Mensile di progettazione elettronica, attualità scientifica, novità tecnologiche.

Elettronica In

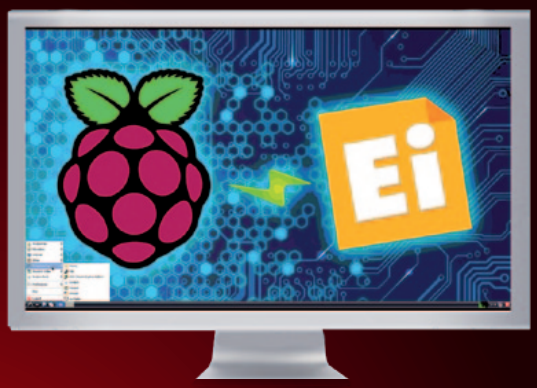
oltre l'elettronica

www.elettronicain.it

Anno XVIII - n. 168
Luglio / Agosto 2012



SPECIALE:
ARDUINO Flash Mob



Tutto il mondo RaspberryPi



Robot Area

- RoboCup Messico: i risultati
- Pancho: il RobotCustom dell'Istituto Vito Volterra

Scheda I/O bluetooth Android based



- Le stampanti 3D approdano nel mondo "fai da te"
- Programmatore HV modulare per MCU Atmel
- Video presentation box
- Minibus, un bus per l'automazione
- Corso Atmel OPEN SOURCE

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

©2010 National Instruments. Tutti i diritti riservati. National Instruments, NI e ni.com sono marchi registrati di National Instruments.
Altri prodotti e nomi aziendali citati sono nomi commerciali delle rispettive aziende. 2010

Elettronica In

www.elettronicain.it
Rivista mensile, anno XVIII n. 168
LUGLIO / AGOSTO 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, PUBBLICITÀ:
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.

Se il futuro non ci aspetta

Un altro anno scolastico è terminato, almeno per quanto riguarda l'attività didattica; le scuole restano aperte per gli esami, un momento di verifica e di riflessione sul cammino percorso e sui risultati ottenuti. Per alcuni una porta che si apre verso il futuro, per altri, sfortunatamente, un gradino troppo alto da superare che costringe a "ritornare al via".
Un momento di verifica, quello degli esami, che siamo abituati ad associare esclusivamente al percorso scolastico: prima quelli delle elementari, poi le medie, le superiori, l'università, il dottorato, l'esame di stato e infine si entra nel mondo del lavoro, che siamo stati abituati a considerare il nostro punto di arrivo, qualcosa su cui contare per tutta la vita fino alla pensione, garantita.



Ma qualcosa sta cambiando, e sempre più rapidamente. Quest'anno non si sono chiuse solo le scuole, ma anche molte attività produttive; si sono persi molti, moltissimi di quei posti di lavoro che fino a poco tempo fa si davano per certi e duraturi, e ci dicono che altre attività chiuderanno, altri posti di lavoro si perderanno. E quanti hanno superato gli esami scolastici faranno sempre più fatica a trovare quel lavoro che sembrava garantito. La scuola a settembre riaprirà, le aziende che hanno chiuso no.
Cosa sta succedendo? Senza farci prendere da visioni pessimistiche, probabilmente quello che sta succedendo è che si sta alzando il gradino per superare gli esami che il mondo globale ci presenta con frequenza sempre più ravvicinata. E non parliamo di esami scolastici. Il famoso lavoro garantito in realtà era comunque un percorso, ma caratterizzato da una evoluzione sociale, economica e tecnologica tanto lenta da farlo durare per una vita intera, fino alla pensione, appunto. Oggi l'evoluzione economica e quella tecnologica sono in continua accelerazione, quella sociale e culturale segna il passo. Il futuro corre, non ci sta aspettando, anche i cicli legati ai posti di lavoro si accorciano sempre più. Che fare? Qual è la soluzione? Probabilmente la stessa di quando arriva l'autobus e noi siamo un po' indietro: mettersi a correre. Essere innovativi, ci dicono. Bene, essere innovativi non significa rincorrere, significa superare, il primo è innovativo, il secondo no. Come fare? Riconoscere i segni dell'evoluzione. Studiare. Cosa? Le lingue, innanzi tutto. Poi le discipline, infine le tecnologie. Riconoscere dove va l'evoluzione, dal disegno manuale, al CAD alla modellazione tridimensionale. Dalla lima, al tornio, alla CNC alle stampe 3D. Dal transistor, agli integrati, ai microcontrollori, ai sistemi embedded. Dove? A scuola, innanzi tutto, e se non basta a casa, di sera, nei week-end e nelle vacanze.
Poi saper fare. Sperimentare, investire nel proprio laboratorietto personale (vale anche e soprattutto per le scuole). In questo mondo che corre anche i prezzi diminuiscono, i prodotti abbondano, ma mancano le idee, le conoscenze e capacità per integrarli e trasformarli in cose utili e produttive. Creare, usare la fantasia, sognare. Fare. Trovare il modo di realizzare in modo concreto ciò che si è imparato. Trasformare i sogni in progetti concreti, che non c'erano prima. Essere il primo. Fare sapere. Comunicare le proprie conoscenze e le proprie realizzazioni, rendersi appetibili, affidabili, concreti, competitivi. Come i campioni di surf, un poco più avanti rispetto all'onda più veloce. Per lasciarci il futuro, e i problemi, alle spalle.
Per ora buone ferie a tutti.
Ci rivediamo a settembre. Noi riapriamo, anzi, proprio non chiudiamo, per non perdere l'allenamento.
Gli esami non finiscono mai.

Arsenio Spadoni
(Arsenio.Spadoni@elettronicain.it)

Sommario

Luglio / Agosto 2012
n. 168

ARTICOLI

23 MODDING
PRESENTATION BOX

Come fare impeccabili presentazioni dal vivo riprendendo oggetti e operazioni manuali ad alta definizione e con una spesa minima.



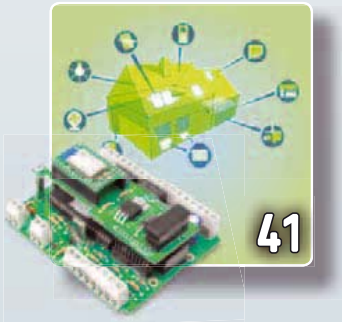
29 DIDATTICA
UN PROGRAMMATORE HV MODULARE PER MCU ATMEL

Dopo avere visto l'organizzazione del firmware per il nostro Programmatore HV approfondiamo la logica del programma e le routine che lo compongono, facendo qualche prova d'uso. Quarta e ultima puntata.



41 NETWORKING
SCHEDA DI I/O BLUETOOTH ANDROID BASED

Permette di comandare con uno smartphone Android 4 uscite a relé e altrettante a transistor, ma anche di leggere lo stato di 4 ingressi digitali.



52 EVENTI
ARDUINO FLASH MOB: È QUI LA FESTA!

In occasione della presentazione del libro "Arduino Uno, Programmazione avanzata e Librerie di sistema", abbiamo organizzato un hacker space: ecco la cronaca ed i progetti con i quali si sono cimentati i partecipanti.





IN COPERTINA:
Finalmente iniziano le consegne della RaspberryPi ed ecco il nostro primo articolo dedicato a questa board: impariamo a caricare il software e, come da prassi, ad accendere un LED collegato al connettore di I/O.

ELENCO INSERZIONISTI

- Fiera di Gonzaga
- Fiera di Novegro
- Futura Elettronica
- ITN
- Lumen Fortronic
- National Instruments
- RM Elettronica
- Robotica
- SMAU
- www.mdsrl.it

Robot Area

91

- **RoboCup MESSICO: COME É ANDATA?**
- **Pancho: il Robot reduce dalla Robocup Jr Italia 2012**



La tiratura di questo numero è stata di 18.500 copie.

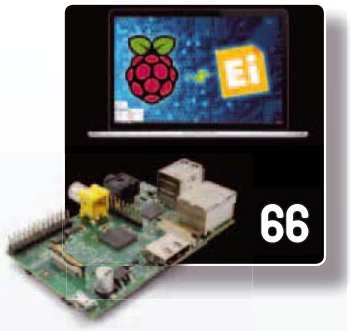
Mensile associato all'USPI,
Unione Stampa Periodica Italiana

RUBRICHE

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

66 DIDATTICA
METTIAMO IN FUNZIONE LA BOARD RASPBERRYPI

Un primo approccio alla scheda RaspberryPi con gli occhi dell'appassionato di elettronica. Oltre a scrivere "Hello World", proviamo ad accendere un LED. E senza scrivere una riga di programma!



79 DIDATTICA
ALLA SCOPERTA DI OPEN PICUS

Finalmente affrontiamo la caratteristica peculiare del mondo openPICUS che permette di realizzare un'applicazione basata su web server utilizzando l'IDE di sviluppo. Ultima Puntata.



83 PROTOTYPING
LE STAMPANTI 3D APPRODANO NEL MONDO "FAI DA TE"

Esistono da decenni, ma solo da circa due anni sono entrate a far parte del mondo degli hobbysti, diventando oggi uno dei temi più caldi del momento.



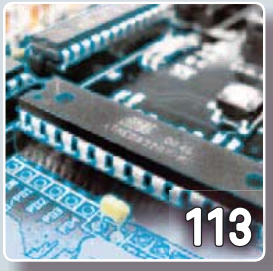
99 DOMOTICA
MINIBUS, UN BUS PER L'AUTOMAZIONE

Proseguiamo la presentazione del sistema flessibile di automazione domestica basato sul protocollo I2C potenziato, con la descrizione degli ultimi moduli periferici e del firmware del modulo Master. Terza puntata



113 DIDATTICA
CORSO ATMEL OPEN SOURCE

Descriviamo il funzionamento di alcune periferiche di input/output presenti nella scheda demoboard: buzzer, LED, tasti e display alfanumerico. Settima puntata.



Affidabilità & precisione... ...nel palmo di una mano!



TERMOMETRO DIGITALE IR / TERMOCOPPIA

Strumento portatile per la misura della temperatura con o senza contatto. Gamma di misurazione da -50 a +300°C (con IR), da -200 a +1300°C (con termocoppia); risoluzione 0,1°C; emissività 0,95; puntatore laser <1mw; dimensioni 121 x 60 x 30mm. Batterie incluse.

€ 82,00 *cod. DVM8090*



LUXMETRO DIGITALE

Strumento portatile per la misura dell'illuminamento ambientale. Unità di misura lux (lm/m²), foot-candle (lm/ft²); gamma di misurazione da 0 a 50000 lux (0÷8300 ftc); risoluzione 1 lux, 0,1 ftc; dimensioni 121 x 60 x 30mm. Batterie incluse.

€ 66,00 *cod. DVM8050*



ANEMOMETRO / TERMOMETRO DIGITALE

€ 88,00 *cod. DVM8020*



TACHIMETRO DIGITALE

Strumento portatile per la misura della velocità angolare di alberi motore, ruote dentate e pulegge. Unità di misura rpm, rps; gamma di misurazione da 100 a 30000rpm, (1,7÷500rps); risoluzione 0,1rpm (0,1rps); dimensioni 121 x 60 x 30mm. Batterie incluse.

€ 59,00 *cod. DVM8030*



FONOMETRO DIGITALE

Strumento portatile per la misura dell'intensità sonora. Gamma di misurazione da 30 a 120dB; curva di pesatura A; risoluzione 0,1dB; gamma di frequenza da 31,5Hz a 8kHz; microfono electret; dimensioni 156 x 60 x 32mm. Batterie incluse.

€ 79,00 *cod. DVM8080*

Strumento portatile per la misura della velocità e temperatura del vento. Differenti unità di misura; gamma di misurazione da 0,4 a 20m/s; da -10°C a +50°C; risoluzione 0,1 m/s, 0,1°C; dimensioni 127 x 60 x 30mm. Batterie incluse.



TERMOIGROMETRO DIGITALE

Strumento portatile per la misura della temperatura (-10 ÷ 50°C) e dell'umidità relativa ambiente (0 ÷ 100%RH) e la determinazione del punto di rugiada (Td). Risoluzione 0,1°C, 0,1%RH; dimensioni 121 x 60 x 30mm. Batterie incluse.

€ 64,00 *cod. DVM8010*

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.

Prezzi IVA inclusa

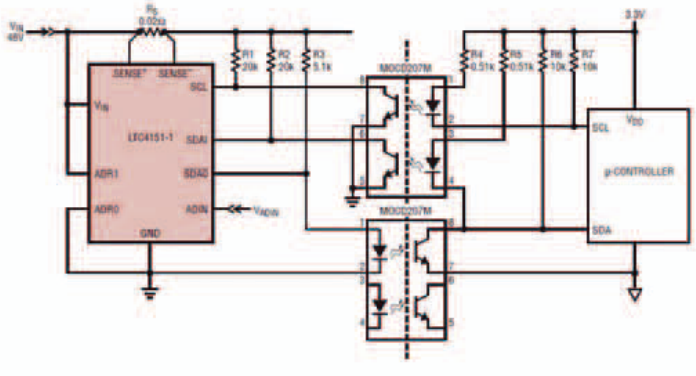
Queste pagine sono dedicate ai suggerimenti, alle richieste ed alle segnalazioni dei lettori. Raccomandiamo, per quanto possibile, di proporre argomenti di interesse generale. Contattateci numerosi all'indirizzo: *redazione@elettronicain.it*

Current Sensing con I²C optoisolato

Vorrei monitorare l'assorbimento di corrente in un circuito in corrente continua (nello specifico, l'alimentazione principale presa dalla batteria di un radiomodello) restando galvanicamente isolato per ragioni legate ai disturbi; la mia idea sarebbe verificare la corrente con un microcontrollore, lo stesso che uso per governare il modello, o meglio ancora con Arduino, quindi mi servirebbe un convertitore da corrente a dati TTL o qualcosa che fornisca informazioni in formato digitale su bus I²C o SPI. Avete qualcosa da suggerirmi? Magari un misuratore di corrente (mi serve misurare circa 30 A) e possibilmente galvanicamente isolato in modo che il micro non venga disturbato dai picchi di assorbimento dei motori?

Ottavio Serra - Como

Nel tuo caso, un'ottima soluzione viene fornita da Linear Technology, che propone il suo LTC4151: si tratta di un integrato per power-sensing, che mediante un resistore shunt rileva la corrente assorbita da un utilizzatore e la converte in tempo reale, tramite un precisissimo A/D converter con risoluzione di 12 bit, in byte che rende poi disponibili su un I²C-Bus. L'integrato può lavorare con un ampio range di tensione di alimentazione (da 7 a 80 V) che preleva dalla fonte dalla quale misurare l'erogazione di corrente (nel tuo caso, la batteria dell'aeromodello) ed ha la particolarità di disporre di linee I²C sdoppiate, proprio in previsione del disaccoppiamento ottico; in altre parole, l'ingresso SCL è unico mentre per l'SDA, che è una linea dati bidirezionale, ci sono un ingresso ed un'uscita. In caso contrario, sarebbe impossibile accoppiare otticamente l'SDA.
Il circuito che ti consigliamo è tratto da una nota applicativa della Linear Technology e impiega, per accoppiare le linee SCL, SDA input e SDA output, degli optoisolatori di produzione Fairchild, ognuno dei quali contiene due canali optoisolati formati da un LED e un fototransistor.
Questi componenti, siglati MOCD207M, hanno peraltro la prerogativa di presentare una curva di CTR (Current Transfer Ratio), ossia della funzione di trasferimento, abbastanza lineare tra 0 e 2 mA di corrente di collettore, il che in questa applicazione è poco rilevante, ma può diventare interessante in altri impieghi.
Per dimensionare la resistenza di Sense (ossia quella collegata tra i piedini SENSE+ e SENSE-) dell'LTC4151, considera che il fondo scala dell'A/D converter all'interno dell'integrato corrisponde ad una differenza di potenziale di 81,92 mV tra SENSE+ e SENSE-; il resto devi farlo con il software, nel senso che devi poi convertire il valore binario fornito dall'ADC in tensione, ragionando in proporzione. Tieni conto che per avere un fondo scala di 30 A con 81,92 mV, serve una resistenza molto piccola, da 2,73 mohm, ossia 0,00273 ohm, che puoi realizzare con uno spezzone di filo in rame, sapendo che il rame ha una resistività di 16 mohm x mm²/m; in breve, uno spezzone di filo di sezione pari a 2 mm, lungo 8,5 cm e ripiegato ad U.



Circuito misuratore di corrente I²C-Bus basato sull'integrato LTC4151, dotato di blocco di accoppiamento ottico al microcontrollore.

ArdupilotMega si fa in due

Mi piacerebbe sperimentare l'ArdupiloMega che avete presentato nei numeri scorsi della rivista, però mi spaventa un po' la necessità di saldare e configurare le varie schede che compongono il sistema finale. Ma soprattutto mi scoraggia la procedura da seguire per la acquisizione, la compilazione e l'upload del firmware sulla scheda. Esiste la possibilità di avere una versione già configurata e pronta all'uso?

Lorenzo Milici - Pordenone

La tua richiesta ci permette di aggiornare la situazione relativamente al progetto ArdupilotMega che dalla data di pubblicazione degli articoli è stato oggetto di numerosi aggiornamenti ed evoluzioni sotto molti aspetti.
A questo riguardo promettiamo di pubblicare quanto prima sulla rivista un articolo che aggiorni il quadro di riferimento del progetto. Per ora ti anticipiamo le novità più importanti. Dal punto di vista delle schede hardware la ... vecchia configurazione di ArdupilotMega, basata su diverse schede assemblate tra loro a pacchetto, è stata sostituita dall'ArdupilotMega2, una nuova configurazione compatta, dalle dimensioni molto più piccole della prima e che viene fornita già assemblata, superando la necessità di saldature (vedi **Fig. 1**).
E anche il prezzo si è ridotto notevolmente. Veniamo ora alla configurazione del firmware, che è stata semplificata di molto.
Pur rimanendo la possibilità

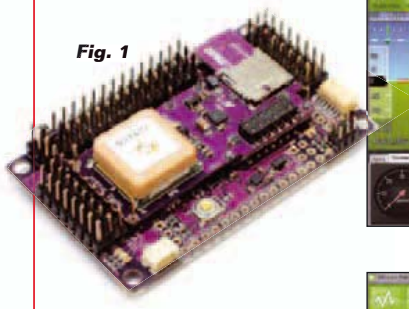


Fig. 1



Fig. 2

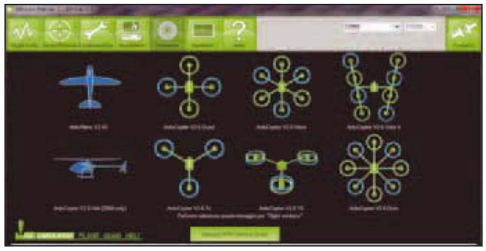


Fig. 3

Qui sopra, la nuova scheda ArdupilotMega 2; accanto, due videate del software Mission Planner per la telemetria e la configurazione dei radiomodelli.

di compilazione per permettere agli appassionati di modificare e personalizzare il software, per l'utilizzo standard tutte le operazioni vengono eseguite dalla MissionPlanner (vedi **Fig. 2**): il download, l'installazione sulla scheda e tutte le configurazioni necessarie sono disponibili con comodi pannelli dall'interfaccia grafica. In **Fig. 3** ti anticipiamo la finestra di download del firmware, che consente di scegliere per quale tipo di velivolo vogliamo configurare la scheda.
Una volta installato il firmware sull'ArdupilotMega2 si può iniziare la calibrazione premendo il pulsante "Settaggi APM". Per farti una idea sulla nuova MissionPlanner, in attesa dell'articolo, la puoi scaricare liberamente all'indirizzo *http://code.google.com/p/ardupilot-mega/downloads/list*.
La estrai direttamente, rispettando le cartelle impostate nello zip, sul disco C del computer.
Per eseguirla basta lanciare il programma MissionPlanner.exe.

Quando il sensore rileva la pioggia

Ho notato che l'impianto di irrigazione del mio condominio si accende ad orari prestabiliti ma lo fa anche se è piovuto da poco, con evidente dispendio di energia elettrica ed acqua; probabilmente ciò dipende dal fatto che nella centralina c'è un semplice timer e manca un sensore in grado di rilevare le condizioni del tempo o più semplicemente l'eventuale presenza di acqua al suolo. È possibile modificare l'impianto aggiungendo un sensore di umidità o qualcosa di simile? Costa molto? E come rispondono gli attuali sensori in caso di gelo?

Saverio Donadio - Vicenza

Parti dalla considerazione che, di norma, un buon impianto di irrigazione automatica deve avere un sensore di pioggia e magari uno di temperatura o luce ambiente: il primo serve ad evitare che venga aperta l'acqua e quindi annaffiato il prato quando sta piovendo (allorché è inutile e deleterio) mentre il secondo evita che l'irrigazione parta quando c'è pieno sole, cosa poco utile perché molta acqua evapora, e dannosa in quanto bisogna annaffiare erba e piante quando non c'è pieno sole, altrimenti si fanno dei danni.
Il sensore di pioggia è un accessorio molto semplice e, nella sua forma elementare, molto a buon mercato; puoi anche farlo in casa accostando ad un paio di millimetri di distanza due lamine o elettrodi metallici, in modo che in presenza d'acqua si passi dal circuito aperto alla conduzione. Collegando questo sensore tra l'alimentazione positiva e la base di un transistor Darlington, il collettore può attivare l'ingresso "pioggia" della centralina di irrigazione, ovvero alimentare un relé il cui contatto, posto in serie all'elettrovalvola dell'acqua, blocchi l'erogazione se è appena piovuto. Perché la cosa funzioni, questo sensore deve stare in un posto tale da non essere a sua volta "annaffiato" quando l'irrigazione entra in funzione, altrimenti blocca l'acqua. Un buon sensore può essere quello di tipo a dischi igroscopici, basato su un materiale idroassorbente: il vantaggio di questa soluzione rispetto ai sensori a lamine è che in esso l'acqua non può mai accumularsi né gelare col freddo. Un buon sensore da pioggia di questo tipo è prodotto dalla Irritrol (*www.irritrol.it*).



SD-Card: quale tipo devo scegliere?

Posseggo un vecchio lettore per SD-Card che usavo per scaricare dei dati dalle SD montate in un data-logger e nella macchina fotografica digitale, solo che di recente ho acquistato una card da 4 GB e sto trovando qualche difficoltà, nel senso che non riesco a leggerne il contenuto. C'è per caso un problema di compatibilità tra SD?

Tommaso Gervasi - Aosta

Le SD o Secure Digital che dir si voglia, non sono tutte uguali, almeno se guardiamo cosa c'è dentro: infatti, malgrado formato fisico e connessioni siano rimaste sempre le stesse, qualcosa è cambiato per quel che riguarda la memoria e l'interfaccia.
Le prime SD-Card, dette semplicemente SD, non sono compatibili con il tipo più diffuso attualmente, che è l'SD HC: in altre parole, il lettore SD comune non legge le HC. Lo stesso vale per fotocamere ed apparati che utilizzano le SD-Card come supporto di memoria (ad esempio i navigatori GPS).
I tipi esistenti di SD-Card sono SD, SDHC, SDXC, SDHC UHS-I. Le SD originarie hanno una capacità di immagazzinamento dei dati che non supera i 2GB; questo limite è stato superato dalle SD HC, che non a caso si chiamano High Capacity e che vanno da 2 ad un massimo di 32 GB. Le SD HC si riconoscono sia perché dispongono sempre di almeno 2 GB, sia perché riportano il simbolo HC (in commercio sono anche apparse SD da 4 GB non marchiate HC: queste sono fuori standard) ed un bollino con un numero indicante la velocità di scrittura continua garantita dei

dati, che può essere 2 (2 MBps) 4 (4 MBps) 6 (6 MBps) e 10 (10 MBps). Nella marchiatura delle SD HC si usa anche un fattore di moltiplicazione tipo 100x, 133x o 200x, dove x corrisponde a 0,15 MBps, ovvero direttamente la velocità di trasferimento in MB/S.
Dopo le HC, sono nate le SDXC, abbreviazione di Secure Digital eXtended Capacity, che hanno capacità comprese tra 32 GB e 2 TB. Oltre ad esse, esistono le SDHC UHS-I, capaci di arrivare a velocità teoriche di 312 Mbyte al secondo.
Riguardo alla compatibilità, tieni presente che i lettori e i sistemi possono gestire, oltre alle SD-Card con cui vengono forniti, anche quelle di tipo precedente, ma non possono riconoscere e usare i modelli successivi; perciò un lettore per SD HC legge e scrive sia le SD HC che le normali SD, ma non funzionano con le SD XC o con le SDHC UHS-I. Dispositivi progettati per funzionare con le SDXC, invece, leggono tranquillamente le SD e le SD HC e le SD HC UHS-I.
Quanto alla velocità, tieni presente che normalmente la classe di appartenenza può fare fede più del fattore di moltiplicazione scritto sulla SD, in quanto quest'ultimo o la velocità riportata è quella dichiarata dal costruttore senza specificare le condizioni del test o senza che sia stata verificata da alcun organismo di verifica; quindi in pratica la velocità reale può essere ben diversa. Se invece una SD viene collocata in una certa classe, deve rispondere ai parametri che la contraddistinguono. Fai anche attenzione che alcuni costruttori scrivono "up to" accanto al valore della velocità, in modo da tutelarsi quando non sono certi che tutti i campioni prodotti rispettino quella specifica: in questo caso è quasi certo che si tratta di SD di scarsa qualità.
Se vuoi maggiori chiarimenti puoi visitare il sito della SD Association, *www.sdcard.org*, che raggruppa i principali costruttori mondiali di SD-Card.

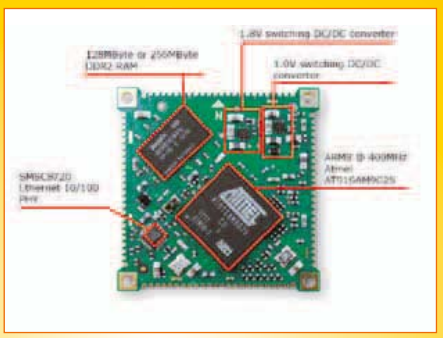
RaspberryPi ha già una concorrente?

Ho letto su questa stessa rubrica nel numero 165 le istruzioni per emulare sul PC il software della RasperryPi. L'argomento mi incuriosisce ma mi scoraggiano un po' la disponibilità ed i tempi di consegna della scheda. Mi chiedo se non esistano alternative alla scheda RaspberryPi per sperimentare con Linux senza dover affrontare la spesa di un PC dedicato...
Gennaro Amendola - Roma

In effetti la RaspberryPi ha rappresentato un punto di discontinuità tra alcuni filoni tecnologici classici, probabilmente accelerando, almeno da un punto di vista culturale, quella che sembra essere una tendenza ormai abbastanza delineata. Come puoi leggere nell'articolo dedicato in questo numero alla RaspberryPi, la divisione abbastanza netta tra elettronica ed informatica si sta affievolendo. I prodotti per elettronici, come i microcontrollori si stanno evolvendo incorporando funzionalità culturalmente informatiche come i server web presenti nei vari OpenPicus, Tibbo e anche Arduino, mentre dal mondo dei PC si sta distaccando un filone di micro PC, come appunto la RaspberryPI, ma che portano a bordo funzionalità di I/O digitale proprie dei microcontrollori. In mezzo, quelli che prima erano sistemi embedded veri e propri, si stanno miniaturizzando e i loro prezzi diminuiscono di molto come la scheda Aria, recensita in questa stessa rubrica e di prossima disponibilità. Il fattore comune è la presenza su tutti di un vero sistema operativo, che di fatto fa avvicinare e in parte sovrapporre le conoscenze e

ARIA NUOVA NEL MONDO EMBEDDED





Dopo l'annuncio fatto in primavera da ACME Systems, ecco finalmente in dirittura d'arrivo ARIA, una piattaforma di sviluppo low-cost basata su Linux Embedded, che utilizza come processore un AT91SAM9G25 ad architettura ARM9 a 400 MHz. Questa board, disponibile da agosto, nasce da sette anni di ricerca che hanno prodotto già ottime board come Daisy (ideale per la gestione di display, relé, servo, RS232, RS485/422, CAN Bus, GSM) e Bee Board (compatibile Arduino) e integrabile con la Daisy. A bordo di ARIA, una RAM di 128 o 256MB DDR2, un'interfaccia Ethernet 10/100 Mbps, 3 porte USB 2.0 High/Full speed, I²C-Bus, SPI ed un lettore SD da cui si può fare il boot. Tutto questo in appena 40x40 mm. Tutte le connessioni, compresi i ben 60 GPIO (I/O General Purpose), sono accessibili da un connettore miniaturizzato a passo 1,27 mm e da una porta JTAG. Per l'alimentazione bastano appena 3,3 volt (il consumo è appena 0,3 W); a bordo, la scheda dispone di due regolatori switching capaci di ricavare 1 e 1,8 V per l'alimentazione del core e della RAM DDR2.
ARIA sarà disponibile in due versioni: quella con 128 MB costerà 24 euro mentre quella equipaggiata con 256 MB costerà 29 euro. Più avanti ACME rilascerà una variante basata sul modello X25, che dovrebbe disporre di due porte Ethernet e di 2 CAN.
Si tratta di un sistema per cui si prevede un grande successo e una diffusione massiccia, almeno quanto è stato per la scheda FOX Board G20 basata anch'essa su Linux Embedded e di cui sono stati venduti circa 40.000 esemplari in oltre 50 Paesi nel mondo; ciò perché ARIA mette a disposizione di una fascia molto più vasta di sviluppatori uno strumento dalle potenzialità enormemente superiori rispetto a quelle delle board oggi presenti sul mercato in questa categoria di prezzo: per intenderci Arduino e simili.
Il filmato di presentazione è disponibile al link *http://goo.gl/M40av*



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

le professionalità legate alle discipline elettroniche e informatiche. Nel tempo gli informatici dovranno capire come integrare e far comunicare i loro sistemi con le sezioni digitali, analogiche e di comunicazione dei nuovi dispositivi, dall'altra ci sono gli elettronici, che dovranno aprire le loro applicazioni al mondo esterno, predisponendole a integrarsi e comunicare con sistemi di controllo remoti fissi o mobili, come i tablet e gli smartphone. I sistemi operativi che si stanno diffondendo sui micropc e sui sistemi embedded sono varie versioni di Linux, Android e WindowsCE.
Qui proviamo a rispondere alla tua





domanda: per soddisfare curiosità e sperimentare senza patemi di combinare pasticci o dedicare un PC, allo stato attuale puoi prendere in considerazione il microPC MK802, in formato poco più che pen drive, accetta sia il sistema operativo Android 4 che Linux (per ora Ubuntu). Il sistema operativo sta su microSD, così con due schedine puoi sperimentare contemporaneamente su entrambi i sistemi. Possiede un processore Allwinner A10 Cortex-A8 da 1GHz di clock, 1GB di RAM, 4GB di memoria Flash, porta USB, video HDMI 1080P e 2160P, audio, WiFi integrato, tastiera e mouse via USB oppure wireless integrato come la microtastiera nella foto, alimentazione a 5 V micro USB come i telefonini. Sia con sistema operativo Android 4 che con Linux può essere configurata con un server DroidMote per essere gestita wireless con uno smartphone Android. Chiaramente manca di connettori GPIO, ma si può sperimentare ugualmente l'integrazione con sistemi come per esempio Arduino o OpenPICUS utilizzando comunicazioni di tipo Web Service (o API), che ti permetteranno di approfondire il tema dell'Internet nelle cose, argomento anche questo al quale pensiamo di dedicare presto qualche articolo.

Se l'APN non è quello esatto

Sto utilizzando un cellulare UMTS/HSDPA come modem per connettermi ad Internet mediante il computer portatile e di recente ne ho aggiornato il firmware; il problema è che i parametri di configurazione che avevo inserito sono stati resettati e perciò ho perso le impostazioni per la connessione. Guardando nel menu ho visto che nei profili web bisogna inserire parametri come l'indirizzo IP, l'APN ecc. In particolare, nel manuale ho letto che l'IP ed il proxy possono non essere necessari (dipende essenzialmente dal gestore) mentre l'APN è indispensabile; il fatto è che il mio gestore ha più APN e in particolare, uno per le connessioni WAP ed uno per quelle web. Sapete dirmi quale devo usare, perché le connessioni non mi riescono o quando mi riescono costano tantissimo? Devo abbandonare il mio cellulare e passare a una chiavetta UMTS?

Luciano Torrisi - Gorizia

Per riuscire ad affacciarsi sul web, quindi connettersi ad Internet, il cellulare deve sempre essere impostato (le impostazioni dipendono dal telefono e sono spiegate nel manuale d'uso fornito dal produttore)



Inserimento del comando AT addizionale per gestire il modem GPRS/UMTS; la finestra di dialogo è di Windows XP e vi si accede da Pannello di controllo>Opzioni Modem e telefono. In Windows 7 è simile e vi si accede da Pannello di controllo>Telefono e Modem.

con almeno l'APN, che è l'Access Point Name; questo rappresenta per il telefono il punto dal quale connettersi sul web. Normalmente i gestori inviano al cellulare degli SMS di configurazione automatica, però talvolta bisogna procedere a mano. La prima cosa da fare è richiedere ai gestori i nomi degli APN, quindi inserirli nei profili web; i nomi che iniziano con WAP servono per l'accesso ad Internet dal cellulare, mentre quelli che iniziano con web o internet vengono usati quando il telefono fa da modem a un computer. Quindi è importante che verifichi prima di tutto che tipo di connessione e opzione Internet hai sottoscritto col gestore, perché se hai un piano WAP (ad esempio Super Internet di TRE o 7 giorni di PosteMobile) devi usare un APN di tipo WAP; invece se navighi da PC, devi inserire un APN di tipo internet o web. Teoricamente nel cellulare possono essere memorizzati vari APN anche di gestori differenti senza che si verifichino problemi: il telefono punta a quello corrispondente al gestore della SIM; nella pratica può capitare che se il gestore si appoggia in roaming su altri, il cellulare punti su APN differenti e quindi la navigazione avvenga in roaming, con le conseguenze del caso (costi molto elevati malgrado si siano sottoscritti piani a costo fisso). Può anche accadere che, malgrado si sia sottoscritta un'opzione WAP, il cellulare si colleghi in web perché nei profili è presente l'APN internet. Per tale ragione ti consigliamo di rimuovere dalla lista degli APN quelli che non c'entrano, onde evitare sgradevoli impatti sulla bolletta telefonica; poi, usa sempre la connessione appropriata. Gli APN

comunemente utilizzati per le connessioni WAP sono *wap.tim.it* per TIM, *live.vodafone.com* per Vodafone, *mobile.libero.it* per WIND, *wap.postemobile.it* per PosteMobile; per le connessioni web (cellulare usato come modem) gli APN sono *tre.it* per TRE, *internet.postemobile.it* per PosteMobile, *web.omnitel.it* per Vodafone. Inoltre, se utilizzi il modem come cellulare, ricorda che in alcuni casi, nella configurazione sul sistema operativo è richiesta una stringa di comando addizionale, che varia da gestore a gestore, anche se ha un formato comune; senza questa, il modem non si connette (PosteMobile) o pur connettendosi mostra le pagine web vuote (può accadere con Vodafone). Le stringhe di configurazione sono comandi AT estesi e vanno inserite, per Windows XP, 7 ecc., nella casella *Comandi addizionali* della scheda *Avanzate* della finestra di dialogo *Proprietà Modem* (vedi la figura); comandi del genere sono +cgdcont=1,"IP","internet.wind" per Wind, +CGDCONT=1,"IP","datacard.tre.it","",0,0 oppure +CGDCONT=1,"IP","naviga.tre.it","",0,0" per TRE, +CGDCONT=1,"IP","ibox.tim.it","",0,0 per TIM e +CGDCONT=1,"IP","web.omnitel.it","",0,0 per Vodafone. Per PosteMobile il comando è +cgdcont=1,"IP","internet.postemobile.it",,0,0. Nota che in tutti i comandi, 1 indica la posizione in cui l'APN è memorizzato nella lista degli APN del tuo telefono, quindi se hai più APN e carichi questo comando addizionale, stai bene attento a mettere il numero della posizione esatta, altrimenti il cellulare punta ad un altro APN o la connessione non funziona correttamente.

RaspberryPi in scatola

Ho prenotato la scheda RaspberryPi e spero che prima o poi mi arrivi, nel frattempo sto pensando di utilizzarla come computer didattico per i miei figli, utilizzando qualche software di disegno e di giochi logici che sono disponibili in Internet. Per evitare danni alla scheda mi piacerebbe racchiuderla in un contenitore che la protegga da contatti indesiderati e ne valorizzi l'estetica.

Davide Bentivoglio - Ravenna



L'apposito contenitore Teko fornisce un aspetto professionale alla board RaspberryPi.

La popolarità della RasperryPi ormai è fuori discussione e anzi, a nostro parere, sta segnando una delle vie di evoluzione nel mondo dei micropc e dei sistemi embedded. Naturale che in molti si siano posti il problema di rivestire la RaspberryPi con una custodia adeguata. Ad esempio noi ne abbiamo prodotte alcune con la stampante 3D che stiamo progettando e che presenteremo in un prossimo numero della rivista. I file in formato .stl si trovano in diversi siti, fra tutti www.thingiverse.com. Per un utilizzo professionale TEKO ha appena reso disponibile il contenitore TEK-BERRY, in stile TEKO, che contiene perfettamente la RaspberryPi lasciando disponibili tutti i connettori tranne il pettine GPIO. A



proposito, in questo stesso numero troverai un articolo sulla RaspberryPi destinato agli appassionati di elettronica e che inizia a spiegare l'utilizzo del GPIO. Proveremo a modificare la TEK-BERRY con la nostra CNC per fare uscire i segnali GPIO da una apposita fessura, eventualmente con un cavo flat. Infine un'ultima curiosità, all'indirizzo *http://www.raspberrypi.org/archives/1310* puoi trovare i link per scaricare un modello per costruire un contenitore in cartoncino, ed anche il file originale per poterla personalizzare. Buon divertimento.

RM ELETTRONICA
Forniture per hobbisti, scuole e industria
Il tuo punto di riferimento per l'elettronica a ROMA
Distributore autorizzato:
FUTURA ELETTRONICA
ELSE Kit
ELETTRONICA
Via Val Sillaro 38 • 00141 Roma • Tel: 06-8104753

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

4ª edizione

infrastrutture e tecnologie per la **Smart City**

itn

27 - 28 settembre 2012
Centro Congressi Lingotto - Torino
www.itnexpo.com

INGRESSO GRATUITO | RISERVATO AGLI OPERATORI DEL SETTORE



INFRASTRUCTURA
- Open Data & Open Goverment
- Digital out of Home
- Wireless & Satellite Connectivity
- Smart Grid & Car recharge
- Wireless Sensor Networks
- Expo 2015
- GIS for Infrastructures
- Land Monitoring & Videosurvelliance
- Fund Raising & Financing

TELEMOBILITY
- On Board Telematics & Car Sensors
- AVL-AVM & Fleet Management
- Smart and Electronic Ticketing
- GNSS (GPS, Galileo, Glonass)
- Telematics Services & Location Based Systems
- Traffic information & Digital Signage
- Parking Solution and Systems
- Geocontent & Apps
- RF-ID & Asset Tracking
- Emergency and Safety
- Traffic Management & ZTL

partner organizzativo
CAMERA DI COMMERCIO INDUSTRIA ARTIGIANATO E AGRICOLTURA DI TORINO
TORINO 2015 9th World Congress BID

in collaborazione con
innovability · ClickUtility · TELEMOBILITY

networking partner
alps · enterprise europe network

con il supporto di
REGIONE PIEMONTE · infrastructura

Made by GL events

www.lingottofiere.it · @LingottoFiere · LingottoFiereTorino

LINGOTTO FIERE TORINO

7-11 SETTEMBRE 2012

IBC2012

Amsterdam • Olanda

IBC è la più importante manifestazione a livello mondiale dedicata alla produzione, gestione e distribuzione dei contenuti audio-video destinati al mondo dei media; un evento che mette in mostra gli sviluppi più recenti nel broadcasting, mobile TV, IPTV e digital signage, e in generale la tecnologia d'avanguardia e le nuove opportunità di business che si spingono anche oltre il mondo del broadcasting. Piattaforme di ultima generazione in 3D, fotocamere integrate, servizi video streaming basati sul Cloud, sono alcuni degli argomenti chiave di IBC2012.
Organizzato anche quest'anno presso il RAI Convention Center, conta oltre 1.300 espositori ed attende circa 50.000 visitatori da più di 160 paesi.
Nei padiglioni dello spazio espositivo, ognuno dedicato ad un tema differente, i marchi leader nel settore proporranno i loro prodotti e servizi più recenti offrendo una panoramica a 360° sulle più avanzate soluzioni tecnologiche di questo settore, con 250 categorie di prodotto coinvolte nella produzione, gestione e distribuzione di contenuti nei media e intrattenimento. Tra le aree di maggior interesse, IBC Connected World, che con un'area dimostrativa al padiglione 14 cercherà di interpretare l'impatto che le tecnologie wireless a banda larga (TV, 3G e 4G) avranno sull'industria di produzione dei Media; o anche Production Village, dedicata alla più recente tecnologia della produzione cinematografica, che al padiglione 11 vedrà allestito uno studio professionale dove provare i più recenti ritrovati come ad esempio telecamere ad alta velocità e sistemi 3D. Non mancherà la suggestiva Future Zone, dove verranno proposte e ipotizzate le tecnologie per il futuro, che per ora vivono solo nei laboratori di ricerca e sviluppo e che un giorno potranno diventare reali per tutti.
A IBC2012 non mancherà il consueto ciclo di conferenze, che ne fanno il principale forum internazionale per l'analisi dell'industria dell'intrattenimento elettronico.
Alle conferenze (che avranno inizio dal 6 settembre) parteciperanno i rappresentanti delle più importanti aziende del settore, offrendo una visione unica sullo stato attuale e cercando di tracciare una roadmap per il futuro. Tra gli appuntamenti in agenda, IBC Leader's Summit, conferenza riservata ai grandi operatori nel settore dei media elettronici e dell'industria dell'intrattenimento. Durante IBC2012 verranno assegnati gli Innovation Awards, che premieranno le persone e le realtà che si sono distinte per l'eccellenza del proprio business tecnologico e per le applicazioni rivolte al settore del digital broadcasting.
www.ibc.org



5-7 SETTEMBRE 2012

ZEROEMISSION Rome 2012

Roma

Fervono i preparativi per ZEROEMISSION ROME, la manifestazione dedicata a energie rinnovabili, sostenibilità ambientale, lotta ai cambiamenti climatici ed emission trading, che si svolgerà alla Fiera di Roma e che, su 45.000 metri quadrati, distribuiti in quattro padiglioni espositivi, includerà la decima edizione di EOLICA EXPO MEDITERRANEAN e la sesta edizione di PV ROME MEDITERRANEAN; temi dei settori espositivi, anche energia solare termica e termodinamica, batterie e sistemi di immagazzinamento dell'elettricità, veicoli elettrici, energia geotermica, efficienza energetica, energia mini-idroelettrica, biogas e cogenerazione. In collaborazione con AzzeroCO2, inoltre, ZEROEMISSION ROME propone quest'anno l'Help Desk Energia: dedicato alle imprese, offre loro consulenze personalizzate gratuite e proposte per migliorarne l'efficienza energetica.
ZEROEMISSION ROME ripropone anche altre iniziative di successo quali l'area Compomat Wind (materiali compositi per l'industria eolica) e MyPartnering, un efficace strumento per il business one-to-one che offre incontri mirati tra aziende e pubblico. Ritornano inoltre Solartech (dedicato all'energia solare termica e termodinamica) ed Electric Cars, area dedicata alla mobilità sostenibile nella quale sono esposti veicoli elettrici (che è anche possibile provare) stazioni di ricarica e pensiline.
ZEROEMISSION ROME sarà completato anche quest'anno da un qualificato programma congressuale, che conta 70 conferenze tenute da qualificati rappresentanti del mondo istituzionale, accademico, associativo e imprenditoriale, e che offriranno interessanti opportunità di aggiornamento professionale.
L'edizione 2012 di ZEROEMISSION Rome sarà caratterizzata da una forte internazionalità, grazie alla partecipazione di delegazioni provenienti da Albania, Argentina, Australia, Bulgaria, Cipro, Egitto, Giordania, Israele, Montenegro, Romania, Serbia, Slovenia e Sud Africa, che presenteranno in fiera le opportunità di investimento nei loro Paesi.
www.zeroemissionrome.eu

zeroEmission ROME
5-7 SETTEMBRE 2012 • ROMA
ENERGIE RINNOVABILI
PER IL MEDITERRANEO

■ 24-28 SETTEMBRE 2012

European Photovoltaic Solar Energy Conference and Exhibition

Francoforte • Germania

Presso la Fiera di Francoforte si terrà la 27ª edizione dell'European Photovoltaic Solar Energy Conference and Exhibition, uno degli eventi più importanti al mondo riguardante il fotovoltaico, dove cinque giorni di conferenze e quattro di esposizione consentiranno di fare il punto sullo sviluppo tecnologico e industriale del settore. La parte convegnistica inizierà già dal 24 settembre e vedrà, fra gli spazi in fiera e le sale del centro congressi, il susseguirsi di circa 1.600 tra keynote e sessioni plenarie (333), approfondimenti tecnologici e tavole rotonde dedicate alle ricadute economiche del fotovoltaico (86 eventi) nonché 1.300 presentazioni visuali.

Gli interventi sono stati selezionati dal Comitato Scientifico della manifestazione, composto da esperti di fama mondiale, per offrire ai partecipanti una panoramica completa e approfondita dello stato della ricerca scientifica e dello sviluppo industriale in campo fotovoltaico; altre presentazioni saranno focalizzate sugli aspetti economici di questo mercato, sulla diffusione e sulle stime di crescita, sui costi dei componenti e sulle possibilità di raggiungere in breve tempo la cosiddetta "grid parity". Il programma delle conferenze si articolerà tra argomenti come la tecnologia del fotovoltaico standard e ad alta efficienza, componenti per sistemi fotovoltaici e diffusione su larga scala del fotovoltaico

Non meno importante per il pubblico sarà l'esposizione dei prodotti industriali, dove saranno in mostra pannelli fotovoltaici proposti dai numerosi produttori mondiali, ma anche inverter, sistemi e software per il monitoraggio degli impianti e delle reti.

Non mancheranno le società di consulenza e progettazione e quelle che installano e conducono gli impianti.

EU PVSEC vedrà il susseguirsi, nei giorni della manifestazione, di eventi collaterali come il 9° European PV Industry Summit, l'Investors Day (dedicato agli aspetti economici del fotovoltaico) il PV Production Forum, Photovoltaics, Forms, Landscapes (dedicato all'integrazione del fotovoltaico nel paesaggio) e Photonics Forum 2012. Anche per quest'anno è annunciata la consegna degli EU PVSEC Student Awards, i riconoscimenti al lavoro svolto dai ricercatori più giovani nel settore del fotovoltaico.

www.photovoltaic-conference.com



■ 25-28 SETTEMBRE 2012

Security Essen 2012

Essen • Germania

Ad Essen, centro industriale nella regione della Ruhr la cui storia recente è strettamente legata alla famiglia Krupp e alle industrie siderurgiche ThyssenKrupp, ogni due anni si svolge la più importante manifestazione europea dedicata alle tecnologie della sicurezza ed alla protezione antincendio: Security Essen. Oltre 1.100 espositori provenienti da oltre 40 paesi proporranno prodotti e servizi destinati a questo settore la cui importanza è in continua crescita; anche la manifestazione tedesca, nonostante la recessione economica che sta investendo l'Europa, ha aumentato di circa il 10% il numero di espositori, con una crescente presenza di produttori dell'Estremo Oriente ma anche con una massiccia partecipazione di aziende italiane. Security Essen si appresta dunque a proporre una panoramica completa delle tecnologie più innovative sia nel campo della sicurezza che nella prevenzione incendi così come in quello della sicurezza meccanica e della protezione IT.

Il giorno precedente l'apertura della manifestazione verranno consegnati i premi "Security Innovation Award" alle aziende più innovative delle varie categorie. Quest'anno, a causa del crescente numero di espositori, l'area destinata alla prevenzione antincendio occuperà per la prima volta ben due padiglioni (1 e 9) mentre nel padiglione n. 6, nell'ambito del Security Forum, verranno tenute lezioni teoriche e pratiche sulla gestione dei sistemi di videosorveglianza; il padiglione 8 ospiterà invece il forum dedicato ai temi della ricerca e sviluppo.

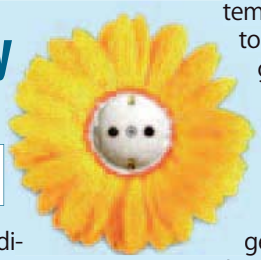
www.security-messe.de/en/security/index.html

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

20-22 SETTEMBRE 2012

Klimaenergy

Bolzano



Giunta alla sua quinta edizione, torna a Fiera Bolzano, Klimaenergy, l'esposizione fieristica che fornirà una panoramica sulle più moderne tecnologie nel campo delle energie rinnovabili. Il ricco spazio espositivo e l'articolato programma di eventi collaterali -tra cui un congresso internazionale, i tour guidati a impianti a fonti rinnovabili, workshop e conferenze specializzate, nonché la seconda edizione del salone per la mobilità sostenibile "Klimamobility"- sono i presupposti per la riconferma di una formula fieristica vincente. Con circa 200 aziende espositrici, Klimaenergy propone un ampio ventaglio di tecnologie e soluzioni all'avanguardia: non solo pannelli solari, ma una panoramica sul mondo delle rinnovabili che tocca solare termico, fotovoltaico, raffrescamento solare, illuminazione, biomasse, biogas e biocarburanti, idroelettrico, geotermia, cogenerazione, gassificazione del legno, idrogeno, celle a combustibile, eolico e servizi finanziari.

A fianco dello spazio espositivo, Klimaenergy propone il tradizionale congresso internazionale organizzato da Fiera Bolzano in collaborazione con la Provincia Autonoma di Bolzano, Eurac Research, Accademia europea di Bolzano, TIS Innovation, Ecoistituto Alto Adige, Fraunhofer Italia, IIT Istituto per le Innovazioni Tecnologiche e l'Unione Energia Alto Adige. Suddiviso in cinque moduli tematici, il congresso 'esplora' il mercato delle energie rinnovabili tra incentivi e sfide future. Il primo modulo, previsto per giovedì 20 settembre, offre un aggiornamento sulle nuove normative e sui nuovi incentivi in campo di rinnovabili. Venerdì 21, invece, sono in programma due sessioni parallele: una dedicata alle specifiche esigenze energetiche dei Comuni e delle istituzioni pubbliche con un focus sull'illuminazione, sui PAES e sul Patto dei Sindaci, l'altra incentrata sul tema della gassificazione del legno e sulla micro cogenerazione. Nell'ambito di Klimaenergy sono previsti, inoltre, un seminario specifico sul tema del solar cooling organizzato da Eurac Research, un convegno dell'Unione Geotermica Italiana sulle novità della geotermia in Italia e un convegno sull'energia alternativa in Alto Adige, organizzato dall'Unione Energia Alto Adige. Anche quest'anno, inoltre, Klimaenergy prevede quattro enertour a impianti funzionanti sul territorio altoatesino. Per gli spostamenti dal quartiere fieristico agli impianti più vicini sono previsti mezzi ecosostenibili.

L'evento è rivolto ad un pubblico altamente specializzato, composto da responsabili di uffici pubblici e istituzioni nonché da rappresentanti di settori ad alto fabbisogno energetico, quali industria, artigianato ed alberghiero.

www.fierabolzano.it/klimaenergy/

MOSTRE MERCATO

MONTICHIARI (BS)
RADIANTISTICA EXPO
Centro Fiere del Garda, Montichiari (BS)
Organizzazione: CENTRO FIERE DEL GARDA SPA
Tel: 030961148 – Fax: 0309961966
radiantistica@centrofiera.it
www.radiantistica.it
01 e 02 Settembre 2012

BUSTO ARSIZIO (VA)
COMPUTERFEST & RADIOAMATORE
Quartiere Fieristico, MalpensaExpo
Organizzazione: Compendio Fiere
info@compendiofiere.it
www.compendiofiere.it
08 e 09 Settembre 2012

CESENA (FC)
EXPO ELETTRONICA
Quartiere Fieristico – Cesena Fiera
Organizzazione: Blu Nautilus srl
Tel: 054153295 - Fax: 0541439584
email: info@blunautilus.it
www.blunautilus.it
08 e 09 Settembre 2012

VENTURINA (LI)
ELECTRONIC DAYS
Fiera di Venturina (LI), Via della Fiera 3
Organizzazione: Italfiere Srl
Tel: 0547415674 - Fax: 0547417357
info@italfiere.net
web: www.italfiere.net
15 e 16 Settembre 2012

PIACENZA
TELE RADIO - MONDO ELETTRONICA
Piacenzaexpo, Località Le Mose S.S. per Cremona
Organizzazione: Exposition Service
Tel: 054527548 - Fax: 0545291179
info@expositionservice.it
www.expositionservice.it
15 e 16 Settembre 2012

MONZA
EXPO ELECTRONICS MONZA
Palalper - Via Stucchi – Monza
Organizzazione: PrintMix
Tel: 0362999186 - Fax: 0362978042
printmix@printmix.it
www.printmix.it
15 e 16 Settembre 2012

MONTESILVANO (PE)
ELECTRONIC DAYS
Palacongressi d'Abruzzo, Montesilvano (PE)
Organizzazione: Italfiere Srl
Tel: 0547415674 - Fax: 0547417357
info@italfiere.net
www.italfiere.net
22 e 23 Settembre 2012

NOVEGRO (MI)
RADIANT & SILICON
Parco Esposizioni di Novegro (MI)
Organizzazione: Comis Srl
Tel: 027562711 - Fax: 027020835
radiant@parcoesposizioninovegro.it
www.parcoesposizioninovegro.it
22 e 23 Settembre 2012

OCCHIEPPO INFERIORE (BI)
BIELLARADIO – 12MA EDIZIONE
Centro polivalente via Caralli 7
Organizzazione: A.R.I. Biella
segretario@aribiella.it
www.aribiella.it
23 Settembre 2012

RAVENNA
MONDO ELETTRONICA
Pala De Andrè - Organizzazione: Exposition Service
Tel: 054527548 - Fax: 0545291179
email: info@expositionservice.it
www.expositionservice.it
29 e 30 Settembre 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.

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

Tubo luminoso 56 LED



€ 48,00
cod. T8-HWSMD

- Flusso luminoso 840 lm
- Consumo 10 W

Pannello da incasso 36 LED



€ 72,00
cod. P1717-WW

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

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

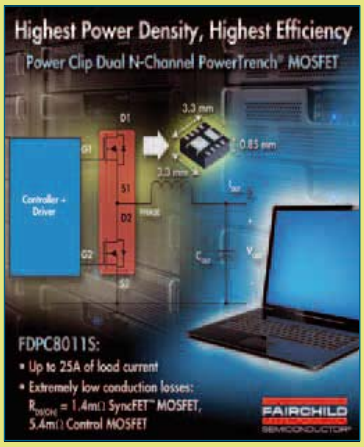
www.futurashop.it

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

Nuovi MOS PowerTrench e Power Clip da Fairchild

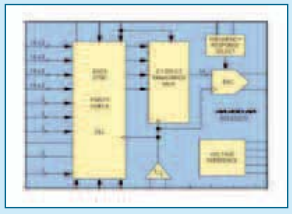
Fairchild Semiconductor annuncia due nuove gamme di MOSFET per le applicazioni mobile e DC/DC; per le prime rende disponibili FDZ661PZ e FDZ663P, due MOSFET PowerTrench a canale-P da 1,5V in package WL-CSP da 0,8mm x 0,8mm che, sfruttando un processo di packaging WL-CSP "fine pitch" allo stato dell'arte, minimizzano gli ingombri su scheda e il valore di R_{DSON}, raggiungendo caratteristiche termiche eccellenti per un componente tanto miniaturizzato. Per gli alimentatori DC-DC embedded ad alta densità, Fairchild ha sviluppato invece l'FDPC8011S, un modulo Power Clip Asymmetric Dual a canale-N



da 25V in un package 3,3×3,3mm² a basso profilo. Sviluppato per funzionare a frequenze di commutazione più elevate, l'FDPC8011S integra una tecnologia SyncFET da 1,4 mohm e un MOSFET di controllo a canale-N e basso fattore di merito da 5,4 mohm all'interno di un package all-clip che contribuisce a ridurre il numero di condensatori e le dimensioni degli induttori nelle applicazioni buck sincrone. L'FDPC8011S fornisce oltre 25A di corrente di uscita migliorando di più del doppio le caratteristiche di altri dual MOSFET convenzionali da 3x3 mm.

www.fairchildsemi.com

La tecnologia RF-DAC per le Macro e Small Cells Base Stations



Maxim annuncia il MAX5879, un RF-DAC multi-Nyquist per trasmettitori digitali SDR (Software Defined Radio), attraverso l'implementazione di una piattaforma hardware comune per Base Stations multi-carrier, multi-banda e multi-standard. Usando la sintesi digitale diretta (DDS) su FPGA, l'architettura del MAX5879 trasferisce nel dominio digitale le funzioni del modulatore in quadratura analogico e dell'oscillatore locale (LO); ciò permette di eliminare gli errori di fase, di quadratura e il feed through dell'oscillatore locale, fenomeni solitamente associati al processo di upconversion analogica. Le modalità di risposta in frequenza selezionabili col MAX5879 supportano la sintesi RF diretta delle bande cellulari 3G/4G in molteplici zone di Nyquist, comprese tra 700 MHz a 2,8 GHz. Un solo DAC RF MAX5879 può supportare standard quali GSM multicarrier, WCDMA e LTE. Il MAX5879 è incapsulato in case BGA a 256 pad da 17x17 mm.

www.maxim-ic.com

Il sensore NXP rileva la posizione della valvola a farfalla nei motori

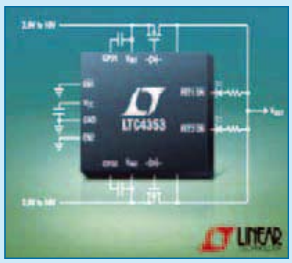
NXP Semiconductors ha annunciato il KMA220, un doppio sensore magnetico progettato per tutte le applicazioni automotive in cui serva misurare angoli con precisione e pensato per rilevare la posizione della valvola a farfalla che regola la portata d'aria nei motori a ciclo Otto. Il KMA220 viene fornito completamente calibrato e incorpora tre condensatori in aggiunta ai due sensori e due segnali di condizionamento ASIC. Ciò riduce i costi di sistema per i clienti e semplifica il processo di assemblaggio. Attualmente, per ottenere la ridondanza nelle applicazioni di controllo della valvola a farfalla le case automobilistiche devono installare due sensori su telaio in piombo o su un PCB; il KMA220 sostituisce questa soluzione. Il sensore KMA220 è eccezionalmente robusto, in grado di funzionare a temperature fino a 160 °C (la media dei concorrenti arriva a 150) il che gli permette di durare a lungo nei motori, dove l'uso della valvola EGR (ricircolo dei gas di scarico, necessario a ridurre gli NOx) alza la temperatura nel collettore d'aspirazione.



www.nxp.com

LTC4353, controller a doppio diodo ideale

Linear Technology presenta l'LTC4353, un controller a doppio diodo ideale che sostituisce due diodi Schottky ad alta potenza e consente il collegamento a bassa perdita tramite circuito OR di più alimentatori, garantendo nel contempo una perdita ridotta e un disturbo minimo delle tensioni di alimentazione. L'LTC4353 regola la caduta di tensione diretta attraverso MOSFET esterni a canale N per assicurare il trasferimento lineare della corrente tra gli alimentatori nelle applicazioni OR con diodo. Nei sistemi a bassa tensione un rilascio lento tra i controllori causa una caduta di tensione durante la commutazione dell'alimentatore. L'accensione in <1µs e l'architettura avanzata dell'LTC4353 garantiscono una commutazione lineare da un percorso all'altro, senza alcuna oscillazione. In caso di guasto o cortocircuito dell'alimentatore di ingresso, lo spegnimento rapido riduce al minimo la corrente inversa. Oltre a offrire il collegamento tramite circuito OR degli alimentatori ridondanti, l'LTC4353 è una buona alternativa al diodo nelle applicazioni di blocco dell'alimentazione nelle quali la perdita temporanea della tensione di alimentazione è isolata dal carico. L'LTC4353 dispone di ingressi di attivazione che possono essere utilizzati per garantire la priorità all'alimentatore principale quando le tensioni di alimentazione rientrano nella caduta reciproca dei diodi. Con entrambi gli ingressi di attivazione a zero logico, l'integrato consuma solo 75 µA per alimentatore.



Il controller funziona con alimentatori da 2,9 a 18 V e offre la flessibilità necessaria per controllare le tensioni fino a 0 V tramite un alimentatore esterno.

www.linear.com



Da Infineon, nuovi Transceiver per CAN

Infineon presenta l'ultima generazione di ricetrasmettitori ad alta velocità per applicazioni automotive CAN: si tratta della famiglia TLE7250 comprendente il TLE7250G e TLE7250GVIO. Il primo si interfaccia con i microcontrollori alimentati a 5 V, mentre il TLE7250GVIO può dialogare con micro sia a 5 che a 3,3 V. I nuovi transceiver offrono migliore immunità alle interferenze elettromagnetiche (EMI) nella gamma di frequenza da 1 MHz a 1 GHz e ridotta emissione elettromagnetica (EME) tra 150 kHz e 1 GHz. Inoltre, sono caratterizzati da una ESD di almeno ± 8 kV (norma IEC61000-4-2). Grazie alla loro alta impedenza, quando non sono alimentati i chip TLE7250 riducono la corrente di dispersione del CAN Bus a solo 5 μA in power-down. Insieme alla bassa corrente di riposo (inferiore a 15 μA) questo porta al miglioramento dell'efficienza energetica del sistema In-Vehicle Network (IVN). Tutti i membri della famiglia TLE7250 si basano sulla tecnologia Smart Power (SPT) di Infineon e hanno velocità di trasmissione dei dati fino a 1 Mbps.

www.infineon.com

Amplificatore stereo in classe G ad alta efficienza con interfaccia I²S



Da Toshiba arriva un nuovo amplificatore integrato in classe G a bassissima potenza per cuffie stereo, che combina straordinarie prestazioni audio con un'interfaccia digitale I²S. L'integrato è siglato TC94B15WBG ed è l'ideale per cellulari, lettori MP3, cuffie e auricolari Bluetooth, navigatori satellitari e altri dispositivi a batteria che richiedono un'amplificazione audio efficiente e di alta qualità. Gli amplificatori in classe G funzionano con basse tensioni di alimentazione (nel nostro caso 1,8 V per l'analogica e 4,5 V per la parte digitale) per bassi livelli del segnale audio, ma possono temporaneamente estendere la dinamica quando ricevono segnali più ampi; perciò il TC94B15WBG raggiunge una maggiore efficienza rispetto alla classe A/B, pur mantenendo bassa la distorsione del segnale audio. Rispetto alle soluzioni in classe D, non richiede il filtro di uscita. Il TC94B15WBG è stato progettato per minimizzare il numero dei componenti e lo spazio occupato sulla scheda. Il circuito di conversione DC/DC incorporato facilita il collegamento diretto alla batteria; uno stadio di conversione D/A sigma-delta integrato con filtri passa-basso digitali supporta l'interfaccia digitale I²S. Inoltre, un controllo interno del volume (con possibilità di spegnere completamente l'audio grazie alla funzione "mute") estende la gamma dinamica da -63dB a 0 dB in 32 passi. La distorsione armonica THD+N è inferiore allo 0,02% e il SNR è pari a 92 dB. L'integrato è disponibile in case WCSP a 20 contatti con passo da 0,4 mm.

www.ti.com

Modem Power Line da Semiconductor basato su ARM Cortex



Il nuovo NCN49597 di ON Semiconductor è un modem Power Line che sfrutta l'architettura ARM Cortex e innovative funzioni per migliorare l'affidabilità della comunicazione e la versatilità nei sistemi di telemisura del consumo di elettricità, nell'home automation, nel controllo dell'illuminazione e negli impianti fotovoltaici. L'NCN49597 è un driver Power Line in classe A/B ad alta efficienza da 2 A, che integra un filtro del 4° ordine e un limitatore di corrente programmabile con una resistenza; combina un processore ARM Cortex M0 a 32-bit con un front-end analogico ad alta precisione; basato sulla tecnologia S-FSK (Spread Frequency Shift Keying) a 4.800 baud, è stato ottimizzato per garantire alta efficienza e affidabilità anche negli ambienti più disturbati. Il componente è riprogrammabile in campo e si sincronizza automaticamente. La possibilità di scegliere la frequenza della portante tra le bande A e D CENELEC e la possibilità di operare con frequenze di rete elettrica sia di 50 che di 60 Hz, ma anche in continua, assicura la massima versatilità d'impiego. Inoltre il componente dispone di 10 I/O generici che permettono di collegarlo a LCD e integrati in grado di compiere misure sulla rete elettrica, il che permette di realizzare un sistema completo senza bisogno di un microcontrollore. L'NCN49597 è disponibile in case QFN-52 da 8x8x1 mm.

www.onsemi.com

Micro-fotoaccoppiatore per moduli di potenza intelligenti funzionante fino a 125 °C

Toshiba Electronics Europe presenta un nuovo fotoaccoppiatore integrato ultra miniaturizzato dedicato agli azionamenti industriali realizzati con moduli di potenza intelligenti (IPM) che devono funzionare su un esteso range di temperature. Il componente, siglato TLP714, può operare in un intervallo di temperatura compreso fra -40 a 125 °C ed è disponibile in un contenitore SDIP6 compatto che sulla scheda occupa uno spazio di soli 9,7x4,58 mm. L'ingombro verticale è di appena 4,0 mm. L'ampia dinamica di uscita, da 4,5 V a 30 V, e la velocità di trasmissione, pari a 1 Mbps, rendono il dispositivo la scelta ideale in una vasta gamma di applicazioni, come azionamenti, servocomandi e sistemi di automazione industriale, che richiedono un isolamento tra il modulo di potenza intelligente e i circuiti integrati di comando.
Sebbene lo spazio occupato sulla scheda sia circa la metà rispetto quello di un dispositivo equivalente in contenitore DIP8, il fotoaccoppiatore TLP714 soddisfa i requisiti degli standard internazionali di sicurezza per le apparecchiature a isolamento rinforzato.
La tensione di isolamento minima è pari a 5.000 Veff, mentre uno schermo antirumore incorporato garantisce un'immunità ai transitori di modo comune di ±20 kV/μs.



www.toshiba-components.com

Nuovi operazionali zero-drift da Microchip

Due nuovi prodotti ampliano l'offerta di operazionali zero-drift di Microchip: sono l'MCP6V11 e l'MCP6V31, basati sulla tecnologia CMOS, e possono funzionare con un'alimentazione di appena 1,6V assorbendo, a riposo, solo 7,5 µA. La nuova famiglia MCP6V11/31 offre grande precisione e package di dimensioni ridotte, ideali per le applicazioni portatili legate a progetti consumer, industriali e medicali. Dispositivi come l'MCP6V11/31 permettono lo sviluppo di prodotti medicali portatili più efficienti e dotati di hardware e software di condizionamento dei segnali; in essi, i progettisti di applicazioni industriali che richiedono bassa potenza, ingombri ridotti, gestione termica semplificata e controllo dei costi (quali sensori di condizionamento e strumentazione portatile) possono trarre vantaggio dalle prestazioni degli amplificatori operazionali MCP6V11/31. L'architettura self-correcting della famiglia MCP6V11/31 offre una tensione massima di offset di ingresso di soli 8 µV, con deriva molto contenuta.



L'MCP6V11 raggiunge un prodotto banda-guadagno di 80 kHz, con una corrente tipica di riposo di soli 7,5 µA; la versione MCP6V31 offre invece un prodotto banda-guadagno di 300 kHz e una corrente tipica di riposo di 23 µA. MCP6V11 e MCP6V31 sono incapsulati in package a 5-pin SOT-23 e SC-70.

www.microchip.com

Da Avago, il primo TX/RX Versatile Link in fibra ottica a 3,3 V, da 0 a 50 Mbps



Il trasmettitore AFBR-1624Z utilizza una sorgente a LED da 650 nm con ottica e driver integrati, per un efficiente accoppiamento con fibre ottiche polimeriche (POF, Polymer Optical Fiber) da 1 mm. L'ingresso del trasmettitore è compatibile con le famiglie logiche TTL. Il trasmettitore opera con ogni tipo di segnale da DC a 50 MBd fino a una distanza di 50 metri con POF da 1 mm. Il trasmettitore della Avago è un dispositivo a 3 pin inserito in un contenitore Versatile Link. I componenti Versatile Link possono essere intrecciati (collegati insieme a N) per minimizzare lo spazio occupato e per realizzare un doppio collegamento con i connettori duplex. Per i componenti Versatile Link sono disponibili da Avago anche vari connettori simplex e duplex, oltre ai cavi POF.

