	v
p20	Analog input 3	v
p21	digital out 5	x
p22	GND	x
p23	Analog input 1	v
p24	+3.3V (LDO out)	x
p25	Analog input 2	v
p26	Reset	x

Tabella 2

PERIFERICHE D'INGRESSO	PERIFERICHE DI USCITA
in (digital input)	out (digital output)
inup (digital input with pullup internal resistor)	
indown (digital input with pulldown internal resistor)	
UART1RX	UART1TX
UART1CTS (not enabled in default UART initialization)	UART1RTS (not enabled in default UART initialization)
UART2RX	UART2TX
UART2CTS (not enabled in default UART initialization)	UART2RTS (not enabled in default UART initialization)
UART3RX	UART3TX
UART3CTS (not enabled in default UART initialization)	UART3RTS
UART4RX	UART4TX
UART4CTS (not enabled in default UART initialization)	UART4RTS
EXT_INT2	SPICLKOUT (for SPI Master mode, Clock Output Signal)
EXT_INT3	SPI_OUT (Data Output Signal)
EXT_INT4	SPI_SS_OUT (for SPI Master mode, Slave Select Signal)
SPICLKIN (for SPI Slave mode, Clock Input Signal)	
SPI_IN (Data Input Signal)	
SPI_SS (for SPI Slave mode, Slave Select Signal)	
TIM_4_CLK	

più avanzate, come per esempio aggiungere altre porte UART. Consultando il datasheet del FlyPort, infatti, si nota che sono indicate ben quattro porte UART, mentre nella configurazione di default ne è presente una sola. I pin che non supportano la funzionalità "rimappabile", possono essere configurati soltanto come ingresso o uscita digitale, mentre quelli rimappabili possono ospitare anche le altre periferiche elencate in **Tabella 2**. Gli elementi della lista possono essere utilizzati come parametri della funzione IOInit(numero pin, funzionalità). I pin non rimappabili saranno quindi compatibili solo con i valori delle prime tre righe della **Tabella 1**.
Vediamo quindi come sia possibile aggiungere una porta seriale a quella presente all'origine. Innanzitutto nel wizard di configurazione andranno abilitate due porte invece di una, come si vede nella **Fig. 1**. A questo punto non resta che da scegliere i pin su cui implementare le funzionalità di TX e RX per la seconda UART ed in seguito inizializzare e attivare la porta. I pin utilizzabili sono i cosiddetti pin "rimappabili", riportati in **Tabella 1**. Ad esempio scegliamo il pin 5 per il TX e il pin 6 per l'RX, con un baudrate di 19200. Il codice da usare nel file "*taskFlyPort.c*" sarà quindi il seguente:

```
// Configurazione dei pin
IOInit(p5, UART2TX);
IOInit(p6, UART2RX);
// Inizializzazione e attivazione della
porta seriale
UARTInit(2, 19200);
UARTOn (2);
```

Da questo momento avremo quindi due porte seriali sul nostro device, la prima la potremo usare come debug, e ricevere messaggi sul pc tramite un comune terminale seriale, la seconda potrà tornare utile per comunicare con device esterni al FlyPort. Le funzioni di scrittura e lettura della porta seriale, da questo punto in avanti, potranno essere utilizzate sia per la seriale 1 che per la seriale 2 (essendo stata attivata tramite UARTOn).
Se si volesse impostare il pin 7 come ingresso, si può utilizzare la funzione

```
// Imposta il pin 7 come ingresso
IOInit(p7, in);
```

Per impostarlo come uscita:

```
// Imposta il pin 7 come uscita
IOInit(p7, out);
```

Per utilizzare le resistenze di pull up o pulldown interne al microcontrollore, basterà utilizzare le funzioni (molto utili nel caso di utilizzo dei pin con dei pulsanti):

```
// Imposta il pin 7 come ingresso con pullup
IOInit(p7, inup);
// Imposta il pin 7 come ingresso con pulldown
IOInit(p7, indown);
```

Per leggere lo stato dei segnali di ingresso si possono utilizzare le funzioni

- `IOGet(pin)` per leggere lo stato "TRUE" o "FALSE" di ogni ingresso
- `IOButtonState(pin)` nel caso si utilizzino le configurazioni inup o indown, che restituisce lo stato del pulsante come "PRESSED" o "RELEASED". La funzione IOButtonState integra al suo interno un meccanismo automatizzato di *debounce* che permette di filtrare i segnali di rimbalzo, spesso generati dai pulsanti meccanici.

Per modificare lo stato delle uscite

- `IOPut(pin, valore)` per cambiare lo stato con i valori:

- "on", valore logico alto (3.3V);
- "off", valore logico basso (0V);
- "toggle", per invertire il valore presente sul pin di uscita specificato.

Oltre alle periferiche riportate in **Tabella 1**, si possono configurare fino a 9 moduli PWM sui pin rimappabili. Il funzionamento è semplice:

- Si inizializza il modulo PWM tramite la funzione
  `PWMInit(BYTE pwm, float freq, float dutyc);`
  dove **pwm** è il numero del modulo (da 1 a 9)
  freq è la frequenza di oscillazione in Hz(per es: 56.5)
  dutyc è la percentuale di duty cycle (da 0.0 a 100.0 %);
- Si associa il modulo PWM ad un pin rimappabile, di fatto accendendolo, tramite la funzione
  `PWMOn(Byte io, BYTE pwm);`
  dove **io** è il pin rimappabile da associare
  pwm è il numero del relativo modulo da utilizzare;
- Si può cambiare il duty cycle tramite la funzione
  `PWMDuty(float duty, BYTE pwm);`
  dove **duty** è la percentuale di dutycicle (da 0.0 a 100.0 %)
  pwm è il numero del relativo modulo da utilizzare;
- Per disabilitare il modulo è sufficiente spegnerlo tramite la funzione
  `PWMOff(BYTE pwm);`
  dove **pwm** è il numero del relativo modulo da disabilitare.

FUNZIONALITÀ WI-FI

Altra caratteristica peculiare del modulo FlyPort Wi-Fi è la presenza del transceiver Wi-Fi. Questo aggiunge delle funzionalità speciali, che i microcontrollori non dispongono. Per avere un assaggio di come si possa sfruttare il connubio transceiver wi-fi e microcontrollore possiamo analizzare una applicazione di scanner intelligente di reti wi-fi. Il modulo FlyPort Wi-Fi infatti dispone di funzionalità di analisi delle reti Wi-Fi, raggiungibili dal transceiver, in modo da poter tentare o meno la connessione a determinate reti. L'applicazione proposta, tramite appositi comandi sulla porta seriale, può eseguire le seguenti funzioni:

- Stampare su porta seriale l'elenco delle reti disponibili;
- Eseguire una connessione su una rete adhoc (se non esiste, ne crea una),
- Tentare di eseguire una connessione ad una rete infrastruttura predefinita;
- Tentare di eseguire una connessione ad una rete infrastruttura selezionata tra i risultati della scansione, con richiesta della eventuale password all'utente.

Per tentare le diverse connessioni il firmware utilizzerà i parametri **WF_CUSTOM**, che possono essere modificati, caricati, e salvati tramite firmware. Infatti a seconda della scelta dell'utente, il firmware modificherà i parametri "*WF_CUSTOM*" tramite le apposite funzioni del framework, e si connetterà alla rete.
Per esempio, in caso di selezione di rete "adhoc" tramite il comando sulla seriale "conadhoc", il firmware eseguirà le seguenti operazioni:

```
// command: "conadhoc"
if (strstr(uread,"conadhoc")!=NULL)
//cerca la stringa "conadhoc"
{
 // Set parameters
   UARTWrite(1,"Connecting AdHoc
   Network...\r\n");
   WFSetParam(MY_IP_ADDR, "192.168.1.115");
   WFSetParam(PRIMARY_DNS, "192.168.1.1");
   WFSetParam(MY_GATEWAY, "192.168.1.1");
   WFSetParam(SUBNET_MASK,
   "255.255.255.0");
   WFSetParam(SSID_NAME, "FLYADHOC");
   WFSetParam(DHCP_ENABLE, "DISABLED");
   WFSetParam(NETWORK_TYPE, "ADHOC");
   WFSetSecurity(WF_SECURITY_OPEN, "", 0, 0);
   WFCustomSave();
   // Start Connection
   WFConnect(WF_CUSTOM);
   while (WFStatus != CONNECTED);
```

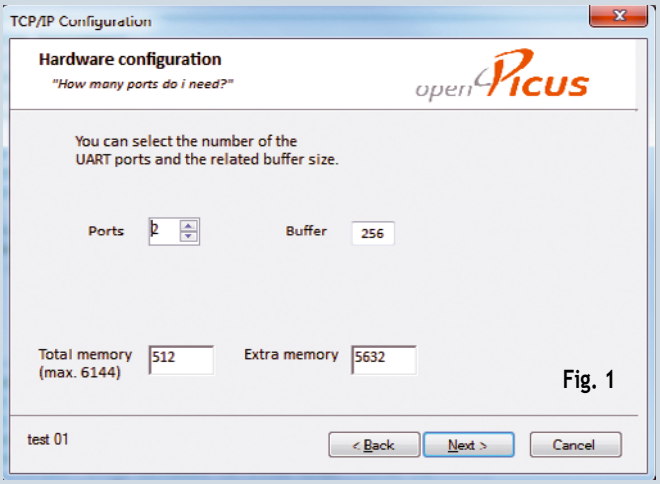


Fig. 1

```
    UARTWrite(1,"Flyport connected to AdHoc
    network...\r\n");
}
```

In caso di rete infrastruttura, alcuni parametri saranno da impostare in maniera differente, quali per esempio il NETWORK_TYPE, il DHCP_ENABLE, etc...
Anche la funzione WFSetSecurity avrà altri parametri (per maggiori informazioni si può vedere il codice sorgente ed il webinar03 sul sito *www.openpicus.com*, sezione downloads).
Chiaramente i parametri possono essere modificati anche durante l'esecuzione del firmware. Per non incappare in reti inesistenti, però, l'utente può eseguire una scansione delle reti disponibili tramite il comando "scan" che scrive su porta seriale l'elenco delle reti rilevabili dal modulo.
Sfruttando questa caratteristica, l'utente potrà scegliere la rete tramite un indice, e potrà inserire le password di sicurezza aggiungendo gli opportuni valori usando la porta seriale.
I comandi utilizzabili sono i seguenti (attenzione, le stringhe vanno sempre termitate con il carattere '\0', altrimenti la ricezione non viene considerata terminata dalle funzioni della porta seriale):

- "**scan**" per far partire la scansione delle reti;
- "**conadhoc**" per far connettere il FlyPort Wi-Fi ad una rete adhoc predefinita (ne crea una in caso non esistesse);
- "**coninfra**" per far connettere il FlyPort Wi-Fi ad una rete infrastruttura predefinita;
- "**conuserX**" per far connettere il FlyPort Wi-Fi alla rete relativa all'indice X (con X= 1, 2, 3, etc...) degli ultimi risultati della scansione (*per esempio conuser2*, comanderà al modulo di connettersi alla rete riferita al secondo risultato scritto sulla seriale);
- **OPEN**: Nel caso di rete "open", ovvero senza sicurezza, non è necessario specificare la password;
- **WEP**:
  - WEP a 40 bit: "**WEP4Yelenco_cifre_chiavi_WEP40**", dove Y è l'indice della chiave da usare ed elenco_cifre_chiavi_WEP40 sono le cifre esadecimali delle chiavi;
    per esempio WEP42AA43AEC7FEA7C-00583C6EB614007F70C1563F777
  - WEP a 104 bit: "WEP10elenco_cifre_chiavi_WEP104", dove elenco_cifre_chiavi_WEP104 sono le cifre esadecimali delle chiavi;
    per esempio WEP106f71652 cc069677573236574313034
- **WPA**:
  - WPA con passphrase: "WPAp_chiave_passphrase_WPA", dove chiave_passphrase_WPA è la chiave di accesso della rete;
    per esempio WPAp12345678
  - WPA con passkey: "WPAk_elenco_cifre_chiave_WPA", dove elenco_cifre_chiave_WPA sono le cifre esadecimali della chiave;
    per esempio WPAk7a00f9648e479c0e8dd-5d5367f8a8acaff91f091beeba0bfe-66eb78f556c0106
- **WPA2**:
  - WPA2 con passphrase: "WPA2p_chiave_passphrase_WPA2", dove chiave_passphrase_WPA2 è la chiave di accesso della rete;
    per esempio WPA2p12345678
  - WPA2 con passkey: "WPA2k_elenco_cifre_chiave_WPA2", dove elenco_cifre_chiave_WPA2 sono le cifre esadecimali della chiave;
    per esempio WPA2k7a00f9648e479c0e8dd-5d5367f8a8acaff91f091beeba0bfe-66eb78f556c0106

L'applicazione non prevede l'utilizzo del webserver, quindi è stato disattivato dal Wizard della openPICUS IDE 2.1. Esiste comunque una application note con un webserver che integra una pagina di configurazione dei parametri di rete, con un'interfaccia che richiama le pagine di configurazione dei comuni router wi-fi. Nella prossima puntata del corso vedremo come customizzare il webserver del FlyPort, per utilizzarlo con una piattaforma hardware di espansione.

I moduli FlyPort Wi-Fi sono disponibili nella versione con antenna integrata (Cod: 8100-FLYPORTWI-FI, Euro 59,90) e con connettore per antenna esterna (Cod: 8100-FLYPORTWIFI-uFL, Euro 59,90) presso: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA) Tel: 0331/799775 - Fax:0331/792287 http://www.futurashop.it

Il 7 Luglio, vieni a Ostia presso il lab di AreaMobile e partecipa al

ConnecThing Hackathon

organizzato da

openPicus, AreaMobile e ElettronicaIn



Cosa portare

Porta il tuo PC, i tuoi strumenti di lavoro preferiti e se hai già qualcosa che vuoi connettere alla rete e hackerare insieme portala con te!

Requisiti

I partecipanti devono essere maggiorenni, avere familiarità con il computer, la programmazione o l'hardware elettronico e avere voglia di imparare e partecipare a una giornata di collaborazione e creatività!

Registrati qui!!!

http://community.openpicus.com/Hackaton

Come arrivare

Vieni presso l'Areamobile lab in Via Delle Baleniere, 151, 00122 Roma (Ostia) - http://goo.gl/maps/cMwH (a pochi metri dalla fermata del Treno Roma-Lido "Lido di Ostia Nord").

Metteremo a disposizione degli hacker romani e non i **prototipi** e i **kit di sviluppo** più incredibili e innovativi per una giornata di **creatività hands on** in cui i migliori talenti avranno inoltre l'opportunità di candidarsi per una posizione da Emdedded Development Engineer proprio per openPicus, la startup Romana che sta rivoluzionando il **Web of Things**.

Grazie a un supporto tecnico e creativo, sarete guidati attraverso la creazione di **oggetti connessi** e di **idee innovative**: i partecipanti saranno divisi in gruppi e alla fine della giornata ci sarà un contest che premierà il **miglior progetto**.

Partecipa per conoscere nuove tecnologie, incontrare persone con cui condividere la tua passione e avrai anche l'opportunità **candidarti** per lavorare nel **core team** di openPicus.



FIRST LEGO League ARRIVA IN ITALIA

Sta per partire il primo campionato italiano tra robot didattici costruiti da giovani Under 14 con i prodotti LEGO MINDSTORMS.

di Davide Scullino

Sono nel pieno dello svolgimento i campionati mondiali Robocup 2012 e da buoni patrioti dividiamo il nostro cuore tra la Nazionale di calcio e le squadre delle nostre scuole partecipanti alla competizione tra robot di Città del Messico, cui facciamo gli auguri affinché escano vincitrici o quantomeno si qualifichino nelle posizioni di vertice della classifica. Mentre seguiamo le imprese delle squadre italiane non possiamo non notare che sull'onda del successo delle competizioni Robocup sta prendendo il via un altro evento di robotica didattica di rilevanza nazionale: si tratta di FIRST LEGO League (FLL), un vero e proprio campionato a squadre che si disputa tra giovani ragazzi che progettano, costruiscono e programmano robot autonomi; l'evento nasce dalla collaborazione tra l'organizzazione FIRST (For Inspiration and Recognition of Science and Technology) e il gruppo LEGO. FIRST (*www.usfirst.org*) è una realtà nata nel New Hampshire per avvicinare i giovani alla scienza e alla tecnologia attraverso programmi pratici ispirati all'innovazione; oltre ad FLL, FIRST organizza Robotic Competi-



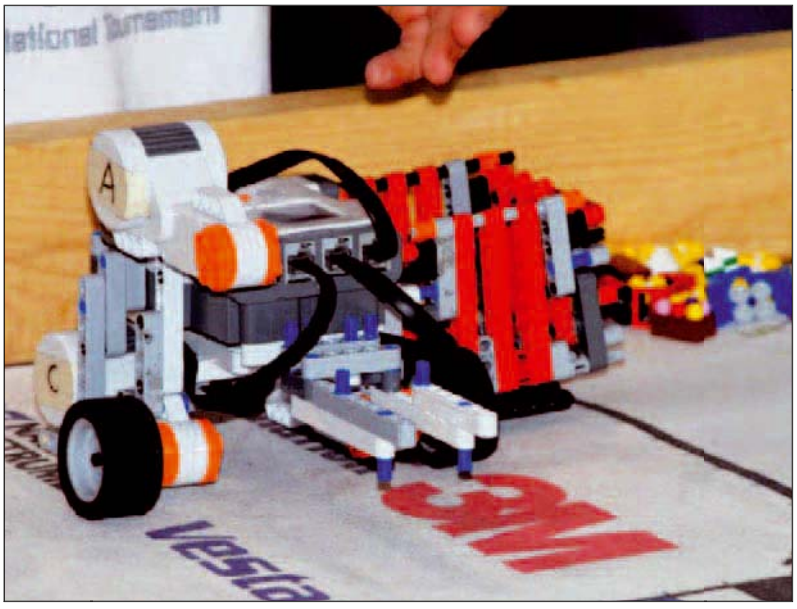
tion e FIRST Tech Challenge. FLL è un progetto di ampio respiro, nato ed evolutosi all'estero, che conta edizioni locali in varie nazioni del mondo, alle quali, nel 2011, hanno partecipato quasi 200.000 ragazzi; da quest'anno approda anche in Italia per avvicinare i giovani alla scienza e alla tecnologia mediante attività divertenti ed entusiasmanti, con lo scopo di far acquisire loro autostima, conoscenza e competenze che serviranno loro nel lavoro e nella vita. I giovani che partecipano alla FLL non apprendono solo nozioni di scienza e tecnologia, ma imparano anche a essere buoni cittadini; infatti i valori guida della competizione sono il senso di squadra e di cooperazione, la ricerca di successi da condividere con il team, lo spirito sportivo e la ricerca del divertimento prima che del successo individuale. FLL coinvolge team composti da non più di dieci ragazzi, di età compresa tra i dieci e i sedici anni (dalla quarta elementare alla seconda superiore) e almeno un tutor adulto.

FLL sfida i giovani a pensare come gli scienziati e gli ingegneri: insieme a tutor adulti, le squadre devono risolvere problemi reali attraverso concetti ingegneristici e matematici. Servendosi della tecnologia e dei prodotti LEGO MINDSTORMS, i ragazzi lavorano in gruppo e imparano a superare ostacoli, interagendo tra loro e con gli adulti che fanno parte della loro squadra. Collaborando tra loro, scambiandosi idee, cercando di risolvere problemi pratici e di coinvolgere sponsor per raggiungere precisi obiettivi, i ragazzi si mettono alla prova, accrescendo la fiducia nelle proprie capacità e sviluppando spirito d'iniziativa. In team i ragazzi con più esperienza aiutano e motivano quelli meno preparati, arrivando a comprendere che ognuno, con le proprie capacità, può contribuire al successo di un progetto condiviso. FLL è un'esperienza unica, che offre ai giovani l'opportunità di confrontarsi con coetanei in tornei e competizioni a elevato tasso di energia agonistica e di essere valutati da una giuria autorevole per la loro capacità di progettare e presentare lavori scientifici, lavorare in squadra, ottenere alte prestazioni nel campo della robotica.

La competizione FIRST LEGO League è strutturata in veri e propri campionati, che si svolgono nelle nazioni più importanti, e fasi di qualificazione, che hanno luogo dove non è previsto lo svolgimento dei campionati veri e propri.

L'edizione italiana è un campionato organizzato con la partnership del Museo Civico di Rovereto; il Museo si articola in sette sezioni unite dalla volontà di coniugare la conservazione della memoria e la valorizzazione delle collezioni storiche con le nuove tecnologie. Maggiori informazioni sull'attività del Museo si trovano al sito *www.museocivico.rovereto.tn.it*.

I team che intendono partecipare a FLL Italia hanno tempo fino a settembre per iscriversi on-line sul sito della competizione (*www.firstlegoleague.org*); la costruzione dei robot dovrà avvenire entro novembre e da novembre a gennaio prossimo avrà luogo la competizione vera e propria. □



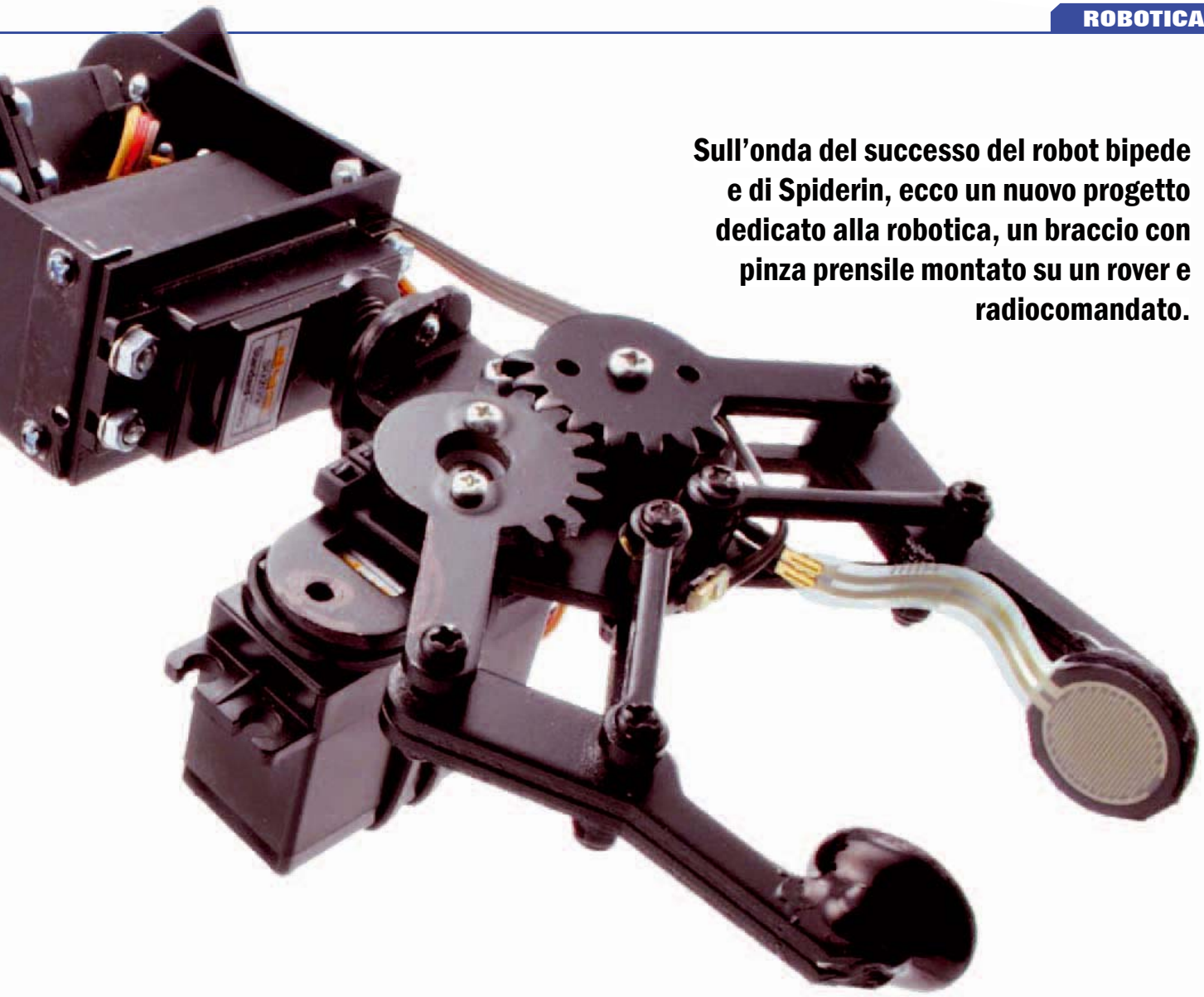
ROVER CINGOLATO CON BRACCIO A PINZA

di FABIO FIORAVANZO





Nel panorama della robotica, ormai si sono realizzate macchine, mezzi e automi di tutte le forme e dimensioni, e capaci di fare cose a dir poco incredibili. Ecco quindi che, dopo il grande successo del robot bipede, e del robot-ragno, Spiderin appunto, abbiamo deciso di presentare sulle pagine della rivista un robot cingolato, dotato di pinza prensile che, comandato dall'apposito telecomando, può effettuare tutti i movimenti di un braccio umano. Le caratteristiche sono di tutto rispetto e ve le andiamo ad elencare.
Come detto l'idea di partenza era quella di realizzare una pinza robotica, ma qualcosa di diverso da



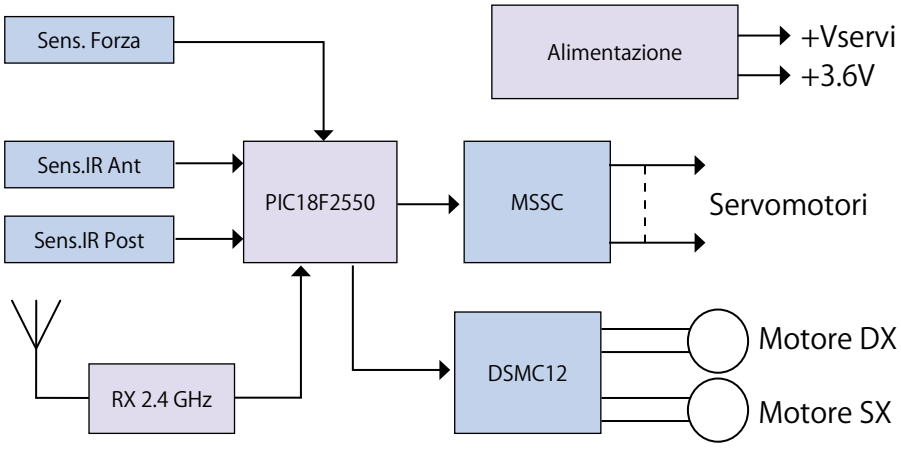
Sull'onda del successo del robot bipede e di Spiderin, ecco un nuovo progetto dedicato alla robotica, un braccio con pinza prensile montato su un rover e radiocomandato.

quelle che si trovano già in commercio su vari siti, che sono per la maggior parte statiche e comandabili solo via PC con programmi dedicati. La nostra idea era di fare un robot mobile, che potesse muoversi anche su terreni accidentati, che potesse essere teleguidato da distanze elevate con assoluta affidabilità, e che fosse anche in grado di muoversi autonomamente. Come prima cosa abbiamo ragionato su quali dimensioni dovesse avere. Per poter superare agevolmente degli ostacoli ed avere un buon appoggio per il braccio robotico, mantenendo il baricentro sempre all'interno della propria figura (come saprete quando il baricentro di un oggetto cade al di fuori del suo perimetro, l'instabilità è garantita), abbiamo realizzato una base di dimensioni 16x22cm. Anche le dimensioni e lunghezze delle varie sezioni del braccio sono state decise a tavolino, anzi al PC, dopo aver disegnato e simulato i vari movimenti al CAD. Il braccio è costituito, così come gli altri pezzi meccanici del robot, con vetronite ramata doppia faccia. Solo per la pinza vera e propria siamo ricorsi ad un pezzo pre-assemblato, e comunque facilmente re-

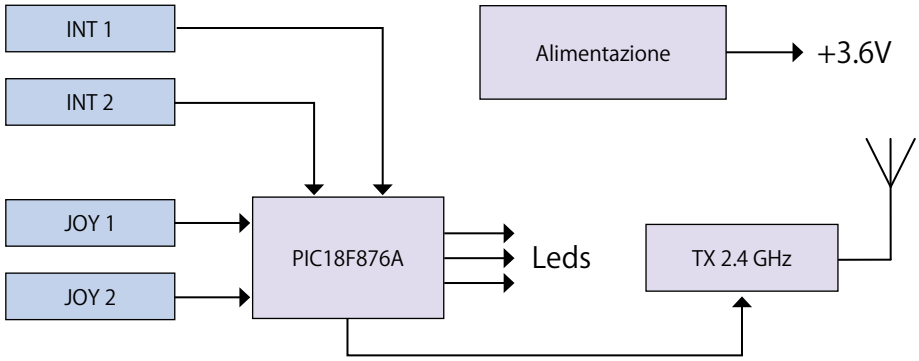
Tabella 1 - Caratteristiche principali pinza robotica

Alimentazione	8x1,2V (Pinza), 9V (Telecomando)
Frequenza di trasmissione	2.4GHz - 78 canali
Sensori di prossimità	1 anteriore, 1 posteriore
Gradi di libertà	6
Sensore di forza	Su chiusura pinza
Buzzer e led di illuminazione	Led bianchi
Telecomando	2 joystick, 2 interruttore, 3 led di segnalazione

Schema a blocchi *Pinza Robotica*



Schema a blocchi *Telecomando*



peribile. Grazie agli 8 servomotori, il braccio ha ben 6 gradi di libertà (per alcuni movimenti usiamo doppi servi per avere più coppia). Il robot può fare in pratica tutti i movimenti di un braccio umano. Riferendoci alla **Fig. 1**, abbiamo una base con cingoli, mossa da due motoriduttori (M1 e M2), al di sopra della quale c'è il braccio vero e proprio che può ruotare di circa 180° sopra la base stessa tramite il servo N°0 chiamato "Braccio". Sulla base sono posizionati gli altri componenti e servi per formare le giunture di "Spalla" (servi N°1 e 2), "Gomito" (servi N°3 e 4) e "Polso" (servi N°5 e 6). Infine troviamo la pinza vera e propria, anch'essa dotata di un servo (servo N°7) per l'apertura e la chiusura della stessa, nonché di un sensore di pressione al fine di dare la giusta presa sul pezzo da prendere. Iniziamo illustrando schematicamente come sono composte le due schede. Nello schema a blocchi della Pinza Robot possiamo vedere lo schema del robot vero e proprio. Il micro è il PIC18F2550 più performante di quello che useremo per il telecomando. C'è uno stadio di alimentazione doppio per poter alimentare i servi (+5V) e la parte logica (+3.6V). Il micro legge i sensori, gestisce il modulo Aurel per la ricezione dati dal telecomando, invia i comandi al modulo MSSC e al modulo DSMC12 che gestiscono rispettivamente gli 8 servocomandi e i 2 motoriduttori dei cingoli. Nello schema a blocchi Telecomando possiamo vedere che la gestione avviene sempre tramite un micro della Microchip, in particolare il modello PIC16F876A che gestisce tutte le funzioni del trasmettitore e cioè lettura dei due joystick, lettura dei due switch, gestione del trasmettitore Aurel a 2.4GHz, gestione dei LED di segnalazione. Il circuito è alimentato da una normale pila alcalina da 9V, nella scheda è presente un regolatore a 3.6V che alimenta tutto il circuito. Descritto sommariamente come sono state impostate le due schede passiamo ad una analisi più approfondita.

