

160 PAGINE!

Mensile di progettazione elettronica, attualità scientifica, novità tecnologiche.

Elettronica In

oltre l'elettronica

www.elettronica.in.it

Anno XVII - n. 162
Dicembre 2011 /
Gennaio 2012



**Tecnologia
In volo con
l'ELETTRONICA**



Sagome natalizie a LED

**Costruiamo un Rover
controllato da iPad**



- TiDiGino: telecontrollo GSM basato su Arduino
- Antifurto per pannelli solari
- Alimentatore duale regolabile
- DSO QUAD, l' "iPhone" degli oscilloscopi digitali
- Android Arduino Sequencer
- Interfaccia LabVIEW per K8055
- Ricevitore AM per microspia UHF
- Corso NI Multisim
- Corso Processing

**IN OMAGGIO
LA RIVISTA**

**LabVIEW
World**



• NOVITÀ •

Corso AMEL OPEN SOURCE

160 PAGINE!

Mensile di progettazione elettronica, attualità scientifica, novità tecnologiche.

Edizione
per iPad

Anno XVII - n. 162
Dicembre 2011 /
Gennaio 2012

Elettronica In

oltre l'elettronica

www.elettronica.in.it



**Tecnologia
In volo con
l'ELETTRONICA**



Sagome natalizie a LED

**Costruiamo un Rover
controllato da iPad**



- TiDiGino: telecontrollo GSM basato su Arduino
- Antifurto per pannelli solari
- Alimentatore duale regolabile
- DSO QUAD, l' "iPhone" degli oscilloscopi digitali
- Android Arduino Sequencer
- Interfaccia LabVIEW per K8055
- Ricevitore AM per microspia UHF
- Corso NI Multisim
- Corso Processing

**IN OMAGGIO
LA RIVISTA**

**LabVIEW
World**



• NOVITÀ •

Corso AMEL OPEN SOURCE

IL LABORATORIO NELLE TUE MANI

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

cod. HPS140

€139,00



OSCILLOSCOPIO PALMARE

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

Fornito con caricabatterie USB

ACCESSORI
OPZIONALI:



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



Micro adattatore
presa accendisigari/USB
(cod. CARSUSB1 • € 6,70)

Custodia
protettiva
(cod. BAGPS
€ 10,50)



Prezzi IVA inclusa.



cod. HPG1

€139,00

GENERATORE DI FUNZIONI

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

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

**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.

Sommario

Dicembre 2011 /
Gennaio 2012
n. 162



IN COPERTINA:

Le sagome luminose rappresentano una delle tante possibili applicazioni della nostra CNC: in questo numero spieghiamo come realizzare bellissime figure luminose con cui decorare l'albero di Natale o la nostra casa in queste feste di fine d'anno.

ELENCO INSERZIONISTI

- Blu Nautilus
- Comis Radiant
- Exposition Service
- Fiera di Genova
- Fiera di Scandiano
- Futura Elettronica
- RM Elettronica
- www.mdsrl.it

ARTICOLI

21

GADGET

STRENNE NATALIZIE A LED

Pochi componenti, qualche lastra di plexiglass e la nostra CNC sono gli ingredienti per realizzare magnifiche figure luminose con cui decorare l'albero di Natale o la nostra casa in queste feste di fine d'anno. Vediamo come farle con un PIC ed una scheda basata su Arduino.



21

33

DIDATTICA

CONOSCERE E USARE NI MULTISIM

Occupiamoci di altre due funzionalità di Multisim: MCU, che permette la simulazione di microcontrollori, ed i LabVIEW instruments, che consentono di utilizzare strumenti programmati in LabVIEW. Decima puntata.



33

45

HI-TECH

TIDIGINO, IL TELECONTROLLO BASATO SU ARDUINO

Sfruttando un ATmega 2560, ovvero il cuore di Arduino, abbiamo realizzato un telecontrollo GSM universale dotato dell'hardware che permette di ottenere le funzioni di controllo a distanza 2 IN/2OUT, chiave DTMF, apricancello e termostato attivabile a distanza: ognuna delle quali, in precedenza, richiedeva una scheda specifica.

57

RADIO

RICEVITORE AM PER MICROSPIA UHF

Pur essendo basato su un RX per dati, sfrutta la possibilità di prelevare il segnale all'uscita del demodulatore in modo da essere usato per ricevere segnali BF trasmessi da qualsiasi ricevitore operante sulla frequenza su cui è accordato. Dispone di amplificatore di potenza e regolazione del volume.

63

DIDATTICA

ATEL OPEN SOURCE

Impariamo ad utilizzare i microcontrollori Atmel e il loro ambiente di sviluppo, partendo da nozioni propedeutiche all'apprendimento della programmazione e dell'uso dell'ATmega16. Prima puntata.



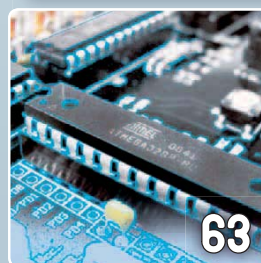
45

71

SICUREZZA

ANTIFURTO PER PANNELLI SOLARI

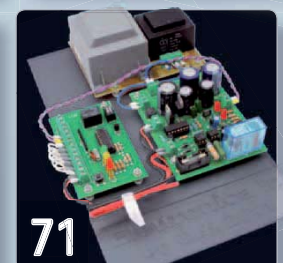
Rileva l'asportazione o l'interruzione di uno dei pannelli costituenti un impianto fotovoltaico, dando l'allarme localmente o, tramite telecontrollo, in remoto.



63



57



71



95

NEWS ED EVENTI

**COSTRUIAMO
UN ROVER
CONTROLLATO
DA REMOTO
MEDIANTE IPAD**



107

Robot Area

95

AUDIO

ANDROID ARDUINO SEQUENCER

Realizziamo un apparato in grado di interfacciarsi ai sintetizzatori musicali dotati di porta MIDI e far loro suonare sequenze di note comandandoli da uno smartphone Android, mediante il protocollo OSC.

121

LABORATORIO

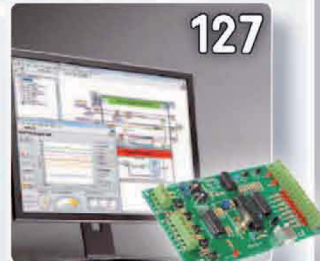
ALIMENTATORE DUALE REGOLABILE

Alimentato con un trasformatore a presa centrale, può erogare una tensione continua simmetrica o asimmetrica rispetto a massa regolabile tra $\pm 1,25$ e ± 18 V ed una corrente di 1A.

121



127



127

APPLICAZIONI

INTERFACCIA LABVIEW PER K8055

Sfruttiamo le potenzialità di LabVIEW per realizzare un'interfaccia grafica con cui controllare la scheda a relé Velleman K8055; un'occasione per imparare ad usare l'ambiente di sviluppo per la gestione di dispositivi non NI, sfruttando periferiche presenti nei PC come l'interfaccia USB.

137



147



137

IN VETRINA

DSO QUAD: L'IPHONE DEGLI OSCILLOSCOPI DIGITALI

Abbiamo provato questo eccezionale strumento di misura e vi descriviamo le nostre impressioni. Ideale per alimentare amplificatori audio e per strumentazione.

147

DIDATTICA

INTRODUZIONE A PROCESSING

Scopriamo l'Object Oriented Programming, che rappresenta un modo più semplice di costruire le nostre interfacce. Quarta puntata.

RUBRICHE

- 01 EDITORIALE
- 05 ELETTRONICA INSIEME
- 13 EVENTI
- 14 APPUNTAMENTI
- 17 COMPONENTI & SISTEMI
- 145 SCIENZA & TECNOLOGIA
- 157 RINNOVABILI
- 160 MERCATINO

La tiratura di questo numero è stata di 18.500 copie.



Mensile associato all'USPI,
Unione Stampa Periodica Italiana

82

TECNOLOGIA

**In volo con
l'elettronica**

Dai fratelli Wright ad oggi,
scopriamo come si sono evoluti
i sistemi di bordo grazie a circuiti
elettronici che assistono i piloti per
garantirci voli sempre più sicuri e
confortevoli, ma anche più ecologici.

Expo Elettronica

MOSTRA MERCATO DI

- > ELETTRONICA
- > INFORMATICA
- > TELEFONIA
- > DVD > GAMES
- > HOBBISTICA

PREZZI SEMPRE PIÙ BASSI!



Calendario
Gennaio/Maggio

2012

MODENA	21-22 GENNAIO
VICENZA	4-5 FEBBRAIO
MORCIANO DI ROMAGNA	11-12 FEBBRAIO
CARRARA	18-19 FEBBRAIO
FAENZA	3-4 MARZO
BASTIA	17-18 MARZO
FORLÌ	5-6 MAGGIO

INFO, NEWSLETTER E TICKET RIDOTTO WWW.EXPOELETTRONICA.IT
DIVENTIAMO AMICI SU WWW.FACEBOOK.COM/EXPO.ELETTRONICA

con il patrocinio di

Ministero dello Sviluppo Economico

organizzato da:



tel 0541 439573
www.blunautilus.it

Info e ticket ridotto:

www.expoelettronica.it

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@elettronica.in.it

“

Cosa bolle in pentola?

In uno spot televisivo ho visto in funzione un dispositivo per cuocere ad induzione, ma che non funziona con le microonde: infatti si tratta di una piastra che a quanto pare permette di usare le pentole tradizionali e che non crea pericolo anche avvicinando una mano.

Insomma, un sistema che non trasferisce il calore per contatto e che però, a quanto pare, non induce neppure la radiofrequenza come nel forno a microonde.

Ma come funziona? E funziona, poi, veramente?

Luca Ponti - Guardiagrele (CH)

La tecnica di cui parli si basa sull'induzione di frequenze più basse di quelle del forno a microonde e non cuoce per diatermia, come invece avviene, appunto, nel forno a microonde: le onde, generate da un grosso avvolgimento di filo in rame (non da un magnetron) investono le pentole, che devono essere di materiale ferromagnetico, scaldandole e facendo così cuocere i cibi in modo convenzionale. La differenza sta nel modo in cui vengono scaldate le pentole: queste ricevono il calore non per il fatto di essere vicine ad una fonte di calore (fiamma o piastra a resistenza), ma lo sviluppano al loro interno a causa dell'esposizione alle onde elettroma-

gnetiche. Questa tecnica, basata sulle perdite per effetto Joule, e che sfrutta la riluttanza magnetica delle pentole, è decisamente innovativa perché consente di realizzare dispositivi non pericolosi in caso di contatto o vicinanza con il nostro corpo, come potrebbero essere i fuochi, le piastre a resistenza elettrica o le piastre ad infrarossi; inoltre sfrutta un principio simile a quello del forno a microonde, senza tuttavia il pericolo delle microonde, rappresentato dal fatto che avvicinando una mano al magnetron, si rischierebbe di cuocersela come vengono cotti gli alimenti.

Le piastre di questo nuovo sistema ad induzione restano fredde ed anche avvicinandovi la mano, non si percepisce alcunché o al massimo si sente un lieve tepore. Il limite di questa tecnica è rappresentato dal fatto che richiede pentole di materiale ferromagnetico, ovvero di acciaio inox con fondo pesante; con l'alluminio ed il rame non funzionerebbe, quindi non permette di cuocere con pentole antiaderenti. I pregi risiedono nella sicurezza d'uso, nella possibilità di porre i piani di cottura vicino a materiali che non sopportano il calore e nell'elevata efficienza, pari a circa il 92 %, che si traduce in una riduzione dei tempi di cottura a parità di potenza impiegata. Se pensi che in un fornello a gas si perde quasi il 60 % dell'energia, il conto lo fai alla svelta...

Prossimamente pubblicheremo il progetto di un semplice dispositivo per fare un po' di esperimenti con il riscaldamento da induzione elettromagnetica.

Continua a seguirci!

Il DCC del ferromodellismo

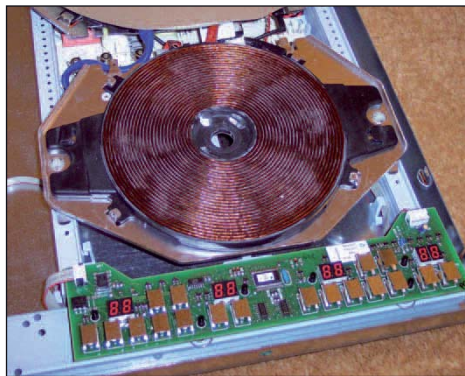
Sto cercando di risolvere un piccolo problema col mio plastico ferroviario: per far circolare più treni senza installare una quantità enorme di binari, dovrei trovare il modo di comandare le motrici dei convogli separatamente, così da poter far circolare più treni sulla stessa linea.

Il controllo distinto è indispensabile perché, pur mantenendo inizialmente distanti due treni che viaggiano sugli stessi binari, le inevitabili differenze di carico e di velocità delle ruote tra i due porterebbero uno ad avvicinarsi sempre più all'altro.

Purtroppo normalmente i treni si controllano variando la tensione tra i binari, almeno nei plastici tradizionali, ma ho sentito di alcuni controller che utilizzano una tecnica in grado di mandare sui binari comandi distinti, interpretabili univocamente dalle singole motrici. Sapete come si chiamano questi apparati? Richiedono treni speciali?

Alberto Calvi - Capua (CE)

Probabilmente ti stai riferendo alla tecnica di controllo DCC, il



Una piastra di cottura ad induzione come appare dall'esterno e vista all'interno: sotto la piastra si trova un potente solenoide avvolto perpendicolarmente rispetto al piano di cottura.

cui funzionamento è grosso-modo quello che hai descritto. Il controllo digitale permette di comandare manualmente moltissime locomotive contemporaneamente senza la necessità di fare sezionamenti sulle rotaie; in pratica con soli due fili e quindi una coppia di rotaie, il DCC permette di comandare in modo indipendente un grande numero (circa un migliaio) di locomotive o anche accessori posti a lato della ferrovia, alimentati dalla tensione presente sui binari.

Il funzionamento del DCC è relativamente semplice: insieme all'alimentazione in corrente alternata per i treni, la centralina DCC invia dei dati digitali che vengono decifrati da speciali locomotori compatibili DCC, all'interno dei quali si trova un decoder. Sostanzialmente si tratta di un telecomando codificato a filo, i cui segnali viaggiano sui binari e di cui ciascun canale (ossia il comando diretto a un locomotore o un accessorio) ha una precisa combinazione; sul locomotore o nell'accessorio del plastico, si imposta la codifica in modo che riconosca una certa combinazione. Quindi basta assegnare ad ogni treno o accessorio un codice diverso da quelli degli altri e siamo certi di poter comandare distintamente ogni elemento, pur disponendo di un'unica linea elettrica.

L'uso di un controller DCC è semplice: se vogliamo che un certo locomotore (ad esempio il locomotore n° 1) si muova, bisogna selezionare il numero 1 sul controller (l'impostazione dipende dal modello di centralina) e girare la

manopola della velocità, come si fa con i variatori tradizionali. Il controller DCC invia comandi in binario che contengono essenzialmente l'indirizzo (codice) dell'unità destinataria ed il livello di tensione che deve essere inviato ai motori (basso, medio, alto). L'introduzione del DCC ha notevolmente ampliato le funzioni dei plastici e consentito di azionare, a bordo dei treni, più dispositivi, contribuendo a migliorare il realismo: se un tempo tutto quel che si poteva fare era accendere le luci di un lato o dell'altro in funzione della direzione (la cosa si realizzava semplicemente alimentando le lampadine mediante diodi, in modo che si accendessero solo con una determinata polarità fra i binari), oggi, con i canali a disposizione, una centralina DCC può comandare distintamente le luci di marcia, quelle delle carrozze, azionare l'avvisatore acustico ecc. Il tutto, naturalmente, funziona sui locomotori predisposti per ricevere comandi DCC e dotati di decoder.

I componenti esatti dell'oscilloscopio

Ho letto l'articolo sull'oscilloscopio palmare che avete pubblicato nel fascicolo n° 160, ma ho trovato alcuni errori di stampa nell'elenco componenti a pag 70/71: R18 è data con il valore di 0 ohm e poi C18 viene indicato due volte; vedendo il piano di montaggio e le foto,

C18 sembrerebbe un elettrolitico. È esatto così?

Giacomo Rizzotto - Asola (MN)

In effetti c'è stato un problema con i condensatori ma solo con quelli: il primo C18 (prima riga di pagina 71) è effettivamente da 10 μ F 16 V, mentre il secondo, ossia quello alla quinta riga, è in realtà C19; quindi quello che è indicato come C19, è in realtà C20. Puoi verificarlo anche dallo schema elettrico della scheda principale, che non lascia spazio a dubbi e puoi anche vederlo nella foto in basso a pagina 71, dove appaiono i valori dei ceramici C19 e C20.

Invece, R18 è esatta: per 0 ohm si intende un ponticello, ovvero una resistenza da 0 ohm (ne esistono in commercio e sono a tutti gli effetti dei componenti in cortocircuito); è stata inserita perché in un primo tempo pensavamo di inserire una resistenza, ma poi ci siamo accorti che non serviva e che potevamo unire le due masse.

Ci scusiamo con te e con tutti i lettori per l'inesattezza e pubblichiamo questa tua lettera per segnalare l'errore.

Il bootloader degli ATmega Arduino

Approfitando del fatto che di Arduino sono disponibili (visto che si tratta di un progetto open-source...) gli schemi elettrici ed in qualche caso anche le tracce del circuito stampato, e considerata la semplicità dell'hardware, vorrei provare a realizzare in casa un modulo come Arduino Duemilanove.

Mi viene però un dubbio che riguarda il bootloader, che, mi sembra d'aver capito, è il programma per far funzionare l'ATmega 328: dove trovo questo programma? È già compreso nei microcontrollori dell'Atmel? È una prerogativa di Arduino? Si può scaricare da Internet e si carica sempre dalla seriale?

Roberto Cuzzolin - Padova

Arduino è effettivamente un progetto open-source, tuttavia la replicabilità delle schede e in generale del suo hardware è vincolata proprio dal bootloader; questo è un semplicissimo programma che si avvia dopo ogni reset e che serve a predisporre il microcontrollore al caricamento degli sketch, ossia dei programmi che scriviamo con l'IDE di Arduino per realizzare le nostre applicazioni.

Il bootloader è quindi come il BIOS dei computer, che imposta la macchina per leggere dal disco rigido o dall'unità



Una tipica centralina DCC per modellismo ferroviario.

di memoria di massa da cui caricare il sistema operativo e in generale i programmi da eseguire. Questo bootloader si deve reperire e poi caricare nella memoria Flash dell'ATmega (nel tuo caso, dell'ATmega328). Questa operazione, però, può essere effettuata solo se hai un programmatore per questa classe di microcontrollori; visto che l'attrezzatura non è proprio economicissima, per uno sperimentatore che voglia fare i suoi piccoli progetti non ha senso comperare il programmatore: è molto più semplice acquistare un Atmega in cui è già stato caricato il bootloader. A tal proposito, sul web puoi trovare diverse aziende (cerca anche nel sito ufficiale www.arduino.cc) che vendono microcontrollori con precaricato il bootloader e che puoi inserire nei tuoi circuiti basati sull'hardware di Arduino; una di queste è la Futura Elettronica (www.futurashop.it).

Il bus VME

Vorrei interfacciare un apparato dotato di sensori con un dispositivo di elaborazione, solo che secondo la documentazione è compatibile con VME Bus. Di cosa si tratta? È qualcosa tipo i vari PC-Bus? È standardizzata? E dove si usa??

Luigi Romano - Torino

Il VME Bus è un bus standard per computer, tipicamente industriali, sviluppato originariamente dalla Motorola circa trent'anni fa ed ancora utilizzato ampiamente in informatica e in alcune schede di controllo per svariate applicazioni, macchinari, controlli automatici ecc. VME è l'acronimo di VERSABus Module Eurocard, perché il sistema deriva dai moduli VERSA Bus, sviluppati dalla Motorola (durante il periodo dello sviluppo del microprocessore 68000) per realizzare un bus di comunicazione (chiamato appunto VERSABus) impiegando unità in formato Eurocard; questo standard prevedeva e prevede tutt'oggi schede di formato 100x160 mm (peraltro il più usato nei tagli delle basette ramate in bachelite e vetronite vendute al dettaglio negli ultimi decenni...) con connettori a due o tre file da 32 contatti a passo 2,54 mm, maschi, collocati sul lato corto. Lo standard Eurocard è stato la base per la realizzazione dei sistemi a rack e prevede, oltre al formato euro standard, il "doppio Eurocard", adottato per le

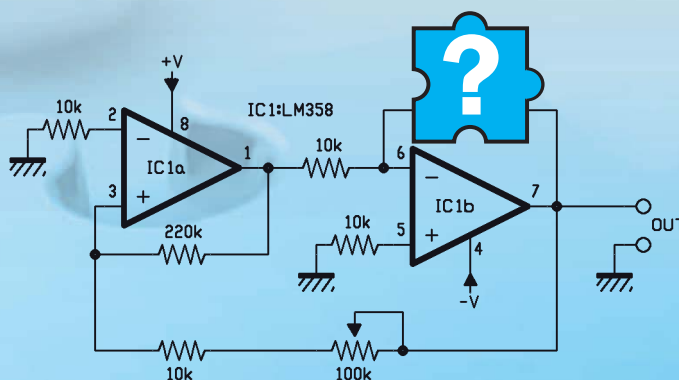
schede un po' più complesse; il doppio Eurocard usa schede grandi circa 230x160 mm (il doppio di una card euro con lo spazio in mezzo fra i due connettori) e prevede due connettori a passo 2,54 mm a due o tre file, posti sul lato lungo. Tutti i connettori sono ad inserzione diretta su un back-panel, ossia la scheda che fa da sfondo al rack e realizza tutte le interconnessioni; nel caso del VME, trasporta il bus. Il bus VME utilizza un protocollo di comunicazione caratterizzato da linee dati ed indirizzi separate: quelle di indirizzo corrispondono a 32 bit (ma possono anche essere 24, 16 o 8 bit) mentre per i dati si utilizzano 4, 16, 24 o 32 bit. Tipicamente le schede VME Bus si collegano mediante due connettori, generalmente indicati con P1 e P2, entrambi dotati di cinque file di 32 contatti; P1 contiene i 24 bit di indirizzo ed i 16 di dato, più tutti i segnali di controllo, come i data_strobe (DS1 e DS0), gli address modifier, eccetera. Adottato da numerosi costruttori, il VME Bus ha ottenuto a suo tempo anche

la "benedizione" della IEC, prendendo il nome di IEC 821 VMEbus, nonché di IEEE ed ANSI, da questi chiamato ANSI/IEEE 1014-1987. Il bus VME si è evoluto negli anni, fino ad arrivare all'attuale versione: il VME64, che gestisce fino a 64 bit di indirizzamento e permette velocità di comunicazione dell'ordine dei 40 MB/s. Il protocollo più recente è il VME64x, che in più implementa l'hot-swap, ossia la possibilità di estrarre le schede dal rack mentre stanno funzionando, senza correre il rischio di danneggiarle.

Il radiocomando α 5,8 GHz

Da qualche tempo trovo in commercio radiocomandi per radiomodellismo ed apparati di trasmissione audio/video (per telecamere con microfono) operanti alla frequenza di 5,8 GHz. Sono ancora molto costosi ma vorrei provarne uno, anche perché ormai sui 2,4

Cosa manca a questo circuito?



Prosegue il quiz proposto nel fascicolo n° 161 di novembre: sapete indovinare che tipo di circuito è quello illustrato qui sopra? E che funzione dovrebbe svolgere? Eventualmente, che tensione possiamo rilevare tra i morsetti siglati OUT?

Attendiamo le vostre risposte direttamente nell'apposita sezione del nostro sito www.elettronica.in. Al vincitore, andrà in premio un abbonamento annuale ad Elettronica In.



GHz trasmette mezzo mondo ed è difficile trovare canali liberi. Mi viene, però, un dubbio: fino ad oggi credevo che si potesse operare liberamente su frequenze come i 434 MHz, gli 868 MHz ed i 2,4 GHz, che corrispondono alla cosiddetta banda ISM. Peraltro gli apparati a 5,8 GHz vengono definiti ISM anch'essi, ma è giusto?

Francesco Cifalà - Erice (TP)

La banda dei 5,8 GHz (per l'esattezza è estesa tra 5.725 e 5.875 MHz) è effettivamente classificata anch'essa come ISM, perché vi si può operare come nella 2,4 GHz, dove sono ammessi sistemi wireless di comunicazione dati, radiocomandi, sistemi di telemetria e diagnostica, e altri ancora.

Proprio la forte diffusione di apparati Bluetooth, Wi-Fi, router wireless e reti wireless private, ricetrasmittenti audio/video, ha spinto a cercare un'alternativa alla banda dei 2,4 GHz, che pure conta su numerosi canali e dove operano molti sistemi in grado di commutare da un canale all'altro quando quello su cui lavorano è troppo disturbato o non permette una ricezione del segnale ad un livello adeguato ad ottenere una corretta discriminazione.

È nata così la banda dei 5,8 GHz, dove si può trasmettere a patto di rimanere in locale, ossia nell'ambito di un centinaio di metri o poco più; infatti, come cita il PNRF (Piano Nazionale di Ripartizione delle Frequenze, varato nel 2009 dall'allora Ministero delle Comunicazioni) ed in conformità alla norma internazionale ERC/REC 70-03, sui 5,8 GHz si possono usare trasmettitori che abbiano una potenza fino a 25 mW e.i.r.p., con una specifica spaziatura tra i canali che, ovviamente, va rispettata. Prima di acquistare un apparato operante a 5,8 GHz, assicurati quindi che sia omologato CE e risponda alle normative del Ministero P.T.