Caratteristiche:
Compatibili RoHS • Trasmissione dati per segnali da DC a 50 Mbps • Integrano LED da 650 nm e driver con logica d'ingresso TTL • Distanza fino a 50 metri con fibra ottica in plastica (POF) da 1 mm • Gamma della temperatura operativa -40 °C ÷ +85 °C • Isolamento ad alta tensione • Eliminazione dei loop di massa • Riducono la suscettibilità ai transitori di tensione.

www.avagotech.com

TE Connectivity: relé RZ da 12/16 A con isolamento rinforzato



La nuova serie RZ di TE Connectivity comprende relé a singolo polo con contatti normalmente aperti o di conversione ed offre la scelta fra materiali di contatto AgNi o AgSnO, comunque in grado di garantire elevate prestazioni. A una temperatura ambiente di 85 °C, con una corrente di commutazione di 16 A e 50.000 cicli, sono approvati VDE e riconosciuti UL. Sono disponibili sia i modelli standard per una temperatura ambiente di 85 °C, sia le versioni HOT, capaci di funzionare fino a 105 °C, oltre ai modelli con coperchio trasparente. La potenza nominale della bobina è di 400 mW. Questi prodotti soddisfano il test a filo incandescente descritto nella specifica IEC60335-1 e sono conformi alla Direttiva RoHS. Concepito principalmente come relé per circuito stampato, l'RZ può essere montato anche su zoccoli DIN rail. La recente acquisizione dell'approvazione CQC, che si aggiunge alle precedenti approvazioni UL e VDE, aumenta il livello di idoneità del relé RZ per le applicazioni in tutto il mondo. Le principali caratteristiche della serie RZ sono:
- contatto da 1 polo 12/16 A, 1 form C (CO) o 1 form A (NO);
- bobina DC da 400 mW;
- contatto bobina 5 kV/10 mm ad isolamento rinforzato;
- temperatura ambiente 85 °C (con coperchio chiaro 70 °C).

www.te.com

Microfoni MEMS in plastica più eleganti e più robusti

STMicroelectronics è la prima al mondo a produrre, in grandi volumi, microfoni MEMS utilizzando package di plastica. Ciò permette di risparmiare spazio e aumenta la vita delle applicazioni professionali e consumer che vanno dai cellulari e tablet ai rilevatori di rumorosità ambientale, alle cuffie acustiche con funzioni di cancellazione del rumore; inoltre garantisce eccellenti prestazioni acustiche ed elettriche, un'imbattibile robustezza meccanica (i dispositivi resistono a 40N di forza) e spessori più ridotti, mentre riduce ulteriormente le dimensioni del chip del microfono a soli 2×2 mm. Questo sviluppo tecnologico apre la strada all'introduzione di microfoni incorporati all'interno di cavità del silicio, come ulteriore e definitivo passo avanti nella miniaturizzazione dei dispositivi. I microfoni MEMS della ST sono adatti all'assemblaggio su schede a circuito stampato a cavo piatto che semplificano la progettazione dei più avanzati sistemi di elettronica di consumo in cui gli spazi a disposizione sono molto limitati. La tecnologia permette di collocare il "foro" da cui passa il suono sia sulla parte superiore sia su quella inferiore del package, in modo da ridurre al massimo lo spessore del sistema completo e garantire che il suono proveniente dall'esterno raggiunga il microfono percorrendo il percorso più breve possibile.

www.st.com



Power Fortronic



La continua evoluzione dell'elettronica di potenza richiede un costante aggiornamento da parte dei progettisti e degli operatori. Per fare il punto su questa tecnologia, l'Electronic Community si incontra al CenterGross Zanhotel di Bentivoglio (Bologna) il prossimo 20 settembre, in un evento che coniuga a momenti formativi e convegni, un'area in cui le aziende saranno presenti con piccoli stand che consentiranno di accogliere direttamente i visitatori ed esporre le ultime novità.
All'area espositiva, nel corso della mattinata si affiancherà una sessione plenaria introduttiva con tre percorsi (Very high Power -Power Mosfet, IGBT, IGCT, IPM-; Nuove tecnologie -SiC, GaN, ecc.-; Power Management) dando al visitatore la possibilità di scegliere il tema più vicino ai propri interessi e che in ogni caso fornirà alla platea una puntuale informazione sullo stato dell'arte delle tecnologie più significative; il pomeriggio sarà caratterizzato da una serie di workshop organizzati direttamente dalle aziende partner su sistemi, prodotti e soluzioni specifiche per l'elettronica di potenza che saranno raggruppati per temi applicativi e tecnologia. In particolare si parlerà di controllo motori, strumentazione per applicazioni di potenza, conversione AC/DC e DC/DC, normative e standard negli alimentatori e componenti passivi (condensatori, filtri magnetici, elementi per thermal management) impiegati nel controllo dei circuiti di potenza.
L'evento offrirà quindi una panoramica generale a cura di esperti e accademici, in una serie di convegni verticali, con lo scopo di approfondire gli aspetti più significativi dell'elettronica di potenza. Le sessioni convegnistiche prevedono il coinvolgimento di alcuni dei più importanti produttori di componenti di potenza e fornitori di prodotti e servizi complementari (Strumentazione, Circuiti stampati, eccetera).
L'ingresso al forum è gratuito e riservato a chi opera in questo ambito; ai partecipanti verrà distribuita la presentazione delle aziende espositrici con i relativi prodotti di punta e una serie di articoli a commento del convegno plenario e delle sessioni pomeridiane.
www.fortronic.it

Nanoforum 2012

Dal 24 al 26 settembre torna nella Capitale, presso l'Università La Sapienza, l'evento in cui ricerca e impresa si incontrano per dibattere sulle micro e nanotecnologie, argomenti quanto mai attuali, se è vero che entro il 2020 circa il 20% di tutti i prodotti fabbricati nel mondo impiegherà una certa quota di nanotecnologie e che il mercato dei nanomateriali potrà muovere, nello stesso periodo, 3.000 miliardi di dollari. Queste previsioni sono a livello mondiale ma l'Italia non è certo rimasta fuori, dato che già l'anno scorso esistevano circa 200 realtà impegnate in attività di ricerca e sviluppo sulle nanotecnologie, il 57% delle quali faceva riferimento ad istituzioni pubbliche ed il restante 43% ad imprese private.
Nanoforum, giunto alla sua ottava edizione, nasce proprio per fare il punto sullo stato della ricerca e dello sviluppo industriale delle nanotecnologie in Italia (ma anche all'estero), conoscere le più recenti applicazioni industriali e anticipare gli sviluppi nel futuro prossimo, grazie a un ampio programma di convegni e ad un'area espositiva molto specializzata.
Accanto al ciclo di conferenze ed all'area espositiva, l'edizione di quest'anno proporrà il Nanoforum Tour, una sessione dove aziende e centri di ricerca apriranno al pubblico le porte dei propri laboratori, per far conoscere più da vicino la realtà del mondo delle nanotecnologie.
Grazie al contributo delle istituzioni partner, degli espositori e degli sponsor, la partecipazione a nanoforum è completamente gratuita previa iscrizione, effettuabile dal sito dell'evento.
www.nanoforum.it



Machine Automation

Il 27 settembre, presso il Centro Congressi Cosmo Hotel Palace Cinisello Balsamo (MI), si terrà un "one-day event" che consentirà a tecnici, progettisti e specialisti che operano nel mondo delle macchine automatiche, di scoprire in anteprima le tendenze tecnologiche del settore. La mattinata sarà aperta da una sessione plenaria nel corso della quale saranno affrontati i temi caldi dell'automazione a bordo macchina, a partire dalle problematiche legate alla sicurezza, per arrivare ai bus di campo più utilizzati. Nel pomeriggio si terranno workshop e seminari focalizzati su tecnologie e soluzioni. Per quanto riguarda la parte convegnistica, nel corso della giornata si susseguiranno seminari tecnici tenuti dalle aziende espositrici, della durata di 30 minuti ciascuno.
Accanto al ciclo di conferenze, è previsto uno spazio espositivo che consentirà ai visitatori di incontrare esperti e fornitori e mettere a confronto il meglio dell'offerta disponibile sul mercato: motori, azionamenti, inverter, controllori, interfacce, comandi, pannelli, sensori, bus di campo e software per l'automazione a bordo macchina.
Machine Automation è un evento rivolto e riservato ad un pubblico specializzato formato da direttori tecnici, progettisti, tecnici di manutenzione, responsabili di qualità e della sicurezza, tecnici e responsabili di produzione, buyer, ricercatori e responsabili R&S che operano per OEM, costruttori di impianti industriali, system integrator e utilizzatori finali.
Nell'ambito della mostra convegno Machine Automation si terrà l'edizione 2012 di IA Forum, la cui keynote verterà sulla sicurezza di macchine e impianti; il convegno di quest'anno, organizzato da Anie-AssoAutomazione e dalla rivista Automazione e Strumentazione farà il punto sulle diverse tecnologie esistenti per realizzare installazioni sicure, illustrando le tipologie di prodotti per comando e controllo e le possibili applicazioni. Il convegno si concluderà con una tavola rotonda che vedrà protagonisti gli utilizzatori, invitati a condividere con la platea casi concreti di problematiche risolte con successo.
www.mostreconvegno.it

ITN 2012

Ritorna a Torino, con una formula rinnovata, l'evento ITN, realizzato da GL events Italia-Lingotto Fiere in partnership con la Camera di commercio di Torino e con la collaborazione di Innovability e Clickutility. Dedicato a infrastrutture e tecnologie per la smart city, ITN si svolgerà il 27 e 28 settembre 2012 presso il Centro Congressi Lingotto insieme a Telemobility Forum, per dare forma all'evento italiano di riferimento per le nascenti smart city e smart community. ITN farà il punto sulle infrastrutture digitali per le future smart city e l'open government, mentre Telemobility affronterà le tecnologie e le soluzioni per la smart mobility, dalla navigazione satellitare all'infomobilità. Il tutto con un format di mostra-convegno dove i partecipanti potranno usufruire di due giornate di interventi, con presenze sia italiane che internazionali, approfondimenti di carattere tecnico e commerciale, incontri continui con gli operatori del settore e con i rappresentanti delle istituzioni nell'area espositiva, one-2-one business meeting.



Ad ITN, sessioni specifiche riguarderanno lo stato dell'arte e le opportunità per il comparto attraverso workshop dedicati, in cui verranno presentati numerosi esempi di best practice, dove la parola d'ordine sarà "intelligenza": nel progettare, nell'ottenere e gestire le informazioni, nell'utilizzare le Infrastrutture digitali utili alla creazione delle future Smart City.

Un workshop tematico e una gallery espositiva presenteranno i progetti più innovativi e faranno incontrare le realtà innovative e le loro offerte con le istituzioni chiamate a gestire un grande appuntamento come l'Expo 2015.

Evento speciale all'interno di ITN, Brokerage Event favorirà incontri bilaterali tra imprese, università e centri di ricerca provenienti da tutta Europa e impegnati nello sviluppo di nuove soluzioni e servizi ICT per il settore dei trasporti in generale.

www.itnexpo.it/itn/

Light 2012

In tutta Europa, la notte del 28 settembre sarà dedicata ai ricercatori e alle loro scoperte; in Italia, per l'occasione, al Planetario di Roma (dove nell'edizione 2011 sono arrivate circa 20.000 persone) andrà in scena Light 2012, un evento per aprire le porte al futuro, sperimentare, provare e -perché no- giocare con la ricerca, con i suoi protagonisti ed altro ancora, durante la notte dei ricercatori. Promozione, comunicazione e divulgazione scientifica: sono questi gli scopi dell'evento, organizzato da PON Ricerca e Competitività in collaborazione con l'IRPPS del CNR. Il tema della notte sarà "Science on Breaking News" cioè "La Scienza in prima pagina". Perché la curiosità non è mai stata spinta così al limite e può essere soddisfatta anche in modo leggero e divertente, come l'evento romano propone, attraverso laboratori, esperimenti, momenti di divulgazione scientifica per grandi e piccoli, accompagnati da spettacoli dal vivo. Un'occasione per trascorrere una notte diversa dove imparare divertendosi.

L'evento è gratuito e aperto a tutti. Light, la Notte dei Ricercatori, si svolge a Roma presso il Planetario e Museo Astronomico, ma anche in contemporanea in quattro città del Sud Italia: Bari, in piazza Della Libertà e C.so V. Emanuele II, a Benevento, presso la Rocca dei Rettori, a Rende (Cosenza) al Museo del Presente ed a Palermo, presso l'Orto Botanico.

www.eventolight.it



EIPC Summer Conference

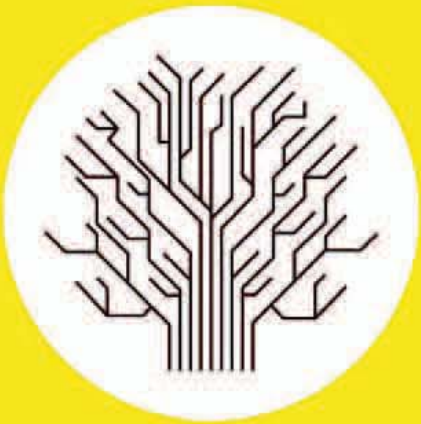
Il 13 e 14 settembre 2012 presso l'Hotel Ramada Plaza di Milano, andrà in scena l'EIPC Summer Conference, un evento organizzato dall'EIPC (European Institute of Printed Circuits) al quale parteciperanno delegati e ospiti provenienti da tutta Europa; tra i temi affrontati spiccano la keynote sull'electromobility come importante opportunità per l'industria europea dei PCB e le sessioni dedicate a nuovi modi di fare innovazione come i progetti di ricerca collaborativi. Il convegno rappresenta un "must" per le aziende che hanno rapporti con l'industria di produzione dei PCB e che desiderino restare aggiornati sui trend delle esigenze del mercato, su materiali, attrezzature e processi di produzione e per coloro che abbiano la necessità di migliorare l'efficienza dei propri processi e l'affidabilità dei loro prodotti.

Lo scopo dell'evento è mettere a disposizione degli invitati e dei delegati una piattaforma che permetta di scambiarsi informazioni sulle condizioni del mercato e sulle innovazioni future relative al mondo delle connessioni e del packaging.

EIPC Summer Conference è rivolto a progettisti e decision maker nel campo degli OEM, degli EMS, dell'assemblaggio e della produzione di circuiti stampati, così come a progettisti di schede e sistemi. I progettisti di schede alla ricerca di soluzioni per abbattere i costi e perseguire la miniaturizzazione, acquisiranno numerose informazioni circa lo sviluppo delle tendenze nelle diverse applicazioni. I produttori di schede e gli ingegneri di sviluppo acquisiranno informazioni che li aiuteranno a focalizzarsi su prodotti più redditizi. Specialisti e marketing manager guadagneranno in competenze sulle applicazioni più avanzate nel settore dei PCB. Infine, gli OEM avranno modo di prepararsi per affrontare la prossima generazione di prodotti elettronici.

Per i visitatori che desiderano approfondire punti e argomenti specifici, è prevista un'area dedicata agli incontri one-to-one. Durante lo svolgimento del convegno, EIPC offrirà l'opportunità di scambiare informazioni e organizzare discussioni in un'area dedicata dove i visitatori potranno ottenere informazioni aggiornate sugli ultimi prodotti o sul relativo know-how in modo informale o mediante incontri one-to one.

www.eipc.org



Fiera dell'Elettronica e del Radioamatore

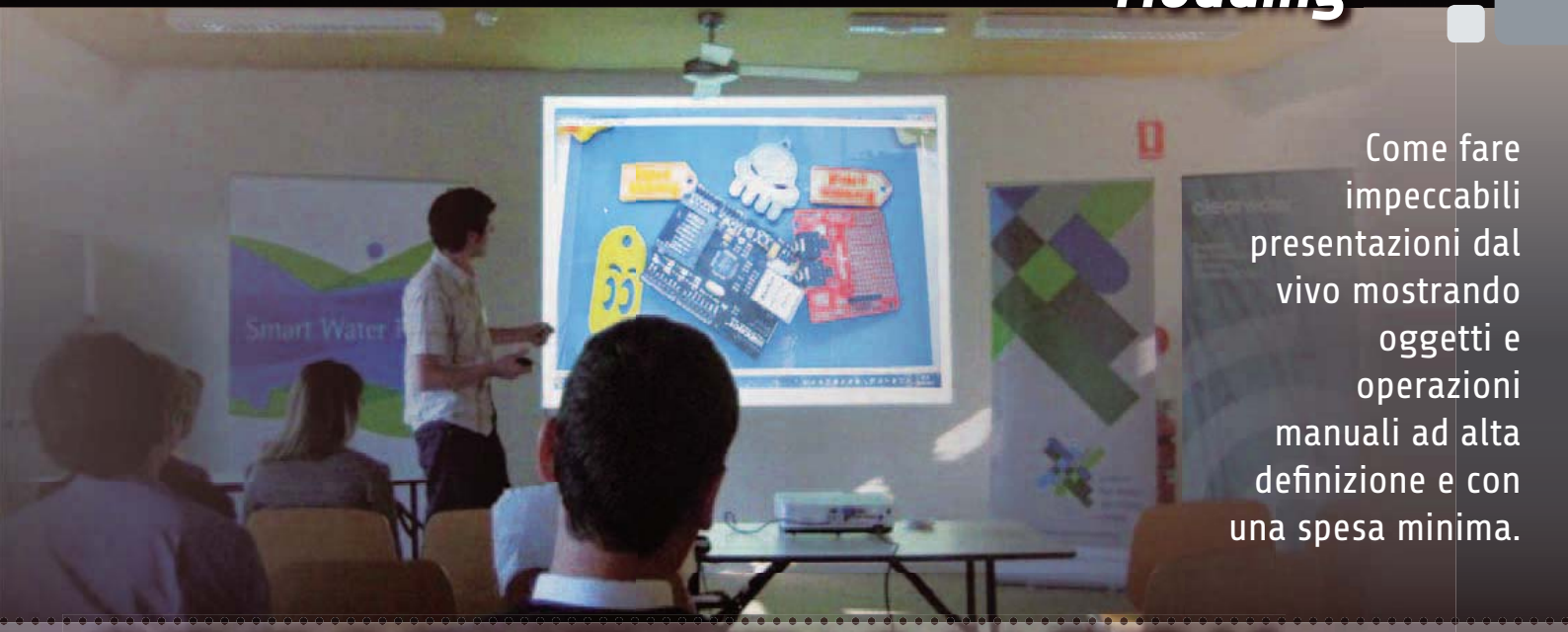
Fiera Nazionale.
Hardware, software, componentistica, telefonia ed elettronica di consumo

29 e 30 settembre 2012
dalle ore 8.30 alle 18
Padiglioni Fiera Millenaria
Gonzaga (Mantova)

fuchsia.com

FIERA 1000 NARIA

Fiera Millenaria di Gonzaga srl
Via Fiera Millenaria 13, Gonzaga (MN)
T +39 0376 58098 F +39 0376 528153
info@fieramillenaria.it **www.fieramillenaria.it**

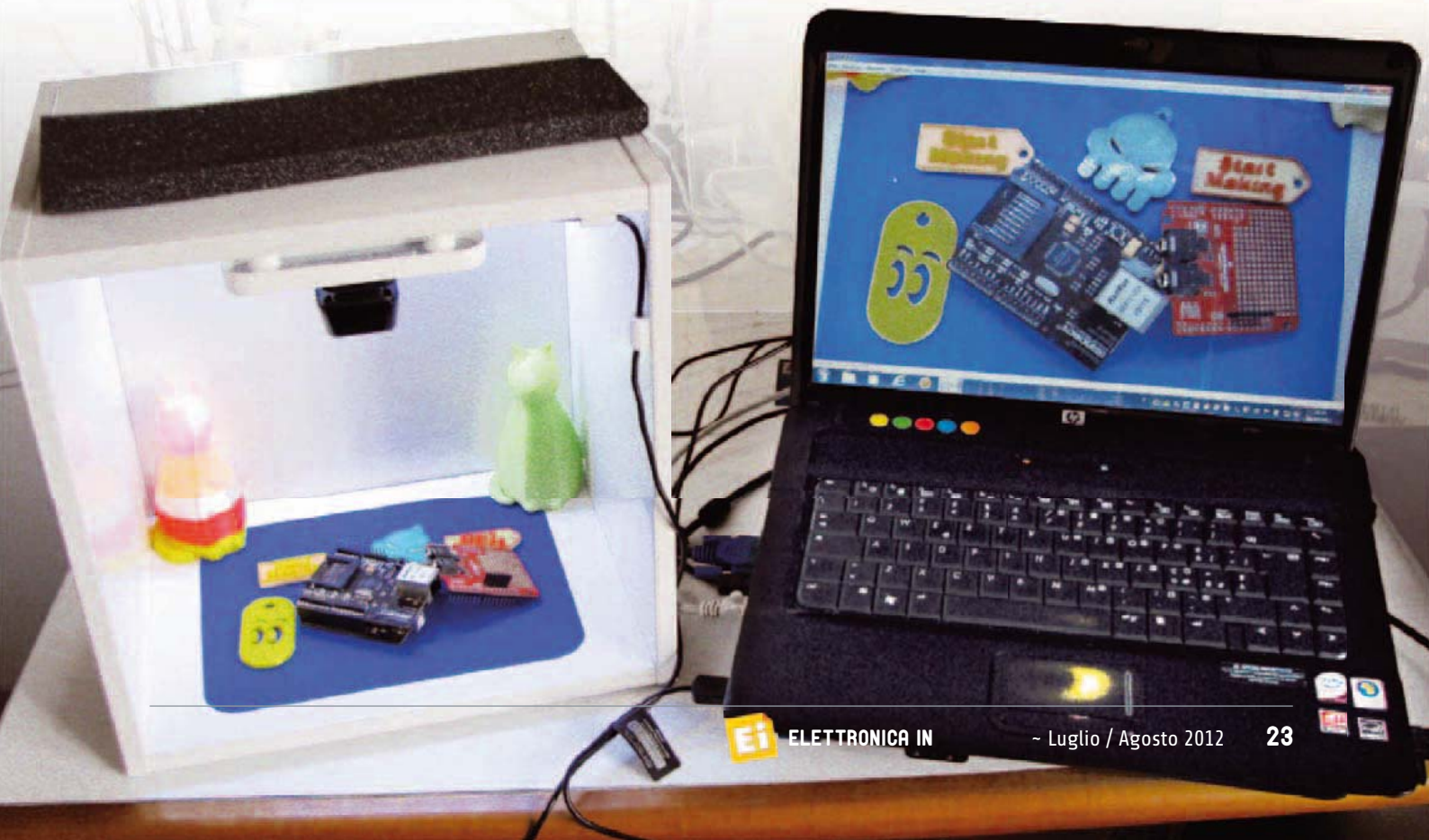


Come fare impeccabili presentazioni dal vivo mostrando oggetti e operazioni manuali ad alta definizione e con una spesa minima.

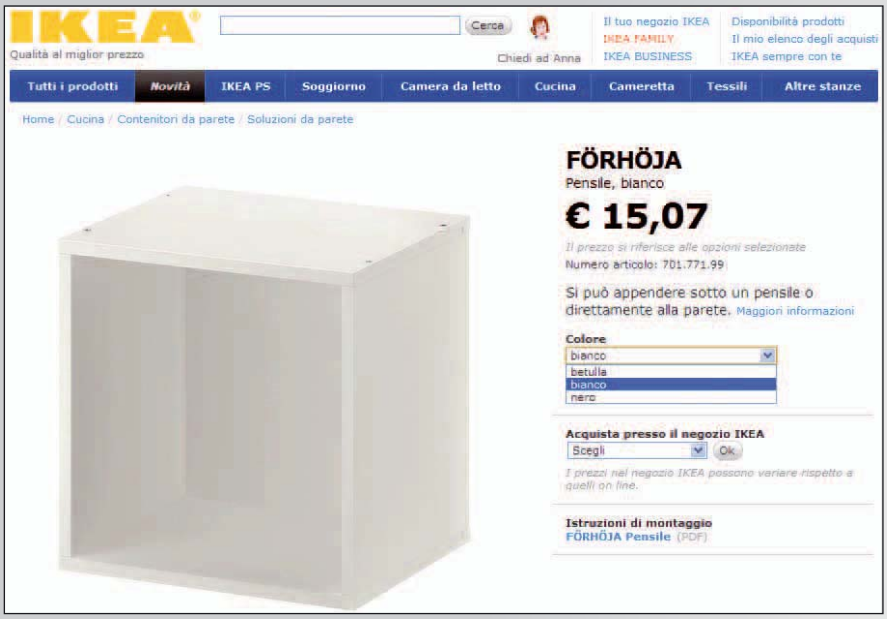
PRESENTATION BOX

di SIMONE MAJOCCHI

I mobili di Ikea sono fra i più comodi per realizzare a basso costo delle soluzioni pratiche ed eleganti. Qualche tempo fa dovevamo organizzare un workshop per dimostrare come utilizzare un Arduino per costruire un progetto ricco di parti e pezzetti. L'audience fatta da quasi una cinquantina di studenti aveva bisogno di vedere bene ogni passaggio del montaggio e quindi abbiamo pensato che una video proiezione sarebbe stata l'ideale per mostrare il tutto. Chi ha provato ad utilizzare una webcam sa però che ci sono alcuni punti critici fra i quali



Materiale necessario



PENSILE FORHOJA



LIFECAM HD-5000



VITE DA LEGNO
PER FISSARE IL MAGNETE

il tenere stabile e ferma l'inquadratura e avere una buona illuminazione degli oggetti inquadrati.
Partendo da questa esigenza, abbiamo dato un'occhiata al nostro "magazzino" delle parti Ikea in cerca d'ispirazione e due oggetti hanno subito attirato la nostra attenzione: il "cubo" pensile Forhoja e il sistema a magneti da cucina Bygel da 120mm per attaccare a parete i coltelli.
Come materiale elettronico avevamo già una webcam Lifecam HD-5000 di Microsoft, e capace di effettuare ottime riprese in HD a 720p e 30 frame al secondo, un pezzo di strip led bianchi e un alimentatore da 12V 500 milliampere.

DA UN'INTUIZIONE ALLA REALIZZAZIONE PRATICA

Immaginandoci in cattedra con il nostro pubblico desideroso di conoscere come avvitare quella specifica vite o di vedere dove collegare il cavetto del motore servo, abbiamo quindi capito che uno spazio chiuso e circoscritto in cui mettere e spostare gli oggetti da riprendere era un buon punto di partenza. Le dimensioni del pensile (27 x 27 cm di apertura frontale e 24 cm di profondità) erano adeguate per contenere le schede e le parti meccaniche del materiale oggetto del nostro workshop che, proprio perché ricco di dettagli, richiedeva una ripresa da una distanza ravvicinata.
La webcam Microsoft LifeCam HD-5000 è un dispositivo di fascia HD che, a un costo molto contenuto, offre delle ottime riprese video sia di tipo Live che per registrazioni su file compatibili anche con YouTube. Trattandosi di una webcam USB, basta il collegamento al PC per alimentazione e video, quindi un solo cavetto piccolo e flessibile da gestire.
Più complesso il collocamento della camera stessa nel box: il primo dubbio che abbiamo dovuto affrontare riguardava l'angolo di ripresa. L'ottica è fissa e quindi in un cubo o si ha la giusta ampiezza o si deve cambiare camera. Ci è andata bene e posizionata sul lato superiore, tutto il piano inferiore veniva correttamente inquadrato con la sola distanza di 20 centimetri dal piano stesso. LifeCam ha l'autofocus, ma spesso fa una rimessa a fuoco passando dallo sfuocato totale all'immagine nitida e quindi conviene disattivare la funzione e passare al modo manuale per evitare questo fastidioso inconveniente.
Chiarito questo importante elemento di compatibilità della webcam disponibile con la sua destinazione d'uso, siamo passati al tema del fissaggio. Le possibilità erano un inelegante approccio con viti e staffettine o… una barretta magnetica a cui solitamente in cucina si attaccano i coltelli.
Tempo fa abbiamo trovato da



STRIP LED WHITE 20 cm

Il nostro progetto è nato un po' per necessità e un po' per caso. I componenti utilizzati possono essere sostituiti con altri elementi simili. Il box Forhoja è a nostro avviso la scelta migliore per costo e dimensioni, mentre il supporto magnetico della Webcam e la strip LED possono essere sostituiti con elementi funzionalmente identici. La nostra Webcam ha il "piede" che contiene metallo e si fissa comodamente con il nostro sistema; se la webcam da voi scelta non ha un piede magnetico potete usare una pinza e fissarla al posto del magnete.



IL MAGNETE PER COLTELLI BYGEL VA FISSATO CON VITI DI LUNGHEZZA NON SUPERIORE A 22 mm

OCCORRENTE:

Articolo	Casa produttice	Prezzo approssimativo
Pensile Forhoja	Ikea	€ 15,00
Magnete per coltelli Bygel	Ikea	€ 5,00
Lifecam HD-5000	Mediaworld o GDO	€ 30,00
20 cm di strip LED white	Futura Elettronica	€ 7,00
Alimentatore 12V	Futura Elettronica	€ 5,00
Mousepad di gomma sottile	Computer shop	€ 2,00
TOTALE		**€ 64,00**

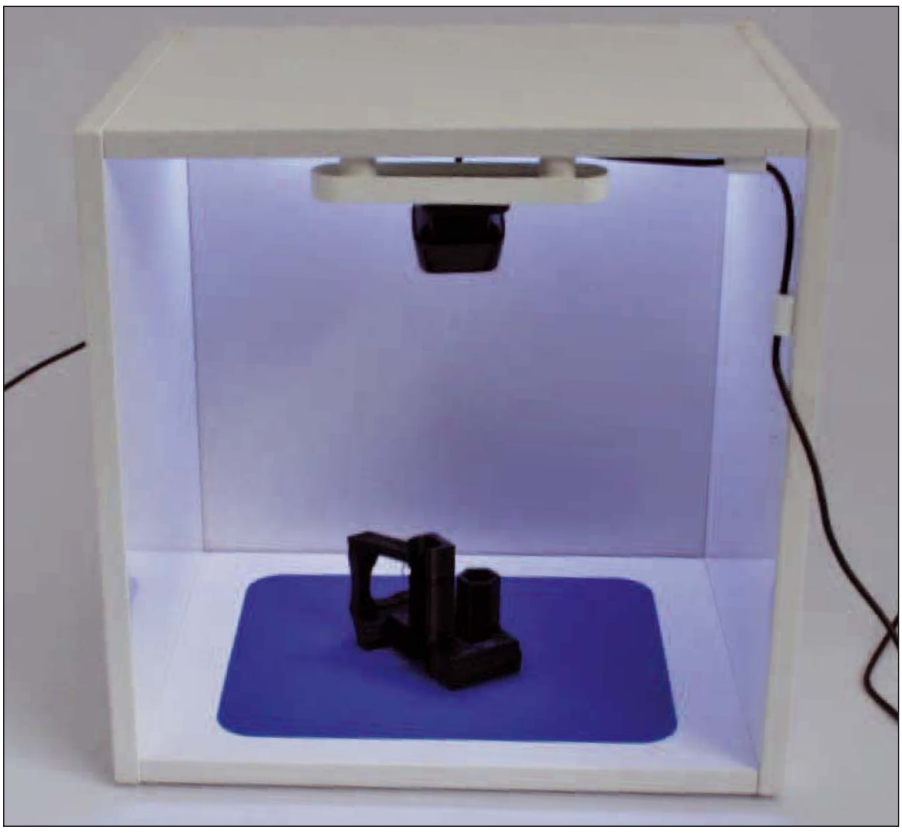
Ikea questi piccoli supporti Bygel da 120 mm, estremamente potenti dal punto di vista magnetico, e con una certa sorpresa abbiamo scoperto che il piede in gomma della Webcam era in realtà una struttura metallica snodata, rivestita in gomma.
Piazzando il supporto magnetico sul bordo superiore, il "piede" in gomma (flessibile e semirigido) può fissarsi ad esso sostenendo egregiamente il ridotto peso della webcam. Come notate dalle immagini, orientare e inclinare a piacimento l'inquadratura è semplicissimo, mentre non ci sono ostacoli e fili a complicare qualsiasi manovra sul piano di lavoro/ripresa.
Al posto della specifica barretta magnetica Bygel Ikea si può usare qualsiasi altra soluzione che è composta da magneti e da un supporto per fissarli. I magneti al Samario Cobalto o comunque non del tipo gommoso a bassa intensità sono quelli da scegliere, per evitare che un semplice urto faccia cadere la webcam sugli oggetti.
Durante l'uso abbiamo anche notato che la struttura semirigida entra facilmente in oscillazione se si danno anche piccoli urti al mobiletto. Il problema è facilmente risolvibile aggiungendo un tassello di gommapiuma fra l'estremità della webcam e la parete superiore. Non importa che faccia una particolare pressione in quanto ha il solo scopo di smorzare e impedire le eventuali oscillazioni di una struttura altrimenti "appesa".
La striscia di LED, infine è l'ultimo elemento significativo del progetto. Grazie alla sua forma risolve il problema di una illuminazione stabile e diffusa. Una singola lampadina o una placchetta di LED concentrati avrebbe creato delle ombre sugli oggetti, mentre la strip unita alle pareti bianche ha ridotto sensibilmente il problema. In più l'ingombro è ridotto, non produce calore ed è anche già dotata di autoadesivo per un pratico fissaggio.

L'ASSEMBLAGGIO

Se vi siete procurati tutto il necessario, vi aspettano circa 30 minuti per l'intero processo di montaggio.
Si comincia con la predisposizione del materiale, ovvero lo spacchettamento del pensile dall'imballo per individuare i due pannelli laterali, con i 4 fori per le 4 viti, i due pannelli superiore e inferiore, più piccoli, e quello più leggero, da inserire come fondo.
Sul pannello superiore, nella parte frontale dovremo fissare il sistema magnetico, mentre in quella posteriore – verso la scanalatura che blocca il fondo – dovremo sistemare la striscia adesiva LED.
Prendete le misure per fare l'invito per le due viti di fissaggio del blocchetto magnetico (l'invito è un piccolo forellino che guida la vite nel legno), quindi scegliete delle viti che non siano troppo lunghe. Lo spessore del pannello è di soli 12 millimetri, mentre Bygel ha dei distanziali e il sup-



Il cavetto bipolare con cui alimentiamo la strip LED va fatto uscire sul retro del pensile praticando una tacca delle giuste dimensioni nel pannello posteriore in masonite, possibilmente vicino a un angolo.



Un tappetino colorato, in silicone sottile, per mouse si è rivelato ottimale per definire l'area di ripresa e per evitare che il legno laccato bianco abbagli la webcam. Evita anche che gli oggetti scivolino.

porto che totalizzano 13 mm. La vite per legno non dovrà quindi essere superiore a 20/22mm. In commercio potete trovare viti che hanno una punta tronca che offrono buona tenuta penetrando il materiale per una lunghezza limitata; quando siete certi che la vite abbia fatto ben presa e il supporto è fissato, non insistete. Fate anche attenzione a quando monterete il magnete sulla parte metallica: l'attrazione è molto forte ed è facile pizzicarsi le dita.



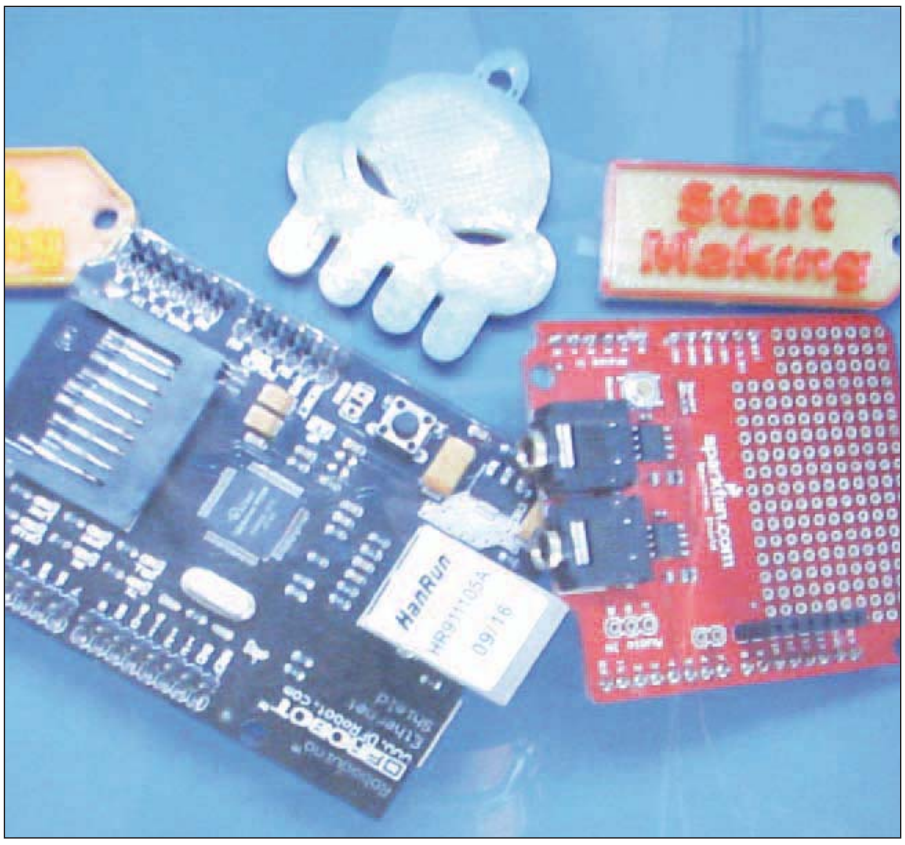
Da questa angolazione si nota come la strip sia in fondo al box e la webcam si posizioni verso il centro dell'area grazie all'aggancio magnetico.

Con il pannello superiore ancora smontato, va posizionata la striscia di led bianchi, effettuando il cablaggio del filo di alimentazione in modo che possa essere fatto passare in una tacca che creerete nel supporto posteriore, vicino al bordo. Noi abbiamo aggiunto una femmina volante così da poter attaccare e staccare tutto, lasciando una situazione pulita e senza fili da gestire. Ricordate che la striscia va alimentata a 12 volt con un alimentatore possibilmente switching o ben rettificato e filtrato, altrimenti potrebbe essere visibile nelle riprese un fastidioso sfarfallio.
Montate la base con il primo pannello laterale e serrate le due viti a brugola, poi fissate il pannello dall'altra parte e infilate il fondo, quindi appoggiate la parte superiore, ora dotata di blocchetto magnetico e strip LED per individuare il punto in cui fare la tacca per il cavo di alimentazione dei LED. Fate caso a quanto pannello entra nella scanalatura superiore perché quel valore va sommato a circa 4 millimetri di agio per il cavo elettrico. Se usate una lima per creare la tacca, potete provare finché il cavo passa, senza essere né schiacciato, né muoversi nella tacca.
Il passaggio successivo riguarda il fissaggio con le 4 viti a brugola del pannello superiore. Serrate nuovamente anche le viti inferiori e verificate che non ci siano giochi fra i pannelli.

I RITOCCHI

Ora che la struttura base è a posto, con il pensile ultimato e dotato di supporto magnetico e strip LED, possiamo passare alla fase dei ritocchi per adattare la funzionalità dell'oggetto alle nostre esigenze. Nel nostro caso, ad esempio, dovendo gestire degli shield di Arduino e delle parti

meccaniche anche molto piccole, abbiamo pensato di rendere la base su cui appoggiare gli oggetti più morbida e antiscivolo con un sottile tappetino per mouse, in silicone blu. Oltre a rendere più gradevole l'immagine ripresa, questo ci permette di non far scivolare le schede sulla superficie laccata del pensile, altrimenti poco aderente.
Il colore del tappetino potrebbe anche essere chiaro o bianco, ma ricordate che la webcam cerca di regolare la luminosità e quindi un fondo chiaro potrebbe abbagliarla, lasciando scuri tutti gli oggetti ripresi.
La Webcam LifeCam, inoltre, ha un ottimo sistema di regolazione del colore e della luminosità e dobbiamo ammettere che le riprese ottenute sono veramente molto fedeli.
Tornando al tappetino, questo aiuta anche ad avere un'area delimitata nella quale sappiamo che c'è l'inquadratura. Posizionando opportunamente webcam e tappetino si crea infatti dello spazio non ripreso in cui si possono parcheggiare dei pezzettini o degli oggetti senza che intralcino le riprese.
Tenete presente che chi guarda il video non va distratto con oggetti che non sono rilevanti e il poterli nascondere nel box è sicuramente un'opportunità interessante.
Tornando ai ritocchi, se notate che la vostra LifeCam sente molto le vibrazioni quando muovete gli oggetti o appoggiate le cose, è necessario smorzare l'oscillazione con un pezzetto di gommapiuma. Va molto bene quella delle spugnette per i piatti sempre di Ikea: il pacco da 6 costa un paio di euro e ha un facile distacco della parte più rigida e abrasiva da quella puramente spugnosa. Ritagliate un rettangolo che sia di un paio di centimetri più spesso del necessario per inserirlo fra la parte superiore e il dorso della webcam. Potete anche intagliare la gommapiuma per ospitare il cavetto USB, così da tenerla salda in posizione.
Ultimo ritocco che consigliamo è l'aggiunta di quattro feltrini antiscivolo sui quattro angoli inferiori del pensile: faranno da piedini, smorzeranno le vibrazioni e renderanno anche più semplice il posizionamento del pensile su un tavolo.



Il nostro box, grazie alle caratteristiche di fascia alta della LifeCam HD-5000, produce immagini ad alta risoluzione in cui si leggono anche le scritte più minute.

L'USO

La webcam LifeCam HD-5000 viene fornita da Microsoft con un buon assortimento di accessori software, utili per divertirsi online con Skype e con gli altri applicativi di video comunicazione. Per i nostri scopi abbiamo invece bisogno di un semplice programma che accede alla webcam e mette sullo schermo quanto viene ripreso. In Windows si chiama Amcap.exe ed è anche facilmente reperibile sia come versione originale rilasciata da Microsoft, sia come utility shareware con prestazioni migliorate. A voi scegliere quello che preferite.
Con Amcap potete impostare tutti i parametri della webcam, dalla risoluzione all'autofocus, al bilanciamento del colore al numero di frame al secondo, quindi potete lanciare l'applicazione a schermo intero per avere su un videoproiettore – come abbiamo fatto noi per il nostro workshop – una ottima ripresa video, quasi full HD e sicuramente in grado di sfruttare la definizione dei più recenti videoproiettori.
In alternativa, con questo box potete realizzare ottimi video in cui spiegare il montaggio o il funzionamento di oggetti anche di una certa dimensione. Ideale per dimostrare telefonini e tablet.



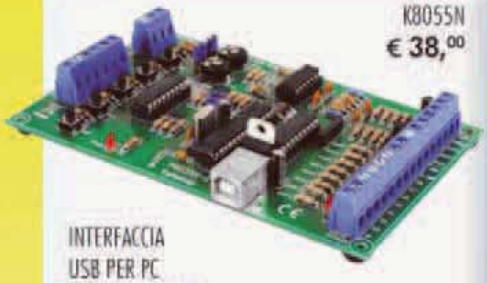
FT908M-GPS
€ 128,00
LOCALIZZATORE MINIATURA GSM/GPS - MONTATO



GSMGPRSSHIELDV2
€ 19,50
GSM/GPRS & GPS SHIELD V2 PER ARDUINO



DSOKIT
€ 69,00
OSCILLOSCOPIO DIGITALE



K8055N
€ 38,00
INTERFACCIA USB PER PC CON DEMOBOARD

TOP KIT

by FUTURA ELETTRONICA

Diventa anche tu esperto di elettronica con le nostre scatole di montaggio e i nostri moduli. Metti in "pratica" le tue conoscenze di programmazione!



FT849K
€ 84,00
TESTER EOBD, UNITÀ DI DIAGNOSI PER AUTO



MODULO JUKEBOX MP3 - MONTATO
VM202
€ 19,90



FT829K
€ 45,00
SERRATURA A TRANSPONDER STAND-ALONE



FT953K
€ 29,00
RICEVITORE 433MHz 4 CANALI AD AUTOAPPRENDIMENTO



MK190
€ 14,00
AMPLIFICATORE STEREO 5W PER LETTORI MP3



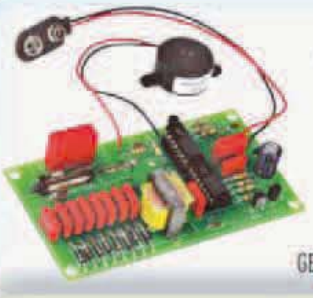
MK174
€ 19,00
REGISTRATORE/ RIPRODUTTORE VOCALE A SINGOLO MESSAGGIO



K8071
€ 9,00
DRIVER PER LED DA 1W/3W



FT972K
€ 5,50
INTERRUTTORE CREPUSCOLARE MINIATURA



K2645
€ 209,00
CONTATORE GEIGER-MULLER

Prezzi IVA inclusa.

www.futurashop.it

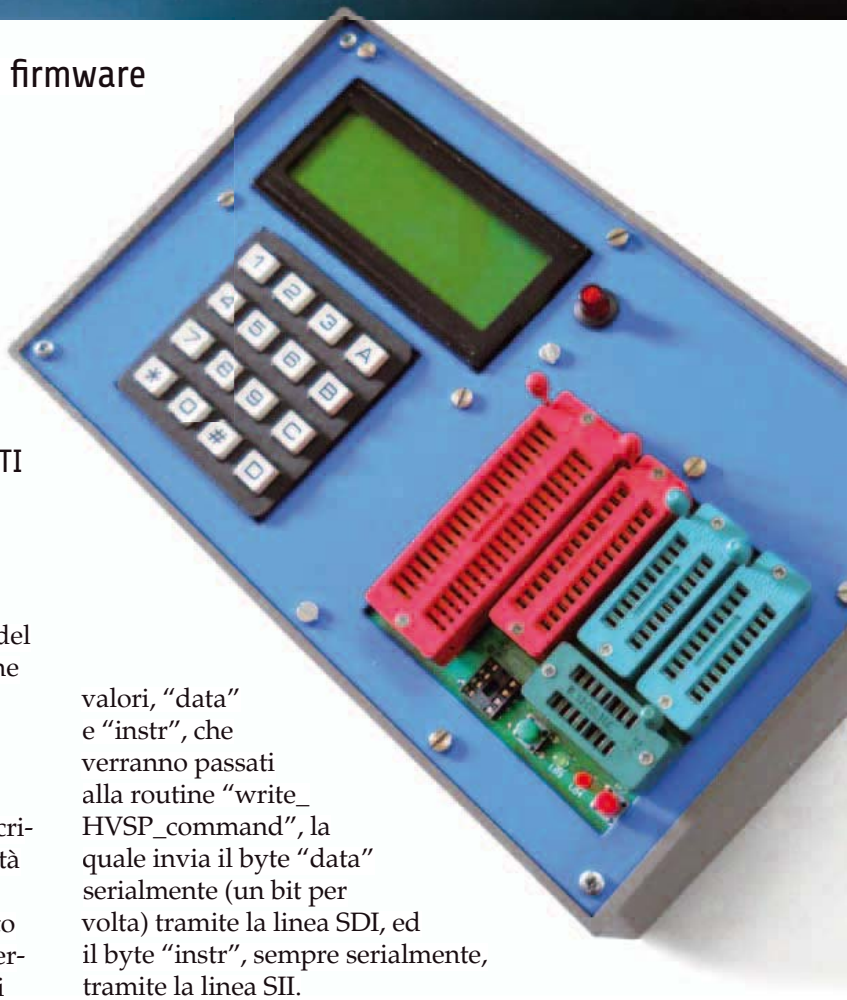
FUTURA ELETTRONICA idee in elettronica
Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775 • Fax. 0331/792287

Maggiori informazioni e schede tecniche dettagliate di tutti i prodotti sono disponibili su www.futurashop.it dove è possibile effettuare anche acquisti on-line.



UN PROGRAMMATORE HV MODULARE PER MCU Atmel

Dopo avere visto l'organizzazione del firmware per il nostro Programmatore HV approfondiamo la logica del programma e le routine che lo compongono, facendo qualche prova d'uso. Quarta e ultima puntata.



del prof. MICHELE MENNITI

Completata la descrizione della sezione hardware del Programmatore HV terminiamo questa serie di appuntamenti con la descrizione del programma di utilizzo sia in modalità stand-alone che collegato ad un PC.

STUDIO DI ALCUNE ROUTINE DEL FIRMWARE DEL PROGRAMMATORE HV

Iniziamo a vedere la procedura che permette di scrivere i fuse di default in un micro target in modalità HVSP (prendiamo ad esempio l'ATtiny85). Una volta inserito il chip nel relativo zoccolo e premuto il pulsante "GO", la procedura è in grado di leggerne automaticamente la signature e di tirarne fuori i relativi dati dalla tabella, quindi sono già noti i valori dei fuse. Alla comparsa del menu si seleziona la voce "1: Ripristino fuse default" ed il programma esegue la routine "fuses_write_HVSP". In questa routine sono presenti tre serie di operazioni, ognuna costituita da cinque istruzioni, per la scrittura dei tre fuse. Ogni istruzione richiede l'inserimento di due valori, "data" e "instr", che verranno passati alla routine "write_HVSP_command", la quale invia il byte "data" serialmente (un bit per volta) tramite la linea SDI, ed il byte "instr", sempre serialmente, tramite la linea SII.

Vediamo il procedimento per il Low Fuse Byte. Per seguire i vari passaggi di programmazione HVSP vi consigliamo di aprire il firmware con l'IDE in modo da avere una visione globale dello stesso. Troverete utilissimo anche il riepilogo, chiamato Listato 1, che vi abbiamo presentato nel numero scorso, per l'immediata reperibilità delle varie routine.

- La prima istruzione:
  write_HVSP_command(HVSP_WRITE_LFUSE_DATA, HVSP_WRITE_LFUSE_INSTR1)
  passa i due byte "data"*(HVSP_WRITE_LFUSE_DATA)* e "instr"*(HVSP_WRITE_LFUSE_INSTR1)*, i cui valori abbiamo definito nella relativa sezione iniziale del firmware, sulla base delle informazioni di programmazione trovate nel data-sheet del micro (v. riferimenti alla fine della puntata precedente, pag. 164 Table 20-16). In pratica l'invio del byte "data" (B01000000) e del byte "instr" (B01001100) istruisce il micro per un'operazione di scrittura di fuse; potete notare come il valore di questa coppia di byte sia identico per i tre fuse;
- La seconda istruzione:
  write_HVSP_command(lfuse, HVSP_WRITE_LFUSE_INSTR2) passa i due byte "data"(lfuse) e "instr"(HVSP_WRITE_LFUSE_INSTR2) e serve per inviare al micro il byte del fuse da scrivere;
- La terza istruzione:
  write_HVSP_command(0x00, HVSP_WRITE_LFUSE_INSTR3) passa i due byte "data"*(0x00)* e "instr"*(HVSP_WRITE_LFUSE_INSTR3)* e serve per dire al micro che il byte inviato è il Low Fuse Byte;
- La quarta istruzione:
  write_HVSP_command(0x00, HVSP_WRITE_LFUSE_INSTR4) passa i due byte "data"*(0x00)* e "instr"*(HVSP_WRITE_LFUSE_INSTR4)* e serve per dire al micro che l'operazione di invio e scrittura del Low Fuse Byte è conclusa.
- La quinta ed ultima istruzione:
  while(digitalRead(SDO) == LOW) ferma l'esecuzione delle operazioni in attesa che la linea SDO vada su HIGH, segno che l'operazione di scrittura del Low Fuse Byte è effettivamente conclusa.

A questo punto il programma riprende e, con le stesse modalità appena viste per il Low Fuse Byte, programma anche l'High Fuse Byte e l'Extended Fuse Byte. In quest'ultimo caso vi facciamo notare che l'operazione viene eseguita solo se il valore del flag BURN_EFUSE è settato su 1, segno che il micro target possiede anche questo fuse, diversamente questa serie di istruzioni viene saltata.
Subito dopo viene eseguita la routine "fuses_read_HVSP" che, con le stesse modalità viste per la scrittura, esegue tre serie di tre istruzioni ciascuna, sempre con l'invio di byte "data" e "instr" alla routine "read_HVSP_command", per effettuare la rilettura dei valori appena scritti. Infine viene eseguita la routine "current_fuses_display" in modo da farvi controllare che l'operazione sia andata a buon fine, mostrandovi i valori appena riletti dal micro.

Vediamo ora invece come avviene la programmazione dei fuse in ambiente HVPP (prendiamo ad esempio l'ATmega328P). Anche questa volta, dopo l'avvio della procedura, disponiamo in automatico dei valori di default dei fuse del micro target. Effettuata la scelta "1: Ripristino fuse default" il programma esegue la routine "fuses_write_HVPP". Come si può notare, in questo caso le istruzioni sono soltanto una per ogni fuse, a motivo del fatto che la programmazione HVPP opera in parallelo, cioè inviando un byte per volta tramite le linee DATA. Vediamo il procedimento per l'High Fuse Byte.
Il comando *F_write_HVPP_command(hfuse, HFUSE_SEL)* passa alla routine "F_write_HVPP_command" i due valori "fuse" *(hfuse)* e "select" *(HFUSE_SEL)*; il primo è un byte, il secondo un valore numerico che servirà per identificare il tipo di fuse da programmare.
Questa routine inizia col trasmettere al micro target il byte "B01000000" mediante l'ulteriore routine "send_cmd_HVPP", ponendolo sulle linee DATA (*PORTD = command*), e attivando queste ultime come uscite (*DDRD = 0xFF*); tale byte ha il significato di iniziare un'operazione di scrittura dei fuse (come abbiamo già visto per la modalità HVSP).
Subito dopo sulle linee DATA viene posto il byte relativo al fuse da programmare (*PORTD = fuse*) ed anche in questo caso l'invio avviene impostando tali linee come uscite (*DDRD = 0xFF*).
Nel passaggio successivo vengono settate le due linee BS1 e BS2, la cui combinazione HIGH/LOW serve per stabilire qual è il fuse che si vuole scrivere; in pratica abbiamo:
Programmazione dell'High Fuse Byte: BS1 = HIGH e BS2 = LOW;
Programmazione del Low Fuse Byte: BS1 = LOW e BS2 = LOW;
Programmazione dell'Extended Fuse Byte: BS1 = LOW e BS2 = HIGH.
Successivamente avviene la scrittura vera e propria mediante un impulso LOW sulla linea "WR".
Infine, con l'istruzione while(digitalRead(RDY) == LOW) il programma resta in attesa che la linea RDY passi su HIGH, ad indicazione che la procedura è conclusa.
Finita l'operazione di scrittura dell'High Fuse Byte il programma procede con lo stesso metodo per scrivere anche il Low Fuse Byte e l'Extended Fuse Byte (se previsto).
Quindi viene eseguita la routine "fuses_read_HVPP" che semplicemente invia al micro target il byte di comando "B00000100" (operazione di lettura dei fuse), poi imposta le linee DATA come ingressi (*DDRD = 0x00*) ed infine setta le linee BS1 e BS2 in base a quale fuse si sta leggendo; in questo caso abbiamo:
Lettura dell'High Fuse Byte: BS1 = HIGH e BS2 = HIGH;

Lettura del Low Fuse Byte: BS1 = LOW e BS2 = LOW;
Lettura dell'Extended Fuse Byte: BS1 = LOW e BS2 = HIGH.
Naturalmente, anche in questo caso la routine "current_fuses_display" mostra a video i valori dei fuse appena riletti, per l'opportuna verifica.
Vediamo ora la procedura di lettura della signature; in questo caso, bisogna leggere tre byte per avere un solo valore. Iniziamo dalla lettura della signature in modalità HVSP. Come già spiegato questa operazione viene eseguita in automatico dal programma, quindi non esiste una voce di menu.
Subito dopo l'avvio della fase di programmazione (routine "begin_programmer") viene eseguita la routine "signature_read_HVSP"; come abbiamo già spiegato, la lettura della signature senza conoscere a priori il micro target, conviene farla in modalità HVSP.
Tale routine è costituita da tre serie di istruzioni, una per ogni byte della signature.
Come per le operazioni descritte in precedenza, anche qui la modalità HVSP richiede, per ogni istruzione, l'inserimento di due valori, "data" e "instr", che verranno passati alla routine "read_HVSP_command", la quale invia il byte "data" serialmente (un bit per volta) tramite la linea SDI, ed il byte "instr", sempre serialmente, tramite la linea SII. In questo caso, poiché si tratta di un'operazione di lettura, il micro target risponderà al comando fornendo sulla linea SDO il byte di risposta, sempre inviandolo serialmente.
Vediamo ora tutto il procedimento di lettura.

- La prima istruzione: *read_HVSP_command(SIGN_READ_DATA, SIGN_READ_INSTR1)* passa i due byte "data"*(SIGN_READ_DATA)* e "instr"*(SIGN_READ_INSTR1)*; anche questi valori sono stati definiti nella relativa sezione iniziale del firmware, sulla base delle informazioni di programmazione trovate nel data-sheet del micro. In pratica l'invio del byte "data" (B00001000) e del byte "instr" (B01001100) istruisce il micro per un'operazione di lettura della signature; questa istruzione viene inviata una volta soltanto; poi, per ogni byte della signature, vengono eseguite ulteriori tre istruzioni;
- La seconda istruzione: *read_HVSP_command(SIGN_READ_ADDR1, SIGN_READ_INSTR2)* passa i due byte "data"*(SIGN_READ_ADDR1)* e "instr"*(SIGN_READ_INSTR2)*; in questo caso viene inviato l'indirizzo del primo byte da leggere;
- La terza istruzione: *read_HVSP_command(0x00, SIGN_READ_INSTR3)* passa i due byte "data"*(0x00)* e "instr"*(SIGN_READ_INSTR3)*, chiudendo il processo di richiesta;
- La quarta istruzione: *read_signature1=read_HVSP_command(0x00, SIGN_READ_INSTR4)* passa i due byte "data"*(0x00)* e "instr"*(SIGN_READ_INSTR4)*, che restituiscono sulla linea SDO il valore del byte, che a sua volta viene assegnato alla variabile "read_signature1".

La seconda, terza e quarta istruzione vengono poi ripetute due volte cambiando esclusivamente il valore di indirizzo nella seconda istruzione (in pratica *SIGN_READ_ADDR2 e SIGN_READ_ADDR3*) in modo da chiedere al micro target di mandare gli altri due valori dei byte della signature, che saranno assegnati rispettivamente a *"read_signature2"* e *"read_signature3"*.
Una volta ottenuti i tre byte, essi vengono sottoposti al controllo della routine "signature_check", che semplicemente verifica se il primo byte è "1E" (lettura valida) o "FF" (il micro target richiede modalità di programmazione HVPP) o "00" (il micro target è danneggiato o non è inserito nello zoccolo corretto o non è presente). Nel primo e nel terzo caso la routine "signature_display" mostra a video il valore della signature o un messaggio d'errore, nel secondo caso

IS2T, la piattaforma Java per l'STM32



Grazie alle nuove tecnologie di ST i dispositivi computerizzati come gli elettrodomestici, le attrezzature per l'ufficio ed i controlli industriali potranno presentarsi con interfacce grafiche accattivanti, complete di animazioni come quelle presenti sugli smartphone. Un motivo che ha permesso questa evoluzione è la disponibilità di ambienti di progettazione e sviluppo software in tecnologia Object-Oriented Programming (OOP) quali Java.
Lo starter kit STM3220G-JAVA contiene una versione di valutazione del Software Development Kit (SDK) MicroEJ® di IS2T e la scheda di valutazione della serie di microcontrollori IS2T, con l'intento di fornire ai progettisti in un'unica confezione tutto quanto necessario per iniziare a progettare.
GeeseWare contribuisce al progetto con la sua soluzione integrata hardware e software, che comprende il MicroEJ SDK di IS2T e una scheda di sviluppo integrata che supporta sia i microcontrollori STM32F2 che STM32F4.
www.is2t.com

invece si passa automaticamente alla rilettura della signature in modalità HVPP.
A questo punto viene eseguita la routine "signature_read_HVPP", che è costituita da tre sole istruzioni, una per ogni byte da leggere, le quali passano alla routine "S_read_HVPP_command" i rispettivi valori numerici (SIGN_SEL1, SIGN_SEL2 e SIGN_SEL3), che serviranno per gestire la sequenza di indirizzi da inviare al micro target. Infatti tale routine esegue il comando send_cmd_HVPP(B00001000) che avvia la procedura di lettura della signature del micro target (similmente a quanto visto a proposito della scrittura dei fuse in HVPP). Subito dopo imposta le linee DATA ponendole come ingressi per la lettura, infine esegue la routine "send_address_HVPP" che si occupa di inviare sequenzialmente i tre indirizzi dei byte (0x00, 0x01, 0x02) al micro target, ottenendo in risposta i valori dei rispettivi byte della signature, che vengono assegnati alle solite variabili *"read_signature1"*, *"read_signature2"* e *"read_signature3"*. Anche in questo caso, dopo opportuna verifica, i valori dei tre byte della signature vengono mostrati a video. Giunti a questo punto possiamo dire di aver concluso lo studio del firmware.
Vi ricordiamo che nella sezione download del nostro sito *www.elettronicain.it* (relativa al numero 167) potete trovare le due diverse versioni del firmware: "HV_Master_for_PC" e "HV_Master_for_LCD"; la prima vi servirà se avete deciso di realizzare la versione del Programmatore HV collegata al PC, la seconda nel caso in cui invece abbiate realizzato la versione stand-alone, con display LCD e tastierino esadecimale. Per quanto riguarda le funzionalità i due firmware sono assolutamente uguali. Il firmware scelto andrà caricato sul microcontrollore ATmega328P che poi sarà installato sulla scheda Master; ricordiamo che tale microcontrollore dovrà avere il bootloader di Arduino Duemilanove, per le ragioni già spiegate nelle puntate precedenti.

IL PROGRAMMA FUSECALC

Vi presentiamo ora un sito internet che mette a disposizione un comodissimo programma per il calcolo dei fuse dei numerosi microcontrollori ATMEL; si chiama "FuseCalc", e viene eseguito automaticamente all'indirizzo *http://www.engbedded.com/fusecalc*.
In **Fig. 1** potete vedere la maschera iniziale, nella quale stiamo selezionando l'MCU ATmega328P dal lunghissimo elenco, visibile parzialmente, in quanto allo stato attuale i modelli presenti sono ben 141.
Subito dopo la selezione appare una nuova maschera che rappresenta tutte le impostazioni di default dei singoli bit dei tre Fuse della MCU (**Fig. 2**).
In genere, quando si seleziona un qualsiasi microcontrollore, il FuseCalc presenta le impostazioni di default, cioè quelle con cui il chip esce dalla fabbrica; questo però non sempre è vero, quindi a volte bisogna andarsi a cercare nell'elenco la combinazione che all'estrema destra riporta la dicitura "default value". Addirittura in alcuni casi questa dicitura non c'è quindi per risalire alle impostazioni di default bisogna far

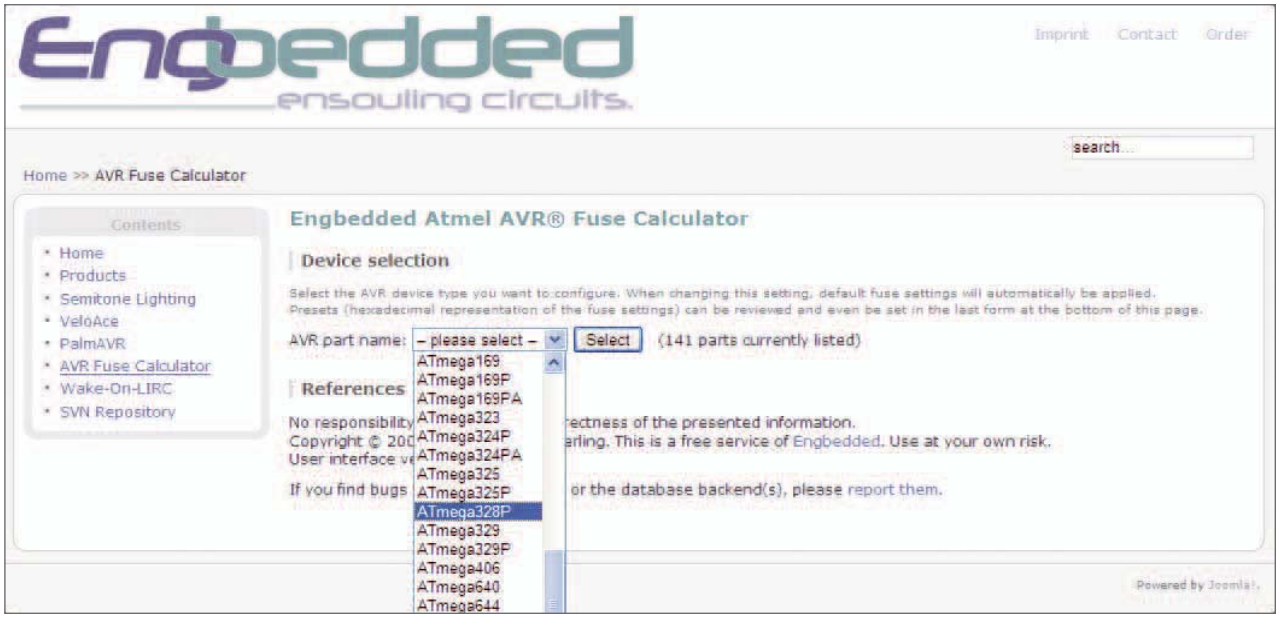


Fig. 1 *- La maschera iniziale del programma Fusecalc.*

riferimento al data-sheet, vedremo più avanti come fare.
La maschera si presenta divisa in tre sezioni:

Feature configuration: in questa sezione vengono elencate le caratteristiche tecniche impostabili via fuse, in forma descrittiva; in base alla combinazione selezionata variano i flag sui singoli campi e di conseguenza variano i valori dei tre fuse, che vengono ricalcolati istantaneamente e mostrati nella sezione "Current settings". La combinazione prescelta descrive le caratteristiche fondamentali; vediamo l'esempio della **Fig. 2**, che ci dà le seguenti informazioni:
Il chip è settato per lavorare con l'oscillatore interno a 8MHz; quindi i quattro bit **CKSEL3--0** del Low Fuse Byte (v. spiegazione sul numero 167) sono settati su 0010.
Il riavvio dopo uno sleep di tipo pwrdown o un RESET prevede 6+14 cicli di clock ed un ritardo di 65msec; quindi i bit **SUT1-0** del Low Fuse Byte sono settati su 10.
Nel rigo successivo si vede che il bit **CKOUT** del Low Fuse Byte, la cui funzione è quella di riportare il clock sul pin PB0 del chip (il 14 nel caso dell'ATmega328P), è disabilitato.
Invece il bit **CKDIV8** del Low Fuse Byte, la cui funzione è quella di dividere il clock per 8, è abilitato, per cui l'oscillatore interno in realtà lavora a 8MHz/8 = 1MHz (che poi è l'impostazione di fabbrica di tutti i micro ATMEL).
Il bit **BOOTRST** dell'High Fuse Byte è disabilitato; la sua funzione è quella di settare il puntamento del vettore di reset. Se questo bit è disabilitato il vettore punta alla sezione Application della memoria Flash (0x0000); in caso contrario punta alla sezione Bootloader, con indirizzo dipendente dalla memoria riservata a questa funzione.

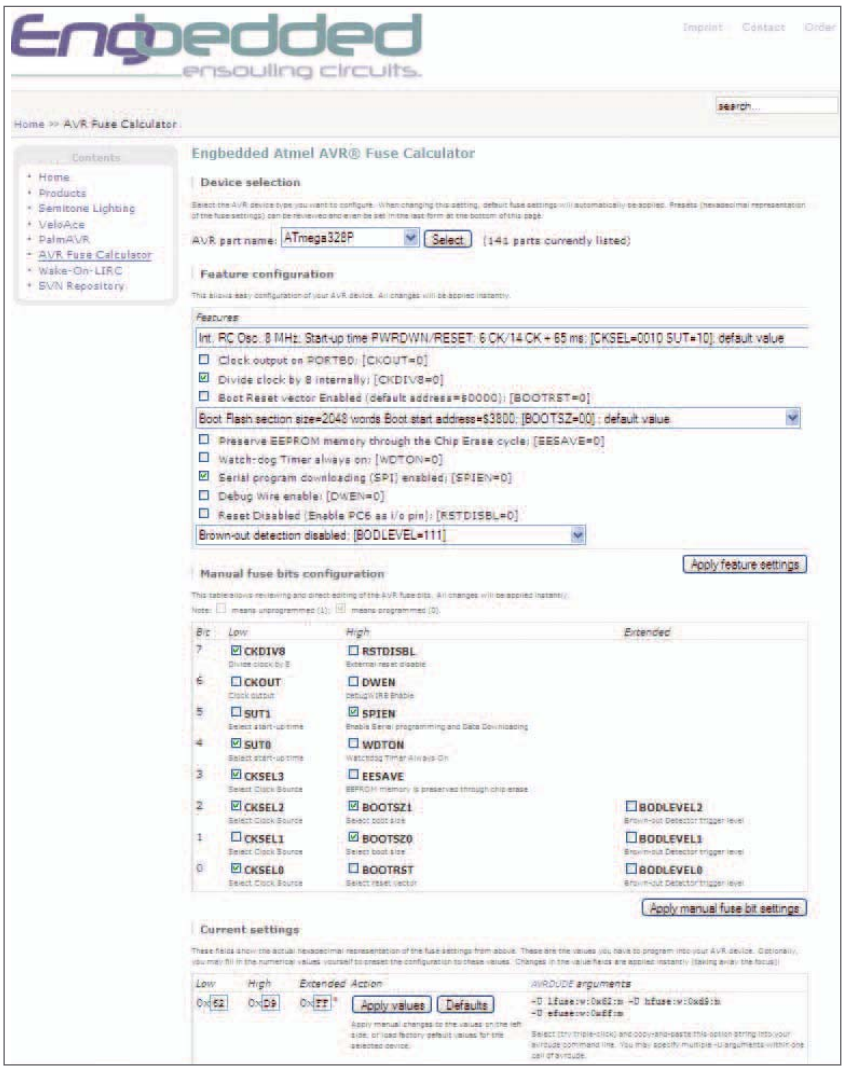


Fig. 2 *- I valori di default dei Fuse dell'ATmega328P.*

I bit **BOOTSZ1-0** dell'High Fuse Byte sono settati su 00, cioè per riservare un'area di 2048 word (4KB) al bootloader, ma di fatto vengono ignorati in quanto il bit BOOTRST è disabilitato. Altre combinazioni possibili sono: 11 (512B), 10 (1KB), 01 (2KB).
Il bit **EESAVE** dell'High Fuse Byte, la cui funzione è quella di preservare l'area di memoria della EEPROM interna del micro dalle operazioni di cancellazione del "Chip Erase", è disabilitato, quindi la memoria viene cancellata ad ogni Chip Erase dato al micro.
Il bit **WDTON** dell'High Fuse Byte (attivazione del Watch Dog Timer) è disabilitato.
Il bit **SPIEN** dell'High Fuse Byte è abilitato, quindi la programmazione seriale del micro è attiva.
Il bit **DWEN** dell'High Fuse Byte, che serve per attivare il Debug Wire, è disabilitato.
Il bit **RSTDISBL** dell'High Fuse Byte, la cui funzione è quella di disattivare il reset del chip, è disabilitato.
L'ultima voce, presente sotto forma di menu a discesa, serve per il settaggio dei tre bit **BODLEVEL2--0,** gli unici dell'Extended Fuse Byte; la funzione di questi bit è quella di settare una tensione minima di alimentazione, al di sotto della quale il micro viene messo nella condizione fissa di reset. Di default sono disabilitati (111 = BOD Disabled).