SCHEMA ELETTRICO PINZA ROBOTICA

Partiamo dalla sezione di alimentazione, ove troviamo due distinti stadi regolatori, U2 che è un regolatore lineare da 3,6V, uguale a quello che vedremo utilizzato nel trasmettitore. Questo alimenta il micro U3, il transceiver U5, la EEPROM esterna U4, l'amplificatore operazionale U6, i sensori I.R.. Inol-

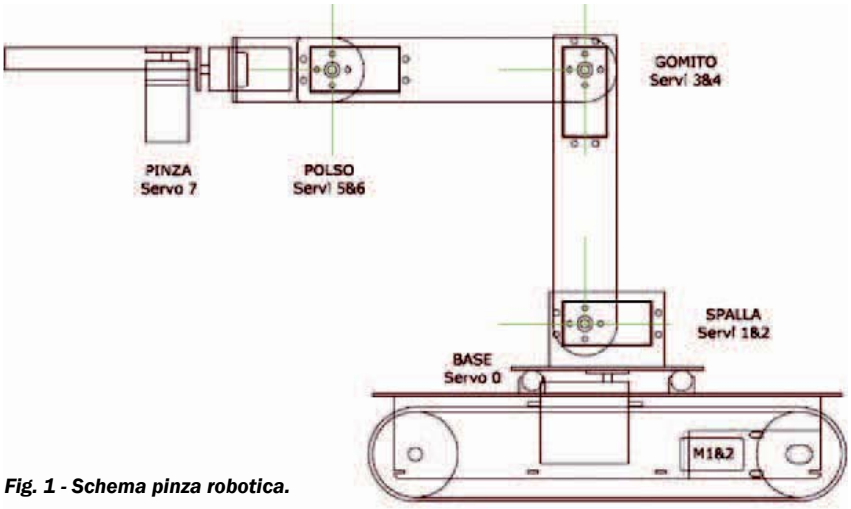


Fig. 1 - Schema pinza robotica.

Schema elettrico *Pinza Robot*

tre ad esso è collegato il led rosso LD1, la cui accensione segnala la presenza di tensione nel circuito. Il regolatore switching costruito attorno ad U1 fornisce invece i 5V per i servi attraverso il modulo di comando U6, nonché alimenta i due led bianchi di illuminazione che trovano posto sul corpo della pinza. Lo schema dello switching è stato ripreso da quello già adottato per il robot Spiderin, ma sono stati rivisti alcuni componenti per irrobustirlo e garantire la corretta alimentazione in ogni condizione. Il condensatore C3 è passato da 1.000 a 2.200uF, l'induttanza L1 è da 5A, mentre il diodo D1 è ora da 4A. Come fonte di alimentazione è stato utilizzato un pacco batterie formato da 8 elementi ricaricabili da 2500mAh posti in serie che danno come risultante una tensione di circa 10V. Notate che l'interruttore SW1 quando è in posizione di OFF, collega il pacco batterie al connettore J1, tramite il quale è possibile effettuare la ricarica del pacco stesso. Come per lo Spiderin, la programmazione del micro viene agevolata dal bootloader precaricato nel PIC, che consente una veloce programmazione via USB. Il rovescio della medaglia di questa soluzione è che il bootloader occupa, oltre ad una parte della memoria FLASH, anche la EEPROM interna del PIC. Ecco quindi che abbiamo aggiunto una memoria esterna (U4) dove memorizziamo le posizioni di neutro dei servi. Tale memoria dialoga con il micro attraverso il bus I2C (pin 7 e 13 del micro). L'ingresso analogico AN0 viene utilizzato per monitorare, tramite il partitore resistivo R1/R2, la tensione di batteria, e se questa è al di sotto del limite impostato, disabilita i servi, ed emette una nota particolare attraverso il buzzer. Altri due ingressi analogici (AN2,3) sono collegati ai due sensori I.R. posti uno nel fronte ed uno nel retro del robot. Infine l'ingresso

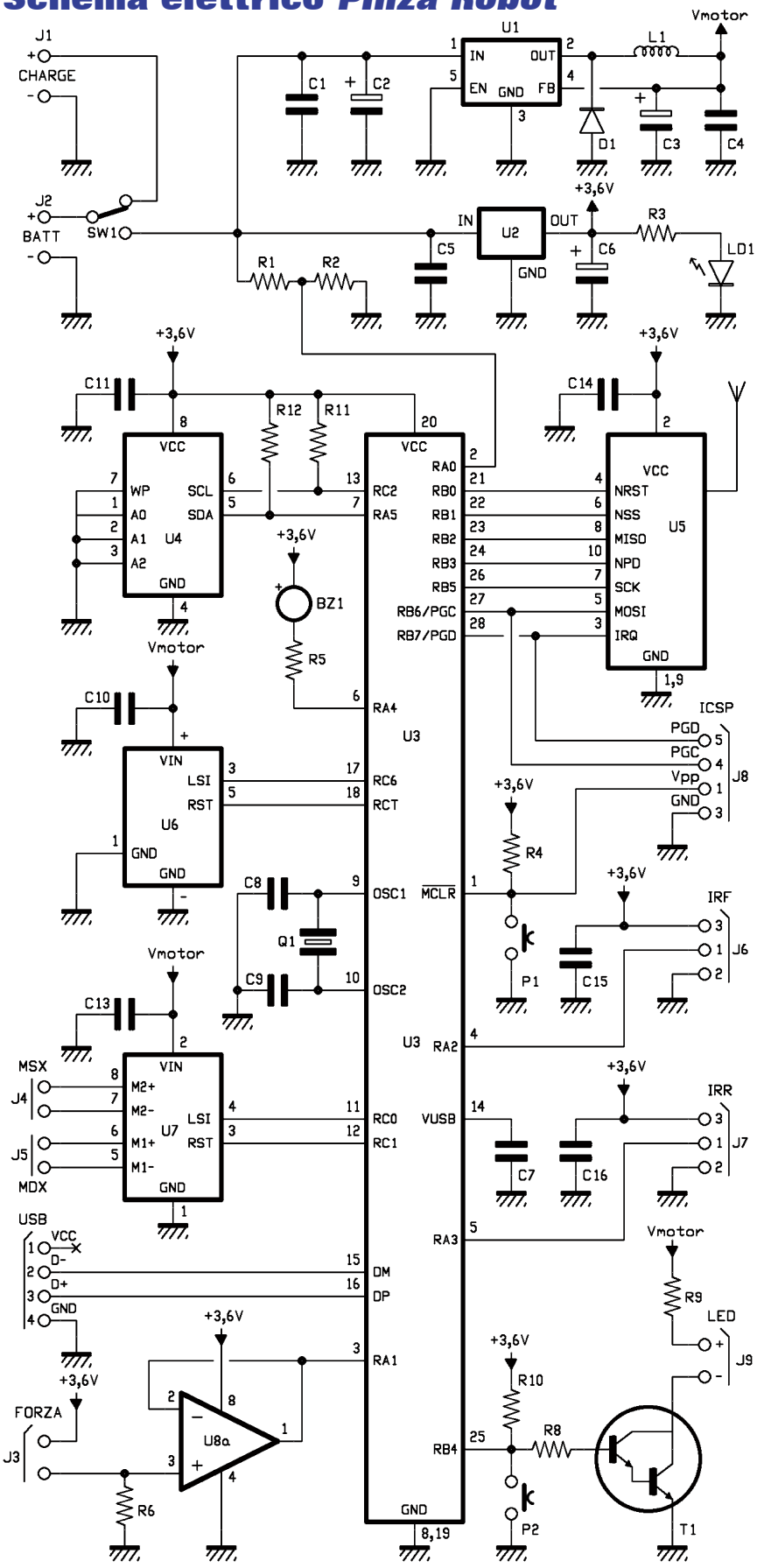


Tabella 2 - Caratteristiche principali modulo Aurel XTR CYP 2.4GHz

Alimentazione	2.7 ÷ 3.6V
Banda di frequenza	2402 ÷ 2481 MHz
Numero canali	78
Tipo di modulazione	GFSK
Potenza RF (ERP) in trasmissione	15dBm
Consumo corrente (RX mode)	60mA
Consumo corrente (TX mode)	100mA

Tabella 3 - Funzioni telecomando

LED SX	JOY SX	Movimento	LED DX	JOY DX	Movimento
OFF	UP	Base avanti	OFF	UP	Pinza apre
OFF	DOWN	Base indietro	OFF	DOWN	Pinza chiude
OFF	DX	Base gira DX	OFF	DX	Braccio gira DX
OFF	SX	Base gira SX	OFF	SX	Braccio gira SX
ON	UP	Abbassa spalla	ON	UP	Alza gomito
ON	DOWN	Alza spalla	ON	DOWN	Abbassa gomito
ON	DX	Abbassa polso	ON	DX	Ruota pinza DX
ON	SX	Alza polso	ON	SX	Ruota pinza SX

Tabella 4 - Funzioni switch telecomando

SW SX	SW DX	Premuti all'accensione
GIU'	GIU'	Leds pinza OFF
GIU'	SU'	Centraggio
SU'	GIU'	Regol. posizione neutro
SU'	SU'	Leds pinza ON

Tabella 5 - Funzioni Joy sinistro in regolazione posizione neutra

LED SX	JOY SX	Movimento
OFF	UP	Incrementa N° servo
OFF	DOWN	Diminuisci N° servo
OFF	DX	Ruota in senso antiorario
OFF	SX	Ruota in senso orario

Tabella 6 - Collegamenti servi al modulo MSSC

Posizione	Servomotore	Prolunga
0	Braccio	NO
1	Spalla 1	NO
2	Spalla 2	NO
3	Gomito 1	NO
4	Gomito 2	NO
5	Polso 1	150mm
6	Polso 2	150mm
7	Pinza	300mm

analogico AN1 è collegato al sensore di forza. Il sensore di forza, posizionato ad una delle due estremità della pinza, presenta in uscita un valore di resistenza che varia in base alla pressione esercitata su di esso (vedere la caratteristica in **Fig. 2**). Viene letto tramite partitore resistivo (R6) e poi bufferizzato tramite U8, un amplificatore operazionale tipo LM358 configurato come inseguitore di tensione. In questo modo abbiamo il corretto adattamento di impedenza verso l'ingresso analogico del micro.
La forza a cui la pinza bloccherà la sua azione di chiusura, dipende da un parametro interno al software di gestione. Una variante al progetto potrebbe essere quella di far variare tale parametro, in dipendenza del valore letto da un trimmer esterno. Nella nostra applicazione però tutti gli ingressi analogici sono occupati, anzi tutti i pin del micro sono occupati, e non è quindi stato possibile implementare questa funzione. La trazione è assicurata da due motoriduttori tipo GM3 che ruotano a 38RPM a 5V, posti sulle ruote posteriori dei cingoli. La gestione di questi due motoriduttori è facilitata dall'utilizzo di un circuito dedicato (DSMC12) che tramite semplici comandi seriali permette di comandare direzione e velocità di due motori DC con assorbimento massimo di 1A cadauno. Gli 8 servi del braccio sono controllati dall'integrato U6 (MSSC), già

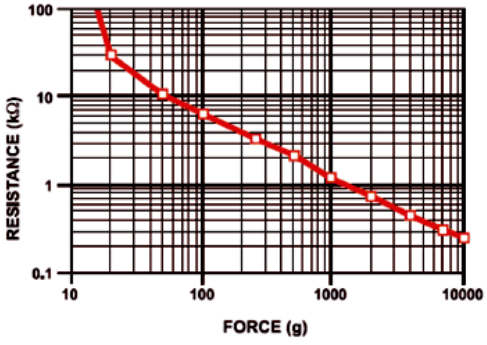
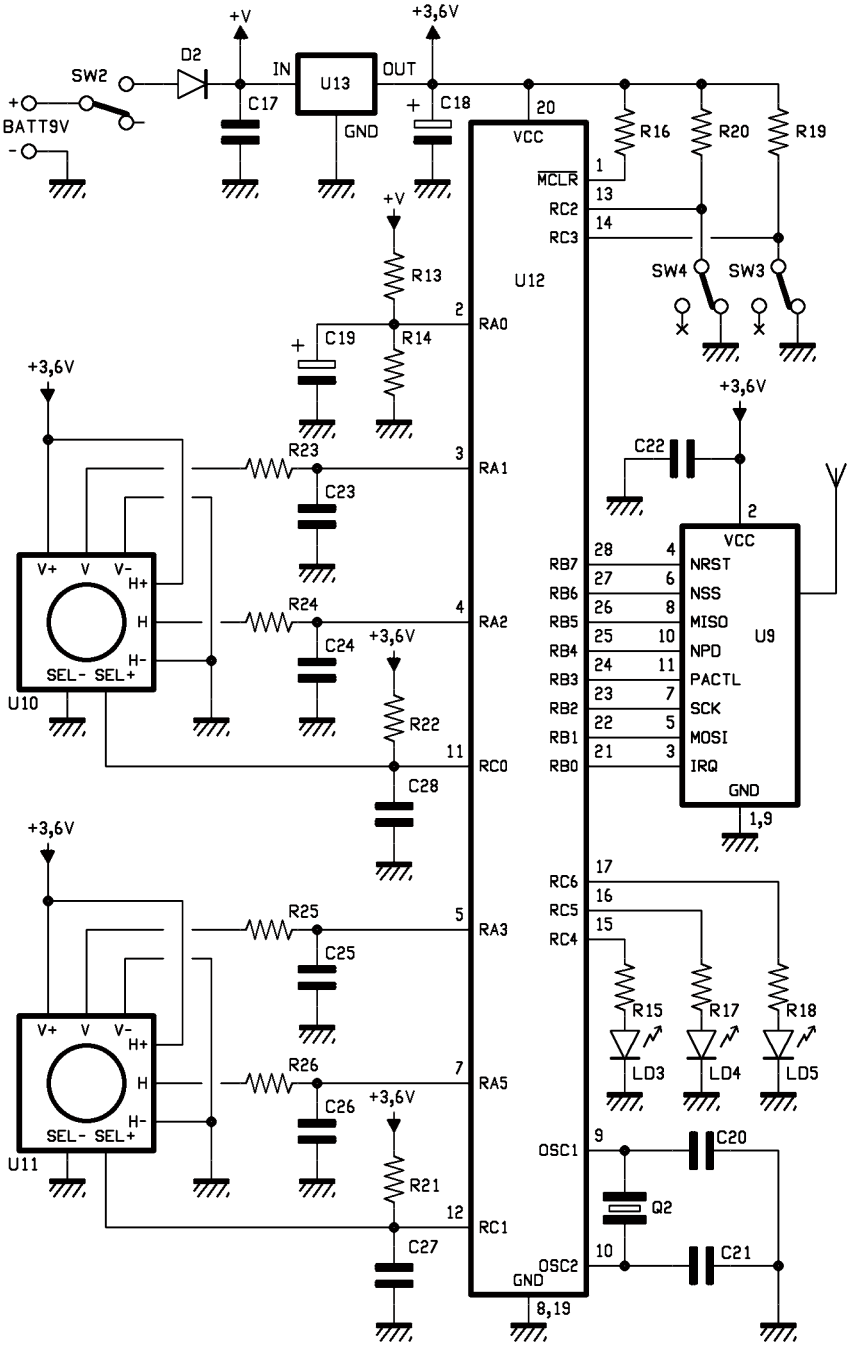


Fig. 2 - Caratteristica di uscita del sensore di forza.

Schema elettrico *Telecomando*



utilizzato in altri progetti, ai quali rimandiamo per la descrizione. Completano il tutto un buzzer per segnalare anche acusticamente i comandi ricevuti dal trasmettitore o, ad esempio, la batteria scarica. Non manca il connettore ICSP per la programmazione del micro vergine tramite il programmatore PICKIT2/3 senza necessità di toglierlo dallo stampato. Per la programmazione tramite bootloader si utilizza invece la presa USB. I due pulsanti P1 ed P2 servono per resettare il micro, ed avviare la procedura di riprogrammazione tramite l'apposito programma per PC.

SCHEMA ELETTRICO TELECOMANDO

Per la trasmissione/ricezione dati, anziché utilizzare il "solito" modulo RF a 434MHz, abbiamo scelto un più performante transceiver multicanale della Aurel a 2.4GHz con velocità di trasmissione fino a 64Kbit/sec. Le principali caratteristiche del modulo sono riassunte in **Tabella 2**.

Lo schema elettrico del telecomando si descrive facilmente. Il circuito è alimentato da una normale batteria alcalina a 9V. Dopo l'interruttore di accensione SW2 troviamo un regolatore lineare (U13) che, unitamente ai condensatori di contorno, genera in uscita 3.6V perfettamente stabilizzati e filtrati. Abbiamo dovuto utilizzare un regolatore a 3.6V, anziché il classico a 5V, in quanto questa è la massima tensione a cui può operare il modulo Aurel. Per evitare di dover introdurre circuiti traslatori di livello, e dover complicare inutilmente il circuito, il micro deve essere alimentato alla stessa tensione. Di conseguenza anche tutti gli altri componenti del circuito sono alimentati a 3.6V. Per impartire in modo agevole i comandi, abbiamo pensato di usare due joystick del tipo di quelli utilizzati nel controller della famosa Playstation 2. Ogni joystick incorpora due potenziometri lineari da 10KΩ, uno per l'asse verticale ed uno per quello orizzontale, inoltre è presente un pulsante che si attiva quando il joystick viene premuto. Le uscite assi sono lette dagli ingressi analogici del micro (U12) tramite degli opportuni filtri passa basso per eliminare eventuali disturbi. I pulsanti sono invece letti da due ingressi digitali, al pari dei due switch a slitta SW3 ed SW4. Questi quattro interruttori sono tutti dotati di resistenza di pull-up esterna. Un altro ingresso analogico (AN0) viene usato per monitorare la tensione di batteria, tramite un partitore resistivo, e segnalare quindi eventuali anomalie sull'alimentazione. La porta B del micro è interamente utilizzata per la gestio-

Fig. 3 - Assemblaggio base.

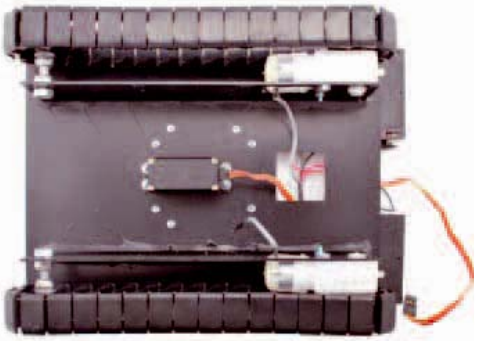
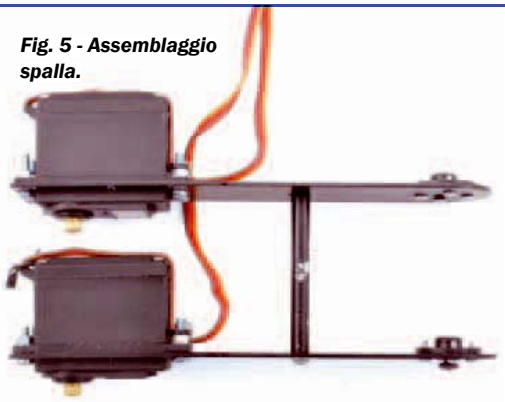


Fig. 4 - Montaggio base rotante.



Fig. 5 - Assemblaggio spalla.



ne del modulo trasmettitore U9. Infine il micro comanda anche tre led di segnalazione. Il led rosso LD4 serve ad indicare visivamente lo stato dell'alimentazione. Se acceso fisso, indica alimentazione presente e stabile, se acceso in modo intermittente, vuol dire invece che la batteria è scarica (tensione inferiore a 7V circa). Gli altri due led, di colore verde, indicano le modalità di lavoro dei due joystick. Infatti, vista la varietà di comandi necessari per movimentare la pinza robotica, ogni joystick ha due serie di funzioni, una quando il led relativo è spento, ed un'altra serie quando è acceso. La variazione si ottiene quando si preme il joystick, cioè si aziona il pulsante al suo interno. Tutti i comandi sono riassunti nelle **Tabelle 3 e 4**. Una spiegazione più approfondita sarà data in seguito quando analizzeremo il software di gestione.

REALIZZAZIONE E MONTAGGIO MECCANICA

Veniamo ora alla parte pratica del progetto, ovvero alla realizzazione del dispositivo. Tralasciamo per il momento la parte di realizzazione dello stampato e montaggio dei componenti, e concentriamoci sulla realizzazione meccanica del robot. Tutti i pezzi sono ricavati da vetronite doppia faccia spessore 1.6mm. Come spiegato per altri progetti, l'utilizzo della vetronite è motivato dal fatto che è sufficientemente robusta per fare strutture "portanti", è facilmente reperibile e lavorabile dal comune hobbista, e soprattutto, grazie ai file Gerber che vi forniamo, con poca spesa si possono far realizzare i pezzi da una delle varie ditte on-line che realizzano circuiti stampati. Prima di iniziare il montaggio è necessario assicurarsi che tutti i servi, siano nella loro posizione di centro. Per far ciò si può utilizzare un qualsiasi circuito di pilotaggio servo. Centrati i servi, possiamo iniziare con l'assemblaggio. Sulla grande base per prima cosa si saldano le due traverse laterali, stagnando abbondantemente entrambi i lati. Attenzione ad inserire correttamente i denti delle due traverse nella base (**Fig. 3**).
La disposizione dei denti è fatta in modo tale da evitare inserimenti errati. Successivamente si montano i due motoriduttori tramite due viti M3 con dado. Sui terminali di ogni motore si saldano due fili che terminano ciascuno in un connettore femmina 2 vie passo 2.54mm. Il cablaggio del motoriduttore destro dovrà essere lungo circa 15cm, mentre per il sinistro sono sufficienti 5cm. Sui due motoriduttori si avvitano le ruote di trazione. Le altre due vanno fissate nella parte anteriore delle traverse tramite viti M6X40mm e dadi autobloccanti.
Si montano i cingoli sulle ruote e, a questo punto, si può regolare la tensione dei cingoli facendo scorrere i due motoriduttori (notate che le sedi di fissaggio sono state opportunamente sagomate per permettere questa operazione). Una volta raggiunta la corretta tensione, si fissano definitivamente le viti dei motoriduttori. Sopra la base si fissa al centro uno dei servomotori e le 4 ballcaster con spessore sottostante (quello più spesso dei due forniti a corredo). Le viti che fissano le ballcaster devono essere accorciate per evitare interferenze con i componenti della scheda elettronica che verrà montata in seguito. A questo punto si può fissare la base girevole, non prima di avervi saldato i due supporti. La base girevole si fissa al servo tramite l'adattatore tondo fornito con il servo stesso e 4 viti autofilettanti da 1.9mm. Anche per tutti i restanti collegamenti dei servi si utilizza tale tipo di adattatore. Sui due supporti si fissano altri due servi, utilizzando per ognuno quatro viti 3MAx10mm con relativi dadi e rondelle (**Fig. 4**).
Passiamo ora alla realizzazione di spalla e gomito. La spalla si assembla utilizzando due pezzi lunghi uniti da un distanziale da 6x60mm filettato alle estremità. Si fissano poi due adattatori tondi per servi con le viti autofilettanti 1.9mm, e altri due servi con le viti 3MAx10mm (**Fig. 5**). In maniera del tutto simile si realizza il gomito, solo che in questo caso si monta un solo servo (**Fig. 6**). Dall'altro lato si fissa una vite 4MAx20mm. Passiamo al polso. Si monta la basetta su un servo e le due squadrette metalliche a 90°, si fissa il tutto con le solite viti 3MA. Alle due squadrette si fissano i due supporti corti con viti 2MA. (**Fig. 7**). La pinza è disponibile già assemblata, è sufficiente avvitarla al servo del polso tramite due viti autofilettanti. Per aumentare l'area di presa, e poter fissare adeguatamente il sensore di forza, sono state incollate alle estremità della pinza due basi tonde, sempre ricavate

Fig. 6 - Assemblaggio gomito.



Fig. 7 - Assemblaggio polso.

da ritagli di vetronite, da 20mm di diametro. Inoltre abbiamo sostituito il servo di serie con uno avente una coppia maggiore. Vi ricordiamo che il sensore di forza non deve essere incollato con ciano acrilato o colle simili, pena il suo danneggiamento. Noi abbiamo utilizzato del nastro biadesivo. Una volta assemblate le varie parti, si uniscono assieme sui perni dei vari servi. Cercate di rispettare le angolazioni il più possibile, come da **Fig. 1**.
Eventuali piccoli disallineamenti potranno essere corretti successivamente via software, variando le posizioni di neutro dei servi. I cavetti dei servi vanno raccolti lungo il braccio e vanno passati nell'apposito foro dietro la base per essere connessi al modulo MSSC. Per alcuni servi occorrono delle prolunghe le cui lunghezze si ricavano dalla **Tabella 6**. La scheda elettronica principale viene montata ad incastro, negli appositi fori della piastra base. I due porta batterie si fissano alle estremità posteriori della base tramite del biadesivo. Sulla sommità della pinza sono posizionati due LED di illuminazione di colore bianco collegati in parallelo. Una piattina a 4 fili collega il sensore di forza e i LED fino ai relativi connettori sulla scheda (J3 e J9). Anche in questo caso per il collegamento si usano strip femmina a 2 poli, passo 2,54mm.

REALIZZAZIONE E MONTAGGIO ELETTRONICA

Il montaggio dei componenti sulle schede segue le solite regole; facendo riferimento ai rispettivi piani di montaggio, si parte a montare i componenti a più basso profilo (resistenze, pulsanti, zoccoli, ecc.) e si lasciano per ultimi i più ingombranti. In particolare per la scheda del ricevitore, i due condensatori elettrolitici C2 e C3 e l'induttanza L1. Raccomandiamo di dotare i regolatori U1 e U2 di idonei dissipatori. Per montare i due sensori I.R. abbiamo utilizzato distanziali esagonali da circuito stampato M3 sovrapponendone uno da 10mm su uno da 20mm. Il connettore è passo 2mm, si può quindi utilizzare le stesse strip femmina utilizzate per la connessione dei transceiver Aurel. Prima di procedere al montaggio degli integrati, si alimenta la scheda e si verifica che siano presenti le alimentazioni ai pin di massa e Vcc del micro, della memoria e dei due moduli MSSC e DSMC12, avendo l'accortezza di montare provvisoriamente una resistenza da 100Ω tra i terminali di alimentazione dell'MSSC al fine di dare un minimo carico all'alimentatore switching. Se tutto è ok, potete montare gli integrati. L'assemblaggio del trasmettitore è abbastanza semplice. Le uniche precauzioni riguardano la saldatura dei 3 LED, rialzati rispetto alla scheda, in modo che una volta montata la scheda stessa dentro al contenitore, risultino a filo del coperchio o poco più. Stessa avvertenza per i due switch a levetta che andranno saldati non sulla scheda, ma rialzati, tramite spezzoni di reofori. La scheda viene poi fissata al coperchio del contenitore tramite distanziali da 15mm. Il contenitore utilizzato è il modello COFFER3 reperibile presso Futura Elettronica. Sul sito è disponibile il disegno della cover da porre sul coperchio del contenitore. Una volta assemblata la meccanica e l'elettronica, passiamo alla programmazione dei micro.