Come funziona il Tutor autostradale?

Come tanti automobilisti, mi sono chiesto qual è il meccanismo di funzionamento del sistema Tutor installato in molti tratti autostradali e, in special modo, come faccia a rilevare la velocità media dei veicoli e ad identificare senza sbagliare una specifica automobile o un camion. So un po' come funzionano i vari Teleser e Autovelox, che rilevano l'infrazione istantanea, ossia il superamento della velocità in un certo punto ed all'istante, ma del Tutor ci capisco poco. Sapete dirmi qualcosa di più?

Samuele Congiu - Sassari

Visto che siamo a cavallo tra la fine di quest'anno e l'inizio del nuovo, quindi in un periodo di partenze, rispondiamo volentieri alla tua lettera perché può essere utile per te, come per chi si metterà in viaggio, sapere come funziona questo importante strumento che, a differenza di quelli cui hai accennato, permette di sanzionare chi ha per abitudine l'eccesso di velocità, senza andare a punire chi, magari per un istante ed una situazione momentanea, preme un po' di più sull'acceleratore. Ebbene, il Tutor si basa su un sistema di computer interconnessi in rete, nei quali gira un sofisticato software di riconoscimento d'immagine che collabora con dei sensori capacitivi di passaggio sepolti pochi centimetri sotto il manto stradale. I sensori sono posti ad una certa distanza, che permette di rilevare il numero degli assi del veicolo e valutare se è un'automobile o un camion a tre o più assi. Il sistema è composto da due barriere: una d'entrata ed una d'uscita, che sono l'inizio e la fine del tratto in cui viene effettuato il monitoraggio della velocità. Alla barriera d'inizio, mediante telecamere, il sistema

fotografa gli automezzi che passano nelle varie corsie (c'è una telecamera per corsia) e memorizza i fotogrammi, estraendo da essi i dati relativi alla targa; quindi, alla barriera d'uscita, un altro set di telecamere fotografa i veicoli che passano, sempre rilevandone il passaggio mediante sensori posti sotto il manto stradale ed estraendo l'ID della targa. I dati relativi alle targhe correlati dall'ora precisa di passaggio vengono inviati ad un server centrale che effettua l'elaborazione. Man mano che i veicoli "visti" all'entrata passano dalla porta di uscita, il sistema cerca le loro targhe e quando le trova può operare in due modi: se il rapporto tra la distanza (che è nota) tra le due porte ed il tempo di percorrenza equivale ad una velocità superiore a quella consentita, salva le immagini della vettura corredate da informazioni come orario d'entrata ed uscita, velocità media, eccetera, e trasmette il tutto alla Polizia Stradale; se, invece, la velocità desunta rientra nei limiti, scarta i fotogrammi e cancella tutti i dati dall'archivio.

Va notato che se tra le due barriere del Tutor si trova un'uscita dell'autostrada (può accadere, perché i tratti in cui è installato il Tutor possono essere lunghi fino a 25 km circa) il sistema ne tiene conto, nel senso che se la vettura rilevata all'inizio del tratto non transita sotto il secondo rilevatore Tutor entro un tempo corrispondente ad una velocità inferiore a quella sanzionabile, i dati della vettura vengono automaticamente cancellati. Il Tutor funziona in qualsiasi condizione atmosferica e di illuminazione (di notte, con la pioggia, in presenza di nebbia con visibilità fino a 30/40 metri).

Come è stato per sistemi che l'hanno preceduto (pensiamo al Teleser, che secondo alcuni si poteva eludere appendendo un CD-ROM allo

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.elettronica.in/webforum



specchietto retrovisore dietro il parabrezza: oggi non se ne vede più perché in breve tempo chi l'ha provato ha scoperto a sue spese che non funziona), anche sul Tutor sono state create "leggende" e false credenze, soprattutto sulla possibilità di eludere tale meccanismo.

Ad esempio, capita di vedere in giro automobilisti poco disciplinati (ed è già una gentilezza definirli così) che arrivando in prossimità del Tutor si spostano sulla corsia d'emergenza per evitare i sensori posti nelle corsie; purtroppo per loro, da tempo il Tutor funziona anche se si guida sulla corsia di emergenza, perché è in grado di rilevare questo tipo di infrazione e comunicarla alla Polizia Stradale, che in questi casi non va troppo per il sottile: si pagano circa 170 euro di multa, ci si vede ritirare la patente da due a sei mesi e quando la si riprende, ci sono 10 punti in meno! Anche guidare con le luci della targa spente o le luci spente, non serve a nulla, perché le telecamere del Tutor effettuano anche riprese all'infrarosso. Per eludere il Tutor bisognerebbe fermarsi per abbassare la velocità media prima della porta d'uscita, ma allora tanto vale restare nei limiti di legge.

Il telefono al riparo dal telemarketing

Come tutti i possessori di un telefono fisso il cui numero è negli elenchi, sono spesso disturbato da telefonate di pubblicità e telemarketing. Esiste un circuito o un sistema elettronico da applicare al telefono per identificare queste chiamate e bloccarle? Insomma, una sorta di discriminatore telefonico per tutelare la mia privacy?

Alessio Piccioli - Ferrara

Per risolvere il tuo problema, che poi è quello di tanti utenti, non serve l'elettronica: da un anno è attivo il Registro Pubblico delle Opposizioni, realizzato dal Ministero dello Sviluppo Economico e che accoglie le richieste dei cittadini che non desiderano essere contattati telefonicamente a fini commerciali o promozionali. L'iscrizione al registro si può fare on-line, accedendo al sito Internet www.registrodelleopposizioni.it, oppure con una semplice chiamata al numero verde 800.265.265 dalla linea telefonica corrispondente al numero che si vuole escludere dalle chiamate commerciali, ovvero per e-mail.

”

FIERE ELETTRONICA INFORMATICA e RADIANTISMO



con Mercatino Radio Amatoriale e Surplus



FIERA DI PIACENZA

PIACENZA EXPO' - LOCALITA' LE MOSE

**17-18
DICEMBRE**

ORARIO CONTINUATO 9.00 - 18.00

**2012 PROSSIMI
APPUNTAMENTI**



4-5 FEBBRAIO 2012

RAVENNA

PALAZZO MAURO DE ANDRE'

11-12 FEBBRAIO 2012

FIERA DI FERRARA

VIA DELLA FIERA 11 - FERRARA

24-25 MARZO

BOLOGNA

PALA NORD - PARCONORD

21-22 APRILE

PIACENZA

PIACENZA EXPO' - LOCALITA' LE MOSE

ELETRONICA IN

Cell. 348 9030832

www.mondoelettronica.net



VALE COME RIDOTTO ANCHE IN FOTOCOPIA



La Più Prestigiosa Rivista di Progettazione Elettronica
per un anno, direttamente a casa tua!

Campagna
abbonamenti



2011
2012



Solo per te ...

- ☐ L'abbonamento annuo (di 10 numeri) costa € 45,00 anziché € 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.elettronica.in.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

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.

NEWScoprila anche su
iPadÈ 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. 113.
- Un volume a scelta della collana "L'ELETTRONICA PER TUTTI".
- La possibilità di scaricare gratuitamente dal sito www.elettronica.in 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:

X 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

Nome:		Cognome:	
Via:		N°:	
CAP:	Città:	Provincia:	
Telefono:		e-mail:	
Data:	Firma:		

Resto in attesa di vostre disposizioni per il pagamento.

Al 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

Resto in attesa del primo numero e degli omaggi.

LA PRIMA RIVISTA ITALIANA
PER LA COMUNITÀ LABVIEW

LabVIEW

World



IN QUESTO NUMERO:

- Novità dal mondo LabVIEW
- Tutorial
- Tips & Techniques
- Interviste
- Applicazioni
- Scuola di LabVIEW
- Eventi, seminari...

e molto altro ancora

**abbonati
subito!**

Rivista a cadenza trimestrale. Costo abbonamento annuale € 20,00.

Abbonamenti, numeri arretrati, download su www.labviewworld.it

NIDays 2012

Forum Tecnologico sulla Progettazione Grafica di Sistemi

NIDays
FORUM TECNOLOGICO
SULLA PROGETTAZIONE GRAFICA DI SISTEMI

Torna a Roma il 29 febbraio presso l'Hotel Ergife, NIDays, l'annuale Forum Tecnologico sulla Progettazione Grafica di Sistemi organizzato da National Instruments, un'intera giornata dedicata alla presentazione dei più recenti prodotti nel campo della progettazione, controllo, automazione, produzione e test con sessioni interattive, demo e premiazioni, nonché un'area espositiva con numerose aziende partner che propongono le applicazioni realizzate con i prodotti National Instruments. Un appuntamento che è anche l'occasione per fare il punto sulle grandi sfide che ci riserva il futuro e sulle tecnologie che utilizzeremo per affrontarle e risolverle.

L'anno scorso NIDays ha scelto quale simbolo dell'innovazione tecnologica la figura di Thomas Edison passando in rassegna innovazioni del passato e progetti per il prossimo futuro; quest'anno NIDays ricorderà i primi 35 anni dalla nascita di National Instruments per cercare di scoprire cosa ci riserveranno i prossimi 35 anni.

Nella Keynote del mattino Eric Starkloff, Vice President Product Marketing, illustrerà come percorrere le nuove vie di innovazione grazie alla Progettazione Grafica di Sistemi.

Durante le sessioni pomeridiane "Programmatori di LabVIEW" verrà spiegato come ottenere il massimo da LabVIEW, sfruttando tutte le risorse per realizzare interfacce professionali, approfondendo la conoscenza dell'ultima versione del programma. Nelle quattro conferenze tecniche "Acquisizione Dati e Collaudo" gli esperti NI illustreranno come ottimizzare le prestazioni dei sistemi di test e realizzare misure automatizzate più performanti. Per quanti hanno come obiettivo la realizzazione di applicazioni HIL o real-time c'è la conferenza "Controllo e Monitoraggio Embedded" focalizzata sulle potenzialità dell'architettura RIO e della programmazione FPGA. In un'area appositamente pensata per i professori e gli ITP degli istituti tecnici denominata "Area Pratica NI Scuola", i docenti potranno provare in diretta le funzionalità della piattaforma didattica National Instruments, in totale autonomia e per tutto il pomeriggio. Le postazioni saranno attrezzate con PC, NI Multisim e NI MyDAQ

per imparare a conoscere, attraverso semplici esercizi, le potenzialità offerte da questa piattaforma.

NI Robotics Summit

Il tradizionale summit pomeridiano di NIDays sarà dedicato al mondo della robotica col "NI Robotics Summit" che rappresenterà l'occasione per discutere, insieme ad esponenti d'eccezione della ricerca e dell'industria italiana, delle principali sfide della robotica moderna: tre "Case History" illustreranno concretamente come progettare sofisticate architetture e soluzioni innovative per lo sviluppo del software, della piattaforma di controllo e dell'integrazione con i sistemi di visione.

Durante NIDays verranno premiati i vincitori della 7^a edizione del Premio Nicola Chiari per la Miglior Applicazione di Misura e Automazione. Dal 2006 sono state raccolte quasi 500 applicazioni realizzate dai clienti National Instruments nei settori applicativi più vari: dall'automotive alla ricerca biomedicale, dal monitoraggio delle grandi strutture alla robotica: in comune la voglia di innovare e condividere le proprie idee con la comunità tecnico-scientifica e industriale.

L'area espositiva

È lo spazio dedicato alle idee che prendono forma, ai prodotti di eccellenza e alle applicazioni che possono fare la differenza e rappresenta il cuore del networking tra National Instruments, Clienti, Partner Tecnologici e Alliance Partner NI.

In un'area espositiva di oltre 1000 metri quadri, sarà possibile vedere demo significative, piattaforme e tecnologie NI applicate alla realtà industriale italiana. I visitatori potranno valutare in prima persona le novità di mercato nei settori dell'elettronica, della misura e dell'automazione industriale direttamente dalle aziende leader di settore e interagire con il supporto tecnico di National Instruments.

<http://italy.ni.com/nidays>



Arte e Scienza in piazza



Dopo il grande successo di pubblico della precedente edizione che ha visto la presenza di ben 41.000 persone ai vari eventi, torna dal 2 al 12 febbraio 2012, Arte e Scienza in Piazza, manifestazione di diffusione della cultura scientifica organizzata dalla Fondazione Marino Golinelli in collaborazione con il Comune di Bologna. Oltre 100 eventi tra mostre, spettacoli, incontri con personalità del panorama scientifico e culturale, proiezioni di film, concerti, giochi in piazza e laboratori creativi tra arte e scienza. Il centro storico di Bologna, da Palazzo Re Enzo alla Biblioteca Sala Borsa, da Palazzo D'Accursio a Piazza Nettuno, si trasformerà per 11 giorni in un grande Art + Science Center che coinvolgerà il pubblico di ogni età. Cuore dell'edizione 2012 sarà la mostra di arte e scienza "Da 0 a 100, le nuove età della vita", una mostra d'arte contemporanea e di scienza per capire come e perché il nostro corpo e la nostra mente sono molto diversi da quelli delle generazioni precedenti, e come svilupparne meglio le potenzialità.

Il ciclo di incontri di approfondimento sul rapporto tra arte e scienza verrà coordinato da Alessandro Cecchi Paone e vedrà la presenza di importanti ospiti del panorama culturale e scientifico; non mancheranno incontri con giovani ricercatori e innovatori ed una serie di conferenze sugli eventi astronomici che influenzano la nascita, l'evoluzione e la fine della vita.

Al Science Center si potranno scoprire e sperimentare in prima persona molte meraviglie della scienza: dalla mostra sulle nuove tecnologie delle immagini, al grande planetario full dome per esplorare paesaggi di stelle, pianeti, galassie, all'avventura nello spazio, per provare l'assenza di gravità e guidare un robot su Marte a caccia di forme di vita aliene.

Per i più piccoli, START Laboratorio di Culture Creative, proporrà una mostra laboratorio per raccontare ai ragazzi come l'arte si possa sposare con la scienza e la tecnologia con, ad esempio, un'area dove trasformare giocattoli elettronici in sorprendenti strumenti musicali.

www.lascienzainpiazza.it

■ 27 FEBBRAIO-1 MARZO 2012

MOBILE WORLD CONGRESS 2012

Barcellona • Spagna

Il GSMA Mobile World Congress 2012 che si terrà a Barcellona dal 27 febbraio al 1 marzo è sicuramente l'evento più importante al mondo per la telefonia mobile e le attività connesse. Organizzato dalla GSMA Association, l'associazione che riunisce quasi 800 operatori di telefonia mobile di tutto il mondo e più di 200 produttori di telefoni mobili e sistemi wireless di comunicazione, il MWC si terrà anche quest'anno (e fino al 2018) nella città catalana dopo che le candidature di Milano, Monaco e Parigi non erano riuscite a convincere l'associazione che, a luglio, ha riconfermato la sede di Barcellona per altri 5 anni.

"Redefining Mobile" è il tema al centro dell'edizione di quest'anno che vedrà la partecipazione di tutti i principali produttori mondiali di telefoni cellulari e smartphone, ma anche di operatori telefonici, società di software e fornitori di servizi.

Un'importante serie di keynote, convegni e seminari tecnici affronterà gli aspetti più "caldi" delle tecnologie legate al mondo del "mobile" nonché le soluzioni che la tecnologia wireless può offrire nel

settore dei pagamenti, dell'assistenza medica, del controllo del traffico ma anche in campi attualmente completamente inesplorati o quasi, grazie anche alla maggior disponibilità di banda, garantita dalle tecnologie 4G, LTE e LTE Advanced. Oltre a prendere parte ai vari convegni, il pubblico potrà visitare centinaia di stand con le ultime novità in fatto di telefonini, smartphone e servizi connessi; in occasione del MWC molti produttori presenteranno in anteprima mondiale i loro prodotti con nuovissime tecnologie hardware e software, dalle configurazioni quad-core ai sistemi con Ice Cream Sandwich, la versione 4.0 di Android. Come consuetudine, l'unico grande assente alla kermesse di Barcellona sarà, almeno ufficialmente, Apple, mentre Samsung e HTC prenderanno parte in forze al Mobile World Congress. Evento nell'evento, l'App Planet – mostra, convegno e concorso – per quanti sviluppano applicazioni per smartphone. Da segnalare, infine, nell'ambito delle conferenze, il Mobile Security Forum, l'LTE Forum e il Mobile Cloud Forum.

www.mobileworldcongress.com



■ 28 FEBBRAIO-1 MARZO 2012

Embedded World 2012

Norimberga • Germania



La più importante manifestazione mondiale nel settore della tecnologia embedded si prepara a celebrare il decimo anniversario con un incremento del 25% sia del numero di espositori che della superficie espositiva. Dopo nove anni di costante crescita i padiglioni e le sale per conferenza utilizzate in passato sono diventate troppo anguste per la vasta comunità embedded e così da quest'anno Embedded World avrà a disposizione nuovi spazi andando ad occupare i padiglioni 1, 2, 4, 4 bis e 5 del Centro Esposizioni di Norimberga con ingresso da NCC Ost e NCC Mitte. "Siamo molto lieti che il decimo anniversario coincida con questa nuova configurazione che consentirà di avere più spazio sia per l'area congressuale che per quella espositiva, pur mantenendo una struttura compatta su un unico livello con percorsi brevi tra una struttura e l'altra e con una minore distanza tra le sale congressuali e l'area espositiva" spiega Alexander Mattausch, Exhibition Manager della Mostra. Quest'anno l'evento si svolgerà dal 28 febbraio al 1 marzo 2012, una cadenza ormai stabile nel calendario fieristico della comunità internazionale; all'edizione 2012 sono attesi oltre 20.000 visitatori dopo il record di 19.025 presenze del 2011. Come cambierà il mondo embedded nei prossimi 10 anni? Quali innovazioni tecnologiche dobbiamo aspettarci? Sono le domande a cui tenterà di dare una risposta l'"Embedded World Conference" che si affiancherà all'area espositiva con un mix di seminari e workshop.

Altra conferenza di grande importanza è l'"Electronic Displays Conference 2012" con esperti provenienti dal mondo dell'industria e della ricerca che faranno il punto sulle tecnologie più avanzate in questo settore. Anche il mondo M2M (Machine-to-Machine) trova grande spazio nella manifestazione tedesca con un'area espositiva nella quale vengono presentati terminali, moduli, applicazioni e servizi legati a questo argomento ed un'area congressuale all'interno dell'"Embedded World Conference" a cura della testata Computer&AUTOMATION in collaborazione con l'M2M Alliance.

www.embedded-world.de

■ 7-9 FEBBRAIO 2012

EMV 2012

Düsseldorf • Germania

EMV è l'evento leader a livello europeo sulla compatibilità elettromagnetica: attrezzature, componenti e servizi all'avanguardia in questo settore saranno in mostra dal 7 al 9 febbraio presso il Centro Congressi di Düsseldorf, in Germania. L'edizione 2012 vedrà un ulteriore incremento degli espositori, rispetto ai 111 dell'edizione precedente; tutte le principali aziende del settore – provenienti principalmente da Europa, USA e Asia – hanno riconfermato la loro presenza ampliando, in alcuni casi, lo spazio espositivo. Le nuove tecnologie che stanno emergendo (dall'eMobility alle Smart Grid agli impianti di generazione da energie rinnovabili) presentano tutte nuove e complesse problematiche dal punto di vista della compatibilità elettromagnetica che richiedono at-

trezzature e servizi specifici; la prossima edizione di EMV offrirà l'occasione alle aziende del settore di proporre le proprie soluzioni anche in questi campi. La manifestazione tedesca ospiterà anche la conferenza biennale EMV dedicata ai temi della compatibilità elettromagnetica ed all'evoluzione della normativa del settore; numerose altre conferenze e seminari dedicati ad argomenti più specifici si terranno durante i tre giorni della manifestazione caratterizzando

EMV quale piattaforma ideale per il confronto tra ricerca, sviluppo dei prodotti e delle applicazioni in un settore molto delicato dell'industria elettronica: nel nostro mondo nulla più funziona senza dispositivi elettrici ed elettronici, e questi debbono poter funzionare correttamente senza influenzarsi l'uno con l'altro.

www.mesago.de/en/EMV/



MOSTRE MERCATO

CEREA (VR)
COMPUTERFEST & RADIOAMATORE
AreaExp - Cerea (VR)
Organizzazione: Compendio Fiere
Tel: 337676719 Fax: 0572498205
info@compendiofiere.it
www.compendiofiere.it
14 e 15 Gennaio 2012

NOVEGRO (MI)
RADIANT & SILICON
Parco Esposizioni di Novegro (MI)
Organizzazione: Comis Srl
Tel: 027562711 - Fax: 027020835
radiant@parcoesposizioninovegro.it
www.parcosposizioninovegro.it
28 e 29 Gennaio 2012

RAVENNA
MONDO ELETTRONICA
Pala De André
Organizzazione: Exposition Service
Tel: 054527548 Fax: 0545291179
info@expositionservice.it
www.expositionservice.it
04 e 05 Febbraio 2012

MANTOVA
ELECTRONIC DAYS
PalaBam, Via Melchiorre Gioia 3
Zona favorita Boma
Organizzazione: Italfiere Srl
Tel: 0547415674 - Fax: 0547417357
info@italfiere.net
www.electronicdays.it
11 e 12 Febbraio 2012

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

BUSTO ARSIZIO (VA)
COMPUTERFEST & RADIOAMATORE
Quartiere Fieristico - MalpensaExpo
Organizzazione: Compendio Fiere
Tel: 337676719 Fax: 0572498205
info@compendiofiere.it
[web: www.compendiofiere.it](http://web.compendiofiere.it)
28 e 29 Gennaio 2012

FERRARA
MONDO ELETTRONICA
Quartiere Fieristico - Ferrara
Organizzazione: Exposition Service
Tel: 054527548 Fax: 0545291179
info@expositionservice.it
www.expositionservice.it
11 e 12 Febbraio 2012

SCANDIANO (RE)
FIERA DELL'ELETTRONICA
Quartiere fieristico Scandiano
Organizzazione: Comune Scandiano
Tel: 0522857436 - Fax: 0522854222
entefiere@comune.scandiano.re.it
www.fierascandiano.it
18 e 19 Febbraio 2012

MODENA
EXPO ELETTRONICA
Quartiere Fieristico - Modena
Organizzazione: Blu Nautilus
Tel: 054153295 - Fax: 0541439584
info@blunautilus.it
www.blunautilus.it
21 e 22 Gennaio 2012

VICENZA
EXPO ELETTRONICA
Quartiere Fieristico - Vicenza
Organizzazione: Blu Nautilus
Tel: 054153295 - Fax: 0541439584
info@blunautilus.it
www.blunautilus.it
04 e 05 Febbraio 2012

MORCIANO DI ROMAGNA (RN)
EXPO ELETTRONICA
Quartiere Fieristico: Morciano Fiere
Organizzazione: Blu Nautilus
Tel: 054153295 - Fax: 0541439584
[email: info@blunautilus.it](mailto:info@blunautilus.it)
www.blunautilus.it
11 e 12 Febbraio 2012

CARRARA (MS)
EXPO ELETTRONICA
Quartiere Fieristico: Carrara Fiere
Organizzazione: Blu Nautilus
Tel: 054153295 - Fax: 0541439584
[email: info@blunautilus.it](mailto:info@blunautilus.it)
www.blunautilus.it
18 e 19 Febbraio 2012

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

DAI VOLUME ALLA TUA MUSICA!



DIFFUSORI PASSIVI PROFESSIONALI IN ABS

€ 179,00
cod. VDSA15

- 500W - 2 vie
- Woofer 15"
- Titanium tweeter 3,5"

- 450W - 2 vie
- Woofer 12"
- Titanium tweeter 3,5"

€ 149,00
cod. VDSA12

Amplificatore audio di potenza "SAIGOR" 2x120W rms

€ 158,00
cod. VPA5G100

€ 98,00
cod. PROMIX220

Mixer professionale 3 canali

- 4 ingressi stereo su 2 canali
- Ingresso microfonico con switch "Talk Over"

Mixer stereo 3 canali

- Ingressi MIC, CD/Line e Phono

€ 39,00
cod. PROMIXSUS

Portamicrofono in ABS per aste con attacco Ø5/8"

€ 1,90
cod. MICAS10

Microfono dinamico professionale

- Unidirezionale

€ 17,00
cod. MICPROB

Stand per microfono*

- Altezza regolabile da 95 a 160cm

€ 22,50
cod. MICSS

* Portamicrofono non incluso

Treppiede per casse acustiche

- In alluminio
- Altezza regolabile da 108 ÷ 200cm

€ 42,00
cod. VDL155

Cuffia stereo professionale per DJ

€ 31,00
cod. HPD15



€ 10,00
cod. HPD13

Cuffia stereo con auricolari dinamici



**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.

MCU con RFID reader per controllo accessi, automazione industriale e ... identificazione animali



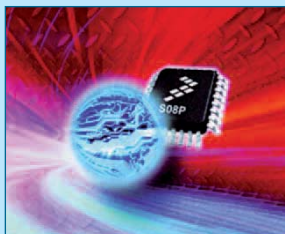
Atmel ha annunciato la disponibilità di un microcontrollore AVR con integrati un lettore RFID e flash memory "in-system self-programmable" da 16 Kbyte. Disponibile in case QFN da 5 x 7mm, l'Atmel ATA5505 opera nella gamma di frequenza compresa tra 100-150kHz (LF) ed è adatto a tutti i tipi di lettori RFID (LF-ASK) e programmatori per applicazioni di controllo accessi, automazione industriale e applicazioni per identificazione di animali.