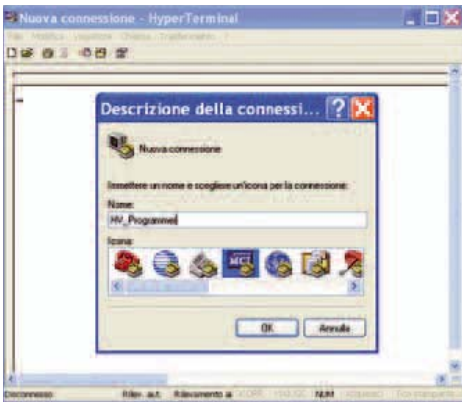


Fig. 3 *- L'emulatore di terminale seriale Windows HyperTerminal.*

Manual fuse bit configuration: in questa sezione tutti i bit dei tre Fuse Byte del micro sono visibili e settabili singolarmente; sono disposti su tre colonne, in base al Fuse Byte di appartenenza ed è sufficiente aggiungere o togliere il flag alla relativa casella per attivarli o disattivarli singolarmente. Questa modalità d'uso manuale richiede naturalmente una perfetta conoscenza dei singoli bit e delle loro combinazioni, in caso contrario si rischia di creare delle combinazioni di Fuse pericolose per il micro. Infatti è noto che impostare un micro ATMEL con una combinazione di Fuse non corretta significa bloccarlo (in gergo tecnico si parla di condizione "bricked") irrimediabilmente. Solo la programmazione HV a quel punto sarà in grado di rimettere a posto le cose. E' evidente che tale condizione si può avere anche con l'uso della "Feature configuration", ma in quel caso le descrizioni estese dovrebbero aiutare a capire più facilmente cosa si sta settando, ecco perché ai meno esperti consigliamo di iniziare a cimentarsi con la "Feature configuration". La modifica di ogni singola casella bit determina l'immediata variazione dei valori dei Fuse nella sezione "Current settings".

Current settings: questa sezione presenta i tre valori dei Fuse Byte risultanti dalle impostazioni fatte in precedenza in una delle due sezioni precedenti. Si è potuto notare, appena selezionata la combinazione di default, che i valori dei tre fuse sono: Low = 62, High = D9, Extended = FF. Questi valori non sono altro che l'espressione esadecimale delle combinazioni dei relativi bit.
Vediamo:
Il Low Fuse Byte è costituito dai bit (partendo dal più significativo): CKDIV8, CKOUT, SUT1, SUT0, CKSEL3, CKSEL2, CKSEL1, CKSEL0, che sono impostati sulla combinazione 01100010, byte che, espresso in esadecimale, vale appunto "62".
L'High Fuse Byte è costituito dai bit (partendo dal più significativo): RSTDISBL, DWEN, SPIEN, WDTON, EESAVE, CKSEL2, BOOTSZ1, BOOTSZ0, che sono impostati sulla combinazione 11011001, byte che, espresso in esadecimale, vale appunto "D9".
L'Extended Fuse Byte usa solo i tre bit meno significativi: BODLEVEL2, BODLEVEL1, BODLEVEL0, i bit dal 7 al 3 sono tutti settati a 1 (disabled), quindi il byte risultante è: 11111111, cioè l'esadecimale "FF".
Una particolarità vantaggiosa di questa sezione consiste nel fatto che è possibile effettuare il percorso inverso, cioè inserire i tre valori dei Fuse per risalire alla relativa combinazione dei bit. Questa opzione è utile in quanto potremmo trovarci tra le mani un chip già programmato; il nostro Programmatore HV ha la possibilità di leggere subito il valore corrente dei tre Fuse che, inseriti nei relativi campi, determineranno appunto i settaggi dei bit, che saranno visibili nelle due sezioni precedenti. Come accennavamo prima, il FuseCalc non sempre presenta una combinazione di default per il micro prescelto, la si può ricavare leggendo i valori dei fuse mediante il Programmatore HV ed inserendoli nei

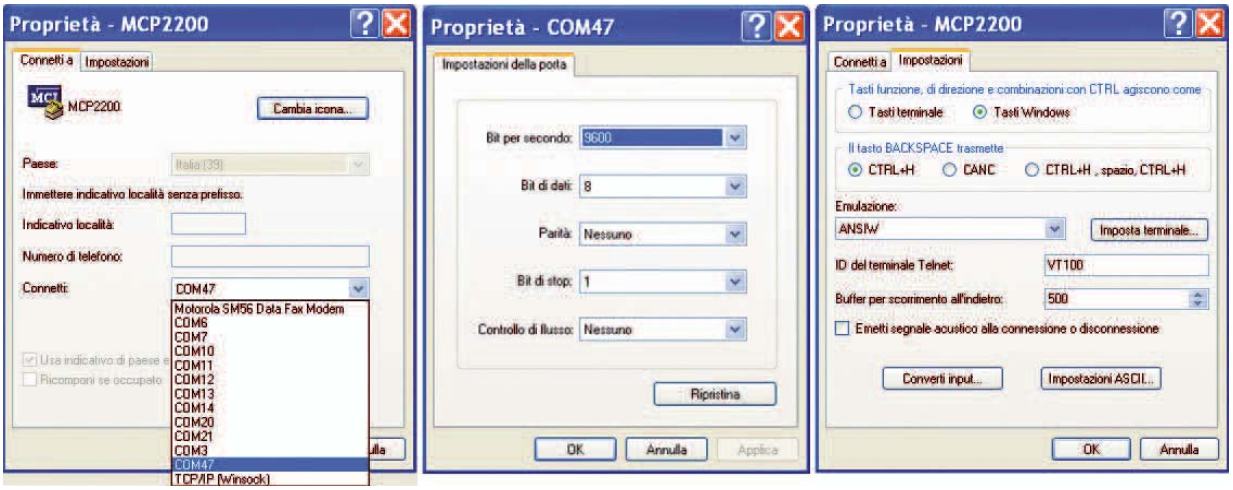


Fig. 4
Le impostazioni di Windows HyperTerminal.

relativi campi. Ma qualora siate nel dubbio, la cosa più sicura da fare è reperire sul sito *www.atmel.com* il data-sheet (è un file pdf) del micro target e cercare il capitolo denominato "Memory Programming", all'interno del quale troverete le tabelle dei fuse con i valori di default.
Viene facile comprendere come, con questo semplice ma potente strumento, potrete calcolare i settaggi dei vostri micro con facilità e rapidità.

L'IMPOSTAZIONE DI HYPERTERMINAL

La versione "PC" del firmware richiede ovviamente il collegamento del Programmatore HV ad un computer, inoltre è necessario un programma di emulazione terminale seriale. Tale programma può essere benissimo rappresentato dal "Serial Monitor" dell'IDE di Arduino, ma siete costretti ad aprire l'IDE, ecco perché abbiamo pensato di usare il comunissimo HyperTerminal di Windows, programma in dotazione a tutte le versioni di Windows e quindi gratuito, inoltre ben si presta a gestire il nostro strumento.
Il programma si trova nella sezione Programmi – Accessori – Comunicazioni e, una volta aperto, dovete creare una nuova connessione, darle un nome, scegliere un'icona e configurarlo (solo la prima volta) per poterlo utilizzare. In **Fig. 3** la schermata iniziale del programma con la creazione della nuova connessione.
La configurazione è piuttosto semplice. La prima operazione da fare è settare la porta COM a cui è collegato il Programmatore HV (nel dubbio potete ricavarvela dalle proprietà hardware del PC). Cliccando sull'icona delle "proprietà" è possibile che esca un messaggio di avviso che confermerete con OK e si aprirà una piccola maschera con due schede; nella scheda "***Connetti a***" bisogna aprire il menu a discesa del campo "Connetti" e selezionare la porta COM a cui è collegato lo strumento, quindi occorre cliccare su "***Configura***" e dare le impostazioni che vedete nell'immagine centrale della **Fig. 4**, tenendo conto del fatto che la velocità (espressa qui in bit per secondo) deve essere uguale a quella impostata nel firmware, con la costante "BAUD". Infine selezionate la scheda "***Impostazioni***" e settate le varie voci come nell'immagine a destra della **Fig. 4**.

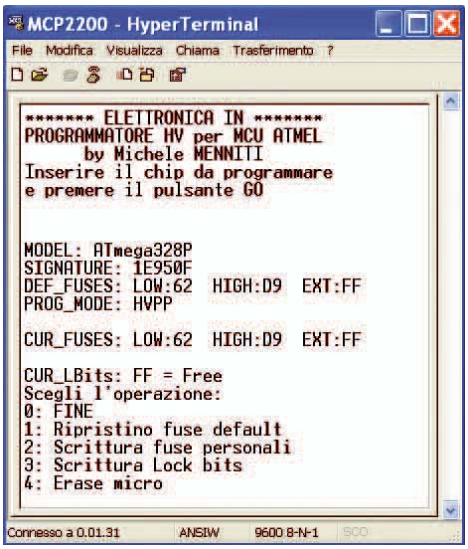


Fig. 5 *- La schermata iniziale del firmware PC su HyperTerminal*

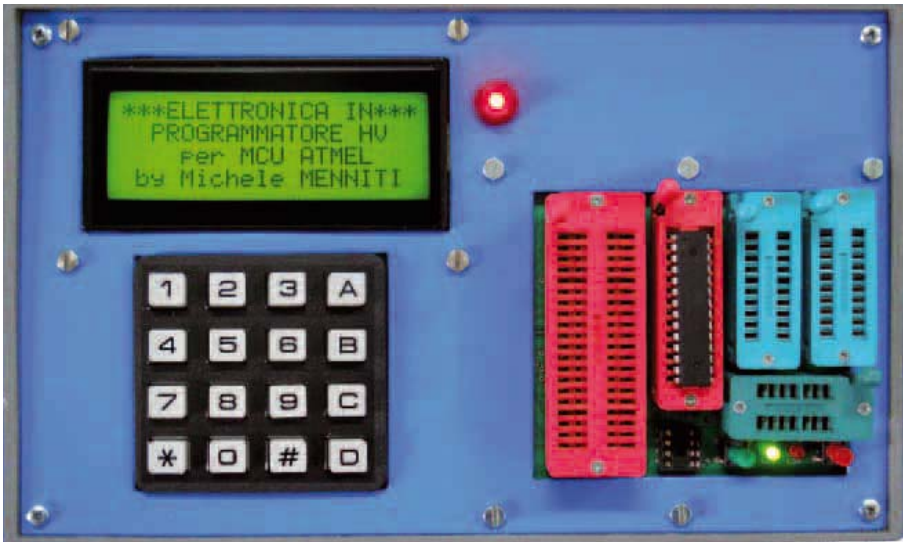


Fig. 6 *- L'avvio della Programmazione HV sullo stand-alone.*

Il programma è quindi pronto per funzionare, ma prima non dimenticate di salvare le impostazioni ricorrendo al classico comando "salva con nome" e quindi chiudetelo; in tal modo potrete crearvi un'icona di esecuzione da richiamare ogni volta che dovrete usare il Programmatore HV.
Per iniziare a lavorare aprite la connessione appena creata e cliccate sull'icona "chiama", premete il pulsante RESET sul Programmatore HV, nell'area di lavoro del programma apparirà la schermata di presentazione; dopo aver inserito il micro target e premuto il pulsante "GO" appariranno i dati del micro ed il menu principale (**Fig. 5**).
Per l'uso del Programmatore HV fate riferimento al prosieguo dell'articolo; non ci resta che ricordarvi che quando terminate le operazioni di uso del Programmatore HV è bene disconnettere l'HyperTerminal cliccando sulla relativa icona.



Fig. 7 - *Le impostazioni di default e correnti della MCU target.*

L'USO DEL PROGRAMMATORE HV STAND-ALONE

Ora trattiamo tutta la parte di uso del Programmatore HV in versione stand-alone; tutto ciò che vedremo vale anche per la versione "PC", come detto più volte le uniche differenze tra le due versioni consistono esclusivamente nella diversa gestione del menu e dei messaggi, ma la funzionalità è identica. Per tale motivo consigliamo la lettura di questa sezione anche a coloro che decideranno di realizzare la versione "PC" del Programmatore HV.

Appena si accende lo strumento sul display LCD appare la schermata iniziale con i "credits" (**Fig. 6**), e lo strumento resta in attesa della pressione di un tasto qualsiasi per procedere; quindi appare il messaggio che invita ad inserire il micro target nel proprio zoccolo e successivamente a premere il pulsante "GO" (quello rosso nella **Fig. 6**). Fino a che non si preme il pulsante "GO" vedrete acceso il LED verde, che indica che lo strumento è in modalità di attesa.

Nel nostro caso usiamo un ATmega8, lo inseriamo nello zoccolo ZIF a 28 pin e, appena premiamo il pulsante "GO", il display mostra tutti i parametri in due schermate sequenziali: modello di MCU, signature, valori di default e attuali dei fuse (in questo caso sono presenti solo Low e High Fuse Byte), nelle prime due righe della seconda schermata apparirà il valore del Lock Bit. In pratica, una volta letta la signature, dalla tabella iniziale vengono estrapolati modello del micro e valori di default dei fuse, quindi vengono letti e visualizzati i valori correnti dei Fuse Byte e del Lock Bit Byte; se il micro è vergine certamente i Fuse Byte di default combaceranno con quelli letti nella MCU (è proprio il nostro caso, come si può vedere in **Fig. 7**).



Fig. 8 - *Il Menu Principale del programmatore HV.*

Per quanto riguarda il Lock Bit Byte, la prima riga mostra il valore corrente del byte, la seconda illustra la corrispondente modalità di impostazione; nello specifico (**Fig. 7**) tutti i bit sono impostati a 1 (FF) quindi nessuno di essi è abilitato, per cui la configurazione è "Free", cioè libera da blocchi. Le altre due combinazioni principali sono FE (Write_lock) e FC (Write&Verify_lock), che poi sono quelle più importanti in quanto riguardano i due lock bits; infatti, come spiegato nella precedente puntata, esse agiscono sull'intera memoria flash e sulla memoria EEPROM.

Naturalmente potrete impostare anche altre combinazioni, usando i boot lock bits, ma in questo caso il display mostrerà solo il valore esadecimale, senza alcuna descrizione aggiuntiva.

Qualche istante di attesa ed il display viene ripulito per dare spazio al Menu Principale; a questo punto il programma si ferma e resta in attesa della pressione di un tasto da 0 a 4. (**Fig. 8**).

Nota importante: durante tutto il processo di programmazione il micro target è sottoposto alla modalità HV, quindi con i 12V applicati al pin RESET, come è facilmente deducibile dallo stato "acceso" del LED rosso (quello situato accanto al pulsante "GO"). Tale modalità deve durare il minor tempo possibile, ecco perché le varie schermate si susseguono rapidamente, ed ecco perché non conviene restare molto tempo in modalità Menu Principale; quindi effettuate rapidamente la vostra scelta in modo da procedere nel più breve tempo possibile con l'operazione da effettuare. Abbiamo fatto centinaia di prove su una decina di MCU "cavie", senza che esse abbiano subito particolari danni, però è bene seguire le indicazioni che vi stiamo fornendo.



Fig. 9 - *La Programmazione dei Fuse Byte.*

Vediamo ora l'esecuzione delle varie voci di menu.

- **Tasto 0 = Fine lavoro**: si chiude la fase di programmazione senza svolgere alcuna funzione, il display mostra un messaggio che indica la necessità di rimuovere il chip dallo zoccolo e premere il pulsante "RESET" (quello verde nella **Fig. 6**) per iniziare una nuova operazione;

a questo punto si spegne il LED rosso e si riaccende il LED verde;

- **Tasto 1 = Fuse def:** con questa operazione nel micro target vengono programmati i fuse attribuendo i valori di fabbrica (presi dalla Tabella iniziale); il display mostra un breve avviso di programmazione in corso e quindi effettua una rilettura e visualizza i nuovi valori dei fuse, affinché si possa verificare l'effettivo buon fine dell'operazione (v. prima immagine in alto della **Fig. 9**);
- **Tasto 2 = Scrivi Fuse pers:** in questo caso i valori dei fuse da scrivere dovranno essere inseriti tramite il tastierino numerico, direttamente in valore esadecimale (v. seconda immagine della **Fig. 9**), in base alle proprie esigenze (abbiamo approfondito in precedenza il programma FuseCalc e nella puntata precedente il significato dei valori dei singoli bit dei tre Fuse Byte). Subito dopo il solito messaggio di programmazione in corso ed infine, come per il punto precedente, la rilettura di verifica dei fuse appena scritti (v. terza e quarta immagine della **Fig. 9**);
- **Tasto 3 = Scrivi Lock bits:** questa scelta serve appunto per programmare il Lock Bit Byte; il display mostra le due possibili impostazioni standard (ma, come già detto, si possono usare anche quelle relative ai soli boot lock bits) ed anche in questo caso bisogna inserire il valore del byte scrivendolo direttamente in esadecimale. Subito dopo la solita rilettura e visualizzazione per la verifica dell'operazione (v. prima e seconda immagine della **Fig. 10**). A proposito della scrittura del Lock Bit Byte bisogna evidenziare un particolare comportamento: l'impostazione di un valore di blocco superiore impedisce il ritorno al valore di blocco inferiore, proprio per una questione di protezione. In pratica, se dalla condizione FF (Free) si passa alla FE (protezione in scrittura), non è più possibile tornare alla FF (livello inferiore), mentre è possibile passare alla FC (protezione in lettura/scrittura, livello superiore). Dalla FC non si può più passare alla FE o FF (livelli inferiori).
  L'altra cosa molto importante è che settare il Lock Bit Byte sul valore FE o FC implica il blocco della programmazione dei fuse, cosa di cui bisogna tener conto in fase di programmazione, come vedremo più avanti. Solo il comando di "Chip erase" può resettare il Lock Bit Byte riportandolo al valore FF e, a quel punto, permettere nuovamente la programmazione dei fuse.
- **Tasto 4 = Erase micro:** è il comando di Chip erase; come spiegato più volte questa operazione resetta il Lock Bit Byte e cancella irrimediabilmente il contenuto delle aree di memoria flash e memoria EEPROM, mentre non ha alcuna influenza sui valori dei fuse. La pressione del tasto è seguita semplicemente da un messaggio di cancellazione in corso e quindi dall'invito a rimuovere il chip dallo zoccolo. (v. terza e quarta immagine della **Fig. 10**).

Come potete vedere l'uso dello strumento è davvero molto semplice ed intuitivo. Ora faremo un esempio pratico, che ci permetterà di usare il nostro Programmatore HV per svolgere una serie di operazioni su una MCU che, questa volta, è un ATmega328P, cioè il cuore delle Arduino UNO e Duemilanove.

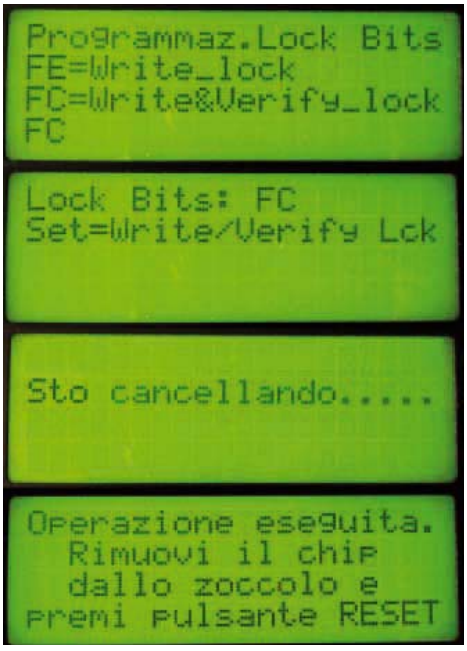


Fig. 10 - *La Programmazione del Lock Bit Byte e il Chip Erase.*

Prima di procedere con la sperimentazione programma dovete necessariamente attrezzarvi per la programmazione ISP; immaginiamo che se intendete seguire questi passaggi non abbiate alcuna difficoltà in tale senso; comunque per approfondimenti vi rimandiamo alla lettura degli articoli pubblicati sui numeri 159, 160 e 164; in quest'ultimo numero abbiamo presentato un Programmatore ISP, se l'avete realizzato certamente ora vi faciliterà enormemente le operazioni. Quindi dovrete avere ora Arduino collegato normalmente al PC e tramite i segnali ISP al micro in stand-alone.
Vi presentiamo ora un altro software: avrdude.exe, che è il compilatore utilizzato dall'IDE quando si effettua una qualsiasi operazione di scrittura del micro montato su Arduino o in configurazione stand-alone.
Avrdude deve essere usato da linea di comando, quella che conosciamo come "emulazione DOS"; poiché l'uso dei percorsi non è agevole vi consigliamo di mettere direttamente in **C:** una copia della cartella "avr", che



Fig. 11 - *L'elenco delle opzioni di AVRDUDE.*

si trova all'interno della cartella dell'IDE, nel percorso: *C:\MyPath\arduino-xxxx\hardware\tools*, in cui MyPath è il percorso che avete scelto per installare il programma per Arduino, e "xxxx" rappresenta la versione dell'IDE (p.es. 0022 o 101).
Se ora entrate in modalità DOS (normalmente da Programmi – Accessori – Prompt dei comandi) vi apparirà la tipica finestra con schermata nera e caratteri bianchi. Al prompt (solitamente **C:\percorso>**) digitate *c:\avr\bin\avrdude.exe* ed otterrete la schermata visibile in **Fig. 11** (abbiamo invertito bianco e nero per una migliore leggibilità), cioè l'elenco delle possibili opzioni del programma.
Vediamone alcune, quelle più importanti e più usate:

- -p <partno> : modello di micro da programmare;
- -b <baudrate> : velocità impostata per la porta seriale;
- -C <config-file> : percorso del file di configurazione (avrdude.conf) ;
- -c <programmer> : tipo di programmatore usato;
- -D : disabilita la cancellazione automatica della memoria flash;
- -P <port> : specifica la porta seriale usata;
- -F : disabilita il controllo della signature;
- -e : esegue il Chip erase;
- -U <memtype>:r | w | v:<filename>[:format] : questo comando serve per operare con la memoria del micro ed ha diverse opzioni; viene usato p.es. per leggere o scrivere il contenuto della memoria flash o per operare sui valori dei fuse;
- -V : disabilita la verifica dell'operazione effettuata;
- -v : attiva la modalità "verbose", una sorta di log di attività della programmazione.

Notate che alcune lettere sono usate sia in minuscolo che in maiuscolo, con funzioni differenti, quindi prestate massima attenzione quando scriverete i vostri comandi!
Per eseguire un comando con avrdude conviene creare un file batch; l'operazione consiste nell'usare un comune programma di elaborazione testi per scrivere la riga di comando e poi salvare il file dandogli estensione .bat; a quel punto sarà sufficiente dare un doppio clic per la sua esecuzione. Il vantaggio è che in tal modo non bisogna ogni volta riscrivere tutto, in caso di ripetizione del comando o di errore, come si dovrebbe invece fare lavorando direttamente da riga di comando DOS. Iniziamo a vedere come caricare uno sketch su un micro configurato in modalità stand-alone; poiché useremo avrdude ci servirà un file con estensione ".hex", cioè un firmware scritto in linguaggio di livello superiore (C++ nel nostro caso), trasformato in linguaggio macchina (quindi scritto in codice esadecimale). Quando inviate uno sketch all'Arduino, l'IDE ricorre ad avrdude per compilare il firmware, trasformandolo in un file .hex e quindi scriverlo nella memoria flash del micro target. Per cui ora vi sarà sufficiente caricare nell'IDE uno sketch qualsiasi (p.es. l'arcinoto "blink") ed effettuare la sola compilazione (verify) in modalità "verbose", in modo da vedere nella sezione di stato della finestra dell'IDE il percorso in cui il file .hex viene temporaneamente collocato. Per attivare la modalità "verbose " dell'IDE, in modo da avere un log dell'intera operazione, con le versioni 00xx è sufficiente effettuare la compilazione tenendo premuto il tasto SHIFT, mentre con le versioni 1.0.x bisogna attivare la relativa voce nelle preferenze. Eseguite un verify e vedrete che la sezione inferiore della maschera dell'IDE si riempirà di informazioni. Dovete localizzare una stringa simile a questa:
C:\DOCUME~1\PROF~1.MIC\IMPOST~1\Temp\build5681027819420524399.tmp\Blink.cpp.hex

In pratica questo è il percorso temporaneo in cui è stato messo il file compilato; basta copiare tale percorso (escluso "Blink.cpp.hex"!) ed incollarlo nella barra degli indirizzi di una qualsiasi finestra di Windows e troverete il file Blink.cpp.hex; copiatelo in C:\.
Ora, con un programma tipo Wordpad o Notepad (non usate programmi evoluti come Word!) scrivete la seguente riga unica (senza andare daccapo!):

se disponete dell'IDE-0022/23:

"C:\avr\bin\avrdude.exe" -C "C:\avr\etc\avrdude.conf" -p m328p -c stk500v1 -b 19200 -P COM2 -U flash:w:C:\Blink.cpp.hex:i

Invece, se disponete dell'IDE-1.0.1, dovete scrivere:

"C:\avr\bin\avrdude.exe" -C "C:\avr\etc\avrdude.conf" -p m328p -c arduino -b 19200 -P COM2 -U flash:w:C:\Blink.cpp.hex:i

quindi salvate il file in **C:** con tipo "testo" e nome "Sketch_write.bat".

Nota importante: non usate la versione IDE 1.0, usate invece la nuova versione 1.0.1, in caso contrario la tecnica ISP non vi funzionerà.

Ma cosa significano queste righe? Il programma avrdude.exe localizza, mediante l'opzione "-C" il file di configurazione avrdude.conf; mediante l'opzione "-c" imposta il tipo di programmatore ("stk500v1" o "arduino" in base alla versione di IDE); mediante l'opzione "-p" imposta il modello di micro da programmare (mega328P = ATmega328P); con l'opzione "-b" imposta il baud rate della comunicazione seriale (nota bene: questo valore non ha alcun legame con quello settato nel firmware del Programmatore HV, in questo caso rappresenta la velocità di comunicazione alla quale è impostato lo sketch ArduinoISP); con l'opzione "-P" imposta la porta COM2 a cui è collegato Arduino (naturalmente se la porta è diversa userete quella corretta) ; infine l'opzione -U viene usata per scrivere sulla memoria flash ("flash:w:") il file Blink.cpp.hex che avete copiato poco prima in **C:**.
Ora dovete caricare sul vostro Arduino il programma ArduinoISP ed avere pronto il collegamento richiesto per la programmazione ISP.
A questo punto con un doppio clic sul file Sketch_write.bat lo eseguite e partirà la programmazione ISP. Durante questa fase si aprirà una finestra DOS che mostrerà in modo molto rapido i vari passaggi; se volete vederli e leggerli con calma, non dovrete fare altro che andare al prompt dei comandi e scrivere ***Sketch_write*** seguito da INVIO. In entrambi i casi a fine operazione il blink funzionerà alla perfezione sul micro in stand-alone.
Quindi con questa semplice tecnica potete programmare qualsiasi micro le cui impostazioni siano presenti nel file avrdude.conf, basta ovviamente mettere i parametri adeguati.
Ed ora provate l'operazione opposta, cioè come leggere il contenuto della memoria flash del micro stand-alone e salvarlo in un file .hex. Questa volta la riga da scrivere sarà per la 0022/23:

"C:\avr\bin\avrdude.exe" -C "C:\avr\etc\avrdude.conf" -p m328p -c stk500v1 -b 19200 -P COM2 -U flash:r:C:\ Flash_read.hex:i

oppure, per la 1.0.1:

"C:\avr\bin\avrdude.exe" -C "C:\avr\etc\avrdude.conf" -p m328p -c arduino -b 19200 -P COM2 -U flash:r:C:\ Flash_read.hex:i

Ora salvate il file in C:\ con tipo "testo" e nome "Flash_read.bat"; a questo punto con un doppio clic lo eseguite e vedrete di nuovo la finestra DOS (anche in questo caso potete eseguire il file dal prompt dei comandi), alla fine in **C:** troverete un file "Flash_read.hex" che, in questo caso, non è altro che il blink estrapolato dalla memoria flash del micro di Arduino.
Aver imparato come leggere e scrivere la memoria flash del micro in stand-alone vi sarà utile nel prosieguo della prossima sperimentazione.
A questo punto disponete di una MCU ATmega328P settata per lavorare a 16MHz (se avete seguito tutte le operazioni ISP in base alle tecniche spiegate sui numeri sopraelencati) e contenente il firmware blink. Immaginando ora che tale firmware sia qualcosa di molto potente e delicato e che vada protetto sia da tentativi di copia che da sovrascritture accidentali, vedrete ora quali sono le operazioni da seguire mediante il Programmatore HV, al fine di raggiungere tale scopo.
Prima di iniziare ricordate di tener presente un particolare importante: il keypad esadecimale che abbiamo mostrato nelle foto, al posto delle lettere "E" e "F" ha rispettivamente i caratteri "*" e "#"; ma al momento di usare tali tasti, avendo specificatamente programmato il micro della sezione *Keypad & Display*, sul display vedrete apparire correttamente "E" e "F". Vi consigliamo però di applicare sui tasti due etichette adesive con le lettere corrette, per evitare confusione al momento della programmazione.
Dunque gli elementi di programmazione che servono per proteg-

gere il micro sono il fuse "SPIEN" per l'abilitazione/disabilitazione della programmazione seriale e i due lock bits del Lock Bit Byte, responsabili della protezione in scrittura/lettura delle memorie flash e EEPROM e del blocco della programmazione dei fuse.
Bisognerà prima di tutto disabilitare il fuse "SPIEN" e poi impostare il Lock Bit Byte su FC; la sequenza è obbligata in quanto, come già detto, una volta settato il Lock Bit Byte su FC non sarà più possibile variare i fuse.
Quindi accendete il Programmatore HV e, alla relativa richiesta, inserite nel proprio zoccolo il micro Atmega328P.
Se il micro è stato usato in base alle istruzioni della programmazione ISP, spiegate nei numeri della Rivista indicati in precedenza, saranno rilevati i seguenti valori: LFuse = FF, HFuse = DF, EFuse = FF (o 07, è uguale), mentre il Lock Bit Byte sarà = FF (Free). Alla comparsa del Menu Principale premete "2" per la programmazione personalizzata dei fuse e fornite questi valori: LFuse = FF, HFuse = FF, EFuse = FF. A questo punto, dopo la rilettura di verifica, vi sarà chiesto di togliere il micro dallo zoccolo e premere il pulsante RESET; eseguite alla lettera e reinserite il micro nello zoccolo.
Questa volta, all'apparire del Menu Principale, premete "3" per programmare il Lock Bit Byte, ed inserite il valore "FC" (Write&Verify_lock).
A questo punto la programmazione per la protezione è terminata. Rimettendo il micro sulla sua board e alimentandolo vedrete che il blink continua a funzionare senza problemi, segno che non c'è stata alcuna cancellazione della memoria flash (che peraltro non avete richiesto con le operazioni sopraindicate) in cui precedentemente avevate caricato il file .hex del blink.
Però se provate a ripetere le operazioni fatte in precedenza con avrdude, sia in lettura che in scrittura, vedrete che non andranno a buon fine, proprio perché in precedenza avete disabilitato ogni possibile forma di accesso alla memoria del micro.
Lo stesso accadrà se tenterete di inserire un qualsiasi sketch nel micro, usando Arduino come Programmatore ISP e l'IDE.
Facciamo notare ai più esperti che il primo messaggio di errore che esce è quello di signature non valida, in quanto avrdude, non riesce ad accedere alla memoria flash del micro e quindi nemmeno a quella piccolissima parte (3 byte) che essa riserva alla signature. Poiché lo stesso avrdude consiglia di usare il parametro -F per evitare l'errore, potete anche tentare di aggiungere tale parametro alla riga di comando, nei due file batch; il risultato sarà che le operazioni apparentemente saranno eseguite ma alla fine della scrittura uscirà un altro errore e vedrete che sul micro non avete caricato nulla, e alla fine della lettura otterrete un file .hex che però non contiene il firmware presente nel micro.
In parole povere non sarà ora più possibile copiare, leggendolo, il contenuto del micro, né sovrascrivere (magari dopo aver apportato delle modifiche) il firmware presente nel micro.
Ormai sappiamo che questo status non è definitivo; in qualsiasi istante decidiate di resettare il micro vi basterà inserirlo nuovamente nel Programmatore HV e scegliere l'opzione "4", cioè effettuare l'operazione di Chip erase, che resetterà istantaneamente il Lock Bit Byte su "FF" (Free); a questo punto sarà anche possibile riportare i tre fuse del micro su FF-DF-FF, come in origine; il micro sarà nuovamente utilizzabile appieno.
Qualcuno potrà obiettare che la protezione non è così efficace se un Chip erase può azzerare tutto. Bisogna però aver chiaro che comunque non c'è alcun modo per copiare il firmware inserito precedentemente e nemmeno si può sovrascrivere accidentalmente, quindi la protezione funziona. Infatti il micro può essere solo definitivamente cancellato, e con un'operazione che non può essere certo eseguita per sbaglio!
Bene, con questo esempio di programmazione vi abbiamo mostrato la potenza del Programmatore HV che, in pochi istanti, vi ha permesso di manipolare a piacimento fuse e lock bits e che, naturalmente, si presta per svariate possibili applicazioni.
Terminiamo quindi, con quest'ultima puntata, la trattazione di uno strumento che non può mancare sul banco di chiunque operi frequentemente con i microcontrollori delle varie famiglie del produttore ATMEL.

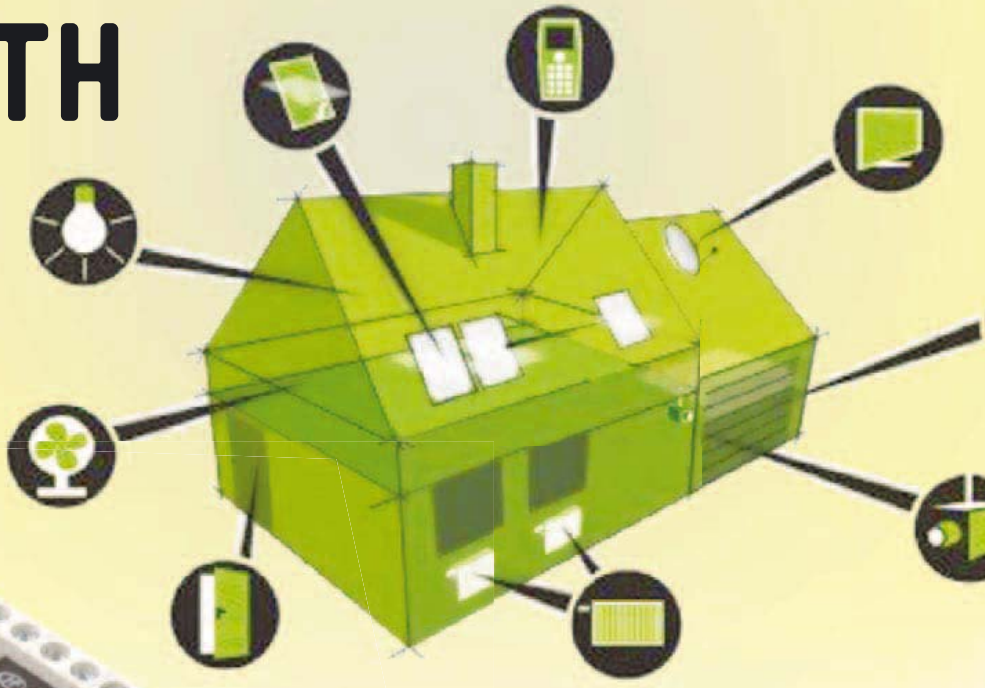


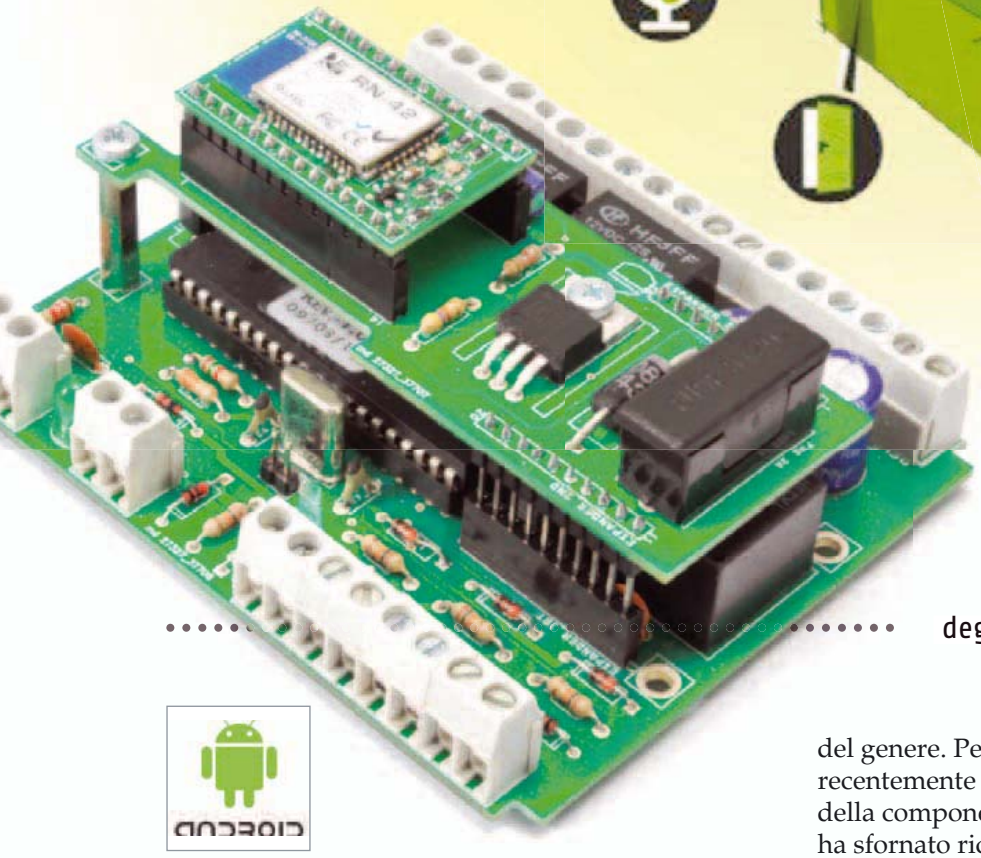
per il MATERIALE

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

SCHEDA DI I/O BLUETOOTH ANDROID BASED





Permette di comandare con uno smartphone Android 4 uscite a relé e altrettante a transistor, ma anche di leggere lo stato di 4 ingressi digitali.

degli Ingg. TOMMASO e ALESSANDRO GIUSTO

Realizzare un'interfaccia o una scheda di controllo via Bluetooth era, fino a qualche anno fa, piuttosto costoso, tanto che sovente, pur trovando in Internet numerose applicazioni gratuite per il comando di luci e domotica da iPhone e smartphone Android-based, molti di voi hanno rinunciato a costruirsi dispositivi del genere. Per fortuna, recentemente il mercato della componentistica ha sfornato ricetrasmittenti Bluetooth low-cost, qualitativamente valide e di facile impiego; un ottimo esempio è il modulo RN-42 della Roving Networks (commercializzato dalla Futura Elettronica, *www.futurashop.it*, al prezzo di soli 21 euro) sul quale si basa il progetto descritto in queste pagine. Ciò che vi proponiamo di realizzare è una scheda di I/O ad interfaccia wireless Bluetooth, provvista di quattro relé ed altrettanti ingressi, rispettivamente comandabili e acquisibili a distanza

Il modulo **Bluetooth**

La connessione wireless del nostro sistema viene operata dal modulo RN-42 della Roving Networks (*www.rovingnetworks.com*), un dispositivo il cui funzionamento può essere visto dall'utente come un collegamento seriale; ciò, grazie al fatto che lo stack Bluetooth è completamente incapsulato, quindi utilizzando, ad esempio, un programma di emulazione di terminale, l'unità ricevente restituisce ogni volta il carattere corrispondente al tasto premuto sulla tastiera. In altre parole, la gestione del protocollo Bluetooth avviene all'interno del modulo ed è trasparente all'utente, il che ci permette di realizzare una comunicazione su Bluetooth semplicemente gestendo l'RN-42 da una seriale.



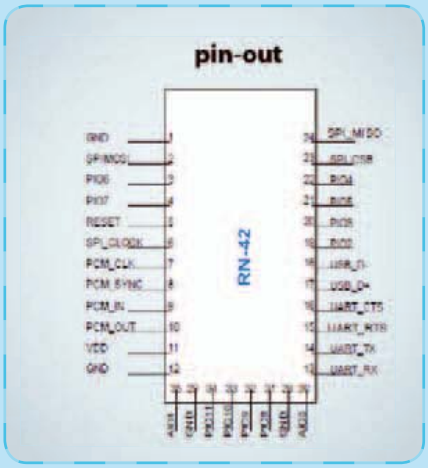


Il nostro modulo è compatibile Bluetooth 2.0 per quanto riguarda la velocità di comunicazione, mentre sul piano della potenza irradiata si colloca in Classe 2, il che significa che può garantire una copertura di circa 20 metri; il basso consumo (solo 26 microampere in sleep-mode) rende l'RN-42 ideale per le applicazioni mobili e comunque per i sistemi alimentati a batteria.
Il componente supporta i profili Bluetooth SPP e HID e dialoga con l'esterno mediante un'interfaccia UART con data-rate fino a 3 Mbps; sul link Bluetooth, la velocità massima è di 721÷2.000 kbps. Il modulo funziona con una tensione di 3,3 V.
Nel riquadro a lato vedete la piedinatura dell'RN-42.
Per quanto riguarda le connessioni, il componente presenta 35 piedini, che hanno il seguente significato:

1) GND; massa del modulo;
2) SPIMOSI; MOSI dell' interfaccia SPI (utilizzato solo per la programmazione);
3) PIO6; pin generico di I/O digitale oppure utilizzato per configurare il modulo come Master;
4) PIO7; pin generico di I/O digitale oppure utilizzato per configurare il baud-rate (chiuso=9.600 baud, aperto=configurabile da software);
5) RESET; reset hardware del modulo (attivo a livello basso);
6) SPI_CLK; clock dell'interfaccia SPI (utilizzato solo per la programmazione);
7) PCM_CLK; clock dell'interfaccia PCM;
8) PCM_SYNC; SYNC dell'interfaccia PCM;
9) PCM_IN; Input dell'interfaccia PCM;
10) PCM_OUT; Output dell'interfaccia PCM;
11) VDD; alimentazione (3,3V) del modulo;
12) GND; massa del modulo;
13) UART_RX; canale di ricezione interfaccia UART del modulo (da collegare al pin TX interfaccia UART dell'eventuale MCU);
14) UART_TX; canale di trasmissione interfaccia UART del modulo (da collegare al pin RX interfaccia UART dell'eventuale MCU);

mediante uno smartphone dotato di sistema operativo Android e di un semplice software di gestione.
Per essere esatti, l'hardware dispone di 4 relé, 2 uscite di tipo open-collector (impostabili in configurazione standard oppure, montando resistenze aggiuntive nelle apposite piazzole, in configurazione per pilotare dei LED), 2 uscite di potenza open-drain (il cui stadio finale è realizzato da MOSFET BUZ11) e infine 4 ingressi digitali.
Con un opportuno software (sviluppato ad hoc) installato ed in esecuzione su smartphone Android, sarà possibile collegarsi al sistema, leggere e verificare a colpo d'occhio (grazie a una chiara interfaccia grafica) lo stato corrente degli input e output e, grazie al display con tecnologia touch-screen, premere dei pulsanti virtuali per inviare la richiesta di modificare il livello logico delle uscite. Alcuni esempi di applicazioni della scheda possono essere il controllo remoto tramite smartphone di apparati elettronici all'interno di impianti automatizzati oppure la gestione di sistemi domotici. Quest'ultima applicazione è, oltre che una comodità, un indispensabile ausilio a chi ha disabilità motorie o è impossibilitato a muoversi perché si trova in convalescenza dopo un trauma o un intervento chirurgico; in tal caso poter controllare con lo smartphone le tapparelle motorizzate, l'elettroserratura del cancelletto, le luci, ma anche rispondere al citofono interfacciando la scheda di I/O con la fonìa, allevia la condizione di disabilità e permette di fare ciò che altrimenti sarebbe impossibile far da soli.
Il limite alla distanza di funzionamento della comunicazione tra scheda elettronica e smartphone è dato dal Bluetooth, ovvero dalla portata dell'RN-42, che si colloca su 15÷20 metri, nonché dalle prerogative del telefono. Le prove che abbiamo fatto hanno dimostrato che all'interno di un'abitazione tali distanze vengono rispettate, se non addirittura superate (siamo passati da una stanza all'altra, da un'abitazione all'altra, da un piano ai superiori o inferiori, fino a una cantina seminterrata senza alcun problema); invece all'aperto pensiamo che possano essere facilmente superate, anche di parecchie volte. Sarebbe possibile aumentare la portata, magari cambiando modulo e antenna, ma si uscirebbe dagli scopi del progetto; inoltre, pur essendo il Bluetooth un protocollo abbastanza sicuro, aumentando la portata andrebbero previsti dei meccanismi di maggiore sicurezza (magari aggiungendo alla keypass una password di collegamento).

15) UART_RTS; segnale Request To Send dell'UART del modulo;
16) UART_CTS; segnale Clear To Send interfaccia UART del modulo;
17) USB-D+; dato positivo interfaccia USB;
18) USB-D-; dato negativo interfaccia USB;
19) PIO2; I/O digitale generico, anche utilizzato dal modulo per segnalare il proprio stato di connessione in corso;
20) PIO3; I/O digitale generico, anche utilizzato per configurare l'autodiscovery del modulo;
21) PIO5; I/O digitale generico, anche oppure utilizzato dal modulo per segnalare, tramite opportuni lampeggi, il proprio stato di funzionamento;
22) PIO4; I/O digitale generico, anche utilizzato per resettare le configurazioni standard di fabbrica;
23) SPI_CSB; pin CSB dell'interfaccia SPI (utilizzato solo per la programmazione);
24) SPI_MISO; pin MISO dell'interfaccia SPI (utilizzato solo per la programmazione e non per il normale utilizzo);
28) GND; massa del modulo;
29) GND; massa del modulo;
30) AIO0; pin generico di I/O analogico;
31) PIO8; I/O digitale generico, anche utilizzato per segnalare la comunicazione UART;
32) PIO9; pin generico di I/O digitale;
33) PIO10; I/O digitale generico, oppure segnale DTR remoto;
34) PIO11; I/O digitale generico, oppure segnale RTS remoto;
35) AIO1; pin generico di I/O analogico.



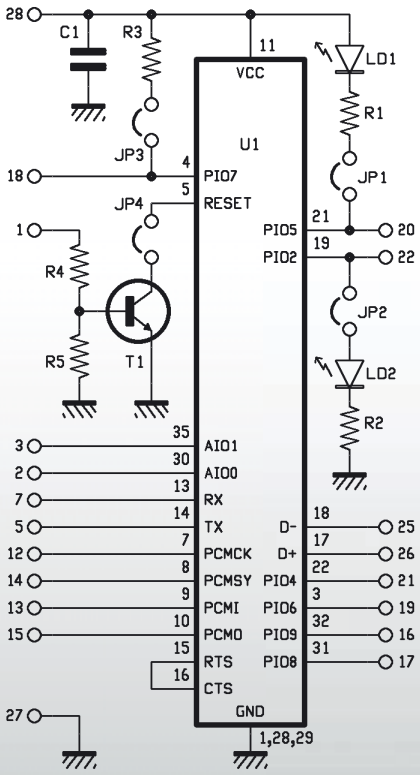
Entriamo dunque nel vivo del progetto, dicendo che consta di tre circuiti stampati da montare uno sull'altro: la scheda base, il modulo d'alimentazione ed il socket, che è l'adattatore su cui si monta il modulo Bluetooth.

IL MODULO RN-42

Quest'ultimo è un transceiver Bluetooth 2.1 di classe 2 con antenna di trasmissione integrata (esiste anche in versione predisposta al collegamento di un'antenna esterna) capace di comunicare a una velocità fino a 3 Mbps. La versione con antenna integrata ha un range teorico di ricetrasmissione di circa 15/20 metri, mentre quella con antenna esterna può spingersi fino a 1.000. Il modulo supporta le modalità di funzionamento denominate SPP (Serial Port Profile) e HID (Human Interface Device); implementa un efficace meccanismo di riduzione del consumo che lo porta, nelle migliori condizioni, ad assorbire circa 26 µA.
L'RN-42 dispone di una semplice interfaccia hardware e software UART che può essere utilizzata per il collegamento a qualsiasi microcontrollore allo scopo di realizzare sistemi embedded; inoltre, avendo integrato lo stack software Bluetooth, richiede al micro risorse limitate. Il modulo supporta anche la modalità di comunicazione denominata *UART over-the-air*, che ne permette la gestione anche da remoto (questa modalità può essere disabilitata). Il modulo implementa la cifratura della comunicazione (vengono utilizzati algoritmi a 128 bit) e la correzione automatica degli errori di trasmissione; queste ultime due caratteristiche rendono l'RN-42 ideale anche per applicazioni particolarmente critiche, in cui vengono richieste la massima sicurezza e le minime perdite di dati in trasmissione.

SCHEMA SCHEDA SOCKET

Passiamo ora ad analizzare lo schema elettrico dell'adattatore socket da noi progettato, che è piuttosto semplice, essendo un adattatore che consente di montare l'RN-42 sulla scheda base; in pratica il circuito dispone, sui due lati lunghi, di un connettore formato da due pin-strip a passo 2,54 mm (ciascuno da 14 contatti) attraverso i quali vengono portati all'esterno quasi tutti i piedini del modulo.
Il componente principale della scheda è ovviamente U1, ossia il modulo Roving Networks, i cui pin PIO5 e PIO2 possono avere una doppia funzione: se realizziamo i due ponticelli a stagno JP1 e JP2 sul lato inferiore dello stampato comandano i LED rispettivamente di *stato* e *connect*; lasciando aperti questi jumper, sono disponibili come generici I/O digitali. Per semplificare il software, il pin PIO7 è invece



[schema elettrico SOCKET]

Piedinatura **Socket**

Nel nostro sistema, il modulo Bluetooth RN-42 viene sistemato su un circuito stampato adattatore che ne agevola il montaggio e l'eventuale sostituzione; infatti rende accessibili su due file di 14 pin a passo 2,54 mm (quelli dei comuni integrati) i contatti del modulo che sono a saldare e a passo dimezzato.
Con il socket possiamo quindi montare l'RN-42 come un qualsiasi integrato ed avere disponibili su di esso le principali segnalazioni.
Il socket peraltro è stato progettato cercando di mantenere la compatibilità con la scheda di controllo via Bluetooth che abbiamo proposto nel fascicolo n° 112.
Le connessioni del socket realizzate dalle due file di pin sono:

1) RESET: reset hardware del modulo RN-42 (a differenza del modulo RN-42 originale, questo segnale non è attivo basso ma attivo alto);
2) AN0: collegato direttamente al segnale AI00 ingresso analogico 0 del modulo RN-42;
3) AN1: collegato direttamente al segnale AI01 ingresso analogico 1 del modulo RN-42;
4) CTS: segnale Clear To Send interfaccia hardware UART (come specificato dal datasheet dell'RN-42, se non si utilizza nessun controllo di flusso, collegato direttamente al segnale RTS);
5) TX: segnale UART-TX interfaccia hardware UART (porta trasmissione modulo RN-42; in sistemi embedded andrà collegato alla porta di ricezione interfaccia UART MCU);
6) RTS: segnale Ready To Send interfaccia hardware UART (come specificato dal datasheet dell'RN-42, se non si utilizza nessun controllo di flusso, collegato direttamente al segnale CTS);
7) RX: segnale UART-RX interfaccia hardware UART (porta ricezione modulo RN-42; in sistemi embedded andrà collegato alla porta di trasmissione interfaccia UART MCU);
8) RI: piedino non collegato in quanto non presente un equivalente nel modulo RN-42;
9) DCD: piedino non collegato in quanto non presente un equivalente nel modulo RN-42;
10) DSR: piedino non collegato in quanto non presente un equivalente nel modulo RN-42;
11) DTR: piedino non collegato in quanto non presente un equivalente nel modulo RN-42;
12) PCM-CLK: segnale di clock interfaccia PCM del modulo RN42;
13) PCM-IN: segnale di input interfaccia PCM del modulo RN42;
14) PCM-SYNC: segnale di sincronismo interfaccia PCM del modulo RN42;
15) PCM-OUT: segnale di output interfaccia PCM del modulo RN42;
16) IO9: collegato direttamente al segnale PIO9 del modulo RN42 (segnale I/O digitale numero 9);
17) IO8: collegato direttamente al segnale PIO8 del modulo RN42 (segnale I/O digitale numero 9);
18) IO7: collegato direttamente al segnale PIO7 del modulo RN42 (segnale I/O digitale numero 7).
19) IO6: collegato direttamente al segnale PIO6 del modulo RN42 (segnale I/O digitale numero 6);
20) IO5: collegato direttamente al segnale PIO5 del modulo RN42 (segnale I/O digitale numero 5).
21) IO4: collegato direttamente al segnale PIO4 del modulo RN42 (segnale I/O digitale numero 4);
22) IO2: collegato direttamente al segnale PIO2 del modulo RN42 (segnale I/O digitale numero 2).
23) IO1: piedino non collegato in quanto non presente un equivalente nel modulo RN-42;
24) 3V3MONITOR: piedino non collegato in quanto non presente un equivalente nel modulo RN-42;
25) USB-: segnale negativo trasmissione USB del modulo RN-42;
26) USB+: segnale positivo trasmissione USB del modulo RN-42;
27) GND: segnale negativo (massa) di alimentazione del modulo RN-42;
28) 3V3: segnale positivo di alimentazione del modulo RN-42.

Il pin 18 può essere collegato tramite un jumper e una resistenza di pull-up per impostare fisso a 9.600 bps il baud rate porta seriale; nel socket abbiamo previsto il jumper e una resistenza di pull-up per impostare fisso a 9.600 bps il baud rate porta seriale.
Il pin 20 può essere collegato a un LED mediante cui il modulo Bluetooth indica il proprio stato di funzionamento (lo vedete nello schema del socket).
Quanto al pin 22, può essere collegato a un LED gestito automaticamente dal modulo per indicare lo stato di connessione Bluetooth attiva; infatti nel socket è presente il jumper JP2 per sconnettere il LED e utilizzare il pin come generico piedino di I/O.

collegato (tramite il jumper JP3) a una resistenza di pull-up, in modo che il baud-rate della porta UART venga forzato a 9.600 bps.
Nel socket è presente una sezione (centrata intorno al BJT T1) relativa alla gestione del reset hardware; l'ingresso della sezione viene preso dal pin 1 del connettore (la coppia di pin-strip) del socket, mentre l'uscita è applicata al pin RESET dell'RN-42. Quando il contatto 1 del connettore del socket viene posto a livello logico alto, il collettore di T1 resetta il modulo Bluetooth; è possibile disattivare questa funzione lasciando aperto il jumper JP4.
Un'ultima nota riguarda il condensatore C1, collegato in parallelo ai pin di alimentazione di U1, che serve a fornire una piccola scorta di energia durante la trasmissione in Bluetooth, allorché il modulo causa picchi di assorbimento che potrebbero originare disturbi e propagarli lungo la linea di alimentazione.
Dal punto di vista software, il modulo viene gestito tramite opportuni comandi inviati attraverso il collegamento UART.
All'accensione, l'RN42 si avvia in configurazione Data (trasmissione e ricezione trasparente dei dati); per entrare nella configurazione Command, in cui il modulo accetta i comandi, è necessario inviare i caratteri **$$$**. Per uscire dalla stessa basta inviare i caratteri **---** seguiti da **<cr>**.
Il modulo può funzionare secondo sei distinte modalità Bluetooth, selezionate tramite il comando **SM** e descritte qui di seguito.

1) Slave Mode (comando **SM,0**): modalità predefinita in cui il modulo è in continua attesa di richieste di connessione, che può accettare o rifiutare (non può invece avviare connessioni di propria iniziativa).
2) Master Mode (comando

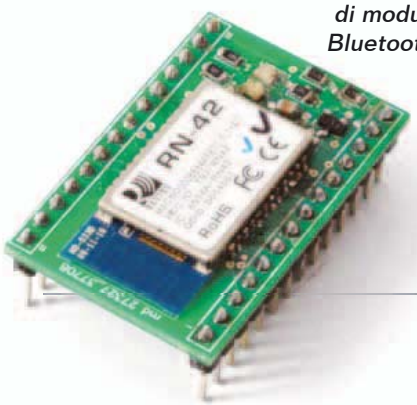
SM,1): modalità opposta alla precedente in cui il modulo può avviare connessioni di propria iniziativa ma non può essere localizzabile da altre periferiche.

3) Trigger Master Mode (comando **SM,2**): in questa modalità il modulo, ogni qual volta viene ricevuto un carattere sulla UART locale, si connette automaticamente a uno slave cui era stato precedentemente abbinato. La connessione rimane attiva per un intervallo programmabile tra 1 e 255 secondi.

4) Auto-connect (Master Mode) (comando **SM,3**): in questa modalità, all'accensione il modulo si connette automaticamente a uno slave cui è stato precedentemente abbinato. La connessione rimane attiva per tutto il tempo in cui il modulo risulta alimentato; nel caso in cui la comunicazione cada, il modulo tenta automaticamente di ricollegarsi allo stesso slave.

5) Auto-connect (DTR Mode) (comando **SM,4**): modalità simile alla precedente, tranne per il fatto che è possibile avviare e chiudere la connessione tramite lo stato del pin PIO6.

6) Auto-connect ANY Mode (comando **SM,5**): modalità simile alla precedente, tranne che per il fatto che ogni volta che varia lo stato del PIO6 viene avviata una nuova procedura di abbinamento a uno slave.



Socket completo di modulo Bluetooth.

iPhone si fa la tastiera **Bluetooth**



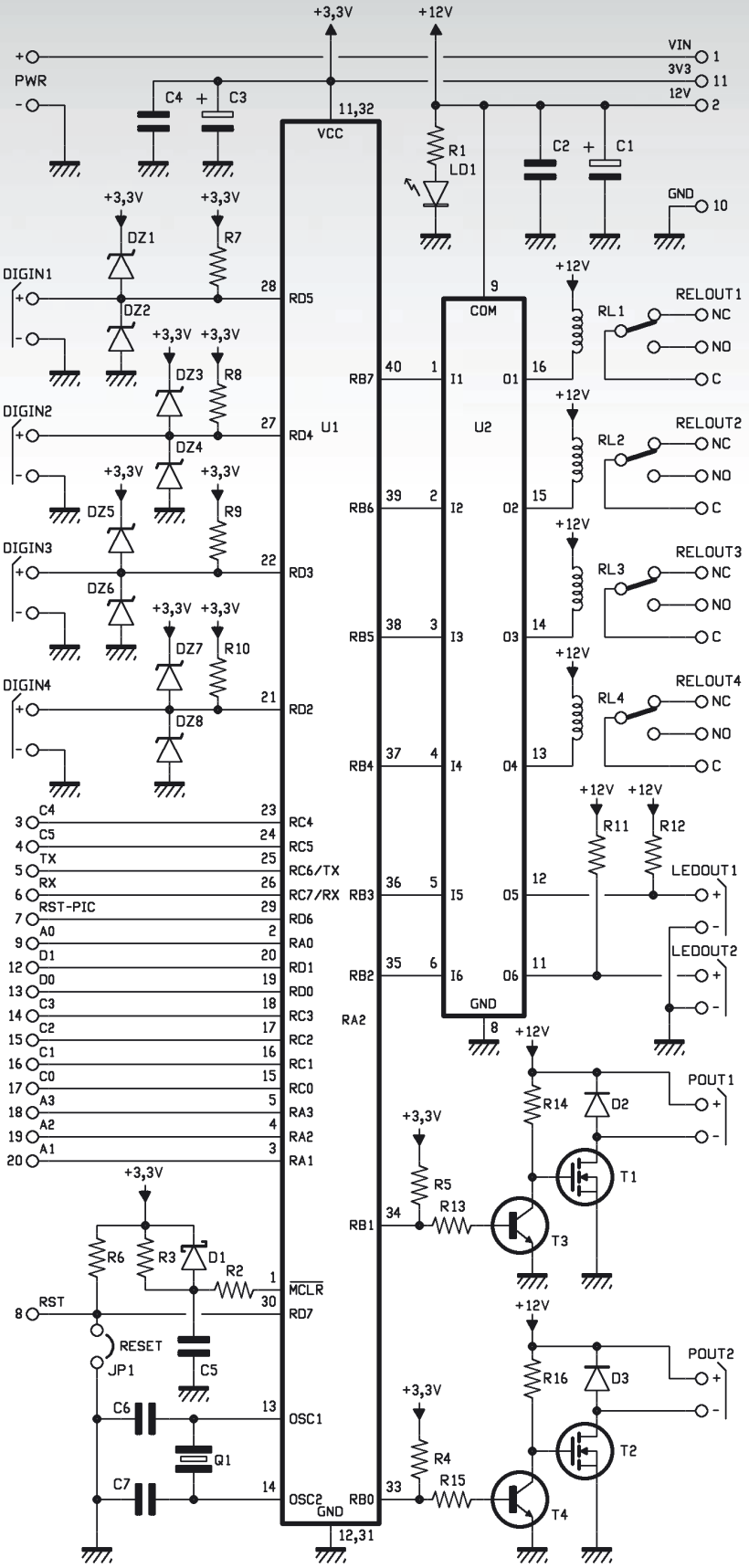
Sappiamo che il celebre smartphone di Apple ha un touch-screen che funge anche da tastiera, però una tastiera virtuale non è come una vera. Devono aver pensato così i realizzatori di Miia Bluetooth Keyboard Case, la custodia per iPhone 4/4S che integra una tastiera Bluetooth a scorrimento. Si tratta di un case che oltre a proteggere l'iPhone 4/4S da urti e graffi, lo completa con una tastiera full- qwerty slide-out a scorrimento. La tastiera è di tipo ultrapiatto ed è alimentata da una propria batteria ricaricabile (dura tipicamente 45 giorni); si collega all'iPhone via Bluetooth, quindi non occupa alcun connettore e anzi, il case è fatto in modo da lasciare libero accesso a tutte le porte e ai pulsanti dello smartphone. Un pulsantino permette un passaggio veloce dal sistema di scrittura touch-screen alla più familiare ed agevole tastiera. Alcuni tasti aggiuntivi consentono, inoltre, di accedere alle funzioni più utilizzate come copia & incolla e alla navigazione; un ulteriore tasto permette di disattivare il Bluetooth quando non si vuole utilizzare il case con la tastiera. La dotazione prevede anche un cavo USB per la ricarica della batteria interna.

I comandi supportati dal modulo sono composti da uno o due caratteri più alcuni parametri e vengono suddivisi in cinque classi principali:

- SET COMMANDS, utilizzati per la configurazione;
- GET COMMANDS, servono per leggere la configurazione corrente;
- CHANGE COMMANDS, usati per modificare temporaneamente la configurazione;
- ACTION COMMANDS, utilizzati per richiedere al modulo l'esecuzione di alcune azioni (avvio connessione, interrogazione dispositivi, ecc…);
- GPIO COMMANDS, usati per configurare e manipolare i pin di I/O generici (GPIO) del modulo.

Un'analisi completa ed approfondita di tutti i comandi richiederebbe molte pagine (eventualmente vi rimandiamo al manuale utente disponibile sul sito della Roving Networks) perciò in questo articolo prendiamo in considerazione solo i principali ed utilizzati:

- comando **SA,<1,0>**; attivazione/disattivazione autenticazione. Se attivata, per ogni nuova connessione, verrà richiesta l'autenticazione;
- comando **SE,<1,0>**: attivazione/disattivazione cifratura comunicazione;
- comando **SF,1**: forzatura configurazione di fabbrica (default);
- comando **SM,<5,4,3,2,1,0>**: impostazione modalità di funzionamento (vedere quanto detto



al punto **6)** dell'elenco modalità Bluetooth);
- comando **SN,<name>**: impostazioni nome Bluetooth associate al modulo RN-42;
- comando **SP,<text>**: impostazione pin code di sicurezza Bluetooth;
- comando **SU,<rate>**: impostazione baud-rate della porta seriale UART;
- comando **D**: lettura impostazioni base Bluetooth del modulo;
- comando **E**: lettura impostazioni estese Bluetooth del modulo;
- comando **GK**: lettura stato corrente connessione Bluetooth (1: modulo connesso; 0: modulo disconnesso);
- comando **G&**: lettura stato pin I/O PIO;
- comando **$$$**: passaggio dalla modalità dati alla modalità comandi dell'RN-42 (dopo i caratteri $$$, tutti i caratteri ricevuti attraverso la porta UART vengono considerati dal modulo come dei comandi).
- comando **---<cr>**: passaggio dalla modalità comandi alla modalità dati dell'RN-42 (tutti i caratteri ricevuti dopo **---** vengono considerati come dati da trasmettere attraverso il canale Bluetooth);
- comando **C,<address>**: richiesta connessione ad un dispositivo specificato attraverso indirizzo MAC;
- comando **K,**: chiusura forzata connessione attiva;
- comando **L**: lettura indice qualità della connessione attiva;
- comando **R,1**: forzatura reboot software del modulo RN-42.