INSTALLAZIONE FIRMWARE E CALIBRAZIONE

Lato trasmettitore si deve caricare il firmware PinzaTX.hex. Lato ricevitore è necessario prima eseguire l'inizializzazione della EEPROM. Bisogna cioè caricare nella memoria i valori di default delle posizioni di neutro dei servomotori. Per far ciò nessun servo deve essere collegato, solo la fonte di alimentazione, cioè la batteria. Se il micro è vergine, deve essere prima programmato tramite un apposito programmatore (va benissimo il PICKIT2/3 che è compatibile con il connettore ICSP presente sulla scheda) caricando il file "HID Bootloader - PIC18F2550.hex". Questa procedura non è invece necessaria se acquistate il micro presso Futura Elettronica. Le modalità di programmazione del micro tramite bootloader sono già state ampiamente descritte nell'articolo sullo Spiderin, a cui vi rimando per eventuali chiarimenti. I due pulsanti per attivare la procedura sono stati posti sul fronte della scheda, ed abbiamo utilizzato switch a montaggio verticale per facilitarne l'utilizzo. Si carica sul micro, tramite il bootloader, il programma "Setting EEPROM.hex", si accende la scheda, e quando si sentono i due suoni del

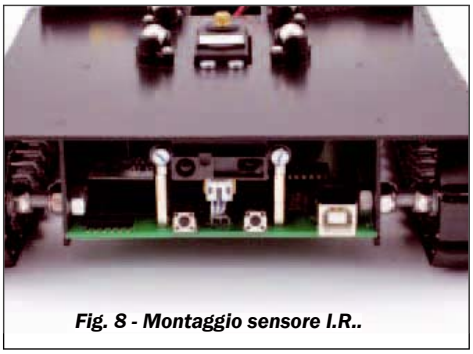


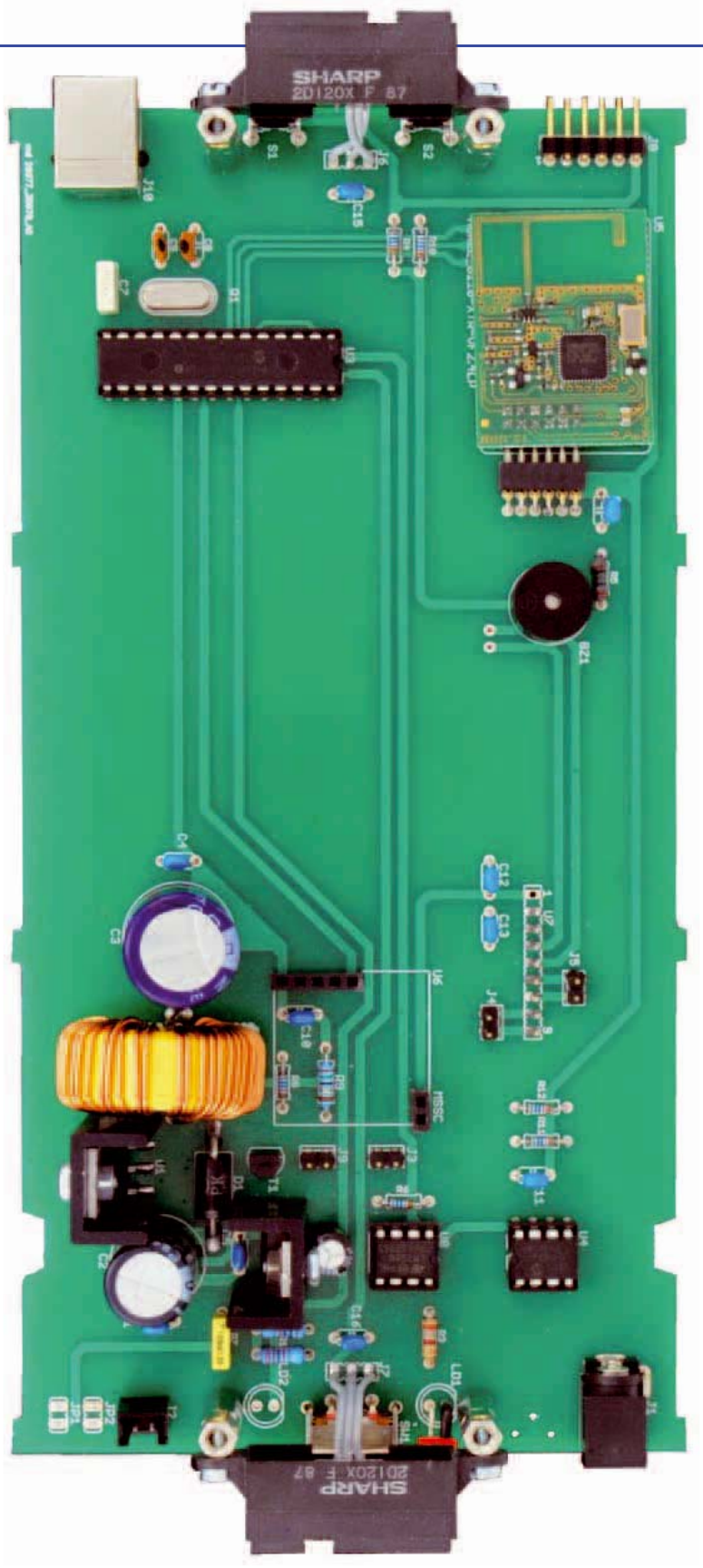
Fig. 8 - Montaggio sensore I.R..

Piano di montaggio
Pinza Robot

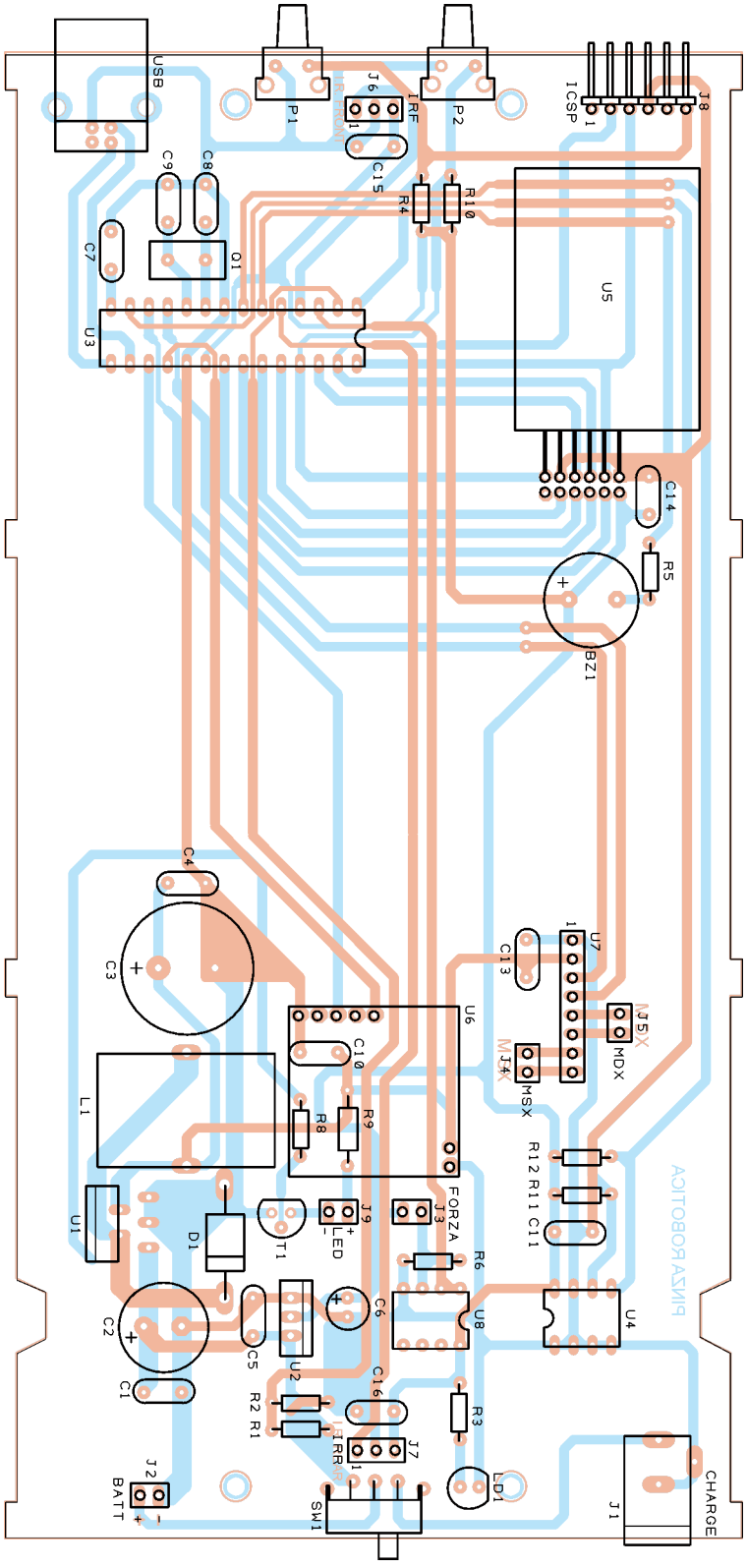
Elenco componenti:

R1, R2: 10 kohm 1%
R3: 180 ohm
R4, R6, R8, R10÷R12: 10 kohm
R5: 100 ohm
R9: 33 ohm 1/2W
C1: 100 nF 63 VL poliestere
C2: 1000 µF 25 VL elettrolitico
C3: 2200 µF 50 VL elettrolitico
C4, C5, C10÷C16: 100 nF 63 VL poliestere
C6: 100 µF 25 VL elettrolitico
C7: 220 nF 63 VL poliestere
C8, C9: 22 pF ceramico
D1: MUR420RL
T1: BC517
LD1: LED 5 mm rosso
Q1: Quarzo 20 MHz
L1: Bobina 220 µH-5.5A
U1: LM2576T-5
U2: LM1086-3.6
U3: PIC18F2550-I/SP (MF1012RX)
U4: 24LC256P
U5: XTR-CYP-24
U6: MSSC
U7: DSMC12
U8: LM358N
BZ1: Buzzer da c.s.
SW1: Interruttore a slitta 90°
P1, P2: Microswitch
SENS1: 7300-SENSOREFORZA
SENS2, SENS3: GP2D120

Varie:
- Plug di alimentazione
- Pin strip femmina 2x6 vie passo 2 mm 90°
- Pin strip maschio 2 poli polarizzato
- Pin strip maschio 2 poli (4 pz.)
- Connettore femmina 3 poli passo 2 mm (2 pz.)
- Pin strip maschio 6 poli 90°
- Connettore USB-B
- Zoccolo 14+14
- Zoccolo 4+4 (2 pz.)
- Dissipatore TO220 (2 pz.)
- Motoriduttori GM3 (2 pz.)
- Portabatterie BH341B (2 pz.)
- Ballcaster PBCASTER38 (4 pz.)
- Ruote per cingoli IMSS4 (4 pz.)
- Cingoli MTLP10 (8 pz.)
- Prolunghe servi SERVOEXT150 (2 pz.)
- Prolunghe servi SERVOEXT300
- Servomotore SERVO213 (8 pz.)
- Circuito stampato meccanica
- Circuito stampato



buzzer, significa che la procedura è stata completata. A questo punto potete collegare i servi, e caricare il programma principale "Pinza RX.hex". Accendete ora la scheda del ricevitore. Il braccio si posizionerà prima a 90°, come in **Fig. 1**, e poi nella cosiddetta posizione di riposo. Per effettuare degli aggiustamenti delle posizioni dei servi è stata prevista una procedura apposita. Per entrare nella procedura



di regolazione dello zero dei servi occorre posizionare lo switch destro in alto e il sinistro in basso, poi accendere il ricevitore. L'entrata nella procedura è segnalata da un suono del buzzer. Tramite il joystick sinistro, muovendolo in senso orizzontale si regola la posizione dello zero avanti e indietro. Muovendo il joystick in alto o in basso si passa alla regolazione del servo successivo/precedente (vedere **Tabella 5**). Effettuata la regolazione si toglie alimentazione per rendere effettive le modifiche effettuate. Attenzione ai servi del braccio e della spalla.

Poiché in questo caso abbiamo 2 servi che lavorano in parallelo, assicuratevi che all'atto del montaggio siano "in linea", cioè che i loro perni entrino bene nelle interfacce senza sforzi. Se ciò non avviene, significa che non sono perfettamente allineati, e ciò ha come effetto un rumore continuo (perché uno o entrambi i servi sono sotto sforzo anche staticamente) ed un "attorcigliamento" del braccio. Il problema si risolve aggiustando lo zero di uno dei due servi finché i perni siano perfettamente allineati. A questo punto il robot è tarato e perfettamente funzionante e potete metterlo all'opera!

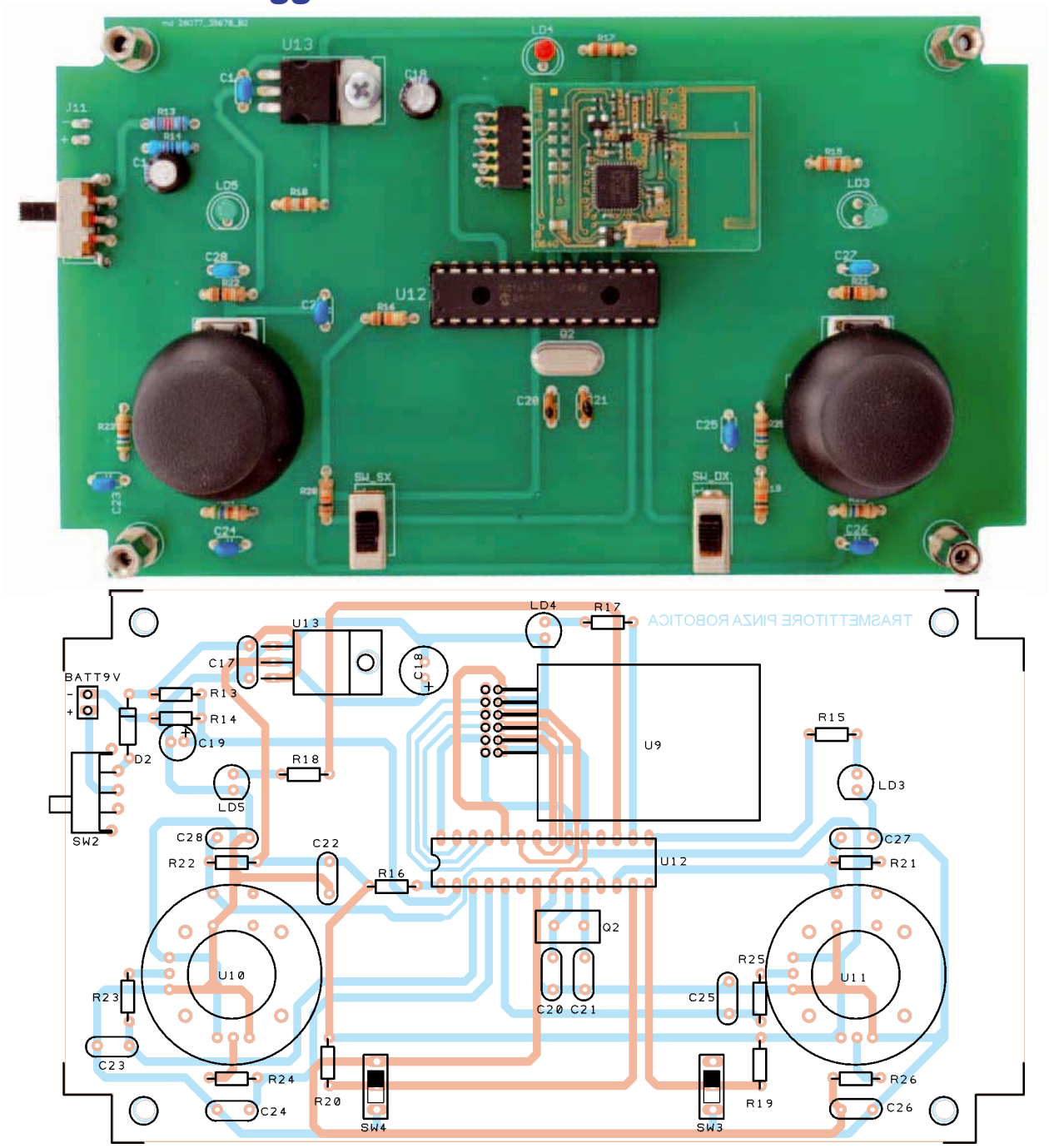
FIRMWARE TELECOMANDO

Diamo un occhio al firmware che gestisce il trasmettitore della pinza robotica. Il programma è stato scritto in Basic, utilizzando il noto compilatore PICBasicPro. Dopo le dichiarazioni iniziali, tra cui si evidenzia l'impostazione del convertitore A/D a 10bit, si inizia effettuando il reset e la configurazione del modulo Aurel.

Seguendo le specifiche del costruttore, vengono impostati, tra gli altri parametri, la potenza d'uscita del modulo a 15dBm, la velocità di trasmissione a 16Kbit/sec e come canale di trasmissione il 40, corrispondente ad una frequenza di 2440MHz.

Tutte le operazioni di comunicazione tra micro e modulo avvengono tramite il bus SPI che fa capo ai pin NSS, MOSI, MISO e SCK del radio modem. Le due routine di lettura e scrittura dei dati sono semplificate dall'utilizzo delle istruzioni SHIFTIN e SHIFTOUT del compilatore Basic. Il MAIN è un ciclo infi-

Piano di montaggio *Telecomando*



Elenco componenti:

R13: 18,3 kohm 1%
R14: 10 kohm 1%
R15, R17, R18: 180 ohm
R16: 10 kohm
R19÷R22: 10 kohm
R23÷R26: 560 ohm
C17: 100 nF 63 VL poliestere
C18: 100 µF 25 VL elettrolitico
C19: 10 µF 50 VL elettrolitico
C20, C21: 22 pF ceramico
C22÷C26: 100 nF 63 VL poliestere
C27, C28: 1 nF 63 VL poliestere
D2: 1N4007
Q2: Quarzo 4 MHz
U9: Aurel XTR-CYP-24
U10, U11: Joystick da CS (CSJOY-STICK)
U12: PIC16F876A (MF1012TX)
U13: LM1086-3.6
LD3, LD5: LED 3 mm verde
LD4: LED 3 mm rosso
SW2: Interruttore a slitta da CS 90°
SW3, SW4: Interruttore a slitta da CS

Varie:
- Zoccolo 14+14
- Clip batteria 9V
- Pin strip femmina 2x6 vie passo 2 mm 90° (2 pz.)
- Circuito stampato

nito dove vengono chiamate in sequenza le funzioni di controllo del livello batteria, lettura degli switch, lettura dei joystick e invio dei dati al transceiver.
La subroutine di controllo del livello tensione batteria effettua, tramite l'ingresso analogico AN0, una serie di 10 misure delle quali poi viene fatta la media. Questo per eliminare eventuali spike o falsi buchi di alimentazione che potrebbero dar luogo a errate segnalazioni. Se il livello risulta inferiore a quello di guardia, viene bloccata l'esecuzione del programma, e viene fatto lampeggiare il led rosso in modo intermittente.
La subroutine lettura switch, legge lo stato dei due pulsanti interni ai joystick e dei due interruttori a slitta, e forma il primo byte da trasmettere.
Allo stesso modo la subroutine lettura joystick, legge la posizione dei due assi di ciascun joystick tramite gli ingressi analogici, e crea il secondo e terzo byte da spedire. Notate che la sensibilità di azionamento può essere aggiustata variando il valore delle due costanti SOGLIA_L e SOGLIA_H. Infine la routine di trasmissione provvede ad attivare il radio modem, ed ad inviargli il buffer dati cui viene aggiunto un header che vale "*" per poter sincronizzare la ricezione dei dati stessi. Il buffer dati è composto come visibile in **Tabella 7**. I tre byte "Switch", "Joy SX" e "Joy DX" sono composti come visibile in **Fig. 9**.
L'invio di ogni singolo byte viene scandito dalla transizione da basso ad alto della linea IRQ del transceiver. Finita la trasmissione del buffer dati, si prepara il modulo ad una nuova trasmissione tramite la subroutine SETRX.

Fig. 9

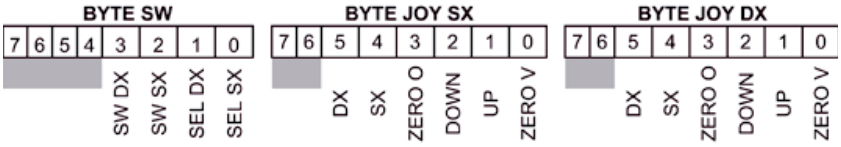
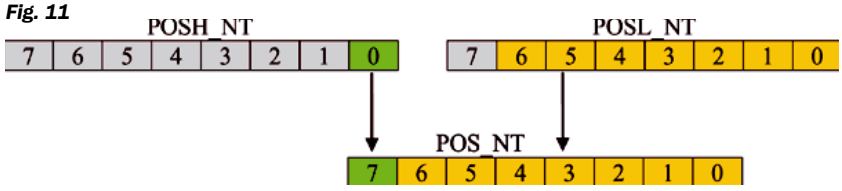


Fig. 10



Fig. 11



FIRMWARE RICEVITORE

Il firmware di gestione del ricevitore è ben più complesso, dovendo il micro gestire, oltre alla ricezione dati, le numerose periferiche di ingresso e uscita. Iniziamo dalla gestione del modulo MSSC, che comanda i servi del braccio. Pur essendoci delle analogie con il firmware dello Spiderin, qui la situazione è decisamente diversa. Rispetto allo Spiderin, dove la rotazione massima dei servi era di 90°, e quindi bastavano 7 bit per gestire l'intero range di rotazione dei servi (0-127 con neutro a 63), qui necessitiamo della massima rotazione possibile cioè di 180° (0-255 con neutro a 127). Per far ciò useremo il COMMAND #3 del modulo MSSC ed invieremo 2 byte di dati dove il DATA1 contiene il bit più significativo, mentre il DATA2 conterrà i restanti 7 bit (**Fig. 10**).
Facciamo un breve cenno al programma "Setting EEPROM.bas". Questo programma si occupa essenzialmente di caricare nella memoria esterna le posizioni di neutro standard dei servi (che corrispondono al valore numerico 127). Quindi per ogni servo andrà caricata la coppia di valori POSH_NT[n]=0 e POSL_NT[n]=127, nelle posizioni di memoria evidenziate in **Tabella 8**. La coppia di valori viene poi conglobata in un unico byte per semplificarne la gestione all'interno del programma (**Fig. 11**). Da sottolineare che anche qui, come nel programma principale, è presente la subroutine di monitoraggio della tensione batteria, che fa la media di 10 misure consecutive, e blocca l'esecuzione del programma (evidenziata dal suono prodotto dal buzzer) nel caso in cui la tensione batteria sia troppo bassa. Questo programma deve essere caricato prima di quello principale, e con il modulo MSSC scollegato. L'esecuzione di questo programma dura pochi istanti ed è scandita da due suoni prodotti dal buzzer all'inizio e fine della procedura.
Veniamo ora al programma principale. Si inizia con le definizioni necessarie per poter utilizzare il bootloader, le inizializzazioni dell'ADC, che opera a 10 bit, la dichiarazione di tutte le variabili e costanti necessarie al programma, per poi passare alla denominazione degli ingressi/uscite e alla configurazione degli stessi. Poi si effettua il reset e con-

Tabella 7 - Composizione buffer dati

BUFF[0]	BUFF[1]	BUFF[2]	BUFF[3]
*	Switch	Joystick SX	Joystick DX

Tabella 8 - Dati memorizzati nella EEPROM esterna

Posizione memoria EEPROM	Servomotore
0→7	POSH_NT 0→7
8→15	POSL_NT 0→7

Guarda il nostro robot



Vuoi vedere come si muove il nostro robot controllato per mezzo del telecomando? Nulla di più semplice! Abbiamo realizzato e postato un video su YouTube all'indirizzo *http://youtu.be/I63XMX49I9w*

figurazione dei moduli esterni, cioè il transceiver Aurel, il modulo comando motori e il modulo comando servi. Notate che prima di resettare l'MSSC, il programma legge dalla memoria esterna i valori delle posizioni di neutro degli 8 servi e ne converte i valori da 2 byte ad un unico byte per poterli gestire più facilmente, come descritto sopra. Concluse le operazioni preliminari, il programma porta i servi in posizione neutra, quindi la pinza assume la posizione di riposo, e si mette in attesa di ricevere i comandi dal trasmettitore. Se all'accensione viene rilevato lo switch destro attivato, si entra nella modalità di centraggio dei servi, dove i servi mantengono costantemente la posizione di neutro. Se invece viene rilevato lo switch sinistro attivato, si entra nella modalità di regolazione della posizione di neutro dei servi. Per uscire da entrambe queste modalità è necessario spegnere e riaccendere la scheda del ricevitore. Si passa poi al MAIN, dove il robot resta in attesa dei comandi che gli arrivano dal telecomando come descritto nelle **Tabelle 3 e 4.** Se si attivano sia lo switch destro che il sinistro, si accendono i led di illuminazione sulla pinza.

Anche in questo caso, il Main è un ciclo infinito dove si eseguono in sequenza le routine di controllo della tensione batteria, di verifica della presenza di eventuali ostacoli (routine Distances), di ricezione dei comandi e di esecuzione dei movimenti. Infine si controlla lo stato dei due switch del telecomando per accendere o meno l'illuminazione della pinza. Il controllo tensione batteria segue la stessa logica di quello implementato nel telecomando, effettuando una media su 10 misure. Se il risultato è sotto il valore limite, determinato dalla costante VBATT_LIM, si disattivano i servi, e il buzzer produce un suono alternato ad indicare la necessità di ricaricare le batterie. La routine Distances legge i valori dei due sensori I.R. e ricava la distanza in cm da eventuali ostacoli. Ricordiamo che i sensori utilizzati possono riconoscere la presenza di oggetti distanti da un minimo di 4 ad un massimo di 30 cm. Durante il funzionamento del robot, se il sensore anteriore rileva ostacoli a distanza inferiore di LIM_F blocca l'avanzamento, così come se il sensore posteriore rileva ostacoli a distanza inferiore di LIM_R blocca l'arretramento del robot. La routine Ricevi, si occupa appunto di ricevere i comandi dal trasmettitore, e di assegnarli ai rispettivi byte in maniera equivalente a quanto viene fatto in trasmissione. Le routine di scrittura, lettura, settaggio e configurazione del transceiver, sono le stesse utilizzate nel trasmettitore. La routine più corposa è sicuramente quella dei movimenti, ove, partendo dai dati ricevuti, li si traduce nelle azioni effettive che devono essere eseguite dal robot. Per i movimenti dei motori, bastano semplici comandi seriali che vengono inviati al MDSMC12 con i quali impostiamo velocità e senso di rotazione dei due servomotori. Per i servi, utilizziamo un'altra subroutine (AZIONA_SERVO) a cui vengono passati come parametri il numero del servo e il verso di rotazione. A sua volta la subroutine AZIONA_SERVO verificherà se la posizione del servo da azionare è all'interno dei limiti prestabiliti, e scomporrà la posizione nei due byte da inviare serialmente all'MSSC. I programmi, come avete visto, sono realizzati con il compilatore PICBASIC-PRO e sono disponibili sul sito della rivista, così come gli schemi per i circuiti stampati e per la meccanica. Chi ha dimestichezza con il Basic potrà personalizzarli a suo piacimento per inventare nuovi movimenti e azioni da far compiere al robot. In particolare i due sensori di prossimità montati sul robot, possono essere utilizzati per creare programmi di navigazione autonoma del robot. Non ci resta quindi che augurarvi buon divertimento! ☐

per il MATERIALE



Tutti i componenti utilizzati in questo progetto sono facilmente reperibili in commercio, mentre gli stampati ed i firmware per il telecomando ed il ricevitore sono scaricabili gratuitamente dal sito della rivista (www.elettronicain.it). Il microcontrollore già programmato per il ricevitore costa 12,00 Euro (codice MF1012RX), il microcontrollore già programmato per il trasmettitore costa 12,00 Euro (codice MF1012TX)
I moduli Aurel XTR-CYP-24 costano 19,00 Euro ciascuno (codice 8110-XTR-CYP-24).
I servocomandi SERVO213 costano 19,50 Euro ciascuno (codice 7300-SERVO213),
Il sensore di forza codice 7300-SENSOREFORZA costa 8,00 Euro. I sensori GP2D120 costano 18,00 Euro ciascuno (codice 7300-GP2D120).
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
www.futurashop.it

IL LABORATORIO NELLE TUE MANI

STRUMENTI PALMARI PRATICI, PRECISI ED AFFIDABILI, ADATTI PER TUTTE LE ESIGENZE, DAL LABORATORIO PROFESSIONALE, AI SERVIZI DI ASSISTENZA TECNICA, ALL'IMPIEGO HOBBISTICO.

cod. **HPS140**

€ 139,00



OSCILLOSCOPIO PALMARE

- 10 MHz di banda passante
- Frequenza di campionamento a 40 MS/s
- 0,1 mV di sensibilità
- Da 1 mV a 20 V/div regolabile in 14 passi
- Base dei tempi da 250 ns a 1 h/div
- Funzione di memorizzazione del segnale (2 memorie)
- Auto set up dei parametri
- Display LCD con retroilluminazione bianca
- Batteria ricaricabile Ni-MH integrata

Fornito con caricabatterie USB



cod. **HPG1**

€ 139,00

GENERATORE DI FUNZIONI

- Gamma di frequenza da 1 Hz a 1 MHz
- Forme d'onda generate sinusoidale, quadra e triangolare
- Risoluzione DAC 10 bit
- Generatore tipo DDS (Direct Digital Synthesis)
- Funzione sweep con modalità bidirezionale
- Distorsione tipica sinusoidale (THD) < 0,1% @ 0 dB / 600 ohm
- Ampiezza massima d'uscita 8 Vpp
- Menu multilingua
- Display LCD con retroilluminazione bianca
- Batteria ricaricabile Ni-MH integrata

Fornito con adattatore BNC maschio / RCA femmina, adattatore BNC maschio / 2 prese banana 4 mm e caricabatterie USB.