Per realizzare lettori RFID col chip ATA5505 sono sufficienti pochi altri componenti esterni ed una bobina d'antenna. L'uso del nuovo dispositi-

vo consente l'eliminazione di costosi componenti discreti come i cristalli, attualmente richiesti nei sistemi RFID a 2 chip nonché l'utilizzo di circuiti stampati di dimensioni più contenute. La nuova MCU Atmel dispone di SRAM da 512byte, EEPROM da 512byte, 16 I/O, 32 registri, 2 timer/contatori a 8-bit, universal serial interface (USI), un controller LIN, interrupts interni ed esterni, un convertitore A/D a 10-bit 11 canali, watchdog timer programmabile con oscillatore interno e tre modalità di risparmio energetico selezionabili via software.

www.atmel.com

Nuova generazione di MCU ad 8 bit da Freescale



arrivare sul mercato più rapidamente con una distinta base (BoM) più ridotta.

La famiglia MCU S08P comprende tre classi di dispositivi (PT, PA e PL) ed è conforme alla norma di sicurezza IEC-60730 per gli elettrodomestici. Garantisce un'ottima performance EFT/ESD (electrical fast transient/electrostatic discharge) all'interno di ambienti difficili e rumorosi dove sono collocati appliance, gruppi di continuità, impianti di riscaldamento e condizionamento, controlli intelligenti di illuminazione ed edifici, contatori intelligenti e applicazioni generali di controllo industriale.

www.freescale.com

Freescale Semiconductor ha annunciato due nuove famiglie di microcontrollori (MCU) 8 bit 5V caratterizzati da un ottimo livello di affidabilità e durata per applicazioni industriali e automobilistiche. Scalabili e compatibili con i dispositivi esistenti, questi MCU entry level di facile utilizzo sono stati studiati per aiutare gli OEM ad

LIS3DSH, accelerometro MEMS per sensoristica intelligente

STMicroelectronics ha presentato oggi il primo accelerometro industriale a tre assi ed elevata risoluzione con due macchine a stati finiti al proprio interno. Questi blocchi programmabili permettono di integrare nel sensore funzioni dedicate personalizzabili per riconoscere i movimenti, facendo così un significativo passo avanti nel ridurre la complessità di sistema e i consumi di potenza dei telefoni cellulari e di altri smart gadget sensibili al movimento. Gli utenti possono, ad esempio, rispondere a una telefonata, silenziare la suoneria o attivare un'applicazione come un misuratore di passi, con un sem-

plice movimento predefinito. Le macchine a stati finiti integrate permettono all'utente di implementare con grandissima flessibilità applicazioni personalizzate basate sulla rilevazione del movimento, riducendo il carico di lavoro del processore.

www.st.com



PIC12LF1840, trasmettitore wireless sub-GHz con MCU 8-bit

Microchip annuncia il nuovo PIC12LF1840T48A, primo di una famiglia di dispositivi single-chip ad integrare un microcontroller PIC® a 8-bit in tecnologia eXtreme Low Power (XLP) con un trasmettitore RF sub-GHz. La combinazione di funzionalità assicurata dal PIC12LF1840T48A - alloggiato in package TSSOP a 14-pin - rende il dispositivo una soluzione ideale nelle applicazioni vincolate in termini di spazio, consumi e costi, quali i telecomandi per automobili, per l'apertura di garage, per sistemi di sicurezza domestica ma anche per un ampio range di applicazioni di home e building automation. Il dispositivo è anche ottimizzato per la tecnologia avanzata di code-hopping royalty-free KEELOQ® di Microchip, una collaudata soluzione di pro-

tezione utilizzata dai principali fornitori di tutto il mondo.

Caratteristiche:

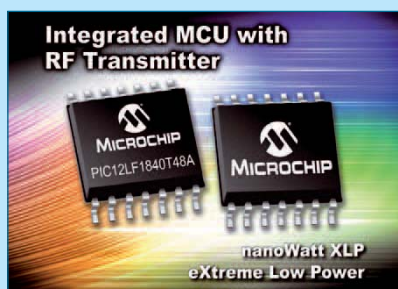
- Architettura single-chip con MCU in tecnologia eXtreme low-power, trasmettitore RF a 418/434/868 MHz e tecnologia

KEELOQ®;

- Mette a disposizione dei progettisti una soluzione completa di controllo remoto single-chip;
- Massimizza la durata delle batterie grazie a una tensione operativa estremamente bassa: solo 1,8V;
- Semplifica la progettazione di un vasto spettro di applicazioni di accesso remoto;

Oltre che per comunicazioni wireless "sicure" il PIC12LF1840T48A è stato progettato per massimizzare la durata delle batterie grazie a una tensione operativa estremamente bassa: solo 1,8V. Il PIC12LF1840T48A è disponibile in package TSSOP a 14-pin.

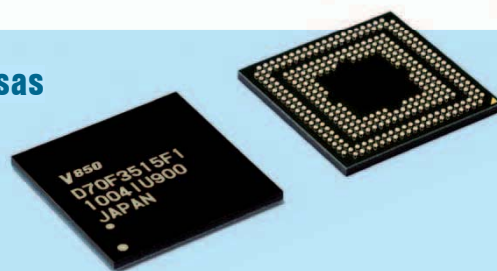
www.microchip.com



Quattro nuovi microcontrollori 32-bit All Flash™ da Renesas

Renesas Electronics Europe ha annunciato la disponibilità di quattro nuovi microcontrollori a 32 bit della famiglia All Flash™: i modelli μ PD70F3515, μ PD70F3514, μ PD70F3512 e μ PD70F3510. Questi dispositivi presentano una capacità di elaborazione massima di ben 1024 MIPS (milioni di istruzioni al secondo) ad una velocità di clock di 200 MHz. Ciò permette a questi chip di occupare il primo posto al mondo

nella classifica degli MCU embedded Flash più performanti. I nuovi controllori dispongono di uno o due core CPU, 2 MB di memoria Flash e sono disponibili in package FBGA da 304 connessioni. Le versioni a singolo core dispongono di una memoria RAM da 64 KB mentre quelle dual-core montano una RAM da 128 KB. La famiglia V850E2M/MN4 a singolo core è particolarmente indicata per sistemi di automazione



industriale, controllo di circuiti di potenza, inverter ed apparecchiature domestiche.
www.renesas.eu

Da NI prime interfacce CAN e LIN a bassa velocità per CompactDAQ

National Instruments ha annunciato le interfacce della Serie C NI 9861 CAN e NI 9866 LIN, i nuovi dispositivi della famiglia di prodotti NI-XNET e i primi moduli CAN e LIN a bassa velocità compatibili con l'intera piattaforma NI CompactDAQ. In quanto parte della famiglia NI-XNET, i nuovi moduli offrono dei vantaggi in termini di produttività come la possibilità di messaggistica hardware accelerata ed elaborazione a bordo scheda. Il modulo single-port di interfaccia CAN NI 9861 della Serie C a bassa velocità dispone di supporto al database CAN per importare, modificare e utilizzare segnali da file FIBEX, .DBC

ed .NCD. Il modulo ha una capacità di carico di comunicazione del 100 per cento fino a 125 kbit/s senza perdere alcun frame. Il modulo LIN NI 9866 è un'interfaccia monoporta con supporto integrato in grado di importare e utilizzare segnali da database LDF assieme al supporto master/slave e gestione temporizzata tramite hardware per applicazioni master. Il modulo ha una capacità di carico di comunicazione del 100 per cento fino a 20 kbit/s senza perdere alcun frame. È possibile utilizzare i nuovi moduli CAN e LIN con lo stesso codice software NI LabVIEW o ANSI C/C++ su una varietà di piattaforme inclusi NI CompactDAQ, CompactRIO, PXI e PCI. Il riutilizzo del progetto consente di risparmiare tempo dato che è possibile utilizzare la stessa applicazione, ad esempio, in laboratorio con PCI, nella produzione di test di fine linea con PXI, nell'impostazione di sistemi per comunicazioni portatili a bordo veicolo con NI CompactDAQ e nella registrazione senza host a bordo veicolo con CompactRIO.

<http://ni.com>



Nuovi transceiver integrati da ON Semiconductor

ON Semiconductor ha presentato due nuovi dispositivi ricetrasmittenti (transceiver) per realizzare reti di comunicazione LIN (Local Interconnect Network) e CAN (Control Area Network) in applicazioni automotive e di controllo industriale.

I transceiver NCV7425 LIN+LDO e NVC7441 con doppia interfaccia CAN rappresentano una soluzione economica, robusta e compatta per facilitare la realizzazione di funzionalità dedicate al confort nella nuova generazione di automobili, nonché sistemi di controllo industriale che utilizzano le reti a standard LIN e CAN.

Il dispositivo NCV7425 integra un regolatore di tensione a bassa caduta (LDO) e un circuito di ri-

pristino (reset). Supportando le specifiche LIN versione 2.x /SAE J2602, agisce da interfaccia tra il controllore del protocollo LIN e il bus fisico con velocità di trasmissione fino a 20 kilobit al secondo (kb/s). Disponibile con tensioni di uscita a 3,3 volt (V) e 5 V, il transceiver NCV7425 supporta carichi fino a 150 milliamper (mA) ed è conforme alle più recenti norme applicabili al settore automotive. Una funzionalità di controllo integrata della velocità dei fronti d'onda consente di ottimizzare il comportamento per quanto riguarda le interferenze elettromagnetiche (EMI).

www.onsemi.com



SSM3J13xTU, nuova generazione di MOSFET

Toshiba Electronics presenta una nuova generazione di MOSFET a media potenza per la gestione dei circuiti di alimentazione dei dispositivi portatili. Gli ultracompatte transistor MOSFET SSM3J13xTU soddisfano i requisiti di commutazione a bassa potenza, alta velocità e alta corrente dei circuiti di alimentazione periferici, in applicazioni come telefoni cellulari, fotocamere digitali e altre apparecchiature alimentate a batteria. Realizzati con la tecnologia di processo Toshiba U-MOSVI a canale P, i

MOSFET SSM3J133TU, SSM3J134TU e SSM3J135TU hanno una corrente di drain nominale (ID) pari rispettivamente a -5,5 A, -3,2 A e -3,0 A. Tutti i dispositivi hanno una tensione nominale tra drain e source (VDSS) pari a 20 V e una tensione nominale tra gate e source (VGSS) pari a ± 8 V. Oltre alla tensione massima tra gate e source, i MOSFET presentano una tensione di soglia di gate (V_{th}) non superiore a 1,0 V, il che permette di avere commutazioni stabili già a partire da 1,5 V.

www.toshiba-components.com



LTC3103, convertitore buck sincrono con 1,8µA a riposo

Linear Technology presenta l'LTC3103 e l'LTC3104, due convertitori buck sincroni da 15V a frequenza fissa in grado di fornire 300mA in corrente continua per tensioni di uscita comprese tra 0,6V e 13,8V. L'LTC3103 consuma solo 1,8µA di corrente di riposo, mentre l'LTC3104 (che offre un'uscita LDO program-

mabile a 10mA) richiede solo 2,6µA di corrente di riposo. Questi dispositivi sono ideali per un'ampia gamma di applicazioni alimentate a batteria, tra cui controllo industriale, reti di sensori remoti, sistemi wireless a bassa potenza e applicazioni di energy harvesting. Entrambi i dispositivi funzionano in un range di tensioni di ingresso tra 2,5V e 15V, supportando quindi un'ampia serie di sorgenti d'ingresso, incluse batterie a singo-

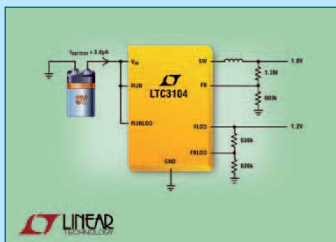
la cella e multicella, supercondensatori e rail di sistema.

L'LTC3103 e l'LTC3104 utilizzano una frequenza di commutazione fissa di 1,2MHz e un raddrizzatore sincrono per raggiungere un rendimento di ben il 95%. Per le applicazioni che richiedono un basso livello di rumore, l'LTC3103 e l'LTC3104

possono funzionare in una modalità "forzata continua", per cui il componente utilizza una frequenza di commutazione di 1,2MHz. Il

dispositivo può anche effettuare la transizione automatica dalla frequenza fissa al funzionamento Burst Mode® per un'elevata efficienza su un ampissimo range di correnti di carico. I package DFN e MSOP richiedono solo pochi componenti esterni.

www.linear.com



LM48901, processore per riproduzione sonora di tipo cinematografico

Texas Instruments ha annunciato l'introduzione di un nuovo circuito integrato ideato con lo scopo di semplificare la progettazione e la programmazione di sistemi audio con migliori caratteristiche di spazializzazione del suono (per indurre nell'ascoltatore una percezione più precisa della posizione delle sorgenti sonore) per prodotti portatili che utilizzano più altoparlanti

tra cui laptop, tablet, sound bar e docking station audio. L'array spaziale quadruplo in classe D LM48901 è il capostipite di una nuova famiglia di integrati audio che utilizza l'elaborazione del suono distribuito accoppiata alla tecnologia ad array di altoparlanti per produrre una fruizione audio coinvolgente in applicazioni dove lo spazio è limitato. LM48901 e il relativo tool software

permettono di ottenere ottimi risultati con una programmazione estremamente semplice con effetti audio che trasformano un palcoscenico sonoro ridotto in una fruizione sonora coinvolgente che può essere resa disponibile sull'intera gamma di prodotti di un costruttore.

www.ti.com



FAN6756 con tecnologia mWSaver per bassi consumi

Con la costante contrazione dei limiti di consumo in standby da parte di dispositivi quali notebook, stampanti, monitor e televisori LCD, limiti che in molti casi scendono a livelli inferiori a 50mW, i progettisti necessitano di

una soluzione in grado di soddisfare requisiti progettuali di questo genere. Per aiutare i progettisti a superare le sfide associate ai bassi consumi in standby utilizzando un numero inferiore di componenti, Fairchild Semiconductor ha sviluppato il controller PWM FAN6756 con tecnologia mWSaver, che consente la riduzione della perdita di corrente in standby nel circuito esterno



senza necessità di circuiti di supporto addizionali.

La soluzione single-chip FAN6756 si riferisce a un controller PWM green-mode altamente integrato capace di ridurre significativamente le perdite in standby nei design di dispositivi SMPS (Switched-Mode Power Supply), eliminando fino a 15 componenti esterni.

www.fairchildsemi.com

Display Rugged+, l'ultima novità da Hitachi

Hitachi Europe Display Products Group (DPG) ha annunciato l'introduzione di un nuovo display 5.0" VGA TFT dalle alte performance, il TX13D04VM2CAA.

Questo display trasmissivo è l'ultima novità della famiglia di TFT Hitachi Rugged+, caratterizzato da una risoluzione VGA di 640 x 480 pixels e dalle ottime performance ottiche. Il nuovo display Rugged+ è stato ideato specificatamente per soddisfare le condizioni industriali più esigenti ed operare in condizioni ambientali estreme.

Mark Stephenson, Product Marketing Manager di Hitachi DPG, ha commentato: "Alcune applicazioni in ambienti difficili richiedono qualità e prestazioni consistenti a temperature estreme, shock meccanici e vibrazioni. L'introduzione del Rugged+ TX13D04 offre ai clienti un display 5.0" VGA robusto e resistente, dalle prestazioni ottiche eccezionali, ad alto contrasto e con una retroilluminazione LED

capace di offrire immagini chiare e luminose anche in ambienti ad alta luminosità".

Il display è caratterizzato da retroilluminazione LED a lun-



ga durata di 70k ore che offre una luminosità di 600cd/m², un contrasto di 500:1 ed eccellente uniformità d'immagine. Il display inoltre presenta un'interfaccia standard 40-pin CMOS, dimensioni compatte di 119.4mm (w) x 89.1mm (h) x 9.3mm (d) ed è stato progettato per operare in condizioni ambientali estreme anche a temperature comprese fra -30°C and +80°C. Il TX13D04 ha un peso di soli 110g.

www.hitachi-displays-eu.com/

RADIANT

A N D • S I L I C O N

56° MERCATO DEL CONSUMER ELECTRONICS

28 - 29 GENNAIO 2012

sab. 9.00-18.00 - dom. 9.00-17.00



PARCO ESPOSIZIONI NOVEGRO

MILANO/LINATE AEROPORTO ✈



Con il patrocinio di
Città di Segrate
Assessorato alla Cultura

www.parcoesposizioninovegro.it

Strenne natalizie a LED

di BORIS LANDONI

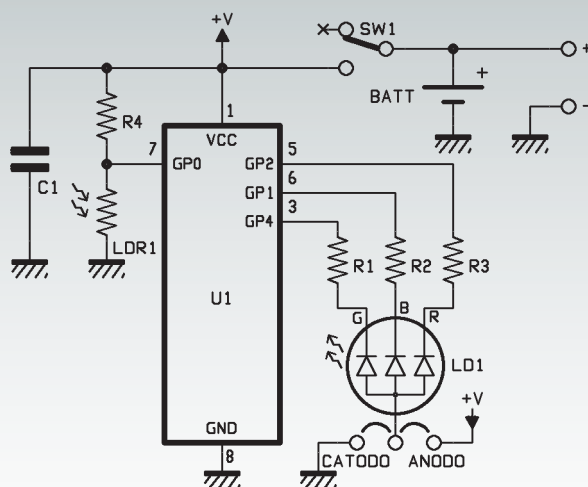
Pochi componenti, qualche lastra di plexiglass e la nostra CNC sono gli ingredienti per realizzare magnifiche figure luminose con cui decorare l'albero di Natale o la nostra casa in queste feste di fine d'anno. Vediamo come farle con un PIC ed una scheda basata su Arduino.



Con i LED è stato possibile fare cose che con le tradizionali fonti di luce artificiale non era permesso; tra queste, un posto di rilievo lo meritano le insegne e le figure luminose ottenute proiettando la luce dei diodi (possibilmente quelli ad alta efficienza, che presentano un ristretto angolo di irraggiamento) trasversalmente in una lastra di plexiglass o policarbonato trasparente su cui sono state scavate le lettere o i disegni da rappresentare. Questa tecnica coniuga il basso consumo e la durata dei LED con le proprietà della luce di fermarsi e percorrere

i solchi fatti nei materiali trasparenti: infatti se puntiamo un diodo lateralmente, ossia parallelamente alla superficie di una lastra di vetro o plastica trasparente, la sua luce attraversa il materiale e frontalmente non si vede; al più, appare colorata la sezione in uscita (se mal levigata) nella direzione della luce. Incidendo dei solchi sulla lastra, la luce si ferma in parte in essi, percorrendone le pareti e determinando un disegno luminoso la cui traccia è quella delle cave. Ecco, questa è la tecnica su cui si basano le moderne insegne a LED e che vi propo-

[schema elettrico
DRIVER con PIC]



niamo di sperimentare realizzando degli interessanti e suggestivi oggetti luminosi da usare come decorazioni natalizie di casa, della vetrina di un negozio o di quel che preferite.

Ogni oggetto si compone di un circuito elettronico che fa accendere un LED RGB in modo da comporre varie tonalità di colore e di una "meccanica", se così possiamo chiamarla, formata da un oggetto in plexiglass cui darete la foggia che più vi aggrada e su cui inciderete la figura che dovrà apparire luminosa.

Già, ma come possiamo lavorare

il plexiglass? Semplice, con una macchina CNC; naturalmente non una industriale, che costerebbe una fortuna, ma con quella commercializzata dalla ditta Futura Elettronica e che certo avrete visto in qualcuna delle pagine pubblicitarie di Elettronica In: si tratta di una macchina economica per lavorazioni di piccole dimensioni, particolarmente robusta e dotata di carrello portautensile con movimento a tre assi, operato da altrettanti motori passo-passo gestiti da una scheda a microcontrollore dotata di bridge ad H per il controllo di

potenza. Questo piccolo gioiello si comanda da PC mediante la porta parallela Centronics, il che potrebbe creare qualche problema a chi ha computer di ultima generazione, in special modo notebook, nei quali ormai parallela e seriale sono in disuso; quindi verificate di avere la parallela. Bene, dopo questo preambolo passiamo ai circuiti: parliamo al plurale perché in realtà vi proponiamo due soluzioni, che sono un semplicissimo driver a singolo LED RGB realizzato con un microcontrollore Microchip ed un circuito a più LED, facilmente configurabile perché basato su architettura Arduino e quindi su un micro ATmega.

IL CIRCUITO CON IL PIC

Ma procediamo con ordine ed occupiamoci prima del semplice circuito che fa accendere il LED RGB usato per produrre la luce che ci renderà visibili i disegni dei nostri gadget; il relativo schema è proposto nel riquadro *Schema elettrico DRIVER con PIC*. Come vedete, il circuito è semplicissimo e consta praticamente di un microcontrollore Microchip PIC12F675, basato su architettura RISC ad 8 bit e programmabile in PICBasic mediante istruzioni in word da 14 bit; dotato di Flash e incapsulato in un contenitore plastico DIP a 4+4 pin, questo componente fa proprio al caso nostro, perché può far girare il semplice firmware che realizza le combinazioni luminose, può pilotare direttamente dei LED con i propri I/O in modo sia source che sink, e funziona perfettamente anche con 3 volt, il che ci consente di alimentare il circuito con una pila al litio a bottone di tipo CR2032, molto comune perché prende posto nelle mainboard dei computer (dove serve a

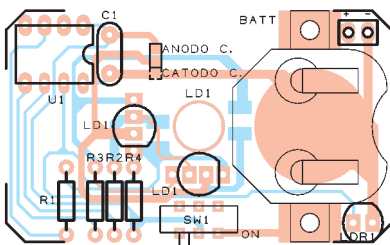
[piano di montaggio **DRIVER con PIC**]

Elenco Componenti:

R1: 47 ohm
R2: 47 ohm
R3: 82 ohm
R4: 10 kohm
C1: 100 nF multistrato
U1: PIC12F675-I/P (MF973)
LD1: LED 5 mm RGB Anodo/Catodo comune
LDR1: Fotorisistenza 2÷20 kohm
SW1: Deviatore a slitta

Varie:

- Zoccolo 4+4
- Porta batteria CR2032 da CS
- Strip maschio 2 poli
- Circuito stampato



mantenere il setup e l'orologio di macchina) nei moduli RTC, negli orologi LCD, nelle radiosveglie, in alcuni lettori MP3 ecc.

Una volta alimentato il circuito, dopo il power-on-reset il PIC inizializza le proprie linee di I/O assegnando GP1, GP2, GP4 come uscite dedicate al comando delle giunzioni, rispettivamente, blu, rossa e verde del LED tricolore LD1; GP0 è invece inizializzata come ingresso assegnato all'A/D converter (AN0) e ad essa abbiamo collegato un partitore resistivo nel quale la resistenza verso massa è in realtà un fotoresistore. Questa soluzione ci permette di far accendere il circuito solamente se la luce nell'ambiente è abbastanza debole (o se è buio...) da lasciar intravedere la colorazione della figura scavata nella plastica; se invece l'ambiente è troppo illuminato, il circuito si spegne, così da non consumare inutilmente la batteria.

Le tre uscite del microcontrollore pilotano direttamente il LED RGB in PWM; la gestione del PWM viene fatta via software e non tramite PWM hardware, in quanto il PIC12F675 non ha al proprio interno il modulo PWM, come i suoi "fratelli maggiori". Notate che nel circuito possiamo utilizzare LED RGB sia ad anodo comune, sia a catodo comune; il firmware è però stato previsto per l'anodo comune, ma adattarlo è semplicissimo.