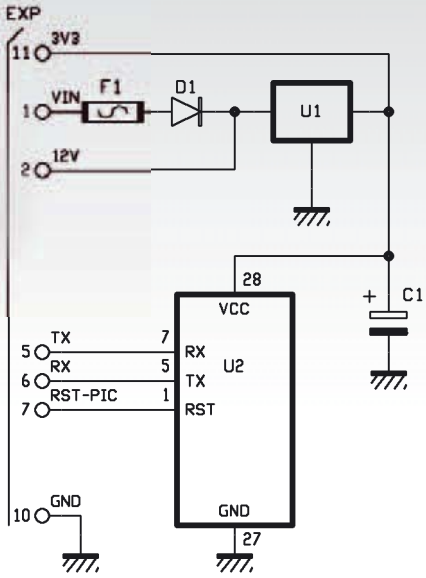
SCHEDA BASE E ALIMENTAZIONE

Vediamo adesso le due schede più grandi del sistema, ovvero quella inferiore (*scheda base*) che comprende quasi tutti i componenti, e quella intermedia (*interfaccia*) contenente parte

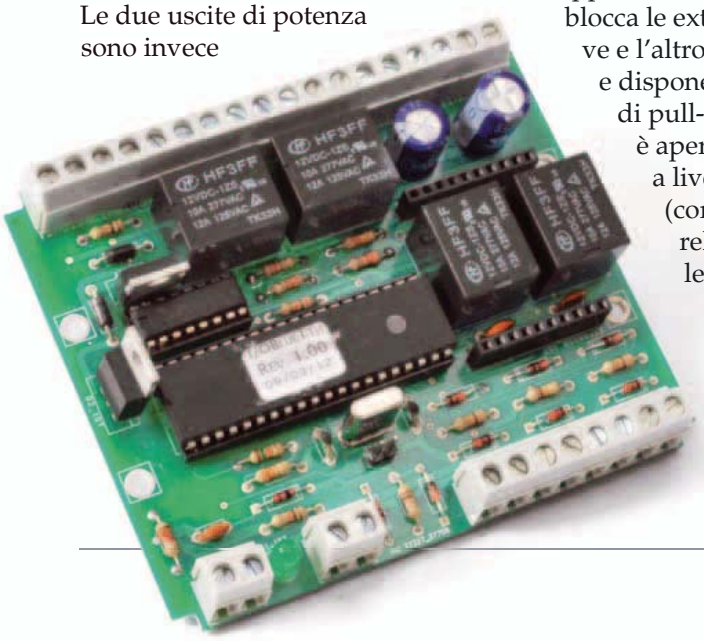
[schema elettrico INTERFACCIA]



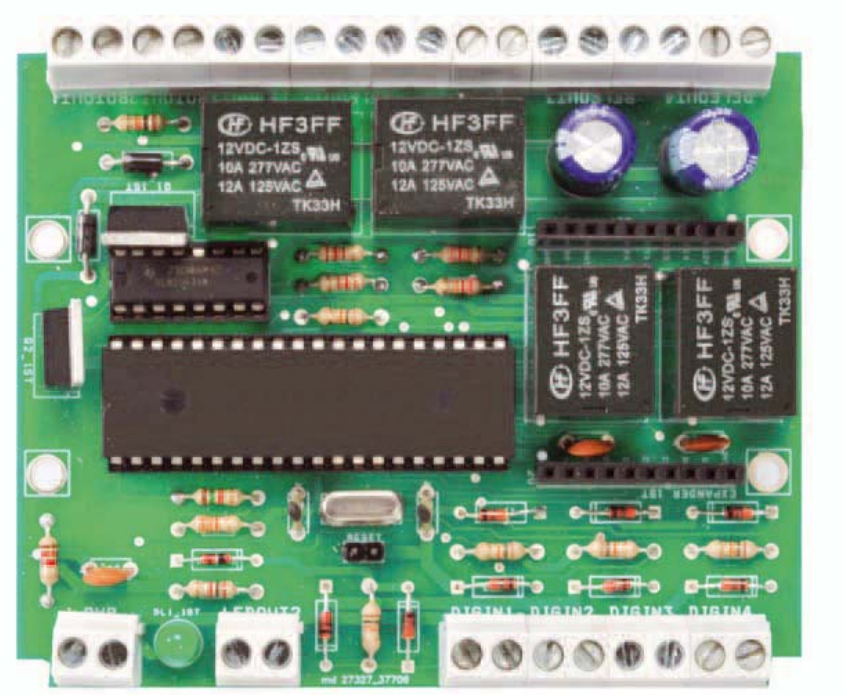
Scheda interfaccia.



della sezione di alimentazione e i connettori che ospitano il socket del modulo Bluetooth. La realizzazione modulare è stata voluta sia per interporre una scheda tra il modulo RF e il microcontrollore (onde evitare interferenze tra le onde elettromagnetiche e le piste di rame della scheda o tra le bobine dei relé e il modulo radio) sia per adattare la base ad eventuali altri moduli RF.
Il "cervello" della scheda è la MCU utilizzata, di tipo PIC-16F877A (indicata nello schema con U1); di essa, i pin di I/O della porta B vengono utilizzati per gestire lo stato delle uscite (relé, LED e potenza) attraverso l'integrato ULN2003 (un array di 7 stadi open-collector Darlington appositamente progettato per pilotare diversi tipi di carico tra cui relé, solenoidi, LED, motori DC, ecc…) che viene utilizzato per il comando dei quattro relé e delle due uscite open-collector (LEDOUT 1 e 2). Queste ultime due uscite sono già state predisposte per comandare dei LED: infatti hanno in serie altrettante resistenze da 820 ohm; se desiderate comandare utilizzatori che non richiedono resistenze (ad esempio ulteriori relé) dovete cortocircuitare R17 ed R18.
Le due uscite di potenza sono invece realizzate con i MOSFET BUZ11 (o equivalenti) T1 e T2, utilizzati per chiudere a massa il pin 2 delle morsettiere di uscita in modo da alimentare gli utilizzatori che vi vengono collegati (il pin 1 è alimentato con il +12 V). I transistor scelti sopportano correnti di drain fino a 30 ampere, quindi sono sufficienti per pilotare la quasi totalità dei carichi senza problemi. Per i nostri test abbiamo collegato dei carichi induttivi (i più critici per questo tipo di uscita) senza mai riscontrare problemi, riscaldamenti anomali o cedimenti dei componenti; ciò, grazie anche al fatto che i MOSFET hanno integrato il diodo di protezione dalle tensioni inverse tra drain e source.
I quattro ingressi digitali vengono gestiti dalla porta D del PIC; ciascuno è protetto da una coppia di Zener da 3,3 V (uno blocca le extratensioni positive e l'altro quelle negative) e dispone di un resistore di pull-up che quando è aperto lo mantiene a livello logico alto (cortocircuitando i relativi piedini delle morsettiere, gli ingressi vengono portati al livello logico basso).
Nel circuito è presente un ponticello sulla linea di reset, collegato alla porta D7 della MCU, utilizzato durante il normale funzionamento per azzerare dall'esterno lo stato delle uscite del PIC.
E passiamo adesso alla scheda superiore, che contiene uno zoccolo formato da due file da 14 contatti a passo 2,54 mm, da utilizzare per inserire il socket del modulo Bluetooth RN-42; confrontando le piedinature notiamo che vengono utilizzati solo i due pin relativi alla porta UART di comunicazione (TX e RX, pin 5 e 7), i due pin di alimentazione (3V3 e GND, pin 14 e 13) ed il collegamento di reset (pin 1). La porta UART del socket è chiaramente collegata all'analoga porta di comunicazione del PIC ed utilizzata da quest'ultimo per gestire il collegamento con il modulo Bluetooth (serve ad inviare comandi di configurazione dell'RN-42 e leggere/trasmettere i dati da inviare) mentre il pin di reset è collegato al piedino D6 del PIC e gestito per effettuare il



La scheda base.



Elenco Componenti:

R1: 220 ohm
R2: 1,5 kohm
R3: 33 kohm
R4, R5: 100 kohm
R6÷R10: 10 kohm
R11: 820 ohm
R12: 820 ohm
R13: 3,3 kohm
R14: 100 kohm
R15: 3,3 kohm
R16: 100 kohm
C1: 470 μF 35 VL elettrolitico
C2: 100 nF ceramico
C3: 220 μF 16 VL elettrolitico
C4: 100 nF ceramico
C5: 100 nF ceramico
C6: 15 nF ceramico
C7: 15 nF ceramico
D1: SD103
D2: 1N4007
D3: 1N4007

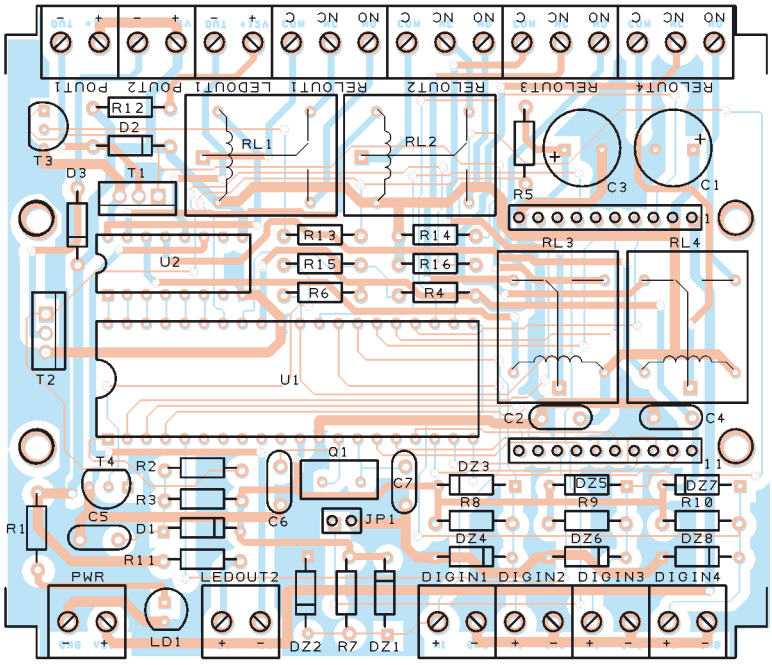
reset hardware del modulo.
Nella scheda superiore, oltre alle connessioni con la scheda socket abbiamo il blocco alimentatore, che partendo dalla morsettiera PWR preleva i 12 volt in continua (forniti da un alimentatore esterno) tramite un fusibile (F1) di sicurezza e un diodo (D1) per proteggere da eventuali errori di inversione di polarità, quindi da essi ricava, mediante il regolatore LM1085IT-3.3, una componente continua stabilizzata al valore di 3,3 V. Con questa tensione alimenta sia il modulo Bluetooth, tramite lo zoccolo per il socket, sia il microcontrollore PIC della scheda base, attraverso il connettore con cui la basetta si applica a tale scheda. Sia la linea dei 12 volt, sia quella dei 3,3 volt, è abbondantemente filtrata da diversi condensatori che stabilizzano le tensioni durante i picchi di assorbimento che possono verificarsi durante le fasi di trasmissione del modulo Bluetooth, l'attivazione delle bobine dei relé o l'applicazione di eventuali carichi collegati alle uscite di potenza.

IL FIRMWARE DEL PIC

Bene, ora passiamo al software del sistema, che si compone di due parti: un firmware da caricare nel microcontrollore affinché la nostra unità di I/O possa assolvere ai propri compiti, ed un programma che gira in ambiente Android e va caricato nello smartphone perché questo possa comandare la nostra unità.
Descriviamo per primo il firmware caricato nel microcontrollore, che dopo la fase iniziale di avvio svolge in parallelo due compiti principali: la gestione della comunicazione Bluetooth (ricezione e trasmissione da parte del modulo RN-42) ed il controllo dei pin di I/O (lettura degli ingressi digitali ed impostazione dello stato delle uscite).
Nella fase di avvio vengono configurate le porte di input/output del PIC ed in seguito, tramite opportuni comandi, il firmware provvede a programmare il modulo RN-42 per instaurare il collegamento Bluetooth. Facciamo notare che tra i comandi che il microcontrollore impartisce, ci sono la selezione dell'ingresso nella modalità Command Mode (comando $$$), il reset delle impostazioni a quelle di fabbrica (comando SF,1), la selezione dell'operation mode come slave (comando SM,0), la programmazione del nome Bluetooth (comando SN, Nuova Scheda I/O), la programmazione del PIN Bluetooth (comando SP,1234) ed infine il reboot software (comando R,1).
Dopo la fase di avvio, il firmware entra nel ciclo infinito principale di esecuzione in cui viene testato l'ingresso di reset ed eventualmente azzerato lo stato logico delle uscite, ma anche gestita la comunicazione. Essendo impostata come slave Bluetooth, la nostra periferica non può iniziare una connessione di propria iniziativa ma può solo accettare

DZ1÷DZ8: Zener 3,3V 400mW
U1: PIC16F877A (MF1005IO)
U2: ULN2003
T1: BUZ11
T2: BUZ11
T3: BC547
T4: BC547
Q1: Quarzo 4 MHz
LD1: LED 5 mm verde
RL1: Relé 12V 220V/10A
RL2: Relé 12V 220V/10A
RL3: Relé 12V 220V/10A
RL4: Relé 12V 220V/10A

Varie:
- Morsetto 2 poli passo 5 mm (9 pz)
- Morsetto 3 poli passo 5 mm (4 pz)
- Zoccolo 8+8
- Zoccolo 20+20
- Strip femmina 10 poli (2 pz.)
- Circuito stampato



(o rifiutare) richieste provenienti dall'unità master, che nello specifico è lo smartphone; per questo motivo il firmware è sempre in attesa di ricevere dati dal modulo RN-42 e, in caso ne riceva, esegue delle verifiche di richiesta di nuova connessione.
Quando tali richieste arrivano, la connessione viene accettata e vengono letti ed analizzati i caratteri ricevuti per verificare se si tratta di un comando riconosciuto dal protocollo definito. In caso affermativo vengono eseguite le operazioni richieste (attivazione/disattivazione uscite oppure lettura ingressi) e trasmesse le opportune risposte.
Ultima segnalazione degna di nota è la presenza di un time-out (di circa 5 secondi) utilizzato per verificare eventuali "false" connessioni Bluetooth; infatti se il PIC rileva che da un collegamento correttamente stabilito non arrivano dati per un periodo di circa 5 secondi, ne forza la chiusura onde evitare che se il link si blocca tenga inutilmente occupato il canale e l'hardware della scheda.

SOFTWARE PER ANDROID

Il software sviluppato per smartphone Android permette il collegamento e la gestione da remoto della scheda: è possibile verificare lo stato corrente degli ingressi digitali, ma anche impostare o verificare la condizione (ON o OFF) delle uscite secondo le proprie necessità.
All'avvio, dopo una schermata iniziale di presentazione del software (**Fig. 1**) viene avviata la scansione per il rilevamento di tutti i dispositivi Bluetooth presenti all'interno della zona di copertura dello smartphone, che vengono elencati in una lista come mostrato nella **Fig. 2**. Per connettersi con la nostra unità



Fig. 1

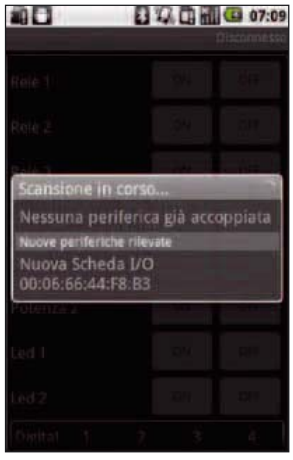
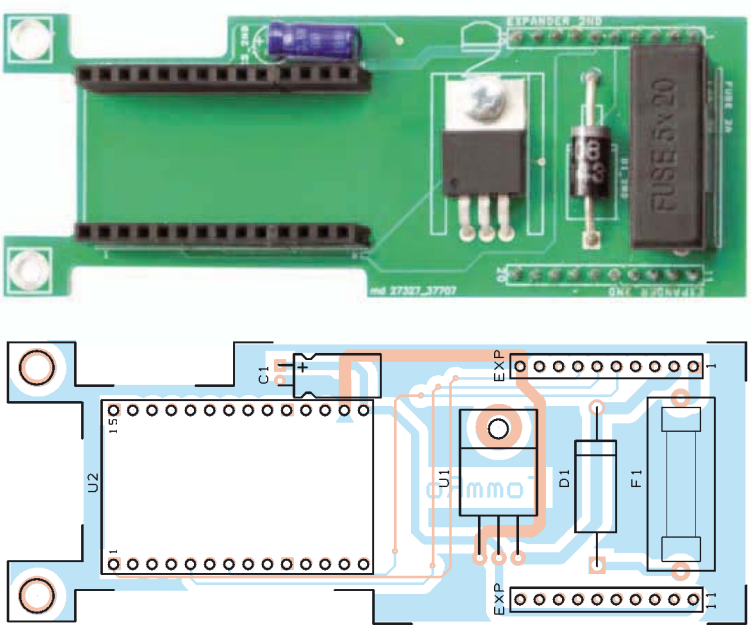


Fig. 2



Fig. 3

[piano di montaggio **INTERFACCIA**]



Elenco Componenti:

C1: 10 μF 63 VL elettrolitico
D1:1N5408
U1: LD1086-3.3
U2: Scheda Bluetooth
F1: Fusibile 2A

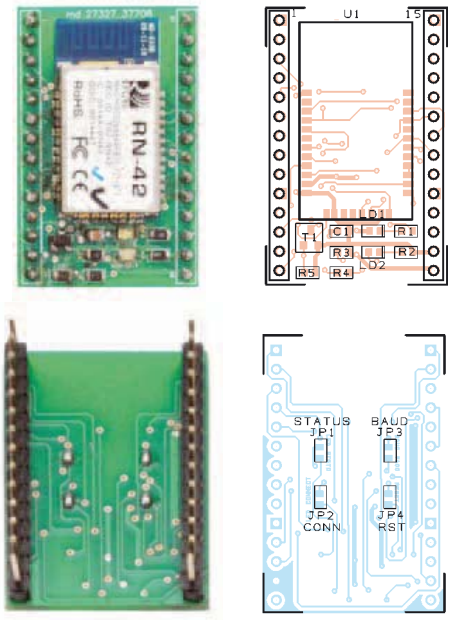
Varie:
- Portafusibile da CS
- Strip femmina 14 poli (2 pz.)
- Strip maschio 10 poli (2 pz.)
- Vite 10 mm 3 MA
- Dado 3 MA
- Circuito stampato

[piano di montaggio **SOCKET**]

Elenco Componenti:

R1: 220 ohm (0805)
R2: 220 ohm (0805)
R3: 1 kohm (0805)
R4: 3,3 kohm (0805)
R5: 10 kohm (0805)
C1: 100 nF multistrato
LD1: LED arancio (0805)
LD2: LED verde (0805)
U1: RN-42

Varie:
- Strip femmina 14 poli (2 pz.)
- Circuito stampato



di I/O basta fare clic sulla voce corrispondente. In ogni momento, tramite l'apposito pulsante virtuale, potete rieseguire la scansione e l'aggiornamento della lista.
A questo punto si passa nella schermata principale del programma (**Fig. 3**); partendo dall'alto viene visualizzato lo stato delle uscite a relé, di quelle di potenza e dei due output LED. In ultimo, nella parte finale viene mostrato lo stato dei quattro ingressi digitali.
Per ogni singola uscita viene utilizzata la seguente convenzione: sulla sinistra è presente un'etichetta di testo che indica lo stato corrente; più a destra ci sono invece due pulsanti (etichettati con ON e OFF di cui solo 1 risulta abilitato) utilizzabili per modificare lo stato corrente.
Se l'uscita è attiva, l'etichetta di testo riporta ON e risulta abilitato il pulsante OFF, altrimenti l'etichetta riporta OFF ed è abilitato il pulsante ON. Nella fotografia in **Fig. 3** sono attivi i relé 1 e 3, le uscite di potenza 1 e LED 1, mentre sono disattivi i relè 2 e 4 e le uscite di potenza 2 e LED 2.
Per ogni ingresso digitale è presente una sola etichetta di testo riportante lo stato corrente: ON nel caso che l'ingresso del PIC risulti alto; OFF in caso contrario (prendendo ancora come riferimento la **Fig. 3**, si vede che tutti e 4 gli ingressi sono ON).
Tornando ancora alle uscite, per richiedere la modifica dello stato corrente è sufficiente fare clic sul relativo pulsante attivo; a questo punto il comando selezionato viene inviato alla scheda ed, al completamento, aggiornata l'interfaccia grafica.
Infine, notate la presenza di un timer di aggiornamento della grafica: ciclicamente il software si collega all'unità di I/O, ne legge

La comunicazione con lo **smartphone**

Il Bluetooth prevede due distinte classi di dispositivi: master e slave; della prima fanno parte tutti quei dispositivi che, di propria iniziativa, possono avviare una nuova connessione, mentre appartengono alla seconda classe quelli che possono instaurare una connessione solamente dietro richiesta di un'unità master. Facendo un parallelo con la comunicazione TCP/IP, i master sono equivalenti ai client, mentre gli slave corrispondono ai server. Nel nostro caso lo smartphone si comporta da master, mentre l'unità di I/O da slave.
Il protocollo prevede delle trame strutturate come descritto qui di seguito.

- Primo byte: tipo di operazione (lettura o impostazione stato I/O scheda); GETNFO_CMD (byte 0x41) per lettura; SETNFO_CMD (byte 0x42) per l'impostazione.
- Secondo byte: identificativo tipo della risorsa. Sono previste 5 possibilità: stato completo scheda (STATUS_CMD; byte 0x40); singola uscita a relé (OUTRLE_CMD; byte 0x41); singola uscita di potenza (OUTPOT_CMD; byte 0x42); singola uscita a LED (OUTLED_CMD; byte 0x43) ed infine singolo ingresso digitale (INPDIG_CMD; byte 0x44).
- Terzo byte: identificativo numero della risorsa (solo per comandi singole uscite o ingressi).
- Quarto byte: stato di impostazione risorsa (solo per settaggio uscite).

Il protocollo prevede inoltre che, ad ogni richiesta di comandi proveniente dal master, lo slave risponda con pacchetti di conferma.

lo stato corrente e in conseguenza alle informazioni ricevute, aggiorna l'interfaccia grafica (in particolare se è variato lo stato degli ingressi digitali, anche la relativa sezione grafica viene modificata).

REALIZZAZIONE PRATICA

Il sistema è facilmente realizzabile grazie al fatto che le due schede più grandi da cui è formato impiegano componentistica a montaggio tradizionale; l'unico elemento in SMD è il socket adattatore RN-42, che però può essere acquistato in Futura Elettronica (*www.futurashop.it*) già montato e collaudato. Dopo esservi procurati gli stampati (trovate le tracce nel nostro sito *www.elettronicain.it*) se volete fare da voi partite proprio con il modulo socket; per il montaggio sappiate che vi serve un saldatore a punta molto sottile e del filo di lega saldante del diametro di 0,5 mm.
Dunque, montate prima i passivi (per saldarli bene spalmate del flussante a bassa densità sulle piazzole, prima di appoggiare i componenti e stagnarli) e poi il modulo, che si stagna abbastanza facilmente, dato che ha i contatti laterali: basta centrarlo nelle rispettive piazzole, bloccarlo stagnando un pad per lato e poi saldare i restanti contatti procedendo alternativamente con i due lati. Se avete a disposizione una lente d'ingrandimento, verificate di non aver fatto colare dello stagno tra due pad adiacenti e, nel caso abbiate fatto un cortocircuito, rimuovetelo con il saldatore e della trecciola dissaldante; fate la stessa verifica con tutti i componenti SMD. Per vedere meglio le saldature vi suggeriamo di pulirle con dello spray rimuovi flussante o dell'alcol isopropilico.
Passate ora ai due circuiti che restano: partite saldando per primi i componenti più bassi e via-via procedete, badando, per gli elementi polarizzati, a rispettare l'orientamento che vedete indicato nei piani di montaggio della tre schede. In ultimo vanno montati i connettori, ricordando che la scheda base e quella che le sta sopra si connettono tramite 2 connettori ciascuno a 10 poli: nella scheda base vanno montati (lato componenti) due connettori SIL femmina a passo 2,54 mm, mentre in quella sopra bisogna stagnare, ma dal lato saldature, due pin-strip da 10 poli ciascuno. La scheda di interfaccia deve avere montati due connettori SIL femmina a passo 2,54 mm da 14 poli l'uno per ospitare il socket; questi strip devono prendere posto dal lato componenti.
Una volta terminato il montaggio dei componenti e verificato che sia tutto a posto, nella scheda base inserite il microcontrollore già programmato con l'apposito firmware che potete scaricare dal nostro sito web *www.elettronicain.it*. Fatto ciò potete assemblare il sistema: come si vede dalle foto in queste pagine, sotto va la scheda base, sopra quella con l'alimentatore (interfaccia) e in cima il socket; siccome le due schede hanno i fori di fissaggio, il nostro suggerimento è di utilizzare colonnine distanziali da 25 mm per ottenere un insieme più solido. Assemblata la parte bassa, sulla scheda d'interfaccia inserite il socket. Ora il sistema è pronto, almeno sul piano hardware. Non vi resta che caricare l'applicazione software (trovate anche questa sul nostro sito web) nel vostro smartphone e fare subito qualche prova.



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono di facile reperibilità. Il firmware ed il software Android nonché i master dei CS sono scaricabili gratuitamente dal sito della rivista (www.elettronicain.it). Il Socket Bluetooth già montato e comprensivo del modulo RN-42 (cod. FT1018M) è disponibile al prezzo di Euro 29,00 mentre il singolo modulo RN-42 costa 21,00. Tutti i prezzi si intendono IVA compresa.

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

ARDUINO Flash Mob

In occasione della presentazione del libro "Arduino Uno, Programmazione avanzata e Librerie di sistema", abbiamo organizzato un hacker space: ecco la cronaca ed i progetti con i quali si sono cimentati i partecipanti.

È qui la FESTA!

a cura della Redazione

Più che un evento è stata una grande festa, l'Arduino-FlashMob che abbiamo organizzato domenica 24 giugno, una festa che ha visto la presenza di tantissime persone molte delle quali hanno partecipato attivamente ai laboratori ed alla gara tra creativi. Inizialmente l'evento era stato pensato per la presentazione del nuovo libro di Simone Majocchi "Arduino UNO, Programmazione avanzata e Librerie di sistema" da noi edito: l'intenzione era quella

www.futurashop.it
www.elettronicain.it



di fare la classica presentazione con un piccolo buffet, i pasticcini, eccetera. Poi, a poco a poco, si sono aggiunte altre attività ed alla fine la situazione ci è letteralmente scappata di mano ed il tutto è diventato un evento vero e proprio, con area espositiva (nostra e di altri makers), Scuola di Saldatura, Area Riciclo, area demo per stampanti 3D e CNC e, soprattutto, una interessante gara tra team di creativi con in palio una stampante 3D.

In queste pagine il racconto della giornata con le foto dei momenti più significativi. Negli spazi di Futura Elettronica, in Gallarate, abbiamo dunque allestito una serie di laboratori didattici, dimostrazioni e aree ludiche con l'intento di attrarre il maggior numero possibile di appassionati di elettronica, makers, creativi, ma anche di semplici curiosi, affascinati da questo mondo e desiderosi di imparare. Questo è stato l'Arduino Flash Mob, un nome che è tutto un programma: ARDUINO perché questa piattaforma di prototipazione rapida è alla base di quel nuovo rinascimento tecnologico che ha al centro il fare, specialmente il fare tecnologico e creativo; FLASH perché è un'idea che ci è venuta all'improvviso e che altrettanto rapidamente abbiamo messo in pratica; MOB: perché tutti i

Spilletta SmiLED

Scuola di saldatura, evento di Arduino Flash Mob dedicato a chi muove i primi passi nel mondo dei montaggi elettronici, ha proposto ai partecipanti due esercitazioni distinte per livello di difficoltà: due circuiti che vi riproponiamo e che potete realizzare da voi. Il primo ed anche il più semplice consiste in un gadget grande come una spilla ed essenziale, grazie al fatto che tutta l'elettronica che serve a farlo funzionare sta in due LED; lo schema è illustrato in questo riquadro e, come vedete, consta di una pila, un interruttore e due diodi luminosi RGB che, una volta chiuso l'interruttore, si accendono inizialmente a luce rossa, per poi virare verso il lilla, il viola, il blu, il verde, il giallo e poi di nuovo il rosso, ricominciando il ciclo.
Ma come si fa, vi chiederete, a far cambiare la colorazione di due LED senza alcun circuito elettronico? Ebbene, la spiegazione sta nei particolari diodi luminosi utilizzati, che integrano un circuito di controllo e che in realtà sono formati ognuno da tre LED emittenti luce, uno rossa, uno verde e l'altro blu. Un LED (Light Emitting Diode) è un particolare diodo in grado, se polarizzato direttamente (ossia con l'anodo positivo e il catodo negativo) di emettere dal lato P della giunzione una luce visibile; tutti i diodi a giunzione in silicio e in germanio, in polarizzazione diretta emettono una radiazione luminosa, però la lunghezza d'onda cade al di sopra del visibile (tipicamente intorno agli 800÷1.000 nanometri) mentre i LED hanno la caratteristica di emettere luce nel campo visibile. Ciò grazie ai semiconduttori con cui vengono realizzati (tipicamente arseniuro di gallio).
Ognuno dei nostri LED RGB in realtà contiene tre giunzioni con un terminale in comune, che vengono accese da un generatore d'impulsi PWM comandato da un circuito logico in grado di variare la larghezza degli impulsi di comando di ciascuna giunzione in modo da ottenere i tre colori base e la dissolvenza. I dispositivi impiegati si presentano come dei comuni LED e quindi hanno due soli terminali: uno positivo (il più lungo) ed uno negativo (il più corto, che sta dalla parte smussata del contenitore); si alimentano con 3,2 V ed assorbono, a tale tensione, 15÷20 mA di corrente. La massima luminosità (nella media dei colori) ottenibile in queste condizioni è 1.200 mcd. LED di questo tipo vengono chiamati LED RGB rainbow, ossia diodi luminosi rosso-verde-blu arcobaleno, proprio per la loro prerogativa di produrre tutte le tonalità di luce quando alimentati; in commercio ne esistono in due versioni in base alla velocità con cui il chip di controllo fa compiere la transizione tra i colori: veloci (fast) e lenti (slow). Inoltre si trovano con il contenitore trasparente (tipo vetro) o bianco opaco.
Nel nostro caso, alimentiamo i diodi in parallelo direttamente con la tensione di una pila a bottone di tipo CR2032; in mezzo c'è solo un deviatore (SW1) inserito per poter accendere e spegnere i LED. Non è stata prevista alcuna resistenza in quanto la tensione che forniamo è appena sufficiente a far illuminare i LED e pressappoco uguale a quella minima richiesta; pertanto inserire un resistore di caduta sarebbe stato inutile. In linea di massima, i LED da noi utilizzati possono essere sempre alimentati a 3,2 volt purché stabilizzati, senza inserire alcuna resistenza. L'inserimento di una resistenza (consigliato per tensioni superiori, tipo 5 volt) ha l'effetto di

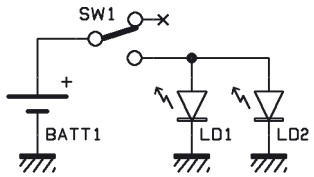


movimenti di questo genere sono improvvisati e spontanei ma debbono essere organizzati nei minimi dettagli per raggiungere lo scopo.

IL PROGRAMMA

Accanto alla presentazione del libro, abbiamo organizzato tre aree dove far svolgere altrettanti eventi: nell'Area Creativi, 5 gruppi di makers, ognuno dei quali composto da 6 persone e contraddistinto da un colore (Rosso, Verde, Giallo, Blu e Bianco), aveva a disposizione un mucchietto di componenti elettronici e di apparecchiature varie, nuove e usate, con cui realizzare un progetto quanto più possibile originale ed innovativo. I team

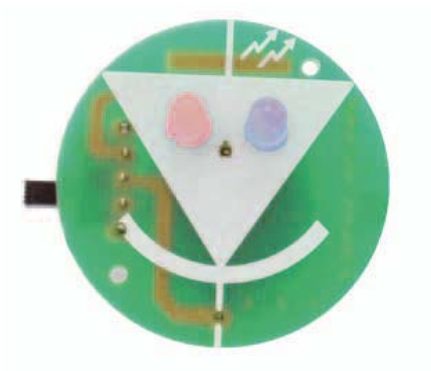
limitare la corrente e variare la velocità di esecuzione della sequenza di colori. Lasciamo adesso gli aspetti teorici e passiamo alla pratica: per costruire questo circuito serve una basetta da realizzare per fotoincisione seguendo la traccia lato rame (il c.s. è un singola ramatura) da noi fornita; la forma definitiva, tonda, si ottiene segando la piastra ramata (già incisa e forata) con un seghetto da traforo in modo da darle la sagoma circolare e poi rifinendo con carta vetrata o con una lima media. Ottenuto lo stampato, montatevi i due LED infilandoli in piedi a fondo nei rispettivi fori e badando che il lato piatto di ciascuno sia rivolto come indicato (obliquo verso i fori riservati all'interruttore); poi, dal lato saldature sistemate il portapile per la pila a bottone CR2032 e il deviatore per l'accensione e lo spegnimento del circuito. A montaggio completato, potete passivare il lato con la pila mediante un dischetto di materiale plastico o di cartoncino, incollato al portapile; potete anche scegliere un piccolo contenitore in plastica sul cui retro disporre una spilla per fissare il gadget alla giacca. *Questo simpatico gadget è disponibile in scatola di montaggio (cod. FT1016K) al prezzo di Euro 5,00. SmiLED è prodotto da Futura Elettronica (www.futurashop.it).*

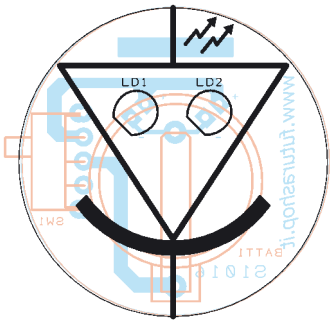


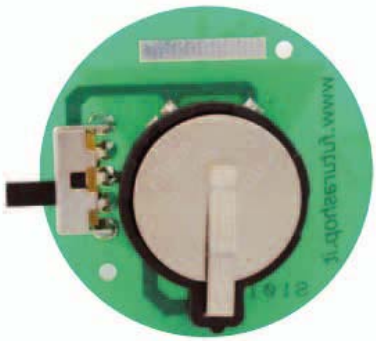
Elenco Componenti:

LD1: LED 5 mm RGB
LD2: LED 5 mm RGB
SW1: Deviatore a slitta 90° da c.s.

Varie:
- Porta batteria CR2032 (BH-001)
- Circuito stampato







avevano a disposizione l'intera giornata con la prospettiva, nel caso di vittoria, di aggiudicarsi una stampante 3D.
Nell'Area Riciclo Materiale, pensata per insegnare a conoscere e riconoscere i componenti elettronici, elettromeccanici e meccanici con i quali vengono realizzati i dispositivi elettronici, vari team di persone si sono alternati attorno ad un "mucchio selvaggio" di apparecchiature usate e schede elettroniche con l'intento di recuperare i componenti funzionanti da fornire ai team dei creativi. Inoltre i partecipanti hanno selezionato il materiale non più utilizzabile in modo da garantirne un corretto smaltimento.
Nell'ultima area - Scuola di Sal-



Ditofono

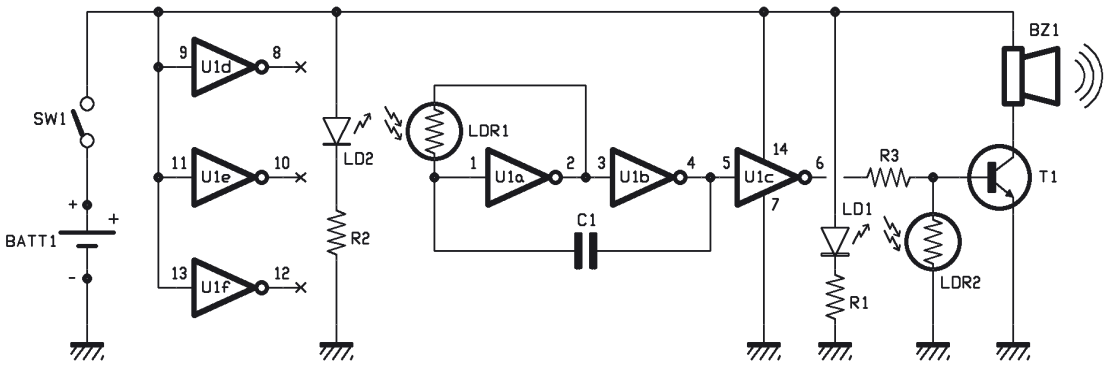
La seconda esercitazione della Scuola di saldatura consiste in un semplice quanto particolare strumento musicale, da suonare con le dita, che perciò abbiamo chiamato "Ditofono": è in grado di generare una nota acustica riprodotta da un altoparlante, della quale facciamo variare tono e volume sonoro semplicemente muovendo due dita davanti ad altrettanti gruppi LED/fotoresistenza. Potete comprendere il funzionamento di questo circuito dando uno sguardo allo schema elettrico, che è incentrato sull'integrato 74HC04, contenente sei porte logiche NOT: tre le utilizziamo, mentre le tre restanti vengono bloccate con l'ingresso a livello alto, in modo da impedire commutazioni logiche casuali originate da disturbi elettrici, le quali potrebbero alterare il funzionamento delle tre NOT che ci interessano. Le NOT sono semplici invertitori della condizione logica all'ingresso (da qui il nome "inverter"): se ricevono uno forniscono zero e se gli si applica 0 logico presentano 1 in uscita. U1a e U1b formano un multivibratore astabile, ossia un circuito elettronico in grado di generare una tensione di forma d'onda rettangolare, che alterna due livelli (uno alto, ossia 1 logico, ed uno basso, ossia lo zero) la cui frequenza dipende, nel nostro caso, dai valori del condensatore C1 e della resistenza inserita tra i piedini 1 e 2 della NOT U1a.



Questo è il funzionamento del multivibratore: partiamo dalla condizione in cui il condensatore è scarico, quindi presenta 0 volt tra i propri estremi, e la porta U1a commuta per prima portando ad 1 logico la propria uscita; tale condizione è verosimile in quanto all'accensione, essendo le due porte concatenate, quella che presenterà per prima uno stato logico certo lo imporrà all'ingresso dell'altra. Quale delle due commuterà per prima dipende dalle caratteristiche costruttive dell'integrato 74HC04 e non è detto che sia la stessa in tutti i circuiti realizzati, ma ciò è irrilevante in quanto imponiamo che U1a porti la propria uscita a livello alto, ma se fosse U1b a farlo per prima, semplicemente cambierebbe la spiegazione, ma non il funzionamento. Dunque, se all'avvio U1a presenta 1 logico sul piedino 2, U1b riceve tale stato e fornisce in uscita il livello basso; ciò rafforza la condizione iniziale, in quando con una differenza di potenziale nulla ai capi del C2, il piedino 1 della U1a è a zero logico. Però la resistenza del fotoresistore LDR1 comincia a caricare C1 con polarità positiva verso il piedino 1, in un tempo che è direttamente proporzionale al prodotto LDR1xC1, ossia la costante di tempo (τ) del bipolo formato dal condensatore e dal fotoresistore. Per come funzionano le NOT contenute nel 74HC04, l'ingresso della U1a riconosce l'uno logico poco prima che sia passato il tempo pari alla costante di tempo τ, dato che la carica del condensatore è ad andamento esponenziale e dopo τ determina una tensione pari a 0,632 volte quella di fine carica, che nel nostro caso è la stessa che alimenta le porte logiche e proviene da una serie di due pile stilo. Dopo questo intervallo, C1 si carica in modo da portare a livello alto l'ingresso della U1a e imporre lo zero all'uscita di quest'ultima, causando la commutazione da zero ad 1 logico del piedino 4 della U1b; adesso il condensatore viene forzato a scaricarsi attraverso la solita LDR1 e a ricaricarsi con polarità opposta, fin quando la tensione sul piedino 1 dell'U1 non scende ad un livello di tensione che la NOT identifica come zero logico. Ciò richiede un tempo pari a quello di scarica del C1 (circa τ) sommato a un'altra

Elenco Componenti

R1: 150 ohm
R2: 150 ohm
R3: 1,2 kohm
C1: 330 nF 63 VL poliestere
LD1: LED 5 mm rosso alta luminosità (L-7113SRD-D)
LD2: LED 5 mm rosso alta luminosità (L-7113SRD-D)
U1: 74HC04B
T1: BC547
AP1: Altoparlante da c.s.
SW1: Deviatore a slitta



90° da CS
LDR1: Fotoresistenza D 5mm (LDR04)

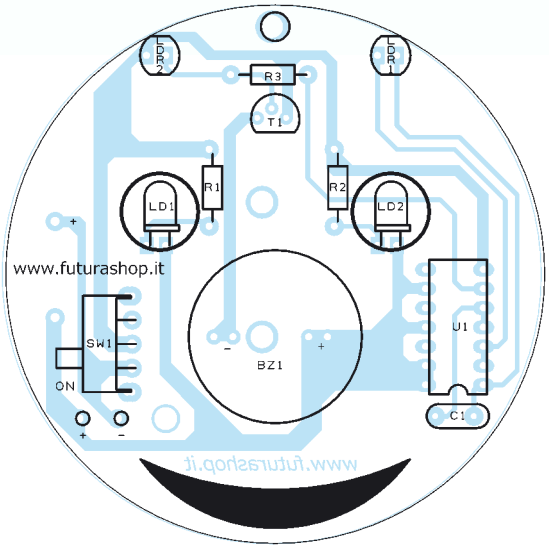
Varie:
- Zoccolo 7+7
- Porta batteria 2xAA

(BH321B)
- Clip batteria 9V (BS-IR)
- Circuito stampato

datura - durante quattro sessioni della durata di un'ora circa (due al mattino e due al pomeriggio) altrettanti gruppi di persone si sono cimentate nel montaggio di alcuni simpatici gadget elettronici, per la precisione quelli descritti in queste pagine. Tutti i dispositivi realizzati sono rimasti

τ, che è all'incirca quanto serve alla tensione sul condensatore per far scendere a zero logico quella sul piedino 1 del 74HC04. Quindi un tempo approssimabile a 2τ.
A questo punto ricomincia tutto da capo e l'uscita della U1b è dunque sede di una tensione rettangolare il cui livello si inverte ogni 2τ, quindi possiamo dire che la frequenza dell'onda generata è pari a $1/2\tau$, ovvero 1/2(LDR1xC1); siccome LDR1 è un resistore variabile in funzione della luce e di fronte ad esso disponiamo (ad un paio di centimetri di distanza) un LED (LD2) acceso mediante la resistenza R2, la frequenza varierà in funzione della quantità di luce che dal diodo luminoso passerà al fotoresistore. Mettendo un dito tra LD2 ed LDR1 e spostandolo in modo da oscurare più o meno la superficie sensibile del fotoresistore, possiamo spostare la frequenza e con essa la tonalità della nota generata.
Proseguiamo l'analisi del circuito con U1c, utilizzata come buffer per trasferire il segnale rettangolare senza caricare il multivibratore (cosa che altererebbe la frequenza generata); l'uscita di tale NOT pilota (tramite la resistenza R3) il transistor NPN T1, usato come amplificatore di potenza per pilotare un cicalino piezo. Questo trasduttore trasforma il segnale elettrico in una nota acustica, alla frequenza di lavoro del multivibratore.
Il controllo del volume viene ottenuto così: un secondo LED (LD1) puntato contro LDR2 illumina la superficie sensibile di quest'ultima, portandone al minimo la resistenza e abbassando (perché fa partitore con R3) quanto possibile la tensione che giunge tra base ed emettitore del T1. Senza ostacoli, la resistenza assunta da LDR2 impedisce la conduzione del transistor.
Mettendo il polpastrello di un dito tra LD1 ed LDR2, in base a quanto oscuriamo il fotoresistore, la tensione base-emettitore del T1 cresce e con essa l'ampiezza della tensione che alimenta il cicalino. Passiamo adesso alla costruzione: ottenuto lo stampato, montate per prime le tre resistenze, cui seguirà lo zoccolo per l'integrato, da posizionare con la tacca di riferimento disposta come indicato nel piano di montaggio. Proseguite con il condensatore e con i LED, i cui terminali dovranno essere piegati a 90° rispetto al corpo; il più corto (il catodo, ossia il negativo) deve entrare nel foro vicino al lato dritto del simbolo, mentre il più lungo (+, ovvero l'anodo) va introdotto dal lato con il dente. Anche LDR1 e LDR2 vanno montati piegandone i terminali ad angolo retto e facendo in modo che il lato sensibile sia rivolto verso il LED corrispondente. Ora montate T1, ricordando che il suo lato piatto deve essere rivolto all'interno dello stampato. In ultimo, dovete saldare il cicalino e il deviatore SW1.
Ora girate il circuito dal lato saldature e fissate il portapile con una vite 3MA più dado, quindi saldate il filo rosso alla piazzola "+" e quello nero alla "-", vicino all'interruttore SW1. Concludete inserendo due pile stilo. Spostando l'interruttore in posizione



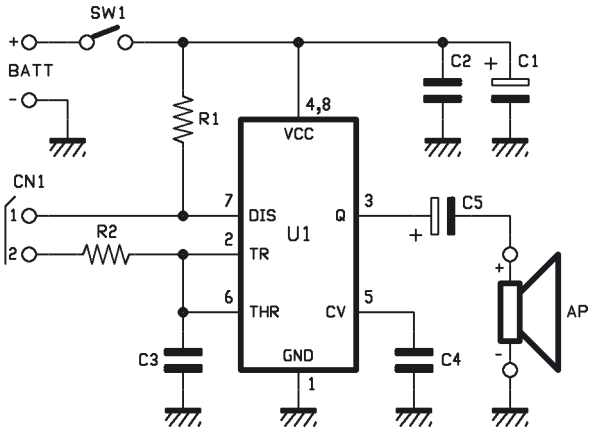


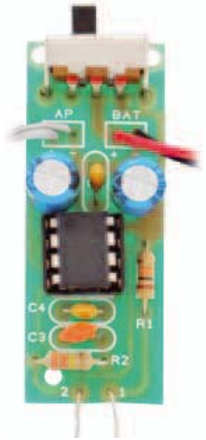
di acceso, vedrete i LED illuminarsi ma il circuito non emetterà alcun suono; appoggiando il circuito dalla parte del portapile e bloccandolo con le mani, muovete le dita tra i LED e le fotoresistenze che si trovano di fronte e noterete cosa si possa fare con uno strumento così semplice. Rammentate che il gruppo LED-fotoresistore di sinistra regola il volume d'ascolto (nullo se non mettete il dito) mentre quello di destra imposta la tonalità (acuto senza interporre il dito e grave se il dito interrompe il fascio di luce).
Questo simpatico gadget è disponibile in scatola di montaggio (cod. FT1014K) al prezzo di Euro 12,00. Il "Ditofono" è prodotto da Futura Elettronica (www.futurashop.it).

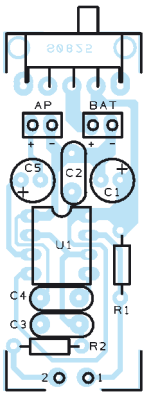
di proprietà di chi li ha montati, offerti gratuitamente da Futura Elettronica. È stata, quella della Scuola di Saldatura, un'occasione per imparare ad utilizzare un saldatore, conoscere alcuni componenti elettronici di base e provare la soddisfazione di aver realizzato un circuito con le pro-

Drawdio: la matita che suona!

Realizzate il circuito, fissatelo con lo scotch alla cima di una matita, quindi collegate i punti 1 e 2 del CN1 a un filo avvolto a spirale nell'impugnatura e ad una puntina infilata nella parte alta della matita in modo da farle toccare la mina; disegnate una linea spessa su un foglio di carta, toccandone un'estremità mentre impugnate la matita dove è avvolto il filo. Come per incanto, l'altoparlante comincerà a riprodurre un suono la cui tonalità cambierà in base allo spessore del tratto e alla forma disegnata. Succede perché il circuito è un astabile basato sul timer NE555 e la mina della matita fa da resistenza, che toccando la traccia (fatta di grafite lasciata quando disegniamo) sul foglio, dove avete appoggiato un dito, chiude il circuito con il vostro corpo. La frequenza prodotta dal 555 cambia con il variare della resistenza formata dalla mina e dal vostro corpo. *Il circuito (già descritto nel fascicolo 141) è disponibile in kit (FT825K) al prezzo di Euro 7,00; ne trovate la sequenza di montaggio sul sito www.futurashop.it.*







Elenco Componenti:

R1: 10 kohm
R2: 330 kohm
C1: 100 µF 16 VL elettrolitico
C2: 100 nF multistrato
C3: 680 pF ceramico
C4: 100 nF multistrato
C5: 100 µF 16 VL elettrolitico

U1: NE555
SW1: Deviatore a slitta 90°

Varie:
- Zoccolo 4+4
- Altoparlante (MLS1)
- Clip per batteria 9V
- 20 cm di filo elettrico
- Circuito stampato





prie mani. Infine, in un apposito spazio, abbiamo esposto alcune delle nostre realizzazioni più innovative, che i visitatori hanno avuto modo di vedere in funzione e, in parte, anche di provare in pratica.

L'EVENTO, MINUTO PER MINUTO

Ed ecco la cronaca della giornata, con partecipanti ai laboratori e visitatori che fin dal primo mattino si sono accalcati ai cancelli della sede di Futura Elettronica. I primi hanno ricevuto in omaggio la T-shirt stampata per l'occasione e la spilla luminosa SmiLED, oltre al braccialetto identificativo.
Mentre inizia la Scuola di Saldatura ed anche il gruppo di riciclo si butta a capofitto negli scatoloni di materiale da recuperare, le squa-



dre dei Creativi iniziano a valutare quale progetto realizzare con il materiale messo a disposizione. Dopo circa un'ora di brainstorming, la decisione è stata presa e i progetti che i team devono realizzare sono stati definiti.
I visitatori, intanto, appena hanno potuto si sono recati a visitare l'Area Makers dove abbiamo esposto alcuni nostri progetti già ultimati o in corso di lavorazione; in particolare, c'erano la nostra autovettura fotovoltaica Aussie, il nuovo Segway (in fase di test), l'UAV di cui ci siamo occupati l'anno scorso, il quadricottero, alcune applicazioni con Arduino, il labirinto comandato dalla pedana (Swinging Labyrinth) pubblicato nel fascicolo di giugno, la nostra CNC e due stampanti 3D. Alle nostre, abbiamo aggiunto le realizzazioni di alcuni makers della zona: dalla motoretta elettrica, alla mega tastiera di Massimo Meli, dalla CNC realizzata con un Lego NXT, ad un bellissimo modello di ruspa radiocomandata.

CREATIVI AL LAVORO!

Dopo aver definito i progetti da realizzare, i Creativi si sono messi al lavoro cercando i vari pezzi necessari, nuovi o di recupero: la Squadra Gialla sulla realizzazione di un sistema luminoso controllato dai suoni ricavato da una nuovissima lampada Ikea; la Squadra Rossa su un lanciatore che simula il gioco del basket con tanto di molla realizzata con una pompetta succhiastagno ed un display per visualizzare il punteggio. La Squa-

Un libro, un evento





Arduino Flash Mob era stato pensato inizialmente come l'occasione per presentare al pubblico il volume "Arduino UNO, Programmazione avanzata e Librerie di sistema" anche se poi è diventato quell'evento tecnologico e in parte anche ludico descritto in queste pagine. In ogni caso una parte della giornata è stata dedicata proprio alla presentazione del libro: poco prima di mezzogiorno Simone Majocchi, l'autore del libro, ha parlato della sua ultima fatica e firmato le copie -ancora fresche di stampa- a tutti i partecipanti che hanno preso parte alla gara tra "creativi".
Il volume, cui anche Massimo Banzi ha voluto contribuire con una introduzione particolarmente affettuosa, è il naturale seguito del già affermato "Primi passi con Arduino", rivolto a quanti, partendo da zero, vogliono imparare i rudimenti della programmazione, utilizzando questa innovativa piattaforma Made in Italy. Chi si è avvicinato al mondo di Arduino con il nostro primo libro dedicato a questo mondo, ha ora la possibilità di proseguire nell'apprendimento della programmazione e degli aspetti di interfacciamento di questa piattaforma attraverso la presentazione delle dieci librerie standard incluse nel sistema di sviluppo, per ognuna delle quali è disponibile un'applicazione che vuole essere punto di partenza da cui trarre ispirazione.
Il libro inizia con una parte teorica e illustrativa dedicata a shield e librerie, quindi propone, in tanti capitoli quante sono le librerie, progetti che usano ciascuno la sola libreria oggetto del capitolo.
Ricordiamo che per entrambi i libri, Futura Elettronica mette a disposizione il materiale necessario alla realizzazione degli esperimenti descritti, rendendo ancora più agevole l'apprendimento di questa materia.







dra Bianca si è impegnata nella realizzazione di una speciale batteria musicale controllata da remoto e capace di scandire il ritmo della musica, mentre la Blu è partita con la realizzazione di una serra con sensori di vento e luce che agiscono sull'aerazione e sull'irrigazione delle piantine. Infine, la Squadra Verde (proveniente da Padova) ha lavorato a un sistema di distribuzione del cibo comandato da remoto.
Accanto alla competizione tra Creativi, si sono succedute quattro sessioni di Scuola di Saldatura, durante le quali giovani e meno giovani hanno montato e testato un piccolo circuito elettronico, imparando a riconoscere i componenti, leggere uno schema elettrico e pratico e, soprattutto, a saldare. Oltre alla spilla SmiLED, gadget dell'evento, i gruppi hanno realizzato il "Ditofono", strumento elettronico che si suona con due dita che controllano il volume e la tonalità.

La Scuola di saldatura è stata un vero successo, perché tutti gli "allievi per un giorno" sono riusciti a realizzare il proprio circuito. Proficua è stata anche l'Area Riciclo, dove i partecipanti si sono divertiti a recuperare componenti elettronici da schede non funzionanti o obsolete, tornando a casa con sacchetti pieni di materiale elettronico per i loro esperimenti.

3D & CNC DANNO SPETTACOLO

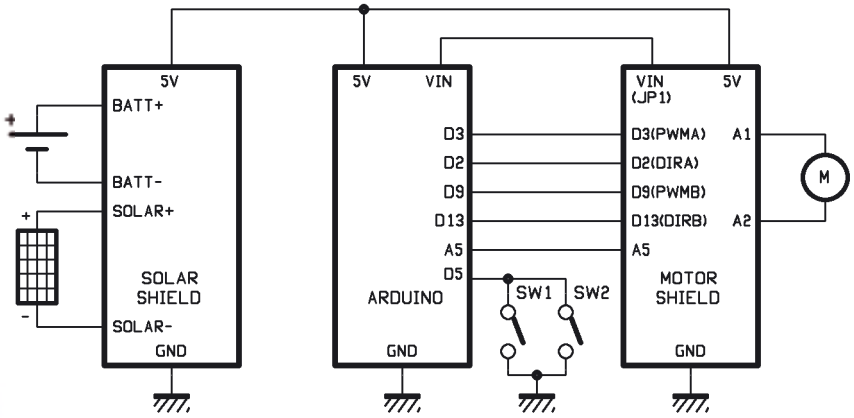
Mentre tutti i gruppi erano completamente assorbiti dai loro progetti, l'area dedicata alle nostre stampanti 3D e CNC è stata attiva e frequentatissima per tutto il giorno, mostrando in funzione la Galileo di Simone Majocchi e il prototipo della nostra RepRap "Prusa" che non è stata da meno. Entrambe le macchine hanno sfornato numerosi modelli nei colori più vari. La nostra "Prusa" (che andrà in premio alla Squadra vincente) è uno dei due modelli che stiamo ingegnerizzando e che sa-





Funivia solare

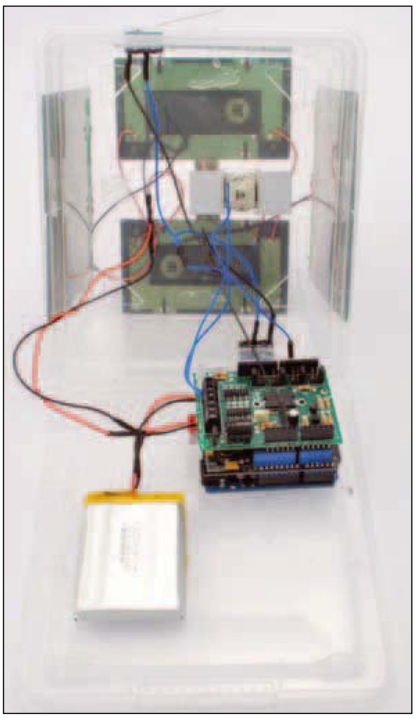




Un pezzo di fune fissato tra due pareti ed una carrucola alla quale è fissata una navicella alimentata da pannelli solari: sono questi gli ingredienti per realizzare una funivia in miniatura da usare in un plastico o come promo pubblicitario.
È lo stesso progetto che i partecipanti all'Arduino Flash Mob hanno potuto vedere in funzione con appeso un cartello di benvenuto che si muoveva avanti e indietro lungo una fune di una decina di metri.
L'hardware si basa su Arduino, un Motor Shield ed un Solar Shield: quest'ultimo, collegato ai pannelli fotovoltaici e provvisto di un regolatore di carica, fornisce 5 V stabilizzati (con cui alimenta Arduino e il Motor Shield) e mantiene in carica una batteria agli ioni di litio. In Arduino gira uno sketch che comanda il motore facendone girare l'albero in un verso e invertendone il moto quando la navicella raggiunge una parete e lo switch fine-corsa viene premuto. SW1 ed SW2 sono tra loro in parallelo, quindi per ottenere l'inversione basta che D5 di Arduino sia portata a livello logico basso.
La navicella incorpora il motore di trazione (un motoriduttore) che aziona una puleggia grazie alla quale può scorrere su una cordicella. L'involucro è un contenitore di policarbonato o di ABS traslucido per alimenti; i pannelli solari vanno collegati in serie a due alla volta. Lo sketch per Arduino è disponibile, insieme ai file del progetto, su *www.elettronicain.it*.

Elenco Componenti:

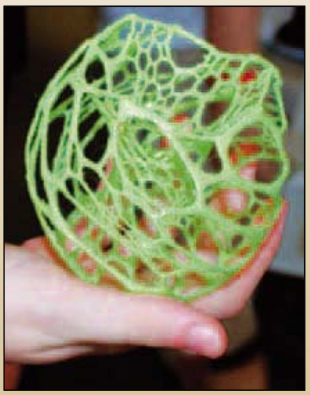
- ARDUINOUNO R3
- SOLARSHIELD
- MOTORSHIELD_FE
- Motore 5V (GM3)
- Deviatore fine corsa (MSWITCH50MM)
- Pannelli solari 2,6V/160mA (6 pz.)
- Batteria 3,7V/2A (PLIB2A)

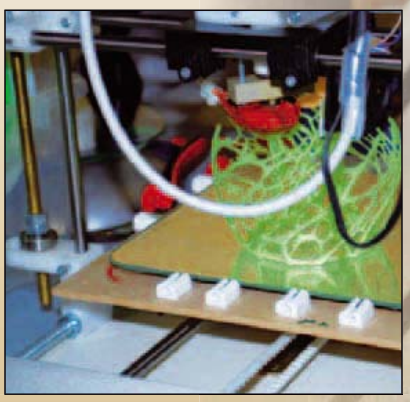


ranno disponibili a breve. Accanto c'era Gabriele Daghetta che, con la nostra CNC, tagliava e incideva le medaglie per la squadra vincente tra quelle dei Creativi.

LA VOLATA FINALE

Dopo la meritata pausa di ristoro (perché non si vive di soli elettroni e, per quanto riesca gradevole alla vista, un circuito non profuma come una salamella...) nel pomeriggio, con sempre maggior frenesia, i gruppi dei creativi hanno continuato a lavorare ai propri progetti. Purtroppo, sarà stato per il caldo e l'afa con cui abbiamo dovuto convivere o perché dopo un'intensa giornata di lavoro sentiva anch'essa il bisogno di un po' di riposo, la nostra rete Wi-Fi ha preso a fare i capricci, mettendo in difficoltà quanti volevano controllare le applicazioni da remoto, e creando in altri casi intoppi di vario genere che hanno ritardato i lavori. Il termine ultimo per la presentazione dei progetti è così slittato di un quarto d'ora, per consentire a tutti di terminare almeno il "grosso" dei progetti.
Alle 16:45 il cronometro si è arrestato: ormai "les jeux sont fait". Chi c'è, c'è... Uno alla volta, i rappresentati delle cinque squadre hanno consegnato e presentato i relativi progetti alla giuria, composta da 5 membri, che ha fatto le sue valutazioni, anche interpellando i





componenti dei team sui particolari sia hardware che software dei progetti da essi realizzati.

AND THE WINNER IS...

Finalmente la giuria emette il proprio verdetto: per un soffio ha vinto la Squadra Bianca, che si è aggiudicata la stampante 3D per il gruppo più creativo (e ritenuto più capace). Il progetto premiato è stato l'ArduBeat, che doveva essere controllato da remoto ma che, alla fine, visti i problemi di rete, è stato controllato direttamente dalla porta USB del PC. Staccata di un solo punto la Squadra Blu, con la sua CyberBotanica, la serra controllata mediante Arduino. Terza la Squadra Rossa, con ArduBasket, quarta la Squadra Verde con XPress ed infine la Squadra Gialla, con Sound Control Lamp. Indipendentemente dal piazzamento, a tutti i team abbiamo regalato una copia del nuovo libro su Arduino autografata dall'autore, l'hacker Passport con lo speciale bollino dell'evento e, naturalmente, gli applausi e le congratulazioni dei presenti.



Via il pubblico, mentre il sole volgeva al tramonto, sfiniti dall'opera alacre e dalla prospettiva di dover smontare tutto e spazzare il cortile, ci ha rincuorato la grande soddisfazione di avere offerto per un giorno al nostro pubblico un'elettronica che dalle immagini bidimensionali della carta stampata è diventata in 3D, sotto tutti gli aspetti...

Ora i riflettori sono spenti e torniamo a scrivere e progettare circuiti nel nostro angolino, ma tavoli, saldatori, robot, bandierine colorate e amenità varie sono lì da parte... casomai ci torni la voglia di ripetere l'esperienza! ■

SMAU

INNOVAZIONE DI CASA NELLE IMPRESE ★



CLOUD

IL CLOUD PUÒ RAPPRESENTARE UN'OPPORTUNITÀ PER LA MIA IMPRESA? COME VALUTARE I BENEFICI REALMENTE OTTENIBILI? QUALI SONO I PROGETTI BASATI SUL CLOUD PIÙ INNOVATIVI IN ITALIA? QUALI SONO I PASSI DA FARE PER ADOTTARE SOLUZIONI CLOUD? COME STA EVOLVENDO L'OFFERTA, CHI SONO I FORNITORI PRINCIPALI E QUALI SONO LE SOLUZIONI PIÙ INNOVATIVE?
A SMAU, PRESENTAZIONE DI CASI DI UTILIZZO, LABORATORI DI APPROFONDIMENTO, WORKSHOP A CURA DI ANALISTI E DOCENTI, PER DARE RISPOSTE CONCRETE A QUESTE E AD ALTRE DOMANDE!

smau 2012

Milano, 17-19 Ottobre – FIERAMILANOCITY

★

Nel 2011 oltre 80.000 imprenditori e manager hanno sfruttato Smau per innovare la propria impresa.
SMAU: I PROTAGONISTI MONDIALI DELL'INNOVAZIONE; IL TALENTO UNICO DEGLI OPERATORI ITALIANI; LE MIGLIORI SCHOOL OF MANAGEMENT. INSIEME. PER TRE GIORNI. A DISPOSIZIONE DELLA TUA IMPRESA.

IN COLLABORAZIONE CON

Gartner | School of Management – Politecnico di Milano – Dipartimento di Ingegneria Gestionale – MIP | SDA Bocconi

CONTATTI
+39.049.8808444
contact@smau.it
www.smau.it

Per Arduino è ora di andare

Impara a programmare con Arduino e i nostri

Per chi parte da zero...



cod. ARDUBOOK1
€ 9,90

Libro "Primi passi con Arduino"

Pensato per i principianti che si avvicinano all'elettronica per la prima volta e per gli hobbisti che non hanno mai affrontato la programmazione. Comprende sei esercitazioni didattiche che consentono di mettere in pratica le nozioni acquisite e, nello spirito che contraddistingue Arduino, rappresentano punti di partenza dai quali trarre ispirazione per arrivare a un buon livello di conoscenza della piattaforma, delle sue possibilità e delle tecniche di programmazione. Pagine: 96.

Sommario:

Capitolo 1: Perché Arduino UNO fa per te
Capitolo 2: Il poker tecnologico vincente
Capitolo 3: Dalle idee al programma
Capitolo 4: Pensato per essere collegato a qualcosa
Capitolo 5: È ora di preparare la nostra cucina tecnologica
Capitolo 6: La nostra prima realizzazione pratica: contiamo fino a tre
Capitolo 7: Basta poco per migliorare le cose
Capitolo 8: Dopo le luci, i suoni
Capitolo 9: Il mondo è a colori
Capitolo 10: Tutti i colori dell'arcobaleno
Capitolo 11: Verso la robotica



Libro "Primi passi con ARDUINO " + scheda Arduino UNO Rev3

cod. ARDUKITBOOK1
€ 29,90

Per chi ha già sperimentato con Arduino...



cod. ARDUADVANCED
€ 22,00

Libro "ARDUINO UNO Programmazione avanzata e Librerie di sistema"

Chi si è avvicinato al mondo di Arduino con il libro "Primi passi con Arduino" ha ora la possibilità di proseguire nell'apprendimento della programmazione e degli aspetti di interfacciamento di questa piattaforma attraverso la presentazione delle dieci librerie standard incluse nel sistema di sviluppo. Con queste dieci librerie abbiamo immaginato altrettante applicazioni di complessità relativa, ma comunque interessanti e anche di qualche utilità. Pagine: 224.

Sommario:

Capitolo 1: Un po' di teoria
Capitolo 2: Preparazione all'ambiente di lavoro
Capitolo 3: EEPROM – Lettura e scrittura nella memoria permanente del microcontrollore. Il progetto proposto (Sequencer con uscite di potenza) utilizza i componenti del set ARDUBOOK2KIT nonché lo shield I/O FT918K e la board Arduino Uno.
Capitolo 4: La libreria Ethernet. Il progetto proposto (Sensori ambientali accessibili da rete) utilizza i componenti del set ARDUBOOK2KIT nonché l'Ethernet Shield e la board Arduino Uno.
Capitolo 5: La libreria firmata. Il progetto proposto (Luci psichedeliche a LED) utilizza i componenti del set ARDUBOOK2KIT nonché l'RGB Shield e la board Arduino Uno.
Capitolo 6: La libreria LiquidCrystal. Il progetto proposto (Termometro digitale con memorizzazione dei valori di minima e massima) utilizza i componenti del set ARDUBOOK2KIT nonché la board Arduino Uno.
Capitolo 7: La libreria SD. Il progetto proposto (Data logging ambientale) utilizza i componenti del set ARDUBOOK2KIT nonché l'RTCShieldKit, l'SDCardShield e la board Arduino Uno.
Capitolo 8: La libreria Servo. Il progetto proposto (Tombola con indicatore laser) utilizza i componenti del set ARDUBOOK2KIT nonché la board Arduino Uno.
Capitolo 9: La libreria SPI. Il progetto proposto (Morra cinese vocale) utilizza i componenti del set ARDUBOOK2KIT nonché l'Audio Shield e la board Arduino Uno.
Capitolo 10: La libreria Software Serial. Il progetto proposto (controllo accessi con RFID) utilizza i componenti del set ARDUBOOK2KIT nonché l'RFID Shield e la board Arduino Uno.
Capitolo 11: La libreria Stepper. Il progetto proposto (Generatore di disegni laser) utilizza i componenti del set ARDUBOOK2KIT nonché due schede MotorShield e la board Arduino Uno.
Capitolo 12: La libreria Wire. Il progetto proposto (Accelerometro a tre assi) utilizza i componenti del set ARDUBOOK2KIT e la board Arduino Uno.

ARDUINO UNO Rev3 con ATMEGA328



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

cod. ARDUINOUNOREV3
€ 24,50

a scuola...
strumenti didattici!

Una vasta gamma di risorse che, sfruttando la piattaforma Arduino, ti consentiranno di imparare a programmare facilmente. La nostra offerta didattica è dedicata sia a quanti partono da zero che a coloro che hanno già un'esperienza di base.
Ideale per corsi scolastici e per quanti affrontano per la prima volta la programmazione di microcontrollori.



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



cod. ARDUKITBOOK2
€ 56,90

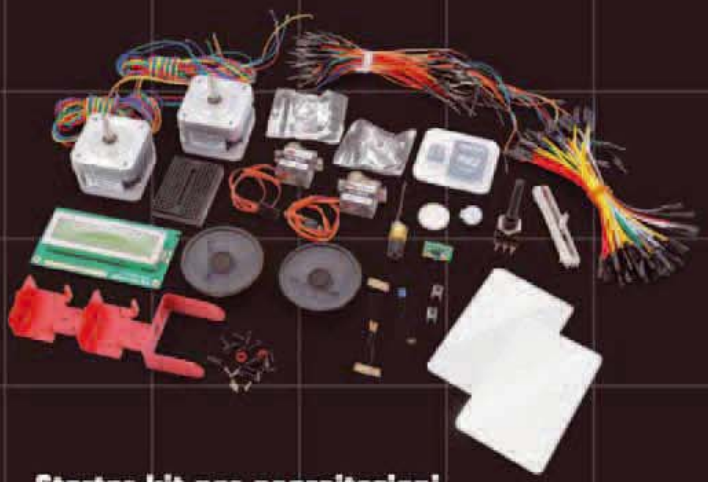
Set composto dal libro "Primi passi con Arduino" e dallo starter kit cod. ARDUKITBOOK comprendente la scheda Arduino UNO Rev3 e tutto il materiale necessario per realizzare gli esperimenti didattici descritti nel libro.

Starter kit con Arduino UNO Rev3



cod. ARDUKITBOOK
€ 52,00

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



Starter kit per esercitazioni del libro ARDUADVANCED

Contiene tutto il materiale necessario per le esercitazioni descritte nel libro "ARDUINO UNO Programmazione avanzata e Librerie di sistema"ad eccezione degli shield e della board Arduino Uno disponibili separatamente.

cod. ARDUBOOK2KIT
€ 134,00

Gli shield impiegati...

Audio shield (kit)

cod. AUDIO_SHIELD
€ 19,50



RGB shield (kit)

cod. RGB_SHIELD
€ 12,00



IN/OUT shield (kit)

cod. FT919K
€ 22,00



Motor shield (kit)

cod. MOTORSHIELD_FE
€ 18,00



Ethernet shield

cod. SHIELDETHERNET
€ 42,00



RFID shield (kit)

cod. RFID_SHIELD
€ 46,00



RTC shield (kit)

cod. RTCSHIELDKIT
€ 11,00



Micro SD Card shield (kit)

cod. SDCARDSHIELD
€ 12,00



Prezzi IVA inclusa.