ACCESSORI OPZIONALI:



Sonda 60MHz x1-x10 (cod. **PROBE60S** - **€ 19,00**)



Micro adattatore presa accendisigari/USB (cod. **CARSUSB1** - **€ 7,00**)

Custodia protettiva (cod. **BAGPS** **€ 10,50**)



Prezzi IVA inclusa.

FUTURA ELETTRONICA

Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775 • Fax. 0331/792287

Maggiori informazioni su questi prodotti e su tutte le altre apparecchiature sono disponibili sul sito *www.futurashop.it* tramite il quale è anche possibile effettuare acquisti on-line.

ROBOTICA

LA FIERA INTERNAZIONALE DELLA ROBOTICA UMANOIDE E DI SERVIZIO

MILANO • 7-9 NOVEMBRE 2012

fieramilano

4ª EDIZIONE

AREE TEMATICHE:

- MECHATRONICS & COMPONENTS
- MEDICAL & HEALTH
- CYBORG & HUMANOID
- SEARCH & RESCUE
- SAFETY & SECURITY
- EDUCATIONAL
- ENTERTAINMENT
- INVENTORS
- 3D & PROTOTYPING
- ARDUINO WORKSHOP



EDIZIONE 2012*

- 15.000 VISITATORI
- 5.000 METRI QUADRI
- 15 CONFERENZE

*DATI PREVISTI

ORGANIZZATO DA

ARTENERGY PUBLISHING S.r.l.

ARTENERGY PUBLISHING S.r.l.
Via Antonio Gramsci, 57 - 20032 Cormano (MI) Italy
Tel.: +39-02-66306866 - Fax: +39-02-66305510
E-mail: info@robotica.pro

IN COLLABORAZIONE CON

ROBOTICA magazine

www.robotica.pro

MINIBUS, UN BUS PER L'AUTOMAZIONE

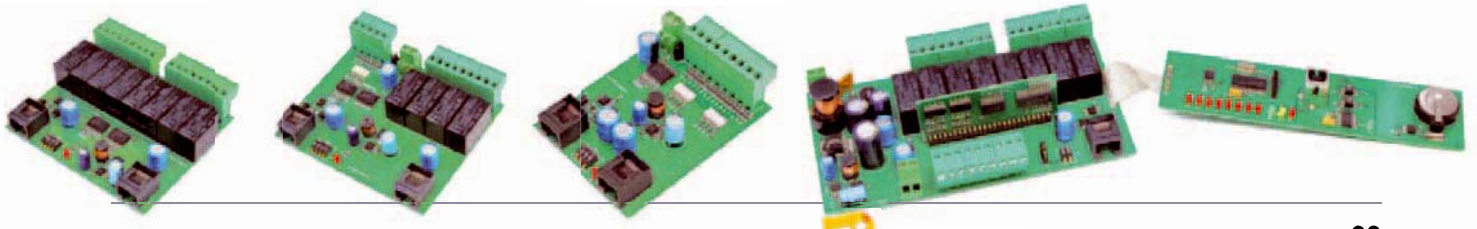
Proseguiamo la presentazione del sistema flessibile di automazione domestica con la descrizione del modulo di espansione e dei moduli periferici specializzati collegati al controllo centrale con il protocollo I²C a livello di segnale potenziato per poter coprire lunghe distanze. Seconda puntata.

di MATTEO DESTRO

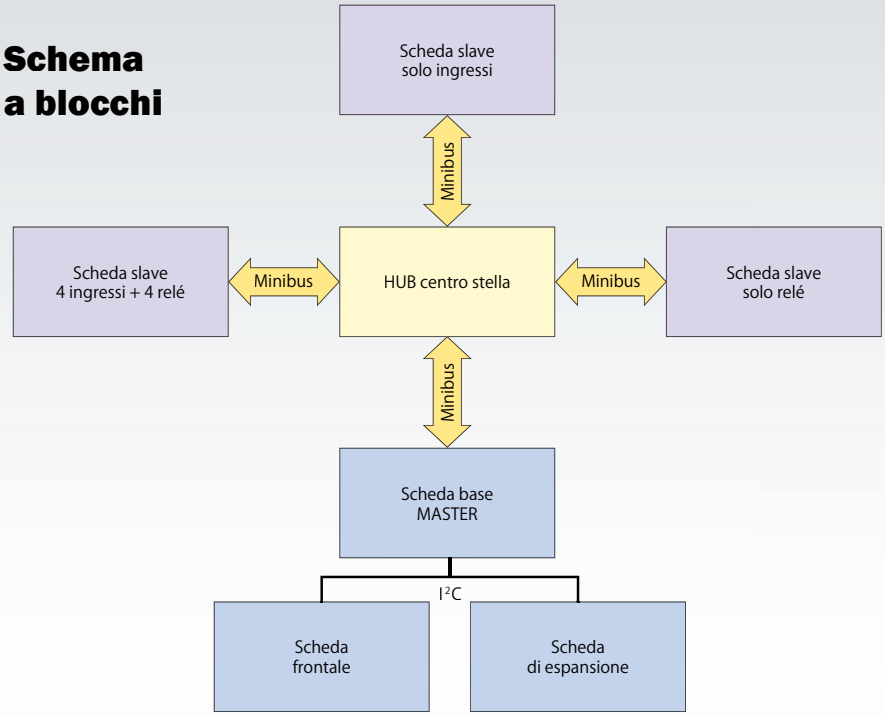
Come già descritto nella puntata precedente la Domotica è la disciplina che si occupa dello studio delle tecnologie atte a migliorare la qualità della vita nella casa e quindi di rendere "intelligenti" apparecchiature, impianti e sistemi. Si può quindi affermare che con "casa intelligente" si indica un ambiente domestico, opportunamente progettato, il quale mette a disposizione dell'utente impianti che vanno oltre il "tradizionale", dove apparecchiature e sistemi sono in grado di svolgere funzioni parzialmente autonome o programmate dall'utente.
Avevamo iniziato la trattazione di un sistema bus che ci permette di gestire in modo automatico dei punti di controllo e comando digitali sia di ingresso che di uscita. In particolare si è parlato del modulo MASTER il quale, grazie all'interfaccia I²C, potenziata per poter comunicare a distanze considerevoli, può interagire con una serie di moduli di ingresso e uscita denominati SLAVE.
In questa puntata tratteremo l'hardware della scheda di Espansione, che completa il modulo MASTER, e dei primi moduli SLAVE. Nella prossima puntata completeremo la descrizione dei moduli SLAVE e introdurremo i concetti base utilizzati per lo sviluppo del firmware memorizzato nel PIC18F2620, cuore del modulo MASTER.

PREMESSA

Come già descritto il sistema si compone di un modulo MASTER, il cervello del sistema, e di una serie di moduli SLAVE (facoltativi) i



Schema a blocchi



Nello schema è evidenziata la logica di collegamento tra i diversi moduli del sistema. L'unità di controllo è composta dalla scheda base MASTER, dalla scheda frontale e dalla scheda d'espansione collegate tra di loro col protocollo I2C. L'unità di controllo si collega alla scheda SLAVE mediante Minibus utilizzando la scheda SLAVE centro stella. Le diverse schede SLAVE sono poi collegate tra di loro, e quindi alla scheda MASTER, utilizzando la scheda HUB centro stella oppure in cascata tra di loro fino ad un massimo di sette schede SLAVE in qualsiasi combinazione tra i diversi tipi disponibili: schede SLAVE solo relé, schede SLAVE solo ingressi e schede SLAVE 4 ingressi più 4 relé.

quali permettono di gestire ingressi e uscite di tipo digitale. Tutti gli ingressi sono optoisolati mentre tutte le uscite sono a relè. Non è obbligatorio avere dei moduli SLAVE collegati al bus, dipende dalle esigenze da soddisfare per il conseguimento del proprio obiettivo progettuale come indicato nella **Fig. 1**, già vista nella precedente puntata. Riprendiamo e completiamo la descrizione del modulo MASTER con lo schema elettrico della scheda di espansione, la quale ospita gli opto-isolatori per gli ingressi digitali, l'integrato ULN2803ADW per la gestione delle uscite a relé (otto transistor Darlington) e l'integrato di espansione degli I/O MCP23017 il quale viene gestito dal PIC18F2620 attraverso l'interfaccia I²C.

SCHEDA DI ESPANSIONE

La scheda di espansione ospita gli opto-isolatori per gli ingressi digitali, l'integrato ULN2803ADW per la gestione delle uscite a relé (otto transistor Darlington) e l'integrato di espansione degli I/O MCP23017 il quale viene gestito dal PIC18F2620 attraverso l'interfaccia I²C.

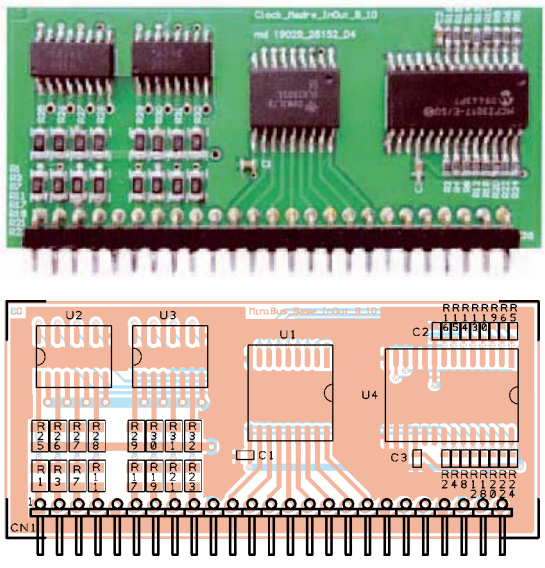
Gli integrati U2 e U3 sono dei classici TLP2804 dove ognuno di essi incorpora quattro opto-isolatori distinti; vedere **Fig. 2**. La sezione di ingresso è formata da due foto-diodi collegati in controfase i quali andranno a pilotare la sezione di uscita formata da un foto-transistor. La tensione di soglia minima dei foto-diodi (VF), affinché entrino in conduzione, è di +1,25V; quindi la tensione in ingresso al modulo deve essere tale da assicurare il superamento di tale soglia. Ovviamente tra la morsettiera e l'anodo del foto-diodo viene interposta una resistenza da 4K7 la quale assicura una protezione in corrente per il foto-diodo. Con questo accorgimento si possono utilizzare tensioni di ingresso da +12Vdc fino a +24Vdc senza che il foto-accoppiatore possa venire danneggiato. La resistenza verso massa da 1K5 serve nel caso in cui l'ingresso di morsettiera venga lasciato flottante assicurando

[piano di montaggio **SCHEDA DI ESPANSIONE**]

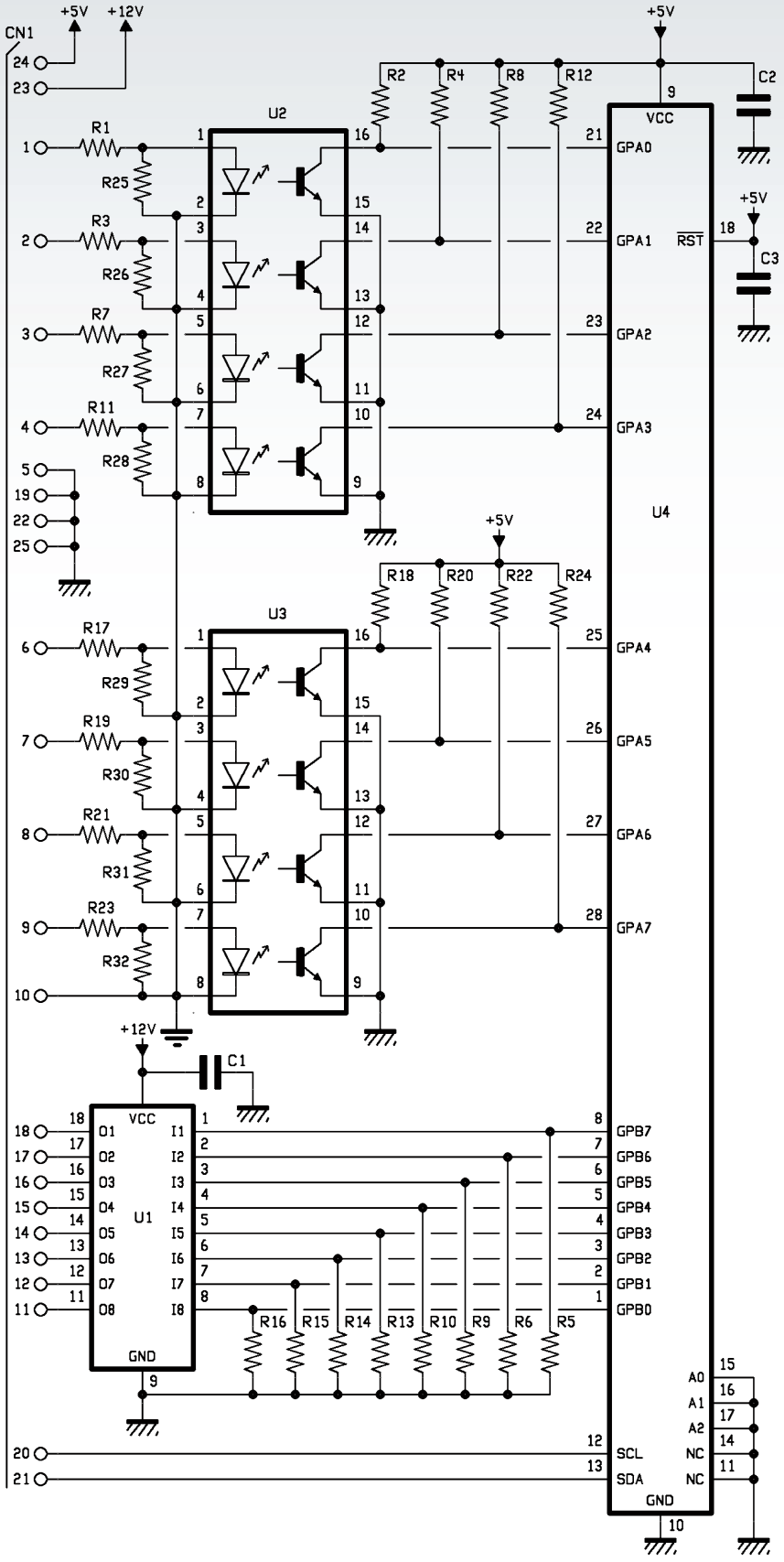
Elenco Componenti:

R1, R3, R7, R11, R17, R19, R21, R23: 4,7 kohm 1% 1206
R2, R4÷R6, R8÷R10, R12÷R16, R18, R20, R22, R24: 4,7 kohm 1% 0603
R25÷R32: 1,5 kohm 1% 1206
C1÷C3: 100 nF 50 VL ceramico 0603
U1: ULN2803ADW
U2, U3: TLP280_4
U4: MCP23017

Varie:
- Strip maschio 25 poli 90°
- Circuito stampato



così un riferimento stabile verso massa. Se si utilizza come tensione di alimentazione degli ingressi i +12Vdc generati dalla scheda base, la massa potrebbe essere quella interna (jumper J1 presente sulla scheda base posizionato su 2-3). Nel caso in cui la tensione di alimentazione degli ingressi sia generata da un alimentatore esterno al modulo stesso deve essere utilizzata anche la massa esterna, jumper J1 presente sulla scheda base posizionato su 1-2. In questo secondo caso si assicura l'opto-isolamento ottimale degli ingressi. Le uscite dell'opto-isolatore, ovvero i collettori dei vari transistor, vanno portati alla porta A dell'integrato U4 la quale è stata configurata come ingresso (lo stesso concetto utilizzato con i PIC). Per ogni ingresso si deve utilizzare una resistenza di pull-up.
Per quanto riguarda le uscite a relé, queste vengono pilotate dall'integrato U1 (ULN2803ADW) il quale a sua volta è direttamente collegato alla porta B di U4, configurata come uscita. In questo caso per ogni pin di collegamento al posto delle resistenze di pull-up devono essere interposte delle resistenze di pull-down.
Parliamo ora dell'integrato U4, un I/O expander della Microchip il quale mette a disposizione ben sedici linee di I/O configurabili indipendentemente come uscite o come ingressi, esattamente come si farebbe con le porte di un microcontrollore. L'integrato MCP23017 può essere ordinato sia con interfaccia I²C sia con interfaccia SPI, nel nostro caso abbiamo optato per l'interfaccia I²C.
Nell'applicazione in esame la porta A (GPAx, pin 21÷28) viene utilizzata come ingresso digitale, mentre la porta B (GPBx, pin 1÷8) viene utilizzata come uscita digitale. I pin 12 (SCL) e 13 (SDA) sono utilizzati per la gestione della comunicazione I²C mentre i pin 20 e 19 (INTA



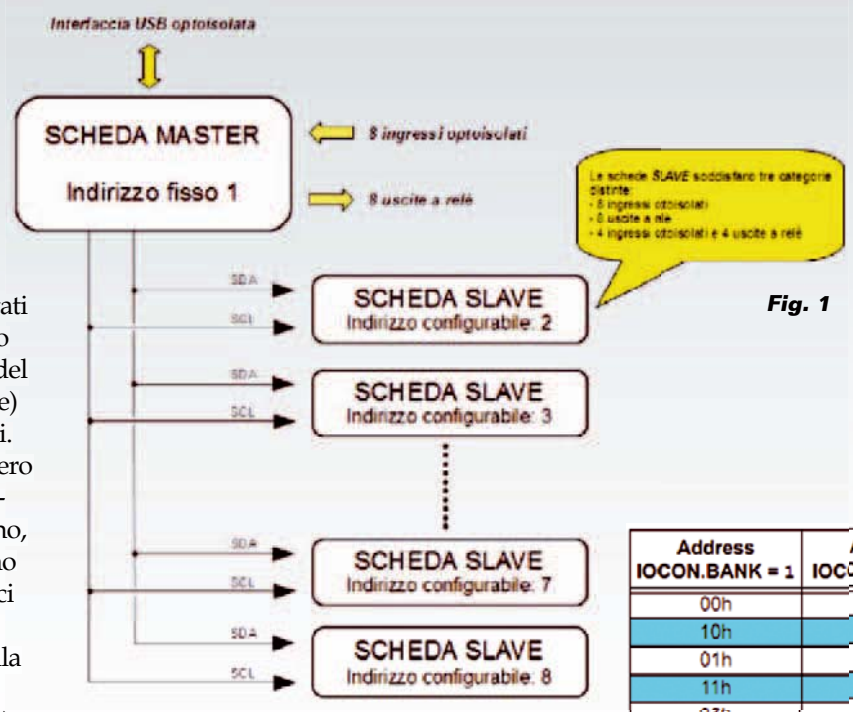


Fig. 1

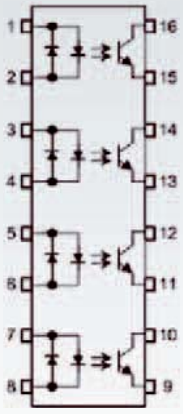


Fig. 2

e INTB) sono lasciati flottanti essendo delle uscite (sono dei pin di interrupt). Nella nostra applicazione non sono stati utilizzati perché è stato utilizzato l'approccio del polling (interrogazione) continuo degli ingressi. INTA e INTB potrebbero venire utili in altri contesti, in quanto indicano, a seconda di come sono stati programmati, se ci sono state delle variazioni sugli ingressi della porta A oppure della porta B. Il pin 19 (reset) viene forzato a +5V.

Internamente l'integrato U4 mette a disposizione una serie di registri i quali devono essere configurati a seconda del tipo di applicazione. La mappatura dei registri interni varia a seconda del valore assegnato al bit ***BANK*** del registro ***IOCON***. Nel caso in cui questo bit sia settato a "1" i registri associati alla porta A e alla porta B vengono separati in due banchi distinti, mentre se il bit è settato a "0" i registri sono memorizzati in modo sequenziale (metodo utilizzato nella nostra applicazione). La **Fig. 3** evidenzia quanto appena detto e mostra l'indirizzamento dei registri in entrambi i casi.

Non descriveremo il significato di ogni singolo registro ma soltanto quelli più significativi per la nostra applicazione. Per primo analizziamo il registro ***IODIRx*** (la x identifica la porta A o B) il quale imposta se i pin della porta x debbano essere configurati come uscite o come ingressi, quindi nel nostro caso la porta A è impostata come ingresso e quindi il registro ***IODIRA*** = 0xFF mentre la porta B come uscita e quindi ***IODIRB*** = 0x00. Il registro ***IPOLx*** serve per impostare la polarità di lettura del registro ***GPIOx***, in altre parole se i bit del registro ***IPOLx*** sono impostati significa che il valore letto dalla porta associata ***GPIOx*** sarà l'inverso del vero valore presente sulla porta, viceversa se i bit di ***IPOLx*** sono a zero. Questo concetto viene sfruttato per decidere come devono lavorare gli ingressi, cioè se devono essere attivi sul fronte di salita o sul fronte di discesa. Il valore da assegnare al registro ***IPOLx*** viene deciso dall'utente attraverso il software di configurazione. Questa configurazione viene applicata alla sola porta A configurata come ingresso. Il registro ***IPOLx*** per la porta B è forzato a 0x00.

Il registro ***GPIOx*** riflette il valore della porta associata, quindi nel caso della porta A, configurata come solo ingressi, riporta lo stato attuale degli ingressi. Invece per quanto riguarda la porta B, configurata come uscita, riporta lo stato assegnato alle uscite.

Infine il registro ***OLATx*** indica il valore che si deve assegnare alla porta configurata come uscita. Per la porta A perde di significato e quindi è forzato a 0x00, mentre per la porta B dipende dal valore che

Address IOCON.BANK = 1	Address IOCON.BANK = 0	Access to:
00h	00h	IODIRA
10h	01h	IODIRB
01h	02h	IPOLA
11h	03h	IPOLB
02h	04h	GPINTENA
12h	05h	GPINTENB
03h	06h	DEFVALA
13h	07h	DEFVALB
04h	08h	INTCONA
14h	09h	INTCONB
05h	0Ah	IOCON
15h	0Bh	IOCON
06h	0Ch	GPPUA
16h	0Dh	GPPUB
07h	0Eh	INTFA
17h	0Fh	INTFB
08h	10h	INTCAPA
18h	11h	INTCAPB
09h	12h	GPIOA
19h	13h	GPIOB
0Ah	14h	OLATA
1Ah	15h	OLATB

Fig. 3

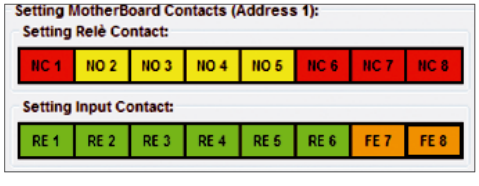
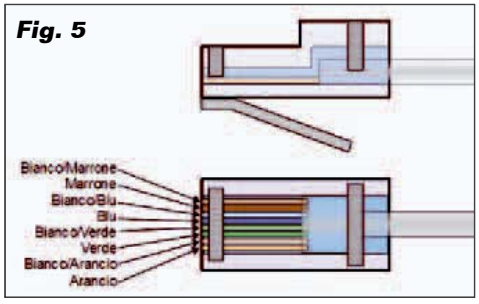
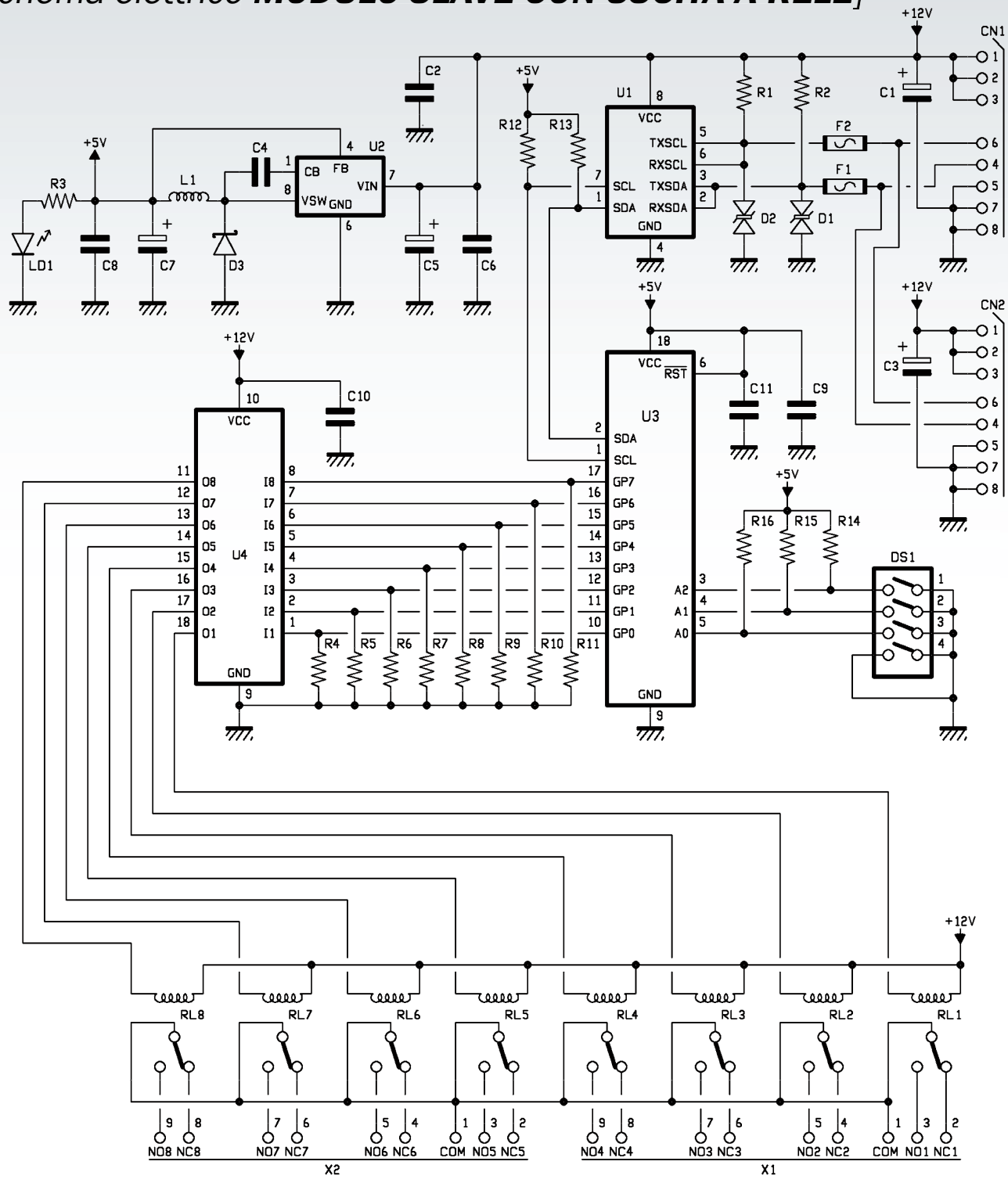


Fig. 4

Fig. 5



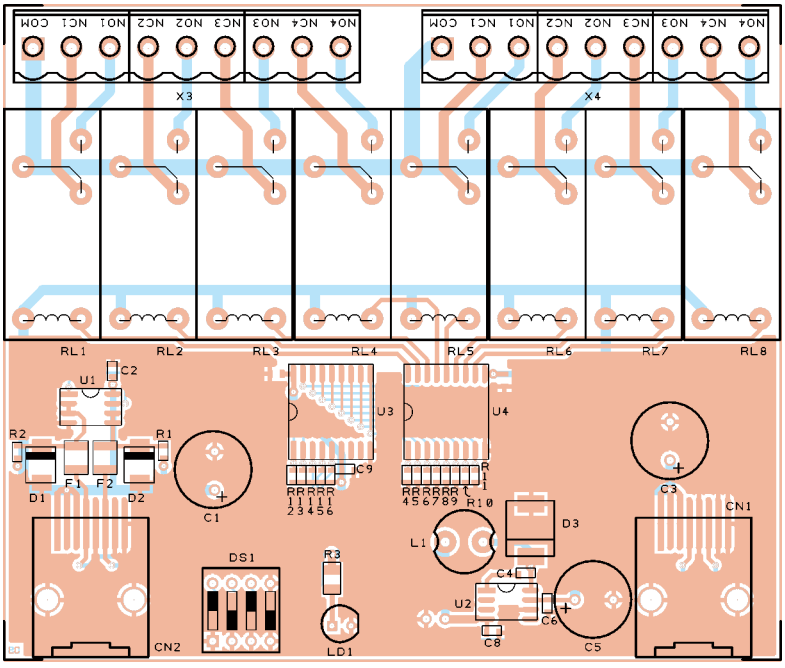
[schema elettrico **MODULO SLAVE CON USCITA A RELÈ**]



in quel momento si vuole assegnare all'uscita (attiva o non attiva). Siccome tutte le uscite sono viste come dei relé l'utente può decidere se queste debbano lavorare come contatto ***NC*** (normalmente chiuso) o ***NO*** (normalmente aperto). Il software genera quindi una maschera da applicare al registro ***OLATB*** attraverso una operazione booleana (***XOR***) tra il valore da passare al registro e la maschera stessa.