All'accensione del circuito, il firmware pone a livello logico alto le linee che comandano le giunzioni del LED, quindi porta a zero GP2 per un istante facendo emettere un impulso a quella rossa, così da indicare che il dispositivo è ok. Subito dopo inizia il fading, ossia la dissolvenza dei colori, ottenuta modulando in PWM progressivamente i canali delle singole giunzioni del

Listato 1

```
SYMBOL G      =GPIO.4
SYMBOL B      =GPIO.1
SYMBOL R      =GPIO.2
SYMBOL AN     =GPIO.0      'INGRESSO AD

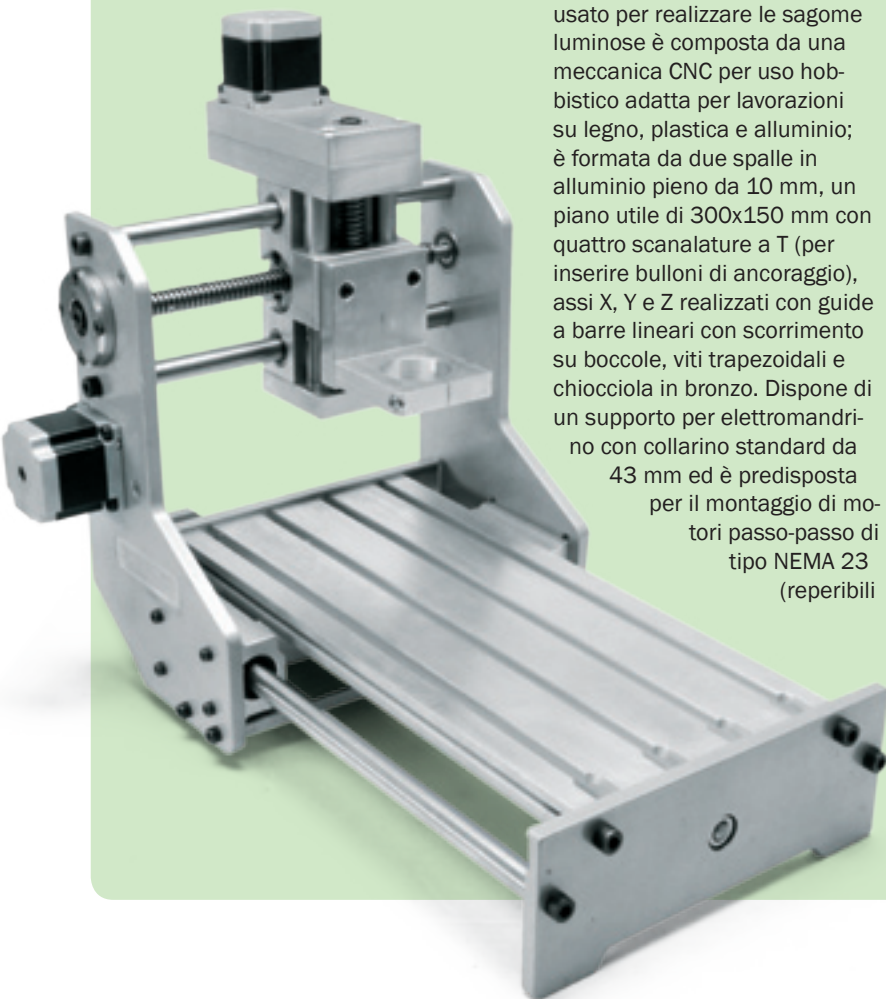
TMP          VAR WORD
TMP1         VAR BYTE
TMP2         VAR BYTE
TMP3         VAR WORD
PAUSA        VAR WORD
LUCE         VAR WORD

TMP1 = 1500 'LUNGHEZZA DEL CICLO
TMP2 = 80   'LUNGHEZZA SINGOLO STEP
TMP3 = TMP1*TMP2 'TMP1*TMP2
PAUSA=200

HIGH R
HIGH G
HIGH B
PAUSE 1000
LOW R
PAUSE 1000
HIGH R
PAUSE 1000

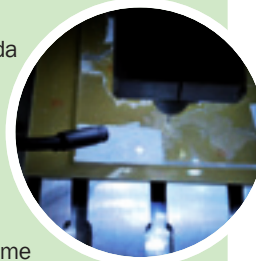
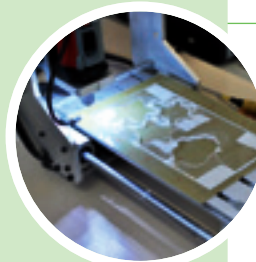
MAIN:
  ADCIN 0,LUCE
  IF LUCE > 100 THEN
    LOW R
    HIGH G
    HIGH B
    PAUSE PAUSA
    FOR TMP=0 TO TMP1
      LOW g
      HIGH r
      PAUSEUS TMP*TMP2
      HIGH g
      LOW r
      PAUSEUS TMP3-(TMP*TMP2)
    NEXT TMP
    LOW G
    HIGH B
    HIGH R
    PAUSE PAUSA
    FOR TMP=0 TO TMP1
      LOW b
      HIGH g
      PAUSEUS TMP*TMP2
      HIGH b
      LOW g
      PAUSEUS TMP3-(TMP*TMP2)
    NEXT TMP
    LOW B
    HIGH R
    HIGH G
    PAUSE PAUSA
    FOR TMP=0 TO TMP1
      LOW r
      HIGH b
      PAUSEUS TMP*TMP2
      HIGH r
      LOW b
      PAUSEUS TMP3-(TMP*TMP2)
    NEXT TMP
  ELSE
    HIGH G
    HIGH R
    HIGH B
    SLEEP 60
  ENDIF
  GOTO MAIN
```


Una CNC per tutti



La macchina che abbiamo usato per realizzare le sagome luminose è composta da una meccanica CNC per uso hobbistico adatta per lavorazioni su legno, plastica e alluminio; è formata da due spalle in alluminio pieno da 10 mm, un piano utile di 300x150 mm con quattro scanalature a T (per inserire bulloni di ancoraggio), assi X, Y e Z realizzati con guide a barre lineari con scorrimento su boccole, viti trapezoidali e chiocciola in bronzo. Dispone di un supporto per elettromandrino con collarino standard da 43 mm ed è predisposta per il montaggio di motori passo-passo di tipo NEMA 23 (reperibili

da Futura Elettronica, con codice STEPMOT02). Questa struttura deve essere gestita da una scheda di controllo che interfaccia il computer ai motori passo-passo che comandano il movimento della testa con l'elettromandrino; l'ideale abbinamento è con la scheda di controllo Futura Elettronica cod. SCHEDA4ASSI, provvista di interfaccia parallela standard, quindi idonea ad essere collegata ai vecchi PC. Per gestire la CNC occorre un software come ad esempio MACH3 (si trova su www.machsupport.com anche in versione demo, limitata a 500 linee, ma si può acquistare in versione non limitata) che comanda la macchina per farle lavorare il pezzo definito nel file di lavorazione (ad esempio in gcode) sviluppato con un CAM; il software CAM da noi utilizzato per disegnare le sagome e creare i file di lavorazione è ArtCAM.



LED tricolore. In pratica si parte dalla rossa, che viene accesa gradualmente, quindi si somma la verde e la rossa va spegnendosi, poi cresce la luminosità della blu, man mano che quella della verde si spegne. Dopo ogni ciclo di fading, il microcontrollore verifica l'ingresso analogico AN0 cui è collegata la fotoresistenza, in modo che se nel frattempo l'ambiente si è illuminato troppo il firmware lo manda in sleep-mode per un minuto. Dopo un minuto il PIC si risveglia e testa nuovamente la fotoresistenza, per vedere se la luce nell'ambiente si è abbassata, allorché ricomincia il ciclo; diversamente, si pone di nuovo in sleep. L'intero circuito si alimenta con una pila o batteria capace

di fornire almeno 3 volt, meglio se 3,6 (va quindi bene una cella agli ioni di litio) o 4,5 volt. Nel nostro caso impieghiamo una pila CR2032, che da nuova fornisce circa 3,2 volt. Notate che, siccome non tutti i LED tricolore hanno le stesse tensioni di soglia (i valori differiscono leggermente da un costruttore all'altro) sarebbe consigliabile usare il componente che abbiamo usato noi, ossia il 1418-540R2GBC_ANODO della Futura Elettronica (lo trovate su www.futurashop.it). L'interruttore SW1 consente di accendere e spegnere il circuito, in modo da non sprecare la pila.

REALIZZAZIONE DEL CIRCUITO CON PIC

Per quanto riguarda l'elettronica, è tutta montata su uno stampato

che dovete realizzare per fotoincisione, facendo attenzione perché si tratta di un doppia faccia; potete scaricare le relative tracce dal nostro sito www.elettronica.in. Incisa e forata la basetta, disponete dapprima le resistenze ed il piccolo interruttore a slitta, quindi saldate l'alloggiamento per la pila a bottone, stagnandone bene le alette laterali; notate che il portapila realizza il contatto positivo, mentre il negativo tocca la piazzola ramata sotto ad esso, la quale conduce alla pista di massa. Sistemato questo, montate lo zoccolo per il microcontrollore e la fotoresistenza, che dovete far spuntare dal contenitore plastico in cui racchiuderete il circuito; il contenitore farà anche da sostegno meccanico per l'oggetto lumino-

so. Il LED RGB si può montare in tre modi: passante attraverso il foro, come mostrano le foto del prototipo visibili in queste pagine (in questo caso, se avete fatto realizzare lo stampato ad un'azienda specializzata ed è a fori metallizzati, dovete fare attenzione a non appoggiare i terminali alla metallizzazione del foro passante), laterale rivolto al lato corto, laterale rivolto al lato lungo della basetta.

Ricordate che il circuito accetta sia i LED ad anodo comune, sia quelli a catodo comune: basta impostare opportunamente il ponticello, ovvero cortocircuitare il terminale comune con la massa se usate il tipo a catodo comune (jumper CATODO) o con il positivo (jumper ANODO) se utilizzate il diodo ad anodo comune; resta inteso che il firmware da noi progettato e che trovate nel **Listato 1**, è adatto a controllare LED ad anodo comune.

Comunque non c'è alcun problema, perché adattare il firmware è piuttosto semplice: basta scambiare lo zero logico con l'1 e viceversa; per esempio, nel blocco seguente:

```
HIGH R
HIGH G
HIGH B
PAUSE 1000
LOW R
PAUSE 1000
HIGH R
PAUSE 1000
```

che segue l'inizializzazione delle linee GP1, GP2 e GP4 e pone inizialmente (dopo l'accensione o il risveglio dallo sleep) le linee di comando delle giunzioni del LED RGB ad 1 logico per spegnere il diodo luminoso, quindi far emettere l'impulso alla



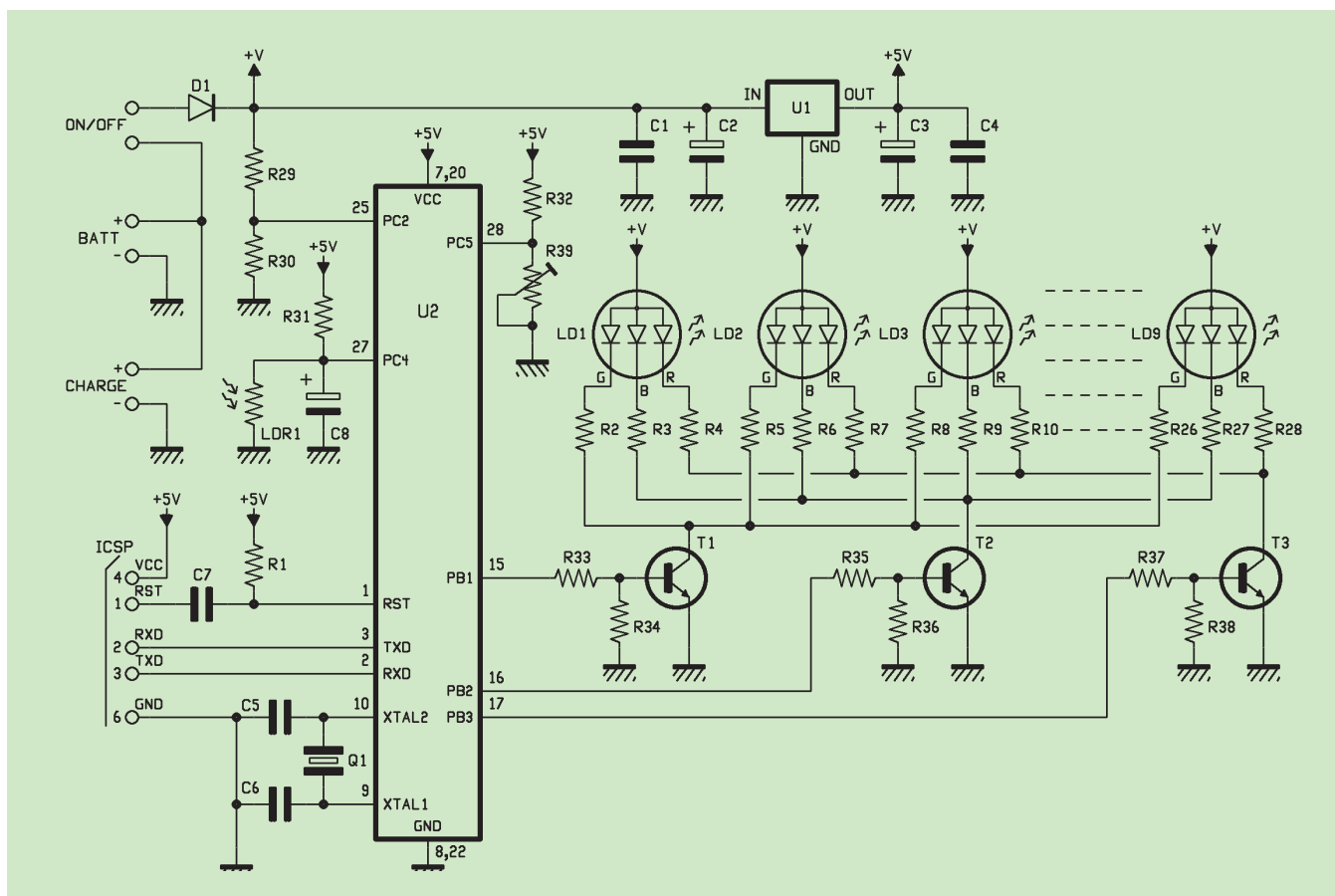
sola giunzione rossa, dove c'è HIGH dovete scrivere LOW e viceversa. Insomma, dovete invertire tutti gli HIGH con LOW. Lo stesso dicasi per il MAIN program, che imposta l'esecuzione del ciclo di fading (ovvero dell'effetto dissolvenza) dei tre colori rosso, verde e blu. Completato il montaggio, inserite il microcontrollore (che avrete avuto cura di programmare caricandovi il firmware ottenuto compilando il sorgente che vi diamo, con l'IDE della Microchip) nel suo zoccolo, orientandolo come mostrato nel piano di montaggio. Controllato il tutto potete inserire la pila a bottone

nell'apposito alloggiamento, badando che il polo + guardi verso l'alto.

IL DRIVER BASATO SU ARDUINO

Bene, ora possiamo descrivere lo schema (*Schema elettrico DRIVER con Arduino*) della versione più grande del driver per figure luminose; si tratta di un circuito basato





sul chip utilizzato in Arduino Duemilanove, UNO ecc. Parliamo dell'ATmega328; se in esso è caricato il bootloader, possiamo realizzare un sistema open source facilmente modificabile, in quanto ad esso si possono applicare le librerie e gli sketch scritti per Arduino. Essendoci il bootloader, la programmazione

avviene da PC tramite un convertitore USB/TTL (ad esempio il 7300-FTDI5V della Futura Elettronica).

Il funzionamento del circuito è molto simile a quello del modello più piccolo: anche qui troviamo la fotoresistenza che permette di verificare la quantità di luce presente nell'ambiente, solo che

in questo caso è possibile aggiustare la sensibilità del circuito mediante un trimmer.

Confrontando il valore letto dal convertitore A/D collegato alla fotoresistenza con quello rilevato dal pin collegato all'R39, il micro decide se avviare la sequenza di fading, ovvero se accendere o meno i LED.

RM ELETTRONICA

Forniture per hobbisti, scuole e industria

Distributore autorizzato:

**FUTURA
ELETTRONICA**

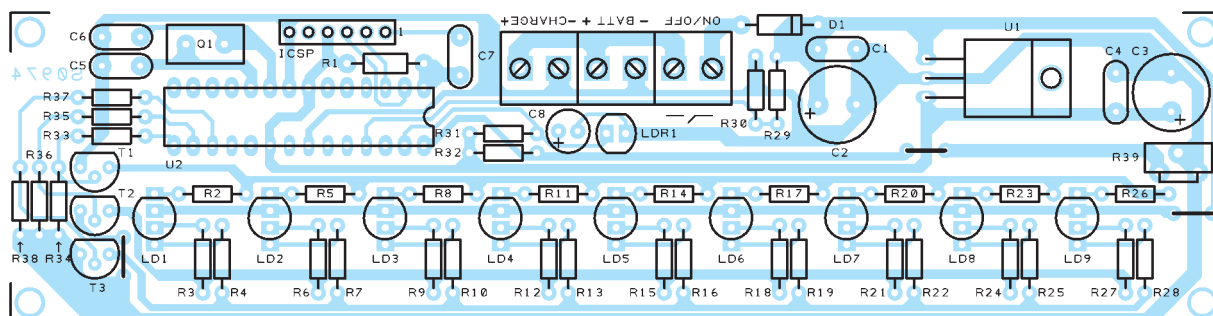
**ELSE
KIT**

ELETTRONICA

Il tuo punto
di riferimento
per l'elettronica
a ROMA

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

[piano di montaggio **DRIVER con ARDUINO**]



Elenco Componenti:

R1: 10 kohm	R22: 1 kohm	R39: Trimmer 4,7 kohm MV	Q1: Quarzo 16 MHz
R2, R3: 820 ohm	R23, R24: 820 ohm	C1: 100 nF multistrato	LDR1: Fotoresistenza 2÷20 kohm
R4, R7: 1 kohm	R25: 1 kohm	C2, C3: 470 µF 25 VL elettrolitico	
R5, R6: 820 ohm	R26, R27: 820 ohm	C4: 100 nF multistrato	Varie:
R8: 820 ohm	R28: 1 kohm	C5, C6: 15 pF ceramico	- Morsetto 2 poli (2 pz.)
R9: 820 ohm	R29: 10 kohm	C7: 100 nF multistrato	- Zoccolo 14+14
R10: 1 kohm	R30: 4,7 kohm	C8: 100 µF 25 VL elettrolitico	- Batteria 12V/2A
R11, R12: 820 ohm	R31: 10 kohm	T1÷T3: BC547	- Strip maschio 6 poli
R13: 1 kohm	R32, R33: 4,7 kohm	LD1÷LD9: LED 5 mm RGB anodo comune	- Plug alimentazione da pannello
R14, R15: 820 ohm	R34: 10 kohm		- Deviatore a levetta da pannello
R16: 1 kohm	R35: 4,7 kohm		- Circuito stampato
R17, R18: 820 ohm	R36: 10 kohm		
R19: 1 kohm	R37: 4,7 kohm		
R20, R21: 820 ohm	R38: 10 kohm		
		U1: 7805	
		U2: ATMEGA328P-PU (MF974)	

I diodi luminosi sono degli RGB anche in questo circuito e vengono pilotati tramite dei transistor; tale scelta nasce dall'esigenza di pilotare, con una sola linea del microcontrollore, più LED, allo scopo di realizzare figure luminose di grandi dimensioni. Come vedete, usiamo una linea dell'ATmega per ciascuno dei colori fondamentali rosso, verde e blu; ciascun transistor pilota in parallelo le relative giunzioni dei LED RGB, perciò è chiaro che tutti i diodi faranno lo stesso

gioco di luci. Naturalmente non è obbligatorio montare tutti i diodi luminosi previsti nel circuito: ognuno di voi potrà montare quelli che servono ad ottenere un buon effetto visivo in base alle dimensioni della sagoma in plexiglass. Notate che sullo stampato sono previste tre morsettiere da due contatti: una per collegare l'interruttore di accensione (ON/OFF) una seconda (BATT) per applicare al circuito l'alimentazione (ottenuta da un alimentatore da

rete o da una batteria da 12 V ed 1 Ah, per esempio) ed una terza (CHARGE) per un eventuale caricabatterie per batterie al piombo o un piccolo pannello solare (tipo SOL6N della Futura Elettronica) da 12 V che integra già il circuito di ricarica. Queste soluzioni consentono di utilizzare a lungo ed autonomamente le figure luminose anche dove manca la rete elettrica, contando su batterie al piombo-gel (o ad acido) ricaricabili con un caricatore da rete o da un pannello

Listato 2

```

//*****
//* Name : RGB controller for common anode led *
//* Author : Landoni Boris *
//* www.open-electronics.org *
//* blog.elettronica.in.it *
//* www.futurashop.it *
//*****

int red = 9; // RED LED connected to PWM pin 3
int green = 10; // GREEN LED connected to PWM pin 5
int blue = 11; // BLUE LED connected to PWM pin 6
int photo = A4; // BLUE LED connected to PWM pin 6
int trim = A5; // BLUE LED connected to PWM pin 6
int volt = A2; // BLUE LED connected to PWM pin 6
int r=50; int g=100; int b=150;
int rup; int gup; int bup;
int fader=1;
int inc=10;
void setup()
{
    Serial.begin(9600);
    Serial.println("Serial READY");
    rgb(r, g, b);
    r = random(0,255);
    g = random(0,255);
    b = random(0,255);
}

void loop() {

    Serial.print("trim ");
    Serial.println(analogRead(trim)*2);

    Serial.print("photo ");
    Serial.println(analogRead(photo));

    Serial.print("volt ");
    Serial.println(analogRead(volt));
    if (analogRead(volt)<600){
        Serial.println("low battery");
        rgb(0, 0, 0);
        delay(500);
        rgb(255, 255, 255);
    }

    if ((analogRead(trim)*2)>analogRead(photo)){
        Serial.println("trim > photo - off");
        rgb(0, 0, 0);
        fader=0;
    }
    else
    {
        if (fader==0){
            r = random(0,255);
            g = random(0,255);
            b = random(0,255);
        }
        fader=1;
    }
    //delay(2000);

    if (fader==1){
        funcfader();
    }
}

void funcfader(){
    Serial.println("fader");
    if (rup==1){r+=1;}
    else{r-=1;}
    if (r>=255){rup=0;}
    if (r<=0){rup=1;}
}

```

fotovoltaico che di giorno immagazzinerà nell'accumulatore la corrente che poi alimenterà i LED quando sarà buio. A tal proposito, un ingresso analogico del microcontrollore è dedicato al controllo della tensione di batteria: quando questa scende al di sotto di 10 V, il circuito farà emettere dei lampeggi ai LED, per avvisare che l'energia sta per terminare. Questo circuito è evidentemente stato pensato per essere utilizzato all'esterno (a patto che sia correttamente isolato) dove trarrà l'alimentazione da un pannello solare e si accenderà e spegnerà autonomamente, grazie al sensore di luce ambiente, che farà illuminare le figure la sera, per lasciarle spegnere al mattino. Insomma, è una soluzione per addobbare anche il giardino o il cortile di casa.

Chiudiamo la descrizione del circuito con la sezione di alimentazione a 5 volt, realizzata da un classico 7805, siglato U1, impiegato nella sua configurazione canonica; l'integrato prende l'alimentazione principale dal catodo del diodo D1, montato per proteggere il circuito dall'inversione di polarità. I 5 volt ricavati, vengono utilizzati per alimentare il microcontrollore. Quanto ai diodi luminosi, vengono alimentati dalla tensione a valle del D1, che è pari a quella applicata a + e - BATT o + e - CHARGE, diminuita di circa 0,7 volt.

Anche per il circuito basato su Arduino.

REALIZZAZIONE DEL DRIVER CON ATMEGA

La costruzione del circuito basato su Arduino richiede le stesse attenzioni richieste dal driver con PIC: una volta preparato il circuito stampato, disponetevi i componenti in ordine di altezza e prestando attenzione alla

(Continua)

Guarda su YouTube come si realizzano le strenne natalizie!



Realizzare le sagome in plexiglass con la nostra CNC è molto semplice e veloce come dimostra il video che abbiamo realizzato e postato su YouTube all'indirizzo: <http://www.youtube.com/watch?v=IwAQo2wGFlc>.

polarità indicata per i diodi ed i transistor. Per l'ATmega328 prevedete uno zoccolo dip in cui inserirlo ed il pin-strip ICSP per la programmazione in-circuit, che consiste nel caricamento dello sketch. I LED vanno orientati come mostra il piano di montaggio e disposti in base al modo in cui andrete a collocare la sagoma in plexiglass.

Quanto al microcontrollore, ricordate che affinché vi si possa caricare lo sketch come previsto dai sistemi Arduino, deve già essere dotato di bootloader; questo semplicissimo firmware serve ad avviare l'acquisizione dal computer quando la linea di reset (che nei convertitori USB/seriale avanzati è collegata al DTR della RS232) viene posta per un istante a zero logico. Il bootloader deve essere caricato con l'apposito programmatore; in alternativa, bisogna acquistare un ATmega328 con precaricato il bootloader.

Lo sketch da caricare nel micro è illustrato nel **Listato 2**.

Passiamo all'assemblaggio, dicendo che il circuito dovrà essere inserito in un contenitore in plastica, possibilmente a tenuta stagna (per esempio una scatola

per impianti elettrici di quelle da esterno, magari di grado IP65); anche in questo caso serve realizzare le cave per affacciare la fotoresistenza sull'esterno e per inserire la sagoma e renderla stabile. Al limite, un rinforzo si può fare facilmente con delle squadrette in alluminio avvitate sia nella scatola che nella base della sagoma.

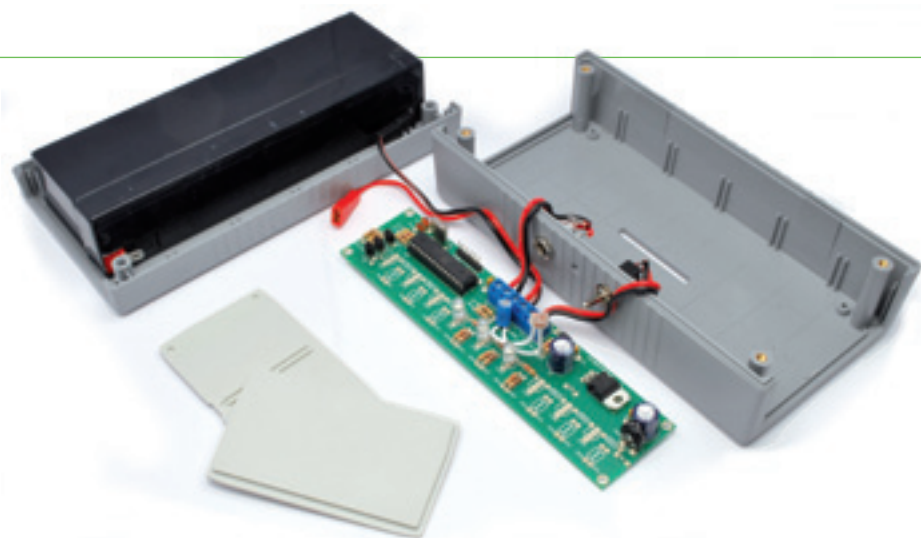
Per sigillare le aperture potete procedere in vari modi; il nostro consiglio è di incollare con del silicone un pezzo di plexiglass nel foro davanti al fotoresistore, in modo da non far entrare l'acqua, mentre per i fori dove entrano le estremità della sagoma che si appoggiano sui LED potete usare ancora del silicone sigillante. Per l'entrata dei cavi provenienti dal pannello fotovoltaico o dal caricabatteria, dovete usare un passacavo a tenuta d'acqua.