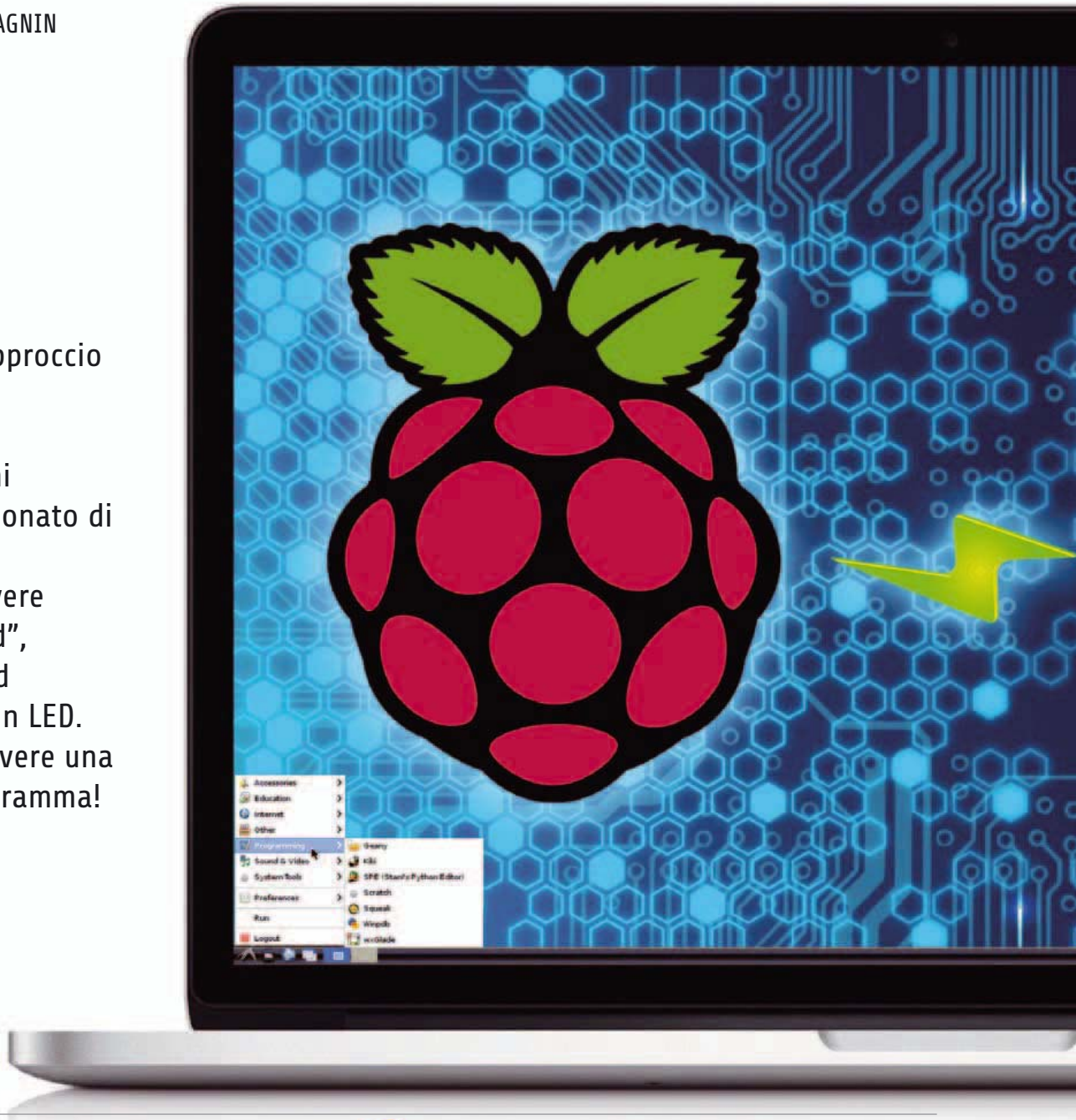
Maggiori informazioni su questi prodotti sono disponibili sul sito *www.futurashop.it* tramite il quale è anche possibile effettuare acquisti on-line.

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

FUTURA ELETTRONICA

METTIAMO IN FUNZIONE LA BOARD RASPBERRYPI

di MARCO MAGAGNIN

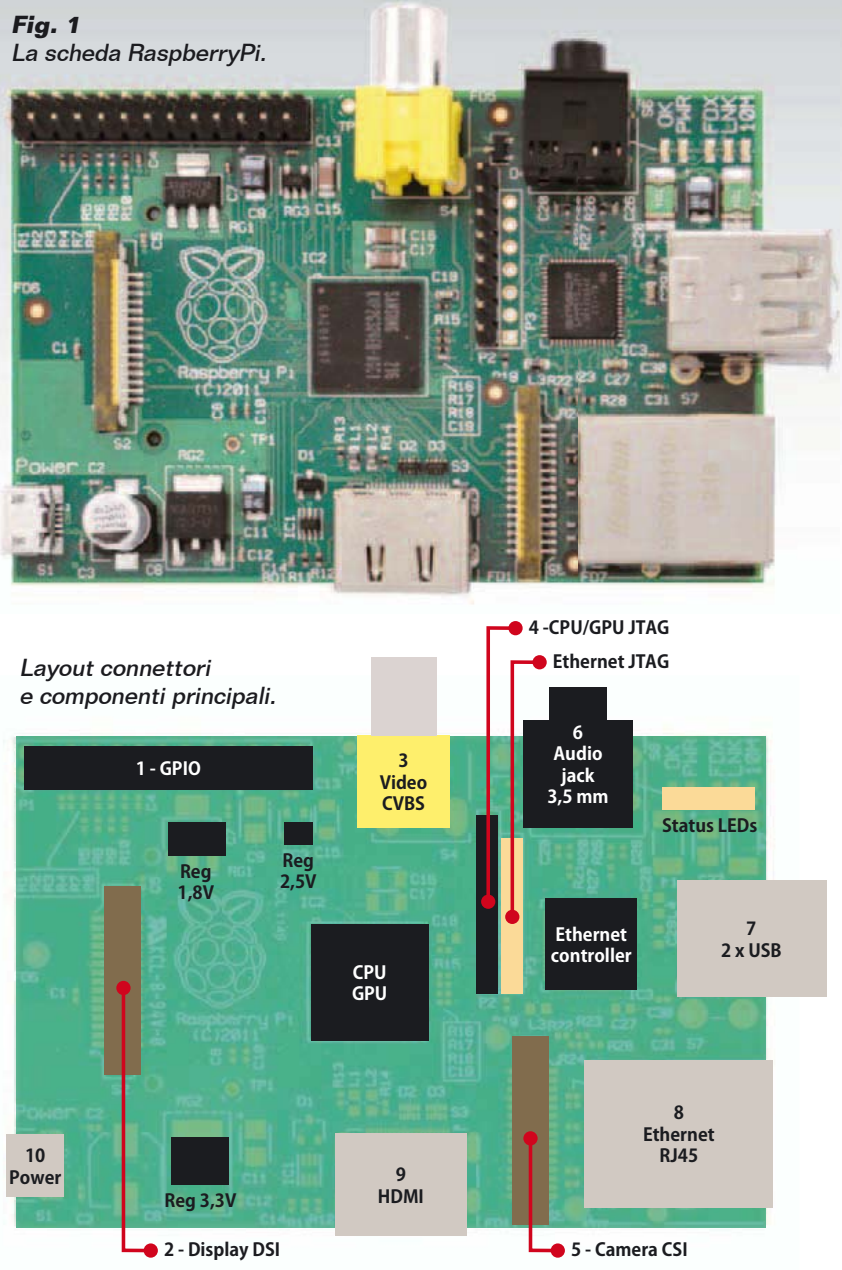


Un primo approccio alla scheda RaspberryPi con gli occhi dell'appassionato di elettronica. Oltre a scrivere "Hello World", proviamo ad accendere un LED. E senza scrivere una riga di programma!



Troppo forte è stata la tentazione di verificare il funzionamento della RasperryPi appena giunta in Redazione e, come avete visto nelle prime news pubblicate on-line, i risultati sono stati decisamente incoraggianti, tanto da indurci a dare il via ad una serie di articoli e progetti che avevamo in mente da tempo ed il cui scopo è quello di avvicinare gli appassionati di elettronica al mondo dei sistemi embedded.
Già in passato abbiamo dovuto adeguarci a nuove realtà con l'arrivo dei circuiti integrati prima e dei microcontrollori poi; con questi ultimi siamo stati costretti ad occuparci di software e programmazione anche se, con i micro, le architetture a nostra disposizione supportano un solo programma (firmware) in esecuzione permanente sull'hardware. Dall'altra parte del "mondo" i computer seguivano la stessa evoluzione costringendo gli informatici ad una analoga corsa all'aggiornamento.
Lo sviluppo dei sistemi embedded, computer dalle dimensioni ormai paragonabili a quelle dei nostri moduli a microcontrollore ma con a bordo un vero e proprio sistema operativo, ha avvicinato questi due mondi tanto che lo scoccare dell'arco ormai è inevitabile. Da una parte i nostri sistemi a microcontrollore si evolvono mettendo a disposizione strumenti che schiudono la porta sul mondo della comunicazione Internet e Web in particolare, dall'altra schede PC-embedded si dotano di pin di I/O. Questo fatto non ci ha sorpresi, se ci riferiamo a schede come la Fox G20 e la sua nuova evoluzione Aria (di cui ci occuperemo prossimamente), ma nel caso della RasperryPi ci ha incuriositi e fatto pensare. Già, perché la RaspberryPi nasce proprio con un connettore che espone una serie di pin di I/O, una interfaccia seriale, una SPI, una I2C e una alimentazione di 5V e 3,3V.
Proprio per questo motivo ci siamo ripromessi di approfondire l'argomento con una serie di articoli destinati agli elettronici puri per capire come integrare e far comunicare tra di loro e con il mondo Intenet e Web le schede

Fig. 1
La scheda RaspberryPi.



Layout connettori e componenti principali.

più eterogenee come appunto la RaspberryPi, le schede Fox G20 e Aria, le OpenPicus, Arduino, Android e così via, sfruttando le opportunità di ciascuno dei singoli sistemi e le potenzialità dei sistemi embedded che, nella maggior parte dei casi sono basati su Linux.
Già, Linux, universalmente noto come il sistema operativo dalle potenzialità pressoché infinite ma che respinge sistematicamente i tentativi di avvicinamento da parte di chi non è "nato" Linux.
In genere ci si prova, si acquista un portatile apposta per sperimentare con Linux, meglio se già installato, si prova a configurare o personalizzare qualcosa e si entra nel file system spaesati; alla fine, delusi, si riformatta tutto e si torna al più amichevole Windows.
Bene, in questa serie di appuntamenti vorremmo provare a sfatare questa fama di Linux proponendo un approccio pratico che permetta di "fare qualcosa da subito", approfondendo poi man mano i diversi argomenti ed aspetti per entrare sempre più a fondo nel mondo Linux Embedded. In più, proveremo a fare tutto questo dal punto di vista del cultore dell'elettronica; in altre parole, se l'esercizio base di tutti i tutorial destinati al mondo dell'informatica è ottenere il famoso "Hello World!", il nostro obiettivo sarà quello - tipico dei tutorial delle board hardware - di fare accendere inizialmente un LED. E, per sfatare ancora di più la nomea di Linux, aggiungiamo: senza dovere scrivere alcun programma!
Da questo punto di vista la RaspberryPi ci offre una serie di vantaggi, costa poco (e prima o poi arriverà), ospita la distribuzione di Linux Debian, universalmente ritenuta la più difficile, così non ci facciamo mancare nulla, tutto il software sta su una SD card, sistema operativo compreso, così se per caso combiniamo qualche pasticcio irreversibile, niente paura, rigeneriamo la SD e ripartiamo; infine, possiede quel curioso pettine di pin con gli I/O che ci permetteranno di utilizzarla come una vera board Linux embedded. Come ulteriore grossa opportunità, tutto ciò che apprenderemo con questa scheda (pardon, vero e proprio computer), lo potremo riutilizzare con altre schede Linux embedded, come la Fox G20, la nuova Aria e tante altre ancora come la MK802, un altro vero microcomputer delle dimensioni di una pen drive, nato per Android ma che gradisce anche Linux. E infine, ovviamente, tutto questo ci spianerà la strada verso Linux desktop, che sta prendendo sempre più piede.
In un articolo destinato ad appassionati di informatica avremmo iniziato con la storia di Linux, per capire le motivazioni alla base della sua architettura e delle

modalità di configurazione ed utilizzo.
Visto che ci dobbiamo rivolgere ad appassionati di elettronica, iniziamo invece con la descrizione della scheda RaspberryPi rimandando gli altri argomenti a box specifici.

LA RASPBERRYPI

La scheda RaspberryPi è un vero e proprio computer su scheda singola delle dimensioni di un cracker.
È stata sviluppata nel Regno Unito dalla RaspberryPi Foundation con l'intento di realizzare un computer a basso costo per stimolare l'insegnamento dell'informatica nelle scuole, in modo particolare nei paesi in via di sviluppo.
Il cuore del computer è il SoC (System On a Chip) Broadcom BCM2835 che include un processore ARM1176JZF-S da 700 MHz, un processore grafico (GPU) VideoCore IV, un processore di segnali digitali (DSP) e 256 Mb di RAM.
Non sono necessari dischi fissi o allo stato solido in quanto il sistema operativo e la memoria di massa sono allocati su SD Card che funge anche da unico device di boot.
Nello schema in **Fig. 1** sono riportati i connettori ed i componenti principali presenti sulla scheda RaspberryPi:

1. Connettore GPIO
2. Connettore per un Monitor LCD/OLED con interfaccia DSI (Display Serial Interface) del tipo montato sui telefoni cellulari.Verificheremo una lista di modelli compatibili;
3. Connettore per un monitor o televisore con uscita video composito;
4. Connettore JTAG;
5. Connettore CSI (Camera Serial Interface). Tutte le videocamere utilizzate negli Smartphone con sensore CMOS dovrebbero essere compatibili con la RaspberryPi come, ad esempio, la economica OV5650 con interfaccia CSI. Anche videocamere con interfaccia I2C come la TCM8240MD dovrebbero poter essere utilizzate con una interfaccia CSI-2 Camera Bridge IC;
6. Jack audio in uscita (in ingresso bisogna utilizzare un microfono USB);
7. Due connettori USB. Per espandere le porte USB si può usare un HUB esterno, però con alimentazione autonoma; a questo connettore si collegano la tastiera ed il mouse;
8. Connettore Ethernet RJ45;
9. Connettore HDMI per collegarsi ad un monitor video ad alta qualità;
10. Connettore di alimentazione, in formato micro USB, quello dei caricabatteria dei telefonini, per alimentare la RaspberryPi è opportuno utilizzarne uno da almeno un Ampere.

IL CONNETTORE GPIO

Vediamo subito come è configurato il connettore GPIO che dovremo usare per accendere il nostro LED e per interfacciare la board con hardware esterno.
Da un punto di vista fisico si presenta come un connettore a passo 2,54 mm su due file di 13 pin ciascuna, per un totale di 26 pin (**Fig. 2**).
Per l'orientamento bisogna tenere conto che il pin 2 è quello in alto a sinistra tenendo la scheda con il connettore della SD Card posto sulla sinistra.
Su questo connettore sono presenti l'alimentazione a 3,3V sul pin P1-01, 5V sul pin P1-02, la massa sul pin P1-06 e poi altri 17 pin di input/output, che possono essere riconfigurati per ulteriori funzioni come I2C, SPI, UART, Seriale e altre. Ricordiamo da subito, prima di iniziare con esperimenti e applicazioni, che i pin di I/O funzionano con livelli di tensione 0 e 3,3V e NON tollerano il livello a 5V; inoltre non è prevista una protezione dalle sovratensioni.
In sostanza, i pin GPIO sono connessi direttamente ai pin del processore.Tutto ciò costringe gli utilizzatori a progettare schede di interfaccia con buffer, conversione di livelli, I/O analogico e

Tabella 1 *- Specifiche tecniche della scheda RaspberryPi.*

Caratteristiche tecniche	Descrizione
SoC:	Broadcom BCM2835 (CPU, GPU, DSP, SDRAM)
CPU:	processore ARM1176JZF-S a 700 MHz (famiglia ARM11)
GPU:	Broadcom VideoCore IV,[55] OpenGL ES 2.0, 1080p30 h.264/MPEG-4 AVC high-profile decoder
Memoria (SDRAM):	256 MB (in condivisione con la GPU)
Porte USB 2.0:	2 (USB hub integrato)
Uscite Video:	Video composito RCA (PAL & NTSC), HDMI (rev 1.3 & 1.4), Pannelli LCD via DSI, 14 risoluzioni HDMI da 640×350 a 1920×1200 (standard PAL e NTSC)
Uscita Audio:	jack da 3.5 mm, HDMI
Memoria di massa:	slot per scheda SD / MMC / SDIO
Network:	10/100 Ethernet (RJ45)
Low-level peripherals:	8 × GPIO, UART, bus I²C, bus SPI con due chip selects, +3.3 V, +5 V, ground
Alimentazione:	700 mA (3.5 W)
Ingressi tensione:	5 volt via MicroUSB o connettori GPIO
Dimensioni:	85.60 × 53.98 mm
Peso:	45 g
Sistemi operativi previsti:	Debian GNU/Linux, Fedora, Arch Linux ARM, RISC OS

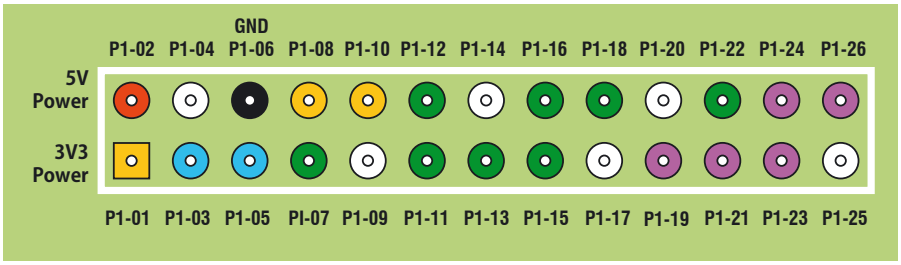


Fig. 2 *- La pinout del connettore GPIO.*

dispositivi di protezione, in modo da evitare il rischio di danneggiare la RaspberryPi.
Vedremo in articolo come accedere ai pin di I/O, come configurarli e cambiarne il livello da alto a basso e viceversa; per il momento teniamo presente che la configurazione predefinita è in direzione OUT.
Nella **Tabella 2** sono descritti i nomi relativi ai vari pin da utilizzare nei programmi e nei comandi. Ricordiamo che ciascun pin è contraddistinto da un numero che ne identifica la posizione sul connettore (da P1-01 a P1-26) e da uno o più nomi che ne specificano la o le funzioni. Ad esempio, il pin P1-07, viene identificato col nome GPIO4. Tuttavia, come abbiamo già accennato, i pin del connettore, possono essere riconfigurati per poter accedere a molti dei segnali del processore. Per fare questo bisogna utilizzare i nomi alternativi di ciascun pin il cui elenco completo è disponibile al link *http://elinux.org/RPi_BCM2835_GPIOs*. In questa pagina riportiamo solamente alcune delle funzioni più utilizzate che sono elencate nella terza colonna della **Tabella 2** chiamata ALTØ. Ad esempio, il pin P1-05 nel caso normale è denominato GPIO1, con funzione di porta di ingresso e uscita. Lo stesso pin è anche assegnabile al clock del bus I2C; in questo caso, in un ipotetico programma, dovrà essere identificato col nome GPIO1_ALTØ.
Alcune note aggiuntive sui pin presenti sul connettore:

- I pin 3 e 5, che possono essere utilizzati per il bus I2C sono dotati di resistenze di pull up da 1,8 kohm;
- Il pin 12 supporta il PWM;
- La massima corrente disponibile sul pin 3,3 volt è di 50 mA;
- La massima corrente disponibile sul pin 5 volt è di 300 mA.

Complessivamente i pin di ingresso e uscita presenti sul connettore GPIO sono 17.
Infine un'ultima nota sul significato dei cinque LED di stato presenti sulla scheda:

Tabella 2 *- Funzioni assegnate ai pin del connettore GPIO.*

Numero pin	Nome pin	Funzione ALTØ	Numero pin	Nome pin	Funzione ALTØ
P1-02	5V0		P1-01	3V3	
P1-04	DNC		P1-03	GPIO0	I2C0_SDA
P1-06	GND		P1-05	GPIO1	I2C0_SCL
P1-08	GPIO14	UART0_TXD	P1-07	GPIO4	
P1-10	GPIO15	UART0_RXD	P1-09	DNC	
P1-12	GPIO18		P1-11	GPIO17	
P1-14	DNC		P1-13	GPIO21	PCM_DIN
P1-16	GPIO23		P1-15	GPIO22	
P1-18	GPIO24		P1-17	DNC	
P1-20	DNC		P1-19	GPIO10	SPI0_MOSI
P1-22	GPIO25		P1-21	GPIO9	SPI0_MISO
P1-24	GPIO08	SPI0_CE0_N	P1-23	GPIO11	SPI0_SCLK
P1-26	GPIO07	SPI0_CE1_N	P1-25	DNC	

- OK - Significa che è presente una scheda SDCard valida;
- PWR - Si accende in presenza di alimentazione;
- FDX - Indica una connessione LAN Full Duplex;
- LNK - Indica attività sulla rete LAN;
- 10M - Velocità della connessione LAN; quando acceso la connessione è a 100Mb.

DIAMO VITA ALLA RASPBERRYPI

Per prima cosa colleghiamo alla RaspberryPi le periferiche necessarie al suo funzionamento e cioè:

- Tastiera, una qualsiasi tastiera USB, non c'è preferenza di porta, il sistema operativo riconosce automaticamente le periferiche;
- Mouse, anche qui un qualsiasi mouse USB nella porta rimasta libera;
- Una connessione di rete Ethernet, via cavo. Nel caso di rete cablata non c'è problema, utilizzate una delle terminazioni disponibili; in casa collegatevi ad una porta del router ADSL;
- Un monitor od un televisore, che può accettare in ingresso il segnale Video Composito o HDMI; consigliamo vivamente di optare per una soluzione HDMI, avrete una risoluzione decisamente migliore; in caso contrario non preoccupatevi, soffrirete un poco di più all'inizio, ma abbiamo già sperimentato il monitor in remoto con VNC che vi permetterà di vedere la vostra RaspberryPi sul monitor del vostro PC con la risoluzione che preferite; in seguito vi spiegheremo passo passo come installarlo e utilizzarlo;
- Un alimentatore a 5V (per ora non collegatelo!) con l'uscita micro USB, per intenderci quelli per ricaricare i telefo-

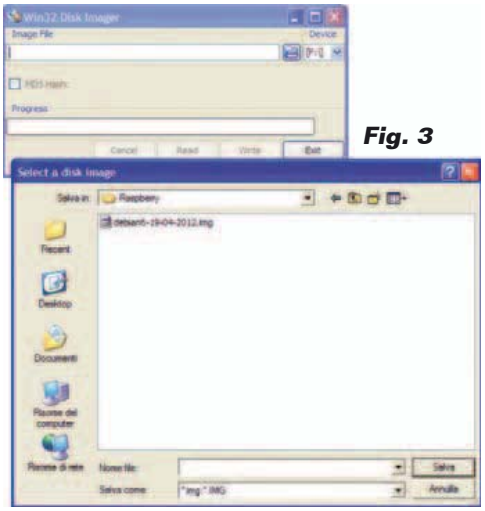


Fig. 3

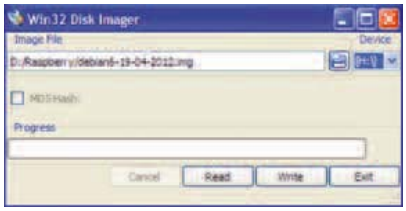


Fig. 4



Fig. 5

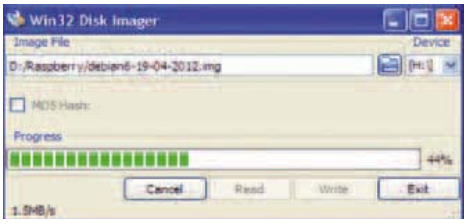


Fig. 6

nini; anche qui in alternativa possiamo collegare un alimentatore stabilizzato a 5V direttamente al positivo e negativo del connettore GPIO, il positivo al piedino 2, il negativo al 6; fate attenzione a non scambiare la polarità.

Passiamo ora al sistema operativo, che è il motore della nostra RaspberryPi. Per sua caratteristica progettuale la scheda accetta il boot, cioè il trasferimento in memoria dei primi programmi necessari al successivo caricamento del sistema operativo, unicamente da SDCard, che rappresenta l'equivalente del disco fisso principale del nostro PC.
In seguito sperimenteremo come velocizzare le operazioni di trasferimento dei file impiegando altre memorie di archiviazione di massa come chiavette USB o dischi fissi, magari allo stato solido.
Come curiosità il termine boot, che è l'abbreviazione di bootstrap, è il laccetto posto dietro a scarponi e stivali che serve per aiutare a calzarli, tirandolo da dietro.
A questo punto, dunque, dobbiamo preparare una SD Card con il software necessario all'avvio della scheda ed al caricamento del sistema operativo Linux.
Nel nostro caso abbiamo utilizzato una SDCard da 8 Gb, ma è possibile impiegare card da 2 a 32 Gb a seconda delle proprie esigenze (la SD verrà utilizzata anche come memoria di massa, ovvero come una sorta di hard-disk). Raccomandiamo di sceglierne una ad accesso veloce in lettura e scrittura (Class 4) perchè da questa caratteristica dipendono i tempi di caricamento dei programmi e dei dati.
A questo scopo utilizziamo il nostro PC connesso a Internet, che deve disporre di uno slot o un adattatore per SDCard.
Descriviamo la procedura riferendoci ad un PC con sistema operativo Windows in quanto riteniamo che quanti utilizzano un PC con Linux non abbiano bisogno di queste note.
La procedura consiste nello scaricare dalla sezione Downloads del sito ufficiale di RasperryPi (*http://www.raspberrypi.org/downloads*) il file "debian6-19-04-2012.zip" o una versione più recente identificata da una data successiva. Il file scaricato contiene una copia in formato binario (ovvero adatto ad una SDCard) del sistema operativo Linux che vogliamo utilizzare e che comprende anche il boot.
Nel nostro caso abbiamo scaricato la *distribuzione* di Linux denominata Debian; nel mondo Linux con la parola "distribuzione" si intende un insieme di componenti di sistema operativo di base con l'aggiunta di applicazioni utente destinate a soddisfare una specifica categoria di utilizzatori, ad esempio grafici, webmaster, sviluppatori di applicazioni e così via. La distribuzione Debian è la più generalista possibile ed anche la preferita dagli hacker.
Scaricato il file in una qualsiasi cartella del PC (nel nostro caso

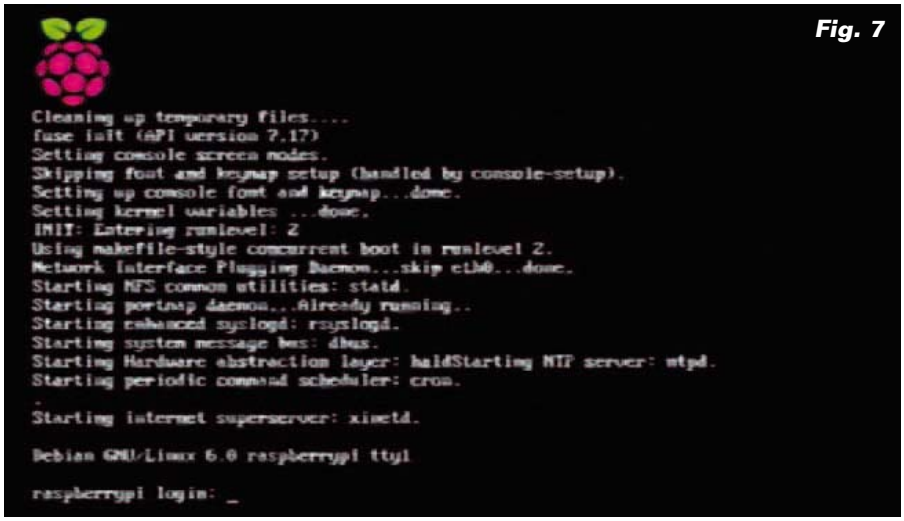
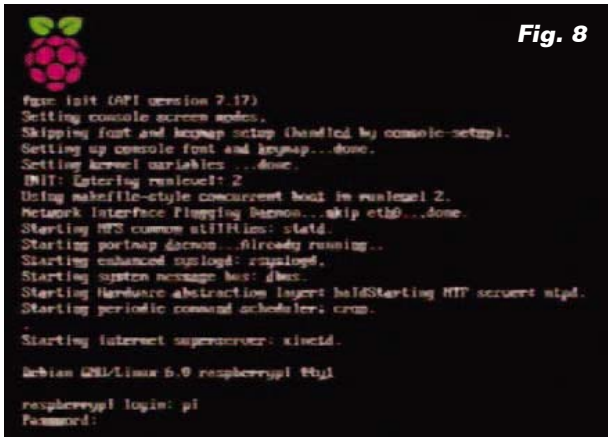


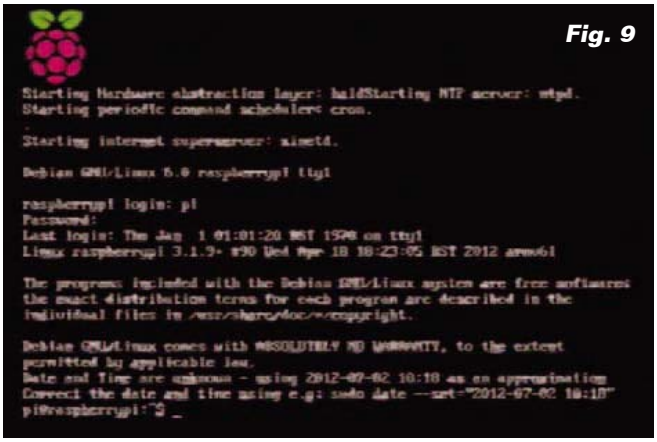
Fig. 7

La storia di **LINUX**

La storia di Linux parte da lontano, un po' come quella di Internet e del suo protocollo di comunicazione TCP/IP.
Agli inizi dell'era informatica i giganteschi e costosissimi elaboratori disponibili allora si comportavano più o meno come i nostri microcontrollori. Accettavano un unico programma alla volta e di conseguenza un solo utente. Di fatto erano privi di un sistema operativo, ovvero di un "programma" centralizzato sempre in memoria che potesse coordinare il funzionamento di diversi programmi, ripartendo risorse e potenza di calcolo e di conseguenza accettare diversi utenti in contemporanea.
I primi sistemi operativi furono sviluppati negli anni '50. Tra gli esempi si può citare il GMOS, il sistema operativo della General Motors sviluppato per l'IBM 701, oppure il FORTRAN Monitor System (FMS) sviluppato dall'aviazione americana per l'IBM 709.
Ciascun produttore investiva in quegli anni per sviluppare sistemi operativi adatti alle architetture delle proprie macchine; tra gli altri, negli anni '60, l'MIT, i laboratori della AT&T Bell e la General Electric iniziarono a sviluppare in consorzio il sistema operativo sperimentale Multics per il calcolatore GE-645 (**Fig. A**), ai tempi molto innovativo per quanto riguardava la flessibilità di gestione delle risorse e la scalabilità della configurazione. Purtroppo non ebbe molto successo per via dei numerosi problemi incontrati nella progettazione e realizzazione. Ma l'idea era buona ed i laboratori AT&T Bell, lentamente si allontanarono dal consorzio e iniziarono una riprogettazione del sistema su una scala più ridotta che denominarono Unics. I ricercatori Ken Thompson, Dennis Ritchie, M. D. McIlroy, e J. F. Ossanna legarono il loro nome a questo progetto, sviluppando, insieme ad altri, su un PDP-7 usato, il primo file system organizzato gerarchicamente, il concetto di processo, i file di mappatura dei dispositivi, oggi noti come driver, e il primo interprete a linea di comando.
Il passo successivo fu lo sviluppo e l'integrazione del linguaggio C in Unics che nel frattempo cambiava nome in UNIX. Space Travel è il video game che segna il raggiungimento degli obiettivi di UNIX come sistema operativo multiutente, multitasking, in grado di gestire la memoria virtuale, portabile, aperto e basato sul linguaggio C. Apertura e portabilità sono le caratteristiche che ne consentiranno il successo; apertura soprattutto verso la comunicazione con gli altri computer in quanto implementava nativamente il TCP/IP, il protocollo di Internet; portabilità perché la sua architettura "a cipolla" prevedeva un kernel centrale che si occupava di tutta la gestione dell'hardware che lo ospitava; per portare UNIX su un altro tipo di hardware ed adattarlo alla nuova architettura bastava

abbiamo chiamato la cartella Raspberry), dobbiamo trasferirlo su SDCard. Per fare ciò dobbiamo utilizzare il programma Win32DiskImager che può essere scaricato gratuitamente dal link *https://launchpad.net/win32-image-writer/+download* e che va installato sul PC.
Scompattiamo nella cartella in cui lo avevamo salvato, lo zip contenente Debian ed otteniamo il file "debian6-19-04-2012.img".
Inseriamo la SDCard nell'apposito slot del PC, se disponibile, altrimenti utilizziamo un Multi Media Card Reader.
Apriamo Win32DiskImager e carichiamo il file di Debian che avevamo scompattato in precedenza (**Fig. 3**) e scegliamo come destinazione l'unità che identifica la SDCard (**Fig. 4**), premiamo "Write" (**Fig. 5**) e poi diamo l'OK.
In f**igura 6** si vede l'avanzamento del trasferimento su SDCard.
Come notate esiste anche un pulsante "Read" che permette di fare un backup della SDCard ogni volta che vogliamo, ad esempio per creare una SDCard per una seconda scheda con le stesse configurazioni.
Lo spazio occupato dalle informazioni trasferite occupa circa 2Gb ma l'eventuale spazio eccedente non potrà essere al momento utilizzato; vedremo in seguito come fare in modo da rendere disponibile al sistema tutta la capacità della SDCard.
A questo punto dimentichiamoci del nostro PC, preleviamo la scheda SDCard e inseriamola nello slot della RaspberryPi alla quale, come abbiamo detto in precedenza, avremo già collegato video, tastiera, eccetera.
La scheda non dispone di interruttore di accensione, per accendere si dà alimentazione, per





riscrivere (portare) il solo kernel (o parte di esso) che allora era di circa 10.000 righe di codice in linguaggio C.
Vent'anni dopo, negli anni '90, sfruttando proprio le caratteristiche di portabilità, Andrew Tanenbaum scrive una versione di UNIX®, chiamata MINIX (MInimal uNIX) basata su un microkernel, che aveva la capacità di girare sui PC. Ed è proprio a questo progetto che, nei primi anni 90, si ispirò Linus Torvalds, il creatore di Linux.
Linux si è rapidamente diffuso da progetto di una singola persona a progetto di sviluppo mondiale grazie al coinvolgimento di una comunità di migliaia di programmatori e dalla adozione della licenza GPL (GNU General Public License). Sotto la GPL, il kernel Linux è stato protetto da sfruttamento commerciale ed ha anche tratto benefici dai programmi disponibili in user-space che facevano parte del progetto GNU (di Richard Stallman, che ha scritto tanto di quel codice che in confronto il sorgente del kernel Linux appare minuscolo).
Questo ha permesso una crescita esplosiva, la diramazione in diverse "distribuzioni" con l'inglobamento di altri progetti, package e applicativi, shell a linea di comando e ambienti grafici note come Debian, Fedora, Ubuntu, Knoppix e infinite altre. Alcune in grado di partire direttamente da CD, da chiavetta USB o da "SD Card".


Fig. A

Alcune distribuzioni sono destinate a specifici campi di utilizzo, come la sicurezza informatica, firewall, gateway, grossi provider di servizi o anche gestione di CNC o la gestione video e grafica. Tutto questo ha portato allo sviluppo di nuove professionalità, molto ricercate nel mondo del lavoro come specialisti di networking, di service providing e di sicurezza (questi ultimi spesso provenienti dalle file degli hacker).
Nel frattempo il "piccolo" kernel ha raggiunto la dimensione di circa 6 milioni di righe di codice, sempre portabili però, come dimostra la nostra RaspberryPi.

spegnere la si toglie. Alcuni LED si accenderanno e dopo qualche attimo vedremo l'immagine di un piccolo lampone sullo schermo, seguito da una lista di comandi che scorreranno velocementre (**Fig. 7**); aspettate alcune decine di secondi fino al termine ed alla comparsa di una riga contenente:

raspberrypi login:

Poche righe prima, probabilmente, troverete l'indirizzo IP della rete interna assegnato al vostro dispositivo dal router DHCP, che sta a significare una corretta connessione alla rete.
Soffermiamoci ora su un concetto importante: Linux permette l'accesso e l'esecuzione di comandi solo a utenti riconosciuti e debitamente autorizzati (userid e password). L'amministratore che ha tutti i diritti di accesso e intervento è l'utente SuperUser al quale competono anche i diritti di "root". Root è la radice dell'albero delle cartelle e dei file che costituiscono la struttura del sistema operativo. Vedremo in seguito come attivare l'utente SuperUser per poter eseguire quei comandi che richiedono questo livello di autorizzazione.
Ogni distribuzione di Linux ha all'interno un utente generico già attivato, che permette di poter iniziare a lavorare da subito. Nel nostro caso l'utente già attivato è "pi" e la password "raspberry" (**Fig. 8** e **9**). Quindi rispondiamo:

raspberrypi login: pi
password: raspberry

Bene, siamo operativi, e andiamo a risolvere subito il primo problemino di configurazione ovvero la tastiera che, di default, è quella inglese e quindi i carat-

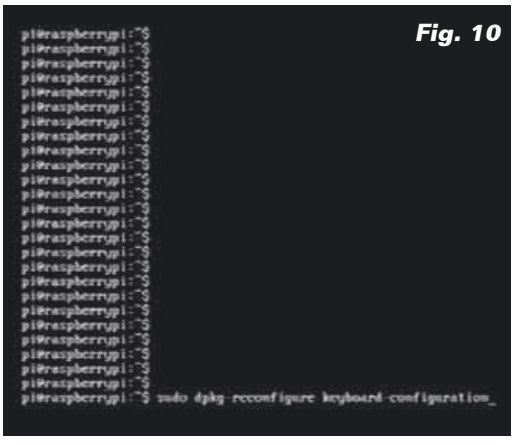


Fig. 10

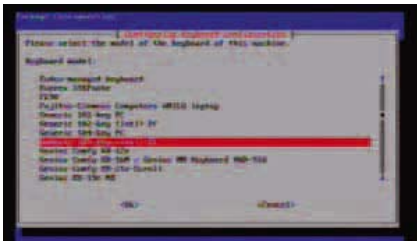

Fig. 11


Fig. 12

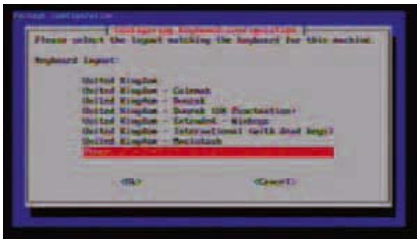
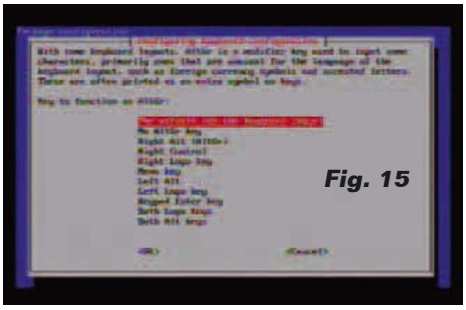

Fig. 13


Fig. 14


Fig. 15

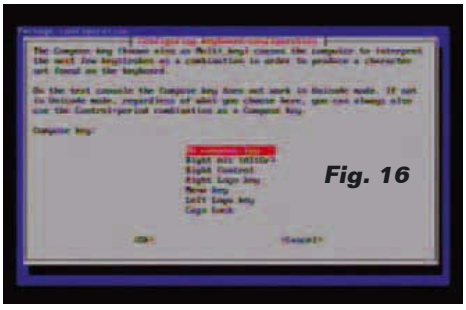
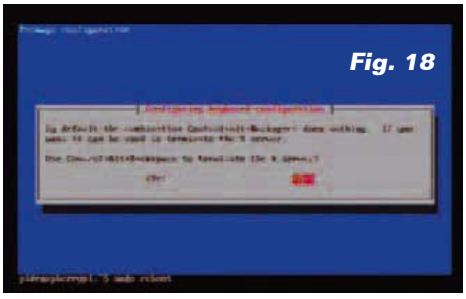

Fig. 16


Fig. 17


Fig. 18

teri speciali non corrispondono.
Come prima operazione configuriamo le impostazioni di tastiera, per adottare quella italiana, con il comando:

sudo dpkg-reconfigure keyboard-configuration

Per ottenere, in questo momento, il carattere - (meno), è necessario digitare sulla tastiera (italiana...) il carattere ? (punto di domanda).
Il comando *dpkg-reconfigure* fa parte di quelli che richiedono i privilegi di SuperUser, il comando *sudo* (superuser do) assegna temporaneamente i privilegi di SuperUser all'utente corrente (**Fig. 10**), in un modo simile a qundo si seleziona in Windows l'opzione "Esegui come Amministratore".
Eseguito il comando, appariranno le finestre di configurazione della tastiera da **Fig. 11**. a **Fig. 18**, assolutamente intuitive.
Per rendere effettive le configurazioni eseguiamo il reboot (riavvio) della scheda con il comando:

sudo reboot

Dopo aver effettuato nuovamente il login, attiviamo ora l'utente SuperUser che ha sempre come nome "root" ed al quale va definita la password per poterlo attivare. Digitiamo il comando (**Fig. 19**):

sudo passwd

ed in seguito inseriamo una password, per due volte, e poi annotiamola accuratamente (**Fig. 20** e **21**).
Attenzione, se digitate unicamente

passwd

cambierete la password all'utente corrente, e non attiverete il SuperUser.
Fate attenzione con le password perché in Linux è difficile recuperare le password e pressoché impossibile recuperare quella del SuperUser.
Al momento non abbiate timore: fortunatamente se qualcosa va storto ricreate una SD con il file d'immagine originale presente sul vostro PC e ricominciate da capo.
Ora concentriamoci sull'accensione del nostro LED; per l'accesso ai pin del GPIO è necessario utilizzare privilegi di SuperUser, e quindi nel nostro caso ci autenticheremo come utente "root" con il comando:

su

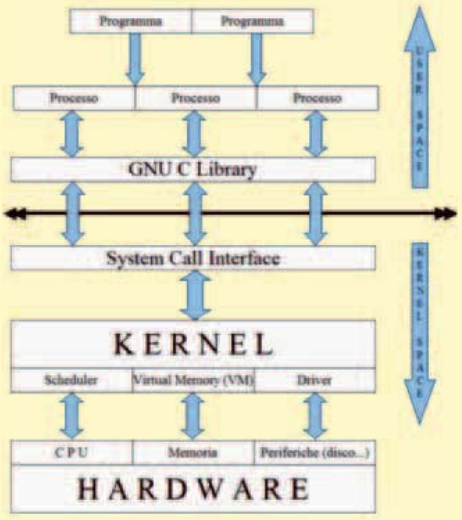
e con la password relativa.
Ora colleghiamo un LED rosso con in serie una resistenza da 120 Ohm tra il pin 7 (GPIO4) del

Linux è un sistema operativo multitasking e multiutente, ossia può gestire più processi e più utenti contemporaneamente. Un processo è una unità minima di elaborazione, se vediamo la funzione di multitasking di un sistema operativo come un tavolo di lavoro con sopra molte scatole, ciascuna delle quali contiene un compito da eseguire, tipo montare giocattoli di legno, e chi si occupa dei montaggi passa da una scatola all'altra per esempio incollando due parti e poi lasciarle asciugare mentre nel frattempo incolla altre parti in una scatola differente, questo è precisamente il comportamento del sistema, dove in ciascuna scatola sono contenuti uno o un insieme omogeneo di quelli che noi chiamiamo programmi.
Il cuore del sistema operativo Linux è il kernel, che in questo sistema è rigidamente l'unico ad avere accesso a basso livello alle risorse di sistema e a presentarle ai livelli superiori in modo standardizzato e indipendente dal processore hardware. I servizi e le applicazioni a livello superiore non devono curarsi su quale hardware verranno eseguite e quindi non devono essere modificate per adattarsi ad hardware diversi. Questa caratteristica di Linux ha contribuito alla sua fortuna, in quanto lo ha reso estremamente portabile, per "portare" Linux su un nuovo processore è sufficiente adattare il kernel alla nuova architettura, tutto il resto rimane invariato.
Anche le interfacce grafiche desktop come Gnome o KDE, non sono funzionalità offerte dal kernel, bensì normalissimi programmi

esterni, al pari di un OpenOffice.org o Firefox. Ecco perché in Linux potete scegliere più interfacce grafiche, è semplicemente un programma come tutti gli altri. In Windows invece l'interfaccia grafica è integrata nel sistema operativo, ed in alcune versioni, anche il browser per la navigazione Internet.
Quella che comunemente viene chiamata distribuzione, Debian, Ubuntu, openSUSE, RedHat solo per fare qualche nome fra le più conosciute, è l'insieme del sistema operativo Linux più una raccolta coerente di programmi usati dagli utenti per le loro normali attività: programmi di grafica, produttività personale, navigazione Internet, player audio/video e così via.
Vediamo ora quali sono le funzioni svolte dal kernel per eseguire i processi, ovvero i contenitori delle applicazioni dei nostri programmi. In figura sono rappresentati schematicamente i tre sottosistemi principali:

- Gestione dei processi tramite lo Scheduler. Questa parte del kernel si occupa di tutto ciò che riguarda la gestione di un processo, ad esempio di stabilire l'ordine di esecuzione dei processi, il tempo da dedicare al processo, la sospensione dell'elaborazione, la ripresa, lo stop e così via;
- Gestione della memoria tramite la memoria virtuale (VM).La memoria fisica non è liberamente accessibile ai processi. E' il VM che, anche con l'aiuto del Memory Management Unit (MMU), apposita unità presente direttamente nell'hardware dei microprocessori, rimappa opportunamente quella fisica in uno spazio virtuale che rende disponibile al processo, la scatola dell'esempio iniziale. In questo modo, ogni processo non accede mai alla memoria reale del sistema, ma solo a quella resagli disponibile dal VM. Rinchiuso nel suo spazio di memoria dedicato, il processo non può creare problemi accedendo per sbaglio allo spazio di un altro processo (chi ha usato le prime versioni di Windows, sa quanto fossero frequenti i blocchi dell'intero sistema causati proprio da problemi di questo tipo). Quando la memoria RAM poi per qualche motivo diventa insufficiente, è sempre il VM a gestire lo swap, lo spostamento cioè di parti della memoria sul disco fisso per liberare spazio, nel caso della RaspberryPi nella SD card.
- Gestione delle periferiche tramite i Driver. Questo sottosistema del kernel si occupa di gestire gli accessi alle periferiche rendendole disponibili ai programmi. La particolarità assolutamente vincente di questo sottosistema è che fornisce una modalità comune (un'interfaccia) di accesso a periferiche anche molto diverse fra loro, porte seriali, dischi rigidi, scheda audio tanto per fare degli esempi, trattandole tutte come se fossero dei file; da qui uno dei motti di Linux: tutto è un file. Ricordiamocene sempre quando cercheremo di installare e utilizzare qualche nuova periferica recalcitrante sul nostro Linux. Ce ne ricorderemo anche quando dovremo accendere il nostro LED.

Ecco, questa è in estrema sintesi l'architettura di un sistema Linux, giusto per iniziare a capire; in realtà le cose sono ben più complesse, come potreste vedere dalle dimensioni di qualunque libro e manuale dedicato al kernel di Linux.



Schema della architettura di Linux.

connettore della Raspberry, e la massa (GND, pin6).
Digitiamo (**Fig. 22**):

echo "4" > /sys/class/gpio/export

Questo comando assomiglia molto ai comandi DOS di Microsoft e di fatto significa: esegui il programma export che sta nella cartella *sys/class/gpio* passandogli il parametro "4", che è il pin che vogliamo usare.
Questo comando crea una struttura di file nella cartella /sys/class/gpio/gpiochip0/subsystem mettendo il device GPIO4 in condizione di essere gestito con i comandi successivi.
Prossimamente spiegheremo i metodi per scorrere le cartelle ed i file, sia da linea di comando che utilizzando l'interfaccia grafica. Per ora ci interessa sapere che in questa cartella sono presenti, tra gli altri, due file: direction e value, entrambi normali file di testo. In direction è contenuta la direzione del pin in o out, la direzione predefinita assegnata quando si esporta un device di tipo pin è out.
Nel file value vi è il livello del pin: "1" per livello alto "0" per livello basso.
A conferma che in LINUX (quasi) tutto è un file, potete verificare il contenuto di questi file con:

cat /sys/class/gpio/gpio4/direction

otterremo "out" e

cat /sys/class/gpio/gpio4/value

otteremo "0".
Per inciso, *cat* è il comando che stampa a console il contenuto dell'argomento del comando, nel nostro caso i file di gestione del pin.
Per assegnare una direzione al pin si usa il comando:

echo "out" > /sys/class/gpio/gpio4/direction

che scrive "out" nel file direction
Per trasformarlo in imput, nel caso il pin sia usato come ingresso, si usa il comando:

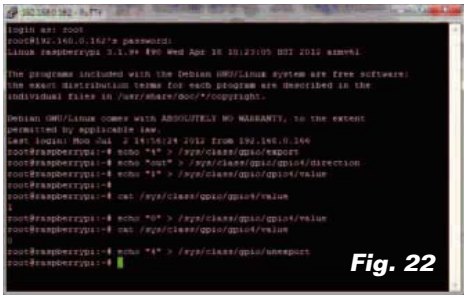
echo "in" > /sys/class/gpio/gpio4/direction

Ricordiamo di fare attenzione ai livelli, i pin GPIO NON tollerano tensioni superiori ai 3,3 volt.
Ora diamo un'occhiata al nostro


Fig. 19


Fig. 20


Fig. 21


Fig. 22

LED e verifichiamo che sia spento, in quanto il valore del pin è "0" (**Fig. 23**)
Per accenderlo basta portare a "1" il livello del pin, con il comando:

echo "1" > /sys/class/gpio/gpio4/value

Verifichiamo ora che il LED sia acceso. Ce l'abbiamo fatta! (**Fig. 24**). Ora lo spegniamo col comando:

echo "0" > /sys/class/gpio/gpio4/value

Per verificare il valore corrente usiamo sempre il comando:

cat /sys/class/gpio/gpio4/value

Bene, alla fine ora "smontiamo" il device GPIO4 con il comando:

echo "4" > /sys/class/gpio/unexport

Questo comando corrisponde, in Windows, a "rimozione sicura dell'hardware". Partendo da questo punto si possono scrivere applicazioni in svariati linguaggi con funzionalità locali e ad accesso remoto, come vedremo nei prossimi appuntamenti. Per spegnere la RaspberryPi date il comando:

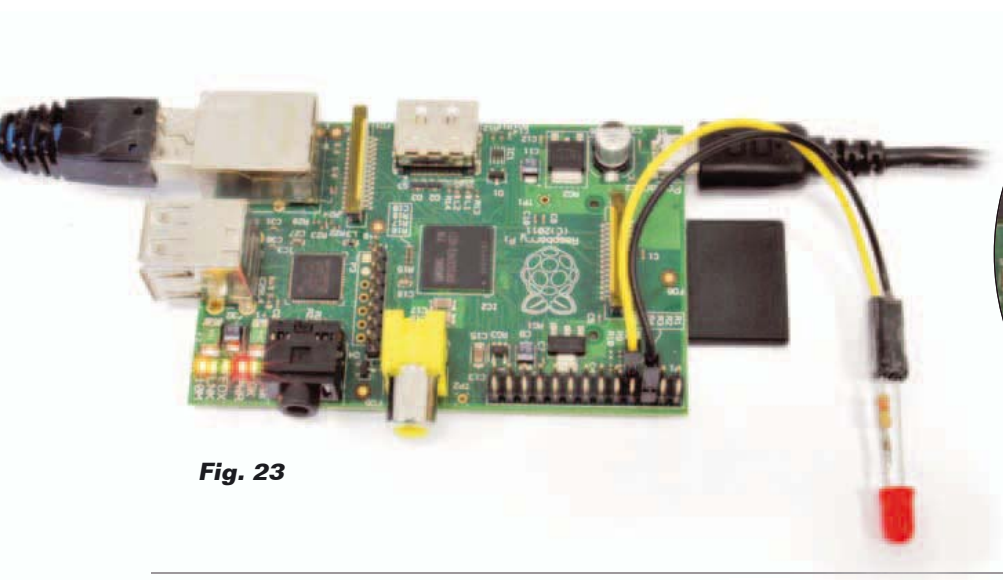
halt

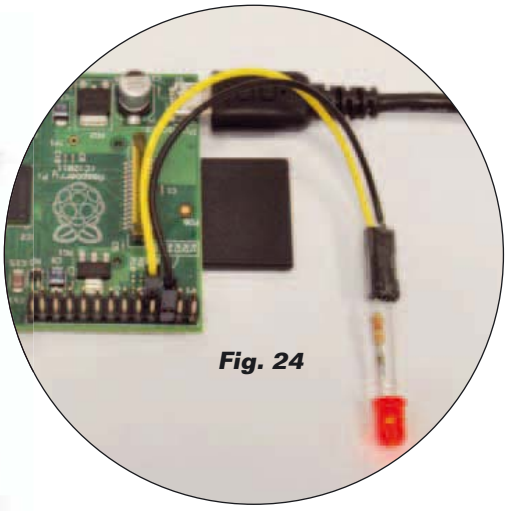
Se invece volete divertirvi con l'interfaccia grafica date il comando:

startx

Giocate fin che volete, per uscire, come in Windows scegliete Logout dal menù sulla barra e poi Shutdown.
Nei prossimi mesi approfondiremo una serie di argomenti legati al mondo Linux embedded, per esempio la possibilità di gestire i nostri pin GPIO dall'esterno, via web, memorizzandone gli stati su database, oppure di gestire applicazioni che girano su altri microcontrollori per mezzo di chiamate web service (API), in modo da realizzare un network di controllori eterogenei gestiti centralmente dal sistema Linux embedded, collegare telecamere al web, integrare il text to speech e altro ancora.
Difficile? Probabilmente meno di quanto pensiate.

La scheda RaspberryPi viene distribuita in esclusiva mondiale da Rs Components e da Farnell.


Fig. 23


Fig. 24

PROMOSSO DA:

assodel

Associazione Nazionale Fornitori Elettronica

POWER FORTRONIC

L'appuntamento italiano dell'elettronica di potenza

L'elettronica di potenza è soggetta a una innovazione continua, che richiede un costante aggiornamento da parte dei progettisti e degli operatori. Per fare il punto della situazione, a un anno dall'ultima edizione, l'**Electronic Community** si incontra di nuovo a Bologna, all'evento riconosciuto come il riferimento italiano per l'elettronica di potenza

Focus su:

- Very High Power (Power Mosfet, IGBT, IGCT, IPM ecc);
- Nuove tecnologie (SiC, GaN ecc);
- Power Management;
- Controllo motori;
- Strumentazione per applicazioni power;
- Normative e standard negli alimentatori;
- Passivi (Condensatori; Filtri; Magnetici; Thermal management ecc).

IL POWER FORTRONIC IN NUMERI

- 8 edizioni realizzate con successo;
- Più di 60 aziende espositrici nella scorsa edizione;
- Più di 400 visitatori intervenuti durante l'edizione 2011;
- Più di 6.000 nominativi professionali in mailing;
- Una audience qualificata: il 72% dei partecipanti sono progettisti e consulenti alla ricerca di un aggiornamento sulle tecnologie per il power

IL TARGET

Tecnici, progettisti, buyer e operatori dell'elettronica di potenza in generale

IL FORMAT

Una mostra convegno con conferenze, dibattiti e workshop pianificati da Assodel con esperti prescelti a fornire lo stato dell'arte che valga a riferimento del settore

Community - area demo per l'incontro tra domanda e offerta di fornitori e utilizzatori

Conference - un filo conduttore di interventi selezionati e dedicati alle novità di prodotto, alle applicazioni tecnico/commerciali, alle case history

Educational - seminari formativi e tutorial applicativo-progettuali per tecnici

20 SETTEMBRE 2012

IX EDIZIONE

BOLOGNA

CENTERGROSS ZANHOTEL

Via Saliceto 8 Bentivoglio (BO)

POWER FORTRONIC

an assodel electronics forum



partecipazione previa registrazione

www.powerfortronic.it

FORTRONIC

Electronics Forum - Fortronic
il "miglior costo contatto"
per qualità ed efficacia

INFORMAZIONI / SEGRETERIA:

Tel 02 210.111.236
marketing@fortronicforum.com

ORGANIZZAZIONE:

TECNO IMPRESE

Via Console Flaminio 19 - 20134 Milano
Tel 02 210.111.1 - **www.tecnoimprese.it**

Rendi il tuo LABORATORIO più completo!

Ti proponiamo strumenti di misura di ultima generazione, alimentatori, generatori di funzione, lenti d'ingrandimento per lavori di precisione, stazioni saldanti e molti altri prodotti con cui effettuare riparazioni o realizzare circuiti elettronici. Tutto questo a prezzi convenienti!



3 STRUMENTI IN 1:
- OSCILLOSCOPIO
- GENERATORE DI FUNZIONI
- ALIMENTATORE

cod. LAB2 ~~€ 290,00~~

NUOVO PREZZO € 249,00



MULTIMETRO DIGITALE PROFESSIONALE MULTI FUNZIONI

cod. DVM4000 € 67,00



LENTE D'INGRANDIMENTO 5 DIOTTRIE CON LAMPADA DA 22W

cod. VTLAMP2WN
€ 45,00



FREQUENZIMETRO 2,4 GHz

cod. DVM13MFC2
€ 96,00



ANALIZZATORE DI COMPONENTI

cod. DCA55 € 75,00



PELLICOLA BLU PER LA REALIZZAZIONE DI CIRCUITI STAMPATI (5pz)

cod. PNP5 € 12,00



STAZIONE SALDANTE 48 W

cod. VTSS4N € 16,90



LAMPADE UVA da 8 e 15W

cod. UVA8 € 5,00
cod. UVA15 € 7,00

TRONCHESINO DI PRECISIONE



cod. VT100 € 2,80

PINZA A BECCHI SEMITONDI RICURVI



cod. VT055 € 2,90

SET 15 CACCIAVITI ASSORTITI



cod. VTSET15 € 12,80

MINI TRAPANO PER LAVORI DI PRECISIONE



cod. VTHD01 € 7,00

PINZA SPELAFILI AUTOMATICA CON TAGLIERINA



cod. VTSTRIP5 € 7,00

Vasto assortimento di prodotti per laboratorio su www.futurashop.it

SET 480 RESISTENZE 1/4W (SERIE E-3)



cod. KRES-E3 € 7,50

SET 610 RESISTENZE 1/4W (SERIE E-12)



cod. KRES-E12 € 9,00

SET 120 DIODI ASSORTITI



cod. KDIODE1 € 8,50

SET 100 TRANSISTOR



cod. K/TRANS1 € 12,00

SET CONDENSATORI CERAMICI



cod. KCAP1 € 12,00

SET CONDENSATORI ELETTROLITICI



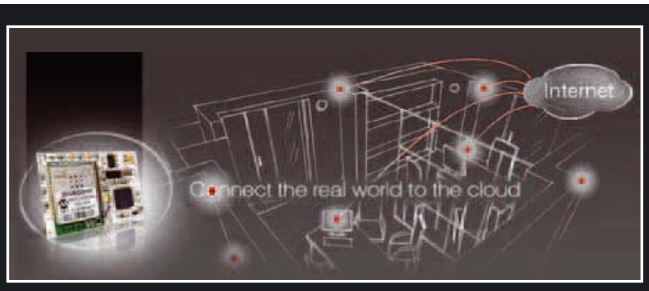
cod. KCAP2 € 11,00

FUTURA ELETTRONICA

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

Prezzi IVA inclusa.

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







Alla scoperta di openPICUS

Finalmente affrontiamo la caratteristica peculiare del mondo openPICUS che permette di realizzare una applicazione basata su webserver utilizzando l'IDE di sviluppo. Ultima Puntata.

4

di Simone Marra

Nella quarta lezione affronteremo uno degli aspetti più caratteristici dei device openPICUS, l'utilizzo della IDE per la creazione di applicazioni dotate di webserver. Partiremo da una pagina web statica, implementando passo-passo variabili dinamiche in grado di aggiornarsi automaticamente, creando una pagina di controllo "realtime" dello stato dell'hardware. In particolare creeremo una pagina web che permetta di controllare lo stato di 5 ingressi digitali, di un ingresso analogico, di pilotare 2 relè e 3 led, e di scrivere delle stringhe su un display LCD 16x2.
Per prima cosa, creiamo un progetto da un template "WiFi 2.1 - Webserver blank example", nominandolo "04 - WebserverCustomElettronicaIN". Se si visualizza il contenuto della cartella "WebPages" dentro le directory del progetto si può vedere che il sorgente delle pagine web è costituito da un solo file chiamato "*index.htm*". Il contenuto del file è il testo "`Welcome to the Flyport blank HTML test page!`" che verrà visualizzato a sua volta tramite browser web.

PERSONALIZZARE UN WEBSERVER

Effettuiamo la prima personalizzazione delle pagine del webserver: all'interno della cartella "webpages" del progetto creiamo il file "index.htm", apriamo il file tramite un editor di testo come Notepad++ o wordpad e inseriamo unicamente il testo "`~stringa~`".

Il testo incluso tra i caratteri ~ viene considerato dal Flyport Wi-Fi come una "variabile dinamica", che viene associata ad una particolare funzione da inserire all'interno del file HTTPApp.c.
Passiamo quindi all'importazione del nuovo webserver tramite il tool integrato della IDE: clicchiamo sull'icona "Web Pages" nella sezione Project Settings, e puntiamo il percorso della cartella del nuovo file index.htm. La OpenPicus IDE provvederà all'importazione e alla conversione della pagina da noi modificata.
A questo punto, se si provasse ad effettuare il processo di compilazione senza modificare il firmware, si riceverebbe un errore: undefined reference to 'HTTPPrint_stringa'. Questo errore viene generato perchè la OpenPicus IDE, durante il processo di importazione delle pagine web, ha rilevato la presenza del testo `~stringa~` ed ha quindi creato un riferimento alla funzione HTTPPrint_stringa() responsabile della restituzione del valore associato al parametro `~stringa~`.
Andiamo quindi a scrivere nel file *HTTPApp.c* la relativa funzione, e cosa debba eseguire. Alla fine del file, prima dell'ultimo *#endif*, aggiungiamo la funzione

```
extern char stringFly [];
void HTTPPrint_stringa(void)
{
TCPPutString(sktHTTP,(BYTE*) stringFly);
}
```

Listato 1

```
#include “taskFlyport.h”
char stringFly[50]=”ElettronicaIN”;

void FlyportTask()
{
        WFConnect(WF_DEFAULT);
        while (WFStatus != CONNECTED);
        vTaskDelay(200);
        UARTWrite(1,”Flyport connected... hello world!\r\n”);
        UARTWrite(1,stringFly);
        int len_s;
        while(1)
        {
//attendi la ricezione di caratteri
                while(!UARTBufferSize(1));
                vTaskDelay(10);                         //attendi il completamento
                len_s = UARTBufferSize(1);    //numero di caratteri ricevuti
                UARTRead(1,stringFly,len_s);//leggi il buffer di ricezione
                stringFly[len_s] = ‘\0’;      //aggiungi il terminatore
        }
}
```

Questa funzione restituirà il valore dell’array di char *stringFly* ogni volta che un client richiederà il valore di `~stringa~`.
Il valore dell’array stringFly sarà invece modificato tramite porta seriale all’interno del FlyportTask, dentro il file taskFlyport.c. (**Listato 1**).
A questo punto compiliamo ed effettuiamo il download del nostro progetto. Accedendo all’indirizzo IP del Flyport Wi-Fi tramite browser web potremo vedere il contenuto della variabile stringFly, che verrà modificata dal task utente.
Per visualizzare i cambiamenti, per ora, è necessario eseguire un refresh della pagina manualmente. Infatti ad ogni refresh verrà nuovamente richiesta la pagina “index.htm”, che contiene la variabile dinamica `~stringa~`. Tale richiesta richiama la funzione HTTPPrint_stringa che leggerà il valore dell’array stringFly, e che quindi aggiornerà il testo del webserver. Vedremo tra poco come far eseguire questo processo di aggiornamento in maniera automatica.

VARIABILI DINAMICHE CON AJAX

Tramite AJAX ed i Javascript è possibile far si che le variabili diventino realmente dinamiche, e che il loro valore venga aggiornato automaticamente all’interno del browser web, senza necessariamente ricaricare l’intera pagina.
Tramite l’utilizzo del file “*mchp.js*” e del file “*status.xml*” possiamo aggiungere funzionalità alla pagina index.htm. Aggiungiamo i due file alla cartella “/WebPages” e modifichiamo il contenuto di status.xml:

```
<response>
<prova>~stringa~</prova>
</response>
```

Listato 2

```
<script src=”/mchp.js” type=”text/javascript”></script>
<div id=”content”>
  <p>Seriale:<br />
        <span id=”prova”>?</span>
  </p>

<script type=”text/javascript”>
<!--
// Parses the xmlResponse from status.xml and updates the status box
function updateStatus(xmlData) {
  // Get the value of stringFly
  document.getElementById(‘prova’).innerText = getXMLValue(xmlData, ‘prova’);
}
setTimeout(“newAJAXCommand(‘status.xml’, updateStatus, true)”,500);
//-->
</script>
```

mentre il nuovo index.htm sarà come riportato nel **Listato 2**.
In questo modo, dichiariamo che l'elemento con id='prova' prenderà il suo valore dal file status.xml che contiene la direttiva ~stringa~ che verrà aggiornata di volta in volta automaticamente dal firmware del flyport tramite la funzione HTTPPrint_stringa, perchè la funzione verrà richiamata grazie ai continui aggiornamenti delle variabili degli script AJAX del file "*mchp.js*".
Per vedere le modifiche della stringa non sarà più necessario premere sul bottone di refresh della pagina web, ma il valore cambierà in tempo reale non appena verrà inviata una nuova stringa sulla porta seriale del Flyport.

WEBSERVER DI ESEMPIO PER UNA SCHEDA DI ESPANSIONE

Come esempio di utilizzo dei precedenti argomenti in un contesto applicativo, riportiamo l'esempio di un webserver in grado di visualizzare lo stato di una grandezza analogica e di 5 ingressi digitali, di controllare 2 relè e 3 led e di pilotare il display LCD 2x16.
Il progetto "***04 - WebserverCustomHardware***" si avvale di 2 librerie esterne, una (*lcd16x2*) per le funzionalità legate al display LCD, un'altra (*DevBoardInit*) per inizializzare la scheda di sviluppo in modo da poter utilizzare i nomi dei segnali della scheda, e non i segnali del modulo (per esempio B1 invece che p5). Il firmware all'interno del task "FlyportTask()" oltre ad alcune inizializzazioni, permette di scrivere sul display alcune informazioni relative ai parametri del Flyport Wi-Fi tramite la pressione dei pulsanti di ingresso:

1. se viene premuto il bottone 1, sul display viene visualizzato l'indirizzo IP attuale;
2. se viene premuto il bottone 2, sul display viene visualizzato l'SSID della rete a cui si è connessi;
3. se viene premuto il bottone 3, sul display viene scritto il nome di *Netbios* del modulo Flyport Wi-Fi (si potrà quindi accedere al webserver del Flyport tramite *http://indirizzoip oppure http://nomenetbios*);
4. se viene premuto il bottone 4, sul display viene visualizzato il valore dell'ingresso analogico;
5. se viene premuto il bottone 5, sul display viene visualizzato il MAC address del transceiver Wi-Fi.

La gestione degli eventi e delle funzionalità del webserver, invece, sono gestite all'interno del file "**HTTPApp.c**".

Fig. 1



La lettura dei valori delle variabili (sia analogiche che digitali) sono effettuate tramite AJAX e le relative funzioni *HTTPPrint_pot*, *HTTPPrint_btn* ed *HTTPPrint_led*. La *HTTPPrint_pot* restituirà il valore dell'ingresso analogico (il potenziometro), la *HTTPPrint_btn* restituirà invece gli stati dei 5 pulsanti di ingresso, mentre la *HTTPPrint_led* resistuirà lo stato delle 5 uscite, ovvero i 3 led ed i due relè. La modifica dello stato delle uscite (i 3 led ed i 2 relè) tramite intervento su pagina web viene effettuata tramite il metodo "GET" con una richiesta CGI al file "*leds.cgi*". Premendo/cliccando i relativi pulsanti virtuali dei 3 led e dei due relè, si ecciterà l'evento "onclick". Questo evento, a sua volta, lancia la funzione "*newAJAXCommand*" che risiede nel file "mchp.js" e che esegue una richiesta CGI. Nella funzione *HTTPExecuteGet* viene prima analizzato il file che ha effettuato la richiesta, se il file è "*leds.cgi*" allora significa che la richiesta *GET* è relativa al cambiamento di stato di una uscita.
A questo punto il firmware analizza quale variabile deve essere modificata, ed esegue il relativo codice. Lo stesso metodo viene utilizzato per aggiornare il display con il contenuto di due *input form* nella pagina del webserver. Tramite la pressione del tasto di invio, il contenuto delle due form viene inviato al display, dato che il metodo GET scatena la funzione *HTTPExecuteGet* con file di provenienza di nome "*index.htm*". Le stringhe da scrivere sul display vengono prese tramite la funzione *HTTPGetArg* la quale legge il contenuto delle form con nome "row1" o "row2", rispettivamente riga 1 e riga 2.