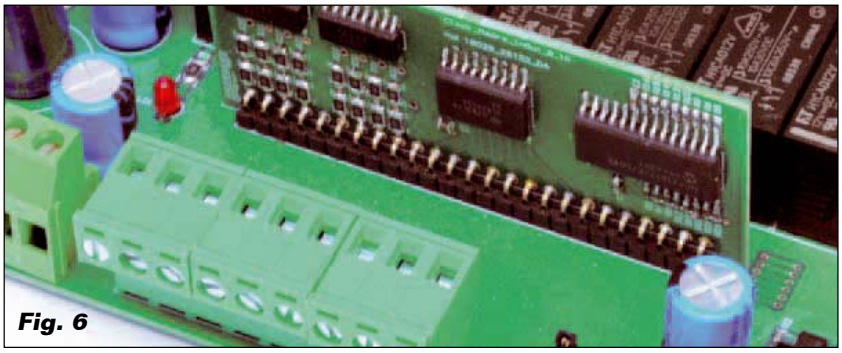
Per chiarire i concetti appena esposti osserviamo la **Fig. 4** la quale evidenzia la configurazione imposta alla sezione di ingresso e alla sezione di uscita per il modulo ***MASTER***. Come si può osservare

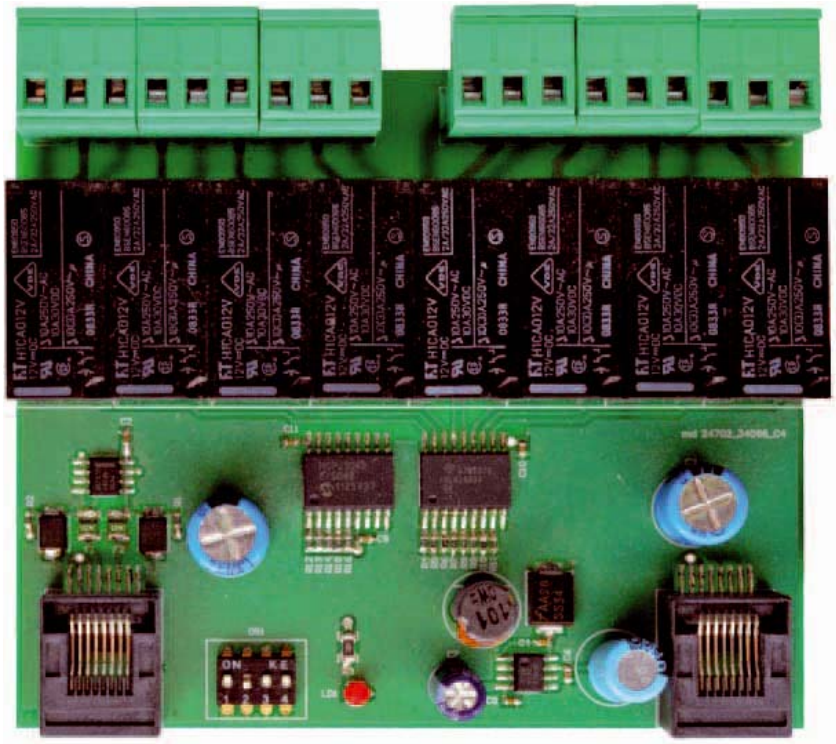
le uscite a relé 2, 3, 4 e 5 sono state configurate come ***NO*** e le uscite 1, 6, 7 e 8 come ***NC***. Ricordiamo che di default i relé montati sulle schede lavorano come ***NC***, quindi con questa configurazione avremo 4 relé non eccitati (relé 1, 6, 7 e 8) e quattro relé eccitati (relé 2, 3, 4 e 5). Attenzione che la maschera ***NON*** è il valore da assegnare all'uscita per eccitare i relé ma dice come l'uscita deve comportarsi a seconda del valore applicato a tale uscita. In più è cura dell'utente collegare in modo corretto le uscite del relé (vedere morsettiera) al carico da pilotare a seconda della maschera applicata. Quindi se il relé n è impostato a funzionare come ***NC***, nel momento in cui l'uscita viene attivata il relé passa nella condizione ***NO*** (se il carico è collegato sul morsetto ***NO*** questo verrà attivato). Se invece il relé è impostato per funzionare come ***NO*** nel momento in cui l'uscita viene attivata il relé passa nella condizione ***NC*** (se il carico è collegato sul morsetto ***NO*** questo verrà disattivato).
Per i dettagli sui collegamenti di morsettiera vedere il piano di montaggio. Ritornando all'esempio per gli ingressi è stato imposto che da 1 a 6 gli ingressi sono attivi sul fronte di salita (***RE***), mentre per gli ingressi 7 e 8 l'attivazione è valida sul fronte di discesa (***FE***). Quindi nel primo caso l'interruttore deve chiudersi (da OFF a ON) portando in conduzione la sezione dell'opto-isolatore associato, attivazione sul fronte di salita. Nel secondo caso l'interruttore deve aprirsi (da ON a OFF) portando in interdizione la sezione dell'opto-isolatore associato, attivazione sul fronte di discesa. La scheda di espansione deve essere inserita nell'apposito connettore sulla scheda *MASTER* come visibile in **Fig. 6**. Per collegare il modulo ***MASTER*** con le schede ***SLAVE*** si utilizza un cavo di rete, categoria 5E, crimpato con i connettori RJ45. La crimpatura del cavo deve essere del tipo 1:1, cioè dritta. Non deve essere assolutamente di tipo incrociato. La **Fig. 5** mostra una possibile crimpatura, entrambe le estremità del cavo devono essere crimpate nello stesso ordine, prestando attenzione alla sequenza colorata dei vari fili.
Il modulo ***MASTER*** è stato progettato per essere inserito in un contenitore modulare per barra DIN. In particolare è stato utilizzato un contenitore 9MH53 della Italtronic (codice 05.0902530) con relativo pannello frontale da lavorare (codice P05090201P oppure P05090201F. Il primo ha la cornice il secondo no).



Elenco Componenti:

R1, R2 R12, R13: 2,2 kohm 1% 0603
R3: 3,3 kohm 1% 1206
R4÷R11, R14÷R16:
4,7 kohm 1% 0603
C1, C3: 220 μF 50 VL elettrolitico
C2, C6, C8÷C11:
100 nF 50 VL ceramico 0603
C4: 10 nF 50 VL ceramico 0603
C5, C7: 100 μF 25 VL elettrolitico
D1, D2: SMBJ13CA
D3: SS34
LD1: LED 3 mm rosso
U1: P82B96
U2: LM2674M_5.0


Fig. 6



U3: MCP23008
U4: ULN2803ADW
L1: Induttanza 100 μH 800 mA
DS1: Dip-Switch 4 vie
RL1÷RL8: Relé 12V JW1FSN
F1, F2: Termofusibile autoripristinante 50 mA

Varie:
- Connettore RJ45 da CS Molex 95501-6889 (2 pz.)
- Morsetto 9 poli ad inserzione (2 pz.)
- Circuito stampato

Passiamo ora a descrivere la famiglia dei moduli SLAVE.

MODULO SLAVE CON USCITA A RELE'

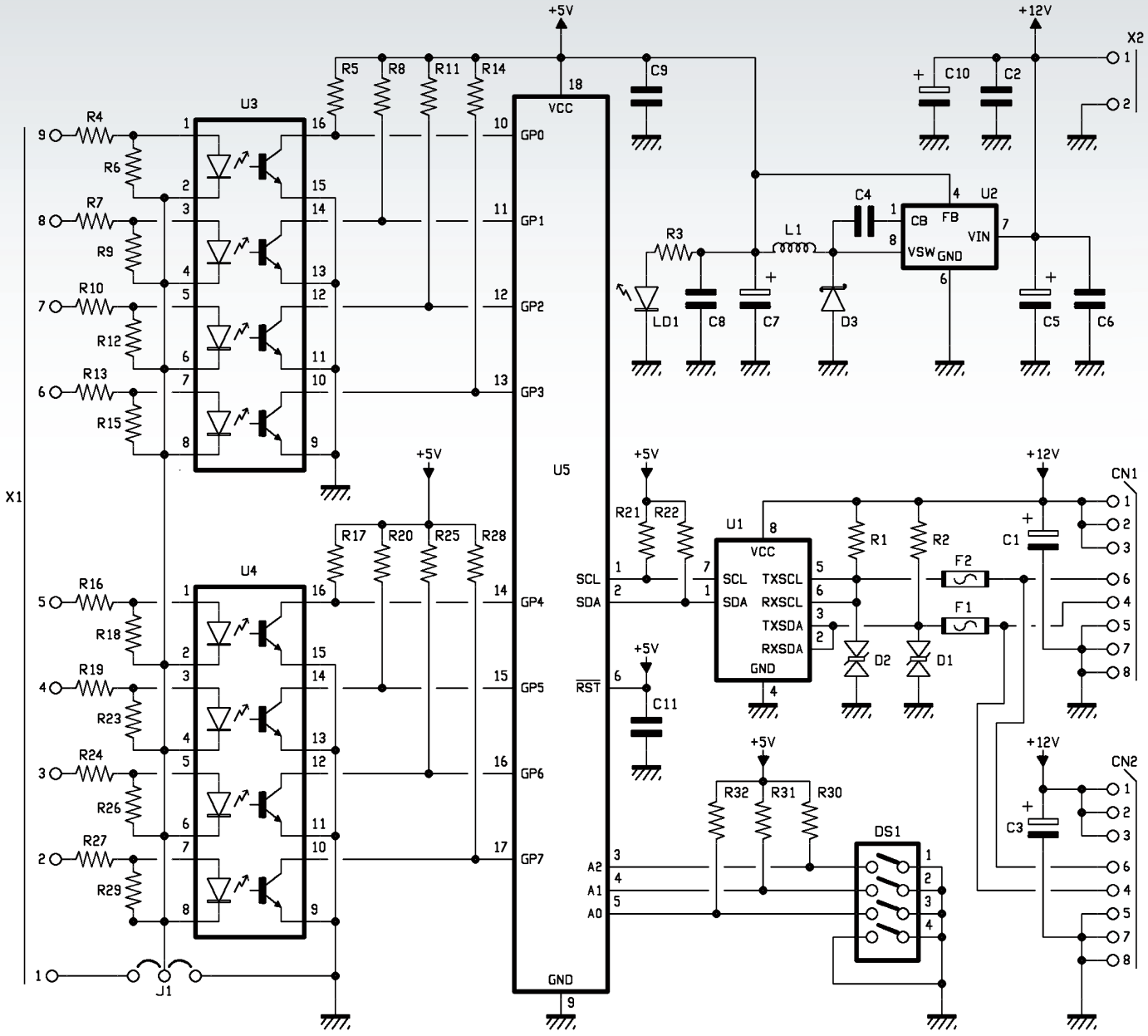
Ricordiamo che il collegamento verso tutti i moduli viene fatto sfruttando il bus I²C in quanto su ognuno dei moduli è presente un integrato MCP23008 (I/O expander) con suddetta interfaccia di comunicazione. La distanza tra il modulo MASTER e i moduli SLAVE può raggiungere anche le decine di metri quindi non è possibile pensare di collegare i moduli utilizzando il segnale I²C con il livello di tensione originale a +5Vdc. Fatta questa premessa possiamo iniziare l'analisi del semplice schema elettrico partendo proprio dall'integrato P82B96 (U1). Il P82B96 è un integrato bipolare in grado di traslare il segnale del bus I²C a un livello di tensione e corrente più alto rispetto a quello di partenza (si passa da +5Vdc a +12Vdc). La **Fig. 7** mostra lo schema a blocchi interno dell'integrato P82B96, come si può notare i pin siglati Sx e Sy vengono utilizzati per collegare, rispettivamente, i segnali SDA e SCL al livello di tensione +5Vdc mentre i pin Tx, Rx e Ty, Ry sono gli omonimi per la parte di segnale traslata di livello (+12Vdc). Come per il modulo MASTER le uscite Tx, Rx e Ty, Ry vengono cortocircuitate assieme e collegate a una resistenza di pull-up a +12Vdc.

Al modulo, collegato con un cavo a 8 poli categoria 5E, giungono i segnali SCL e SDA (traslati di livello) più l'alimentazione a +12Vdc necessaria al corretto funzionamento del modulo. Sulla scheda sono montati due connettori RJ45 (CN1 e CN2) in modo da permettere anche un collegamento a cascata tra diversi moduli. I due connettori sono collegati tra di loro in "parallelo", l'ordine di inserimento dei cavi è indifferente in quanto non esiste un connettore di ingresso e un connettore di uscita.

Il segnale I²C, traslato di livello, viene portato al solo I/O expander MCP23008 (U3) il quale è in grado di gestire fino a 8 linee di I/O indipendenti, ognuna di queste viene collegata agli ingressi dell'integrato ULN2803ADW (U4). Per ognuna delle 8 linee è stata inserita una resistenza di pull-down necessaria nel caso in cui gli ingressi dell'integrato ULN2803ADW rimanessero flottanti (le linee possono rimanere flottanti se l'integrato U3 è configurato in modo errato, se è danneggiato oppure non montato).

Tabella 1

Indirizzo			Dip-Switch			Descrizione
A2	A1	A0	3	2	1	
0	0	0	ON	ON	ON	Non permessa
0	0	1	ON	ON	OFF	Indirizzo logico 2
0	1	0	ON	OFF	ON	Indirizzo logico 3
0	1	1	ON	OFF	OFF	Indirizzo logico 4
1	0	0	OFF	ON	ON	Indirizzo logico 5
1	0	1	OFF	ON	OFF	Indirizzo logico 6
1	1	0	OFF	OFF	ON	Indirizzo logico 7
1	1	1	OFF	OFF	OFF	Indirizzo logico 8



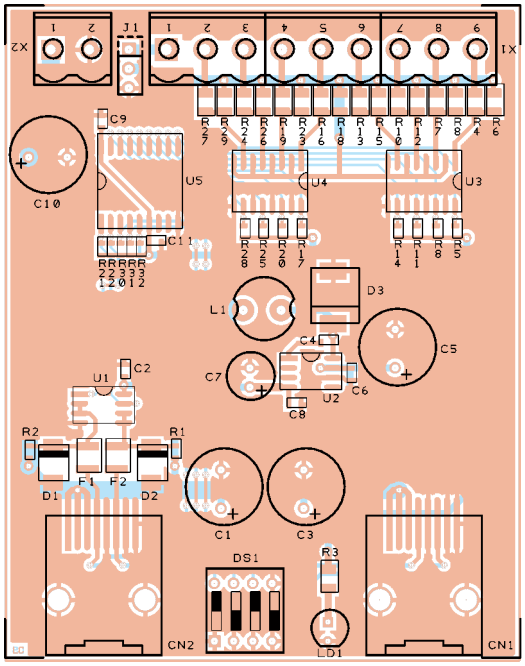
Le uscite dell'integrato U4 vengono portate direttamente alle bobine dei rispettivi relè alimentate a +12Vdc, si ricorda che le uscite di U4 sono dei transistor Darlington in configurazione open-collector.
Ad ogni modulo ***SLAVE*** è necessario assegnare un indirizzo fisico univoco diverso da zero, in quanto tale indirizzo è assegnato al modulo ***MASTER***. Ricordiamo che l'indirizzo logico di ogni modulo è dato dall'indirizzo fisico più uno. Quindi se al modulo **SLAVE** assegniamo indirizzo fisico 1 significa che con il software si dovrà attivare il modulo ***SLAVE*** di indirizzo logico 2 e configurarlo come modulo con sole uscite a relè.
Per assegnare l'indirizzo fisico utilizzare il dip-switch DS1 come evidenziato dalla **Tabella 1**. Per ultimo rimane da trattare la sezione di alimentazione realizzata con il solito integrato LM2674M5.0 (U2) in configurazione step-down già utilizzato per la sezione di alimentazione del modulo ***MASTER***. La corrente massima che può erogare è pari a 500mA con una efficienza massima del 92%. La tensione +5Vdc viene portata a tutti gli integrati presenti sulla scheda tranne che all'integrato U1 che viene alimentato con la tensione +12Vdc.
Sulla scheda è presente il solo led LD1 che evidenzia se il modulo è alimentato e quindi connesso al sistema.
Le morsettiere X1 e X2 mettono a disposizione i contatti ***NC*** (NC = Normalmente Chiuso) e ***NO*** (NO = Normalmente Aperto) dei relè nonché il loro contatto comune COM. Il comune (COM) viene portato solo ai pin 1 delle due morsettiere. Il piano di montaggio evidenzia sia la distribuzione dei

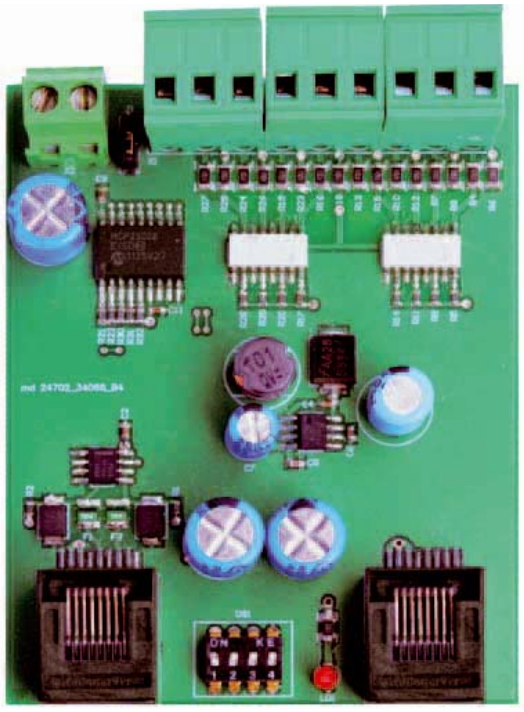
componenti sulla scheda, sia un possibile collegamento dei carichi alla stessa. Si noti che in questo esempio si sfrutta il contatto *NO* per i collegamenti dei carichi.

MODULO SLAVE SOLO INGRESSI OPTOISOLATI

Il modulo in esame ha un hardware pressoché identico al precedente (vedi schema elettrico) salvo per il fatto che gestisce solo ingressi optoisolati. Quindi partendo dal connettore RJ45 ritroviamo l'integrato traslatore di livello P82B96 (U1), l'integrato I/O expander MCP23008 (U5) e la sezione di alimentazione realizzata con l'integrato LM2674M5.0 (U2). L'opto-isolamento degli ingressi viene realizzato sfruttando gli integrati TLP2804 (U3 e U4), ognuno di essi incorpora quattro opto-isolatori distinti; vedere **Fig. 2**. La sezione di ingresso è formata da due foto-diodi collegati in controfase i quali andranno a pilotare la sezione di uscita formata da un foto-transistor. La tensione di soglia minima dei foto-diodi (VF), affinché entrino in conduzione, è di +1,25V; quindi la tensione in ingresso al modulo deve essere tale da assicurare il superamento di tale soglia. Ovviamente tra la morsettiera e l'anodo del foto-diodo viene interposta una

[piano di montaggio **MODULO SLAVE 8 INGRESSI**]





Elenco Componenti:

R1, R2, R21, R22: 2,2 kohm 1% 0603
R3: 3,3 kohm 1% 1206
R4, R7, R10, R13, R16, R19, R24, R27: 4,7 kohm 1% 1206
R5, R30, R31, R32: 4,7 kohm 1% 0603
R6, R9, R12, R15, R18, R23, R26, R29: 1,5 kohm 1% 1206
R8, R11, R14, R17, R20, R25, R28: 4,7 kohm 1% 0603
C1, C3, C10:
220 μF 50 VL elettrolitico
C2, C11:
100 nF 50 VL ceramico 0603
C4: 10 nF 50 VL ceramico 0603
C5: 100 μF 50 VL elettrolitico
C6, C8, C9:
100 nF 50 VL ceramico 0603
C7: 100 μF 25 VL elettrolitico
U1: P82B96
U2: LM2674M_5.0
U3, U4: TLP280_4
U5: MCP23008
LD1: LED 3 mm rosso
D1, D2: SMBJ13CA
D3: SS34
DS1: Dip-Switch 4 vie
F1, F2: Termofusibile autoripristinante 50 mA
L1: Induttanza 100 μH 800 mA

Varie:
- Connettore RJ45 da CS Molex 95501-6889 (2 pz.)
- Morsetto 2 poli ad inserzione
- Morsetto 9 poli ad inserzione
- Strip maschio 3 poli
- Jumper
- Circuito stampato

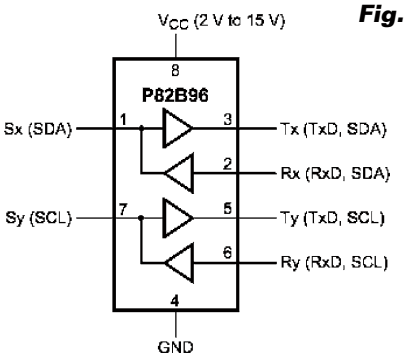


Fig. 7

resistenza da 4K7 la quale assicura una protezione in corrente per il foto-diodo. Con questo accorgimento si possono utilizzare tensioni di ingresso da +12Vdc fino a +24Vdc senza che il foto-accoppiatore possa essere danneggiato.
La resistenza verso massa da 1K5 serve nel caso in cui l'ingresso di morsettiera venga lasciato flottante assicurando così un riferimento stabile verso massa. Il riferimento di massa può essere sia quella interna, se si utilizza come tensione di alimentazione degli ingressi la tensione stabile +12Vdc (jumper J1 posizionato su 2-3), oppure una massa esterna, nel caso in cui la tensione di alimentazione degli ingressi sia generata da un alimentatore esterno al modulo stesso, solo in questo caso si assicura l'opto-isolamento ottimale degli ingressi (jumper J1 posizionato su 1-2). Le uscite dell'opto-isolatore, ovvero i collettori dei vari transistor, vanno portati alla porta dell'integrato U5 la quale è stata configurata come ingresso (si usa lo stesso concetto dei PIC). Per ogni ingresso è stata predisposta una resistenza di pull-up.
Per l'assegnazione dell'indirizzo fisico utilizzare il dip-switch DS1 come descritto per il precedente modulo (vedere la **Tabella 1**).
La morsettiera X1 viene utilizzata per la connessione dei possibili ingressi (pin da 2 a 9, solitamente interruttori e/o pulsanti). Il pin 1 serve per collegare la massa nel caso in cui l'alimentazione degli ingressi venga gestita da una alimentazione esterna alla scheda, in tal caso il jumper J1 deve essere posizionato su 1-2. La morsettiera X2 riporta la tensione +12Vdc utile per alimentare, ad esempio, gli ingressi nel caso in cui si decida di utilizzare l'alimentazione interna stabile +12Vdc, in questo caso il jumper J1 deve essere posizionato su 2-3. Il piano di montaggio evidenzia sia la distribuzione dei componenti sul modulo, sia un possibile collegamento degli interruttori/pulsanti al modulo stesso.

Nella prossima puntata completeremo la descrizione delle schede con il modulo SLAVE 4 ingressi e 4 uscite relè e con il modulo Hub. Parleremo del firmware del sistema e del software sviluppato per la programmazione delle funzioni, messe a disposizione dal sistema stesso, ovvero fasce orarie, configurazione ingressi/uscite, tipologie di moduli ***SLAVE*** collegati al BUS e associazione ingressi/uscite per esecuzione comandi di ***SET***, ***RESET*** o ***TOGGLE*** sull'uscita desiderata. ■



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono di facile reperibilità. I master dei circuiti stampati possono essere scaricati dal sito della rivista così come il firmware utilizzato per programmare il microcontrollore presente sulla scheda master PIC18F2620-I/SO. Quest'ultimo è disponibile non programmato al prezzo di 10,00 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

Elettronica In
Rivista mensile di progettazione elettronica

Sei un appassionato di elettronica? Ti piace progettare, costruire, modificare qualsiasi cosa purché abbia a che fare con transistor, display o microcontrollori? Vorresti condividere le tue conoscenze con gli altri lettori?

Cosa aspetti? ... Entra subito a far parte dei

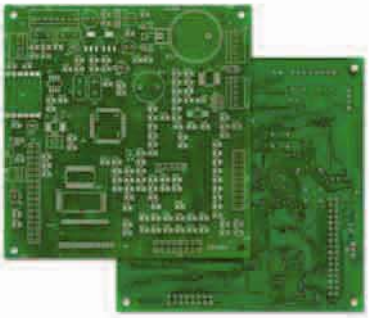
COLLABORATORI HARDWARE della nostra Rivista

Realizzare un progetto per ElettronicaIn è molto semplice!
Dopo aver concordato con la redazione le linee guida del progetto, ti invieremo il materiale necessario con quale farai le prove e realizzerai un primo prototipo, anche su breadboard. Successivamente ci manderai il disegno dello stampato: alla sua realizzazione pratica provvederemo noi che ti rimanderemo la basetta ed i componenti per realizzare il prototipo finale, quello che poi andremo a fotografare e che presenteremo sulle pagine della rivista. Agli schemi elettrici definitivi provvederemo sempre noi mentre a te non rimarrà altro che descrivere il progetto nello stile della rivista.

Non perdere questa opportunità! **Contattaci oggi stesso!**

Manda una mail a: **redazione.hardware@elettronicain.it**

plug in



SINGOLA E DOPPIA FACCIA CON FORO METALLIZZATO; CON SOLDER E SERIGRAFIE PER UNO STAMPATO DI ALTA QUALITÀ O SOLO PISTE STAGNATE PER UN PROTOTIPO A BASSO COSTO

I NOSTRI CIRCUITI STAMPATI HANNO DEI COMPONENTI UNICI. SU MISURA PER VOI.

velocità

TEMPO DI CONSEGNA GARANTITO:
24 ORE O I CIRCUITI SONO **GRATIS**

qualità

I CIRCUITI VENGONO SOTTOPOSTI A **TEST ELETTRICO** E ACCOMPAGNATI DAL **CERTIFICATO DI GARANZIA***;
TUTTI I CIRCUITI SONO REALIZZATI SU LAMINATO **Panasonic**

novità

DA OGGI, ALLO STESSO PREZZO E ALLE STESSE CONDIZIONI DEI NOSTRI CIRCUITI DOPPIA FACCIA È DISPONIBILE **IL LAMINATO IN ALLUMINIO** AD ALTISSIMA DISSIPAZIONE TERMICA INDISPENSABILE PER LA **REALIZZAZIONE DI PROGETTI CON LED** AD ALTA LUMINOSITÀ E OVUNQUE SIA RICHIESTO UN ELEVATO GRADO DI DISSIPAZIONE





facilità

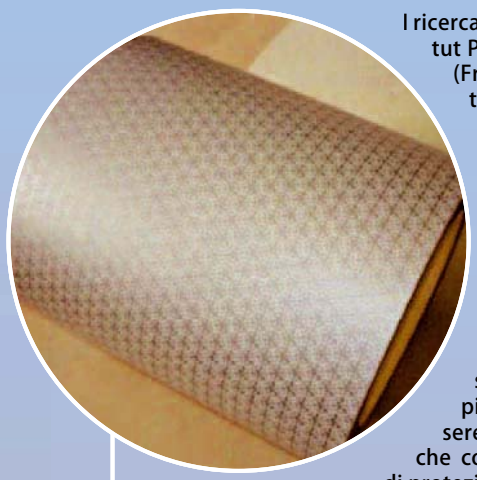
PER CHIARIMENTI, DETTAGLI SULLE NOTE TECNICHE, ORDINI:
www.mdsrl.it
PREVENTIVO ANONIMO, GRATUITO E IMMEDIATO

md millennium dataware
produciamo circuiti stampati

Parco scientifico e tecnologico
15050 Rivalta Scrivia - Tortona (AL)
tel. 0131 860.254 fax 0131 860157 info@mdsrl.it

*la certificazione sarà rilasciata solo per circuiti con consegna oltre le 48 ore

La carta da parati blocca il segnale Wi-Fi

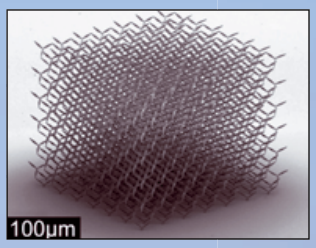


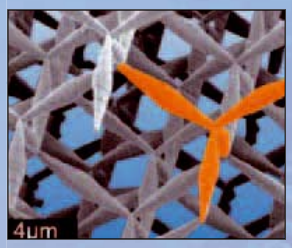
I ricercatori della ESISAR dell'Institut Polytechnique de Grenoble (Francia) stanno sperimentando una nuova carta da parati che riesce a bloccare selettivamente le trasmissioni RF nella gamma di frequenze usate dalle reti Wi-Fi; lo scopo del progetto è isolare alcuni locali dalla rete wireless, per renderli più sicuri da eventuali furti di dati sensibili, tramite "snooping". L'isolamento può essere sia interno che esterno, il che consente un elevato livello di protezione.