Tutte queste accortezze non servono se pensate di collocare le sagome luminose al riparo dalla pioggia e dalla neve.

COSTRUIAMO LE SAGOME

Sistemato il circuito elettronico bisogna preparare il visualizzatore che mostrerà l'immagine luminosa; come detto, i materiali eletti per questa applicazione sono polycarbonato trasparente (o anche metacrilato) oppure plexiglass, cui potete dare la sagoma che preferite.

Poi bisogna incidere, abbastanza in profondità (metà o due terzi dello spessore della lastra) il disegno o la scritta che volete appaia luminosa. Questo lavoro riesce perfettamente con una fresa montata su un elettromandrino fissato alla testa di una macchina a controllo numerico di quelle hobbistiche, ma anche, in modo più spartano, con un trapano a mano (tipo il DRE-



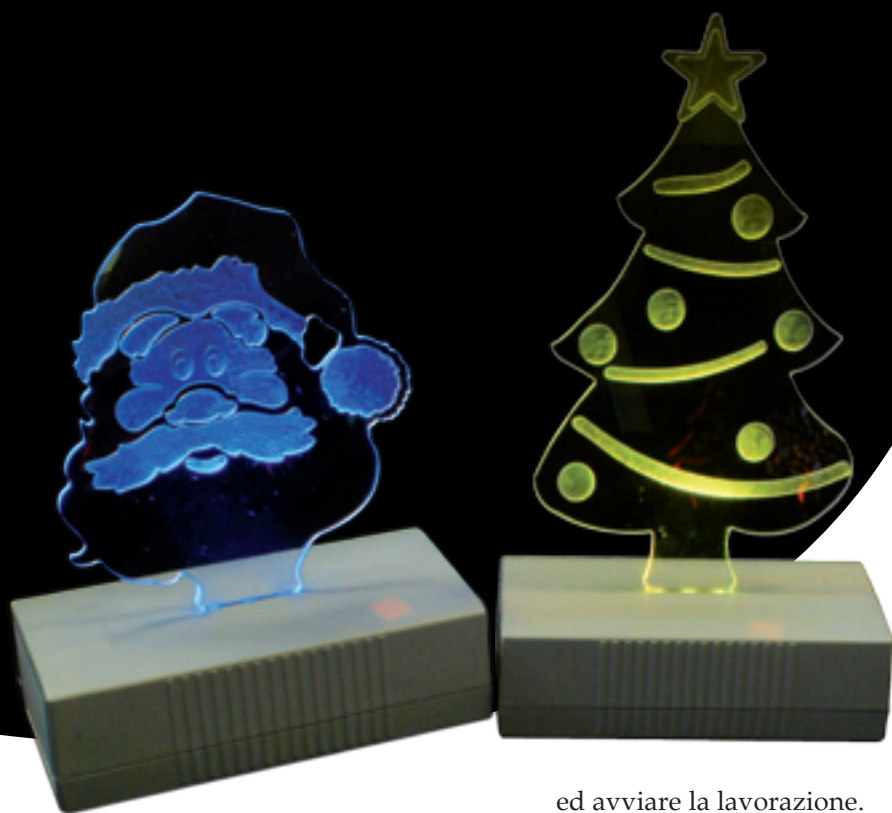
Listato 2 - segue

```
if (gup==1){g+=1;}
else{g-=1;}
if (g>=255){gup=0;}
if (g<=0){gup=1;}

if (bup==1){b+=1;}
else{b-=1;}
if (b>=255){bup=0;}
if (b<=0){bup=1;}

delay(inc*2);
rgb(r, g, b);
}

void rgb(int r, int g, int b)
{
  Serial.print("RGB: ");
  Serial.print(r);
  Serial.print(" ");
  Serial.print(g);
  Serial.print(" ");
  Serial.print(b);
  Serial.println(b);
  if (r>255) r=255;
  if (g>255) g=255;
  if (b>255) b=255;
  if (r<0) r=0;
  if (g<0) g=0;
  if (b<0) b=0;
  analogWrite(red, r);
  analogWrite(green, g);
  analogWrite(blue, b);
}
```

MEL) o elettromandrino più piccolo nel quale è montata una fresa conica. Come macchina CNC vi consigliamo quella della Futura Elettronica (cod. MEC-CANICACNC) equipaggiata con l'apposita elettronica e gestita da un Personal Computer; si tratta di un prodotto molto valido, dal prezzo abbordabile e che potrà servire per realizzare moltissimi oggetti, lavorando plastiche, metalli, legno ecc. Questa macchina può essere gestita con software CAM, che permette di disegnare la lavorazione voluta e pilotare macchine CNC in modo da ottenerla; nel nostro caso, per ottenere i pezzi che vedete in queste pagine abbiamo adottato ArtCAM, un ottimo software che genera file di tipo *gcode*, ma volendo si può cercare sul web qualcosa di più economico o magari di freeware. Per effettuare materialmente la lavorazione bisogna gestire la CNC con un apposito software, come MACH3; in esso occorre importare il file *gcode* (va bene qualsiasi formato) verificare che non vi siano righe incompatibili (nel qual caso basta cancellarle o modificarle con l'editor "gedit")

ed avviare la lavorazione.

Le sagome vanno prima incise e poi tagliate dalla lastra da cui sono derivate. Nel caso della lavorazione su macchina a controllo numerico si fa il contrario, perché l'utensile deve prima scavare il disegno, poi staccare il pezzo dalla lastra, altrimenti durante l'incisione la sagoma si muoverebbe. Sempre per evitare che il materiale si muova durante le lavorazioni, conviene fissarlo sfruttando le viti e le scanalature presenti nel piano della CNC; se temete di graffiare il plexiglass o polycarbonato che usate, potete ricorrere ad un artificio: appoggiare sul piano della macchina un pezzo di vetronite o plastica (ma anche legno compensato) che avvitarete con le viti in dotazione dopo averlo opportunamente forato nella periferia. Poi, su questo falso piano applicate qualche pezzo di nastro biadesivo un po' spesso, in modo da applicarvi il plexiglass, che così rimarrà ben fermo durante la lavorazione senza entrare in contatto con la superficie di appoggio e senza, perciò graffiarsi. I programmi di CAM come ArtCAM permettono di disegnare sagome a piacere, quindi di simulare a video

l'aspetto finale: ciò permette di sapere da subito come verrà la lavorazione ed evitare di buttare via i pezzi alla fine perché non vi soddisfano. Ricordate che ogni sagoma deve riportare una linguetta con una cava grande come la testa del LED tricolore: ciò perché a montaggio ultimato andrà fatta appoggiare proprio sul triplo diodo, in modo che la luce emessa da questo possa essere accoppiata alla plastica con la minor perdita possibile. Per sostenere la sagoma, bisogna montare il circuito stampato in un contenitore plastico, il cui coperchio dovrà riportare una finestrella grande come la sezione orizzontale della linguetta; la finestra potrà essere realizzata con la stessa macchina CNC e dovrà far entrare "a filo" la linguetta alla base della sagoma. ■



per il MATERIALE

Tutti i componenti utilizzati nei progetti natalizi sono facilmente reperibili in commercio; inoltre il firmware per la programmazione dei microcontrollori, i master dei PCB ed i files G-Code per la realizzazione delle sagome sono scaricabili gratuitamente dal sito della rivista (www.elettronica.in.it). I disegni delle sagome sono stati creati utilizzando il programma ArtCam, mentre la CNC viene gestita mediante Mach3.

Ricordiamo che il pantografo CNC da noi utilizzato è distribuito da Futura Elettronica. La sola meccanica (cod. MECCANICACNC) costa 648,00 Euro, la scheda di controllo a 4 assi (cod. SCHEDA4ASSI) costa 210,00 Euro mentre ciascun motore passo-passo (ne servono 3 esemplari) costa 44,00 Euro (cod. STEPMOT02). 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>

PROIETTORI LASER

disegnatori di luce



196.00

cod. VDP1401RGLD5

Proiettore laser "Vega" rosso/verde • 140mW

Modalità di funzionamento DMX (5 canali), auto, master/slave o ad attivazione musicale • Alimentazione 230Vac • Potenza laser DPSSL verde 40mW (532nm), rosso 100mW (650nm) • Classe 3B • Dimensioni 195 x 190 x 110mm.

Mini laser rosso • 100 mW

Modalità di funzionamento automatico (16 patterns), manuale o ad attivazione musicale • Alimentazione 220 Vac • Potenza laser 100mW (650 nm) • Classe 3B • Dimensioni 155 x 75 x 46mm.

69.00

cod. VDL301RL



NEW

209.00

cod. VDP1401RGLD4

Proiettore laser rosso/verde • 140mW

È in grado di proiettare oltre 100 fasci e può funzionare in modalità DMX (4 canali), automatica, master/slave o ad attivazione musicale. Alimentazione 230Vac • Consumo 50W • Potenza laser DPSS verde 40mW (532nm), rosso 100mW (650nm) • Dimensioni 285 x 120 x 80mm.



Proiettore laser verde • 100mW

Modalità di funzionamento DMX (7 canali), automatico, master/slave o attivazione musicale. Alimentazione 220Vac • Potenza laser 100mW (532nm) • Classe 3B • Dimensioni 300 x 280 x 130mm.

348.00

cod. VDP1001GLD7



Proiettore laser verde • 30mW

Compatto proiettore Laser verde con potenza 30mW (532nm). Funzionamento musicale o con programmi preimpostati • Alimentazione 220Vac • Consumo 30W • Angolo di scansione 20° • Dimensioni 180 x 183 x 80mm.

125.00

cod. VDL301GL



340.00

cod. VDL160LL

Laser "Sky Smooth" rosso/verde • 160mW

Crea bellissimi effetti luce grazie all'azione combinata di un LED blu e di due laser, uno verde e uno rosso. Strobo e velocità di rotazione regolabile per entrambi i laser • Luminosità del LED e velocità di rotazione regolabile • Potenza laser rosso 120mW (650nm), verde 40mW (532nm) • Alimentazione 230Vac • Consumo 15W • Dimensioni 260 x 200 x 240mm.



268.00

cod. VDP1601RGLD10



Proiettore laser rosso/verde/giallo • 160mW

È in grado di generare particolari effetti luminosi e può essere controllato tramite protocollo DMX (10 canali). Funzionamento stand-alone, a controllo musicale o con programmi preimpostati. Alimentazione 220Vac • Consumo 15W • Potenza laser verde 60mW (532nm), rosso 100mW (650nm) • Classe 3B • Dimensioni 340 x 210 x 90mm.

Specchi regolabili per effetti laser (4 pz)

- Supporto robusto completo di staffa di fissaggio
- Inclinazione regolabile finemente mediante viti posizionate sul retro;

- Dimensioni 100 x 100 x 90mm.

La confezione comprende 4 specchi.



158.00

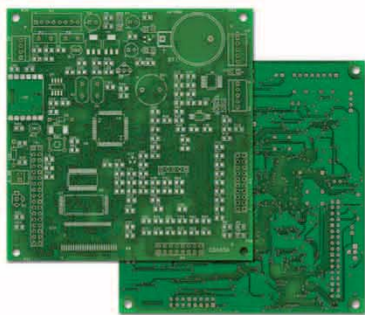
cod. VDLML



FUTURA ELETTRONICA

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

Disponibili presso i migliori negozi di elettronica o nel nostro punto vendita di Gallarate (VA). Ulteriori informazioni e acquisti on line su www.futurashop.it

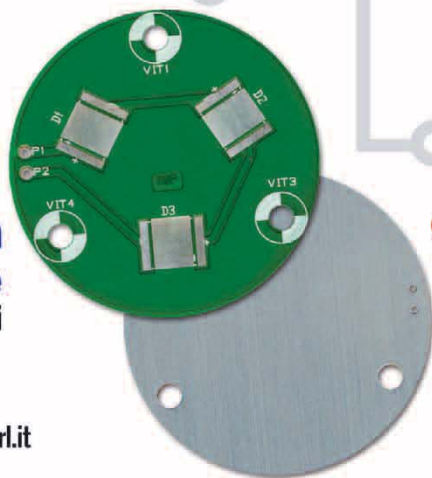


SINGOLA E DOPPIA FACCIA CON FORO METALLIZZATO; CON SOLDER E SERIGRAFIE PER UNO STAMPATO DI ALTA QUALITÀ O SOLO PISTE STAGNATE PER UN PROTOTIPO A BASSO COSTO

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

**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



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 STESSER 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



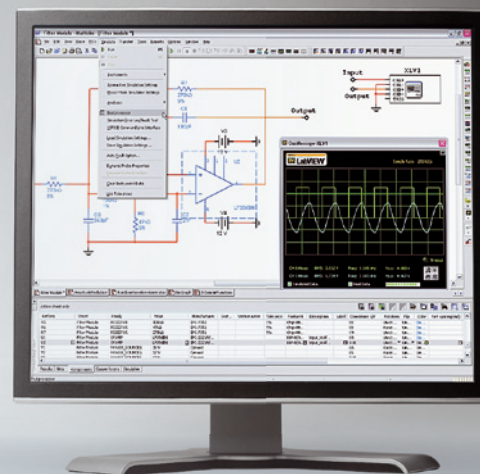
facilità

PER CHIARIMENTI, DETTAGLI
SULLE NOTE TECNICHE, ORDINI:

www.mdsrl.it
PREVENTIVO ANONIMO,
GRATUITO E IMMEDIATO

Conoscere e usare NI Multisim

Progettazione più semplice,
migliore simulazione.



Occupiamoci di altre due funzionalità di Multisim: MCU, che permette la simulazione di microcontrollori, ed i LabVIEW instruments, che consentono di utilizzare strumenti programmati in LabVIEW. Decima puntata.

di Filippo Spadaro

Delle tante preziose funzionalità di NI Multisim, in queste pagine ci occupiamo del modulo MCU, che permette la simulazione dei microcontrollori e risulta utile per lo sviluppo e il test di progetti e circuiti che si avvalgono di tali dispositivi, ormai diffusissimi. Prenderemo in considerazione anche i LabVIEW instruments, che rappresentano la possibilità, offerta da Multisim, di utilizzare al suo interno strumenti programmati in LabVIEW, il popolare linguaggio di programmazione di National Instruments. Iniziamo dunque affrontando l'MCU.

MULTISIM MCU

L'ambiente di sviluppo per microcontrollori (MCU) proposto da Multisim, permette di realizzare circuiti elettronici basati su microcontrollore, al fine di simulare in maniera realistica l'esecuzione di listati o di codice compilato. Il modulo MCU simula i microcontrollori Intel 8051 e 8052, Microchip PIC16F84 e PIC16F84A.

Usando l'**Editor** dell'ambiente MCU, è possibile scrivere il codice sorgente per il microcontrollore scelto direttamente all'interno di Multisim, com-

pilarlo mediante il comando **Build**, eseguirlo ed eventualmente accedere agli strumenti di **Debug** per svolgere il codice passo-passo a livello di istruzione, ottenendo altresì vedute della memoria e dei registri interni del MCU. Le funzionalità per il debug sono raggiungibili usando il menu **MCU** nella barra dei menu di Multisim.

Il modulo di simulazione Multisim MCU prevede la programmazione in assembly e in C, oltre a permettere l'importazione a bordo del MCU di codice macchina già compilato con gli strumenti software propri del microcontrollore che si sta usando. Sarà proprio questo, il metodo operativo che adotteremo nella trattazione di questa puntata del corso, dove adopereremo Multisim per testare codice macchina per il PIC16F84A. Di seguito faremo due esempi di utilizzo del modulo MCU; ulteriori esempi, sia per 805X che per PIC, sono disponibili in Multisim accedendo dal percorso **File>Open Samples>MCU Sample Circuits**.

Esempio 1: programmare l'accensione di un LED
Realizziamo un semplice file che illumina un LED collegato al pin RB0 del PIC. L'ambiente di pro-

Listato 1

```

; LED.ASM
; Accensione di un LED sulla porta RB0

    list p=16F84A, r=dec      ; Scelta uC e rappresentazione decimale dei numeri
    #include <p16F84a.inc>    ; File di definizioni relative al uC
    __config _RC_OSC & _WDT_OFF & _CP_OFF & _PWRTE_ON

                                ; Oscillatore interno
                                ; Watch dog timer          OFF
                                ; code protect             OFF
                                ; Power up timer ON

LED    EQU    0                ; Indica il pin cui è collegato il LED
ORG    0x00                   ; Inizio allocazione istruzioni in area programma

Start:
    bsf     STATUS,RP0        ; Seleziona il banco 1
    bcf     TRISB,LED         ; Dichiaro il pin RB0 come output
    bcf     STATUS,RP0        ; Torna al banco 0
    bsf     PORTB,LED         ; Accendi il LED sul pin RB0
    goto    $                 ; Attendi
end

```

grammazione dei microcontrollori Microchip è il software MPLAB (scaricabile gratuitamente dal sito www.microchip.com).

Creiamo un nuovo progetto e importiamo il file *Led.asm*, oppure scriviamolo direttamente su una nuova pagina dell'editor (**Listato 1**).

Compiliamo il progetto e, dopo che la compilazione è andata a buon fine e che abbiamo ottenuto il file di estensione HEX (contenente il codice in linguaggio macchina) possiamo riversare il codice esadecimale sul PIC, estrarlo dal programmatore collegato al PC ed inserirlo nel circuito in cui dovrà operare, schematizzato in **Fig. 1**. In alternativa, è possibile effettuare una simulazione di come lavorerebbe il PIC-16F84A, sfruttando il modulo MCU di Multisim, raggiungibile mediante il comando di menu **Place>Component**, scegliendo MCU tra i componenti sotto la voce **Group**, quindi impar-

sul microcontrollore (MCU) PIC16F84A il codice da noi scritto, inserendo anche i componenti circuitali necessari. Una volta scelto il PIC-16F84A tra i dispositivi simulati da Multisim, si apre un wizard in tre passi (illustrati in **Fig. 2**). Nello **step 1** indichiamo il percorso di lavoro, scegliendo, per comodità, la cartella di lavoro del progetto "Led" usata in ambiente MPLAB e scegliendo un nome per la cartella di lavoro in cui Multisim metterà i file temporanei.

Nello **step 2** carichiamo a bordo del PIC il file HEX in codice macchina e nominiamo il progetto come Led. Nello **step 3**, in questo caso compare la sola opzione **Create empty project**, che è necessario selezionare (anche se sembra già selezionata) altrimenti non viene attivato il pulsante **Finish**. Completato il MCU wizard, il PIC viene posto sullo schema elettrico ed è possibile inserire i rimanenti componenti.

Il circuito di simulazione (**Fig. 3**) è molto simile a quello reale: il PIC va alimentato a 5 V sul pin 14 (VDD), mentre il pin 5 (VSS) è posto a massa. Il pin 4 (MCLR) è il reset del PIC ed è normalmente mantenuto al potenziale VDD tramite una resistenza di pull-up, ovvero è posto a zero quando si desidera resettare il microcontrollore dall'esterno (si ricorda che il PIC avvia già all'accensione un reset interno, indipendente da questo pin). Nella nostra simulazione in Multisim verrà inserito, per comodità, un pulsante di reset. Al pin 6 (RB0) andrebbe collegato un LED verso massa (come in **Fig. 1**) tuttavia, siccome i simulatori di MCU presenti in Multisim forniscono sui pin in uscita solo dei livelli logici di tensione (alto/basso) ma non la corrente, il circuito virtuale non funzionerebbe. Del resto,

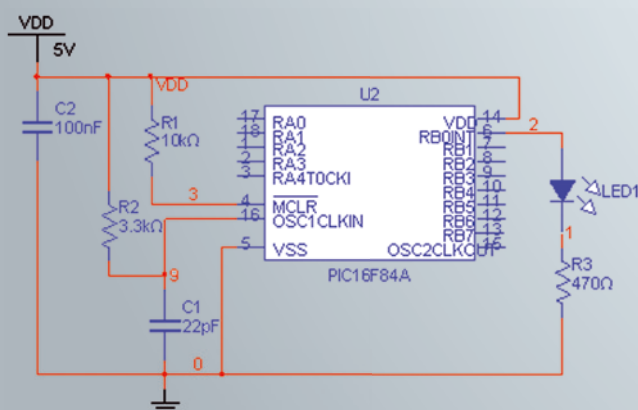
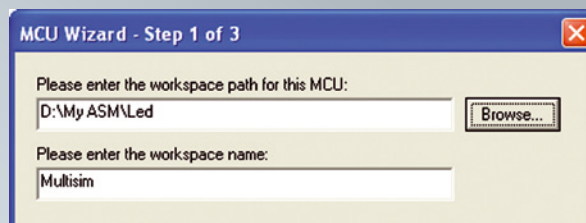


Fig. 1 - Schema circuitale per PIC16F84A del circuito con diodo LED.

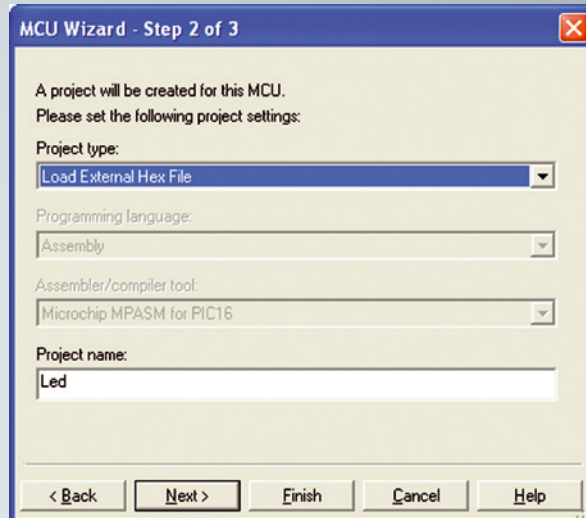
si tratta pur sempre di una simulazione e per verificare una risposta in uscita è sufficiente collegare un **digital probe**, con funzione di LED al pin 6 (RB0), il quale si illumina sul livello alto e si spegne sul livello basso, consentendoci di verificare l'effettivo funzionamento del programma che abbiamo scritto. È possibile superare questo limite del simulatore, collegando un buffer al pin in uscita (ricordando che si tratta di un'aggiunta necessaria solo ai fini della simulazione) il quale permette di comandare dei dispositivi che assorbono corrente (in questo caso un LED) e di eseguire una simulazione completa del nostro progetto. In effetti, nel caso di LED o display a LED il problema si aggira riducendo al minimo la corrente di conduzione (in Multisim), ottenendo così l'accensione dei LED senza aggiungere dei buffer. Tenete comunque presente l'utilità del buffer in tutti i casi in cui occorre effettuare simulazioni con carichi gestiti dai pin di un micro.

Ai fini della simulazione con Multisim non è necessario realizzare la rete esterna di temporizzazione del microcontrollore (pin 15 e pin 16) con oscillatore al quarzo o con oscillatore RC, perché il tempo di clock viene impostato nel box **Properties** del PIC, che si apre cliccando due volte sul dispositivo. Nella sezione **Value**, inseriamo la frequenza di clock, che generalmente, per il PIC16F84A è 4 MHz; dato che la simulazione software con Multisim richiede un certo tempo di elaborazione, 4 MHz allungano i tempi di risposta rispetto a quelli che ci aspetteremmo dal dispositivo hardware, quindi impostiamo un valore più alto (ad esempio 1 GHz) per rendere la simulazione visiva più attinente alla realtà (questo sarà più evidente nell'esempio successivo, in cui si vuole comandare un display 7 segmenti). Adesso è venuto il momento di associare al PIC il codice da eseguire (**Fig. 4**), scegliendo la sezione **Code** e poi **Properties**, la quale aprirà un'ulteriore finestra dentro cui andrà selezionato il nome del progetto Led e andrà indicato il percorso su hard-disk del file di tipo HEX contenente il codice macchina da simulare. Una volta dato l'ok a tutto, è possibile avviare la simulazione, al solito mediante il **Simulate>Run** di Multisim.

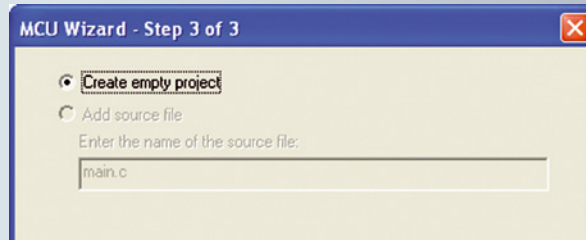
Esempio 2: segnapunti con un display 7 segmenti
In questa seconda applicazione, il PIC16F84A è stato programmato per realizzare un incre-



(a)



(b)



(c)

Fig. 2 - MCU Wizard in 3 passi.