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

RADIANT

A N D • S I L I C O N

59° MERCATO DEL CONSUMER ELECTRONICS

22 - 23 SETTEMBRE 2012

sab. 9.00-18.00 - dom. 9.00-17.00



in contemporanea

NOVEGRO PHOTO Video CINE

17ª Edizione

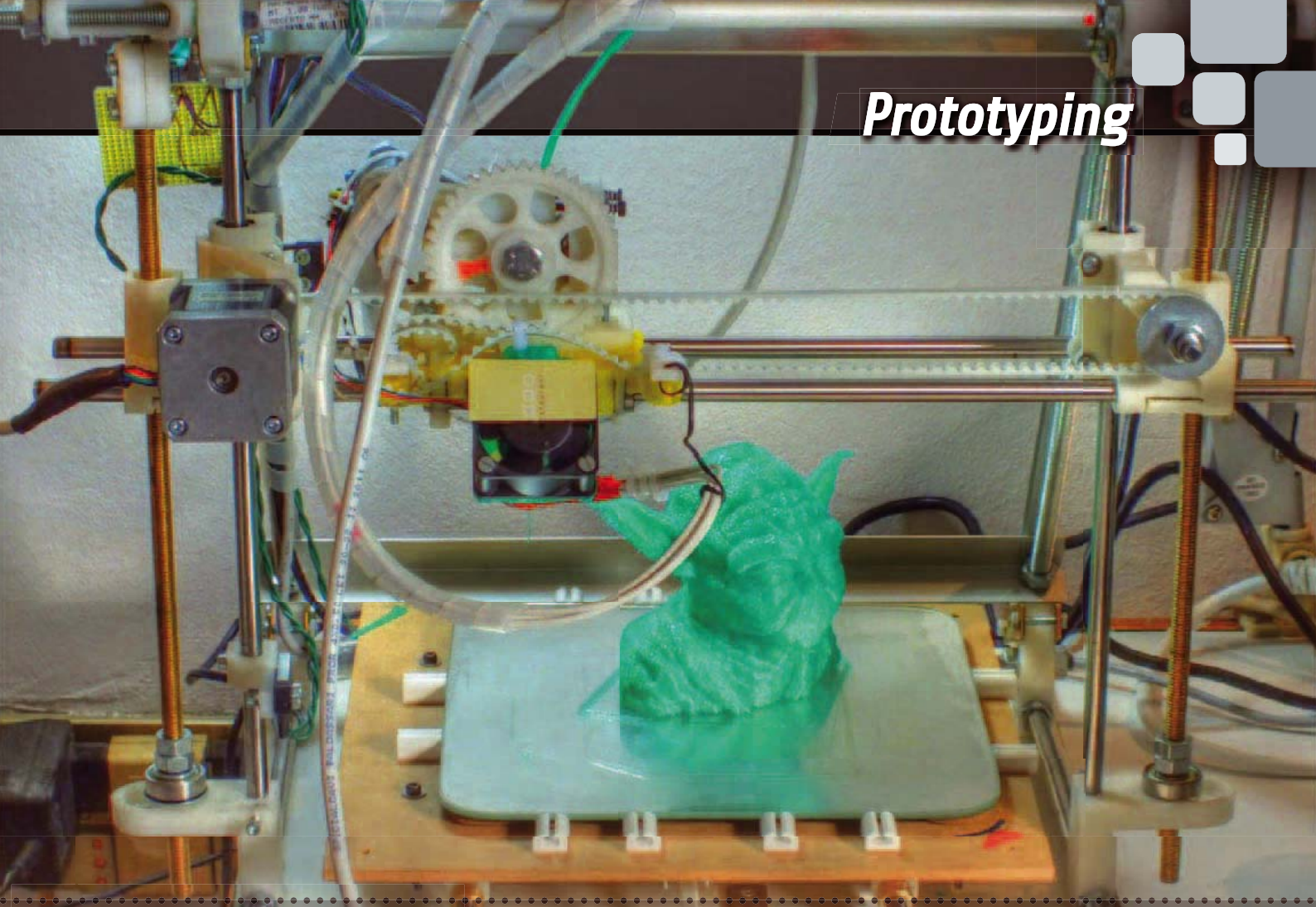
Mostra-Mercato di apparecchi fotografici e cinematografici

PARCO ESPOSIZIONI NOVEGRO

MILANO/LINATE AEROPORTO ✈

Con il patrocinio di
Città di Segrate
Assessorato alla Cultura

www.parcoesposizioninovegro.it



Esistono da decenni, ma solo da circa due anni sono entrate a far parte del mondo degli hobbysti, diventando oggi uno dei temi più caldi del momento.

LE STAMPANTI 3D APPRODANO NEL MONDO "FAI DA TE"

di SIMONE MAJOCCHI

L'idea della prima stampante della famiglia Open Source RepRap – chiamata Darwin - nasce in Inghilterra nel 2006, ma prima di essere rilasciata al pubblico devono trascorrere ancora due anni. Per noi hobbysti, si tratta di un momento fondamentale in una storia che prende l'avvio nel "lontano" 1984; una storia che vede il suo punto di svolta nel 2012, con un'accelerazione tipica delle tecnologie che hanno superato la massa critica. Da qualche mese, infatti, la stampa 3D è il tema tecnologico più caldo su tutti i forum e su tutti i blog orientati alla tecnologia e, a conferma che nulla è mai per caso, dietro a RepRap c'è Arduino e una sua evoluzione.

QUASI 30 ANNI DI STORIA

Charles Hull, nato nel 1939 e titolare di oltre 60 brevetti nel campo dell'ottica e della prototipazione rapida, fondò nel 1984 la società 3D Systems e nel 1986 ottenne il brevetto per un "Apparatus for Production of Three-Dimensional Objects by stereo lithography" (Apparato per la produzio-

Il **formato dei file** di **stampa 3D**

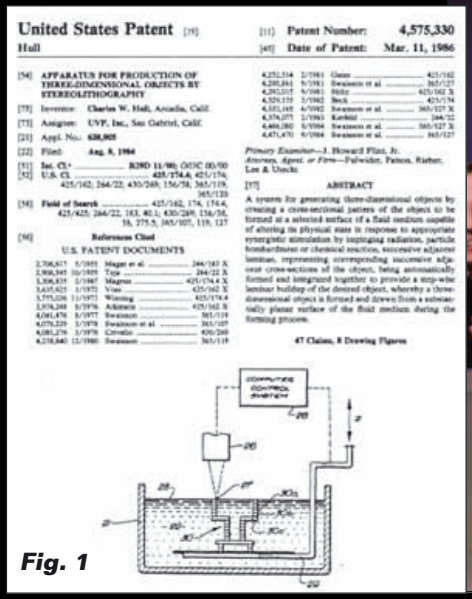
Per poter stampare correttamente un file 3D, non basta che venga definito con la corretta estensione "STL", ma deve anche essere privo di quei difetti che possono creare complicazioni in fase di "slicing". Il formato STL (STereoLithography) nasce da 3D Systems come soluzione per la stampa dei modelli 3D sulle proprie macchine. Dato che per questo scopo non servono particolari informazioni se non il reticolo di triangoli che descrive la superficie dell'oggetto, STL non contiene informazioni relative a colori o ad altri parametri e quindi la sua sintassi, nella versione testuale, è molto semplice.

Es:

```
solid OBJECT
  facet normal 0 -0.99914427180083842 -0.041360900952137368
    outer loop
      vertex -4.607335090637207 9.9937305450439453 -13.645944595336914
      vertex -4.2146701812744141 9.9888019561767578 -13.526885986328125
      vertex -4.607335090637207 9.9888019561767578 -13.526885986328125
    endloop
  endfacet
  facet normal 0 -0.99956544409214843 -0.029477499501591287
    outer loop
      vertex -4.2146701812744141 9.9972267150878906 -13.764497756958008
      vertex -4.607335090637207 9.9937305450439453 -13.645944595336914
      vertex -4.607335090637207 9.9972267150878906 -13.764497756958008
    endloop
  endfacet
  facet normal -0.0018000171410936154 -0.99937040963595669 -0.035433660301355953
    outer loop
      vertex -4.2146701812744141 9.9888019561767578 -13.526885986328125
      vertex -4.607335090637207 9.9937305450439453 -13.645944595336914
      vertex -4.2146701812744141 9.9972267150878906 -13.764497756958008
    endloop
  endfacet
```

ne di oggetti tridimensionali attraverso la stereo litografia). Il concetto era la deposizione di una serie di strati solidi sovrapposti attraverso l'indurimento di una resina sensibile agli ultravioletti. Il brevetto prevedeva varie modalità per l'esposizione agli ultravioletti della resina e il modello solido poteva affondare nel bagno di resina o esserne estratto (potete trovare il brevetto in figura 1 e all'indirizzo : http://www.google.com/patents/US4575330).

Mentre negli anni 80 la stereolitografia veniva sviluppata da 3D Systems, Scott Crump, fondatore di Stratasys, lavorava su un altro tipo di tecnologia, quella del Fused Deposition Modelling, ovvero della modellazione tramite deposizione di materiale fuso, definizione diventata



United States Patent [19]
Hull

[11] Patent Number: 4,575,330
[45] Date of Patent: Mar. 11, 1986

[54] APPARATUS FOR PRODUCTION OF THREE-DIMENSIONAL OBJECTS BY STEREOLITHOGRAPHY

[75] Inventor: Charles W. Hull, Arcadia, Calif.

[73] Assignee: UVP, Inc., San Gabriel, Calif.

[21] Appl. No.: 638,905

[22] Filed: Aug. 8, 1984

[51] Int. Cl.[4] B29D 11/00; G03C 00/00

[52] U.S. Cl. 425/174.4; 425/174; 425/162; 264/22; 430/269; 156/58; 365/119; 365/120

[58] Field of Search 425/162, 174, 174.4, 425/425; 264/22, 163, 40.1; 430/269, 156/58, 59, 275.5, 365/107, 119, 127

[56] References Cited
U.S. PATENT DOCUMENTS

Primary Examiner—J. Howard Flint, Jr.
Attorney, Agent, or Firm—Fulwider, Patton, Rieber, Lee & Utecht

[57] ABSTRACT

A system for generating three-dimensional objects by creating a cross-sectional pattern of the object to be formed at a selected surface of a fluid medium capable of altering its physical state in response to appropriate synergistic stimulation by impinging radiation, particle bombardment or chemical reaction, successive adjacent laminae, representing corresponding successive adjacent cross-sections of the object, being automatically formed and integrated together to provide a step-wise laminar buildup of the desired object, whereby a three-dimensional object is formed and drawn from a substantially planar surface of the fluid medium during the forming process.

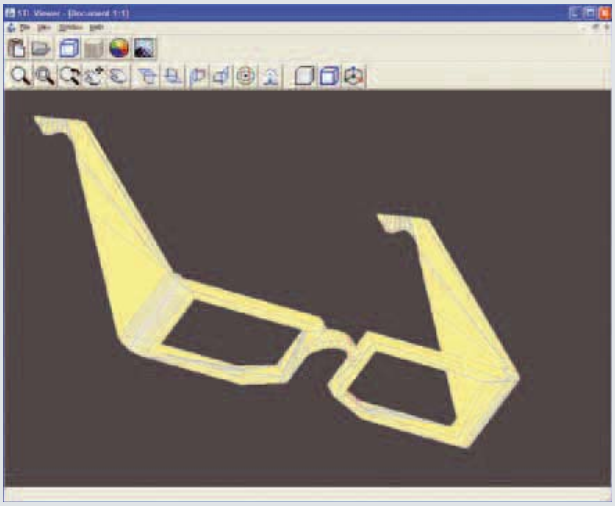
47 Claims, 8 Drawing Figures

Fig. 1



Fig. 2

Dopo l'intestazione "solid OBJECT" (con OBJECT che potrebbe essere un nome qualsiasi), inizia la sequenza dei triangoli che vengono definiti attraverso la normale della faccia e i tre vertici della faccia triangolare. I vertici e la normale vengono definiti come serie di coordinate X, Y e Z indicate come numero decimale a 32 bit di precisione. Usando la regola della mano destra viene definito quale sia l'interno e l'esterno del solido.



Se i triangoli non creano un reticolo contiguo o ci sono problemi di orientamento delle facce, il file può creare problemi di elaborazione successiva da parte del software di "taglio" o slicing.
La versione binaria di STL contiene le medesime informazioni, ma le memorizza in modo molto più compatto come sequenza di 12 valori a 32 bit che corrispondono alla normale e ai tre terzetti di coordinate. L'intestazione è da 80 caratteri seguita da un valore a 32 bit che indica il numero di triangoli del file, dopodiché iniziano le sequenze dei triangoli.
I file di tipo STL possono essere visualizzati da applicazioni specifiche e spesso gratuite come STL Viewer, Netfabb o OpenSTL.



***Fig. 4** - La Genisys di Stratsys, 1996.*

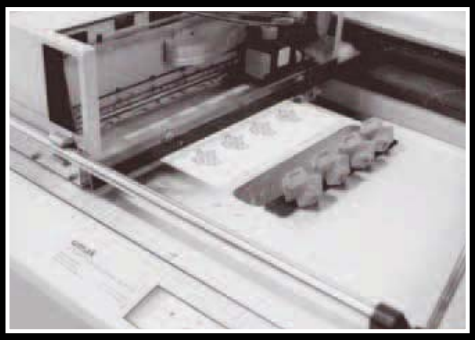
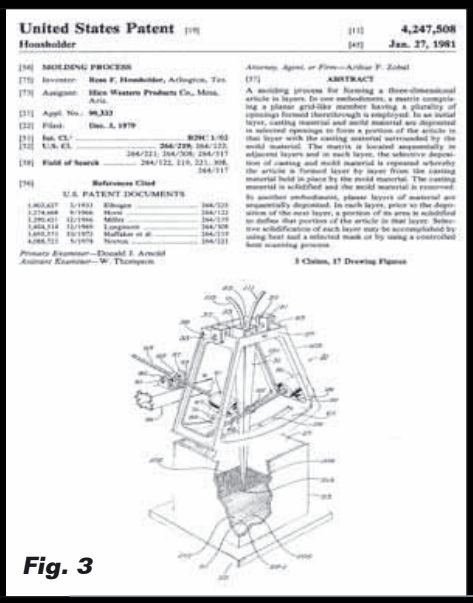


***Fig. 5** - La Z402 di Z Corporation, 1996.*



***Fig. 6** - La Actua 2100 di 3D Systems, 1996.*

marchio di Stratasys già nel 1988 (Fig. 2).
A questa serie di tecnologie si aggiunge quella sviluppata dal MIT che nel 1993 brevetta le "3 Dimensional Printing techniques" (tecniche di stampa su 3 dimensioni) basate sulla modifica di tecnologie delle stampanti 2D. Questo brevetto viene dato in licenza a Z Corporation nel 1995 per lo sviluppo delle proprie 3DP printer (Fig. 3).
Tutto questo sfocia nel 1996 in una prima serie di prodotti commerciali abbordabili e di elevate prestazioni da parte di tutte e tre le aziende, dando il via al mercato professionale della stampa di modelli solidi per la prototipazione rapida e per la realizzazione di modelli e mock up (Fig. 4, 5 e 6).
Per i successivi dieci anni, la stampa in 3D resta un'applicazione professionale e di nicchia, con l'industria che ne beneficia senza che il grande pubblico se ne renda conto.
Nel 2006, però, la strada scelta da Crump e Stratasys – con il Fused Deposition Modelling – diventa anche quella scelta dal progetto Open Source RepRap. Per evitare problemi di tipo legale, MDF diventa FFF ovvero Fused Filament Fabrication e il gruppo iniziale degli sviluppatori è inglese. L'obiettivo del progetto era realizzare una stampante in grado di replicare se stessa stampando le parti in plastica necessarie per la sua costruzione.
In due anni, nel 2008, viene finalizzata la prima RepRap Darwin, capace di produrre il 50% delle sue parti, mentre il resto era di facile reperimento sul mercato. Dal 2008 in poi, per il mondo della stampa 3D cambia un po' tutto: l'interesse è prima limitato a un gruppo ristretto di "pionieri" della tecnolgia, ma con



United States Patent [19]
Housholder
[11] 4,247,508
[45] Jan. 27, 1981

[54] MOLDING PROCESS
[75] Inventor: Ross F. Housholder, Arlington, Tex.
[73] Assignee: Hico Western Products Co., Mesa, Ariz.
[21] Appl. No.: 90,333
[22] Filed: Dec. 3, 1979
[51] Int. Cl.³ B29C 1/02
[52] U.S. Cl. 264/219; 264/122; 264/221; 264/308; 264/317
[58] Field of Search 264/122, 219, 221, 308, 264/317

[56] References Cited
U.S. PATENT DOCUMENTS

Primary Examiner—Donald J. Arnold
Assistant Examiner—W. Thompson
Attorney, Agent, or Firm—Arthur F. Zobal

[57] ABSTRACT

3 Claims, 17 Drawing Figures

Fig. 3

la nascita nel 2009 di Makerbot Industries, la desktop 3D printer alla portata di tutte le tasche è sempre più vicina (Fig. 7). Makerbot Industries è frutto dell'iniziativa di Bre Pettis, Adam Mayer e Zach "Hoeken" Smith; partendo dai primi lavori e dai primi risultati del gruppo RepRap, Makerbot realizza Cupcake CNC e la rilascia sul mercato come progetto Open Source. Questo permette, principalmente attraverso il sito Thingiverse (*www.thingiverse.com*), lo sviluppo di numerose migliorie e modifiche.

A fine 2010, la Makerbot Industries presenta la nuova Thing-o-Matic che recepisce molte delle modifiche sviluppate e condivise dagli utenti, trasformando la precedente Cupcake in una stampante 3D capace di una miglior definizione, una stampa più stabile e anche qualche forma di "produzione a catena". Il successo è notevole e altre aziende iniziano a valutare la possibilità di prendere il notevole lavoro del progetto RepRap per trasformarlo in prodotti in kit, ancora open source, ma di costo sempre più competitivo. La differenza fra una Makerbot e una RepRap restava nella impossibilità di fare altre Makerbot partendo da una Makerbot in quanto parte della struttura era fatta con taglio laser, mentre RepRap ha sempre tenuto i componenti sotto forma di viti, bulloni, barre, cuscinetti e parti in plastica stampabili.

Anche Thing-o-Matic è stata però rilasciata in open source permettendo comunque agli utenti di sviluppare e condividere le proprie migliorie.

Nella primavera 2012 il posto di questa stampante viene preso da Makerbot Replicator, capace di

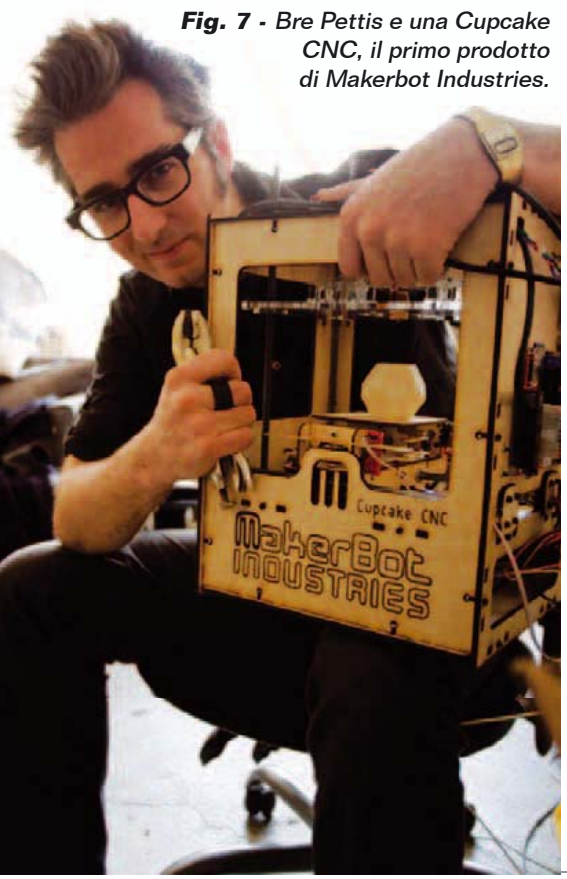


***Fig. 7** - Bre Pettis e una Cupcake CNC, il primo prodotto di Makerbot Industries.*

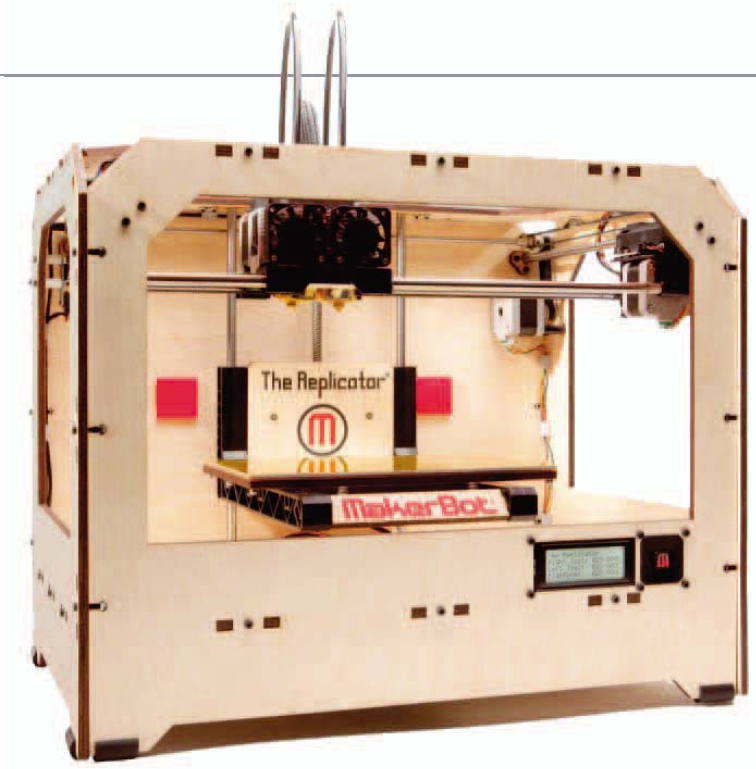


***Fig. 8** Makerbot Replicator è la stampante che dalla primavera 2012 viene venduta da Makerbot Industries, in kit a circa 1700 dollari.*



***Fig. 9** - Printrbot: 800 mila dollari in un mese.*

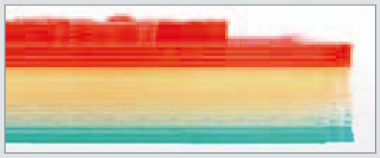
stampare a due estrusori contemporaneamente e con un piano di stampa molto più ampio (**Fig. 8**). Altre aziende scelgono la strada di Makerbot Industries e rapidamente la danese Ultimaker attira l'attenzione degli utenti con il suo prodotto "di riferimento", capace di stampare con precisione ad elevate velocità. Ancora una volta si parte da RepRap, ma con la produzione in serie, alcuni elementi vengono gestiti in modo diverso. Fra le peculiarità di Ultimaker, capace di 350 mm/s come velocità, c'è un'area di stampa di 21x21x21 cm, la più ampia della categoria.

UN MERCATO IN ESPLOSIONE

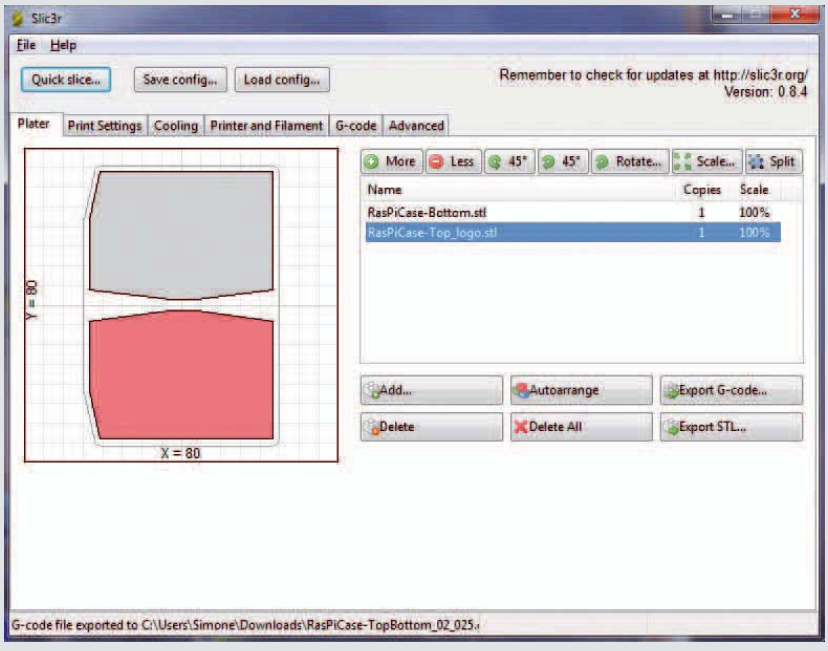
Il termine "esplosione" non è stato scelto a caso in quanto nel giro di pochi mesi sono diventate molte le aziende che offrono kit e stampanti a basso costo, anche ricorrendo al finanziamento come quello di Kickstarter. Il progetto Printrbot, ad esempio, ha raccolto oltre 800 mila dollari di finanziamento contro i 25 mila richiesti, ottenendo 1800 supporter e vendendo di fatto - ancora prima di creare un'azienda - un numero significativo di stampanti (**Fig. 9**).
Altre aziende sono nate in forma più o meno artigianale fornendo la propria interpretazione al progetto RepRap per crearne una versione con migliori prestazioni, un prezzo più abbordabile o un più semplice assemblaggio. Anche in Italia abbiamo l'esempio di Kent's Strapper che, partendo da una RepRap Prusa, ha realizzato la Galileo 1.0, meno costosa e più semplice. La loro iniziativa imprenditoriale ha attirato un interesse mediatico non indifferente su tutto il mondo della stampa 3D dei Makers e dei Fablab, dimostrando come anche in Italia ci sia inventiva e voglia di fare.



Basta osservare una stampante 3D per capire come qualsiasi oggetto sia costruito attraverso la sovrapposizione di una serie di "fettine" orizzontali. Le fettine sono composte da un perimetro e da una parte interna che può anche essere racchiusa fra l'interno e l'esterno del perimetro. Il file di partenza è un STL, ovvero un reticolo composto da triangoli orientati che definiscono la "pelle" dell'oggetto da stampare.
Da quel file alle fettine da stampare il passo è apparentemente breve, ma al tempo stesso molto complesso. I programmi che si occupano di questo processo, chiamato slicing, sono numerosi, ma nel segmento dell'open source sono due a fare la parte del leone: Skeinforge e Slic3r.
Nati entrambi per trasformare i file STL in G-Code, ovvero in una sequenza di movimenti che compone ciascuno strato, nel tempo si sono sviluppati in applicazioni molto sofisticate e parametriche grazie alle quali ciascun utente può cercare di ottenere il miglior risultato dalla propria stampante 3D modificando tutti i parametri principali, dalla velocità di stampa alla quantità di filo da estrudere durante il tracciamento di ciascun singolo segmento.
I programmi di slicing devono quindi creare il modello nello spazio tridimensionale ed intersecarlo con un piano che si sposta pian piano sull'asse Z dal basso verso l'alto. La sezione deve



essere quindi elaborata per individuare il perimetro, le parti vuote e le parti piene, tenendo anche conto dei livelli precedenti e quelli successivi. Nel caso, ad esempio, di uno sbalzo verso l'esterno o di un piano che è presente solo da una certa altezza in poi, va generato un supporto, ovvero del materiale che permette all'estrusore di trovare qualche cosa su cui appoggiare il filo di estrusione durante la deposizione della parte sospesa. Il riempimento può essere realizzato con una percentuale variabile da 0 (vuoto) a 100 (tutto pieno) cambiando in modo significativo il consumo di filo, il tempo di stampa e la robustezza dell'oggetto finito. In pratica, quindi, l'effetto del programma di slicing sul risultato finale è determinante. Nei prossimi articoli entreremo nel dettaglio di questi programmi, valutandone i vari parametri e i loro effetti.



I **materiali di stampa** (ABS e PLA)

Oltre a scegliere quale modello stampare, con una stampante 3D si ha anche la possibilità di scegliere il materiale con cui si vuole alimentare l'estrusore. Nel segmento delle stampanti hobbystiche di tipo RepRap, la scelta è generalmente limitata a due tipi di materiale: ABS e PLA. Il primo è un materiale plastico abbastanza diffuso e conosciuto, con cui sono stampati numerosi prodotti di uso comune, come ciotole, complementi d'arredo e parti di scooter e autovetture. Dal punto di vista chimico, l'ABS è Acrylonitrile butadiene styrene, ovvero un copolimero creato combinando tre diversi materiali per ottenere resistenza, aspetto lucido ed elasticità. L'ABS può essere formulato con diverse proporzioni fra acrilonitrile, butadiene e stirene per variare la temperatura di fusione, la resistenza e la plasticità, mentre l'aggiunta di pigmenti permette di variare la colorazione del materiale che altrimenti sarebbe neutro.
La temperatura di estrusione dell'ABS varia fra i 240 e i 260 gradi, con un effetto di ritiro e distacco dal piano di stampa abbastanza pronunciato e questo richiede solitamente o un piano di stampa riscaldato o un trattamento speciale della superficie di stampa.
Il PLA, Acido Polilattico, è un materiale di origine biologica in quanto è ottenuto attraverso la polimerizzazione dell'acido lattico. Questo deriva da amido di mais e canna da zucchero ed è biodegradabile nelle opportune condizioni climatiche e ambientali. È stato utilizzato per realizzare packaging e tessuti, mentre per la stampa 3D presenta delle interessanti caratteristiche quali una temperatura di fusione attorno ai 160 gradi, che aumenta in base agli eventuali altri componenti aggiunti per la colorazione e una maggiore plasticità. Il PLA si sta rapidamente diffondendo come materiale di stampa sia per i colori e le trasparenze particolarmente accattivanti, sia per i minori problemi legati alla stampa. Le parti in PLA sono comunque sufficientemente resistenti per creare parti di stampante, come per l'ABS. A differenza dell'ABS che può essere trattato con acetone, il PLA richiede trattamenti più complessi e pericolosi per il suo discioglimento e ritocco (tipicamente Soda Caustica).



Nei prossimi mesi vedremo altri casi di successo, altre iniziative e un fiume in piena di stampanti 3D, ciascuna con peculiarità e caratteristiche tali da alimentare un mercato in espansione con la diversità e la varietà.

TUTTE SIMILI, TUTTE DIVERSE

Come vedremo nei prossimi articoli dedicati alla stampa 3D, gli elementi che differenziano e caratterizzano ciascun modello sono numerosi e riguardano principalmente:

- Dimensioni area di stampa
- Numero di estrusori / Materiali
- Velocità di stampa / stabilità
- Risoluzione / estrusore
- Funzionalità per la stampa autonoma
- Open / closed alle modifiche

La stampante ottimale che offre il massimo in ciascuno di questi elementi non esiste e il compromesso fra prezzo e prestazioni che ciascun produttore ricerca con il proprio modello è appunto quello che giustifica la scelta di un modello piuttosto che un altro. Anche a livello pratico, possono servire solo alcune "eccellenze" a scapito di peggiori prestazioni su altri fronti e proprio imparando a conoscere cosa si vuole stampare diventa possibile acquistare il prodotto giusto.
La scelta di un modello Open Source rispetto a uno proprietario, poi, può essere una strada che fa faticare di più in fase di assemblaggio, ma poi consente di modificare con maggiore efficacia e una significativa serie di suggerimenti la propria stampante.

COSA FARE CON UNA 3D PRINTER?

Questa è la domanda più semplice e al tempo stesso più complessa del momento. Le potenzialità

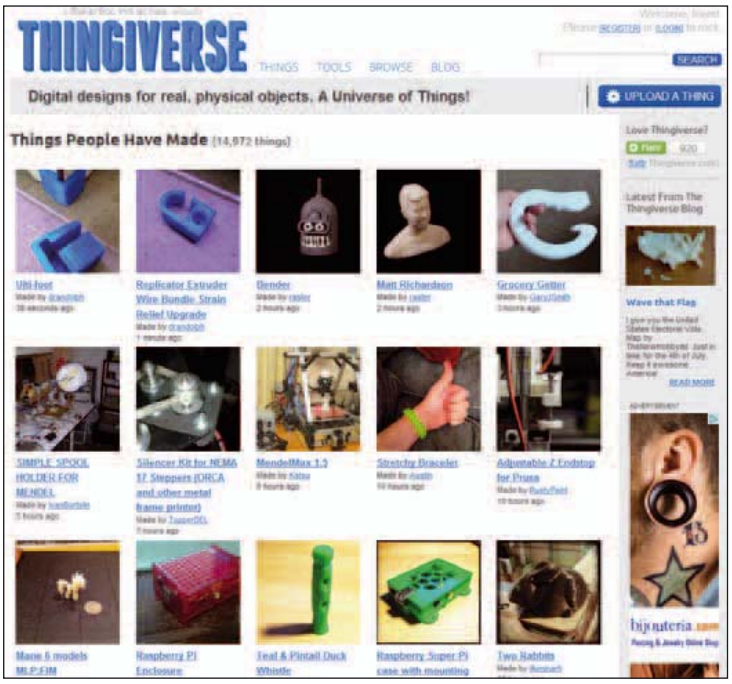


Fig. 10

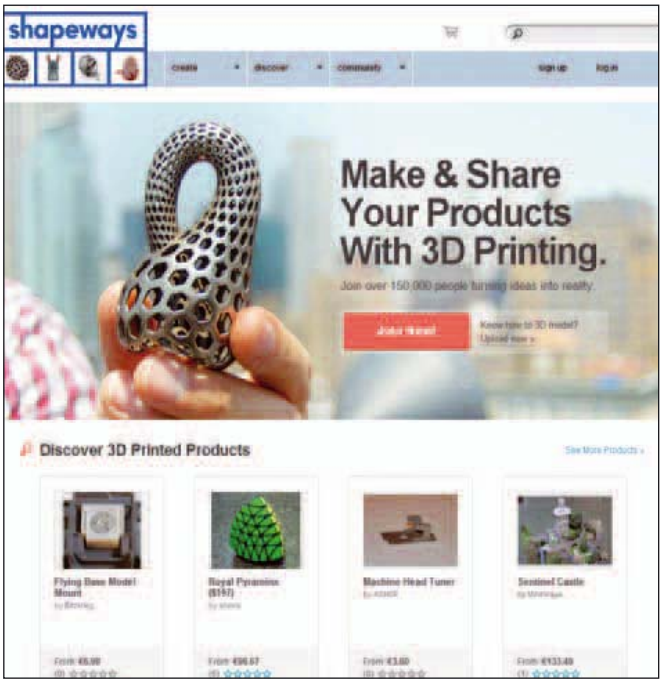


Fig. 11

della tecnologia FFF sono infatti molte, ma come all'inizio dell'era informatica domestica si diceva che "con il computer si può fare tutto" e tutti scoprivano che servono i programmi per far fare ai computer quello che interessava, allo stesso modo siamo all'inizio della rivoluzione del 3D desktop printing e dopo l'iniziale eccitazione del "posso stampare qualsiasi cosa", stiamo scoprendo che imparare a usare un programma per la modellazione 3D non è proprio banale e che ci sono mille problemi da risolvere prima di arrivare al modello che cercavamo di realizzare. Anche una banale fibbia per cintura, nella sua semplicità, va realizzata con perizia, altrimenti "non stampa" come vorremmo.

In risposta a questi problemi – o meglio – a questa inevitabile esigenza di imparare a creare quello che vogliamo stampare, stanno nascendo siti che mettono in comune i modelli già realizzati da altri utenti. Siti come Thingiverse o Shapeways o Tinkercad contengono il lavoro di migliaia di utenti e artisti del 3D che, un po' per spirito di condivisione, un po' nella speranza di essere notati da qualche azienda, hanno accettato di rendere i propri file scaricabili e talvolta modificabili attraverso i sorgenti CAD.

GUARDARE PRIMA DI CREARE

A meno che non siate dei provetti utenti di applicazioni CAD, è improbabile che scopriate un innato talento per domare le complessità di un programma di modellazione tridimensionale. Niente di trascendentale, certo, ma non è come mettere insieme una paginetta su Word o ritoccare un po' un'immagine con Photoshop.

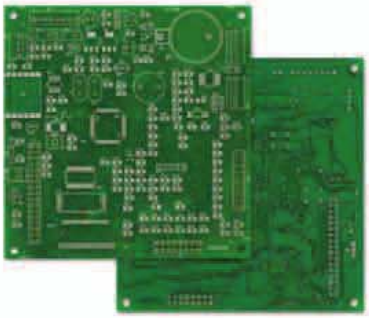
Per avere quindi un quadro più preciso di cosa si riesce a produrre e cosa è facile da stampare, un bel giro fra modelli e fotografie di oggetti su Thingiverse (**Fig. 10**) è la cosa migliore da fare. Qualcuno potrebbe avere delle grosse aspettative e rimanere deluso della relativa semplicità delle cose che si possono fare, mentre altri potrebbero scoprire che la 3D printer è già capace di andare oltre le loro esigenze.

Certo è che le immagini valgono più di mille parole e quindi le fotografie che mostrano il medesimo modello stampato da una decina di utenti diversi, con risultati altrettanto diversi vi aiuterà a capire gli effetti pratici di termini quali la risoluzione, l'incremento sull'asse Z o il diametro dell'estrusore.

Chi invece cerca modelli di elevata qualità e non alla portata di qualsiasi stampante 3D, può far visita su *www.ponoko.com* o su *www.shapeways.com* (**Fig. 11**) per curiosare fra oggetti di vario genere e tipo, fino ad arrivare ai sofisticati gioielli e alle opere d'arte di *i.materialise.com*. Tutti e tre i siti offrono modelli da scaricare, modelli fatti da altri utenti, pronti per essere stampati ed acquistati o la stampa di un vostro modello con materiali e tecnologie diverse.

Il prossimo mese analizzeremo il fenomeno RepRap con tutte le sue derivazioni hobbystiche e commerciali, presentando le più aggiornate versioni di software e firmware per la stampa con questa tipologia di 3D printer.

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**

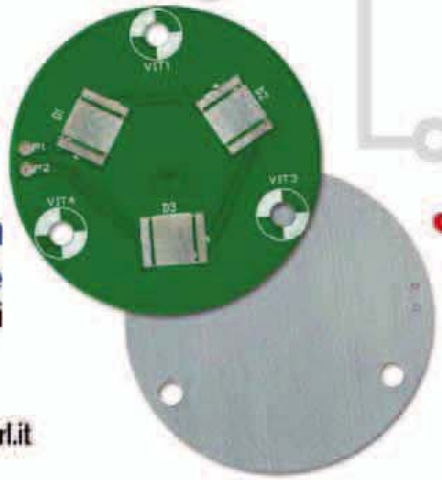
qualità

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



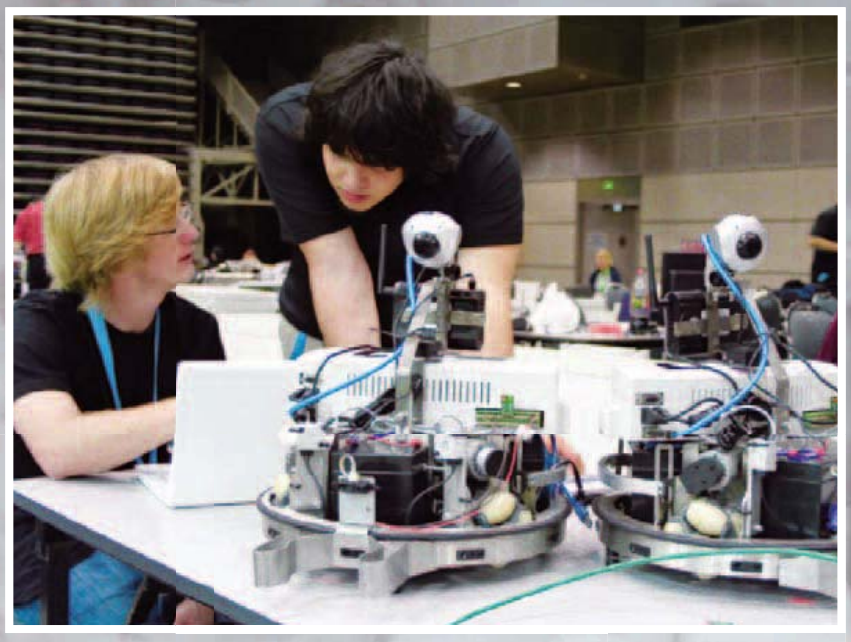
Chi ha vinto e come si sono piazzati i team italiani alla finale mondiale 2012 di Città del Messico?

a cura della Redazione

RoboCup Messico: com'è andata?

Si è da poco conclusa la finale mondiale della RoboCup 2012, la competizione che ogni anno vede misurarsi in prove suddivise per categorie i robot realizzati e programmati da gruppi di giovani studenti provenienti da tutto il mondo e selezionati durante le annuali edizioni locali (nel nostro paese le selezioni riguardano solamente la RoboCup jr). Come saprete, perché vi abbiamo tenuti costantemente aggiornati tramite la rivista ed il nostro blog, per l'Italia alla finale mondiale sono andate 10 squadre. Un po' tutti aspettavamo di sapere se le nostre rappresentative partite alla volta di Città del Messico, sarebbero riuscite a conquistare un posto sul podio. Ma parlare di classifiche e di vittoria in questo ambito ha poco senso, in quanto alla RoboCup sono un po' tutti vincitori, anche solo per il fatto di aver portato ad una finale mondiale i propri progetti, il proprio lavoro e l'impegno che lo ha contraddistinto. RoboCup nasce nel 1997 come iniziativa educativa, con l'intento di promuovere la ricerca nei campi della robotica e dell'intelligenza artificiale, offrendo visibilità e proponendo competizioni spettacolari tra studenti. Per i ragazzi e i docenti che li hanno accompagnati, il viaggio a Città del Messico è stato un'esperienza entusiasmante: gli studenti si sono confrontati con loro pari arrivati da tutto il mondo, hanno studiato le loro strategie e le loro tattiche e si sono scambiati informazioni e consigli con le squadre che hanno partecipato alle precedenti edizioni della manifestazione.
Oltre che per incentivare la ricerca e avvicinare i giovani alla robotica, RoboCup si propone l'ambizioso obiettivo di far nascere dal lavoro degli studenti una squadra di robot umanoidi cui far disputare e vincere, per il 2050, un incontro di calcio contro giocatori in

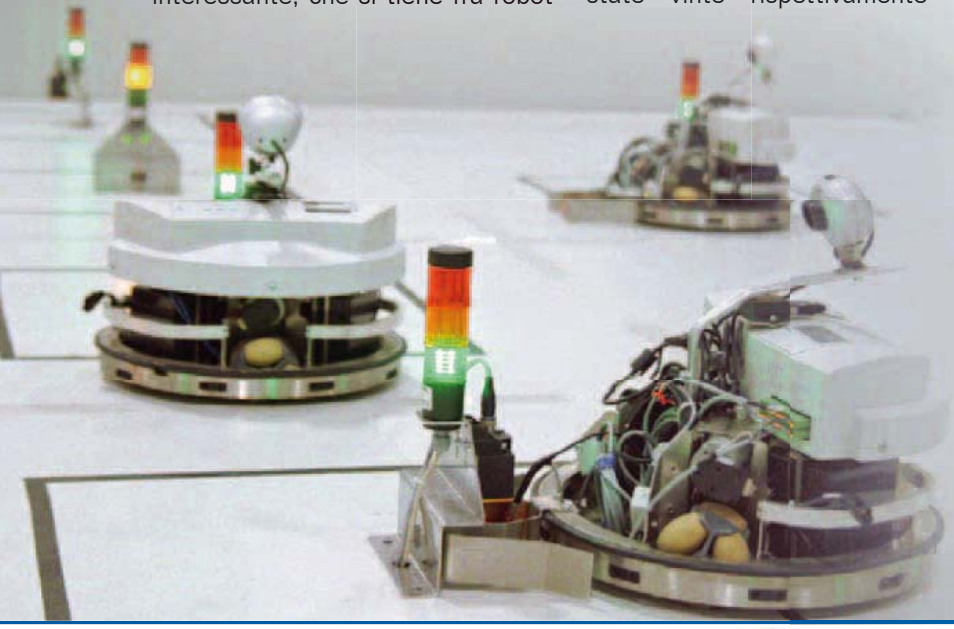




carne ed ossa (la squadra vincitrice dell'ultimo campionato mondiale di calcio FIFA).
Non a caso la categoria più attesa è sempre stata la RoboCup Soccer, che consiste nel far disputare incontri di calcio tra robot costruiti dai team di scuole e nella quale valgono un po' le regole del gioco del calcio tra umani: vince la squadra che segna più reti.
RoboCup Soccer, consiste in una vera e propria partita di calcio tra robot autonomi e punta, attraverso lo sviluppo di strategie di automazione finalizzate al gioco del calcio, a realizzare soluzioni adatte ad applicazioni utili nel campo del sociale e dello sviluppo industriale. La specialità comprende diversi campionati, ai quali partecipano gruppi di età e formazione crescente. Le categorie previste sono:

- Simulation League
- Small Size Robot League
- Middle Size Robot League
- Standard Platform League
- Humanoid League

La prima riguarda simulatori bi e tridimensionali; la seconda, la terza e la quarta le piattaforme robotiche di dimensioni piccole, medie e grandi. La quinta è la competizione più interessante, che si tiene fra robot umanoidi divisi in tre categorie in base all'altezza: Kid Size, 30÷60 cm, Teen Size 90÷120 cm e Adult Size, oltre 130 cm); a vincerla sono stati i ragazzi del Team CHARLI Virginia Polytechnic Institute & State University (U.S.A.) per la categoria Adult Size, quelli del NimbRo Teen Size dell'Università di Bonn (Germania) per la Teen Size e quelli del Team DARwIn delle University of Pennsylvania & VirginiaTech (U.S.A.) per quanto riguarda la Kid Size.
Le gare riguardanti le competizioni con i simulatori 2D e 3D sono state vinte rispettivamente dal team Place HELIOS (Giappone) e dal TT-UT Austin Villa, University of Texas, Austin (U.S.A.); quanto al campionato Small Size Robot, è stato vinto dalla squadra Skuba della Kasetsart University (Tailandia) mentre il primo posto del Middle Size Robot è stato guadagnato dai ragazzi del Tech della United Eindhoven University of Technology (Olanda). Ad aggiudicarsi il primo posto nella Standard Platform League è stata invece la squadra UT Austin Villa del Department of Computer Science, The University of Texas at Austin (U.S.A.).
Ma RoboCup 2012 Mexico non è stata solo una competizione tra robot calciatori; altre due categorie hanno animato l'evento: Rescue e RoboCup Junior.
La prima ha avuto come obiettivo il salvataggio di una "vittima" rappresentata da una lattina: il robot viene programmato per seguire una linea nera rilevata da un sensore ottico e superare ostacoli come discontinuità, rampe e il passaggio da una stanza all'altra, prima di raggiungere la vittima, prelevarla e tornare indietro per portarla in salvo.
Anche per la RoboCup Rescue sono previsti campionati separati per simulatori e per robot reali: per la prima (Rescue Simulation Lea-



Gli ITALIANI in gara

Alla fase finale della competizione mondiale di Città del Messico il nostro paese ha schierato 10 squadre che hanno gareggiato in varie competizioni e categorie:

- SPQR Levis (Soccer Junior Light Weight A) dell'ITIS G. Galilei e dell'IISS J. Von Neumann di Roma;
- Bionics Soccer (Soccer Junior Open League A) dell'IT Archimede di Catania;
- SPQR Gravis (Soccer Junior Open League B) dell' ITIS G. Galilei e dell'IISS J. Von Neumann di Roma;
- CARDANO ROBOTIX (Soccer Junior Open League B) dell'I.I.S. Piazza della Resistenza di Monterotondo;
- Manetti Globetrotters (Soccer Junior Open League B) dell'ISIT Manetti di Grosseto;
- SOSBEARZI (Primary Rescue A) dell'Istituto Salesiano G.Bearzi di Udine;
- Endless Runner (Secondary Rescue A) dell'ITIS G.Chilesotti;
- Jujitsurossi (Rescue B) dell'ITIS A. Rossi di Vicenza;
- GruzanteDance (Dance CoSpace Theatre Demo) dell'ITIS Carlo Zuccante di Venezia;
- Itimonaco1 (Secondary Dance) dell'ITI A. Monaco di Cosenza.

gue) nella categoria Virtual Robot Competition il primo posto se lo sono aggiudicato i ragazzi del PoAReT Politecnico di Milano, mentre nella AGENT COMPETITION al primo posto si sono piazzati i giovani del Ry - One Shiga Japan (Giappone). Nel secondo campionato (Rescue Robot League) il primo posto è stato conquistato dal team iraniano MRL.

La terza competizione della finale di Città del Messico è stata Robocup Junior, articolata in prove di Soccer, Rescue e Theatre o Dance; in quest'ultima i robot devono eseguire prove di danza e movimento a ritmo di musica.

I team italiani hanno guadagnato qualche riconoscimento proprio nella Robocup Junior: sul podio sono salite le squadre SPQR Gravis e Levis, rispettivamente dell'IIS Von Neumann e dell'ITIS Galilei di Roma; per l'esattezza, la SPQR Gravis ha conquistato il 2° posto individuale nella categoria Open Soccer B e 3° del superteam nella categoria Open Soccer B, mentre la squadra SPQR Levis ha conquistato il 3° posto superteam nella categoria Open Soccer B.

Cardano Robotix dell'IIS Monterotondo si è classificata quindicesima su trenta; invece il team di Catania, Bionics Soccer della scuola Itis Archimede, che ha gareggiato nella categoria Open League A non è riuscito ad accedere alla finale.

A parte i risultati degli italiani, questi sono stati i piazzamenti nelle varie gare della Robocup Junior: nella RoboCupJunior Dance, la prima qualificata è stata la squadra Tiger (Cina) mentre al terzo posto si è piazzata l'italiana Gruzante Dance, che ha meritato anche il Best Presentation Award.

Per quanto riguarda la RoboCupJunior Soccer, prima classificata della categoria Soccer A Primary Light Weight è stata la giapponese Gorogoro; per la Soccer A Secondary Light Weight, a classificarsi al primo posto è stata la squadra della Shenzhen Experimental School (Cina). Un altro team cinese si è aggiudicato il primo posto nella categoria Open Soccer A: si tratta di quello di Automation Institute. La Open Soccer B, dove la nostra SPQR Levis si è aggiudicata il 3° posto, è stata vinta dalla squadra EMM CS di Macao.

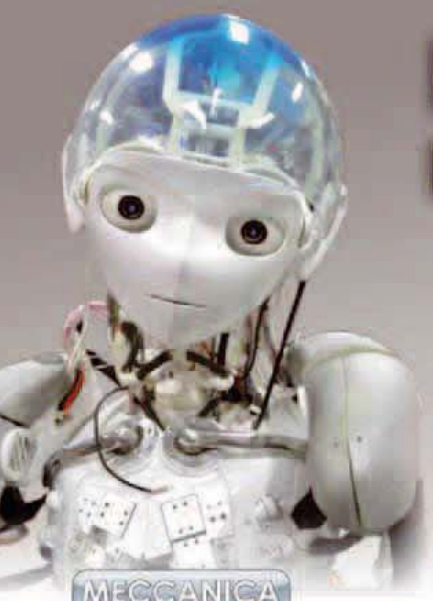
Questo è quanto per le prove di Soccer; per quelle di Rescue, la RoboCupJunior Rescue A Primary è stata vinta dal team giapponese Amalgam, mentre la RoboCupJunior Rescue A Secondary se l'è aggiudicata la cinese Dalian No.8 Senior Middle Schools. Prima classificata nella RoboCupJunior Rescue B è stata la giapponese M&Y, che in team con M&Y+Teamohnename.de ha vinto la SuperTeam Competition, prova in cui le squadre si misuravano a coppie lavorando in team.

Sempre in tema di Rescue, RoboCupJunior CoSpace Rescue è stata vinta dal Team Takosu (Giappone) mentre per quanto riguarda la categoria Dance (Theatre) il primo posto è stato conquistato dalla squadra cinese Tiger.

Come nella passata edizione si è tenuta anche una competizione sponsorizzata dalla FESTO e chiamata Logistics League by FESTO, nella quale ogni squadra partecipante doveva programmare tre robot mobili utilizzati come sistemi automatizzati di trasporto senza conducente, i quali dovevano completare attività impegnative, con (in alcuni casi) sistemi di produzione sconosciuti, e misurare la loro efficacia all'interno di una zona di gioco di 6x6.

L'obiettivo di questa gara è quello di programmare il robot in modo da poter svolgere autonomamente le attività interdisciplinari nei settori della meccatronica, tecnologie dell'informazione e della logistica nelle operazioni di produzione e di comunicare con le macchine con l'intento di produrre il maggior numero possibile di prodotti. A differenza delle altre categorie e competizioni della RoboCup, in questo caso le squadre usano un robot fornito dalla FESTO, chiamato Robotino, progettato dagli esperti della Festo e dotato di intelligenza artificiale, di telecamera e di tecnologia a basso consumo energetico. Questa gara è stata vinta dalla squadra tedesca TUM BendingUnits.

Maggiori informazioni sulla finale di Città del Messico ed i risultati completi sono disponibili sul sito *www.robocup2012.org.*



Entra anche tu nel mondo dei Robot

Sensori, componenti meccanici, sistemi di azionamento e controllo di ultima generazione per realizzare tutti i tuoi progetti, anche quelli più complessi.

SENSORI

posizione

Giroscopio a due assi pitch e yaw



Misura la velocità angolare sugli assi x e y; 2 uscite indipendenti in tensione per ogni asse; dimensioni 12,7x26mm.

LPY510AL - € 26,00

Modulo accelerometro single-chip



Sensibilità ±2g, ±4g, ±8g; dati su bus SPI e I²C-Bus; chip MMA7455L; dimensioni 10x18x3,6mm.

M-ACC - € 13,50

Modulo inerziale



Dispone di giroscopio ITG-3200 e di accelerometro ADXL345 (3 assi).Bus I²C; dimensioni 16,6x15,6mm.

IMUDG8 - € 77,00

infrarossi

Sensore IR 4 ÷ 30cm



Uscita analogica; distanza di rilevazione 4 ÷ 30cm; alimentazione 4,5 ÷ 5,5Vdc; dimensioni 44,5x13,5x19mm.

GP2D120 - € 18,00

Sensore IR 0,5 ÷ 5cm



Uscita digitale; distanza di rilevazione 0,5 ÷ 5cm; alimentazione 2,7 ÷ 6,2Vdc; dimensioni 21,6x8,9x10,4mm.

GP2Y0D805PCB - € 8,00

Sensore IR - 38kHz



Frequenza di lavoro 38kHz/600µs; amplificatore/squadratore integrato; alimentazione 4,5 ÷ 5,5Vdc/0,6mA.

IR38DM - € 2,80

ultrasuoni

Metro ad ultrasuoni 30cm ÷ 5m



Protocollo di comunicazione RS485; alimentazione 8 ÷ 24Vdc/10mA; dimensioni Ø16x38x38mm.

SRF485 - € 32,00

... 15cm ÷ 6m



Protocollo di comunicazione I²C; interfaccia seriale TTL; alimentazione 5Vdc/4mA; dimensioni 24x20x17mm.

SRF02 - € 19,00

... 1cm ÷ 4m



Uscita PWM; frequenza di lavoro trasduttori 40kHz; alimentazione 5Vdc/4mA; dimensioni 43x20x17mm.

SRF05 - € 26,00

MECCANICA

Piattaforma robotica 2WD Arduino compatibile



Corpo in alluminio (Ø170mm) completo di 2 motoriduttori (3 ÷ 6Vdc) con ruote Ø 65mm, ballcaster in metallo, deviatore e presa di alimentazione.

2WDPMA - € 54,00

Piattaforma robotica 4WD Arduino compatibile



Corpo in alluminio (200x170x105mm) completo di 4 motoriduttori (3 ÷ 12Vdc) con ruote Ø65mm, deviatore e presa di alimentazione.

4WDPMA - € 69,90

Pinza in metallo per robot



Dispone di piastra per il montaggio di un mini servo (es. SERVO207, non incluso). Dimensioni (pinza chiusa) 108,3x67x24mm. Apertura massima 50mm.

PINZAROBOT3 - € 14,50

MOTORI

stepper

Passo-passo 15,4Vdc - 200 passi



2 fasi/0,28A; angolo singolo passo 1,8°; coppia di tenuta 2,4kg/cm; dimensioni 42,3 x 42,3 x 37mm.

STEPMOT01 - € 26,00

Passo-passo NEMA23 - 200 passi



2 fasi/2,5A; angolo singolo-passo 1,8°; forza di torsione 9,8kg/cm; albero Ø6,35mm; dimensioni 56,4x56,4x54mm.

STEPMOT02 - € 44,00

SERVOMOTORI



Servocontrolli completi di accessori, ideali in tutte le applicazioni dove è richiesta una dimensione compatta ed una coppia molto elevata. Questi motori sono in grado di mantenere, in modo estremamente preciso, la posizione impostata.

Codice	Alim.	Coppia di torsione	Velocità	Dimensioni (mm)	Peso	Prezzo €
SERVO206	4,8÷6 Vdc	1,5 kg/cm	0,10 s/60°	23x12,5x30	9g	7,50
SERVO207	4,8÷6 Vdc	2,3 kg/cm	0,18 s/60°	28,5x13,3x30,6	17g	8,00
SERVO360	4,8÷6 Vdc	2,8 kg/cm	0,13 s/60°	39,6x20,1x35,6	42g	21,00
SERVO209	4,8÷6 Vdc	3,5 kg/cm	0,18 s/60°	41x20,1x38	38g	8,00
SERVOS03T	4,8÷6 Vdc	7,20 kg/cm	0,33 s/60°	39,5x20x39,6	46g	18,50
SERVO213	5÷7,2 Vdc	13 kg/cm	0,16 s/60°	41x20,1x38	55g	19,50
SERVOS04BBM	4,8÷6 Vdc	13 kg/cm	0,20 s/60°	54,4x26,5x51,5	114g	28,00

ATTUATORI

Attuatore lineare 12Vdc



Corpo in metallo (IP63); motoriduttore 12Vdc; pistone Ø20mm; escursione massima 5cm; coefficiente di carico dinamico 50kg; velocità massima 1,3cm/s.

LACT2 - € 153,00

Modello con escursione massima pistone di 10cm - LACT4 - € 153,00

SUPPORTI

Supporto per attuatore



Corpo in alluminio. Compatibile con attuatori lineari LACT2 e LACT4.

SUPPLACT - € 22,00

Staffa Pan/Tilt per servo 9 grammi



Compatibile con servo cod. SERVO206 (o con dimensioni max. 23x12,5x30mm.) Servo non compresi.

STAFFASERVO9G - € 14,00

RADIOCOMANDO SU RETE Wi-Fi





Utilizza il tuo iPhone o iPad per pilotare da remoto qualsiasi modello sfruttando la telecamera montata a bordo.



WIRC
€ 132,00

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

Prezzi IVA inclusa.

Caratteristiche tecniche e vendita on-line su: www.futurashop.it

Pancho: Il Robot reduce dalla RoboCupJr Italia 2012

a cura degli allievi dell'ITTS Vito Volterra di San Donà di Piave (VE)

L'esperienza di una squadra che ha partecipato alla finale nazionale nella sezione Rescue.



Pancho, questo è il nome del robot, è stato realizzato dagli studenti del 4° anno dell'istituto Vito Volterra di San Donà di Piave per partecipare alla gara nazionale di robotica "RoboCup Jr Italia 2012".
Delle tre competizioni previste alla RobuCup, robot Pancho, è stato sviluppato per partecipare alla gara di rescue. In questo tipo di competizione, i robot vengono progettati e programmati al fine di seguire una linea, aggirare degli ostacoli, salire una rampa e, infine, trovare e "salvare" una lattina.
La premessa era quindi di scegliere una meccanica, una piattaforma di sviluppo e dei sensori più adatti per affrontare tale gara.

STRUTTURA, MECCANICA E COMPONENTISTICA

Per quanto riguarda la struttura meccanica si è optato per una piattaforma robotica in alluminio distribuita da Futura Elettronica con codice 7300-2WDPMA (**Fig. 1**). Questa struttura è costituita da due fogli circolari, spessi circa 2 mm e da una serie di paratie già forate, che hanno facilitato l'inserimento dei motori e dei sensori. In dotazione a questa struttura, ci sono anche due motoriduttori in grado di sviluppare una velocità massima di 70 cm/s, due ruote Tamiya racing di diametro 65 mm ed un portabatteria. La struttura, originariamente circolare, è stata modificata nella parte anteriore per renderla idonea ad accogliere i sensori di linea ed il sensore ad ultrasuoni; questa modifica consente anche una salita più agevole, evitando che la parte anteriore vada a sbattere sulla rampa (**Fig. 2**).
Per afferrare la lattina è stata utiliz-



Fig. 1

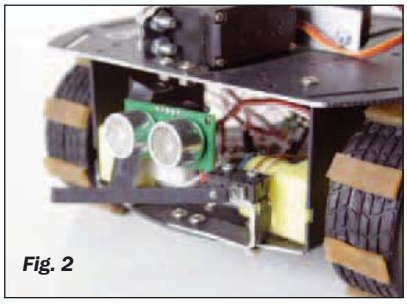


Fig. 2

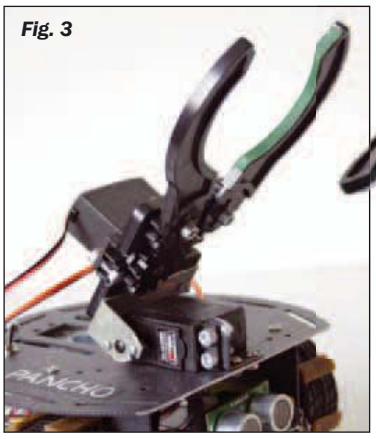


Fig. 3


Fig. 4

zata una pinza azionata da un servomotore e, per migliorare il grip della presa, sono state applicate due strisce in gomma sulle ganasce (**Fig. 3** e **4**). Dopo aver letto il regolamento ed eseguito alcuni test sul campo di gara, sono stati scelti i sensori più idonei. Abbiamo constatato che la linea da seguire era una semplice striscia di nero isolante nero, posta su uno sfondo bianco, perciò abbiamo deciso di utilizzare tre sensori di rifrazione (QTR-1A) che sono composti da un LED IR e un foto-transistor (**Fig. 5** e **6**). Il loro funzionamento è molto semplice: il LED emette gli infrarossi mentre il foto transistor capta gli IR riflessi dalla superficie posta di fronte. Questi sensori dispongono di tre pin di collegamento, due per l'alimentazione mentre il terzo fornisce l'uscita proporzionale alla rifrazione del piano posto di fronte; se il colore del piano è scuro (nero) si avrà una tensione in uscita bassa, se il colore è chiaro (bianco) si avrà un valore di tensione alto. Abbiamo utilizzato anche un interruttore che, a seconda della sua posizione, permetteva a Pancho di avviare la procedura per la taratura dei sensori oppure il programma di gara.

Per la rilevazione dell'ostacolo, posto su un punto della linea da seguire, abbiamo utilizzato un micro-switch a levetta (**Fig. 7**), che chiude il suo contatto quando il robot, avanzando, tocca l'oggetto.

Un ulteriore sensore, usato per rilevare la posizione della lattina, è il sensore ad ultrasuoni SRF05 (**Fig. 8**); quest'ultimo è composto da due trasduttori ad ultrasuoni, Tx (trasmettitore) e Rx (ricevitore), orientati nella stessa direzione. Questo dispositivo fornisce in uscita un impulso la cui durata è proporzionale alla distanza dell'oggetto posto di fronte ed è facilmente gestibile da una scheda a microcontrollore. Durante il funzionamento il sensore invia un impulso ad ultrasuoni a 40 kHz ed attende l'impulso di ritorno dovuto alla riflessione delle onde sonore su di un oggetto; misurando il tempo impiegato fra il momento in cui l'impulso è stato emesso e quello in cui ritorna, è possibile determinare la distanza dell'oggetto, in un range compreso tra 3 cm e 4 metri.

Per sfruttare al meglio questi componenti si è deciso di usare come piattaforma di sviluppo Arduino UNO, che si basa su un microcontrollore della ATMEL. Il microcontrollore dispone di 14 input/output digitali (di cui 6 utilizzabili come PWM), 6 ingressi analogici, una memoria di 32 kB e dispone di una frequenza di clock di 16 MHz, più che sufficienti per la nostra applicazione. Per il comando dei motori ci siamo affidati allo shield Ardu-

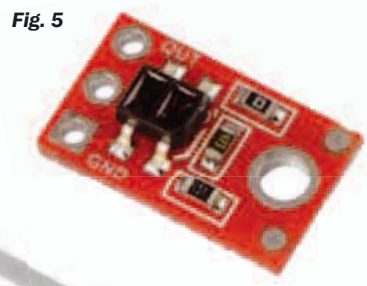
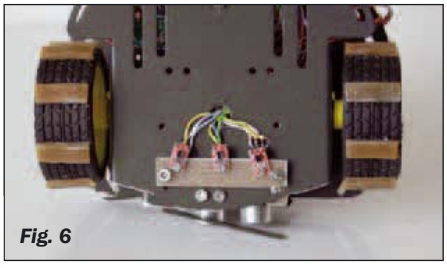


Fig. 5


Fig. 6

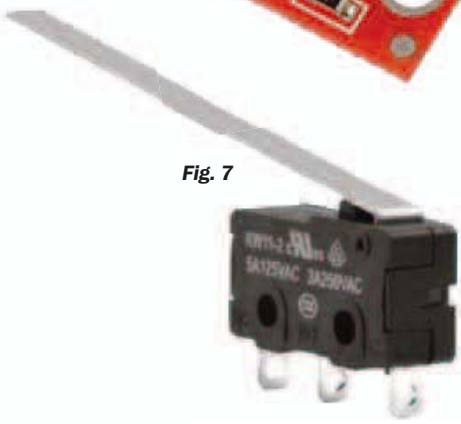

Fig. 7


Fig. 8

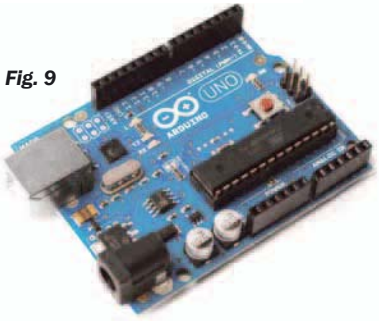


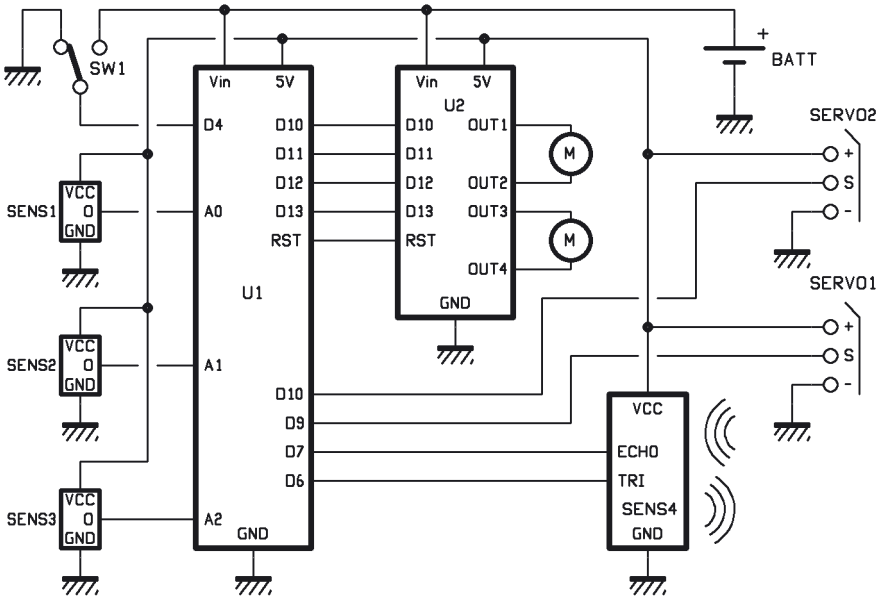
Fig. 9


Fig. 10

Elenco componenti:

U1: ArduinoUNO (ARDUINOUNOR3)
U2: Shield Motori (MOTORSHIELD)
SENS1: QRT-1A
SENS2: QRT-1A
SENS3: QRT-1A
SENS4: SRF05
SW1: Deviatore fine corsa (MSWITCH50MM)

Varie:
- Meccanica completa (2WDPMA)
- Pinza robot (PINZAROBOT1)
- Porta Batteria 6xAA



Motor che permette di comandare entrambi i motori gestendo sia la direzione che la velocità. Ad Arduino è stato connesso anche un ProtoShield necessario per eseguire i cablaggi dei sensori e dei contatti (**Fig. 9** e **10**). Infine, l'alimentazione è stata ricavata utilizzando cinque batterie stilo ricaricabili AA da 1,2 V poste in serie, permettendo di ottenere una tensione nominale di 6 volt, variabile però, a seconda del livello di carica delle batterie. Tuttavia, la diversa velocità dei motori in relazione al livello della tensione della batteria, comportava inevitabilmente delle variazioni rispetto al percorso prestabilito. Questo fatto ha comportato non pochi problemi durante lo sviluppo del software in quanto, per aggirare un ostacolo, è importante che il robot descriva un preciso percorso e non avendo ulteriori riferimenti, come ad esempio un encoder sui motori, le distanze percorse non erano precise. Per la gestione dei motori è stato utilizzato lo shield MotorShield della americana SparkFun, più che idoneo per la gestione di due motori che non assorbono più di 500mA ciascuno.