La speciale carta da parati, chiamata Métapapier, è progettata per bloccare selettivamente solo alcuni tipi di trasmissione radio, lasciando passare, ad esempio, il segnale della rete telefonica cellulare e altri tipi di segnale, in modo da non creare problemi sulle comunicazioni estranee al Wi-Fi; ciò, grazie alla particolare struttura reticolare della carta da parati, la cui forma è simile ad un cristallo di ghiaccio e che può variare in base alle necessità. Il costo della carta da parati anti Wi-Fi, secondo i ricercatori, una volta avviata la fase commerciale sarà pressoché quello dei normali rivestimenti per pareti.

http://esisar.grenoble-inp.fr/l-ecole/

Metamateriali trasformano il suono





Al Karlsruhe Institute of Technology (Germania) hanno realizzato una nuova classe di materiali attraverso la sintesi di un metafluido cristallino stabile: si tratta di "metamateriali pentamodi", che sono costruiti per comportarsi come fluidi. Averli realizzati apre nuove possibilità nel campo della trasformazione acustica. I metamateriali si ottengono grazie alle nanotecnologie e a tecniche di nano-strutturazione e possono essere creati in modo che la loro struttura influisca significativamente sulle loro proprietà elettromagnetiche, facendoli comportare in maniera diversa rispetto ai loro omologhi naturali. Questo traguardo è stato raggiunto dopo mesi di lavoro presso il Centro per le Nanostrutture Funzionali, del DFG, e presso l'Istituto per la Fisica Applicata del Karlsruhe Institute of Technology. Grazie a questi nuovi materiali, sistemi finora solo teorizzati come prismi acustici, mantelli dell'inaudibilità e altri manipolatori del suono, potrebbero presto diventare realtà. Il prototipo costruito dai ricercatori è stato creato sulla base di un polimero, impiegando una tecnica di litografia ottica di nanostrutture tridimensionali basata su laser. La particolarità del composto realizzato è che la materia che lo compone è poca rispetto all'unità di volume (poco più dell'1%) grazie alla particolare struttura cristallina che rende il metamateriale molto leggero.

www.cfn.kit.edu



Iniezione indolore con la forza di Lorentz

Quanti di voi farebbero a meno delle iniezioni e preferirebbero le siringhe che vediamo in TV nei telefilm di fantascienza? Ebbene, il MIT (Massachusetts Institute of Technology) pare abbia trovato il modo di fare le iniezioni senza pungerci. Infatti il professor Ian Hunter ha realizzato una "pistola" per iniettare medicinali attraverso la pelle e nei muscoli a una velocità fino a 340 m/s e in meno di un millesimo di secondo, senza ago e perciò senza dolore. A differenza di sistemi simili realizzati in passato, basati su una molla o altro sistema meccanico per forzare l'uscita del liquido, la siringa messa a punto al MIT utilizza un attuatore a forza di Lorentz, formato da un pistone posto all'interno di una fiala di medicinale e mosso da un magnete attorno al quale è avvolta una bobina, alimentata da un chip per gestire la velocità di iniezione e la pressione del getto, quindi la profondità, con estrema precisione. Appoggiando l'ugello alla pelle e alimentando la bobina, il magnete e con esso il pistone si muovono, espellendo il medicinale attraverso un foro grande come il pungiglione di una zanzara; l'iniezione avviene in due fasi: nella prima la velocità e la pressione sono molto elevate per perforare la pelle, mentre nella seconda si riduce la pressione per dare al medicinale il tempo di diffondersi e non causare dolore. In attesa della sperimentazione sull'uomo, Hunter e il suo gruppo lavorano a un chip integrato che sarà combinato con un dispositivo in grado di rilevare lo spessore e l'elasticità della pelle del paziente, in modo che l'iniettore possa regolare di conseguenza velocità e pressione.

www.mit.edu

Dragon: missione compiuta



Prima era una competizione politico-militare tra le nazioni che aspiravano alla guida del mondo; oggi, complici i buchi di bilancio dei governi, in gara per la conquista dello Spazio sono entrati anche i privati. Lo ha dimostrato Dragon, la prima navicella spaziale interamente finanziata e costruita da privati; partita a bordo del razzo vettore Falcon 9 nelle prime ore mattutine dello scorso 22 maggio dalla base di Cape Canaveral (un tempo esclusività della NASA), la navicella della società privata SpaceX (Space Exploration Technologies Corp.) è approdata il 25 maggio alla Stazione Spaziale Internazionale (ISS) dove ha portato quei rifornimenti che gli Space Shuttle ormai in pensione, e la promessa Orion mai terminata dalla NASA per ragioni economiche, non avrebbero potuto far giungere. Completata la missione, Dragon si è staccata dalla ISS ed è tornata sulla Terra, dove è ammarata il 31 maggio vicino alla costa messicana. Prima di rientrare nell'atmosfera, la navetta si è separata dagli impianti fotovoltaici che le hanno fornito l'elettricità necessaria alla missione ed ha toccato l'acqua, dopo essere stata frenata da tre enormi paracadute. Capace di trasportare 6.000 kg di carico (in un volume utile di 10 m^3) e un massimo di sette passeggeri, Dragon è lunga complessivamente 4,4 metri ed ha un diametro massimo di 3,66 metri; con i pannelli solari completamente estesi, misura 16,5 metri di larghezza. Per le manovre orbitali e il controllo dell'assetto, conta su diciotto motori a getto ad azoto e diazoto/monometilidrazina, capaci di una spinta di 400 N. A proteggere la navicella durante il rientro nell'atmosfera terrestre, c'è uno scudo termico fatto di un materiale chiamato PICA-X, una variante ad alte prestazioni del Phenolic Impregnated Carbon Ablator (PICA) sviluppato dalla NASA.

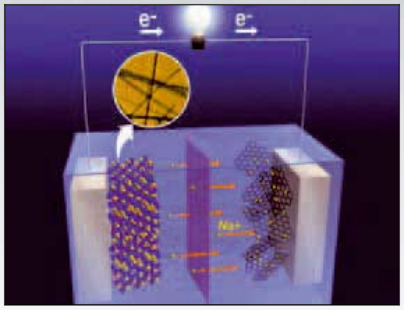




www.spacex.com

Il sodio migliora lo storage dell'elettricità

Le batterie a ioni di litio sono lo stato dell'arte per quanto riguarda i sistemi di accumulo dell'elettricità destinati ai dispositivi elettronici e al settore automotive. Tuttavia, a causa del loro costo elevato, derivante dalla scarsa disponibilità del litio in natura, non si prestano ad applicazioni di grandi dimensioni. L'alternativa per immagazzinare grandi quantità di elettricità (ad esempio negli impianti di produzione da energie rinnovabili o nelle reti elettriche, per accumulare corrente la notte, quando costa meno, e disporne di giorno) è rappresentata dalle batterie a ioni di sodio-zolfo, che però attualmente funzionano a temperature di circa 300 gradi Celsius, il che le rende poco maneggevoli e sicure.
I ricercatori sono quindi impegnati nel trovare una soluzione alternativa: ad esempio mantenere la struttura delle Li-Ion ma farlo utilizzando l'economicissimo sodio nello stesso tipo di elettrodi utilizzati nelle batterie agli ioni di litio.
Al Pacific Northwest National Laboratory (PNNL) del Dipartimento dell'energia degli Stati Uniti, si sta già lavorando per realizzare elettrodi in sodio e ossido di manganese; ciò perché gli elettrodi delle batterie a ioni di litio sono realizzati con ossido di manganese, in cui si formano diversi tunnel a livello microscopico che lasciano transitare gli ioni quando le batterie vengono caricate o erogano corrente.
Sostituire gli ioni di litio con quelli di sodio permette di realizzare batterie ad elevata capacità e basso costo, però gli elettrodi ottenuti sarebbero del 70 per cento più grandi; ciò significherebbe incrementare le dimensioni delle batterie a parità di energia immagazzinata. Per ovviare a questo inconveniente, i ricercatori hanno fatto ricorso a tecniche di manipolazione nanoscopica, grazie alle quali sono riusciti a ingegnerizzare l'ossido di manganese in modo da lasciar fluire anche gli ioni di sodio, pur mantenendo le caratteristiche fisico-chimiche dell'elettrodo.



www.pnl.gov

Telecamere IP

La soluzione ideale per il controllo da remoto tramite LAN o Internet.

Telecamera dome IP

Sensore d'immagine CMOS 1/4" a colori • Risoluzione 640x480 (VGA) • Ottica fissa • Velocità di trasferimento immagine 30fps (QVGA) • Motion Detection • Ingresso e uscita d'allarme • Invio di email con fotogrammi su allarme • Illuminatore a 22 LED IR • Alimentazione 5Vdc (adattatore incluso) • Dimensioni Ø 128 x 96mm.

Versione con interfaccia Wi-Fi / RJ-45 (cod. CP449 • € 89,00)



cod. CP479
€ 74,00
(con interfaccia RJ-45)

Telecamera IP da esterno con Wi-Fi e IR

Sensore d'immagine CMOS 1/4" a colori • Motion Detection • Ingresso d'allarme • Invio di email con fotogrammi su allarme • Formato compressione M-JPEG • Velocità di trasferimento immagine 25fps (VGA) • Illuminatore a 30 LED IR • Alimentazione tramite adattatore da rete (incluso) • IP66 • Dimensioni 146 x 83 x 80mm.



cod. CP391W
€ 139,00
(con interfaccia Wi-Fi / RJ-45)

Versione con interfaccia RJ-45 (cod. CP391 • € 109,00)



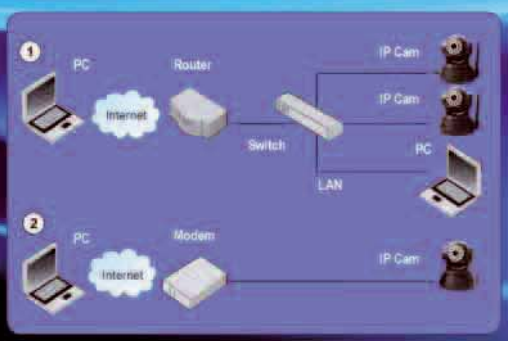
cod. CP439
€ 98,00
(con interfaccia RJ-45)

Telecamera IP motorizzata a colori PAN (270°) / TILT (120°)

Sensore d'immagine CMOS 1/4" a colori • Risoluzione 640x480 pixel (VGA) • Ottica fissa • Brandeggio PAN (270°) / TILT (120°) • Illuminatore a 10 LED IR • Formato compressione M-JPEG • Velocità di trasferimento immagine 25fps (VGA) • Formato video AVI • Alimentazione 5Vdc (adattatore incluso) • Dimensioni 120 x 100 x 135mm



cod. CP439W
€ 118,00
(con interfaccia Wi-Fi / RJ-45)



Telecamera IP da esterno con IR e sensore CCD SONY SUPER HAD

Sensore d'immagine CCD 1/3" Sony Super HAD a colori • Risoluzione 512x582 pixel • Sensibilità 0,5 lux - F1.4 • Ottica f=4mm • Illuminatore a 21 LED IR • Interfaccia Ethernet 10BaseT / 100Base TX • Possibilità di visione da remoto anche tramite telefonino • Formato di compressione MPEG4 (25fps) • Motion Detection • Invio di email con fotogrammi su allarme movimento • Alimentazione 12Vdc • Grado di protezione IP66 • Dimensioni Ø73 x 85mm.



cod. CAMIP4N
€ 290,00



cod. CP360
€ 98,00
(con interfaccia RJ-45)



cod. CP360W
€ 115,00
(con interfaccia Wi-Fi / RJ-45)

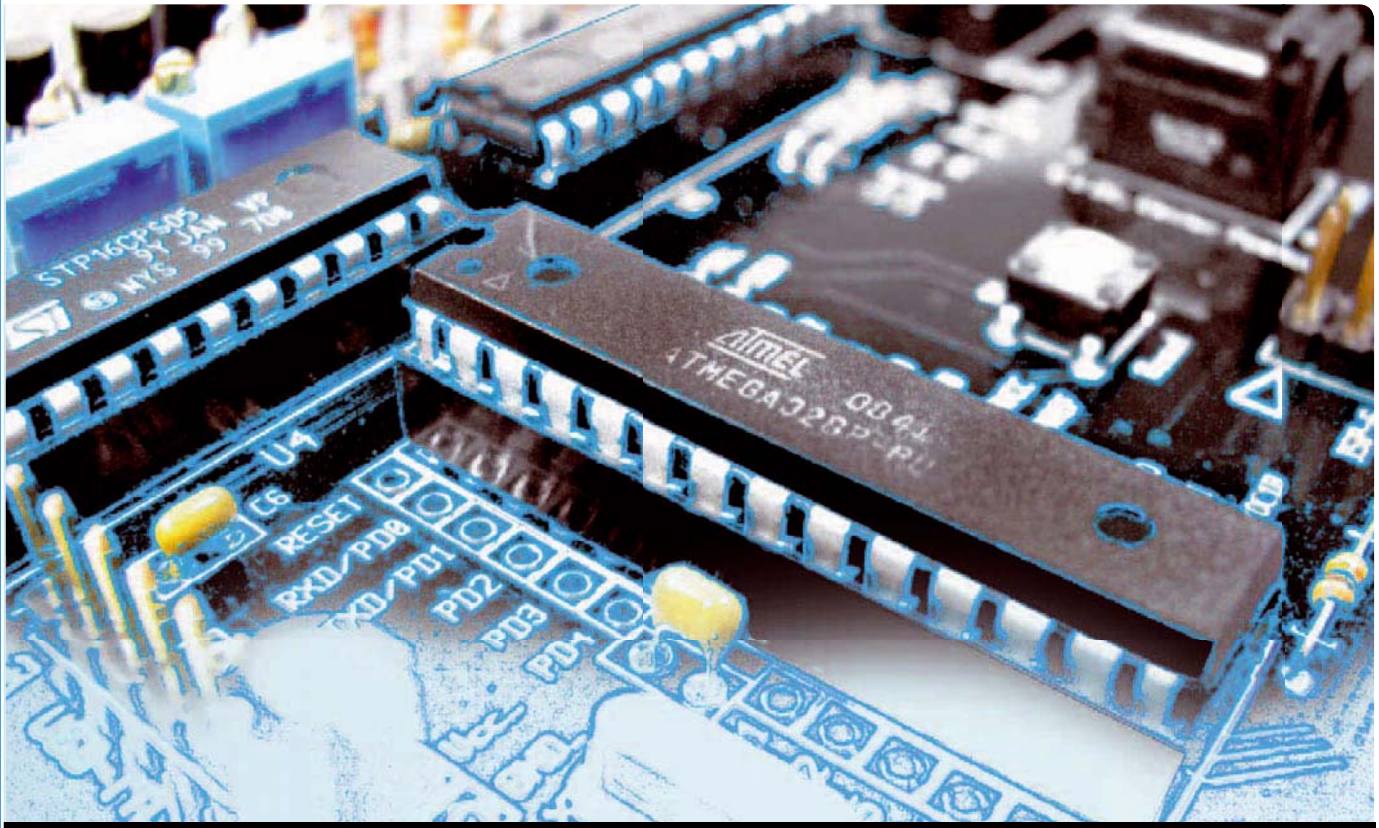
Telecamera IP motorizzata a colori PAN (90°) / TILT (90°)

Sensore d'immagine CMOS 1/4" a colori • Risoluzione 640x480 pixel (VGA) • Ottica fissa • Brandeggio PAN (90°) / TILT (90°) • Illuminatore a 4 LED IR • Formato di compressione M-JPEG • Velocità di trasferimento immagine 30fps (VGA) • Formato video AVI • Alimentazione 5Vdc (adattatore incluso) • Dimensioni 105 x 115 x 100mm.

Prezzi IVA inclusa

FUTURA ELETTRONICA
Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775 • Fax. 0331/792287

Maggiori informazioni su questi prodotti e su tutte le altre apparecchiature sono disponibili sul sito *www.futurashop.it* tramite il quale è anche possibile effettuare acquisti on-line.



ATMEL OPEN SOURCE

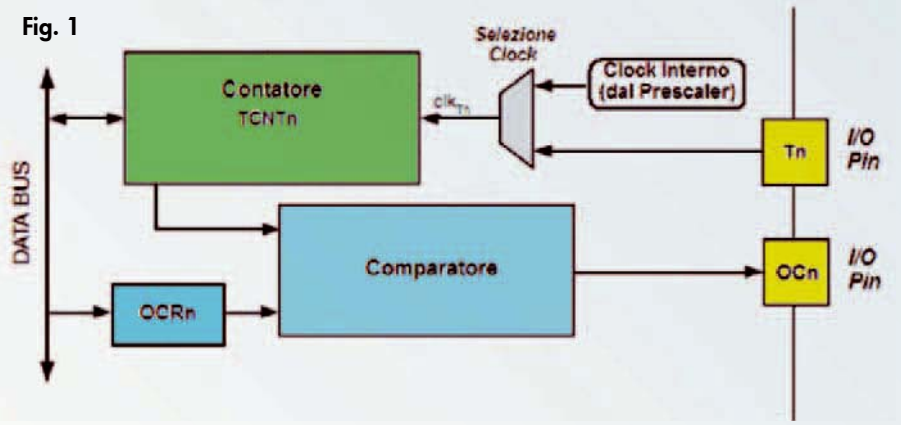
dell'ing. OSVALDO SANDOLETTI

Iniziamo l'esplorazione delle periferiche interne dell'ATMEGA16 analizzando quelle dedicate al conteggio di tempi ed eventi, per capire quali sono i loro compiti e come possano essere utilizzate all'interno di un programma. Sesta puntata.

Iniziamo in questa puntata l'esplorazione delle periferiche interne dell'ATMEGA16 analizzando quelle dedicate al conteggio di tempi ed eventi, per capire quali sono i loro compiti e come possano essere utilizzate all'interno di un programma.
Nella descrizione delle caratteristiche del microcontrollore ATMEGA16, nella prima puntata del corso, avevamo brevemente riportato la lista delle numerose periferiche che esso contiene al suo interno, oltre alla CPU e alle memorie. Tra queste vi sono i ***Timer-Counter*** (abbreviati anche in T/C nel seguito), ovvero circuiti contatori in grado appunto di contare il tempo (come un cronometro) oppure sequenze di '0' e '1' applicate ai loro ingressi. Ad oggi, praticamente ogni microcontrollore sul mercato annovera almeno un timer/counter tra le proprie periferiche; l'ATMEGA16 ne ha 3 distinti (numerati da 0 a 2: T/C0, T/C1, T/C2), e ciascuno può essere programmato per eseguire una delle seguenti funzioni.

- Conteggi di tempo: sull'ingresso del contatore è applicato un segnale di clock (derivato

Fig. 1



da quello della CPU o da un oscillatore) per cui l'uscita del contatore riporta un numero sempre crescente e proporzionale al tempo "che trascorre". Il contatore può anche essere azzerato, od inizializzato a un preciso numero, dal programma. Con questa modalità si può realizzare un cronometro;

- Conteggio di eventi: l'ingresso del contatore è collegato ad un piedino di I/O del micro, per cui l'utente gestisce il conteggio dall'esterno: il contatore incrementa (oppure decrementa) di una unità ad ogni transizione 0 → 1 oppure 1 → 0 applicata al piedino. Con questa modalità si possono realizzare ad esempio dei conta-pezzi;
- Generazione di eventi: l'uscita del contatore (che può essere collegata ad un pin di I/O del micro) può essere configurata per cambiare il suo stato (da '0' a '1' o viceversa) quando il conteggio raggiunge un determinato valore, anch'esso impostabile da programma: con questa modalità si possono realizzare dei temporizzatori.

In **Fig. 1** possiamo vedere lo schema a blocchi semplificato di ciascuno dei T/C, dove al suffisso 'n' delle varie definizioni bisogna sostituire nella pratica il numero 0, 1 oppure 2, a seconda che si consideri rispettivamente il timer/counter 0, 1 oppure 2. Sulla destra della figura vi è il selettore della sorgente di clock del contatore, che può provenire da un piedino di I/O esterno (T*n*) oppure dal clock della CPU (attraverso un "*Prescaler*" tramite il quale possiamo dividere la frequenza di un certo fattore impostabile da programma).
Il segnale così ricavato (detto $\mathbf{clk_{Tn}}$) è utilizzato per pilotare il contatore vero e proprio, chiamato **TCNT*n***. Esso può essere programmato per venire incrementato oppure decrementato ad ogni transizione del segnale $\mathbf{clk_{Tn}}$ ed è collegato al data-bus del microcontrollore: può pertanto in qualsiasi momento essere letto (oppure anche scritto) dal programma. Per esempio, se vogliamo leggere il valore del contatore TCNT0 (ovvero quello relativo al T/C0) e salvarlo nella variabile "Lettura" scriveremo la seguente espressione:

```
Lettura = TCNT0;
```

Oltre alla lettura, nei contatori TCNT*n* possiamo anche scrivervi un qualsiasi valore, dal quale inizierà il conteggio; ad esempio per caricare in TCNT2 il valore zero scriviamo:

```
TCNT2 = 0x00;
```

Se in seguito il contatore TCNT2 riceverà un segnale di clock al suo ingresso, comincerà a contarlo a partire da zero: 1, 2, 3, 4, ...

Nota: nell'ATMEGA16 i contatori TCNT0 e TCNT2 sono da 8 bit mentre TCNT1 è a 16 bit; questo significa che per i primi due il numero massimo raggiungibile è 0xFF (255 in decimale), mentre in TCNT1 è 0xFFFF (65535). Per tutti, il ciclo di clock successivo al raggiungimento del numero massimo causa l'azzeramento del contatore che dunque riparte a contare da 0x00, generando un segnale cosiddetto "*Timer Overflow*" del timer/counter *n* (TOV*n*). Tale segnale è un bit del registro TIFR (*Timer Interrupt Flag Register*, sul quale torneremo più avanti): normalmente è a '0', e se va ad '1' significa che vi è stato un timer overflow; il programma può dunque andare a leggerne lo stato in modo da essere a conoscenza che un contatore "ha sforato" ed agire di conseguenza.
Definizione: i bit che segnalano particolari eventi (come i TOV*n* che segnalano l'overflow dei rispettivi timers) vengono detti flag (bandiera in italiano). Generalmente essi sono posti a '1' per segnalare l'evento, e poi devono essere riportati manualmente a '0' dal programma.
Proseguendo nella descrizione dello schema della **Fig. 1** incontriamo un comparatore, che confronta continuamente il valore di TCNT*n* con il numero contenuto nel registro OCR*n*, che normalmente viene precaricato dal pro-

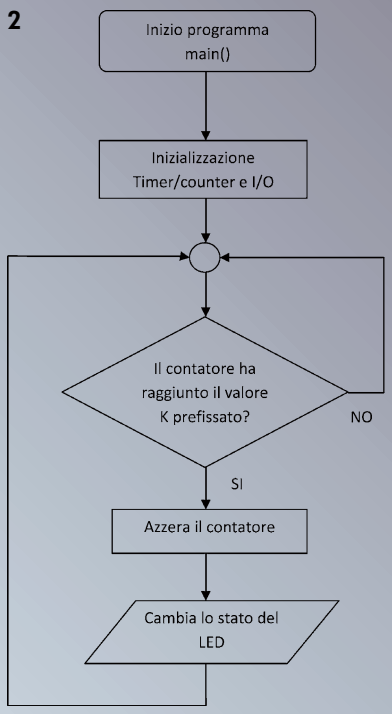
gramma con un valore voluto. Quando TCNT*n* raggiunge (durante il suo conteggio) tale numero, il comparatore "scatta" e segnala l'evento mandando ad '1' il rispettivo bit chiamato OCF*n* (*Output Compare Flag*) pure contenuto nel registro TIFR. Come si nota ancora nella **Fig. 1**, l'uscita del comparatore può anche venire indirizzata su un piedino di I/O (chiamato OC*n*: *Output Compare* del canale *n*) per poter controllare direttamente un circuito esterno al microcontrollore.
Sul datasheet troverete, nelle sezioni relative, gli schemi a blocchi e la descrizione dettagliata di ciascun timer/counter: potrete vedere che, sebbene come principio di funzionamento essi aderiscano tutti allo schema di **Fig. 1**, fra di loro vi sono alcune differenze che possono influenzare il progettista nella scelta di uno piuttosto che un altro. Per evidenziare brevemente queste differenze, riassumiamo la loro descrizione:

- Timer/counter 0: è composto essenzialmente dai blocchi funzionali di **Fig. 1**; il contatore (TCNT0) e il registro del comparatore (OCR0) sono a 8 bit;
- Timer/counter 1: ha due unità indipendenti di "Output Compare", ovvero contiene due registri OCR (chiamati OCR1A e OCR1B), due comparatori e due uscite OC1A e OC1B anch'esse indipendenti tra loro. Possiede anche un'unità di *Input Capture* (ICP) che permette di "fotografare" il contenuto del contatore TCNT1 nel momento in cui un particolare ingresso (collegato ad un pin esterno del microcontrollore) cambia di stato. Il valore catturato in quell'istante viene memorizzato in un registro chiamato ICR1, dal quale il programma può leggerlo. Da ricordare infine che tutti i registri del Timer/counter 1 (TCNT1, OCR1A, OCR1B, ICR1) sono a 16 bit;
- Timer/counter 2: è simile al Timer/counter 0, però l'ingresso di clock esterno è sostituito da un completo modulo oscillatore, cui si può collegare un quarzo esterno dal quale ricavarsi il segnale di temporizzazione indipendentemente dal clock di sistema. I suoi registri sono da 8 bit.

USIAMO I T/C

Per vedere concretamente i timer/counter (T/C) in azione, ne utilizzeremo uno per fare lampeggiare (ancora!) il nostro LED della demoboard.
La scorsa puntata avevamo visto che la soluzione per ottenere un lampeggio "visibile" era l'inserimento di un opportuno ritardo dopo ogni cambiamento di stato del LED; tale ritardo si era ottenuto facendo eseguire al processore, tramite un ciclo '`for`', un elevato numero di istruzioni "inutili".
Adesso vediamo invece come far contare questo ritardo da un T/C, liberando dall'incombenza il processore (che quindi può eseguire altri "incarichi" del programma).