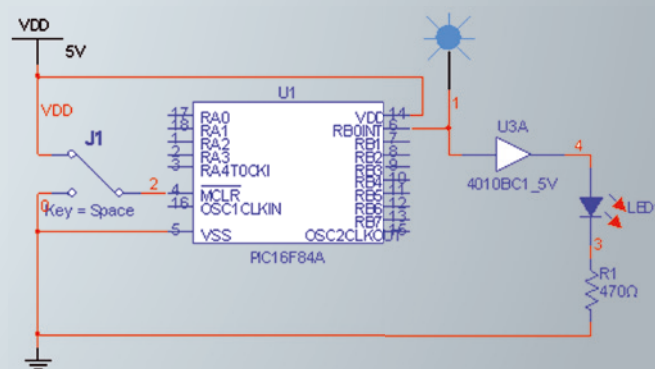


Fig. 3 - Circuito di simulazione con Multisim MCU.

mento di conteggio da 0 a 9 ogni volta che viene premuto uno switch a ritorno automatico, visualizzando il valore su un display a sette segmenti. Nel **Listato 2** viene riportato il relativo programma in C, che non abbiamo commentato. Si sceglie di usare un display

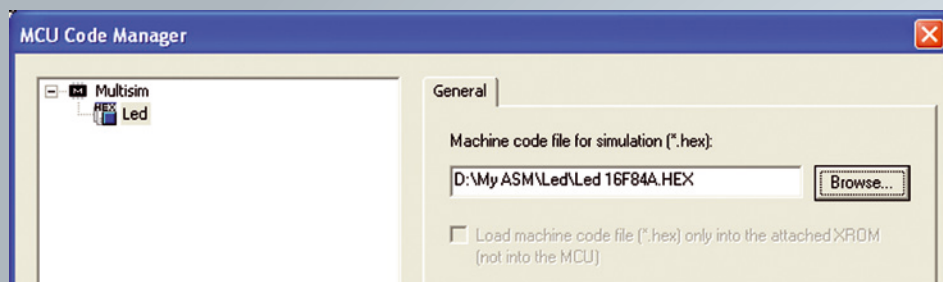


Fig. 4 - Box per l'associazione del codice operativo al PIC.

anodo comune, così tutti gli otto pin del PORTB sono impostati come uscite e ad essi vengono collegati i catodi dei LED del display, secondo lo schema di connessione descritto nel programma e in **Fig. 5**. A seconda del numero da rappresentare, viene assegnato a ciascun LED del display un valore in binario 0/1 di acceso/spento o 0/5V. La funzione **Delay(int n)** produce un ritardo temporale di $n \cdot 0,5$ s (considerando un clock del PIC di 4 MHz) e viene usata per inserire un ritardo da tarare secondo la sensibilità dello switch adoperato, in modo da evitare false commutazioni causate dalla lentezza con cui riusciamo a premere lo switch rispetto alla velocità di controllo del microcontrollore. La funzione principale **main()** esegue un conteggio e tiene in memoria il numero rappresentato su display (fino a 9 e poi ricomincia da 0). Viene

effettuato un controllo ciclico sul pin RA1 del microcontrollore: il pervenire di un livello alto di tensione (5V) a seguito di una pressione dello switch attiva un incremento della memoria e l'aggiornamento del display. Una volta compilato il programma con il compilatore HI-TECH C in ambiente MPLAB, è possibile simularlo con Multisim mediante il circuito di simulazione visibile in **Fig. 5**, procedendo come nell'esempio precedente. Al solito, notate come nella realtà non siano necessari i buffer 4010 tra PIC e display, che servono solo ai fini della simulazione (in alternativa possiamo ridurre al minimo, in simulazione, la corrente nei segmenti). Inoltre è opportuno impostare la temporizzazione del microcontrollore ad un valore più alto, ad esempio di 12 GHz, rendendo così la simulazione visiva più vicina alla realtà.

Listato 2

```
#include <pic.h>
#include <htc.h>           // Richiesta per interfacciarsi con le routine di ritardo

#define XT, WDTDIS, UNPROTECT, PWRTEN /* Oscillatore esterno: XT (interno: RC)
                                     Watch dog timer OFF
                                     Code protect OFF
                                     Power up timer ON */

#define BUTTON RA1         // Al pin 1 di PORTA è collegato il pulsante
#define _XTAL_FREQ 4000000
/* Schema di collegamento tra PIC e display a 7 segmenti

  g f a c a b          RB0 (pin 6) -> dp
  | | | | |          RB1 (pin 7) -> c
  | | | | |          RB2 (pin 8) -> d
  |   a   |          RB3 (pin 9) -> e
  | --- |          RB4 (pin 10) -> b
  | f | g | b |       RB5 (pin 11) -> a
  | --- |          RB6 (pin 12) -> f
  | e | | c |         RB7 (pin 13) -> g
  | --- |
  |   d   |
  | | | | |
  e d a c c dp */

// Lookup table per display anodo comune
const char lookup_7segmenti_anodocomune[10] = {
    0b10000001, 0b1101101, 0b01000011, 0b01001001, 0b00101101,
    0b00011001, 0b00010001, 0b11001101, 0b00000001, 0b00001001 };

// Lookup table per display catodo comune
/* const char lookup_7segmenti_catodocomune[10] = {
    0b01111110, 0b00010010, 0b10111100, 0b10110110, 0b11010010,
    0b1100110, 0b1101110, 0b00110010, 0b11111110, 0b11110110 }; */
```

(Continua)

LABVIEW INSTRUMENTS

Multisim supporta, all'interno della simulazione, l'utilizzo di strumentazione creata con LabVIEW, raggiungibile con il comando **Simulate>Instruments>LabVIEW Instruments**, oppure cliccando sulla relativa icona LabVIEW instruments posta nella toolbar **Instruments** (la stessa barra dei menu che contiene multimetro, oscilloscopio, generatore di funzione, ecc.). La strumentazione può essere di due tipi: di input o di output. Uno **strumento di input** durante una simulazione può ricevere continuamente dati da Multisim. Comunque, se si prevede di costruire o utilizzare strumenti da collegare al mondo reale di I/O (ad esempio DAQ, GPIB, seriale, file, ecc.) bisogna prendere in considerazione il fatto che la simulazione non può avvenire in tempo reale (ciò può dipendere dalla simulazione SPICE, dalla complessità dello schema, dalla velocità della CPU, ecc.). Uno **strumento di output** invia dati all'interno di Multisim. Tale strumento è comunque limitato dal fatto che non può trasferire dati a Multisim mentre la simulazione è in esecuzione; ciò significa che l'acquisizione dei dati o la loro

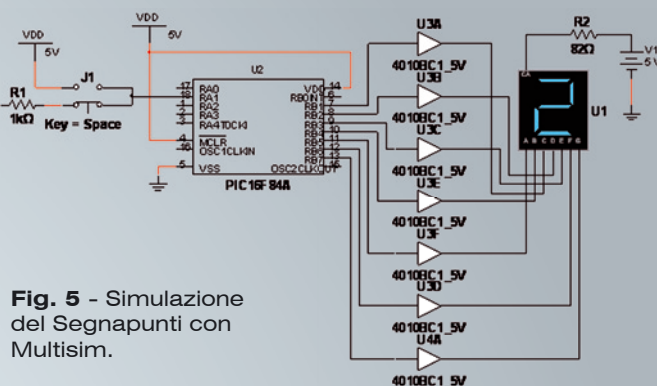


Fig. 5 - Simulazione del Segnapunti con Multisim.

generazione deve avere luogo prima di avviare una simulazione SPICE: prima vanno acquisiti i dati dal mondo reale, ad esempio mediante una scheda di acquisizione, oppure mediante una registrazione microfonica, oppure realizzando una forma d'onda con LabVIEW, ecc. e dopo viene avviata la simulazione con i dati raccolti. La creazione di uno strumento di ingresso o di uscita comporta un procedimento simile.

Esempio: comparatore a finestra

Realizziamo uno strumento LabVIEW di input che permette di simulare la caratteristica di un comparatore a finestra, ossia un dispositivo elettronico capace di riconoscere se una ten-

Listato 2 - seguito

```
//      Delay(int n) produce un ritardo di circa n * 0.5 secondi

void Delay(int n){
    int i;
    for (i=0;i<n;i++){
        __delay_ms(197);           // ritardo di 197 millisecondi
        __delay_ms(197);           // ritardo di 197 millisecondi
        __delay_ms(106);           // ritardo di 106 millisecondi
    }
}

void main(void)
{
    int j = 0;
    TRISB = 0;                     // Tutti i pin di PORTB in uscita
    for(;;) {
        while(j<10) {
            if (BUTTON == 1){      // A switch premuto, incrementa
                Delay(2);           // Ritardo da tarare secondo la sensibilità
                                    // dello switch adoperato
                PORTB = lookup_7segmenti_anodocomune[j];
                j++;
            }
            else {                  // Altrimenti, mantieni
                Delay(2);           // Ritardo da tarare secondo la sensibilità
                                    // dello switch adoperato
                PORTB = lookup_7segmenti_anodocomune[j];
            }
        }
        j = 0;
    }
}
```

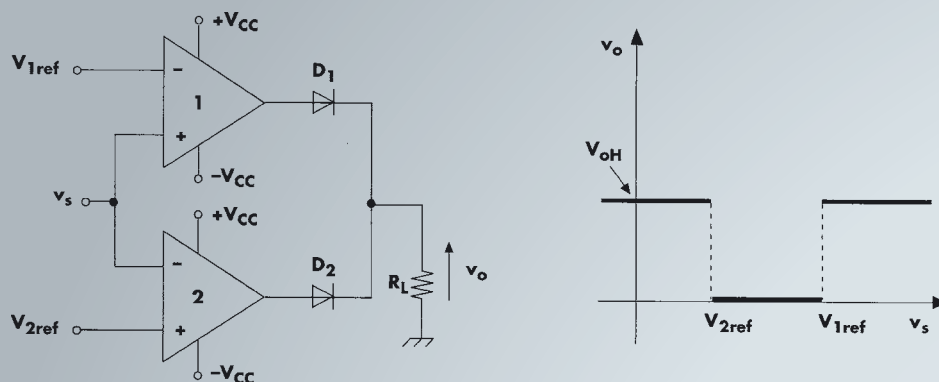



Fig. 6 - Struttura base di un comparatore a finestra.

sione in ingresso risultati compresa o no in un certo intervallo di valori. Nel circuito in Fig. 6 (tratto dal volume *L'Elettronica Analogica*, di E. Ambrosini), V_{1ref} e V_{2ref} sono rispettivamente le tensioni di riferimento maggiore e minore della finestra. L'uscita si pone a livello alto $V_o = V_{oH}$ se l'ingresso è $V_s > V_{1ref}$ o se $V_s < V_{2ref}$, mentre si porta a $V_o = 0V$ se la tensione d'ingresso $V_{2ref} < V_s < V_{1ref}$. Un dispositivo con tale caratteristica viene adesso realizzato in LabVIEW e utilizzato in un circuito elettronico in Multisim. Il comparatore a finestra che andiamo a programmare riceve un segnale da un circuito SPICE e lo confronta con un limite superiore e uno inferiore, settati dall'utente mediante controlli a slide all'interno della relativa finestra, fornendo in uscita la forma d'onda propria di un comparatore a finestra ad operazionali. Qui di seguito sono elencati i

passi da compiere per realizzare il comparatore.

1. Copiare in una nuova directory un modello di progetto che è possibile trovare nella cartella `C:\Users\Public\Documents\National Instruments\Circuit Design Suite 11.0\samples\LabVIEW Instruments\Templates\Input` (percorso valido per Vista/Win 7, oppure `C:\Documents and Settings\All Users\Documenti\National Instruments\Circuit Design Suite 11.0\samples\LabVIEW Instruments\Templates\Input` per Win XP/2000): copiare la cartella **Input** e rinominarla "Comparatore A Finestra".
2. Rinominare il file **StarterInputInstrument.lvproj** in "Comparatore A Finestra Instrument.lvproj" e aprirlo in LabVIEW con doppio clic.
3. Nella finestra Project Explorer (Fig. 7a) che appare, fare clic col pulsante destro su

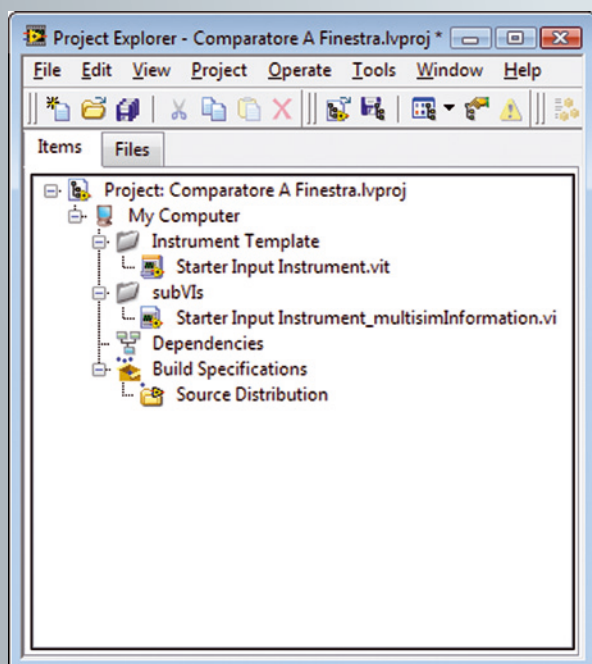
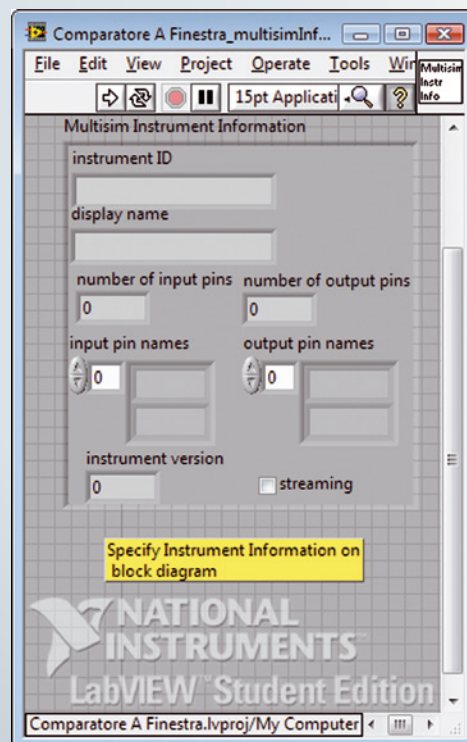


Fig. 7 - (a) Project Explorer, (b) Front Panel di "Comparatore A Finestra multisimInformation.vi".



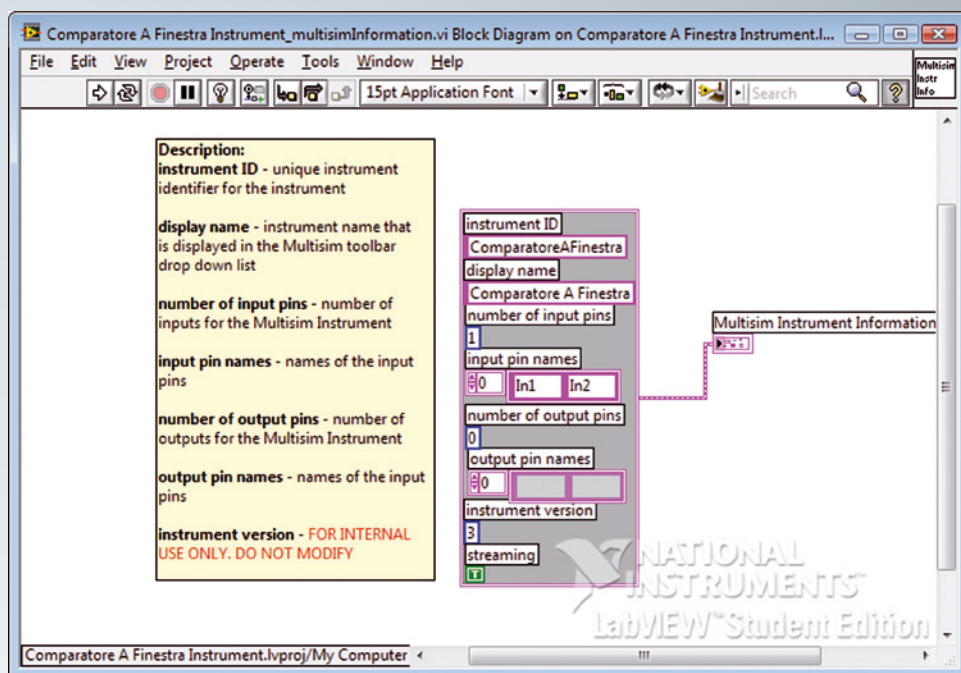


Fig. 8 - Block Diagram di Comparatore "A Finestra Instrument_multisimInformation.vi".

Instrument template>Starter Input Instrument.vit e rinominarlo in "Comparatore A Finestra.vit".

4. Ripetere la stessa procedura per rinominare **SubVIs>Starter Input Instrument_multisimInformation.vi** come "Comparatore A Finestra Instrument_multisimInformation.vi" (indipendentemente dal nome scelto per la SubVI, è necessario mantenere l'estensione "**_multisimInformation.vi**" per abilitare Multisim a caricare l'instrument);
5. Salvare il progetto.
6. Immettere le seguenti informazioni direttamente nel diagramma a blocchi (**Fig. 8**):
 - **Instrument ID** = "ComparatoreAFinestra" (unicamente usato per comunicare tra Multisim e LabVIEW);
 - **Display name** = "Comparatore A Finestra" (il nome che verrà visualizzato nella barra degli instruments in Multisim);
 - **Number of pins** = "1" (imposta il numero di pin input offerti dallo strumento);
 - **Input pin names** = "In1" (nome del pin che verrà utilizzato in una netlist SPICE o in un netlist report).
7. Salvare il VI e chiudere Block Diagram e Front Panel.

Notate che un Instrument può avere pin di ingresso o pin di uscita, ma non di entrambi i tipi; se si imposta un numero di pin di ingresso > 0 e un numero di pin di uscita > 0, lo strumento non sarà trattato come un valido strumento di

LabVIEW per Multisim. Allo stesso modo, se si sta creando uno strumento di uscita, bisogna compilare i valori appropriati per numero di pin di uscita e nomi di pin di uscita.

Si passa, adesso, alla personalizzazione dello strumento e quindi del pannello frontale del VI; successivamente si programmerà il VI.

1. Aprire Comparatore A Finestra.vit, per realizzare l'interfaccia dello strumento che un utente vede e gestisce in Multisim; nel Front Panel muovere via dalla finestra (ma non eliminare) tutti i controlli già presenti, ma che un utente non dovrebbe vedere.
2. Costruire uno schema come riportato in **Fig. 9** composto da:
 - due **Waveform Chart** (uno per visualizzare il segnale in ingresso al comparatore e l'altro per visualizzare l'uscita); si fa notare che per il Waveform Chart che visualizza il segnale in ingresso, può essere usato direttamente quello disponibile nella VI;
 - tre **Pointer Slide** (per settare V_{1ref} , V_{2ref} e V_{oH}), per i quali si è abilitato il display digitale (selezionare ciascun Pointer Slide con il pulsante destro del mouse e quindi scegliere **Visible Items > Digital Display**);
 - uno **Square LED** (collegato ad un controllo booleano, che si illumina a seconda che il segnale d'ingresso V_s sia compreso nel range scelto oppure sia esterno).
3. Passare al Block Diagram dell'instrument (**Fig. 10**) e programmare il processo nella zona "Update Data" della Case Structure

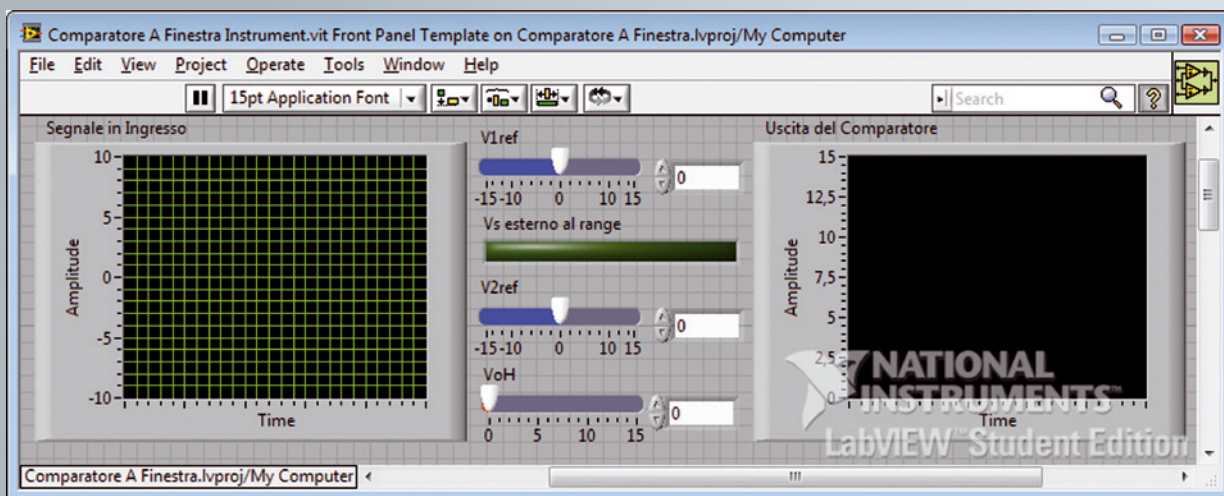


Fig. 9 - Front Panel di "Comparatore A Finestra.vi".

localizzata nella parte bassa del ciclo While Loop, proprio dove è indicato >> **add processing SubVI here** >> (se è necessario, allargate il rettangolo del Case per farvi entrare tutti i blocchetti della propria programmazione):

- la forma d'onda d'ingresso è disponibile al **Multisim input pins** e viene inviato su un **Index Array**, la cui uscita portata su un **Get Waveform Component** è adattata in forma d'onda analogica e inviata all'ingresso x della funzione **In Range and Coerce**, la

quale determina se x rientra nell'intervallo specificato dagli ingressi limite superiore e limite inferiore (settaggi delle slide V_{1ref} e V_{2ref}), emettendo un booleano: se x è esterno si fa accendere il LED (si usa al proposito un altro **Index Array** e un **NOT**);

- il confronto viene inviato anche all'ingresso s della funzione **Select**, che restituisce in uscita l'ingresso t , collegato allo slide V_{oH} , oppure l'ingresso f , collegato alla costante numerica 0, a seconda che s abbia un valore rispettivamente VERO oppure FALSO.
4. Salvare il VI, quindi chiudere sia il pannello frontale che lo schema a blocchi.

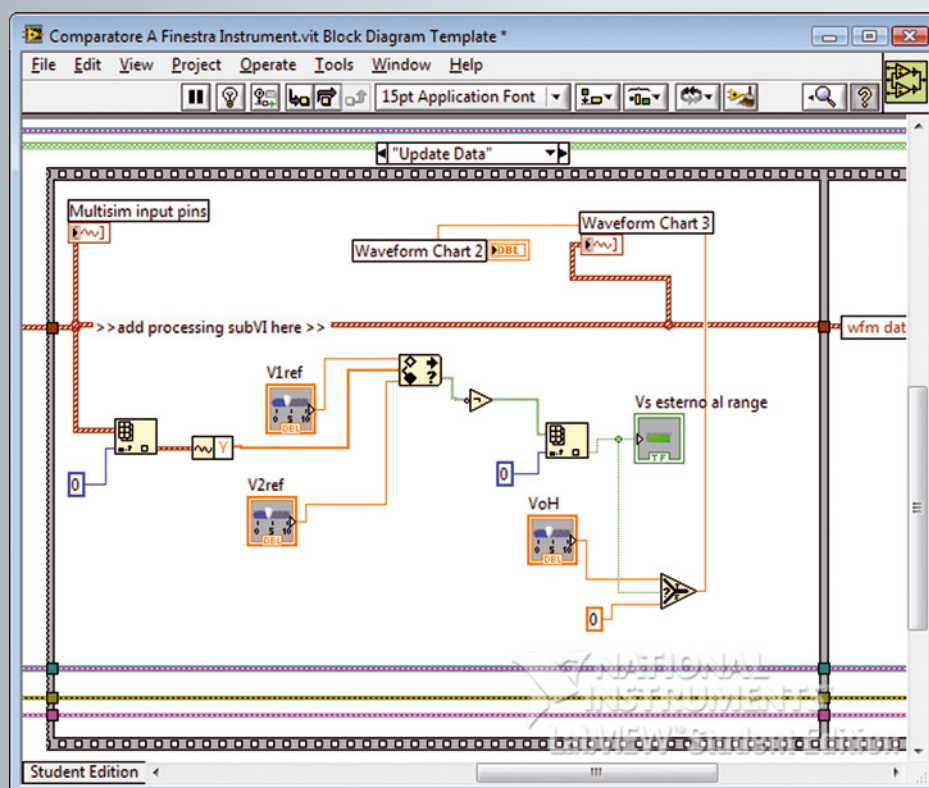
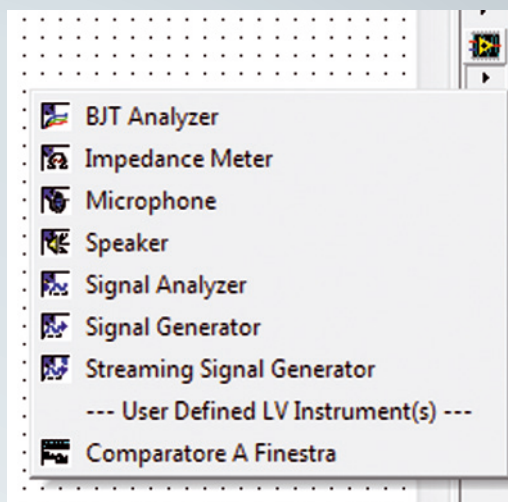


Fig. 10
Block Diagram di
"Comparatore A
Finestra Instrument.vi".