SCHEMA ELETTRICO

L'alimentazione delle batterie è connessa al plug di Arduino e tramite il pin Vin anche alla scheda di controllo motori. Per tutto il resto si utilizza la tensione stabilizzata prelevata dal pin 5 V di Arduino; pur essendo al limite, come massima corrente prelevabile, non ci sono stati problemi. Ai motori, dedotta la caduta di tensione sul diodo di Arduino e le perdite del driver, giunge una tensione di circa 5v sufficiente ad alimentare i motori, per i quali non è richiesta un'elevata velocità.
Lo shield di controllo utilizza la tensione presente sul pin Vin di Arduino per alimentare i motori e la tensione sul pin 5V per alimentare la logica mentre i pin 3, 11, 12, 13 sono usati per il controllo dei motori. Nello specifico, i pin 3 e 11 sono usati per controllare la velocità di rotazione tramite il segnale PWM generato da Arduino mentre i pin 12 e 13 vengono usati per definire la direzione di marcia. Il sensore ad ultrasuoni utilizza il pin 6 per il segnale di trigger, mentre l'impulso in uscita fa capo al pin 7. I tre sensori di linea sono tutti alimentati a 5 V ed i loro segnali in uscita sono connessi agli ingressi analogici A0, A1, A2. Il micro-switch dispone di un contatto di scambio, il cui terminale comune è stato connesso al pin 4 di Arduino, mentre i due terminali denominati NC e NA sono stati connessi rispettivamente a GND e 5 V. I due servo utilizzati per il movimento della pinza sono alimentati anch'essi a 5 V e sono comandati tramite i pin 8 e 9. Riassumiamo in **Tabella 1** i collegamenti elettrici.

PROGRAMMAZIONE

Dato che il percorso richiedeva diversi comportamenti del robot (seguire la linea, evitare l'ostacolo, salire la rampa, trovare la vittima e salvarla), il nostro programma è sviluppato sulla base dell'istruzione switch-case, che permette di poter gestire più casi a seconda del valore di una variabile. Infatti, dopo il settaggio iniziale e la taratura, viene deciso quale parte di codice eseguire secondo la funzione richiesta. La prima e più importante funzione è l'inseguimento della linea e la parte del programma per questo caso sfrutta i tre sensori di rifrazione.

Tabella 1 - collegamenti elettrici

Pin Arduino	Funzione
A0	Sensore di linea SX
A1	Sensore di linea centrale
A2	Sensore di linea DX
3	PWM motorA
4	Micro-switch sensore oggetti
6	Segnale di trigger di SRF08
7	Segnale di ECHO di SRF08
8	Servo sollevamento pinza
9	Servo chiusura pinza
11	PWM motor B
12	DIR motor A
13	DIR motor B

Per compensare la diversa luminosità e la lucidità del materiale, in cui veniva posto il robot, è stata prevista una procedura automatica di auto taratura. Questa procedura consiste nel far passare i sensori sullo sfondo bianco e poi sulla linea nera, in modo che Arduino possa misurare i valori massimi e minimi di tensione dovuti alla rifrazione, successivamente esegue la media per trovare la soglia bianco/nero. Fatto ciò, mentre Pancho avanzava sul campo di gara, se il sensore sinistro rilevava un valore sotto la soglia, ovvero classificato come nero, la ruota destra accelerava e la sinistra si fermava e viceversa per il sensore destro. Se il sensore centrale rilevava un valore sotto la soglia, significava che Pancho stava seguendo la traiettoria giusta e i motori delle ruote procedevano con la stessa velocità.

A seconda della difficoltà del campo di gara, lungo il percorso poteva essere posizionato un ostacolo delle dimensioni di 10 cm x 10 cm x 20 cm. Quando Pancho incontrava questo ostacolo il micro-switch forniva segnale ad Arduino e venivano attivate le seguenti funzioni:

- retromarcia per allontanarsi dall'ostacolo;
- avanzamento verso destra aggirando l'ostacolo;
- deviazione a sinistra fino al ritrovamento della linea nera.

In questa fase, non disponendo di una velocità costante dei motori o altri riferimenti, non era possibile assicurare la sua infallibilità.

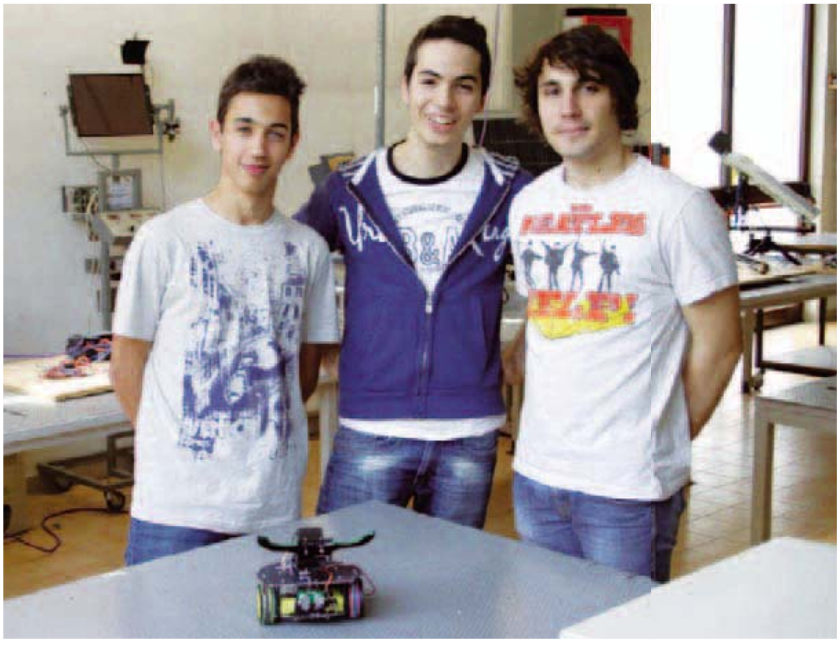
Non pochi problemi si sono riscontrati con la salita della rampa; la modesta potenza dei motori e il limitato attrito delle ruote, spesso impediva la corretta salita, tanto che si è provveduto a dotare le ruote di una ulteriore gommatura.

In cima alla rampa era presente una striscia di materiale altamente riflettente che permetteva di identificare l'entrata nella terza stanza.

A questo punto il robot doveva rilevare la posizione della lattina e per questa funzione abbiamo utilizzato il sensore ad ultrasuoni. Come da regolamento, la lattina poteva essere posizionata come minimo a 5 cm dal bordo della stanza quindi, per il rilevamento di essa è sufficiente far ruotare il robot su se stesso misurando la distanza dal bordo, un brusco cambiamento nella misura significava la presenza della lattina. Il movimento della pinza, tramite i due servi, è stato funzionale ed efficiente e la lattina veniva presa con sicurezza. L'ultima operazione da svolgere era trovare il punto di salvataggio, posizionato in uno dei quattro angoli della stanza.

RISULTATI E OSSERVAZIONI FINALI

Mettere in pratica concetti teorici presenta sempre una serie di difficoltà, imprevisti e complicazioni sono sempre una nuova sfida pronta per essere superata. Questa prima esperienza è stata impegnativa ma ci ha permesso di imparare molte cose nuove, la stessa gara RCJ mette di fronte a problematiche nuove che devono essere risolte con dinamismo e preparazione. Abbiamo capito i punti deboli del nostro robot, sia in termini hardware che software, ma abbiamo già in mente le giuste soluzioni. □



Nella foto gli allievi Cian Marco, Pastò Carlo e Santoro Mattia dell'ITTS Vito Volterra di San Donà di Piave che hanno partecipato, assieme all'allievo Facchin Nicolò, alla RococupJR 2012 Rescue con il robot Pancho.

MINIBUS, UN BUS PER L'AUTOMAZIONE

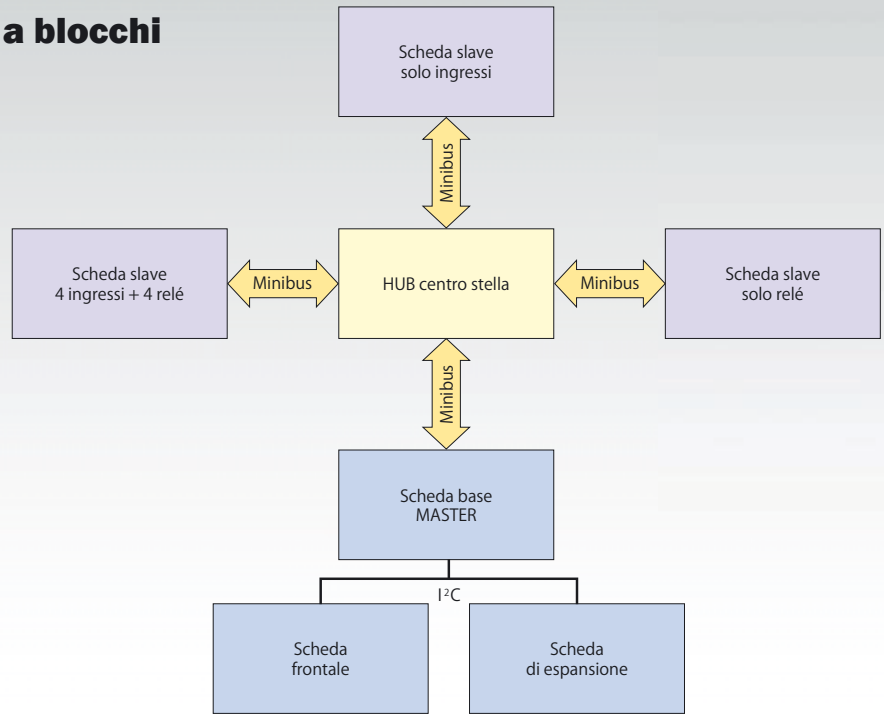


Proseguiamo la presentazione del sistema flessibile di automazione domestica basato sul protocollo I2C potenziato, con la descrizione degli ultimi moduli periferici e del firmware del modulo Master. Terza puntata

di MATTEO DESTRO

Nelle precedenti puntate abbiamo descritto in modo dettagliato l'hardware del modulo MASTER e dei primi moduli SLAVE. In questa puntata completeremo la descrizione dell'hardware della scheda 4 ingressi e 4 uscite relè e della scheda HUB. Introdurremo alcuni concetti base utilizzati per lo sviluppo del firmware memorizzato nel PIC18F2620 montato sul modulo MASTER. Inoltre, il modulo MASTER ha necessità di essere configurato e personalizzato per poter gestire tutti gli ingressi e le uscite ad esso collegati in modo diretto o in modo indiretto attraverso i moduli SLAVE.
Per la configurazione del modulo si deve utilizzare un apposito programma, sviluppato per piattaforma Windows™, il quale permette di personalizzare ogni singola funzione messa a disposizione dallo stesso. Il software in questione si chiama MiniBusTools che provvederemo ad analizzare nella prossima e ultima puntata in ogni singola funzione, in modo da dare al lettore una visione completa delle potenzialità del sistema.
Completiamo ora la descrizione delle ultime schede SLAVE.

Schema a blocchi



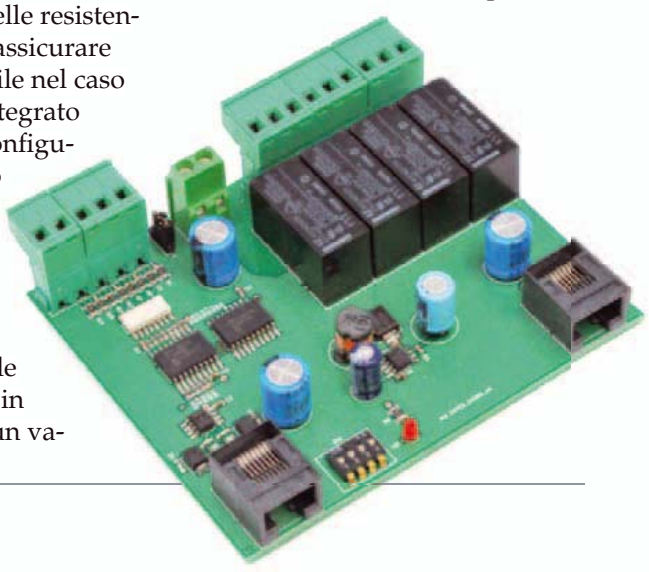
Nello schema è evidenziata la logica di collegamento tra i diversi moduli del sistema. L'unità di controllo è composta dalla scheda base MASTER, dalla scheda frontale e dalla scheda d'espansione collegate tra di loro col protocollo I2C. L'unità di controllo si collega alla scheda SLAVE mediante Minibus 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.

MODULO SLAVE 4 INGRESSI E 4 USCITE RELE'

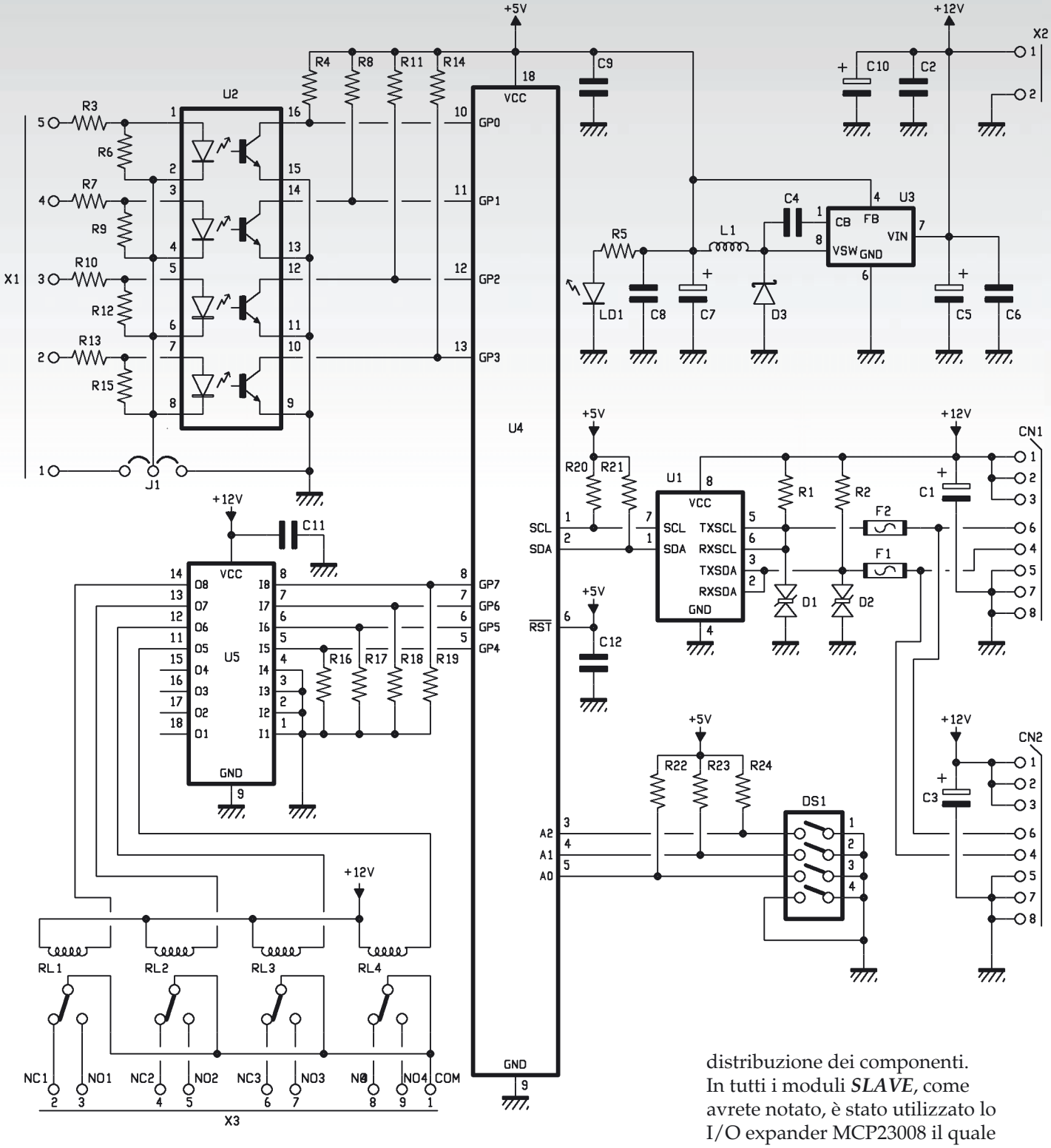
Non rimane che trattare l'ultimo modulo SLAVE il quale è un mix dei precedenti (vedi schema elettrico Modulo 4 IN 4 OUT), ovvero 4 ingressi optoisolati più 4 uscite a relè. Come per i moduli precedenti ritroviamo l'integrato traslatore di livello P82B96 (U1), l'integrato I/O expander MCP23008 (U4) e la sezione di alimentazione realizzata con l'integrato LM2674M5.0 (U3). In questo caso l'I/O expander deve essere impostato per avere metà porta configurata come ingressi (GP0-GP3) e metà porta configurata come uscite (GP4-GP7).
Gli ingressi, come per i precedenti moduli, sono optoisolati attraverso l'integrato TLP2804 (U2) mentre le uscite a relè vengono gestite dall'integrato ULN2803A-DW (U5). Importante notare che gli ingressi da D0 a D3 sono collegati a massa in quanto non utilizzati e quindi NON devono essere lasciati flottanti (integrato U5). Invece per gli ingressi da D4 a D7 sono previste delle resistenze di pull-down per assicurare un valore logico stabile nel caso in cui la porta dell'integrato MCP23008 non sia configurata in modo corretto oppure l'integrato sia danneggiato o non montato.
La sezione di porta utilizzata come ingresso prevede delle resistenze di pull-up in modo da assicurare un valore logico stabile nel caso in cui l'integrato U2 non sia montato o danneggiato. La morsettiera X1 viene utilizzata per la connessione dei possibili ingressi pin da 2 a 5 (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 gli ingressi nel caso in cui si decida di utilizzare l'alimentazione stabile +12Vdc, in questo caso il jumper J1 deve essere posizionato su 2-3. La morsettiera X3 serve per il collegamento dei carichi ai rispettivi relè. Il comune viene collegato al pin 1 della morsettiera.
La serigrafia (vedi Piano di montaggio Modulo 4 In 4 Out) evidenzia sia la distribuzione dei componenti sul modulo, sia un possibile collegamento degli ingressi/uscite al modulo stesso.

MODULO HUB (CENTRO STELLA)

Per ultimo è stato realizzato un modulo HUB che permette di creare un centro stella per il collegamento dei moduli utilizzati nell'impianto. L'elettronica del modulo HUB si limita a connettere tra di loro in parallelo fino a otto connettori RJ45 e a predi-

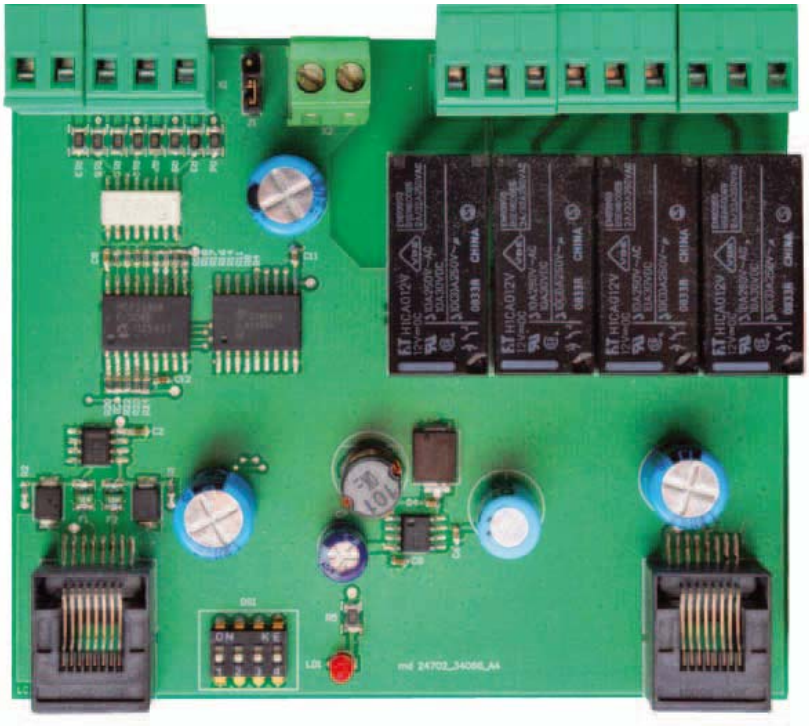


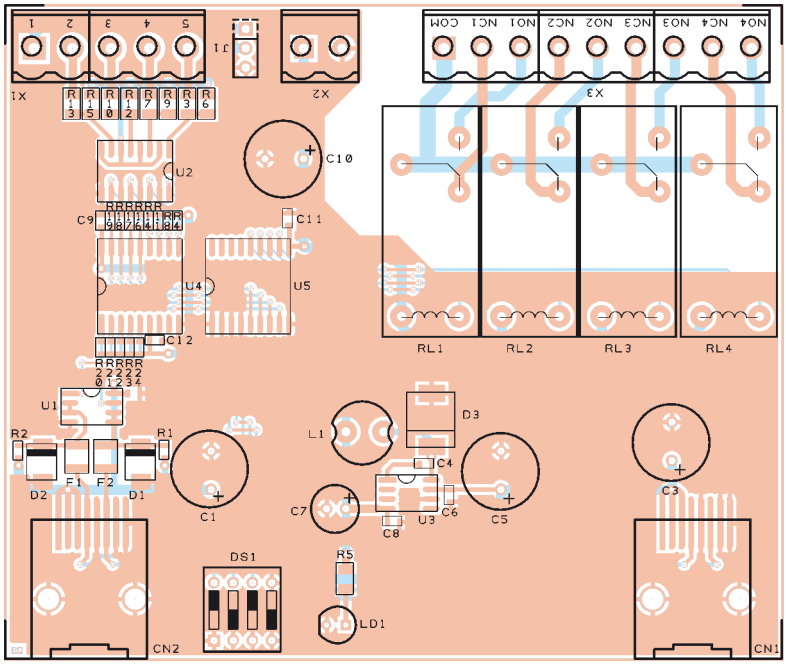
[schema elettrico **MODULO 4 IN 4 OUT**]



sporre l'eventuale inserimento di altrettanti condensatori da 220µF sulla linea dei +12Vdc. Lo schema elettrico evidenzia quanto appena detto (vedi schema elettrico Modulo Hub) mentre la serigrafia (vedi Piano di montaggio Modulo Hub) evidenzia la distribuzione dei componenti.

In tutti i moduli ***SLAVE***, come avrete notato, è stato utilizzato lo I/O expander MCP23008 il quale è il fratello minore dello I/O expander MCP23017 utilizzato nel modulo ***MASTER***. In pratica MCP23008 ha la metà dei registri disponibili in MCP23017 in





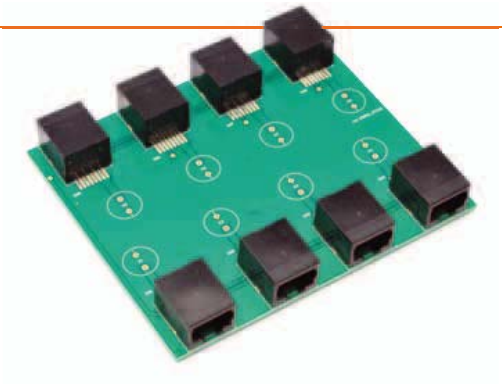
Elenco Componenti:

R1, R2: 2,2 kohm 1% 0603
R3, R7, R10, R13: 4,7 kohm 1% 1206
R4, R8, R11, R14: 4,7 kohm 1% 0603
R5: 3,3 kohm 1% 1206
R6, R9, R12, R15: 1,5 kohm 1% 1206
R16÷R19, R22÷R24:
4,7 kohm 1% 0603
R20, R21: 2,2 kohm 1% 0603
C1: 220 μF 50 VL elettrolitico
C2: 100 nF 50 VL ceramico 0603
C3: 220 μF 50 VL elettrolitico
C4: 10 nF 50 VL ceramico 0603
C5: 100 μF 50 VL elettrolitico
C6: 100 nF 50 VL ceramico 0603
C7: 100 μF 25 VL elettrolitico
C8: 100 nF 50 VL ceramico 0603
C9: 100 nF 50 VL ceramico 0603
C10: 220 μF 50 VL elettrolitico
C11: 100 nF 50 VL ceramico 0603
C12: 100 nF 50 VL ceramico 0603
D1, D2: SMBJ13CA
D3: SS34
LD1: LED 3 mm rosso
U1: P82B96
U2: TLP280_4
U3: LM2674M_5.0
U4: MCP23008
U5: ULN2803ADW
L1: Induttanza 100 μH 800 mA
DS1: Dip-Switch 4 vie
RL1÷RL4: Relé 12V JW1FSN
F1, F2: Termofusibile
autoripristinante 50 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

quanto ha una sola porta di I/O da gestire.
In questo caso i registri sono mappati a partire dall'indirizzo 0x00 fino a 0x0A e non c'è necessità di decidere su quale banco lavorare come per l'integrato MCP23017. Anche in questo caso non descriveremo in dettaglio la configurazione di ogni singolo registro ma soltanto quelli più significativi per la nostra applicazione. Iniziamo dal registro ***IODIR*** il quale configura i pin della porta per lavorare o come ingressi o come uscite. Quindi a seconda del modulo SLAVE in esame avremo:

- Modulo ***SLAVE*** solo ingressi →(***IODIR*** = 0xFF);
- Modulo ***SLAVE*** solo uscite



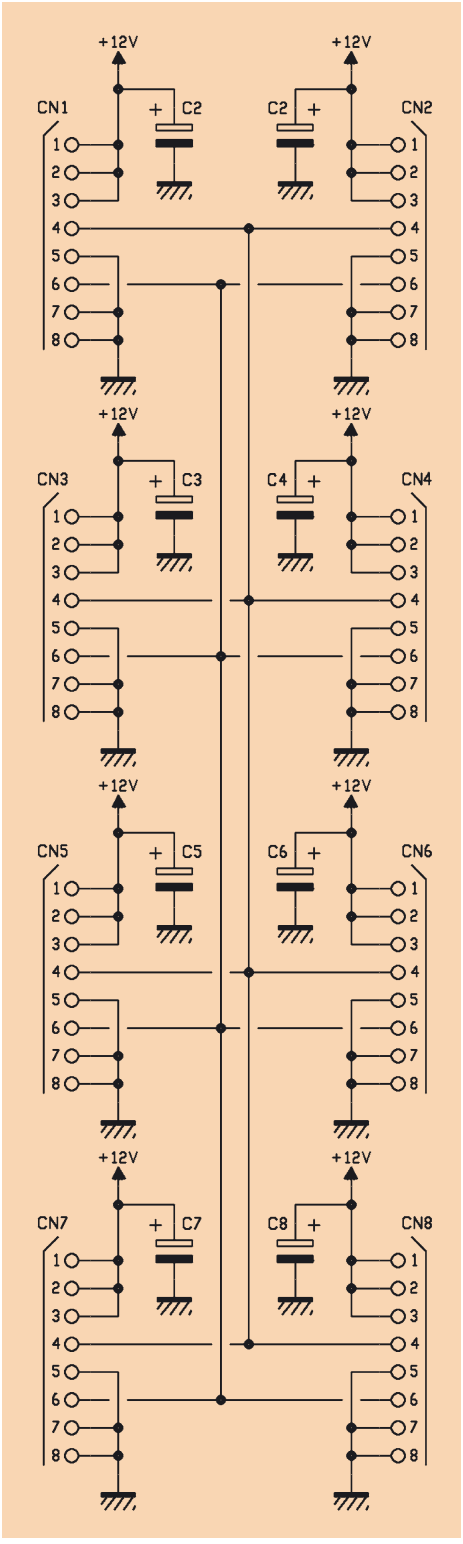
→(***IODIR*** = 0x00);

- Modulo ***SLAVE*** ingressi/uscite →(***IODIR*** = 0x0F).

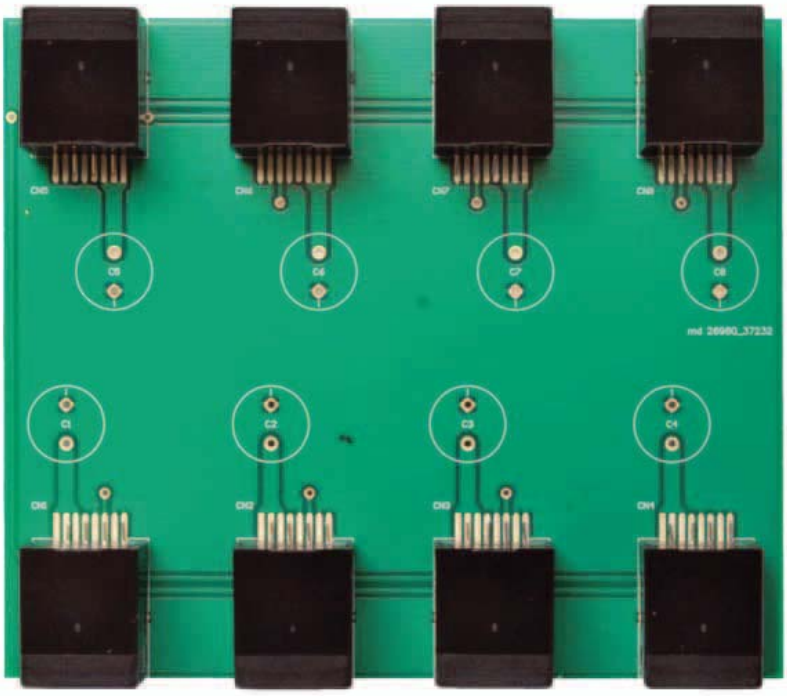
Il registro ***IPOL*** serve per impostare la polarità di lettura del registro ***GPIO***, in altre parole se i bit del registro ***IPOL*** sono settati significa che il valore letto dalla porta ***GPIO*** associata sarà l'inverso del vero valore presente sulla porta, viceversa se i bit di ***IPOL*** 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 ***IPOL*** viene deciso dall'utente attraverso il software di configurazione. Questa configurazione viene applicata ai soli moduli ***SLAVE*** di ingresso o a quelli misti ingresso/uscita.
Il registro ***GPIO*** riflette il valore presente sulla porta quando questa è stata configurata come ingresso, oppure solo una parte di tale porta nel caso in cui il modulo sia di tipo misto (ingresso/uscita).
Infine il registro ***OLAT*** indica il valore che si deve assegnare alla porta configurata come uscita.
Il contenuto del registro dipende dal valore che in quel momento si vuole assegnare all'uscita ovvero se l'uscita debba essere attivata oppure no.
Il valore da assegnare all'uscita dipende da due fattori; il primo è lo stato dell'uscita (attiva o non attiva) il secondo è la configurazione assegnata ad ogni singola uscita, in altre parole la maschera. Per chiarire il concetto si tenga presente che tutte le uscite sono viste, semplicemente, come dei relè e quindi l'utente ha la facoltà di decidere se questi debbano lavorare come contatto ***NC*** (normalmente chiuso) oppure ***NO*** (normalmente aperto). A seconda della scelta fatta dall'utente il software genererà una maschera da applicare al registro ***OLAT***. La maschera viene applicata utilizzando una operazione booleana (XOR) tra il valore da assegnare al registro e la maschera stessa.
Il modulo ***SLAVE***, solo ingressi, è stato progettato per essere inserito nel contenitore modulare da barra DIN 4MH53 della Italtronic (codice 05.0402530) con relativo pannello frontale (codice P05040201P oppure P05040201F. Il primo ha la cornice il secondo no).
I moduli ***SLAVE***, solo uscite e ingressi/uscite, sono stati progettati per essere inseriti nel contenitore modulare da barra DIN 6MH53 della Italtronic (codice 05.0602530) con relativo pannello frontale (codice P05060201P oppure P05060201F).

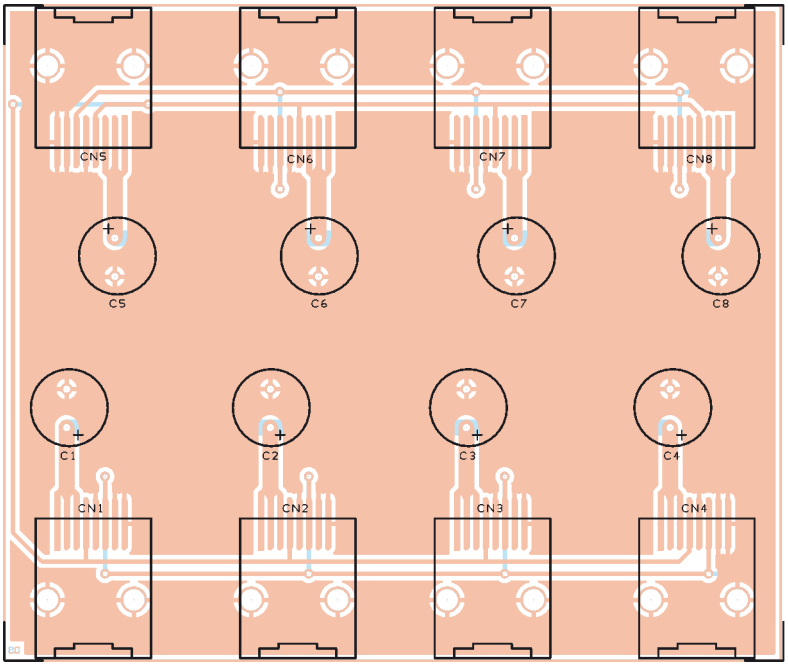
DESCRIZIONE FIRMWARE

Come di consueto facciamo una breve panoramica su come è stato realizzato il firmware di gestione del progetto in esame.
Data la complessità del progetto si è deciso di scrivere il codice direttamente in linguaggio ANSI C, permettendo così di implementare e debuggare codice complesso con molta più facilità rispetto al linguaggio assembler. Inoltre, come con l'assembler, si ha la possibilità di interagire in modo diretto con le periferiche messe a disposizione dal PIC18F2620 così da configurarle e gestirle a seconda dell'esigenza progettuale. In più il linguaggio C ci permette di realizzare e gestire strutture dati complesse e nidificate con molta più semplicità e immediatezza rispetto all'assembler. La **Fig. 5** evidenzia una struttura dati nidificata utilizzata nel progetto per la gestione delle fasce orarie.
Attualmente la quantità di codice utilizzata per l'implementazione



[schema elettrico **MODULO HUB**]





Elenco Componenti:

C1÷C8: 220 μF 25 VL elettrolitico

Varie:
- Connettore RJ45 da CS Molex 95501-6889 (8 pz.)
- Circuito stampato

delle funzioni messe a disposizione ammonta a circa 1/3 di tutta la memoria FLASH disponibile, mentre l'utilizzo della memoria RAM si attesta a circa la metà di quella disponibile (**Fig. 6**). Si conclude quindi che si ha ancora un notevole margine per aggiungere nuove funzioni al sistema.
La **Fig. 4**, invece, mostra come è stato strutturato il progetto ovvero come il codice è stato suddiviso su diversi file in modo da avere una gestione ordinata di tutto l'insieme. Ogni nome assegnato ai file .c richiama in modo inequivocabile le funzioni che in esso sono implementate, inoltre per ogni file .c esiste un equivalente file .h contenente le dichiarazioni delle funzioni implementate ed eventualmente le dichiarazioni delle costanti necessarie all'implementazione del codice. Di seguito, in ordine di importanza, una breve descrizione dei file utilizzati nel codice e il compito che vanno ad assolvere:

- ***main.c*** Contiene il codice necessario a gestire tutte le macchine a stati e a richiamare le funzioni necessarie al corretto funzionamento di tutto il sistema. Dopo la sezione iniziale di inizializzazione del microcontrollore il main entra in un loop infinito dove richiama ciclicamente tutte le funzioni messe a disposizione;
- ***inits.c*** Contiene il codice per inizializzare tutte le periferiche messe a disposizione dal PIC18F2620 compresa l'inizializzazione della memoria RAM;
- ***init_eeprom.c*** Contiene il codice per inizializzare la memoria EEPROM interna al PIC18F2620. Sono i parametri di fabbrica;
- ***init_flash.c*** Contiene una funzione per la lettura della FLASH del PIC18F2620 più una stringa di inizializzazione

che viene salvata in FLASH ad un indirizzo ben preciso;

- ***isr_management.c*** Contiene un'unica funzione la quale viene richiamata ogni qual volta viene generato un interrupt. Quando è in esecuzione richiama le varie sotto-funzioni di gestione dei singoli interrupt, le quali verranno eseguite se, e soltanto se, l'interrupt scatenante le riguarda, altrimenti verranno ignorate;
- ***isr_timer.c*** Contiene le funzioni di gestione degli interrupt inerenti i TIMER messi a disposizione del PIC18F2620;
- ***isr_i2c.c*** Contiene la funzione di gestione dell'interrupt inerente la periferica I²C. Inoltre in questo file sono state memorizzate tutte le funzioni necessarie a gestire le periferiche I²C utilizzate nel progetto,
- ***isr_uart.c*** Contiene la funzione di gestione dell'interrupt inerente la periferica SCI (UART). Inoltre in questo file sono state memorizzate tutte le funzioni necessarie a gestire la comunicazione seriale tra il PC e il PIC18F2620;
- ***debouncing.c*** Contiene la funzione di debouncing degli ingressi e relativa gestione degli stessi. Mette in relazione gli ingressi con le relative uscite, se programmate;
- ***eeprom.c*** Contiene le funzioni di lettura e scrittura della EEPROM interna al PIC18F2620;
- ***led_management.c*** Contiene le funzioni di gestione dei led presenti sul modulo ***MASTER***;
- ***ram.h*** Contiene la dichiarazione di tutte le variabili in RAM;
- ***typedef_generic.h*** Contiene la dichiarazione delle strutture dati generiche utilizzate nel progetto e associate alle variabili allocate in RAM;
- ***typedef_i2c.h*** Contiene la dichiarazione delle strutture dati utilizzate per la gestione del bus I²C e associate alle variabili allocate in RAM;
- ***typedef_uart.h*** Contiene la dichiarazione delle strutture dati utilizzate per la gestione della periferica SCI e associate alle variabili allocate in RAM.

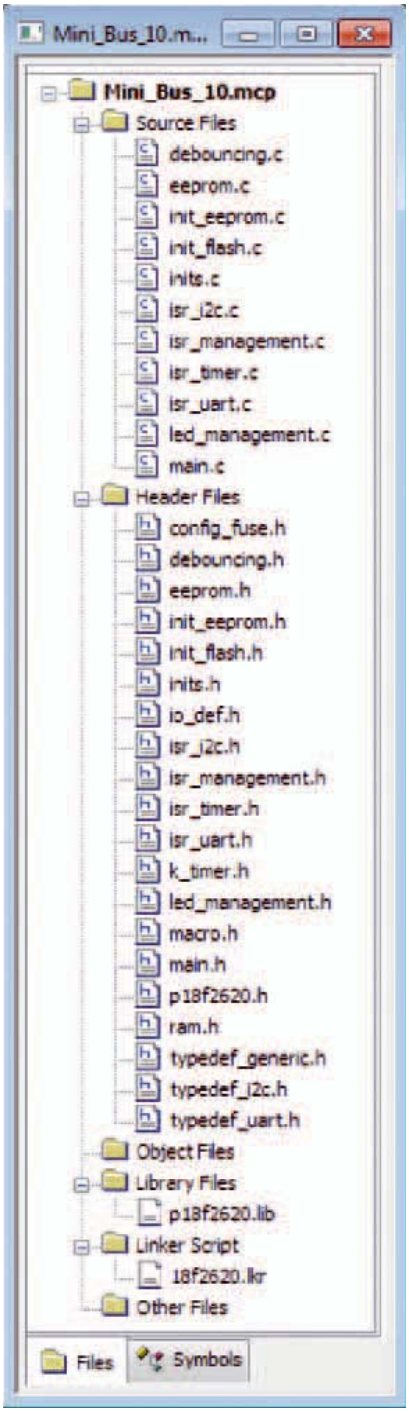


Fig. 4

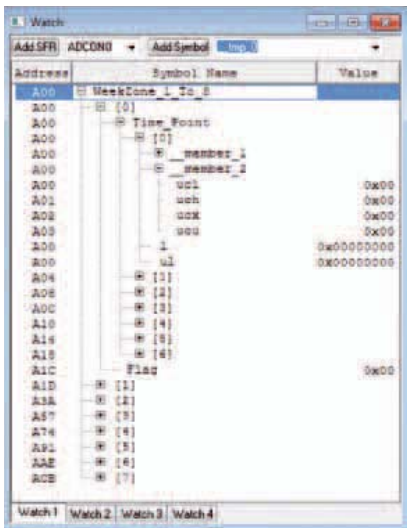


Fig. 5

Data la premessa iniziale passiamo a descrivere brevemente il "nocciolo" del firmware parlando del file main.c e, per aiutarci nella descrizione, facciamo riferimento al flowchart disegnato in **Fig. 7**. Il flowchart è suddiviso in due sezioni distinte; la prima, denominata "***Init***", serve per inizializzare il sistema, ovvero tutte le periferiche a bordo del PIC18F2620 nonché tutte le macchine a stati necessarie a gestire le funzioni messe a disposizione. La seconda, denominata "***Main***", è la sezione in cui vengono processate tutte le macchine a stati, precedentemente configurate, e richiamate tutte le funzioni di supporto al corretto funzionamento del sistema. Ovviamente ci sono una serie di "*semafori*" che discriminano quali funzioni richiamare e quali macchine a stati processare. I "*semafori*" altro non sono che dei semplici flag i quali vengono settati o resettati a seconda dei casi. Ad esempio, se la memoria EEPROM esterna non è programmata significa che non è stato inserito nessun programma utente valido e che quindi alcune funzioni e rispettive macchine a stati non devono essere processate in quanto i relativi dati contenuti in EEPROM non sono coerenti.
Inoltre il test dell'integrità della memoria EEPROM, esterna e

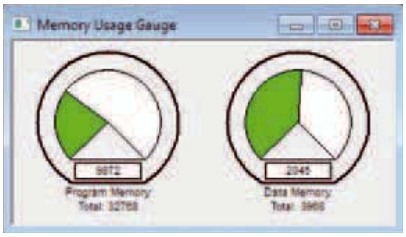
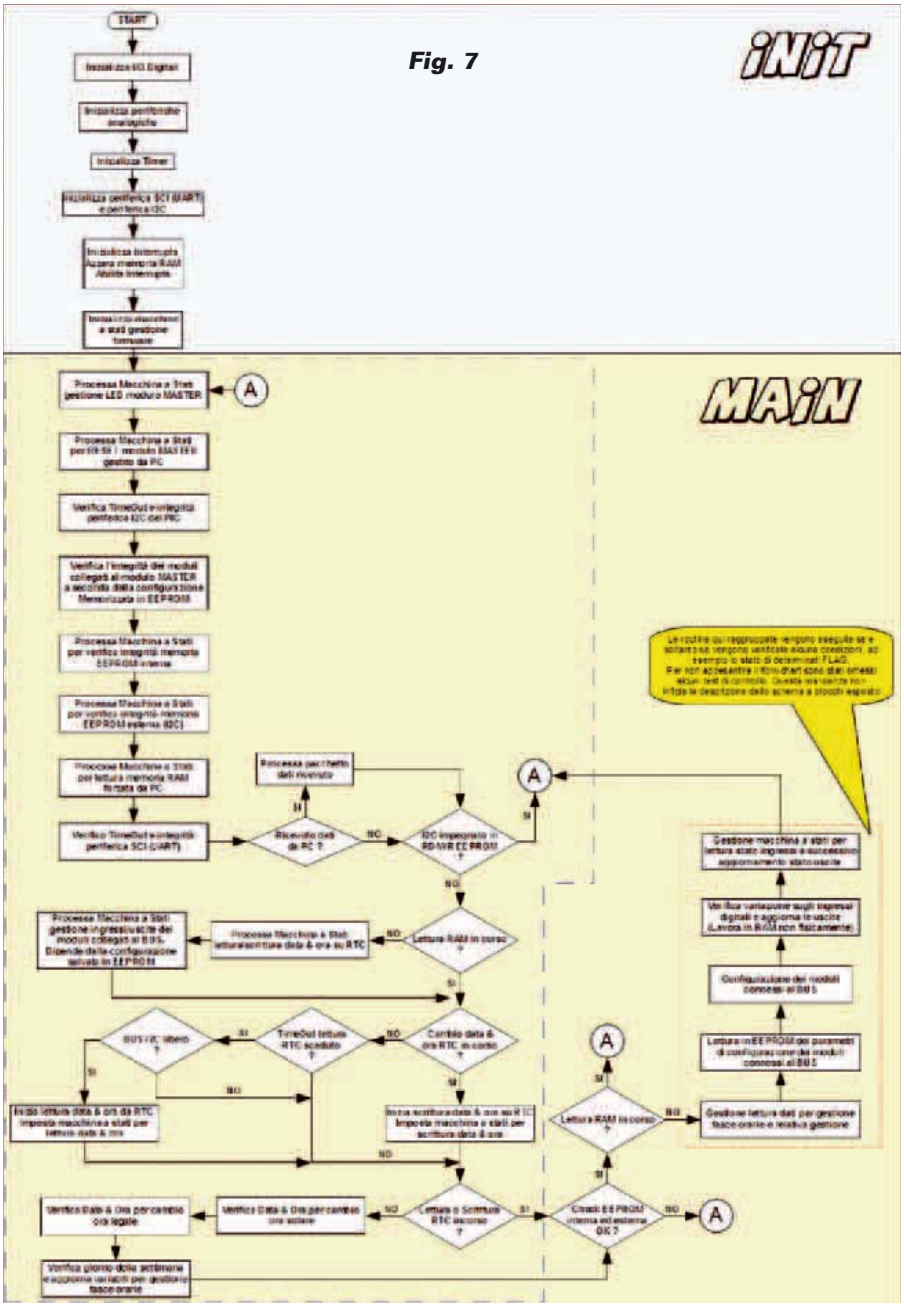
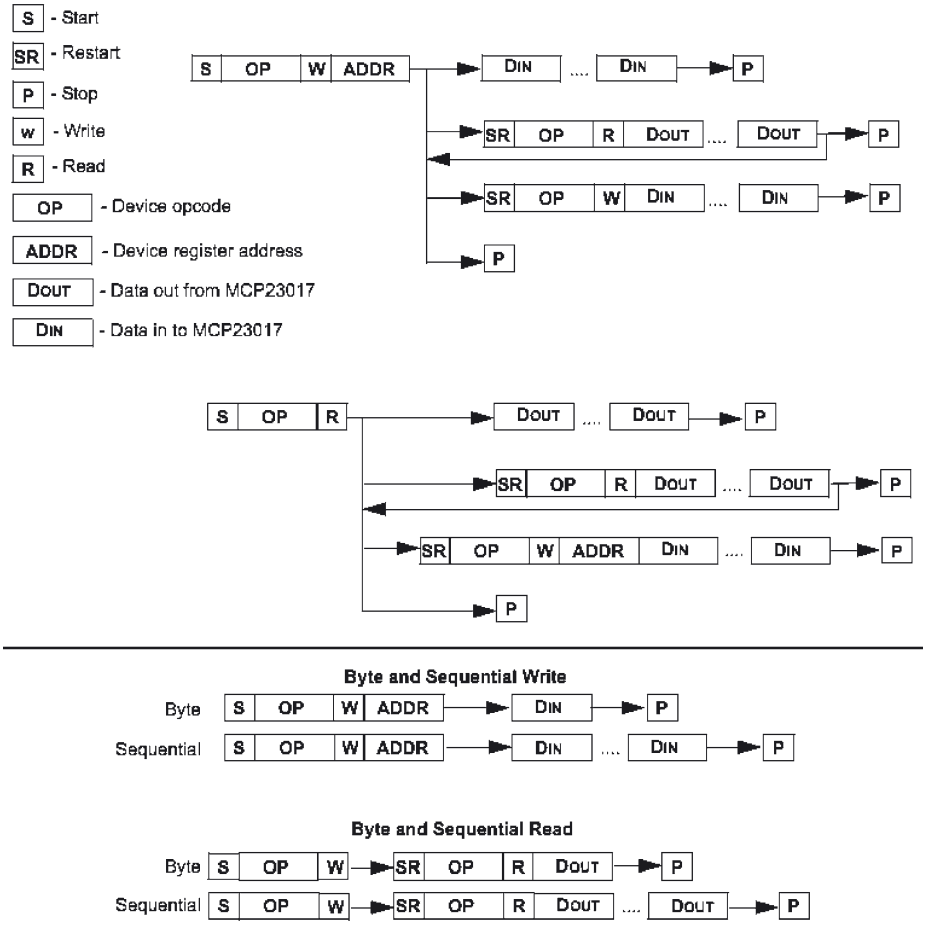


Fig. 6

interna, è necessaria al fine di decidere se è possibile processare i dati in esse contenuti e quindi configurare i moduli ***SLAVE*** collegati al BUS, gestire la dipendenza delle uscite a seconda dello stato degli ingressi e per ultimo, ma non meno importante, gestire tutte le fasce orarie messe a disposizione dal sistema (la configurazione salvata in EEPROM riguarda anche il modulo ***MASTER***).
Le informazioni memorizzate in EEPROM vengono programmate utilizzando il software su PC il quale, sfruttando un apposito protocollo di comunicazione a pacchetti, invia i dati al microcontrollore il quale, a sua volta, provvede a programmare sia la EEPROM su bus I²C sia la EEPROM interna. In più, conoscendo le locazioni di memoria di interesse, è possibile fare della diagnostica leggendo lo stato delle variabili in RAM.
Sempre sfruttando il protocollo implementato è possibile leggere il RTC presente sul modulo MASTER e, se necessario, allineare la data e l'ora a quella del PC oppure forzarla al valore desiderato. Il firmware gestisce in modo totalmente automatico l'ora legale e l'ora solare; quindi è in grado di identificare l'ultima domenica di Marzo e portare avanti di un'ora l'ora corrente (si passa dalle ore 2:00 alle ore 3:00. Ora legale) oppure l'ultima domenica di Ottobre e quindi portare indietro di un'ora l'ora corrente (si passa dalle ore 3:00 alle ore 2:00. Ora solare).
Guardando il flowchart, sezione "***Main***" parte tratteggiata in blu, si nota che ci sono una serie di funzioni (e relative macchine a stati) che vengono processate ad ogni ciclo di main (si ricorda che il main è ciclico ovvero viene ripetuto all'infinito). Dal processo di tali funzioni verranno settati/resettati una serie di flags i quali permetteranno o no l'esecuzione delle funzioni descritte nella parte tratteggiata in rosso. Tutte le funzioni raggruppate nella parte rossa riguardano la configurazione dei moduli ***SLAVE*** e ***MASTER*** nonché le funzioni per leggere lo stato degli ingressi e forzare lo stato delle uscite. Inoltre sono gestite le funzioni per la gestione delle fasce orarie.
Molto importante è la parte di firmware che si occupa di gestire la comunicazione su bus I²C. La gestione viene fatta sfruttando

Fig. 7





l'interrupt di periferica generato dalla periferica hardware messa a disposizione dal PIC18F2620.
La gestione tramite interrupt permette di scrivere un codice molto più "pulito" e comprensibile rispetto ad una implementazione a polling. Il firmware deve gestire i seguenti integrati connessi al bus I²C:

- DS1307 (RTC = Real Time Clock);
- 24FC1025 (EEPROM 1Mbit);
- MCP23017 (I/O Expander. Due porte da 8 I/O ciascuna);
- MCP23008 (I/O Expander. Una sola porta da 8 I/O).

La periferica I²C montata sul PIC18F2620 viene fatta lavorare in ***MASTER MODE***.
Per comunicare con l'integrato desiderato il firmware deve prima verificare che il bus I²C sia libero e che quindi nessuna comunicazione sia in corso, dopodiché può impegnare il bus con una operazione di lettura o scrittura verso l'integrato desiderato. Quando una comunicazione è in corso, è possibile identificare con quale periferica si sta dialogando, semplicemente verificando lo stato di alcuni flag (un flag per ognuna di esse).
Inoltre per ogni periferica è stato allocato un buffer in memoria per gestire i dati da inviare o ricevere, i buffer hanno lunghezza differente gli uni dagli altri in quanto i dati da gestire per ogni periferica sono diversi sia come quantità che come tipologia.
Anche la velocità di comunicazione tra i vari integrati è differente ad esempio: per quanto riguarda la memoria 24FC1025 la linea ***SCL*** (clock) viene fatta lavorare a 400Khz, mentre per quanto riguarda il clock calendar DS1307 la velocità di lavoro della linea ***SCL*** è 100Khz in quanto non accetta velocità superiori.
Diversamente la periferica MCP23017 viene fatta lavorare a 400Khz in quanto è direttamente montata sul modulo ***MASTER*** mentre tutti gli integrati MCP23008 verranno fatti lavorare a 100Khz in quanto montati sulle schede dei moduli ***SLAVE***, i quali possono essere posizionati a diversi metri di distanza dal modulo ***MASTER***. Di conseguenza, il registro ***SSPADD***, necessario a impostare la velocità della linea ***SCL***, verrà continuamente settato a seconda della periferica utilizzata.
Ogni comunicazione verso una qualsiasi delle periferiche inizierà sempre con la generazione, da parte del PIC18F2620, dello "***Start Bit***", che va ad impegnare il bus I²C e verrà conclusa con la generazione di uno "***Stop Bit***". Tra questi due marcatori verranno inseriti i dati, suddivisi in byte, da inviare ai diversi integrati e per ogni byte inviato il PIC18F2620 si aspetta un segnale di acknowledge (ACK) da parte della periferica. Tutte le volte che si conclude la generazione di uno start o stop bit o dopo l'invio di un byte di dati viene generato un interrupt ad indicare che la periferica è libera per passare allo step successivo. Quindi nella routine di gestione dell'interrupt ci sarà una struttura che discriminerà in quale step di gestione della comunicazione ci si trova

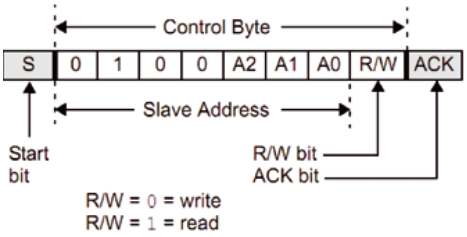


Fig. 8

Fig. 9

e agirà di conseguenza. Inoltre dovrà distinguere verso quale periferica si stanno inviando o ricevendo i dati in quanto la gestione cambia a seconda dei casi.
Per fare un esempio guardiamo la **Fig. 8**, estrapolata dal datasheet, la quale evidenzia una possibile comunicazione verso la periferica MCP23017. Si noti che la comunicazione inizia sempre con uno "Start Bit" per poi concludersi sempre con uno "Stop Bit". Il primo byte che si deve obbligatoriamente sempre inviare dopo uno "Start Bit" è il cosiddetto "Control Byte", vedere **Fig. 9**, il quale deve contenere il codice univoco di identificazione della periferica, in questo caso "0100", l'indirizzo fisico della periferica con cui comunicare (per MCP23017 è sempre "000") e per finire un bit che identifica se si sta eseguendo una scrittura o una lettura sulla periferica.
Nella **Fig. 8** non viene erroneamente evidenziato che, sia per la scrittura/lettura di un singolo byte che per la scrittura/lettura sequenziale di byte, è necessario indicare da quale indirizzo di partenza si vuole iniziare a scrivere o leggere. Questo concetto vale per tutte le periferiche I²C.
Quindi ricapitolando, il firmware, per gestire una comunicazione I²C verso una delle possibili periferiche, deve:

1) Preparare i dati da inviare alla periferica. Per lo scopo si deve utilizzare l'apposito buffer in memoria;
2) Verificare che il bus I²C sia libero, ovvero nessuna comunicazione in corso verso altre periferiche;
3) Generare lo "***Start Bit***" e attendere interrupt prima di passare allo step successivo;
4) Inviare il "***Control Byte***" e attendere l'interrupt prima di passare allo step successivo. Ogni periferica ha il proprio "***Control Byte***" (vedere **Fig. 9**). La periferica interrogata, per ogni byte ricevuto, deve obbligatoriamente generare un segnale di ACK. Se il PIC18F2620 non riceve ACK da parte della periferica dovrà abortire la comunicazione generando lo "***Stop Bit***";
5) Inviare l'indirizzo da cui iniziare a leggere o scrivere e attendere interrupt prima di passare allo step successivo. L'indirizzo potrebbe richiedere uno o due byte a seconda dei casi. Ad esempio per quanto riguarda la memoria EEPROM 24FC1025 sono necessari due byte per l'indirizzamento;
6) Inviare i dati, in caso di scrittura, oppure iniziare la procedura di lettura. In caso di scrittura si deve inviare verso la periferica un byte alla volta, fino a svuotare il rispettivo buffer, e per ogni byte attendere l'interrupt prima di passare al successivo. Come sopra per ogni byte inviato si deve ricevere ACK da parte della periferica. Nel caso di lettura si deve invece:
   a) Generare un "***Re-Start Bit***" e attendere interrupt prima di passare a step successivo;
   b) Inviare nuovamente il "Control Byte" specificando che si sta eseguendo una operazione di lettura. Vedere **Fig. 9**;
   c) Leggere i dati, un byte alla volta, dalla periferica. In questo caso è il PIC18F2620 che genera l'ACK ad indicare che il byte è stato ricevuto. Prima di passare al byte successivo è obbligatorio inviare l'ACK verso la periferica interrogata;
7) Concluso l'invio o ricezione dei dati provvedere a generare lo "Stop Bit".

Questo processo viene re-iterato all'infinito, a turno ognuna delle possibili periferiche (memoria, I/O Expander ecc) verrà interrogata dal sistema mantenendo così allineate le uscite rispetto agli ingressi. Inoltre verrà monitorata la memoria EEPROM per assicurarsi che sia integra e che quindi contenga dei dati congruenti con l'applicazione, e il clock calendar verrà letto con cadenza di un secondo per restare sempre allineati all'ora e data corrente in modo da potere gestire in modo corretto le fasce orarie eventualmente implementate. Nella prossima ed ultima puntata parleremo del software sviluppato per la programmazione delle funzioni messe a disposizione dal sistema 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.

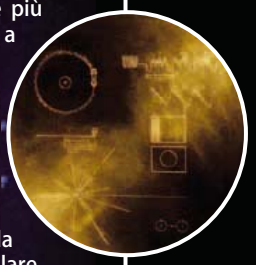
Il materiale va richiesto a:
Futura Elettronica, Via Adige 11,
21013 Gallarate (VA)
Tel: 0331-799775 - Fax: 0331-792287
http://www.futurashop.it

Voyager 1 varca le porte del Sistema Solare



Dopo un viaggio di 35 anni, la sonda della Nasa Voyager 1, lanciata nel 1977, si avvicina ai limiti del nostro Sistema Solare, anche se ancora non sappiamo quale sarà il momento esatto in cui la navicella, lanciata per portare in giro per l'universo la testimonianza dell'esistenza dell'umanità, si addentrerà nell'universo interstellare. Siccome sono passati decenni dal lancio, ormai ce ne siamo quasi scordati, però la missione Voyager 1 è di estrema importanza perché la navicella porta con sé un disco registrato di rame e placcato d'oro che contiene immagini e suoni del nostro pianeta, assieme alle istruzioni su come ascoltarlo e ad un messaggio di saluto in tutte le lingue del mondo; l'intento è portare un messaggio di amicizia e comunicare, ad eventuali forme di vita che incontreranno l'astronave in qualche angolo remoto dell'universo, qualcosa sulla Terra e sull'umanità.

La NASA è certa che Voyager 1 stia uscendo dal Sistema Solare perché i dati ricevuti dalla sonda indicano che essa sta incontrando sempre più frequentemente particelle a grande intensità di energia, le quali provengono certamente dall'esterno del nostro Sistema Solare.



All'inizio di luglio, Voyager 1 si trovava a circa 18 miliardi di km dalla Terra, ma il momento esatto in cui la sonda entrerà nello Spazio interstellare ci verrà comunicato con un certo ritardo, dato che i segnali radio trasmessi dalla sonda e riguardanti le informazioni sulla rotta e sulle particelle incontrate, arrivano sulla Terra con circa 16 ore di ritardo.

Insieme alla compagna Voyager 2 (lanciata un mese dopo), la sonda ha superato Giove, Saturno, Urano e Nettuno (comprese 48 delle loro lune), rivelandosi una delle missioni scientifiche più fruttuose e interessanti dell'esplorazione dello Spazio.

È possibile seguire in diretta il viaggio delle navicelle del programma Voyager sulla pagina web http://voyager.jpl.nasa.gov/where/index.html.

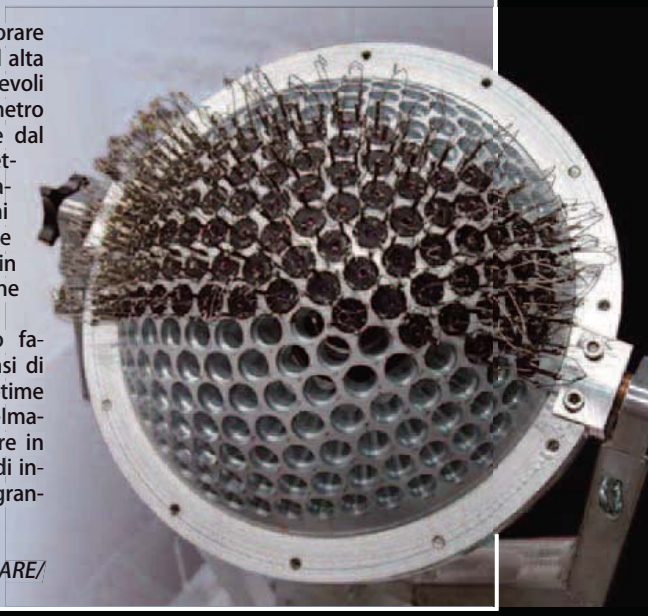
http://voyager.jpl.nasa.gov

1 Gigapixel: dove l'occhio non arriva

Alla Duke University (North Carolina, U.S.A.) hanno realizzato un prototipo di supermacchina fotografica in grado di acquisire un miliardo di pixel, cioè ben oltre la risoluzione dei nostri occhi messi insieme e circa trenta volte più delle migliori fotocamere attualmente in commercio.

Aware-2, così si chiama il prototipo realizzato da un gruppo di scienziati coordinato da David Brady, potrebbe cambiare i sistemi di osservazione e sorveglianza, dato che consentirebbe ingrandimenti notevoli conservando un'elevatissima definizione delle immagini.

Aware-2 può riprendere immagini da mille megapixerl con un refresh di tre fotogrammi al minuto; è un sistema composto da 98 microtelecamere da 14 megapixel ciascuna, montate su un supporto a calotta sferica e puntate verso un punto focale, oltre che da un'unità elettronica progettata per elaborare grandi quantità di informazioni ad alta velocità. Le dimensioni ragguardevoli dell'insieme (75 centimetri di diametro e 50 di profondità) sono dettate dal sistema di raffreddamento dell'elettronica che si occupa dell'elaborazione delle immagini. I fotogrammi ripresi dalle singole telecamere vengono campionati e ricomposti, in quanto ognuno forma una porzione del quadro complessivo.



Aware-2 può essere considerato facente parte dei sistemi statunitensi di tipo Argus-Is (Autonomous Real-time Ground Ubiquitous Surveillance-Imaging System) creati per sorvegliare in maniera continua le aree critiche di interesse e capaci di identificare con grande precisione un bersaglio.

http://disp.duke.edu/projects/AWARE/

Memorizzare un bit in una molecola

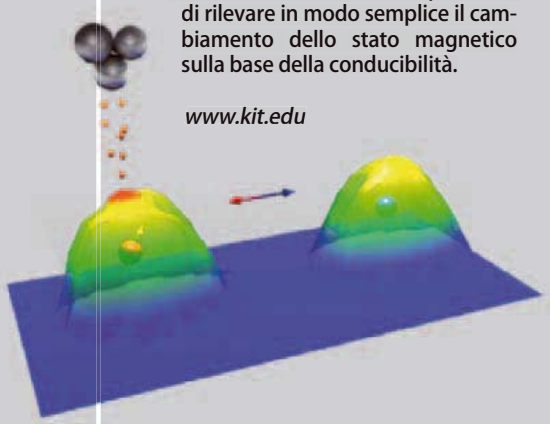
L'attuale limite di densità di memorizzazione degli hard-disk dipende dall'effetto superparamagnetico, che impedisce di memorizzare bit in aree troppo piccole e comparabili con lo spazio di poche molecole. Ma ora una nuova tecnologia potrebbe aumentare enormemente la capacità di storage dei supporti magnetici: usando l'elettricità, i ricercatori del Karlsruhe Institute of Technology sono riusciti a modificare lo stato magnetico e la resistenza elettrica delle molecole. Ciò permetterebbe di memorizzare lo stato di un bit binario in una sola molecola, il che, se consideriamo che negli attuali dischi rigidi un bit richiede circa 3 milioni di atomi di materiale ferromagnetico, è una vera rivoluzione.

Con un impulso elettrico, la molecola metallo-organica può essere fatta commutare stabilmente da uno stato magnetico ad alta conducibilità ad uno non magnetico, a bassa conducibilità.

La tecnica sviluppata al KIT si basa invece su una struttura formata da un singolo atomo ferromagnetico posto al centro di una molecola organica formata da 51 atomi che protegge l'informazione memorizzata nell'atomo. A parte l'elevata densità di storage permessa, questo tipo di memoria (detta "spin crossover molecule") presenta anche il vantaggio di garantire un processo di scrittura puramente elettrico affidabile.