Fig. 2



Per fare ciò utilizzeremo il T/C in modalità di conteggio di tempo: lo inizializziamo per far sì che esso conteggi il clock fornitogli dalla CPU e (sempre tramite programma) leggiamo continuamente il risultato del conteggio: quando viene raggiunto un certo valore 'K' (che poi stabiliremo), il programma cambia lo stato del LED ed azzera il contatore (per fare ripartire il conteggio da 0x00), e il ciclo si ripete.
Chi trovasse complicata la procedura può consolarsi sapendo che esiste una maniera molto più elegante ed intuitiva per descriverla: il *diagramma di flusso* (si veda il riquadro relativo per un approfondimento). In **Fig. 2** viene proposto il diagramma di flusso del nostro nuovo programma, relativo alla procedura sopra descritta (il nostro *algoritmo*).
Prima di iniziare la scrittura del programma (che costituisce l'*implementazione* dell'algoritmo) dobbiamo ancora decidere quale dei tre T/C messi a disposizione dal nostro micro possiamo utilizzare: teoricamente tutti hanno la funzionalità richiesta, però dobbiamo fare due conti per capire quale ritardo massimo possono garantire. Innanzitutto, qual è il segnale di clock che utilizziamo per il conteggio del timer? Siccome nella demoboard non vi sono sorgenti esterne, dobbiamo utilizzare il clock interno "di sistema" del microcontrollore

7	6	5	4	3	2	1	0	
FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0

Fig. 3

	CS02	CS01	CS00	Description
Contatore fermo	0	0	0	No clock source (Timer/Counter stopped).
Clock interno	0	0	1	$clk_{I/O}$/(No prescaling)
	0	1	0	$clk_{I/O}$/8 (From prescaler)
	0	1	1	$clk_{I/O}$/64 (From prescaler)
	1	0	0	$clk_{I/O}$/256 (From prescaler)
	1	0	1	$clk_{I/O}$/1024 (From prescaler)
Clock esterno	1	1	0	External clock source on T0 pin. Clock on falling edge
	1	1	1	External clock source on T0 pin. Clock on rising edge.

che, come avevamo visto nella quarta puntata, può a sua volta essere selezionato da diverse sorgenti: scegliamo quella utilizzata anche nelle esercitazioni passate (quindi non dovremo neanche modificare i fuse-bits nelle impostazioni di avrdude), ovvero il quarzo esterno da 16MHz. Tale frequenza può essere divisa dal *prescaler* del timer/counter per un fattore selezionabile dai tre bit CS*n*<2:0> (ove n è sempre il numero del timer). Con tali bit si sceglie anche quale sorgente di clock utilizzare per il T/C: riportiamo per esempio, in **Fig. 3**, la tabellina della verità dei bit CS0<2:0> (quindi relativi al T/C0); questi bit sono contenuti nel registro TCCR0.
Se questi sono tutti e tre a zero, nessun segnale di clock viene inviato al contatore che resta pertanto fermo. Le combinazioni da 001 (1 decimale) a 101 (5 decimale) permettono di utilizzare il clock di sistema del microcontrollore (qui indicato come "$clk_{I/O}$"), diviso per il fattore mostrato in **Fig. 3**: nel nostro caso, poiché $clk_{I/O}$ = 16MHz, se CS0<2:0> = 001 non vi è divisione e il contatore riceve la frequenza di 16MHz; se CS0<2:0> = 101, il fattore di divisione del prescaler vale 1024 e il contatore incrementerà con una frequenza di 16 MHz / 1024 = 15625 Hz.
Le ultime due combinazioni dei bit CS0<2:0> impostano il timer/counter per prelevare il clock dal piedino esterno T0, e l'incremento del contatore avviene sulla transizione 1→0 (se CS0<2:0> = 110) oppure 0→1 (quando CS0<2:0> = 111) del segnale.
Noi come già detto utilizziamo il clock interno; poiché, come abbiamo visto, la frequenza minima ottenibile è 15625 Hz (usando il rapporto di divisione più elevato), il tempo di ogni impulso di clock che va ad incrementare il contatore sarà pari a 1/15625 = 64µs. Ma allora, se volessimo far lampeggiare il nostro LED a 1Hz, dobbiamo realizzare un ritardo di 500ms sia per la fase 'ON' che per quella 'OFF', e in quell'intervallo di tempo il contatore conterà (scusate il gioco di parole!) 500ms / 64 µs = 7812 impulsi: questo è il valore 'K' con cui il contatore va confrontato nel programma. Questo numero impone anche la scelta sul timer/counter da utilizzare: abbiamo visto che i T/C 0 e T/C 2 hanno contatori (TCNT0 e TCNT2 rispettivamente) da 8 bit, quindi il massimo numero raggiungibile è 255 e quindi non vanno bene: bisogna utilizzare necessariamente il T/C 1 in quanto il suo contatore (TCNT1) è da 16 bit e può contare fino a 65535.
Dal datasheet si evince che anche per questo timer vi sono tre bit di programmazione del prescaler, chiamati CS1<2:0> e contenuti nel registro TCCR1B; anche in questo caso il fattore di divisione massimo è 1024 (ponendo sempre CS1<2:0> = 101) quindi le considerazioni ed i valori calcolati sopra rimangono validi.
Riassumendo:

1) Utilizziamo il T/C 1, alimentato dal clock di sistema (16 MHz) diviso per 1024 (CS1<2:0> = 101): il contatore TCNT1 incrementa pertanto il conteggio ogni 64 microsecondi;
2) Quando TCNT1 raggiunge il numero 7812 significa che sono trascorsi 500 millisecondi, per cui il programma deve cambiare lo stato del LED (da acceso a spento o viceversa) e azzerare il contatore TCNT1;
3) Si aspetta che TCNT1 raggiunga nuovamente il numero 7812, e si torna al punto 2).

INIZIALIZZARE PRIMA DELL'USO

Come (quasi) tutte le periferiche di un microcontrollore, anche i timer/counters richiedono di essere *inizializzati* per divenire operativi, andando ad impostare il valore dei bit di specifici registri di controllo.
Datasheet alla mano, nel paragrafo "Register description" della sezione "16 bit Timer/Counter 1" troviamo la lista e la spiegazione dettagliata di tutti i registri di cui il T/C1 fa uso. I due registri che "contengono" i bit di controllo del T/C1 son TCCR1A e TCCR1B; la modalità di conteggio è stabilita ad esempio dal valore assegnato ai quattro bit WGM1<3:0> (suddivisi tra TCCR1A e TCCR1B) in base alla tabella "Waveform Generation Mode Bit Description"; in **Fig. 4** riportiamo una parte di tale tabella, evidenziando anche la posizione dei bit WGM1<3:0> nei registri TCCR1A e TCCR1B.
Si può notare che ad ogni combinazione di

Listato 1

```
/*********************************************************************
 *   Programma per far lampeggiare il LED LD9 della DEMOBOARD        *
 *    usando il TIMER1 in modalità NORMALE per il conteggio del tempo *
 *********************************************************************/
#include <avr/io.h>

void main(void) {

/* Inizializzazione */
        DDRA = DDRB = 0xFF;  /* Tutti i pin di PORTA e PORTB come uscite */
        DDRC = 0b10111100; /* PC0, PC1 e PC6 sono ingressi */
        DDRD = 0b11110011; /* PD2 e PD3 sono ingressi */
/* impostazione T/C 1 in mod. normale, con prescaler 1024 */
        TCCR1A = 0;
        TCCR1B = (1 << CS12) | (1 << CS10);
        TIMSK = 0;

/* Fine inizializzazione */

        while (1) {   /* ciclo infinito */

                if (TCNT1 == 7812) { /* se il contatore ha raggiunto
7812...*/
                        PORTD = PORTD ^ 0x80; /* complementa PD<7> */
                        TCNT1 = 0;     /* azzera contatore */
                }
        }
}
```

questi bit corrisponde una diversa modalità di conteggio, e ve ne sono ben 16 in totale; siccome noi utilizziamo in quest'applicazione il T/C1 come normale contatore di impulsi a 16 bit dobbiamo impostarlo nel modo '0' (prima riga): infatti la colonna "Timer/counter Mode of operation" riporta la scritta "Normal", e la colonna "TOP" (che dice qual è il massimo numero cui arriva il contatore prima di azzerarsi) riporta 0xFFFF (65535). Sempre dalla tabella si desume che per ottenere il modo '0' dobbiamo impostare WGM1<3:0> = 0000.
Nel registro TCCR1A vi sono altri 6 bit, che però dobbiamo impostare a '0' in quanto servono ad attivare particolari funzioni dell'unità di *Output Compare* che noi non utilizziamo.
Il registro TCCR1B contiene due bit (ICNC1 e ICES1) relativi all'unità di Input Compare: li lasciamo a '0' in quanto anche questa funzionalità non viene utilizzata. TCCR1B contiene infine i tre bit CS1<2:0> di controllo del clock del contatore, che come discusso sopra dobbiamo impostare a 101.
I registri OCR1A, OCR1B, e ICR1 sono relativi alle funzionalità di output compare e input capture del modulo, e quindi non dobbiamo inizializzarli.
Il registro TIMSK serve ad abilitare gli interrupt legati al timer/counter, e vedremo in seguito a cosa serve; per il momento lasciamo disabilitate queste caratteristiche scrivendo '0' nei suoi bit.
Il registro TIFR contiene infine i flag del T/C1, ovvero dei bit che divenendo '1' segnalano il verificarsi di un evento legato alla periferica: il programma può quindi leggerli per sapere se l'evento è accaduto o meno.

SCRITTURA DEL PROGRAMMA

Avendo tutti i dati che ci occorrono passiamo finalmente alla scrittura del programma, riportata nel **Listato 1**. In AVR Studio potete decidere se sovrascrivere il file sorgente "ControlloLED.c" della precedente esercitazione, oppure creare un progetto ex-novo; noi preferiamo risparmiare tempo e sovrascrivere, tanto lo scopo del programma è il medesimo!
In tal modo potrete verificarne il funzionamento con la consueta procedura di compilazione e programmazione. Qui vorremmo soffermarci per commentare alcune righe del programma: nella sezione di inizializzazione dopo aver impostato la direzione dei quattro I/O port del micro (in base a quanto visto nelle puntate precedenti) ed aver forzato a zero il livello dei loro pin impostati come uscite, si passa a caricare i valori prima discussi su TCCR1A (cioè 0) e TCCR1B (cioè zero su tutti i bit tranne CS12 e CS10); per quest'ultimo scriviamo

```
TCCR1B = (1 << CS12) | (1 << CS10);
```

che corrisponde a caricare in TCCR1B il risultato dell'OR logico (che in C si scrive col carattere '|') tra il valore delle due parentesi: (1 << CS12) e (1 << CS10); vediamo allora qual è tale valore.
Durante la compilazione, il preprocessore sostituisce preliminarmente il nome dei bit CS12 e CS10 con la loro rispettiva posizione (2 e 0) nel registro TCCR1B, che è definita (insieme a quelle degli altri registri delle periferiche) nell'header *io.h*; è pertanto equivalente a scrivere (1 << 2) e (1 << 0).
Nel linguaggio C l'espressione "N << X" fa scorrere a sinistra i bit del numero N di X posizioni, quindi se N = 1 (in binario 0000 0001)

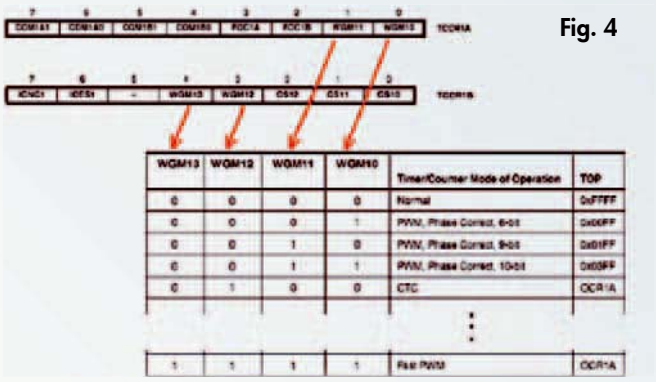


Fig. 4

otteniamo che:
1 << 0 = 1 (0000 0001)
1 << 1 = 2 (0000 0010)
1< < 2 = 4 (0000 0100)
e così via. Quindi i valori ricavati dalle due parentesi tonde sono:

```
(1 << CS10) = 0000 0001 << 0 → 0000 0001
(1 << CS12) = 0000 0001 << 2 → 0000 0100
```

che infine vengono messi in OR tra loro per ottenere il valore 0000 0101 (= 0x05) da caricare in TCCR1B.
Chiaramente avremmo potuto scrivere direttamente

```
TCCR1B = 0x05;
```

oppure, utilizzando la notazione binaria, anche

```
TCCR1B = 0b00000101;
```

ma usando il nome dei bit lasciamo al preprocessore del compilatore il compito di sostituirli con la rispettiva posizione all'interno del registro: il codice rimane più leggibile (scrivendo "`TCCR1B = 0x05;`" come faremmo a capire che sono stati messi a '1' i bit CS0 e CS2?), e a prova di errore!
Tornando al codice dell'inizializzazione, scriviamo 0 in TIMSK per assicurarci che tale sia il suo valore (quindi di tutti i suoi bit), anche se il datasheet afferma che è già lo stato in cui si trova il registro dopo il reset del processore (si veda, nella descrizione del registro, il valore dei bit della riga "Initial value"): riscriverlo *non fa mai male*.
Nel *main-loop* del programma facciamo semplicemente quanto previsto nell'algoritmo e riportato nel diagramma di flusso di **Fig. 2**: TCNT1 è continuamente confrontato col numero K (ovvero 7812), se uguale si aggiorna lo stato del LED e si azzera il contatore. Per complementare il LED usiamo l'operatore EX-OR *bit a bit* del C con la seguente istruzione:

```
PORTD = PORTD ^ 0x80;
```

siccome 0x80 vale in binario 1000 0000, il bit "più a sinistra" (il più significativo) di PORTD viene complementato: se era '0' diventa '1' e viceversa. Poiché tale bit comanda lo stato del piedino PD7 a cui è collegato il LED, otteniamo il lampeggio.

UN'ALTRA POSSIBILITÀ

Abbiamo visto che ogni timer/counter contiene oltre al contatore anche un'unità di comparazione (**Fig. 1**), ed il T/C1 in particolare ne ha addirittura due (canali A e B), possiamo demandare ad essa il compito del confronto del valore di TCNT1 con il numero 7812, anziché andarlo a fare manualmente nel programma. È infatti sufficiente caricare nel registro OCR1A (decidiamo di usare il canale A) il numero 7812 e, ogni volta che TCNT1 raggiunge tale valore, viene automaticamente alzato (va ad '1') il flag OCF1A del registro TIFR. Se inoltre impostiamo il T/C1 per lavorare nella modalità CTC (*Clear Timer on Compare*) che, come si vede dalla tabella di **Fig. 4** si ottiene impostando WGM1<3:0> = 0100 (quindi solo WGM12 a '1') , il contatore TCNT1 si azzera automaticamente quando raggiunge il valore di OCR1A, e pertanto non bisogna più farlo manualmente: il compito del programma nel main-loop rimane quello di aspettare che OCF1A vada a '1' per complementare il LED. Nel **Listato 2** mostriamo dunque il programma completo: si nota che il flag OCF1A del registro TIFR viene letto con l'istruzione:

```
if (TIFR & (1 << OCF1A))
```

ed il "corpo" dell'`if` (tra parentesi graffe) viene eseguito solo se tale flag vale '1'. In tal caso la prima istruzione è il ri-azzeramento di tale flag, necessario in quanto dobbiamo "prepararlo" per il prossimo ciclo: se rimanesse ad '1' non si potrebbe sapere quando il comparatore scatta di nuovo. Per molti flag dell'ATMEGA16 tra cui questo, in base alla descrizione sul datasheet, l'azzeramento si ottiene scrivendovi un '1' (strano, ma è così); nel programma utilizziamo l'operazione OR logico bit a bit del C:

```
TIFR |= (1 << OCF1A);
```

Listato 2

```
/*********************************************************************
 *   Programma per far lampeggiare il LED LD9 della DEMOBOARD        *
 *    usando il TIMER1 in modalità CTC per il conteggio del tempo    *
 *********************************************************************/

#include <avr/io.h>

void main(void) {

/* Inizializzazione */
DDRA = DDRB = 0xFF;  /* Tutti i pin di PORTA e PORTB come uscite */
         DDRC = 0b10111100; /* PC0, PC1 e PC6 sono ingressi */
         DDRD = 0b11110011; /* PD2 e PD3 sono ingressi */

/* impostazione T/C 1 in mod. CTC, con prescaler 1024 */
         TCCR1A = 0x00;
         TCCR1B = (1 << CS12) | (1 << CS10) | (1 << WGM12);
         OCR1A = 7812; /* costante di comparazione */
         TIMSK = 0;

/* Fine inizializzazione */

         while (1) {   /* ciclo infinito */

                  if (TIFR & (1 << OCF1A)) {   /* OCF1A è a 1 ? */
                           TIFR |= (1 << OCF1A);   /* allora riportalo a 0 */
                           PORTD = PORTD ^ 0x80;   /* e complementa PD<7> */
                  }
         }
}
```

Con tale istruzione scriviamo '1' nel bit OCF1A del registro TIFR, lasciando invariati gli altri bit del registro; l'effetto è che OCF1A viene rimesso a zero.

UTILIZZARE LE... INTERRUZIONI

Il compito del *main-loop* del **Listato 2** è quello di verificare ininterrottamente lo stato del flag OCF1A che (avendo caricato OCR1A con 7812) sappiamo andare ad '1' ogni 500ms. Supponiamo che, quando ciò accade, la CPU impieghi quindi 10 microsecondi per azzerare il flag e complementare il pin PD7: allora per i successivi 500.000 – 10 = 490.000 microsecondi la CPU continuerà a controllare (inutilmente) il flag in attesa che torni '1', senza poter eseguire eventuali altre operazioni.

Nota: l'operazione di lettura ciclica dello stato di una periferica (per verificare l'accadere o meno di un evento) si definisce polling.

Pertanto sarebbe molto più efficiente un meccanismo "automatico" che verifichi il cambiamento di stato del flag e lo comunichi alla CPU (che nel frattempo sta eseguendo altri compiti) solo quando ciò accade: in tal caso la CPU interrompe temporaneamente l'esecuzione di quello che stava facendo e va ad eseguire il lavoro richiesto da questo evento, ovvero il complemento dell'uscita PD7; fatto ciò, riprende il compito che aveva interrotto. Questo meccanismo esiste, ed è basato sulle interruzioni (*interrupt*): una interruzione è un segnale (proveniente da una periferica quando accade un particolare evento, oppure dall'esterno) trasmesso alla CPU: quando lo riceve, essa blocca il programma che stava eseguendo in quel momento per andare ad eseguire una particolare funzione di gestione dell'interruzione, detta isr (*Interrupt Service Routine*). Al termine di questa funzione la CPU torna al programma che stava eseguendo nel momento dell'interruzione, riprendendo dal punto esatto in cui era stato interrotto: l'interruzione non ne deve alterare lo svolgimento, anche se ovviamente lo rallenta (la CPU non può eseguire due programmi contemporaneamente: mentre esegue la *isr*, il programma principale è fermo).

Nota: poiché un'interruzione può essere generata da un evento indipendente dal programma principale (può avvenire in un punto qualsiasi durante l'esecuzione dello stesso), diremo che la isr è una funzione asincrona, per differenziarla da una funzione normale (sincrona) che invece viene invocata dal programma principale stesso in un punto preciso.

Esemplifichiamo il concetto delle interruzioni applicandolo al nostro caso: il timer/counter 1 del micro ATMEGA16 permette di inviare segnali di interruzione alla CPU al verificarsi dei seguenti eventi:

- Si è alzato il flag TOV1 (dovuto all'overflow del contatore TCNT1);
- Si è alzato il flag OCF1A oppure il flag OCF1B poiché il relativo comparatore (A o B) è scattato (l'evento è detto pertanto *Output compare match*);
- Si è alzato il flag ICF1 (relativo ad un evento di Input Compare).

Come si vede, ciascuna interruzione che si può inviare alla CPU è legata allo stato di un flag della periferica. In alcuni microcontrollori vi è un solo segnale di interruzione globale che

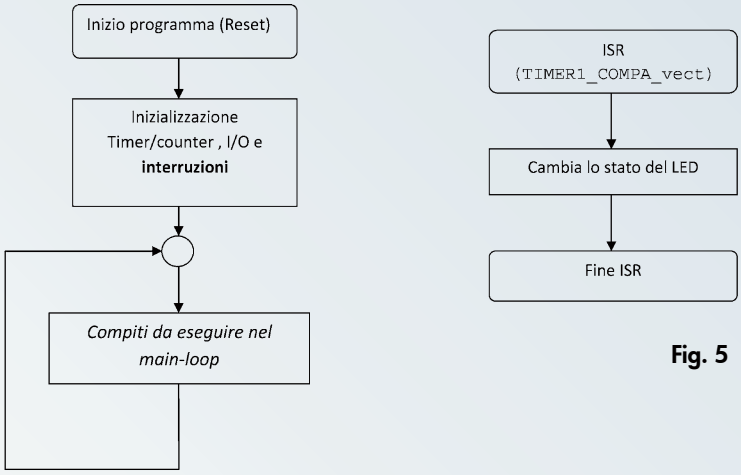


Fig. 5

"lancia" un'unica *isr*; all'interno della quale è compito del programma verificare quale flag è attivo per sapere qual è l'evento verificatosi. Nei micro ATMEGA invece ogni evento chiama una specifica isr; pertanto, a noi interessa implementare la isr relativa all'evento "Output compare match" del canale A, che viene chiamata in concomitanza dell'attivazione del flag OCF1A.
Per far questo dobbiamo scrivere nel sorgente, prima o dopo la funzione main() ma comunque all'esterno di essa, il seguente codice:

```
ISR (TIMER1_COMPA_vect) {
      /* corpo della funzione */
}
```

La parola 'ISR' è una macro, che (semplificando) possiamo paragonare ad una parola chiave del compilatore avr-gcc per informarlo che quella che segue non è una funzione "normale" ma una isr, ovvero una funzione che il processore chiamerà automaticamente al ricevimento di un segnale d'interruzione.
Tra parentesi bisogna inserire il nome preciso della isr che vogliamo invocare; l'ATMEGA16 possiede 20 diverse sorgenti d'interruzione (sommando quelle esterne e quelle legate alle periferiche interne), ognuna delle quali viene collegata ad un proprio indirizzo (detto vettore dell'interruzione) per chiamare una diversa isr. In avr-gcc questi vettori sono associati ad un nome: "`TIMER1_COMPA_vect`" è il nome del vettore associato all'interruzione che ci interessa gestire nel nostro programma (attivata con il flag OCF1A). La lista dei nomi dei vettori d'interruzione del compilatore avr-gcc si trova nel documento "avr-libc-user-manual" contenuta in <cartella installazione WinAVR>\doc, nella sezione "Interrupts". Tra le parentesi graffe della funzione (il suo corpo) si scrive ciò che vogliamo eseguire quando l'interruzione viene attivata. Possiamo adesso stendere il diagramma di flusso del nuovo programma, mostrato in **Fig. 5**.
A sinistra abbiamo il flusso del programma principale, che alla partenza esegue come di consueto tutte le inizializzazioni; inoltre deve *abilitare le interruzioni* che vogliamo vengano attivate, per le quali avremo cura di scrivere una isr di gestione. Per l'ATMEGA16 significa fare due cose:

1) Abilitare l'interruzione legata all'evento "Output compare match" del canale A del T/C 1: si tratta di mettere a '1' il bit OCIE1A (*Output Compare Interrupt Enable 1* del canale A) nel registro TIMSK (Timer/counter Interrupt Mask register). Si dice in gergo che così facendo "smascheriamo" l'interruzione, che in caso contrario non può essere generata;
2) Abilitare la CPU a ricevere le interruzioni attivando un bit di abilitazione "globale"; in *avr-gcc* lo otteniamo scrivendo l'espressione: "`sei();`". Se nel corso del programma volessimo in seguito disabilitare le interruzioni, dobbiamo scrivere "`cli();`". Tenere presente che se nel programma non scriviamo nulla, le interruzioni sono disabilitate per impostazione predefinita.

Tornando al diagramma di flusso, il programma principale si chiude poi nel *main loop* dove può eseguire qualunque compito richiesto (nel nostro caso non deve fare più nulla), sino a che non viene *chiamata* la isr.
Sulla destra vi è il semplice diagramma della isr: l'unico compito è il complemento dell'uscita PD7 per il lampeggio del LED, dopodiché la isr termina (in genere ad eccezione del `main()` tutte le funzioni, e anche le isr, hanno una "fine") e la CPU torna ad eseguire le istruzioni del main-loop.
Il programma completo è quindi mostrato nel **Listato 3**.
Notare che, utilizzando questa logica, il programma è più semplice del precedente: non è più nemmeno necessario dover azzerare manualmente il flag OCF1A in quanto, come riporta il datasheet, viene fatto automatica-

mente alla chiamata del relativo vettore d'interruzione (la isr) .

Listato 3

```
/*********************************************************************
 *   Programma per far lampeggiare il LED LD9 della DEMOBOARD        *
 *    usando il TIMER1 in modalità CTC con le interruzioni           *
 *********************************************************************/

#include <avr/io.h>
#include <avr/interrupt.h> /* necessario includere questo header */

void main(void) {

/* Inizializzazione */

        DDRA = DDRB = 0xFF;  /* Tutti i pin di PORTA e PORTB come uscite */
        DDRC = 0b10111100; /* PC0, PC1 e PC6 sono ingressi */
        DDRD = 0b11110011; /* PD2 e PD3 sono ingressi */

/* impostazione T/C 1 in mod. CTC, con prescaler 1024 */

        TCCR1A = 0x00;
        TCCR1B = (1 << CS12) | (1 << CS10) | (1 << WGM12);
        OCR1A = 7812;
        TIMSK = (1 << OCIE1A); /* abilitazione interrupt Su Output Compare
Match */

        sei(); /* abilita la CPU a ricevere le interruzioni */

/* Fine inizializzazione */

        while (1) {   /* ciclo infinito */
                /* non dobbiamo far nulla! */
        }
}
/* fine programma principale main() */
/* Segue la isr di Output Compare Match del Canale A del Timer 1 */

ISR (TIMER1_COMPA_vect) {
        PORTD = PORTD ^ 0x80; /* complementa PD<7> */
}
```

LA MODULAZIONE PWM

Uno dei sistemi più utilizzati per codificare l'informazione di un segnale digitale è la variazione del tempo in cui resta a livello '1' (detto T_{ON}) rispetto al tempo in cui vale '0' (T_{OFF}), pur mantenendo costante la *somma* dei due (ovvero il periodo T del segnale: $T = T_{ON} + T_{OFF}$); quindi se aumenta T_{ON} deve diminuire T_{OFF} e viceversa. Per quantificare l'informazione codificata con questo sistema non si utilizza la misura del valore di T_{ON} oppure T_{OFF}, ma si ricava il cosiddetto *ciclo utile* (o *duty cycle* in inglese) "D" pari al rapporto percentuale tra il tempo TON e il periodo T: $D = (T_{ON} / T) * 100$; quindi questa grandezza sarà compresa tra 0 (se il segnale è sempre a '0') e 100 (se il segnale è sempre a '1').
In **Fig. 6** potete vedere un esempio di segnale con tre diversi *duty cycle*.
Questa tecnica di codifica dell'informazione (contenuta in D) si dice *modulazione della larghezza dell'impulso* (abbreviato in PWM, dall'inglese *Pulse Width Modulation*). Tra le altre applicazioni, essa è molto sfruttata nel mondo dei microcontrollori per realizzare economici convertitori digitali/analogici: il segnale PWM, generato direttamente da un pin di I/O del micro, può essere filtrato con semplici circuiti per ricavare in uscita una tensione proporzionale a D. Per tale motivo le periferiche timer/counter di molti microcontrollori annoverano la possibilità di generare automaticamente segnali modulati in PWM, ed il nostro ATMEGA16 non fa eccezione: i suoi tre T/C possono essere configurati per tale scopo, ed i segnali PWM risultanti vengono resi disponibili all'utilizzatore sui rispettivi piedini OC_N.

Per vedere un'applicazione pratica, utilizzeremo la modulazione PWM per **variare la luminosità** del 'solito' LED DL9 della demoboard, che finora avevamo fatto solo lampeggiare. Se infatti esso viene alimentato con un segnale PWM, di frequenza abbastanza elevata affinché i nostri occhi non lo vedano lampeggiare, la luminosità che percepiremo sarà proporzionale al *duty cycle* del segnale.
Il LED è collegato sul piedino 21 (PD7) che, guardando sul datasheet al paragrafo "Pin Configurations", è condiviso con il segnale

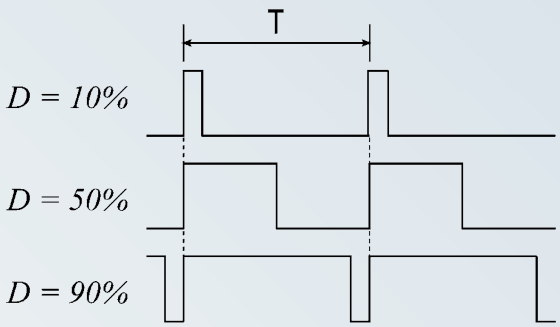


Fig. 6

I diagrammi di flusso

In fase di concepimento di un programma informatico, sia che riguardi grossi calcolatori piuttosto che un microcontrollore, dopo aver capito "bene" quali sono i dati di ingresso a disposizione e che cosa bisogna fare, si delinea il *procedimento* che consenta di arrivare alla soluzione del problema. Tale procedimento viene detto algoritmo e consiste nella descrizione delle operazioni da eseguire, in un preciso ordine, per ottenere il risultato atteso. Per rendere intuitiva questa descrizione, normalmente si rappresenta l'*algoritmo* con uno schema a blocchi grafico in cui le frecce

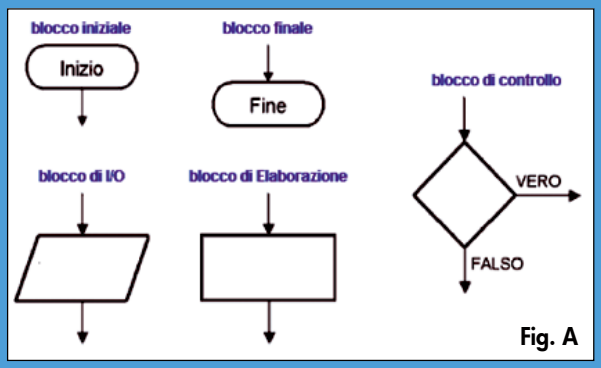


Fig. A

che collegano i vari blocchi danno l'idea della direzione del programma e della sequenza di esecuzione delle varie parti; per tale motivo essi vengono detti *diagrammi di flusso* (*flowcharts*). I blocchi che li compongono possono avere diverse forme, relative al tipo di funzione che devono rappresentare; inoltre normalmente contengono del testo esplicativo dell'operazione che eseguono. Il sito Wikipedia (*http://it.wikipedia.org/wiki/Diagramma_a_blocchi*) descrive bene questi concetti ed i tipi di blocchi che si possono incontrare in questi diagrammi (riportati in **Fig. A**); per la verità ve ne sono anche di altri tipi, ma quelli mostrati sono sufficienti per rappresentare tutte le situazioni (Inizio programma, elaborazione, decisione, input/output, fine programma).
Nel diagramma di flusso di **Fig. 2** ad esempio usiamo già tutti questi tipi di blocchi: notare che il primo è leggermente ovalizzato poiché non rappresenta nessuna operazione ma indica dov'è l'inizio del programma (e non vi è ovviamente la fine, essendo il diagramma della funzione `main()`).
Vi è poi un blocco rettangolare che esegue le operazioni specificate internamente, quindi troviamo un pallino detto "connettore": è semplicemente il punto di unione di più linee (i "flussi" del programma), che affluiscono tutte nel successivo blocco di confronto (rombo). Da lì, se TCNT1 è diverso da K il programma "torna indietro" per fare un nuovo confronto, altrimenti si azzera il contatore e si complementa lo stato del LED (questa è un'operazione di uscita, e dunque va rappresentata con l'apposito simbolo del trapezio, anche se in alcuni casi per semplicità si vede utilizzare sempre il rettangolo).
Siccome la rappresentazione di algoritmi tramite diagrammi di flusso è una pratica molto utilizzata, vi sono numerosi programmi che permettono di realizzarli; alcuni addirittura consentono di estrarre dallo schema il programma vero e proprio, magari in linguaggio C. Chiaramente parliamo di programmi professionali e non gratuiti, se non per periodi limitati di prova. Comunque per realizzare semplici schemi come quelli dei nostri esercizi, vale la pena sapere che anche Microsoft Word mette a disposizione tutti i simboli necessari per il disegno (su Word 2007 vi si accede dal menu Inserisci → Forme → Diagrammi di flusso).

di uscita del timer/counter 2 (OC2). Questo significa che possiamo configurare il T/C2 per generare un segnale PWM su OC2, ed "instradarlo" internamente sul piedino 21 del micro, il quale lo rende disponibile ai circuiti esterni al microcontrollore, nel nostro caso LD9. Si noti anche che questo piedino 21, quando è configurato per riportare il segnale OC2, viene scollegato dal PORTD (pertanto qualsiasi valore che scriveremo successivamente su PD7 tramite programma non verrà più trasferito al piedino).
Questo discorso è valido per tutti i pin di I/O dei micro che sono condivisi con qualche periferica interna: se è configurata opportunamente, essa "prende il controllo" del piedino che non è più gestibile come I/O generico (fino a quando non disabilitiamo questa funzione). Nel datasheet al paragrafo "Alternate Port Functions" del capitolo "I/O Ports" vengono dettagliati questi concetti, oltre a descrivere la "funzione secondaria" che può assumere ciascun piedino. Queste informazioni sono importanti quando, in fase di definizione di un nuovo schema, bisogna decidere i collegamenti e le assegnazioni dei vari pin di I/O del microcontrollore ai circuiti della scheda: nel nostro caso era necessario collegare LD9 proprio al piedino 21, in quanto è l'unico a cui possiamo mandare il segnale di PWM del T/C2. Il T/C2 (ma il discorso è uguale per gli altri 2 timer/counters) prevede due modalità di PWM:

1) "Fast": il contatore TCNT2 incrementa sempre da 0x00 a 0xFF per poi ripartire da 0x00, come nella modalità di conteggio normale; l'uscita OC2 viene alzata quando vi è la transizione 0xFF → 0x00, e riportata a '0' quando TCNT2 raggiunge il valore contenuto in OCR2;
2) "Phase correct": il contatore TCNT2 incrementa da 0x00 a 0xFF e poi decrementa da 0xFF a 0x00; l'uscita è azzerata (0) quando TCNT2 raggiunge OCR2 in "salita" (nella fase di incremento), ed è poi alzata (1) quando TCNT2 raggiunge nuovamente OCR2 in "discesa".