Nel caso, invece, si voglia creare uno strumento di uscita, il procedimento da seguire è molto simile, tranne per il fatto che è necessario collegare con un filo al **Multisim output pins** tutti i dati che si desidera inviare a Multisim, nell'opzione del Case denominata "Update Initial Output Data". Se si intende salvare i dati acquisiti dallo strumento e inerenti all'applicazione Multisim (le impostazioni, i valori di controllo, e così via), è necessario inserire il codice appropriato nella Case Structure nelle sezioni "Serialize Data" e "Deserialize Data".

Siamo arrivati alla conclusione del procedimento che porta alla realizzazione dello strumento

Fig. 11 - Lo strumento "Comparatore A Finestra in Multisim".

personalizzato, che adesso va inserito in una libreria LabVIEW per poi essere utilizzato in ambiente Multisim. Ecco i passi da seguire.

1. Nella finestra Project Explorer (**Fig. 7a**), espandere **Build Specifications** e fare doppio click su **Source Distribution** per aprire le impostazioni di distribuzione.
2. Selezionare la voce **Destination**, modificando la directory di distribuzione **... \Build \InputInstrument.llb** in "... \Build \Comparatore A Finestra.llb" e premere il pulsante **Build**.
3. Premere **Done** dopo che è stato completato il processo di compilazione e salvare il progetto, chiudendo LabVIEW.
4. Per poter usare lo strumento di LabVIEW appena creato, copiare il relativo file *Comparatore a Finestra.llb* nella cartella riservata agli strumenti creati dall'utente: **C:\Users\Public\Documents\National Instruments\Circuit Design Suite 11.0\LVInstruments** (supponendo di avere installato Multisim 11 su disco C in ambiente Vista/Win7). In alternativa, anche se non è la scelta propriamente corretta, è possibile installare lo strumento direttamente nella cartella degli strumenti LabVIEW prodotti da NI per Multisim: **C:\Programmi\National Instruments\Circuit Design Suite 11.0\lv instruments** (se ci sono problemi a localizzare la cartella lvinstru-

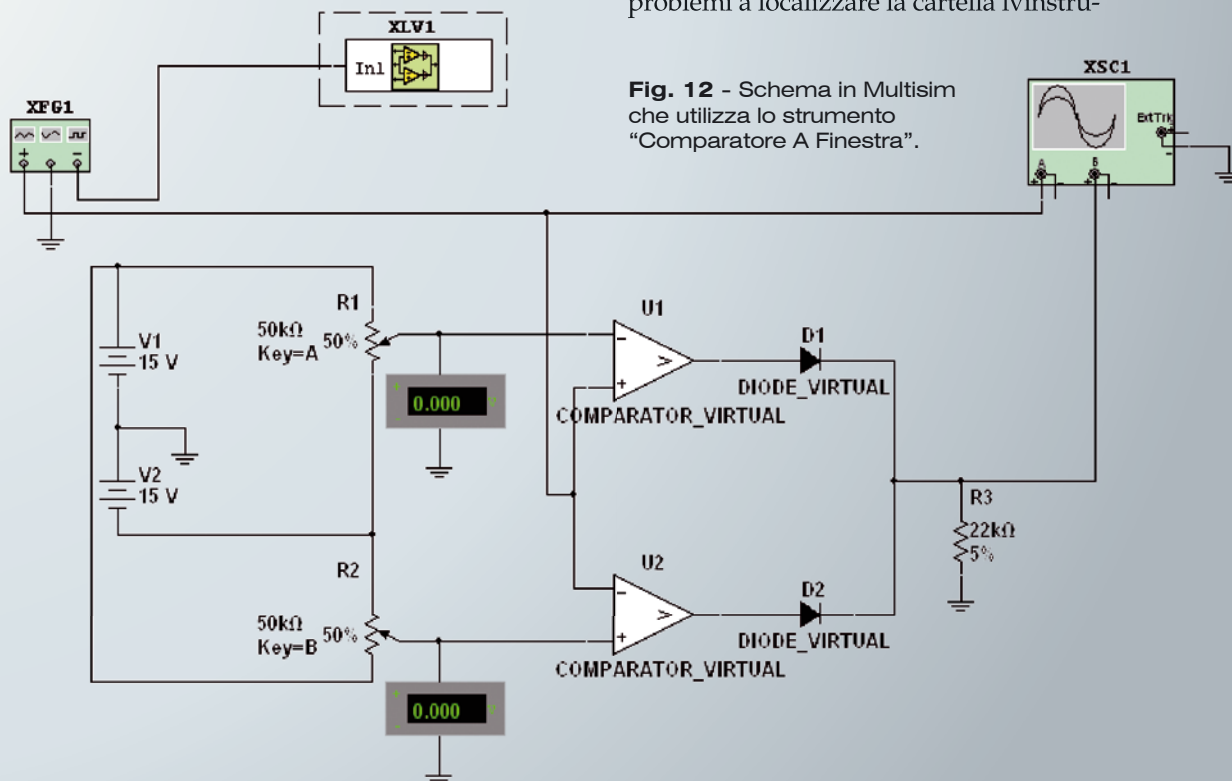


Fig. 12 - Schema in Multisim che utilizza lo strumento "Comparatore A Finestra".

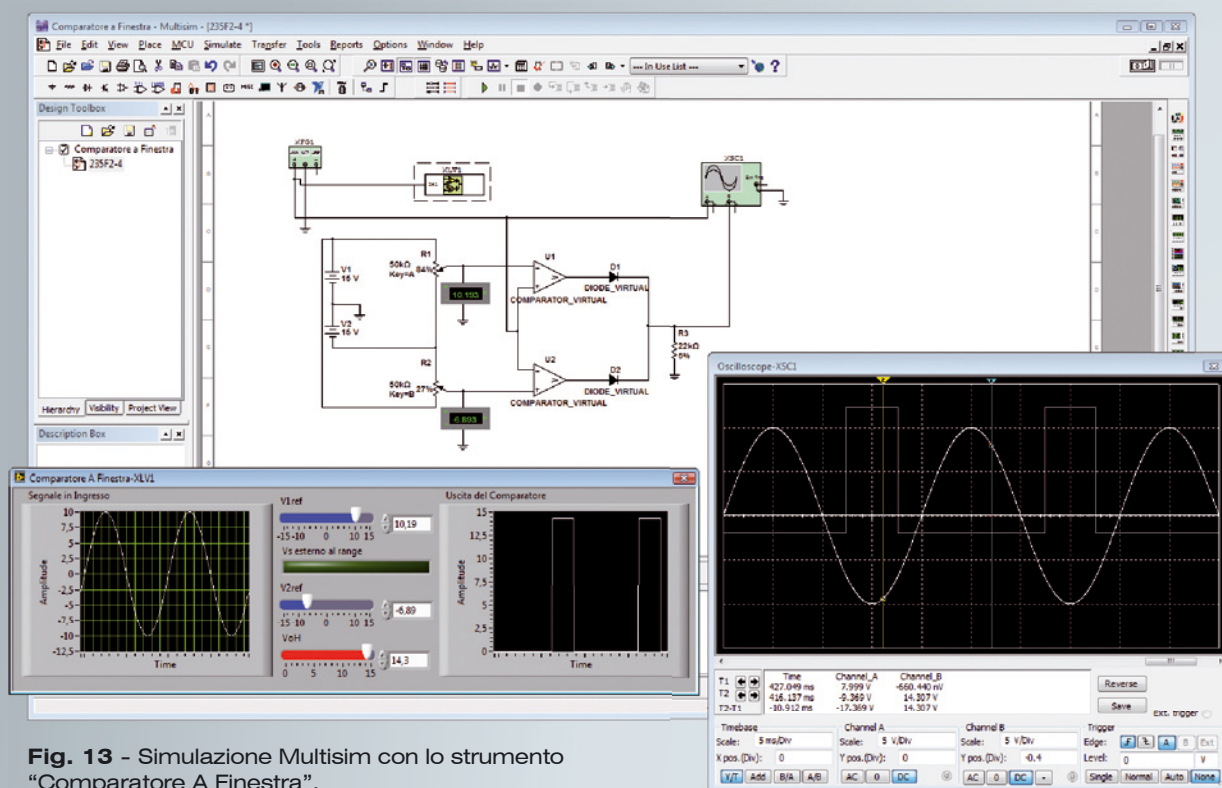


Fig. 13 - Simulazione Multisim con lo strumento "Comparatore A Finestra".

ment si può individuare il relativo percorso in **Multisim** scegliendo **Options>Global Preferences>Paths>User LabVIEW instruments path**.

5. Eseguire Multisim e verificare che lo strumento sia presente tra i LabVIEW instruments, dopodiché farà parte di Multisim e potrà essere utilizzato in qualunque simulazione futura (Fig. 11).

A questo punto bisogna creare uno schema elettrico, come riportato in Fig. 12, che contiene sia la realizzazione circuitale di un comparatore a finestra con operazionali, sia lo strumento LabVIEW "Comparatore A Finestra" appena creato, in modo da poter confrontare le due diverse modalità di simulazione e vedere come Multisim e LabVIEW interagiscono (Fig. 12). Dopo aver lanciato il **Run**, facendo doppio clic sul VI e sull'oscilloscopio si potranno vedere e valutare i relativi grafici. In alternativa è possibile reperire, nel nostro sito www.elettronica.in.it, ambedue i file *Comparatore a Finestra.ms11* e *Comparatore A Finestra.Ilb*, e provarli direttamente (a patto di installare la libreria nella cartella degli strumenti). Lo strumento creato da noi si presenta con l'icona che mostra un comparatore a finestra (visibile in Fig. 12) e con un pannello frontale (Fig. 9). Il Waveform Chart

a sinistra rappresenta il segnale in ingresso proveniente dal circuito di Multisim, mentre in quello a destra è riportata l'uscita del comparatore. In Fig. 13 vedete una simulazione in cui viene analizzata una forma d'onda generata dal generatore di funzione di Multisim e lo stesso segnale viene inviato al circuito del comparatore realizzato con componenti elettronici.

Ulteriori LabVIEW instruments

Come illustrato in Fig. 11, Multisim offre già alcuni strumenti LabVIEW pronti per le nostre esperienze, tra i quali il **Microphone** e lo **Speaker**, che permettono, rispettivamente, di acquisire ed emettere suoni interagendo con un microfono e un altoparlante del PC (tramite la scheda audio), utili ad esempio per testare lo schema di un amplificatore audio o per altre esperienze.

I file delle simulazioni presentati in questo corso sono scaricabili gratuitamente dal sito www.elettronica.in.it.

Il software Multisim fa parte del pacchetto NI Circuit Design Suite Education (cod. 8100-NICDSE) disponibile presso www.futurashop.it. La versione per utenti privati e studenti costa 99,95 Euro.



31⁰ MARC

**mostramercato attrezzature
radioamatoriali & componentistica
hardware • software
ricezione satellitare
editoria specializzata
radio d'epoca**

**Fiera di Genova
17 - 18 Dicembre 2011**

**sabato ore 9 • 18,30
domenica ore 9 • 18**

**Ente organizzatore e segreteria:
STUDIO FULCRO s.a.s. - Genova
Tel. 010.561111
e-mail: info@studio-fulcro.it**

**Ente patrocinatore:
A.R.I. Ass. Radioamatori Italiani
Sezione di Genova
Tel. 010.255158
www.arigenova.it**





Conosci bene ARDUINO?

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

La soluzione è...

TiDiGino

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

Sviluppa il tuo controllo remoto basato su ARDUINO!

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

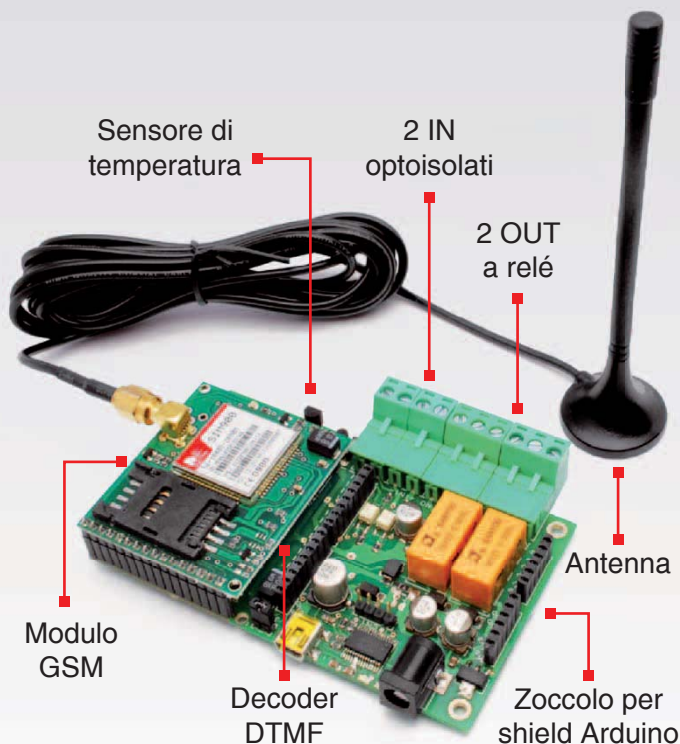
Lo zoccolo disponibile sulla scheda TiDiGino ti permette di utilizzare shield per Arduino progettati per svolgere specifiche funzioni utili ad ampliare le risorse hardware e le potenzialità applicative dell'unità.

Il telecomando GSM dispone di:

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

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

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



€ 138,00
cod. FT924M

Sfruttando un ATmega 2560, ovvero il cuore di Arduino, abbiamo realizzato un telecomando GSM universale dotato dell'hardware che permette di ottenere le funzioni di controllo a distanza 2 IN/2OUT, chiave DTMF, apricancello e termostato attivabile a distanza: ognuna delle quali, in precedenza, richiedeva una scheda specifica.



TiDiGino, IL TELECONTROLLO BASATO SU ARDUINO

di BORIS LANDONI

Ricordate i nostri telecomandi, riproposti in chiave modulare nei fascicoli dal n° 146 al n° 150? Ebbene, abbiamo deciso di realizzarne tutte le funzioni con un solo circuito, però, a differenza di quanto fatto nel sistema modulare, qui usiamo come unità di elaborazione Arduino. Perché questa scelta? Semplice: innanzitutto il suo hardware si può integrare in qualsiasi circuito, essendo open source (quindi lo schema elettrico è reso pubblico) e poi programmare il telecomando è più sempli-

ce, grazie alla disponibilità di numerose librerie che permettono di far fare al microprocessore di Arduino qualsiasi cosa; al limite, se non c'è proprio una libreria specifica è possibile modificarne una esistente. Così è nato TiDiGino, ossia il nostro telecomando della serie TDG, ma basato su Arduino. Integrando l'hardware di Arduino, il nostro sistema dispone di connettori S.I.L. per montare, sul circuito base, eventuali shield; ciascuno di essi è nella stessa posizione in cui si troverebbe nella piattaforma di sviluppo originale, il che permette

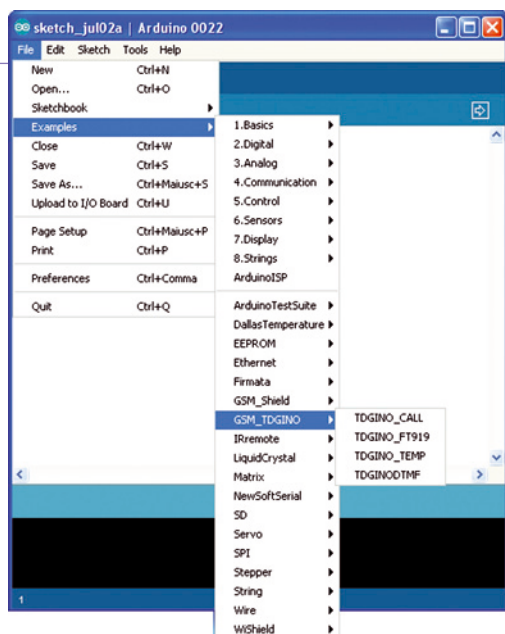


Fig. 1

di utilizzare gli shield commerciali e comunque quelli standard. Abbiamo detto che le funzioni dei nostri telecontrolli, ossia 2IN/2OUT, apriancello, chiave DTMF e termostato GSM possono essere realizzate mediante appositi firmware; ebbene, questi avremmo potuto scriverli noi, ma abbiamo voluto proporre ai nostri lettori che conoscono l'ambiente Arduino di farlo loro. Questo è il senso del TiDiGino Contest, che avete potuto seguire sul nostro sito www.elettronica.in e sul nostro blog <http://blog.elettronica.in> e che si è appena concluso; come promesso, pubblichiamo l'hardware del telecontrollo e qualche routine.

IL CIRCUITO

Il TiDiGino è basato su un chip ATmega 2560, quindi l'architettura da noi scelta è quella dell'Arduino Mega; del processore sono stati utilizzati alcuni pin per gestire le funzionalità GSM, corrispondenti a porte che non sono utilizzate dall'Arduino MEGA 2560 originale. Per questo motivo è necessario sostituire il file `pins_arduino.c` localizzato sotto la cartella `C:\Programmi\arduino-0022\hardware\arduino\cores\arduino` che si crea scaricando l'IDE di Arduino, con quello da noi reso disponibile sul

nostro sito; altrimenti non è possibile far gestire all'ATmega 2560 le linee di I/O non previste dalla piattaforma Arduino MEGA. Tale scelta è stata voluta per lasciare gli I/O dell'ATmega liberi per essere utilizzati dagli eventuali shield che possiamo montare sul TiDiGino, che poi sono gli stessi compatibili con Arduino UNO, Duemilano e Arduino MEGA. Pertanto è possibile utilizzare gli sketch già realizzati per controllare un determinato shield con le Board originali Arduino, anche su TiDiGino.

Per consentire a chi si è cimentato nel Contest e a chiunque voglia utilizzare il TiDiGino, di sviluppare il proprio firmware, nel rispetto della filosofia open-source abbiamo reso disponibili sul nostro sito le librerie da noi realizzate per far funzionare i principali blocchi della scheda; le librerie ci sono peraltro servite per testare il funzionamento dei blocchi del TiDiGino.

Abbiamo dunque realizzato quattro sketch, ognuno dei quali consente di utilizzare una sezione del sistema; sono tutti contenuti nel file `GSM_TDTGINO.zip`, scaricabile dal nostro sito, che contiene la libreria che permette di gestire la sezione GSM del TDTGINO.

Questa libreria deriva da quella sviluppata da HWKitchen, ma è stata adattata al nostro hardware, in quanto, ad esempio, utilizza la seconda seriale dell'ATMEGA2560 per gestire il modulo GSM SIM900 della SimCom. Decomprimendo lo zip sotto la cartella delle librerie di Arduino (ad esempio `C:\Programmi\arduino-0022\libraries`) la libreria è immediatamente utilizzabile. Copiando la libreria, automaticamente vengono anche installati gli esempi che abbiamo sviluppato per gestire le varie periferiche; per poter provare questi esempi è necessario collegare il TiDiGino alla porta USB e fornire al circuito un'alimentazione esterna di circa

12 Vcc (serve 1 A di corrente). Verrà creata una COM virtuale che servirà per programmare il telecontrollo. Selezionate la Board "Arduino Mega 2560" e, dal menu `File->Examples->GSM_TDTGINO` (fig. 1), scegliete l'esempio che volete caricare sul telecontrollo.

Gli esempi in questione sono gli sketch di seguito descritti.

- **TDTGINO_CALL**: permette di capire come funziona la libreria che gestisce il modulo; infatti invia sulla seriale lo stato della chiamata, ovvero se c'è una chiamata in corso, in arrivo o nessuna chiamata. Inoltre controlla i primi 10 numeri di telefono presenti sulla SIM e se arriva un SMS vi riporterà sulla finestra di debug seriale il testo e il numero del mittente.
- **TDTGINO_TEMP**: permette di testare la sonda di temperatura DS18B20. Tale sketch richiede, per poter gestire la comunicazione con la sonda di temperatura della Dallas, la libreria scaricabile dalla pagina web www.milesburton.com/Dallas_Temperature_Control_Library. Una volta caricato lo sketch nel TiDiGino, premendo il pulsante verrà inviato un SMS contenente la temperatura al numero specificato; inoltre, in caso di chiamata il telecontrollo risponderà e premendo i pulsanti sulla tastiera del proprio telefono sarà possibile verificare il funzionamento della sezione DTMF.
- **TDTGINODTMF**; questo sketch che permette di verificare lo stato della chiamata (chiamata in arrivo, chiamata attiva, ecc) e di verificare il corretto funzionamento della sezione DTMF. Restituisce inoltre il pulsante premuto sulla tastiera del telefono.

Un altro sketch che trovate sul nostro sito è il `TDTGINO_FT919`,

pensato per verificare lo stato degli ingressi digitali o analogici della scheda FT919 (pubblicata nel fascicolo n° 154) e gestirne le uscite a relé tramite semplici comandi seriali.

Ovviamente si potrebbero attivare i vari relé semplicemente chiamando la scheda e utilizzando i comandi DTMF.

L'HARDWARE

Passiamo adesso all'analisi del circuito che realizza il TiDiGino, meglio descritto dallo schema elettrico visibile in queste pagine: tutto ruota intorno al microcontrollore U6, che è appunto l'ATmega 2560; completa l'unità, l'alimentatore switching che ricava i 3,3 volt da inviare ai connettori di espansione ed i 4 V per la logica.

Del micro, le linee di I/O utilizzate per le espansioni (ossia portate ai connettori single-in-line destinati a collegare gli shield) sono PB4÷PB7, PE3÷PE5, PG5, PH3÷PH6; ci sono poi PE0 e PE1, che funzionano anche, rispettivamente, da RXD e TXD del primo UART interno al microcontrollore. Vref del connettore di espansione è ricavata da Aref dell'U6, che fornisce il riferimento per gli input analogici; questi ultimi fanno capo alle linee ADC0÷ADC5 del micro. Questo è quanto riguarda le connessioni standard di Arduino.

Ora vediamo le linee di I/O impiegate per gestire le periferiche necessarie ad implementare le funzionalità del telecontrollo, partendo dal 2IN/2OUT: PE6 e PE7, configurate come input senza pull-up (al pull-up provvedono R38 ed R39) servono a leggere lo stato degli ingressi optoisolati, ognuno dei quali rileva la presenza di una tensione continua (da un minimo di 3 ad un massimo di circa 35 volt) o alternata (da un minimo di 2,5 ad un massimo di circa 30 Veff.) applicata ai capi IN1 (circuito di U4) e IN2

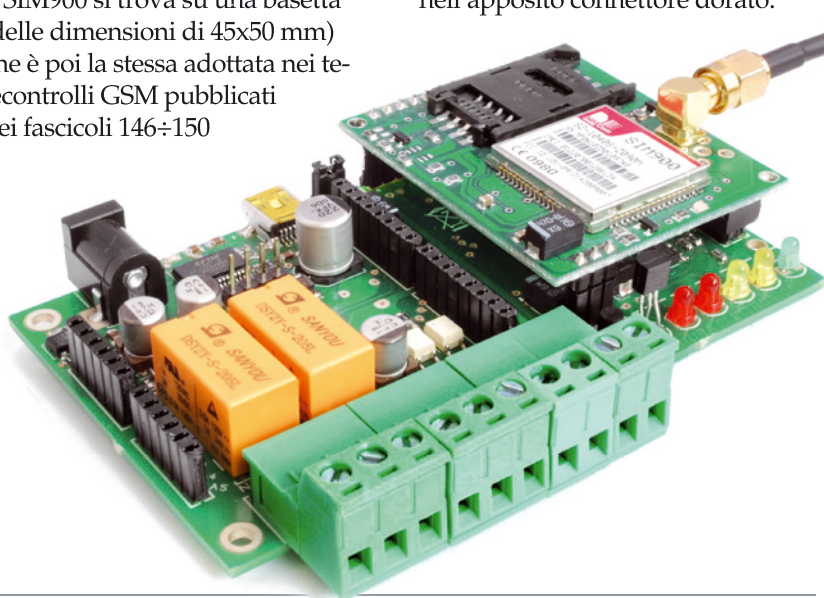
(circuito facente capo ad U5). Di tali ingressi notate le resistenze tratteggiate: sono in realtà dei ponticelli (resistenze da 0 ohm) che dovete montare solo se volete leggere la presenza di tensioni rinunciando all'isolamento galvanico; resta inteso che in tal caso è possibile rilevare solo tensioni continue a livello 0/5 V, perché l'azione rettificatrice del LED interno al fotoaccoppiatore viene meno.