In laboratorio, la polarizzazione elettrica è stata applicata mediante un microscopio a scansione a effetto tunnel, che ha permesso di somministrare alla molecola impulsi di dimensioni nanometriche; questa tecnica cambia non solo lo stato magnetico, ma anche le proprietà elettriche della molecola, il che permette di rilevare in modo semplice il cambiamento dello stato magnetico sulla base della conducibilità.

www.kit.edu





L'Italia ha il suo supercomputer

IBM detiene saldamente il primato del supercomputer più veloce al mondo e ha deciso di donare un po' della propria tecnologia anche al nostro Paese, collaborando alla realizzazione del super cervellone installato al Cineca, il Consorzio interuniversitario italiano per il supercalcolo. Il calcolatore si chiama Blue Gene/Q ma è stato ribattezzato Fermi ed è al settimo posto della Top500 annunciata nel corso della International Supercomputing Conference (ISC12) di Amburgo. Questa classifica, che annovera i computer più potenti al mondo, vede primeggiare Sequoia, il potentissimo elaboratore costruito dall'IBM presso la National Nuclear Security Administration americana: si tratta ancora di un sistema Blue Gene/Q composto da 96 rack ed è dedicato al programma AdvancedSimulation and Computing per la gestione della riserva di armi nucleari statunitense. Sequoia può elaborare alla velocità di 16,32 petaflop (un petaflop equivale a un milione di miliardi di operazioni a virgola mobile al secondo, cioè la capacità di calcolo di circa 35.000 PC). Nella classifica dei supercomputer, a Sequoia seguono il K Computer della Fujitsu, macchina dotata di 705.000 processori Sparc64 e capace di elaborare a 10,51 petaflop ed installata presso il Riken Advanced Institute for Computational Science di Kobe in Giappone, e un altro esemplare di BlueGene/Q, chiamato MIRA, dotato di 786.000 core e capace di una velocità di 8,15 petaflop, in funzione presso l'Argonne National Laboratory dell'Illinois. Il nostro Fermi conta invece 163.000 processori PowerA2 e viene raffreddato con un sistema misto ad acqua ed aria; è tra i migliori per Power Usage Effectiveness, l'indice del livello di efficienza energetica, che vale 1,15.

www.cineca.it

Trovato il Bosone di Higgs



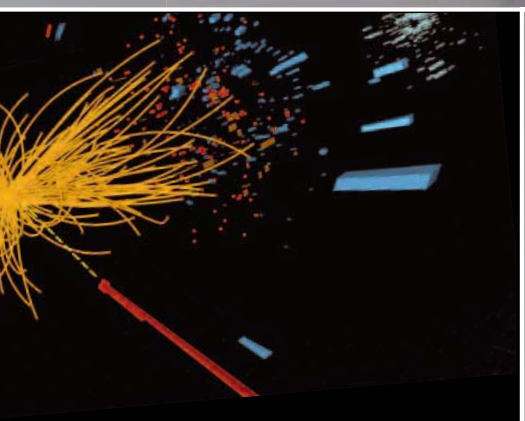
A oltre 50 anni dalla sua teorizzazione, sembra che il tanto bramato Bosone di Higgs stia diventando una certezza; dopo svariati esperimenti, sia al CERN di Ginevra che negli U.S.A., i ricercatori sono praticamente certi di averlo trovato, lui, l'ultimo tassello del puzzle per costruire il quadro delle particelle elementari che costituiscono la materia. Per il momento, tuttavia, gli scienziati europei e statunitensi si dividono sul nome da dare alla loro certezza: quelli U.S.A., dagli esperimenti condotti con il loro acceleratore di particelle Tevatron (in Illinois) sarebbero sicuri a livello 3 sigma, che tradotto significa una certezza del 90%, mentre al CERN l'ottimismo sale, dato che i ricercatori dichiarano di avere oltre il 99,999% di certezza di aver trovato il bosone. Infatti gli esperimenti ATLAS e CMS hanno fatto osservare una nuova particella nella regione di massa attorno a 125-126 GeV, che teoricamente corrisponde al più pesante bosone mai trovato. Un quadro più completo delle osservazioni di luglio uscirà entro la fine dell'anno, dopo che l'LHC fornirà i dati degli altri esperimenti. Il prossimo passo sarà determinare l'esatta natura della particella, ossia se si tratta dell'ingrediente mancante nel modello standard della Fisica delle particelle o proprio del Bosone di Higgs. L'importanza di questa par-



Anche la Cina alla conquista della Luna

Alle 3.05 del mattino di venerdì 29 giugno, dopo una missione di 13 giorni, la capsula cinese Shenzhou 9 è atterrata nella Mongolia centrale, terminando una missione spaziale importantissima per la Cina, sia perché per la prima volta una navicella ha trasportato un equipaggio umano di tre astronauti (tra cui una donna) sia perché è stata la prima ad agganciarsi alla stazione Spaziale Tiangong 1 (cinese anch'essa, messa in orbita lo scorso anno) dove l'equipaggio ha potuto soggiornare. La missione è filata liscia e gli astronauti si sono staccati dalla loro stazione, alla quale avevano attraccato inizialmente con una manovra automatica, poi si erano sganciati ed avevano ripetuto l'attracco comandando la navicella manualmente, compiendo con successo una manovra delicatissima e difficile, finora riuscita solo a Stati Uniti e Russia.
Forte di questo successo e orgogliosa di fronte al mondo che l'osserva, la Cina proseguirà spedita nell'attuare il proprio programma spaziale, che entro il 2020 porterà al completamento della sua stazione spaziale, anche se per mandare nello Spazio le parti mancanti dovrà affidarsi ad un nuovo tipo di vettore (Lunga Marcia 5, capace di portare in orbita fino a 25 tonnellate) perché quello usato finora non è idoneo.

http://news.xinhuanet.com



ticella si comprende sapendo che il Modello Standard descrive le particelle fondamentali di cui siamo fatti, e ogni cosa visibile nell'universo, e le forze che agiscono tra di loro, ma tutta la materia che possiamo vedere sembra essere non più del 4% del totale. La particella di Higgs potrebbe essere un ponte per comprendere il 96% dell'universo che rimane a noi oscuro.

http://public.web.cern.ch

Lenti multifocali da gocce di cristalli liquidi

Per la prima volta i ricercatori dell'Ino-CNR di Napoli hanno realizzato microscopiche gocce di cristalli liquidi con cui si potrebbero mettere a punto microlenti con focale variabile impiegabili nei pannelli fotovoltaici e in fotografia. Ordinati come un cristallo ma fluidi come un liquido, i cristalli liquidi in particolari condizioni si organizzano in fasi intermedie che presentano le caratteristiche dello stato liquido e di quello solido, peculiarità che ne hanno permesso un larghissimo impiego per la costruzione di oggetti tecnologici di uso quotidiano: basti pensare ai display LCD.
Ricercatori dell'Istituto nazionale di ottica del Consiglio nazionale delle ricerche (Ino-Cnr) di Napoli guidati da Pietro Ferraro, in collaborazione con l'Istituto di chimica e tecnologia dei polimeri del Cnr (Ictp-Cnr) e con il Dipartimento di scienze fisiche dell'Università Federico II di Napoli, hanno avviato una ricerca per manovrare i cristalli liquidi in maniera del tutto innovativa.
Per la prima volta i cristalli liquidi sono stati frammentati in goccioline microscopiche che possono assemblarsi nuovamente in gocce più grandi tramite un processo reversibile; in particolare, le goccioline possono diventare micro-lenti con focale variabile in grado di offrire inaspettate soluzioni in fotografia, per mettere contemporaneamente a fuoco oggetti a diverse profondità e, nel fotovoltaico, per catturare la luce solare da qualunque angolazione con una lente di forma sferica. Altre applicazioni si aprono nella sensoristica e nelle nano e biotecnologie.

www.cnr.it



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



NEW

€ 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

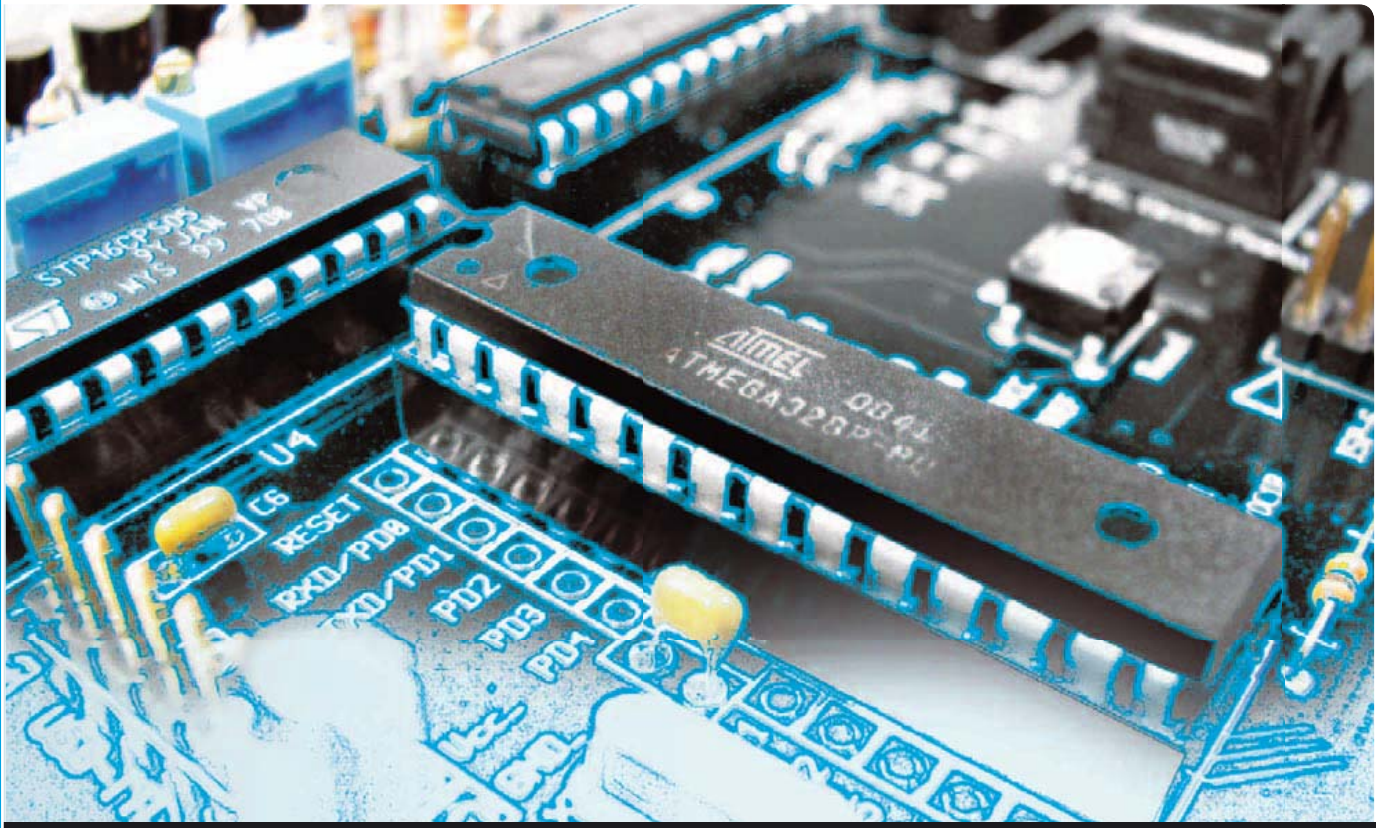
- Memorizzazione dei video (AVI, 30fps) e delle immagini (JPEG) su memoria integrata da 64MB o su micro SD (max. 8 GB, non inclusa)
- Può funzionare anche come Webcam
- Telecamera CMOS 1,3 Mega pixel
- Microfono incorporato
- Dimensioni Ø31 x 101 mm
- Alimentazione tramite batteria ricaricabile al Litio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



ATMEL OPEN SOURCE

dell'ing. OSVALDO SANDOLETTI

Descriviamo il funzionamento di alcune periferiche di input/output presenti nella scheda demoboard: buzzer, LED, tasti e display alfanumerico. Settima puntata.

Nella scorsa puntata ci siamo lasciati con la proposta di realizzare un programma per variare in modo automatico la luminosità del LED LD9 della demoboard, dal minimo al massimo e viceversa, creando un effetto "pulsante". Nel **Listato 1** vi è la soluzione al problema, con la quale potrete confrontare quella scritta da voi, che potrebbe peraltro funzionare bene anche se diversa: ricordate che non c'è quasi mai un modo univoco di concepire e implementare un algoritmo!

In questo programma si utilizzano due *variabili globali*, denominate così perché accessibili e utilizzabili in tutte le funzioni del programma (che nel nostro caso sono la `main`() e la isr (`TIMER1_COMPA_vect`)). Le variabili globali si dichiarano al di fuori delle routine; le nostre le chiameremo '`direzione`' e '`luce`', entrambe da 8 bit. La prima assume solo i valori zero e uno (bastava un bit, ma la variabile più piccola dichiarabile è da 8 bit!) per indicare al programma qual è la *fase* in corso: se zero la luminosità sta crescendo, se uno sta diminuendo; '`luce`' contiene invece il valore di luminosità (da 0 a 255).

Nel *main-loop* non vi sono istruzioni: tutto si svolge nella isr (Interrupt Service Routine) del timer/counter 1 (`TIMER1_COMPA_vect`), che viene attivata ogni 7ms circa poiché abbiamo caricato 100 in OCR1A nella fase di inizializzazione.

Ad ogni chiamata di questa isr si verifica in che fase ci troviamo: se '`direzione`' è zero la lumi-

Listato 1

```
/*********************************************************************
 *   Programma per rendere "pulsante" il LED LD9 della DEMOBOARD     *
 *    usando il T/C 1 per la velocità del lampeggio                  *
 *    e T/C 2 per generare il PWM                                    *
 *********************************************************************/

#include <avr/io.h>
#include <avr/interrupt.h>

/* Dichiarazione variabili globali */
uint8_t luce = 0;          /* livello di luminosità */
uint8_t direzione = 0;     /* 0 = lum. crescente; 1 = lum. calante */

void main(void) {

/* Inizializzazione */

         DDRA = DDRB = 0xFF;  /* Tutti i pin di PORTA e PORTB sono uscite */
         DDRC = 0b10111100; /* PC0, PC1 e PC6 sono ingressi */
         DDRD = 0b11110011; /* PD2 e PD3 sono ingressi */
/* forziamo a '0' logico tutti i pin dei port impostati come uscite */
         PORTA = PORTB = PORTC = PORTD = 0;

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

         TCCR1A = 0x00;
         TCCR1B = (1 << CS12) | (1 << CS10) | (1 << WGM12);
         OCR1A = 100;               /* con 100 il timer scatta ogni 7 ms circa */
         TIMSK = (1 << OCIE1A); /* abilitaz. int. Su Output Compare Match */

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

/* impostazione T/C 2 come fast PWM, no prescaler */
         TCCR2 = (1 << CS20) | (1 << WGM21) | (1 << WGM20) | (1 << COM21);

/* Fine inizializzazione */

         while (1) {   /* main loop: ciclo infinito */

                  /* non dobbiamo far nulla! */

         }
}

/**** ISR T/C 1: con OCR1A = 100 è chiamata ogni 7ms ****/
ISR (TIMER1_COMPA_vect) {
         if (direzione == 0) { /* fase di luminosità crescente */
                  if (luce < 255)  /* limita l'intervallo di 'luce' a 255 */
                           luce ++;     /* incrementa 'luce' */
                  else
                           direzione = 1;/* se 'luce' = 255 inizia la fase calante*/
         }
         else {   /* fase di luminosità calante */
                  if (luce > 0)  /* controlla se 'luce' > 0 */
                           luce --;   /* se sì decrementala */
                  else
                           direzione = 0; /* altrimenti passa a fase crescente */
         }
         OCR2 = luce; /* infine aggiorna OCR2 per variare il PWM sul LED */
}
```

nosità deve crescere e quindi incrementiamo di uno la variabile 'luce' con l'istruzione C: "luce ++;" . Se però 'luce' vale 255 la luminosità è arrivata al massimo, quindi iniziamo la fase calante mettendo '1' nella variabile 'direzione'. Alla prossima chiamata della isr si andrà ad eseguire il codice relativo alla fase di luminosità calante sin quando 'luce', decrementata ogni volta, non arriva a zero (luminosità minima): in tal caso 'direzione' è riportata a zero ed il ciclo ricomincia. Prima di "uscire" dalla isr, viene inoltre sempre aggiornato il valore del PWM al LED (impostato da OCR2) con il valore corrente di 'luce'.

BUZZER

Sulla demoboard vi è un piccolo cicalino piezoelettrico, che se alimentato con una tensione variabile produce una nota acustica. Questo dispositivo è collegato sul piedino PD5, che

se guardate nel capitolo del datasheet "Pin Configurations" corrisponde anche all'uscita *Output-Compare* del canale A del Timer/counter 1 (OC1A): pertanto possiamo utilizzare questa periferica per accendere il buzzer ad una frequenza programmata. Nel **listato 1**, apportate questa semplice modifica nella sezione di inizializzazione: al posto di

```
TCCR1A = 0x00;
```

scrivete

```
TCCR1A = (1 << COM1A0);
```

che corrisponde, come ormai avrete capito, a settare a '1' il bit COM1A0 del registro TCCR1A, mentre negli altri bit del registro viene scritto '0'.
In accordo col datasheet, questo abilita il piedino PD5 ad essere complementato tutte le volte che nel timer/counter 1 vi è un evento di *output compare* del canale A (lo stesso che attiva anche la isr). Riprogrammando il micro col programma così modificato sentirete uscire dal buzzer, più che un suono, una specie di ronzio; d'altra parte il periodo del segnale che "spariamo" dal piedino è pari al doppio del tempo tra due successivi eventi di output-compare, quindi la frequenza è circa 1/(14 ms) = 70 Hz. Per sentire una nota acustica 'degna' (dipende anche dalle caratteristiche del buzzer) dobbiamo andare almeno sui 3-400Hz: per fare questo facciamo contare il T/C 1 solo sino a 20 invece che a 100, quintuplicando la velocità degli eventi di output-compare:

```
OCR1A = 20;
```

La nota acustica prodotta sarà senz'altro più apprezzabile e vedrete che il led, ovviamente, pulserà più velocemente (esattamente 5 volte più veloce). Naturalmente anche con il buzzer potrete sbizzarrirvi nel programma per variarne la frequenza, e magari rendere la variazione automatica realizzando così una sirena!

USCITE DI "POTENZA"

La demoboard mette a disposizione, tramite U3, otto uscite cosiddette "open collector" in quanto si comportano come degli interruttori verso massa, che vengono chiusi quando il rispettivo ingresso IN di U3 è messo a '1', oppure rimangono aperti quando l'ingresso IN di U3 viene posto a '0'. Con questo tipo di uscite si possono, ad esempio, pilotare direttamente piccoli carichi, come lampadine o relè, alimentabili con la tensione di +12V prelevabile dalla demoboard stessa, e resa disponibile (insieme al +5V) su alcune piazzole vicino all'area millefori. Si tenga però presente che la corrente massima disponibile dipende dall'alimentatore della demoboard e comunque non deve superare i 500mA (poiché dev'essere tollerata dal diodo D1). Può pure essere utilizzata la linea a +5V, ma in questo caso la corrente massima non dovrebbe eccedere i 200mA. In **Fig. 1** vi sono alcuni esempi di collegamento dei carichi sulle uscite O<7:0> di U3.
Per monitorare lo stato delle uscite sono stati posti otto LED (LD<8:1>) sugli ingressi di U3, che si accendono quando il rispettivo canale è attivo (a '1') e di conseguenza l'uscita di U3 è a massa.
Gli otto segnali che pilotano gli ingressi di U3 ed i LED non sono collegati direttamente al microcontrollore ma sulle uscite di U2, un *D-latch* ottuplo siglato 74HC573. Questo dispositivo è una sorta di porta di comunicazione a 8 bit tra gli ingressi D<7:0> e le uscite Q<7:0>. Quando il piedino LE (*Latch Enable*) è a '1' logico gli ingressi sono collegati alle rispettive uscite, che ne rispecchiano quindi i livelli logici: se D<7:0> = 0x3F, anche Q<7:0> = 0x3F. Quando

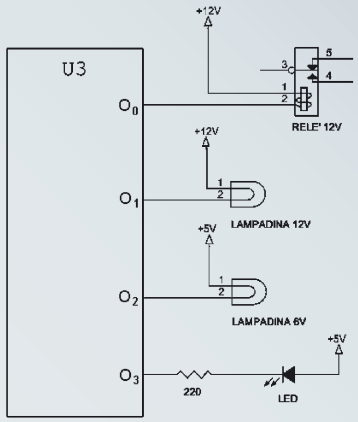


Fig. 1

LE viene posto a '0' le uscite memorizzano il valore che avevano nell'istante in cui 'LE' passa da '1' a '0' e lo mantengono, indipendentemente dalle variazioni successive degli ingressi; quindi se nella transizione 1 → 0 di LE in uscita vi era 0x3F, questo valore permane per tutto il tempo in cui LE è a zero anche se gli ingressi assumono un altro valore.
Anche dopo aver capito come funziona U2 potrebbe però nascere una domanda: perché

usare questo integrato invece di pilotare direttamente U3 con un port del microcontrollore? La risposta è che le linee del micro che utilizziamo per i canali di potenza (quelli relativi a U3) sono anche utilizzate, come si vede nello schema, per pilotare il display LCD di cui parleremo più avanti nella puntata. Il problema è che quando mandiamo i dati relativi al display non vogliamo che essi vadano ad influire lo stato delle linee di U3, pertanto U2 serve a bloccare questi dati (lasciando LE a '0' quando "colloquiamo" col display).
Questo stratagemma nasce dal fatto che il nostro ATMEGA16, seppure ben "dotato", non ha un numero infinito di linee di I/O e questo sistema ci consente di risparmiarne qualcuna: se usassimo tutte linee separate, ce ne servirebbero 8 (per U3) + 6 (per il display) = 14 pin di I/O. Adottando U2 ne bastano 10 (8 linee di dati condivise più i segnali di LE per il latch ed E del display).
Passiamo dunque a scrivere un programmino per gestire le linee di potenza: non ci occorrerà collegare dei carichi sulle uscite di U3 in quanto, come già detto, il loro stato può essere monitorato dai LED LD<8:1>.
Queste otto linee fanno capo, attraverso U2, all'intero PORTB del micro, pertanto per scrivervi un valore l'algoritmo sarà il seguente:

- caricare il valore desiderato sul PORTB;
- alzare (0 → 1) e poi abbassare di nuovo (1 → 0) il segnale LE di U2, per memorizzare il valore.

Con questo sistema si aggiorna contemporaneamente lo stato di tutte le linee.
Il segnale LE fa capo al pin I/O PD6, dunque la transizione voluta si ottiene con le istruzioni:

```
PORTD |= 0x40; /* alza PD6 senza alterare
gli altri bit */
PORTD &= ~0x40; /* abbassa nuovamente PD6
senza alterare gli altri bit */
```

Poiché 0x40 vale 0100 0000 in binario, l'operazione di OR bit a bit del C (operatore '|') va ad alzare il solo sesto bit del PORTD, senza alterare gli altri. Parimenti, siccome l'operatore '~' del C fa la negazione bit a bit, si ottiene che ~0x40 = 1011 1111 per cui l'operazione di AND bit a bit del C (operatore '&') va ad azzerare sempre e solo il sesto bit di PORTD.
Siccome l'algoritmo sopra va eseguito ogni volta che vogliamo aggiornare il valore delle linee di potenza, conviene "rinchiuderlo" in una funzione che poi sarà chiamata dal programma tutte le volte necessarie.
Il programmino di esempio proposto prevede di attivare una uscita per volta, dalla prima alla ottava, per poi invertire il percorso e tornare alla prima, e così via: negli otto LED si visualizzerà un effetto "ping-pong". Per cadenzare lo scorrimento utilizziamo il T/C1, demandando alla sua isr il compito di aggiornare la visualizzazione ed il verso avanti-indietro, esattamente come avevamo fatto per creare il lampeggio pulsante di LD9 (**Listato 1**). Similmente a quel programma, per tenere traccia del verso di percorrenza, utilizziamo la variabile `'direzione'` : 0 = avanti (da LD1 a LD8), 1 = indietro (da LD8 a LD1). Quando gli estremi LD1 o LD8 sono raggiunti, la variabile `'direzione'` viene cambiata ed il ciclo si ripete.
Nel **listato 2** è riportato il sorgente completo del programma: essendo strutturalmente molto simile a quello analizzato nel **listato 1**, ne discutiamo solo le poche differenze. Come già suggerito, abbiamo scritto una piccola funzione (`ImpostaLineePotenza()`) per effettuare l'aggiornamento delle linee di uscita del *D-latch* U2: a tale funzione viene passato come argomento un numero a 8 bit con il valore che vogliamo assumano le linee.
Nella fase di inizializzazione del micro, durante l'azzeramento dei port, è stato aggiunto il codice

```
PORTD = 0x40; PORTD = 0;
```

in modo da generare un impulso sul segnale 'Latch Enable' di U2: poiché i bit di dati sono tutti '0' (l'istruzione precedente azzerava anche PORTB), l'effetto è che all'accensione del sistema tutte le linee di potenza vengono inizializzate a zero (inattive). Questo è importante poiché avrete notato che in mancanza di questa inizializzazione, quando si alimenta la scheda i LED LD<9:1> si accendono in maniera "casuale": il D-latch 74HC573 non possiede un sistema di reset che inizializzi i suoi circuiti interni ad uno stato noto, pertanto all'accensione non è specificato lo stato delle sue uscite.
La variabile `PosizioneLed` viene inizializzata con il bit meno significativo a uno: questo '1' sarà poi fatto scorrere in avanti e indietro rispettivamente con le istruzioni del C:

```
PosizioneLed = PosizioneLed << 1; /* spo-
```

Listato 2

```
/*********************************************************************
 *  Programma per la gestione delle linee di potenza della demoboard  *
 *    con effetto "palleggio"                                          *
 *********************************************************************/

#include <avr/io.h>
#include <avr/interrupt.h>

/* Prototipi delle funzioni */
void ImpostaLineePotenza(uint8_t Dato);

/* variabili globali */
uint8_t PosizioneLed = 0x01; /* il bit a '1' determina il LED acceso */
uint8_t direzione = 0;      /* 0 => da LD1 a LD9; 1 => da LD9 a LD1 */

void main(void) {

/* Inizializzazione */

        DDRA = DDRB = 0xFF;  /* Tutti i pin di PORTA e PORTB sono impostati come uscite */
        DDRC = 0b10111100; /* PC0, PC1 e PC6 sono ingressi */
        DDRD = 0b11110011; /* PD2 e PD3 sono ingressi */
/* forziamo a '0' logico tutti i pin dei port impostati come uscite */
        PORTA = PORTB = PORTC = 0;
        PORTD = 0x40; /* impulso su 'LE' per spegnere le linee di potenza */
        PORTD = 0;

/* impostazione T/C 1 in mod. CTC, con prescaler 1024 */
        TCCR1A = 0;
        TCCR1B = (1 << CS12) | (1 << CS10) | (1 << WGM12);
        OCR1A = 1500;     /* 1500: il timer scatta ogni circa 100 ms */
        TIMSK = (1 << OCIE1A);     /* abilitaz. int. Su Out Compare Match*/
        sei(); /* abilita la CPU a ricevere le interruzioni */
/* Fine inizializzazione */

        while (1) {   /* main loop: ciclo infinito */
                /* non dobbiamo far nulla! */
        }
}

ISR (TIMER1_COMPA_vect) {
        if (direzione == 0) { /* scorrimento LD1 -> LD8 */
                if (PosizioneLed != 0x80)  /* se non è ancora acceso LD8 */
                        PosizioneLed = PosizioneLed << 1;   /* accendi il LED successivo */
                else
                        direzione = 1;/* se era acceso LD8, cambia direzione */
        }
        else {   /* scorrimento LD8 -> LD1 */
                if (PosizioneLed != 0x01)  /* se non è ancora acceso LD1 */
                        PosizioneLed = PosizioneLed >> 1;   /* accendi il LED precedente */
                else
                        direzione = 0; /* se era acceso LD1, cambia direzione */
        }
        ImpostaLineePotenza(PosizioneLed); /* infine aggiorna i LED */
}

/*
 * Funzione ImpostaLineePotenza()
 *   memorizza sulle uscite di U2 il dato contenuto
 *   nella variabile passata come argomento
 */
void ImpostaLineePotenza(uint8_t Dato) {
        PORTB = Dato;
        PORTD |= 0x40; /* alza PD6 senza alterare gli altri bit */
        PORTD &= ~0x40; /* abbassa PD6 senza alterare gli altri bit */
}
```

`sta di una posizione a sx */ PosizioneLed = PosizioneLed >> 1; /* spo-sta di una posizione a dx */`
nella isr del T/C1, nella quale ci si occupa anche di gestire la direzione del "palleggio" ed infine di aggiornare lo stato delle uscite con la chiamata a `ImpostaLineePotenza()`, passandogli come argomento il byte con la posizione aggiornata (il bit che è a '1') del LED da accendere.
Pertanto, anche in questo caso il *main-loop* è vuoto: il processore vi "gira dentro" in attesa

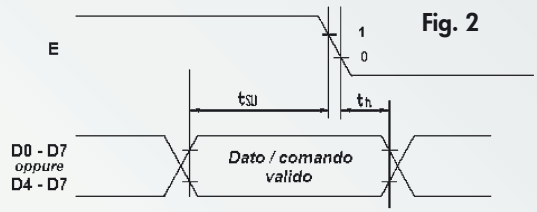


Fig. 2

Characteristic	Symbol	Min.	Unit
Data set-up time	t_{SU}	60	ns
Data hold time	t_h	10	ns

Fig. 3



della isr, la esegue, e torna al *main-loop* in attesa della prossima.
Questo è solo un esempio di come gestire le linee di potenza: di tutto il programma voi potreste mutuare la sola funzione `ImpostaLineePotenza()` per scrivere del codice che soddisfi le vostre esigenze, ad esempio collegando sulle uscite di U3 dei relè potreste pilotare delle lampade o motori a 230V per realizzare lampeggianti, automatismi, e così via.

LEGGIAMO GLI INGRESSI

Passiamo ora a vedere come fare per leggere con il micro ATMEGA16 gli stati logici dei pin di I/O. Come già sappiamo, bisogna innanzitutto configurarli come ingressi, mettendo a zero il bit relativo del registro di direzione: se vogliamo impostare come ingresso il piedino PC0 dobbiamo mettere zero il bit 0 del registro DDRC (Data Direction Register C); ad esempio con l'istruzione

```
DDRC = 0b11111110;
```

poniamo come ingresso il pin PC0, e come uscite tutti gli altri del port C (i piedini PC<7:1>).
Fatto questo, nel programma possiamo leggere (o "campionare", come talvolta si dice) lo stato del piedino PC0 andando ad interrogare il registro "PINC" (Port C Input Pin register), il cui bit 0 rifletterà il valore del piedino, ad esempio:

```
if (PINC & 0x01)  /* se il pin PC0 è '1'
esegui il corpo dell'if */
```

Oppure, utilizzando l'operatore "NOT" del C (il carattere '!'), invertiamo la logica e verifichiamo lo stato '0':

```
if (!(PINC & 0x01))  /* se il pin PC0 è '0'
```

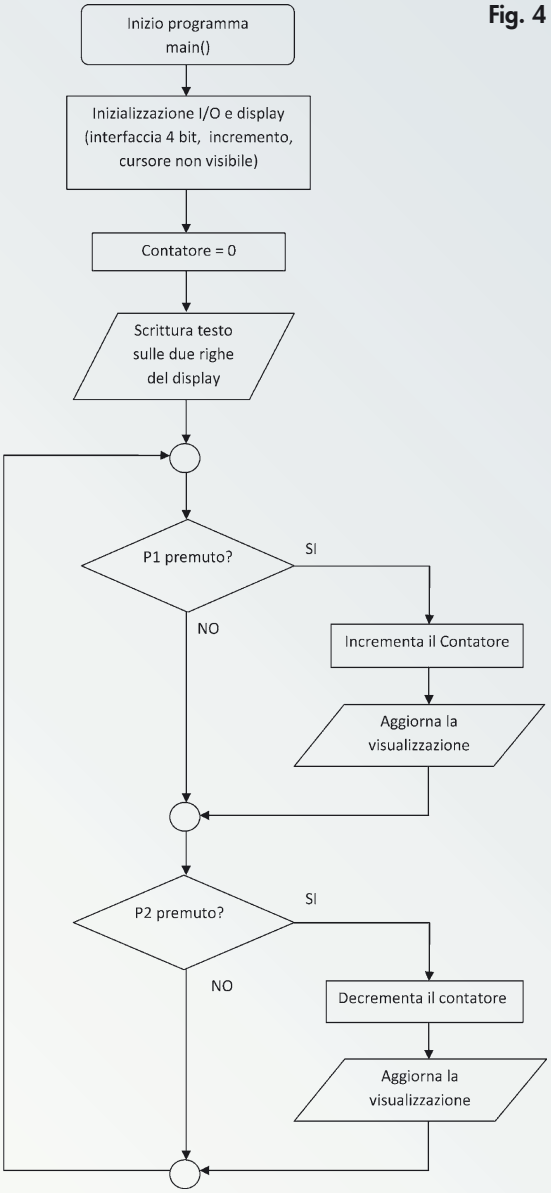


Fig. 4

```
esegui il corpo dell'if */
```

Facciamo un esempio concreto: nella demoboard vi sono due pulsanti, P1 e P2, collegati tramite resistore di pull-up ai piedini PD2 e PD3. Tali piedini sono normalmente a '1' (per via dei resistori di pull-up R17 e R18), e vengono posti a '0' premendo il tasto relativo.
Con queste informazioni, nel listato 3 è mostrato un programmino che accende LD9 quando premiamo P1, e lo spegne premendo P2.
Un'ultima annotazione: l'header "io.h" del WinAVR definisce il numero dei bit di tutti i registri; ad esempio il bit 2 del registro PIND viene definito come:

```
#define PIND2 2
```

E pertanto nel programma per testare il pin PD2, al posto di

```
if (!(PIND & 0b00000100))
```

Listato 3

```
/*********************************************************************
 * Programma per accendere LD9 con P1 e spegnerlo con P2             *
 *********************************************************************/

#include <avr/io.h>

void main(void) {

/* Inizializzazione */

        DDRA = DDRB = 0xFF;  /* Tutti i pin di PORTA e PORTB sono impostati come uscite */
        DDRC = 0b10111100; /* PC0, PC1 e PC6 sono ingressi */
        DDRD = 0b11110011; /* PD2 e PD3 sono ingressi */
/* forziamo a '0' logico tutti i pin dei port impostati come uscite */
        PORTA = PORTB = PORTC = 0;
        PORTD = 0x40; /* impulso su 'LE' per spegnere le linee di potenza*/
        PORTD = 0;

/* Fine inizializzazione */

        while (1) {   /* main loop: ciclo infinito */

                if (!(PIND & 0b00000100)) /* PD<2> = 0 -> P1 è premuto */
                        PORTD |= 0x80;             /* accendi LD9 */

                if (!(PIND & 0b00001000)) /* PD<3> = 0 -> P2 è premuto */
                        PORTD &= ~0x80;            /* spegni LD9 */
        }
}
```

potreste anche scrivere, se vi sembra più chiaro:
`if (!(PIND & (1 << PIND2)))`
e questo ovviamente vale per tutti gli altri piedini dei port.

USIAMO IL DISPLAY LCD

Il display della demoboard è un modulo intelligente in grado di visualizzare due righe di 16 caratteri ciascuna. Cosa vuol dire "intelligente"? Che per pilotarlo bisogna inviargli, in sequenza su apposite linee, il codice dei caratteri desiderati (con codifica ASCII) e la locazione in cui visualizzarli, ed il suo controller interno si preoccuperà di gestire le operazioni necessarie al comando dei *pixel* dello schermo per comporre la visualizzazione richiesta. Il controller in questione è lo "storico" chip HD44780 di Hitachi o equivalente, su cui si basano tutti i display LCD alfanumerici con 8, 16 o 20 caratteri per riga e 1, 2 o 4 righe. Questo tipo di display è da tempo molto diffuso, pertanto sulla Rete vi si trova qualunque tipo di documentazione ed esempi di programmazione in Assembly, C e quant'altro; in ogni caso discuteremo brevemente cosa bisogna fare per renderlo operativo e svilupperemo un semplice programma per visualizzare caratteri e numeri.
Per poter funzionare, il display deve ricevere i dati da visualizzare, ed i comandi per essere impostato (ad esempio per specificare la locazione in cui scrivere); la comunicazione avviene tramite otto linee di dato D<7:0>, il segnale RS (Register Select) per specificare al modulo se sulle linee D<7:0> vi è un comando (RS = 0) oppure un dato da visualizzare (RS = 1), ed un segnale di abilitazione E la cui transizione 1 → 0, analogamente a quanto visto per il segnale LE del *D-latch* 74HC573, dice al modulo di "incassare" il dato presente in quell'istante su D<7:0>. La **Fig. 2** mostra questa fase, evidenziando anche il tempo t_{SU} in cui il dato dev'essere stabile prima della transizione (detto tempo di setup) ed il tempo t_H in cui deve mantenersi stabile dopo la transizione (detto tempo di hold).
Pertanto per trasmettere un'informazione al display dovremo usare il seguente algoritmo:
1) impostare le linee D<7:0> del display con l'informazione da trasmettere;
2) mettere RS a '0' oppure a '1' a seconda che l'informazione sia un comando oppure un dato;
3) alzare E;
4) attendere il tempo di setup;
5) abbassare E;
6) attendere il tempo di hold.

È pure possibile programmare il display affinché accetti informazioni solo su quattro delle otto linee di dato, (D<7:4>), mentre le altre (D<3:0>) possono essere collegate a massa (forzate a '0'). Per fare questo, il microcontrollore deve dividere l'informazione (che è sempre a 8 bit) in due dati di 4 bit (*nibble*) e trasmettere

Il codice ASCII

Questo codice (che è l'acronimo di *American Standard Code for Information Interchange*, ovvero *Codice Standard Americano per lo Scambio di Informazioni*), è nato nei primi anni '60 per uniformare la codifica binaria dei caratteri dell'alfabeto e dei numeri contenuti nelle tastiere dei calcolatori, ovvero per stabilire un'associazione univoca di ogni carattere con un preciso numero binario. Esso è divenuto *de facto* uno standard mondiale di codifica, ed ancora oggi pigiando un tasto sulla tastiera noi trasferiamo al PC la codifica ASCII di tale tasto. Inizialmente la codifica avveniva su 7 bit (128 combinazioni), più che sufficienti per rappresentare tutti i caratteri di una normale tastiera (lettere maiuscole

dec	hex	ascii
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9

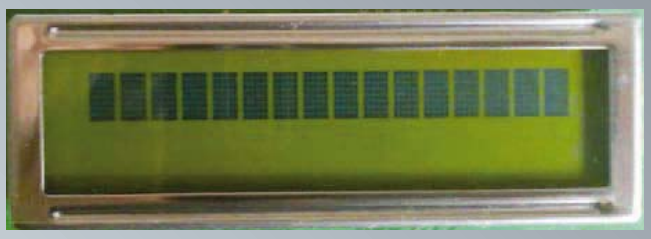
dec	hex	ascii
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
	•	
	•	
	•	
121	79	y
122	7A	z

Fig. A

e minuscole e i numeri), compresi i cosiddetti "codici non stampabili" come DEL, ALT, Control, eccetera. In **Fig. A** mostriamo come esempio le codifiche ASCII dei numeri da zero a nove e delle lettere dell'alfabeto minuscole.
In seguito si estese la codifica su 8 bit per raddoppiare le combinazioni (ora 256) e poter inserire nuovi caratteri "speciali", come i simboli grafici per rappresentare caratteri greci, cirillici, e così via. Va detto però che l'estensione a 8 bit, ovvero la codifica dei caratteri da 128 a 255, *non* è standard (un video terminale potrebbe visualizzare un carattere diverso da quello pigiato con la tastiera ad esso collegata, ma di marca o Paese diversi). I moduli LCD alfanumerici, che ovviamente possono visualizzare solo i caratteri "stampabili", sono aderenti alla codifica ASCII per i codici da 0x20 (il carattere spazio ' ') a 0x7D (il carattere '}'), mentre possono differire per i caratteri che visualizzano con i codici superiori (fino a 0xFF); per sapere quali sono, basta consultare la tabellina normalmente chiamata "Character pattern" sul relativo datasheet.
Da qualche anno è anche comparsa una nuova codifica: l'*Unicode*, pensato con l'ambizione di poter uniformare sotto un unico codice i caratteri rappresentabili di *tutte le lingue del mondo* (ma è ancora in evoluzione), ed associa ogni carattere con un numero binario di ben 21 bit (anche se era nato a 16 bit, dimostratisi subito insufficienti con "soli" 65536 caratteri rappresentabili). Le piattaforme informatiche (linguaggi di programmazione od anche sistemi operativi – Windows CE 6 utilizza solo la codifica Unicode) si stanno adeguando a questo nuovo formato, ma non temete: nel mondo dei microcontrollori e dei moduli LCD il codice ASCII dominerà incontrastato ancora a lungo!
Per maggiori informazioni sul codice ASCII nel web, basta digitarlo in un motore di ricerca e compare "il mondo", a partire dalla sempre apprezzata Wikipedia; per conoscere meglio l'Unicode, potete partire dal sito *www.unicode.org/standard/translations/italian.html*.

sulle linee D<7:4> del modulo prima il nibble superiore (che corrisponde ai bit <7:4> del dato di partenza) e poi quello inferiore (i bit <3:0>); pertanto l'algoritmo visto sopra raddoppia.
Anche nella nostra demoboard usiamo questa modalità poiché, a fronte di un leggero incremento del codice, ci fa risparmiare quattro linee del microcontrollore: i pin di I/O utilizzati sono 4 (dati) + 2 (RS, E) = 6, invece dei 10 necessari per la comunicazione a 8 bit.
Per vedere in concreto come va gestito, scarichiamo dal sito della rivista il codice sorgente di un programma applicativo chiamato "ProvaDisplay.c", che andremo a copiare-incollare dentro l'editor sorgente di un progetto creato in AVR Studio con la consueta procedura analizzata la scorsa puntata; il nome del progetto (e del file sorgente su cui copiamo il contenuto di "ProvaDisplay.c") può essere qualsiasi.
Dopo le operazioni di compilazione e di programmazione, dovremo vedere visualizzati sulle due righe del display i messaggi visibili in **Fig. 3** (se non appare nulla, controllate il trimmer R9 come spiegato nel paragrafo "Il contrasto del display" più avanti in questo articolo).
Il numero della seconda riga, contenuto nella variabile "Contatore", può essere incrementato o decrementato premendo rispettivamente i pulsanti P1 e P2; poiché "Contatore" è definita come `uint8_t` (quindi un byte), si possono rappresentare i numeri da 0 a 255.
Nella **Fig. 4** è mostrato il diagramma di flusso del programma: vi è la solita fase di inizializzazione delle periferiche utilizzate e anche del display: per renderlo operativo dopo che è stato alimentato, infatti, bisogna trasmettergli precise sequenze di numeri per istruirlo sulla modalità di funzionamento, ad esempio per informarlo che l'interfaccia verso il micro è a 4 bit anziché 8 (che sarebbe il valore predefinito).
Nel main-loop viene continuamente controllato lo stato dei pulsanti P1 e P2 per sapere se incrementare o decrementare la variabile "Contatore", ed aggiornare conseguentemente il suo valore sul display.
Poiché lo scopo del programma è abbastanza semplice, ci focalizzeremo sulle funzioni relative alla gestione del display; una volta capito il loro funzionamento, potrete copiare il loro codice per riutilizzarlo in tutte le applicazioni volute; non dimenticatevi anche di inserire pri-

Fig. 5



ma del main() la definizione dei loro prototipi, che riportiamo di seguito:

```
void LCD_Init(void);/* Inizializzazione
display */
void LCD_Command (uint8_t dato); /* Tra-
smissione comando */
void LCD_DisplayChar (uint8_t dato); /*
Visualizza carattere ASCII */
void LCD_DisplayString(char *str_ptr,
uint8_t Nchar); /* vis. Stringa */
void LCD_DisplayChar3Num(uint8_t numero);
/* Visualizza un byte su tre cifre */
void ImpulsoEnable(void);
void RitardoCorto(void);
void RitardoLungo(void);
```

L'INIZIALIZZAZIONE DEL DISPLAY

Il codice necessario all'inizializzazione è contenuto nella funzione `LCD_Init()`, che va necessariamente chiamata prima di poter utilizzare il display. Peraltro è facile sapere se il display non è stato inizializzato (o comunque non correttamente), poiché esso mostrerà (se il contrasto è stato regolato, come spieghiamo più avanti) la prima riga con tutti i pixel accesi e la seconda con tutti i pixel spenti (**Fig. 5**).
In sintesi, nella funzione `LCD_Init()` vi sono le istruzioni per :

- impostare l'interfaccia dati a 4 bit;
- *pulire* il display (spegnere tutti i pixel e portare il cursore sul primo carattere della prima riga (*home*);
- impostare la modalità di visualizzazione su due righe, con cursore invisibile e incremento posizione dopo scrittura di ogni carattere.

Il *cursore*, analogamente a quello lampeggiante che viene visualizzato quando editiamo un documento di testo sul PC, indica la posizione in cui verrà scritto il prossimo carattere. Esso può essere impostato per venire incrementato dopo la scrittura (come quello del PC), decrementato (la scrittura avviene al contrario da destra verso sinistra), oppure rimanere fermo (il nuovo carattere compare sempre nella stessa posizione, ed è l'intera riga a scorrere). Inoltre il cursore può essere visibile come linea (*underscore*), lampeggiare, oppure essere lasciato spento (o invisibile, come nel nostro programma). Per conoscere tutte le modalità operative e come impostarle, invitiamo a fare riferimento al datasheet del display in questione, oppure anche di un qualsiasi altro modulo che utilizzi il protocollo HD44780-compatibile (la quasi totalità dei display alfanumerici). In esso vi troverete una tabella con la lista dei comandi di controllo e di visualizzazione; per il datasheet del nostro modulo (serie 162B della *Displaytech*) questa tabella è chiamata "Control and Display Command", e ve ne riproponiamo la parte con i comandi essenziali in **Fig. 6**. Ogni riga mostra la descrizione di un comando ed il relativo valore da impostare sui bit di dato e del segnale RS (per tutti, 'L' significa *Low* quindi '0', 'H' sta per High, quindi '1', e 'X' significa che il valore è ininfluente). Vi è anche la colonna di un segnale chiamato R/W (Read/Write) che, se posto a '1', permette di leggere dal display i dati (come fosse una memoria) ed il flag di '*Busy*' (indica se il display è ancora "occupato" ad eseguire qualche operazione). Nella nostra demoboard il segnale R/W non viene gestito ed è fissato a zero, in modo che si possa solo scrivere sul display; siccome in tal modo non si sarà in grado di conoscere lo stato del flag 'Busy', inseriamo dopo ogni istruzione un adeguato ritardo per essere sicuri che il display non sia ancora occupato quando trasmettiamo l'istruzione successiva. Tornando alla tabella, essa mostra anche il tempo necessario all'esecuzione di ogni istruzione (colonna *Execution time*), e nel nostro programma abbiamo inserito ritardi "generosi" in modo da esser certi di non violare la tempistica del display in nessuna condizione. A tale proposito, questi ritardi sono stati calcolati per una frequenza massima di clock di 16MHz (che è anche la frequenza massima dell'ATMEGA16); se fate funzionare il micro con frequenze inferiori, potete ridurli in proporzione: in caso contrario il display funziona ugualmente ma per scrivere una riga intera (16 caratteri), se il clock è 1MHz ci vogliono oltre 200ms, che si cominciano a notare: la scritta non appare più "in un colpo solo" ma si intravede l'avanzamento dei caratteri lungo la riga.

LE FUNZIONI DI COMANDO E VISUALIZZAZIONE

Per mettervi in grado di utilizzare la tabella

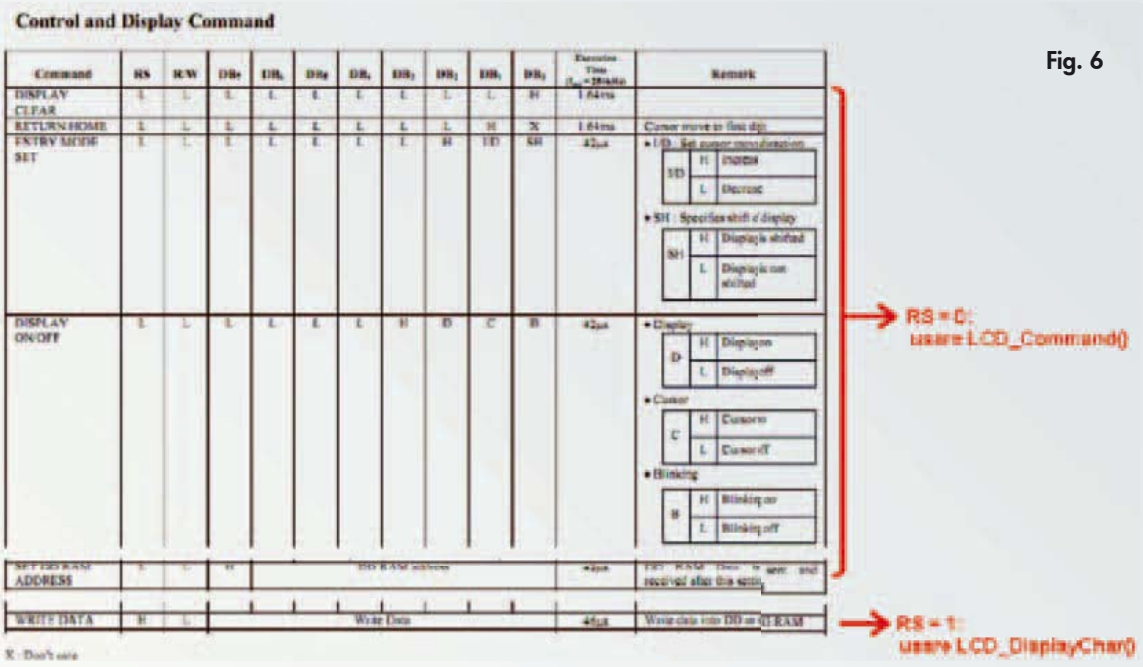


Fig. 6

per personalizzare subito la visualizzazione del vostro display, supponiamo ad un certo punto del programma di volere accendere il cursore per vedere dove si trova, e di farlo lampeggiare; nella quarta riga della tabella di **Fig. 6**, chiamata "Display ON/OFF", troviamo i bit che gestiscono questa opzione: il bit C (*Cursor*) se posto a '1' accende il cursore e il bit B (*Blinking*) ne abilita il lampeggio. Il bit D deve naturalmente essere a '1' poiché accende il display. Pertanto gli otto bit di dato devono essere impostati come 0000 1111 (0x0F in esadecimale). Per trasmettere l'informazione al display usiamo le funzioni *già fatte* del nostro programma "ProvaDisplay.c"; poiché si tratta di un comando (segnale RS = 0, da tabella), dobbiamo usare la funzione `LCD_Command()` e pertanto scriveremo:

```
LCD_Command(0x0F);
```

Per spegnere nuovamente il cursore (ma lasciando acceso il display) scriveremo invece:

```
LCD_Command(0x0C);
```

Se volessimo poi "pulire" il display useremmo il comando della prima riga della tabella ("Display Clear"), in cui si vede che è a '1' solo il bit DB0, quindi l'istruzione sarebbe:

```
LCD_Command(0x01);
```

e così via.

Infine, per muovere il cursore in una data posizione (sulla quale visualizzare poi il carattere voluto), dobbiamo usare il comando "Set DD RAM address", caratterizzato dall'avere DB7 alto ed i restanti 7 bit che specificano la locazione voluta; per semplificare le cose, mostriamo in **Fig.** 7 il numero esadecimale da trasmettere al display per impostare il cursore in una delle 32 posizioni possibili (16 caratteri su due righe).

Ad esempio, per posizionarlo sul terzo carattere della prima riga dovremo scrivere la seguente riga di codice:

```
LCD_Command(0x82);
```

per metterlo sulla prima posizione della seconda riga:

```
LCD_Command(0xC0);
```

e così via.

Notate che se impostate il cursore in locazioni diverse da quelle specificate in **Fig. 7** (ad esempio tra 0x90 e 0xBF), il carattere successivamente scritto sarà "fuori schermo" e dunque non visualizzato.

Per scrivere il carattere da visualizzare (nella posizione corrente del cursore) dobbiamo inviarlo, codificato ASCII, al display mentre il segnale 'RS' è alto, per cui useremo la funzione `LCD_DisplayChar()`. Per scrivere la lettera 'a' l'istruzione è la seguente:

```
LCD_DisplayChar('a');
```

siccome il C trasforma il carattere tra singoli apici nel suo equivalente ASCII. Alternativamente dovremmo scrivere

```
LCD_DisplayChar(0x61);
```

poiché 61 esadecimale è infatti la codifica ASCII del carattere 'a'. Possiamo anche passare alla funzione una variabile:

```
LCD_DisplayChar(MiaVar);
```

stampa la codifica ASCII del valore della variabile `MiaVar`.

Utilizzando la funzione `LCD_DisplayChar()` abbiamo anche costruito la funzione `LCD_Di-`

Fig. 7



splayString(), che visualizza una stringa (ovvero una sequenza di caratteri), a partire dalla posizione corrente del cursore; in tale funzione dobbiamo passare anche il numero di caratteri da scrivere: pertanto per visualizzare la parola "Ciao" dobbiamo scrivere:

```
LCD_DisplayString("Ciao", 4);
```

Al posto della parola esplicita, la funzione può anche accettare un puntatore ad una stringa di caratteri (sui cui chiarimenti rimandiamo ad un qualsiasi manuale del C standard).
Infine, per visualizzare il valore numerico di una variabile abbiamo la funzione `DisplayChar3Num()`: essa converte il numero (da 8 bit) passato come argomento in tre caratteri ASCII corrispondenti al codice dei numeri che lo compongono, e li manda al display. Ad esempio con l'istruzione:

```
DisplayChar3Num(57);
```

inviamo in sequenza al display i codici ASCII '0', '5', e '7' ed esso visualizzerà pertanto `057`.
Nel programma utilizziamo questa funzione per visualizzare il valore della variabile "Contatore"; naturalmente possiamo modificarla per visualizzare solo due cifre (da 00 a 99) oppure al contrario farle accettare variabili più grandi (16 o 32 bit) adeguando il numero di cifre che deve visualizzare.

IL CONTRASTO DEL DISPLAY

Tutti i moduli LCD hanno un piedino (VO) per mezzo del quale, applicando una opportuna tensione, si può regolare il contrasto dei cristalli liquidi del display. Alcuni moduli necessitano di una tensione negativa rispetto a massa, mentre il nostro modello richiede, a temperatura ambiente (in tutti gli LCD il contrasto varia con la temperatura), di applicare su VO una tensione di circa 200mV. Tale tensione è ricavabile agendo sul trimmer R9: regolate tale trimmer sino a quando non vedrete apparire la videata di **Fig. 3** (se avete caricato nel micro il programma "ProvaDisplay"), oppure quella di **Fig. 5** (se il display non è stato inizializzato, magari perché il programma caricato non lo prevede). Se invece il contrasto non è regolato bene il display avrà:

- tutti i pixel accesi se la tensione di VO è troppo bassa;
- tutti i pixel spenti se la tensione di VO è troppo alta.

LETTURA DEI TASTINI

Come potrete verificare eseguendo il programma "ProvaDisplay.c", per quanto velocemente premiate e rilasciate un tastino, la cifra visualizzata incrementa o decrementa di diverse unità, il che non è sbagliato poiché nonostante i ritardi introdotti per la visualizzazione dei caratteri, il processore esegue molte volte al secondo la verifica dei tastini e l'eventuale incremento/decremento della variabile.
Pertanto sarebbe meglio rendere il programma sensibile alle variazioni di stato del tastino, ovvero (consideriamo per brevità solo il tastino P1 collegato a PD2):

- se leggo '1' -> tastino non premuto: non fare nulla;
- se leggo '0' -> tastino premuto: aspetta che venga rilasciato;
- se leggo '1' -> tastino rilasciato: avanzamento contatore.

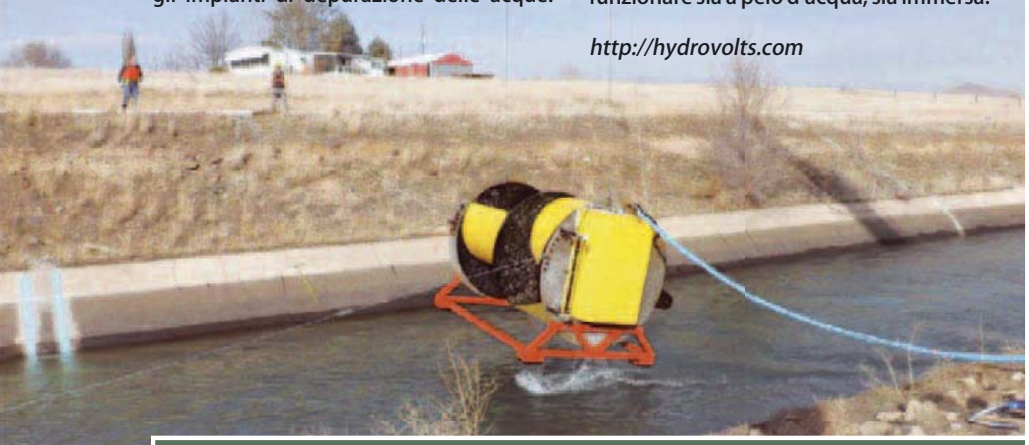
Quindi la cifra verrà incrementata solo quando il tasto è rilasciato. Nella pratica tuttavia si osserva che la cifra incrementa sempre più di una unità ad ogni rilascio del tastino, poiché esso è formato da un contatto tra due superfici metalliche che rimbalzano tra loro ogni volta che lo premiamo o lo rilasciamo: tali rimbalzi vengono interpretati dal programma (che è eseguito molto velocemente) come successive pressioni del tastino e quindi correttamente conteggiate.
Per evitare anche questo problema si inserisce un ritardo di (almeno) qualche decina di millisecondi tra le fasi sopra descritte, realizzando un semplice ma efficace meccanismo detto di antirimbalzo (*debouncing*) del tasto. Abbiamo così modificato "ProvaDisplay.c" nel programma "ProvaDisplay_debounce.c", che potete scaricare e sovrascrivere al primo. Siccome i tasti da verificare sono due, abbiamo raccolto il codice di controllo in una funzione chiamata `ControllaTasto()`, alla quale passiamo il nome del pin da verificare ed essa ci ritorna '1' se il tasto è stato premuto e rilasciato, *filtrando* tutti i rimbalzi. Alla funzione passiamo come argomento anche il port relativo al pin da testare; nel nostro programma è superfluo in quanto è sempre il PORTD, tuttavia l'abbiamo inserito nel caso voleste utilizzare questa funzione in vostri programmi in cui i pin da testare appartengano a port diversi. ■

L'idroelettrico galleggiante porta elettricità ovunque

Lo sfruttamento dell'energia idroelettrica non implica necessariamente grandi turbine poste in centrali di montagna, non a caso nel tempo sono apparsi sul mercato minigeneratori utilizzabili ad esempio in prossimità di piccoli corsi d'acqua. Un recente esempio è la turbina idrocinetica galleggiante della Hydrovolts di Seattle (U.S.A.) progettata esplicitamente per essere inserita nei canali di irrigazione, nei piccoli corsi d'acqua e nelle rogge, negli impianti di depurazione delle acque.

La turbina è del tipo ad asse orizzontale orientato perpendicolarmente alla direzione del flusso d'acqua, che consente l'accesso ad una frazione maggiore della sezione trasversale del canale rispetto ad una soluzione ad asse parallelo. Il risultato è un'elevata efficienza senza significative influenze sul flusso dell'acqua.
La galleggiabilità della turbina può essere regolata in modo da adattarla alla velocità dell'acqua e comunque la turbina può funzionare sia a pelo d'acqua, sia immersa.

http://hydrovolts.com



Il silicone rende giovane il pannello

Sovente sentiamo parlare del "silicone" usato come rimedio estetico, ma indubbiamente ci suona strano pensare che possa far ringiovanire i pannelli solari. Eppure nei laboratori del Fraunofher statunitense si sta sperimentando la laminazione delle celle fotovoltaiche con polimeri inorganici al posto del più tradizionale EVA; usare il silicone nella struttura delle celle solari, peraltro già impiegato nell'incapsulamento dei moduli fotovoltaici dove serve a proteggere le celle e griglie di contatto, nel processo di laminazione (che consiste nell'unione/fusione dei diversi strati da cui è composta la cella) dovrebbe allungarne la vita. Il processo di laminazione attualmente impiega l'etilene-vinil acetato (EVA) però i ricercatori stanno collaborando con la Dow Corning Corporation (nota perché

Il solare funziona meglio in 3D

Il miglioramento dell'efficienza della conversione dei pannelli fotovoltaici è uno dei principali obiettivi dei ricercatori che operano in questo campo; una buona notizia giunge dal MIT di Boston, dove alcuni ingegneri hanno dato vita a celle solari 3D che consentono di sfruttare la quasi totalità della radiazione solare incidente nell'arco della giornata, rispetto a quanto possono fare i pannelli piani installati fissi. Le celle 3D sono composte da colonnine su cui sono montate celle fotovoltaiche disposte con angolazioni differenti, che permettono di intercettare la luce da svariate direzioni. Questa tecnica costruttiva in teoria permette di migliorare lo sfruttamento della radiazione solare nell'arco dell'intera giornata fino a 20 volte, in quanto risolve il problema dell'angolo di incidenza dei raggi del sole che nei pannelli piani è ottimale solo per una ristretta fascia oraria.
Una tecnica analoga è stata approntata più o meno contemporaneamente dagli studiosi del Georgia Tech Research Institute, solo che in questo caso sono state usate celle solari piatte dalla struttura microscopica, simili a tanti piccoli grattacieli (in realtà, nanotubi al carbonio) alti cento micron e con base quadrata.

http://dmse.mit.edu/



Decolla il volo perfetto

Airbus e Air Canada si sono impegnati in un progetto di riduzione dei consumi e delle emissioni di CO_2 dei voli che solcheranno nei prossimi anni i cieli del Canada. Lo scopo verrà raggiunto grazie alla moderna avionica che contraddistingue i velivoli di Airbus e all'alimentazione dei motori turbofan con biocarburanti; il primo volo sperimentale è decollato il 18



produce protesi mediche in silicone e materiali per l'industria automotive) nello sviluppare una nuova soluzione: hanno rivestito le celle fotovoltaiche con silicone liquido che, quando si indurisce, sigilla le celle dando ai componenti elettronici una protezione ottimale.

http://phys.org/news

eMO, la city-car a 4 posti

Tata Motors, in collaborazione con la filiale indiana della Dassault Systemes, sta sviluppando eMO, la prima auto elettrica low cost che verrà proposta entro fine anno ad un prezzo inferiore ai 20.000 $ (circa 16.000 euro). L'azienda indiana, nota per aver prodotto la Nano (l'auto più economica del mondo, sul mercato a 1.700 euro) e per aver acquistato di recente il marchio Jaguar, tenta ora la scalata al mercato delle vetture elettriche. Se è vero che secondo uno studio gli italiani accetterebbero l'auto elettrica purché acquistabile con un massimo di 15.000 euro, la eMO potrebbe davvero dilagare sul nostro mercato, almeno presso il pubblico che dell'auto fa un uso quasi prettamente cittadino. Tata Motors vuole bruciare le tappe, perciò al progetto stanno lavorando 300 ingegneri provenienti da tre continenti. Se tutto andrà bene, presto avremo la possibilità di viaggiare su un'auto elettrica a quattro posti (quindi l'equivalente di una FIAT Cinquecento o Seicento) senza spendere una fortuna, il che potrebbe veramente togliere quel freno che ancora tiene ferma ai nastri di partenza la mobilità elettrica. Certo, un giorno che le auto elettriche saranno diffuse quanto quelle attuali, resterà il problema dell'approvvigionamento di elettricità; ma si risolverà anche quello.

www.3ds.com



giugno da Toronto per atterrare qualche ora dopo a Città del Messico.
L'aeromobile è stato un A319, soprannominato Volo Perfetto per le sue prestazioni in fatto di inquinamento, dato che, numeri alla mano, è riuscito ad emettere meno del 60% di CO_2 rispetto ad un volo tradizionale.
Questo risultato si deve sia all'eco-efficienza dell'A319 che all'impiego di una miscela (al 50%) di biocarburanti ottenuta dal riciclo dell'olio da cucina esausto; a contribuire alla riduzione delle emissioni, il fatto che l'A319 ha seguito la rotta più diretta, usando il profilo di volo verticale più efficiente e attraverso l'avvicinamento in discesa continua (Continuous Descent Approach) a Città del Messico, per risparmiare carburante. Per il rullaggio è stato usato un solo motore (questa tecnica si chiama single-engine taxing) e la fusoliera è stata accuratamente pulita per ottimizzare la penetrazione aerodinamica, cosa che sembra da poco ma che a velocità di crociera quali quelle dei moderni liner ha la sua importanza.

www.airbus.com



43,5% di efficienza dal fotovoltaico a concentrazione



Sharp segna un nuovo record in tema di rendimento delle celle fotovoltaiche a tripla giunzione: l'azienda giapponese ha realizzato una cella a concentrazione usando come semiconduttore l'InGaAs (arseniuro di gallio e indio) la quale ha raggiunto, in laboratorio, un'efficienza di conversione del 43,5%. Il risultato è stato ufficializzato dall'Istituto tedesco per l'Energia Solare Fraunhofer (ISE), che è una delle organizzazioni in grado di certificare formalmente i rendimenti dei dispositivi fotovoltaici.

Il risultato è stato ottenuto (nell'ambito del progetto "R&D on Innovative Solar Cells", promosso dalla New Energy and Industrial Technology Development Organization -NEDO- del Giappone) grazie ad una nuova tecnologia che permette di realizzare uno stack di tre strati fotoassorbenti con l'InGaAs come strato inferiore. Per raggiungere il 43,5% di efficienza di conversione, gli ingegneri della Sharp hanno inoltre ottimizzato la spaziatura tra gli elettrodi sulla superficie del concentratore e minimizzato la resistenza elettrica della cella fotovoltaica.

www.sharp.com

L'eolico si mette in luce



L'elettricità per far funzionare l'illuminazione pubblica costa; per questo si susseguono iniziative volte a ridurre i consumi -come l'adozione di lampioni a LED- o a ricavare la corrente elettrica da sistemi indipendenti o di Energy Harvesting. Una di queste è quella avviata dal comune di Vallarsa (TN) e consiste nel dotare i lampioni di generatori microeolici ELDER (Eolic StreetLight Distributed Energy Resource). Questa soluzione risolverà anche un problema che hanno molti enti: portare l'illuminazione sulle strade lontano dagli abitati, dove è risultato sconveniente effettuare l'allacciamento con la rete elettrica. Ogni dei lampioni che compone il sistema usa lampade a LED e reca in cima un piccolo generatore eolico che si accontenta del vento che soffia nella zona: infatti i lampioni funzionano già quando il vento soffia a 2 metri al secondo, mentre la massima potenza si ottiene con 8 metri al secondo.

Il progetto di Vallarsa costerà trecentomila euro, che saranno completamente finanziati dalla Provincia di Trento.

www.scintec.it

Alle olimpiadi, a bordo di auto elettriche



Per i Giochi Olimpici di Londra si punta sulla mobilità sostenibile, con un progetto cui prenderà parte General Electric Energy Industrial Solutions: 200 auto elettriche saranno a disposizione di atleti e dirigenti delle squadre e per ricaricarle, nella capitale britannica sono in fase di installazione 120 colonnine di ricarica rapida (permetteranno di fare il pieno di energia in sole 4 ore) sparse per il territorio cittadino. Una volta terminati i giochi olimpici, l'infrastruttura entrerà a far parte della rete Source London, la più grande del Regno Unito, che nel 2013 conterà 1.300 stazioni, rimanendo così a disposizione dei cittadini che, trovandosi a disposizione centinaia di punti di ricarica, verranno incentivati a usare le auto elettriche. Non va infatti scordato che a scoraggiare l'acquisto e la diffusione delle auto elettriche non è tanto il costo né la scarsa autonomia, quanto, piuttosto, la difficoltà di ricaricare le batterie; farlo in casa, significherebbe dover disporre di una linea di potenza industriale e non certo domestica, altrimenti la ricarica richiederebbe un'intera giornata. Ogni colonnina sarà in grado di ricaricare sino a 30 veicoli simultaneamente.

www.genewscenter.com
www.tfl.gov.uk



SMAU

INNOVAZIONE DI CASA NELLE IMPRESE ★



E ANCORA...

PERCORSI FORMATIVI, WORKSHOP, CONVEGNI E UN'AREA ESPOSITIVA CON LE MIGLIORI SOLUZIONI DI MOBILITY, UNIFIED COMMUNICATION&COLLABORATION, INFORMATION SECURITY, CRM, FATTURAZIONE ELETTRONICA E DEMATERIALIZZAZIONE, BUSINESS INTELLIGENCE, SISTEMI GESTIONALI INTEGRATI E MOLTO ALTRO.

smau 2012

Milano, 17-19 Ottobre – FIERAMILANOCITY

★

Nel 2011 oltre 80.000 imprenditori e manager hanno sfruttato Smau per innovare la propria impresa.

SMAU: I PROTAGONISTI MONDIALI DELL'INNOVAZIONE; IL TALENTO UNICO DEGLI OPERATORI ITALIANI; LE MIGLIORI SCHOOL OF MANAGEMENT. INSIEME. PER TRE GIORNI. A DISPOSIZIONE DELLA TUA IMPRESA.

IN COLLABORAZIONE CON

Gartner

School of Management – Politecnico di Milano – Dipartimento di Ingegneria Gestionale – MIP

SDA Bocconi

CONTATTI

+39.049.8808444

contact@smau.it

www.smau.it

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".*

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

Gestisci e controlla la tua casa di vacanze con i nostri sistemi GSM

made in Italy

Attiva e controlla da remoto qualsiasi dispositivo elettrico o elettronico: dalla centralina antifurto all'impianto di irrigazione, dalle luci di casa al sistema di condizionamento e raffrescamento. Gestione tramite comandi vocali o messaggi SMS.



cod. TDG135
€ 128,00

Disponibili anche moduli con funzione di:

Combinatore telefonico GSM

- 8 numeri per l'invio degli allarmi tramite chiamata vocale, SMS o entrambi i sistemi
- 2 ingressi d'allarme con messaggi personalizzabili.
- 2 uscite a relé gestibili con toni GTMF e programmabili tramite comandi SMS.
- Possibilità di associare gli ingressi ad eventi di allarme e di controllarli da remoto.
- Alimentazione 9÷32Vdc/1A.





cod. TDG133 • € 96,00
TELECONTROLLO GSM 2 IN / 2 OUT

APRICANCELLO GSM
cod. TDG134 • € 92,00





cod. TDG138 • € 96,00
TELEALLARME GSM CON ANTI JAMMER

TERMOSTATO CON CONTROLLO GSM
cod. TDG139 • € 96,00





cod. TDG140 • € 105,00
TELECONTROLLO GSM CON COMANDI DTMF

Prezzi IVA inclusa.

Tutti i telecontrolli sono facilmente configurabili mediante SMS o in locale tramite PC (per il collegamento al PC è necessario il modulo d'interfaccia USB cod. FT782M • € 15,00 opzionale). I dispositivi vengono forniti montati e collaudati e sono certificati CE-R&TTE.

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

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