Per i nostri scopi scegliamo di usare la modalità "Fast", e il valore di OCR2 è proporzionale al duty cycle voluto: questo registro è da 8 bit e dunque ha 256 combinazioni possibili

Listato 4

```
/*********************************************************************
 *   Programma per variare la luminosità del LED LD9 della DEMOBOARD  *
 *    usando il TIMER2 in modalità "Fast PWM"                        *
 *********************************************************************/

#define DUTY_CYCLE 200 /* impostare un valore tra 0 e 255 */

#include <avr/io.h>

void main(void) {

/* Inizializzazione */

        DDRA = DDRB = 0xFF;  /* Tutti i pin di PORTA e PORTB come uscite */
        DDRC = 0b10111100; /* PC0, PC1 e PC6 sono ingressi */
        DDRD = 0b11110011; /* PD2 e PD3 sono ingressi */

/* impostazione T/C 2 come fast PWM, no prescaler, uscita su pin OC2 */
        TCCR2 = (1 << CS20) | (1 << WGM21) | (1 << WGM20) | (1 << COM21);
        OCR2 = DUTY_CYCLE; /* carica in OCR2 il valore definito sopra */

/* Fine inizializzazione */

        while (1) {   /* ciclo infinito */

                /* non dobbiamo far nulla! */

        }
}
```

(da 0 a 255): pertanto la risoluzione, intesa come la minima variazione di duty cycle che possiamo ottenere incrementando o decrementando di un'unità OCR2, vale 100/256 = 0,39%: questo valore viene detto granularità del PWM per la nostra periferica. Se in particolari applicazioni ci servisse una maggiore "finezza" di regolazione dovremmo usare il T/C 1 poiché ha registri da 16 bit (65536 combinazioni): la *granularità* risultante sarebbe pertanto del 100/65536 = 0,0015%.
L'impostazione del T/C 2 per questa funzionalità è abbastanza semplice: vi è un unico registro (TCCR2) contenente tutti i bit di configurazione necessari:

- Bit del prescaler CS22, CS21, CS20: li impostiamo per il minimo fattore di divisione (1, cioè nessuna divisione del clock) per ottenere una frequenza tale che il nostro occhio non veda "vibrare" la luce del LED (nel datasheet vi è la formula per ricavare la frequenza esatta del segnale PWM in base al clock in ingresso al T/C e alla divisione del prescaler);
- Bit di modalità funzionamento WGM21,WGM20: li mettiamo entrambi a '1' per configurare il T/C nella modalità "Fast PWM";
- Bit COM21, COM20: essi vanno impostati per consentire al timer di "prendere il controllo" del piedino 21 (PD7), in caso contrario in uscita non arriverà il segnale PWM; le loro combinazioni sono esplicate nella tabellina "Compare Output Mode, Fast PWM Mode" nel paragrafo "Register Description" del T/C 2. Nel nostro caso impostiamo COM2<1:0> = 10; si noti che se COM2<1:0> = 11 il segnale PWM sul piedino viene invertito, e la luminosità diminuirebbe al crescere del valore di OCR2.

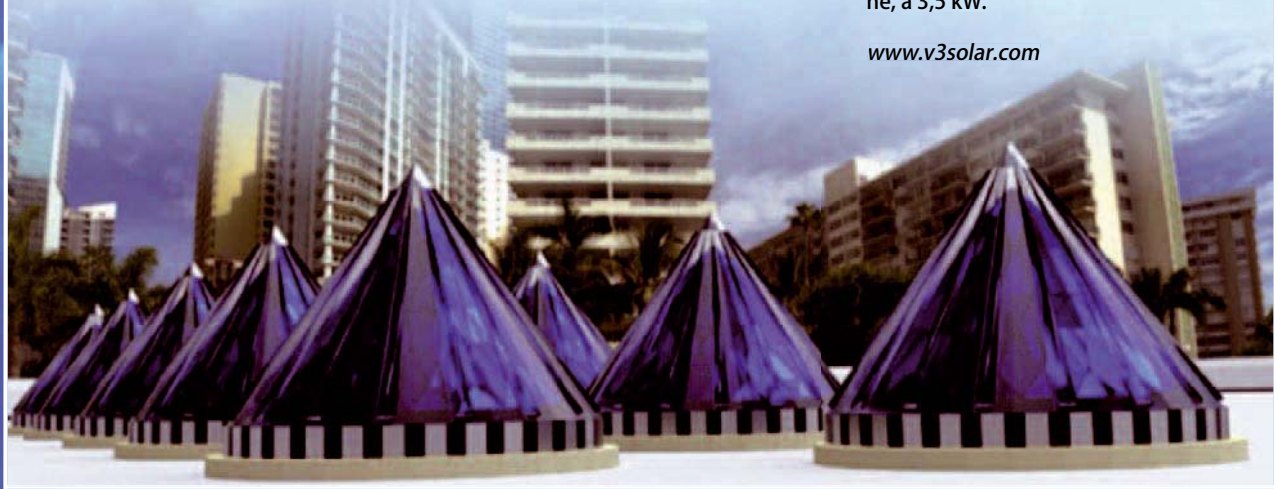
Nel **Listato 4** si può veder il programmino per accendere il LED con una luminosità proporzionale al valore definito per comodità nella costante DUTY_CYCLE: variandolo (e quindi ricompilando e riprogrammando la scheda) potrete vedere come cambia la luminosità. Notare solo che anche mettendo zero il LED non viene completamente spento, poiché la periferica, quando TCNT2 torna a zero, alza sempre il piedino prima di fare la comparazione con OCR2 e abbassarlo di nuovo: questo produce degli "spilli" di tensione che bastano ad accendere, seppur molto brevemente, il LED.
Per rendere più interessante l'esercizio potreste fare incrementare e decrementare in modo automatico la luminosità, caricando OCR2 nel *main loop* non con un valore costante ma con una variabile globale (da 8 bit, dichiarata ad esempio come "`uint8_t MiaVar;`" prima della funzione `main()`), fatta incrementare e/o decrementare ad intervalli di tempo ragionevoli per osservare la variazione; tale variabile potrebbe ad esempio essere incrementata nella isr del T/C1 (nel **Listato 3**, al posto del complemento di PD7). Volete provarci?
Vi diamo un mese di tempo, comunque state tranquilli: nella prossima puntata pubblicheremo la soluzione, oltre a proseguire nell'esplorazione delle funzionalità della demoboard.


I NOSTRI ALIMENTATORI ?
UNIVERSALI
velleman
Scopri la vasta scelta di adattatori da rete
affidabili e ad elevata efficienza,
adatti per alimentare qualsiasi dispositivo
su www.futurashop.it
FUTURA ELETTRONICA
Via Adige, 11 21013 Galiarate (VA)
Tel. 0331/799775 - Fax. 0331/792287

I coni fotovoltaici che piacciono alla California

Uno dei principali limiti all'applicazione della tecnica fotovoltaica è rappresentato dall'estetica dei pannelli, che talvolta abbruttisce realizzazioni rilevanti sul piano architettonico. Per questo, da tempo si cercano soluzioni integrabili e l'industria è alla ricerca di elementi fotovoltaici che abbiano la forma di componenti architettoniche. Un buon esempio sono le volte coniche 3D Spin Cell Generator, moduli fotovoltaici (realizzati dalla californiana V3 Solar) disposti su una struttura a cono capaci di generare 1 kW. Progettati per girare sulla loro base, in maniera da garantire il raffreddamento delle celle e fornire così un 15 per cento di miglioramento delle prestazioni di uscita, questi coni solari impiegano solamente un quarto del silicio, a parità di potenza, rispetto ai pannelli piani. I dispositivi sono dotati di inseguitore solare ad asse singolo che grazie all'angolo di 22° con cui sono stati progettati permette di ottenere gli stessi vantaggi degli inseguitori a doppio asse ma mantenendo i costi contenuti. Inoltre l'unità è in grado di produrre corrente alternata grazie allo statore contenuto alla base e rendere così superfluo l'uso dell'inverter.
V3 Solar sostiene che il 3D Spin Cell Generator permetta di produrre durante la giornata il 40% in più di energia con un guadagno nell'arco dell'anno maggiore del 60%. Il cono fotovoltaico proposto è l'elemento di partenza di una gamma che l'azienda conta di portare, una volta raggiunta la piena produzione, a 3,5 kW.

www.v3solar.com

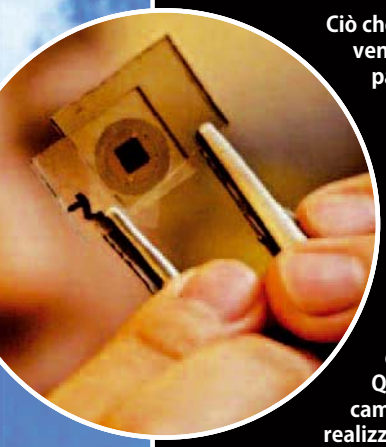


L'eolico fa arredo

Discontinuità a parte, la cosa che più infastidisce dell'eolico è la vista degli aerogeneratori; è per questo che sempre più, almeno nelle città e nelle zone dove ridurre l'impatto visivo è determinante, si punta su piccoli generatori distribuiti ed eventualmente interconnessi. Un buon esempio è Enessere, una serie di piccoli e moderni aerogeneratori che uniscono all'efficienza tecnica un gradevole aspetto estetico. La prima creatura nata dall'impegno dell'omonima società è Hercules, riuscito impasto di tecnologia ed eleganza, già presentato con successo al Fuori Salone del Mobile di Milano; si tratta di un aerogeneratore eolico ad asse verticale il cui rotore è formato da titanio, legno, fibra di carbonio e acciaio, ed è collegato ad un alternatore capace di una potenza installata di 5 kW, montato su un palo alto almeno nove metri. Può soddisfare il consumo medio di elettricità di una famiglia e, in virtù del gradevole impatto visivo (merito del lavoro di un team di artigiani) può essere collocato ovunque, anche vicino a edifici d'epoca e monumenti, senza dare troppo fastidio.

www.enessere.com

La cella di Gratzel diventa solida



Ciò che rende ancora sconveniente l'utilizzo dei pannelli fotovoltaici è un mix di tre fattori: l'elevato costo di produzione delle celle solari, la loro scarsa resa e il degrado, relativamente rapido, delle caratteristiche (perdita di efficienza dell'ordine dell'1% all'anno). Qualcosa potrebbe cambiare con le celle realizzate dalla Northwestern University di Evanston (Illinois – U.S.A.) che sostanzialmente sono una variante migliorata della cella di Grätzel; quest'ultima è un elemento fotovoltaico a basso costo ed ecologico, la cui produzione è per ora ferma ai blocchi di partenza a causa di un difetto: il colorante-sensibilizzate è un liquido organico, che può fuoriuscire e danneggiare la cella solare. Le celle di Grätzel si basano su una tintura organica contenuta tra due pannelli di vetro conduttivi, la quale assorbe la luce solare e la converte in energia elettrica, come la clorofilla fa nelle piante; purtroppo la durata della struttura sifatta non supera i 18 mesi.

Quella realizzata dai ricercatori della Northwestern University è una variante "stabile" della cella Gratzel, basata su un elettrolito solido che è un mix di cesio, stagno e iodio ($CsSnI_3$). In pratica, il nuovo elemento fotovoltaico (che misura 0,5x0,5 cm per circa 10 micron di spessore) utilizza elettrodi esterni in semiconduttore N e P ed una molecola colorante monostrato (quasi sferica, del diametro di 20 nanometri) che serve come giunzione tra i due. Ogni nanoparticella di biossido di titanio, è il semiconduttore di tipo N; il CsSnI3 a film sottile fa da semiconduttore di tipo P.

La cella siffatta ha un'efficienza del 10,2%, simile a quella delle celle di Gratzel ad elettrolito liquido e delle celle a film sottile, ma lontana da quella tipica della tecnica monocristallina (che può arrivare al 17÷18%). L'obbiettivo dei ricercatori è duplice: elevare l'efficienza di conversione e prolungare la vita media fino a portarla ai 20 anni tipici delle celle monocristalline.

www.northwestern.edu/newscenter



In Germania, il solare raggiunge 22 GWh

Nel weekend del 26/27 maggio la produzione fotovoltaica in Germania ha raggiunto il picco istantaneo record di 22.000 MWh: più o meno quello che verrebbe prodotto istantaneamente da 20 centrali nucleari tipo. A darne comunicazione è stato Norbert Allnoch dell'IWR (Internationales Wirtschaftsforum Regenerative Energie) ossia l'ente per le energie rinnovabili in Germania. Il risultato raggiunto significa che le installazioni fotovoltaiche potrebbero coprire circa metà del fabbisogno nazionale tedesco di energia elettrica. Se mai ce ne fosse bisogno, questa notizia dimostra come sia possibile oggi credere nelle rinnovabili e come la Germania abbia da tempo fortemente scommesso sul fotovoltaico, tanto da essere la prima nazione al mondo per produzione fotovoltaica e da avere, sul proprio territorio, una potenza installata che è pari a quella dei restanti paesi del globo. Una simile produzione di energia "verde", oltre che stabilizzare il prezzo dell'elettricità e proteggere maggiormente la Germania dalle fluttuazioni del prezzo dei combustibili fossili, rappresenta un ulteriore passo avanti dal punto di vista della tutela ambientale; non a caso i tedeschi puntano, con l'incremento del fotovoltaico, a una riduzione dei gas serra di circa il 40% entro il 2020: ben oltre il 20-20-20 che impegna le nazioni dell'UE.

www.renewable-energy-industry.com



Solar Impulse vola dall'Europa all'Africa



Finalmente ce l'ha fatta: dopo mesi di fervidi preparativi, Solar Impulse, l'aereo a propulsione elettrica alimentato da 12.000 pannelli fotovoltaici, è atterrato a Rabat, in Marocco, completando un viaggio di 2.500 km svolto in più tappe per consentire ai piloti impegnati nell'impresa di darsi il cambio. Partito da Payerne (Svizzera) il velivolo di Bertrand Piccard ha compiuto un primo scalo a Madrid per cambiare pilota e, una volta ripartito, sorvolando la spagna e lo stretto di Gibilterra ha fatto una trasvolata del Mar Mediterraneo per atterrare in Marocco; questa seconda tappa, della lunghezza di 800 km, ha permesso a Solar Impulse di raggiungere Rabat, meta del suo viaggio.Dopo una sosta di cinque giorni, riprenderà il volo per Ouarzazate, nel sud del Marocco, dove il Re Mohammed VI inaugurerà il più grande impianto solare termico al mondo (ben 160 MW di potenza installata) che sorge laggiù.
Mai momento è stato più atteso, perché la riuscita di questa impresa rende più vicino il giorno in cui gli aerei a propulsione elettrica potranno essere impiegati per i voli internazionali e intercontinentali con passeggeri a bordo; voli che si svolgeranno anche di notte, perché infatti Solar Impulse, pur traendo l'elettricità per i suoi quattro motori elettrici (da 10 CV l'uno) da pannelli fotovoltaici, può volare anche in mancanza della luce del sole, grazie ad un set di batterie che immagazzinano l'elettricità ricavata durante il giorno. Ciò verrà dimostrato dal team di Solar Impulse, che tenterà (si prevede entro il 2014) di compiere il giro del mondo in volo senza soste.

www.solarimpulse.com

Virus piezoelettrici producono elettricità

Che dai batteri e dai dispositivi biologici possa nascere energia, non è una novità; che lo stesso risultato sia ottenibile con i virus, è invece un fatto nuovo. Gli scienziati del Lawrence Berkeley National Laboratory americano sono riusciti a sviluppare un processo basato su virus batteriofagi del tipo M13 che convertono l'energia meccanica in elettricità; la particolarità di tale virus è la capacità di autoassemblarsi naturalmente in sottili film proteici. Aggiungendo al virus, manipolato geneticamente, quattro residui amminoacidici carichi negativamente (ad un'estremità delle proteine elicoidali che rivestono il virus) i ricercatori hanno potuto aumentare la differenza di carica tra i poli positivo e negativo, e di conseguenza anche la tensione generata dal virus.



Da questa struttura, i ricercatori hanno sviluppato un generatore che produce 400 mV di tensione e una corrente sufficiente ad azionare un piccolo display a cristalli liquidi. In questo caso, i virus si sono comportati come un materiale piezoelettrico biologico: disposti su più strati all'interno di un pulsante che misura circa un centimetro quadrato, hanno dato energia al display con la semplice pressione di un dito.

www.berkeley.edu

La Bacheca degli annunci

Vendo:

Da privato "soddisfatti o rimborsati", spedisco pagate alla consegna scontati per urgente realizzo:
- Radiocuffia stereo multifunzione multiaccessoriata Pioneer anche per uso investigativo.
- Microspia americana FM 40,68MHz con auricolare ricevente e certificato di conformità (campionatura).
- Campanello senza fili con funzioni manuali o automatiche (campionatura).
- Proiettore laser a fascio visibile rosso + centralina multieffetti a 3 uscite con portafaretti + lampada di Wood (campionature).
- Accumulatore al gel a tracolla con multiuscite multitensioni ad uso contemporaneo accessoriato tedesco.
- Mini megafono compatto tipo torcia con altoparlante a membrana piatta (campionatura).
- Variatore di velocità professionale fino a 1.320W + variatore universale fino a 1.800W.
- Radiomicrofono con microricevitore a pile.
- Microscopio tascabile tedesco da superficie 100 ingrandimenti con luce per microelettronica, micromeccanica, ecc... (campionatura).
- Vibrografo a battute regolabili + micro incisore multialimentato accessoriato.
- Micro trapano con accessori multialimentato + saldatore ceramico rapido a temperatura regolabile.

Possiedo inoltre utili dispense tecniche collezionabili dal 1900 in poi, che, invecchiandosi, si valorizzano. Contattare Pietro al numero di telefono 0371-30418.

Eseguo:

- Frontalini in policarbonato anche a pezzi unici senza costi di attrezzatura.

Vendo:

- Kit per la stampa in proprio di mascherine in policarbonato composto da 2 fogli formato A4 di speciale policalicarbonato 250 micron stampabile Ink Jet, 2 fogli di biadesivo formato A4, 1 bomboletta di vernice di fondo, istruzioni dettagliate che vi consentono di avere in 10 minuti le vostre mascherine. ideale per progetti.

Contattare Roberto al 338-4317770.

Vendo:

- Saldatore da 50W con regolazione della temperatura;
- Tester digitale (7 campi di misura e 32 portate);
- Trapanino da 12V con alimentatore per forare le basette;
- Terza mano con lente d'ingrandimento;
- Riviste di elettronica.

Il materiale è nuovo, non è mai stato usato. Il tutto a 50 euro. Contattare Francesco al numero di telefono 347-4133862.

Offro:

- Collaborazione anche a progetto o a tempo determinato a ditte del settore elettronico in tutta Italia (sono tra Milano e Salerno).

Realizzo:

- Schede finite, prevalentemente digitali a microcontrollore (Microchip), su specifiche del cliente. Schema, sbroglio del PCB, sviluppo di firmware in Assembler, part list, assemblaggio anche SMD.

Eseguo:

- Riparazioni ed installazioni elettroniche, tecnico per assistenza.

Vendo:

- Duplicatori di EEPROM per serie 24xxx, 93xxx, MDA2061/2 e NVM3060, 27xxx, 28xxx, 29xxx, 49xxx, PIC12C508, PIC16F84, ecc.

Escluso uso hobbistico.

Contattare Vittorio all'email: elettronica.argento@tiscali.it o al 347-3810787.

Sviluppo:

Firmware per microcontrollori PICXX, STXX e Motorola su specifica e realizzo prototipi. Telefonare a Gianni allo 0376-396743.

Vendo:

- Ricetrasmettitore standard C120 (palmare) per i 144MHz con microfono esterno, antenna extra a stilo, contenitore per batterie stilo, manuale in italiano e inglese (originale) perfettamente funzionante. Euro 70,00;
- Ricetrasmettitore standard SR146 per i 144 MHz a quarzi 5 canali più 14 quarzi per ponti radio microfono esterno, antenna extra in gomma, manuale custodia in cuoio, perfettamente funzionante. Apparato adatto anche per i collezionisti. Euro 50,00;

Contattare Vincenzo al 328-9655192.

Cerco:

Nella zona geografica tra Latina nord e Roma sud, laboratorio elettronico o persona specializzata nella installazione elettronica e funzionamento di macchina CNC (pantografo a portale).
Contattare Francesco allo 06-9853114 o inviare e-mail: fracap@iol.it

Cerco:

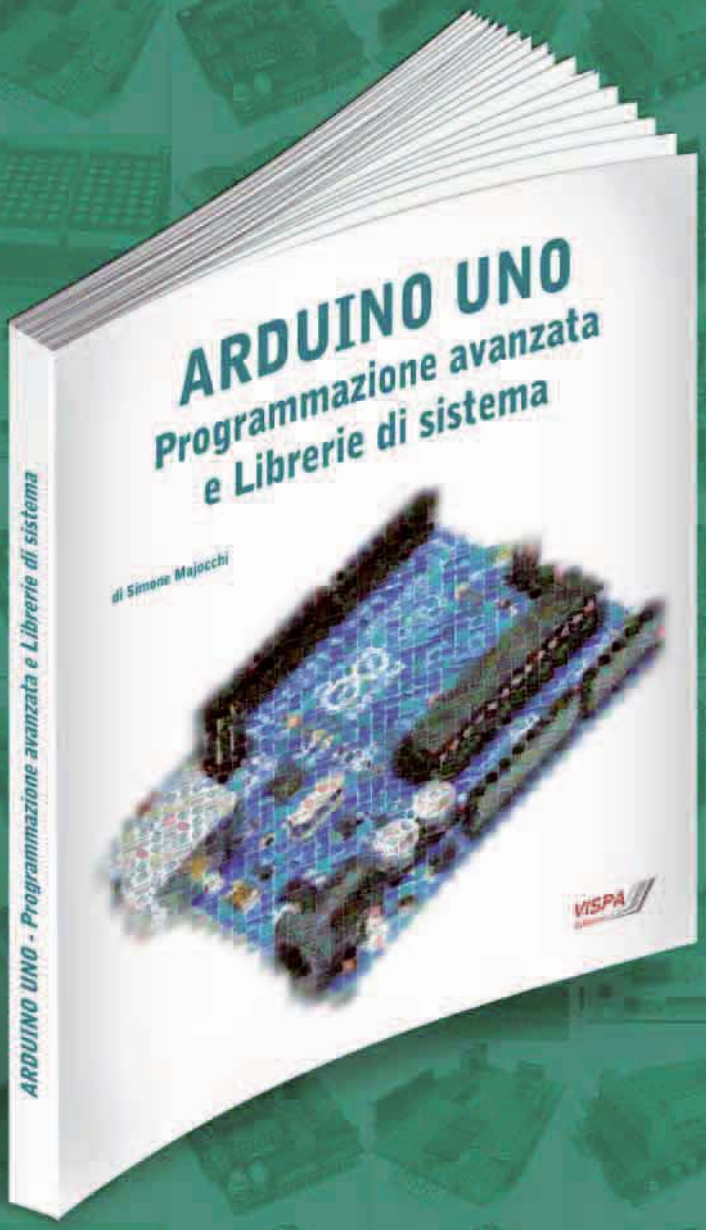
- 4 condensatori elettrolitici a secco per radio a valvole (1930) da 8uF - 500Vp -575Vl;
- scatola di montaggio completa di schemi, mobile e materiale elettronico per radio a valvole supereterodina - 2 gamme d'onda.

Rispondere a Leonardo a leopers@libero.it o al 338-2138943.

Questo spazio è aperto gratuitamente a tutti i lettori. La Direzione non si assume alcuna responsabilità in merito al contenuto degli stessi ed alla data di uscita. Gli annunci vanno inviati via **FAX** *al numero* 0331-792287 *oppure tramite* **INTERNET** *connettendosi al sito* www.elettronicain.it *nella sezione "Annunci".*

"ARDUINO UNO - Programmazione avanzata e Librerie di sistema"

La continuazione del grande successo editoriale del libro *"Primi passi con Arduino"*.



solo € 22,00 IVA inclusa

Gli approfondimenti e le informazioni destinate a chi vuole utilizzare Arduino per progetti evoluti in un libro di facile consultazione e ricco di spunti.

Scritto da Simone Majocchi con l'introduzione di Massimo Banzi. Pagine 224, Euro 22,00.

acquistalo su
www.futurashop.it

FUTURA ELETTRONICA

Via Adige, 11 - 21013 Gallarate (VA)
Tel. 0331/799775 • Fax. 0331/792287

SOLUZIONI PER LA TUA AUTO...

Mantieni sempre in perfetta efficienza il tuo veicolo verificando le condizioni dei principali componenti e sistemi con questi validi ed affidabili strumenti.

TESTER EOBD
Sistema completo, in valigetta

composto da:

- Interfaccia hardware montata
- Software di gestione Unicarscan-Scanmaster
- Cavo di connessione USB

cod. OBDSYSTEM
€ 149,00



... PER LA DIAGNOSI



TACHIMETRO DIGITALE CON PUNTATORE LASER

Permette di misurare, senza contatto, la velocità angolare di alberi motore, ruote dentate e pulegge fino ad un massimo di 99999 rpm, con accuratezza dello 0,05%. Dimensioni 160 x 58 x 39mm.



Prezzi IVA inclusa.

... PER L'ISPEZIONE



cod. DTO6234N
€ 89,00

cod. DVM205AM
€ 61,00

... PER LA MESSA A PUNTO



MULTIMETRO DIGITALE A 4 CIFRE PER AUTOVEICOLI

È in grado di misurare il numero di giri/minuto di un motore, l'angolo Dwell, tensione/corrente alternata e continua, frequenza, resistenza, capacità, diodi e continuità. Completo di batteria, puntali, cavetti con coccodrilli e termocoppia.

SNAKE CAMERA CON MONITOR LCD

Sistema per videoispezione composto da una telecamera CMOS a colori con tubo orientabile da 84,5cm e monitor LCD TFT a colori da 2,36". Angolo di ripresa 50°; illuminatore a LED con intensità regolabile; dimensioni telecamera Ø17x79mm.



cod. CAMCOLI3
€ 120,00

FUTURA ELETTRONICA

www.futurashop.it

Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775 • Fax. 0331/792287

Maggiori informazioni su questi prodotti e acquisti on-line su www.futurashop.it.