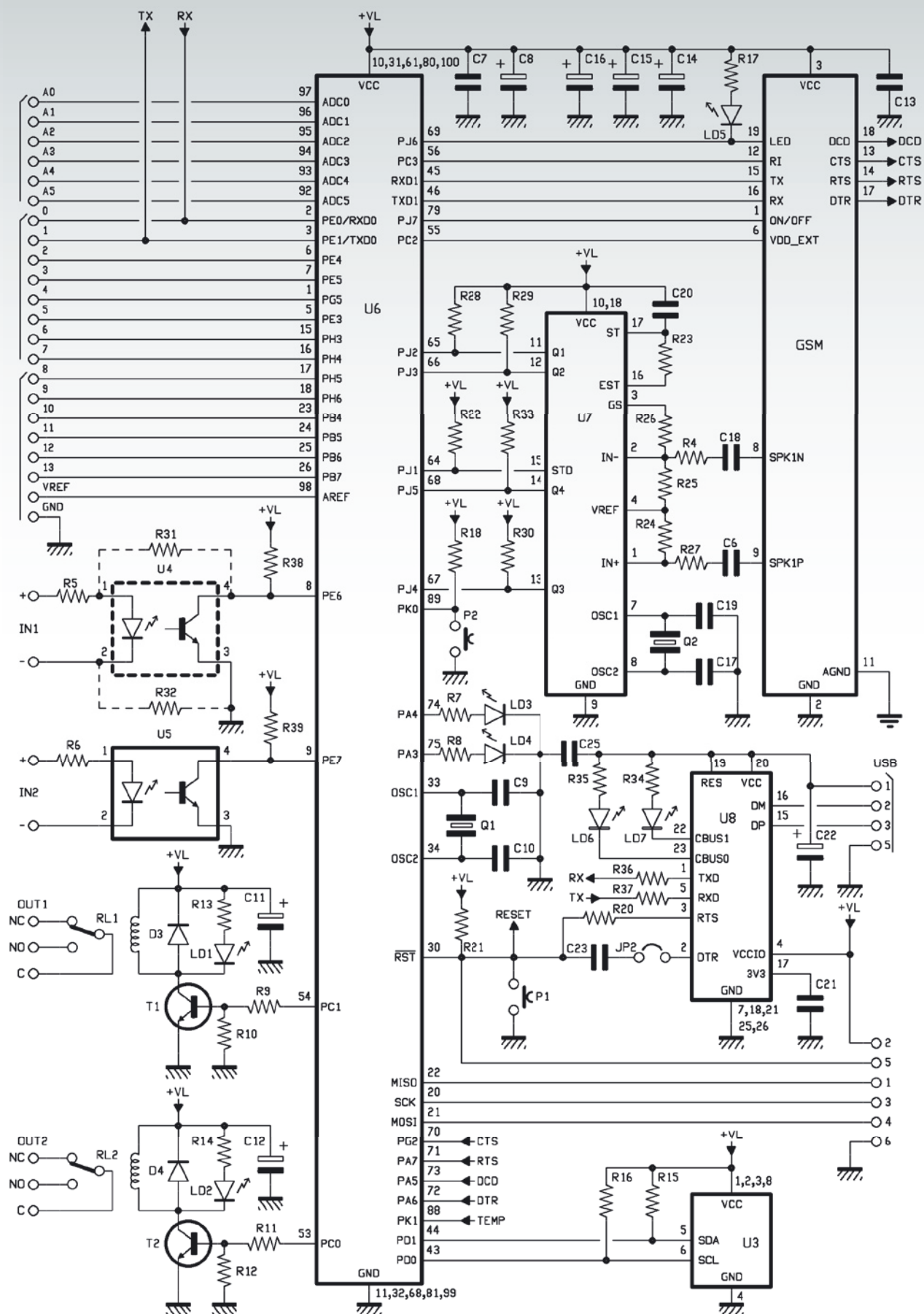
Il comando dei relé di uscita viene ottenuto con PC0 e PC1, inizializzate come uscite; ciascun relé è comandato mediante un transistor NPN, la cui base viene polarizzata dalle linee del micro ed il cui collettore controlla la corrente nella bobina d'innescio.

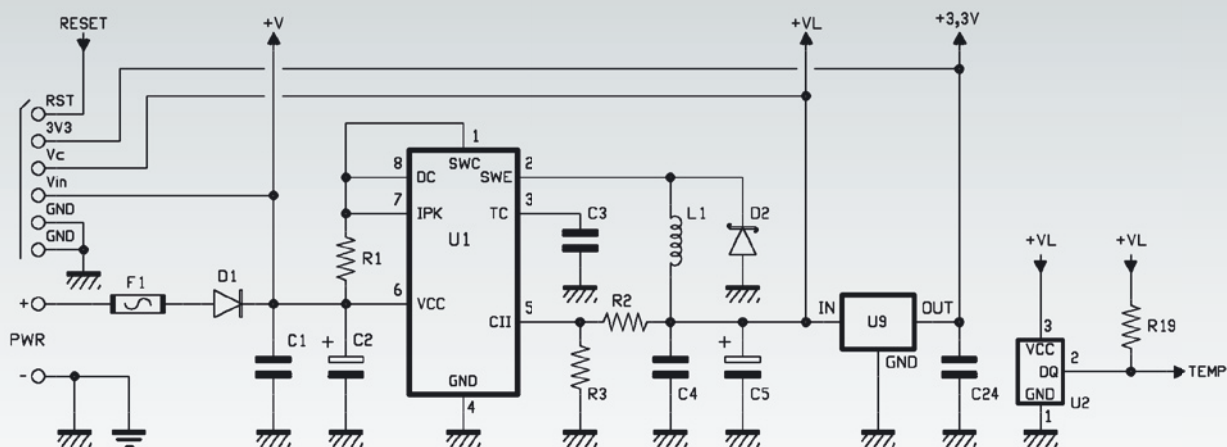
Notate il diodo al silicio sul collettore, che sopprime le tensioni inverse prodotte dalla bobina all'interdizione del rispettivo transistor, oltre al bipolo LED resistenza, che permette l'indicazione visiva dello stato dell'uscita (LED acceso=relé eccitato).

Passiamo adesso alla gestione del cellulare, che è quella più impegnativa; prima, però, è d'obbligo premettere che quello siglato GSM nello schema elettrico non è il modulo GSM, ma un circuito (siglato TDGSM_900) che lo ospita. Infatti il SIM900 si trova su una basetta (delle dimensioni di 45x50 mm) che è poi la stessa adottata nei telecontrolli GSM pubblicati nei fascicoli 146÷150

e descritta, per l'esattezza, nella rivista n° 147, alla quale rimandiamo per tutti i dettagli. In questa sede ci limitiamo a dire che LED è l'uscita che riporta il segnale di controllo del LED di stato della rete, RI corrisponde al criterio Ring Indicator (ossia il segnale che avverte dell'arrivo di una chiamata) TX ed RX fanno capo all'UART del SIM900 e ON/OFF è la linea di comando dell'accensione del cellulare; infine, Vdd_EXT è il segnale di comando usato anche per la gestione del reset. Ebbene, l'ATmega configura PJ6 e PC3 come ingressi per leggere, rispettivamente, LED ed RI, mentre inizializza PJ7 e PC2 come uscite per controllare, rispettivamente, ON/OFF e Vdd_EXT. Le linee del secondo UART sono impiegate per dialogare con il canale seriale del cellulare (TX ed RX); oltre a questi pin, per la gestione della comunicazione con il modem del cellulare vengono utilizzate PG2 (CTS) PA7 (RTS) PA5 (DCD) e PA6 (DTR). La basetta TDGSM_900 dispone di due pin-strip per le connessioni con il circuito base del telecontrollo, uno da 3 pin ed uno da 16 pin. Il modulo viene fornito con la sua antenna GSM (cod. 8170-ANTG-SMSTL-S01 Futura Elettronica www.futurashop.it) da inserire nell'apposito connettore dorato.







Passiamo oltre e vediamo la sezione che interfaccia la memoria in cui scriveremo i dati di configurazione: questa scelta ci risparmia di ingombrare la EEPROM interna al microcontrollore, riservandone l'intera capienza al firmware.

Dunque, per gestire la memoria vengono impiegate le linee PD0 e PD1, usate per costituire un I²C-Bus di cui la prima è il clock e la seconda il canale dati SDA.

La memoria usata è una EEPROM siglata 24FC256-SN; si tratta di una memoria CMOS da 256 kbit ad accesso seriale, con interfaccia I²C-Bus.

Quanto alla gestione del sensore di temperatura, usato esclusivamente per la funzionalità "Termostato GSM", l'ATmega sfrutta PK1 (TEMP) inizializzandola come linea bidirezionale; la sonda termica utilizzata nel telecontrollo è la DS1820 della Dallas, capace di misurare temperature comprese nel range -5÷150 °C con un'accuratezza di ± 0,5 °C (nel campo -10÷85 °C), di digitalizzarle e di renderle disponibili su una linea seriale unidirezionale, collegata al pin RC0 del micro. La sonda dialoga usando il protocollo One-wire, che permette il dialogo bidirezionale su un unico filo; allo scopo viene richiesta la resistenza di pull-up R19.

E veniamo alla chiave DTMF, che sfrutta il decoder U7 (un MT8870,

in SMD): le linee usate del micro sono PJ2, PJ3, PJ4, PJ5 per leggere il bus dati Q1÷Q4 e PJ1 per leggere l'STD, ossia il segnale che comunica, con un impulso a livello logico alto, la decodifica di ogni bitono DTMF.

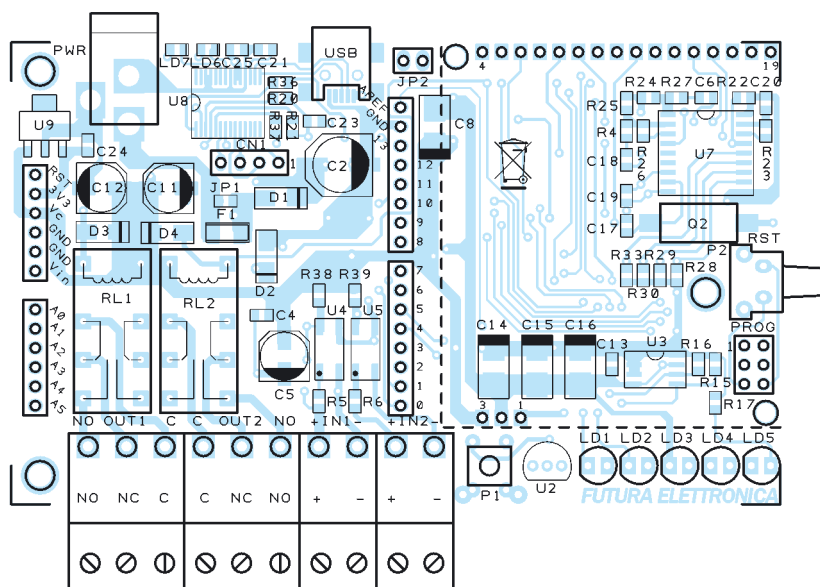
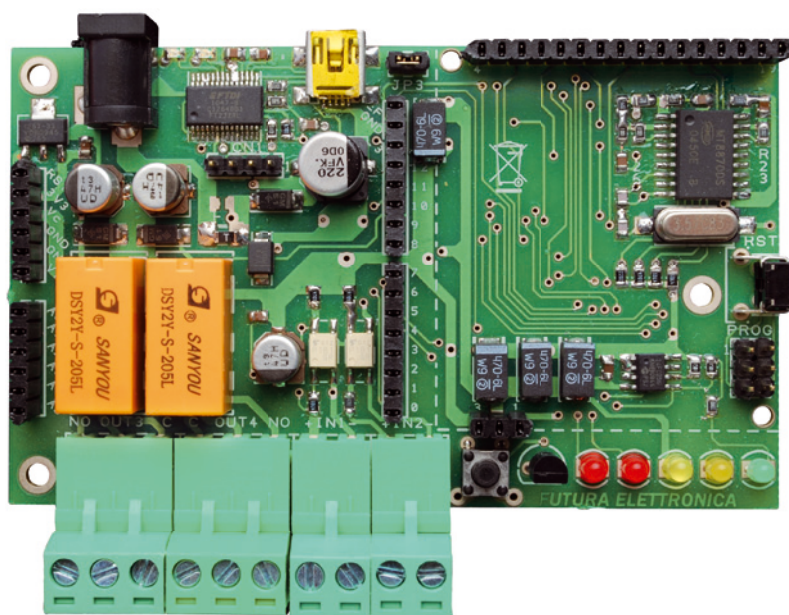
L'MT8870 è un chip capace di decifrare i 16 bitoni standard grazie ad un complesso schema di filtri attivi accordati mediante il segnale di clock generato dal suo oscillatore interno, il quale fa capo ai piedini 7 ed 8 e al quarzo Q2. Ogni volta che riconosce un bitono, U7 ne presenta in formato binario il numero corrispondente sulle uscite Q1 (bit meno significativo) Q2, Q3 e Q4 (bit più significativo); i numeri da 0 a 9 sono rappresentati con le combinazioni binarie da 0000 a 1001, mentre A, B, C, D, * e # vengono espressi con le restanti combinazioni, che vanno da 1010 a 1111. L'MT8870 dispone in uscita di un latch che permette di mantenere su Q1÷Q4 la combinazione logica corrispondente all'ultimo bitono letto.

L'interfaccia USB del circuito è realizzata con l'ausilio dell'integrato U8, il classico FT232RL cui sono dedicati i piedini 3 (TX) e 2 (RX) che corrispondono al primo UART; dato che esse servono essenzialmente per caricare il bootloader (l'operazione si avvale della linea DTR dell'FT232RL, collegata

mediante il condensatore C23 ed il jumper JP2 al reset dell'ATmega) nel circuito.

Il pulsante P1 serve al reset del circuito, che viene forzato anche da DTR, quando dal computer vogliamo caricare il firmware nel micro, sfruttandone il bootloader.

Bene, concludiamo l'analisi circuitale con l'alimentazione, che è a tensione continua, anche non stabilizzata (applicata ai punti + e - PWR) di valore compreso tra 5 e 32 V; tale tensione viene filtrata a valle del diodo di protezione contro l'inversione di polarità (D1) mediante i condensatori C1 e C2; il fusibile F1 ci permette di proteggere il circuito e la fonte di alimentazione in caso di cortocircuito nel regolatore integrato che segue e che ci serve a ricavare i 4 volt necessari a far funzionare il resto del circuito. Il regolatore switching U1 è basato sul chip MC34063, utilizzato come regolatore buck, la cui tensione di uscita dipende dall'energia immagazzinata in L1; viene stabilizzato mediante la componente retrocessa dal partitore resistivo R2/R3, che serve a fissare a 4 V la componente livellata ai capi di C4 e C5. I 4 volt a valle di questi condensatori vengono abbondantemente filtrati da altri condensatori collocati sulle linee di alimentazione del microcontrollore e del modulo GSM, il quale in trasmissione presenta picchi di as-



Elenco Componenti:

- R1: 0,1 ohm 1W (1206)
- R2: 2,2 kohm (0805)
- R3: 1 kohm (0805)
- R4: 100 kohm (0805)
- R5, R6: 4,7 kohm (0805)
- R7, R7: 330 ohm (0805)
- R9, R11: 4,7 kohm (0805)
- R10, R12: 10 kohm (0805)
- R13, R14: 330 ohm (0805)
- R15, R16: 1,5 kohm (0805)
- R17: 330 ohm (0805)
- R18, R19: 4,7 kohm (0805)
- R20: 100 ohm (0805)
- R21, R22: 4,7 kohm (0805)
- R22: 4,7 kohm (0805)
- R23: 330 kohm (0805)
- R24: 39 kohm (0805)
- R25: 56 kohm (0805)
- R26, R27: 100 kohm (0805)
- R28, R30: 4,7 kohm (0805)
- R31, R32: 0 ohm (0805) - Vedi Testo
- R33: 4,7 kohm (0805)
- R34, R35: 470 ohm (0805)
- R36, R37: 10 ohm (0805)
- R38, R39: 4,7 kohm (0805)
- C1, C4, C6, C7: 100 nF multistrato (0805)
- C2: 220 μ F 35 VL elettrolitico (F)
- C3: 100 pF ceramico (0805)
- C5: 100 μ F 16 VL elettrolitico (D)
- C8: 470 μ F 6,3 VL tantalio (D)
- C9, C10: 22 pF ceramico (0805)
- C11, C12: 47 μ F 16 VL elettrolitico (D)
- C13: 100 nF multistrato (0805)
- C14÷C16: 470 μ F 6,3 VL tantalio (D)
- C17: 10 pF ceramico (0805)
- C18: 100 nF multistrato (0805)
- C19: 10 pF ceramico (0805)
- C20, C21: 100 nF multistrato (0805)

sorbimento cui suppliscono C7, C8, C13, C14, C15 e C16, evitando che la richiesta impulsiva di corrente causi impulsi di disturbo al micro.

REALIZZAZIONE DEL CIRCUITO

Per costruire il TiDiGino serve un circuito stampato a doppia faccia con fori metallizzati, sul quale abbiamo previsto di usare quasi esclusivamente componenti SMD. Il primo componente da montare è il microcontrollore, che sugge-

riamo di posizionare ben centrato sulle piazzole corrispondenti e di fermare stagnando leggermente un elettrodo per lato; nel fare le saldature, raccomandiamo di prestare molta attenzione a non cortocircuitare, con lo stagno, piedini vicini (il passo è molto ridotto). Utilizzate la stessa tecnica anche per gli altri circuiti integrati.

Il microcontrollore può anche essere saldato con una stazione ad aria calda, di quelle usate per il rework

degli SMD; in tal caso dovete sciogliere dello stagno su ogni piazzola del c.s. riservata ad esso, quindi, passate tutte le piazzole, deporre del fluxante per saldatura di SMD (va bene anche quello dei BGA...) ed appoggiare il chip centrandolo bene, quindi scaldare con l'aria il perimetro dove si trovano gli elettrodi, badando di tenere la velocità bassa in modo da non spostare il componente. La temperatura deve essere impostata a circa 330°C

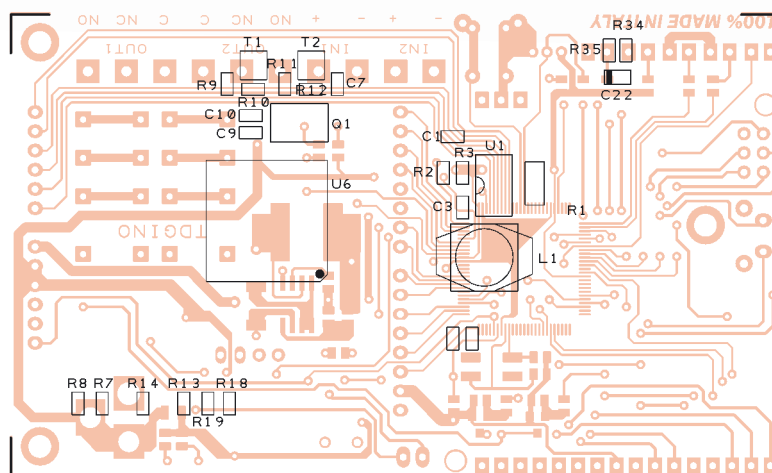
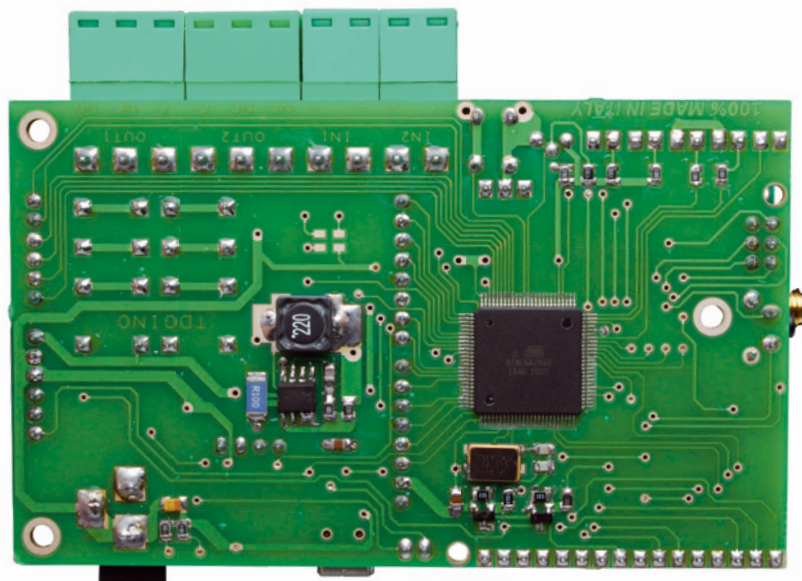
C22: 4,7 μ F 6,3 VL tantalio (R)
 C23÷C25: 100 nF multistrato (0805)
 Q1: Quarzo 16 MHz (C7S)
 Q2: Quarzo 3,579545 MHz
 (HC49/4H SMX)
 U1: MC34063AD
 U2: DS18B20+
 U3: 24FC256-SN
 U4: TLP181
 U5: TLP181
 U6: ATMEGA2560-16AU (MF924)
 U7: MT88L70AS
 U8: FT232RL
 U9: TC1262-3.3 (SOT-223)
 GSM: TDG GSM_900
 D1: GF1M-E3
 D2: MBR5140TRPBF
 D3: GF1M-E3
 D4: GF1M-E3
 T1: BC817
 T2: BC817
 LD1, LD2: LED 3 mm rosso
 LD3, LD4: LED 3 mm giallo
 LD5: LED 3 mm verde
 LD6: LED giallo (0805)
 LD7: LED rosso (0805)
 L1: Bobina 22 μ H
 RL1, RL2: Relé 5V 2 scambi
 P1: Microswitch
 P2: Microswitch 90°
 F1: Fusibile ritardato 2 A (1206)

Varie:

- Morsettiera 2 poli (2 pz.)
- Morsettiera 3 poli (2 pz.)
- Connettore Mini-USB
- Plug alimentazione
- Jumper
- Strip maschio 2 poli
- Strip maschio 3 poli (2 pz.)

- Strip maschio 4 poli
- Strip femmina 3 poli
- Strip femmina 6 poli (2 pz.)

- Strip femmina 8 poli (2 pz.)
- Strip femmina 16 poli
- Circuito stampato



(salite se vedete che lo stagno fatica a dissaldarsi) ed occorre ripassare i contatti fin quando non si vede la lega saldante divenire lucida ed il micro affondare nello stagno. L'operazione riesce meglio se si può scaldare inferiormente la basetta, ad esempio disponendola sopra un getto d'aria calda o una bistecchiera elettrica, ovvero scandandone la superficie per un po' mediante un asciugacapelli prima di appoggiare l'integrato e proce-

dere alla saldatura con la stazione ad aria calda. Completate le saldature, si può passare ai componenti a montaggio normale, ossia LED, pulsante, pin-strip, sonda DS18B20 e relé, iniziando con quelli a più basso profilo e rispettando le polarità indicate dai disegni di montaggio illustrati in queste pagine. Sistemato il tutto, dovete inserire a fondo negli appositi connettori il modulo contenente il cellulare GSM, che si acquista

già pronto dalla Futura Elettronica. Presso Futura Elettronica è anche disponibile il TiDiGino già montato (cod. FT924M, Euro 138,00). Per far funzionare il telecomando occorre un alimentatore capace di erogare da 5 a 32 volt - 1 A.

FUNZIONI

Il telecomando può essere gestito tramite comandi inviati da SMS, ma è anche possibile comandarlo attraverso la porta seriale (attual-

mente collegata al convertitore USB) solo che in questo caso occorre farli precedere dal numero chiamante (basta un numero fittizio) perché il firmware si aspetta che nei comandi sia sempre contenuto l'identificativo telefonico da cui provengono.

Ad ogni comando segue una risposta (via SMS) diretta al numero chiamante verso il telefono originante, ma la risposta automatica può essere disabilitata; le risposte possono anche essere intercettate lungo la seriale. Inoltre è previsto un funzionamento come allarme con l'invio automatico di SMS o chiamate vocali, sulla base di condizioni su ciascuno dei due input. Il telecomando può essere utilizzato anche come apricancello, comandato chiamandolo da un numero di telefono preventivamente

abbinato: non serve attendere alcuna risposta (che peraltro il circuito non dà, così da non farci spendere soldi) perché basta uno squillo. Nella modalità DTMF, il telecomando può essere comandato da un telefono a bitoni in multifrequenza, quindi da qualsiasi cellulare o apparecchio di rete fissa costruito

a partire da una ventina d'anni fa. Oltre a ciò, il circuito può funzionare da termostato, gestendo un impianto di climatizzazione in modo autonomo grazie al rilevamento della temperatura effettuato con la sonda DS18B20, o dietro impostazione da remoto. Riepilogando, sono previste le seguenti funzionalità:

- impostazione; comprende i comandi che inseriscono o cancellano i numeri abilitati, o abilitano su questi numeri la chiamata o l'invio di SMS in caso di allarme;
- gestione operativa del telecomando; riguarda i comandi che definiscono il comportamento del telecomando, mediante la definizione di alcune proprietà, oppure che attivano direttamente i relè di output o leggono lo stato degli input optoisolati;
- teleallarme; al raggiungimento della soglia stabilita (bassa o alta), il telefono abilitato viene raggiunto da una chiamata normale o da un SMS;
- apricancello; all'arrivo della chiamata da un telefono abilitato viene attivato il relè;
- telecomando DTMF; all'arrivo della chiamata da un telefono abilitato, viene aperta la linea

di comunicazione ed il telecomando si mette in ascolto, quindi ad ogni bitono ricevuto, viene attivato il comando associato e viene risposto con una sequenza di toni prestabilita (se non viene chiusa manualmente, la comunicazione viene interrotta automaticamente dopo un certo periodo).

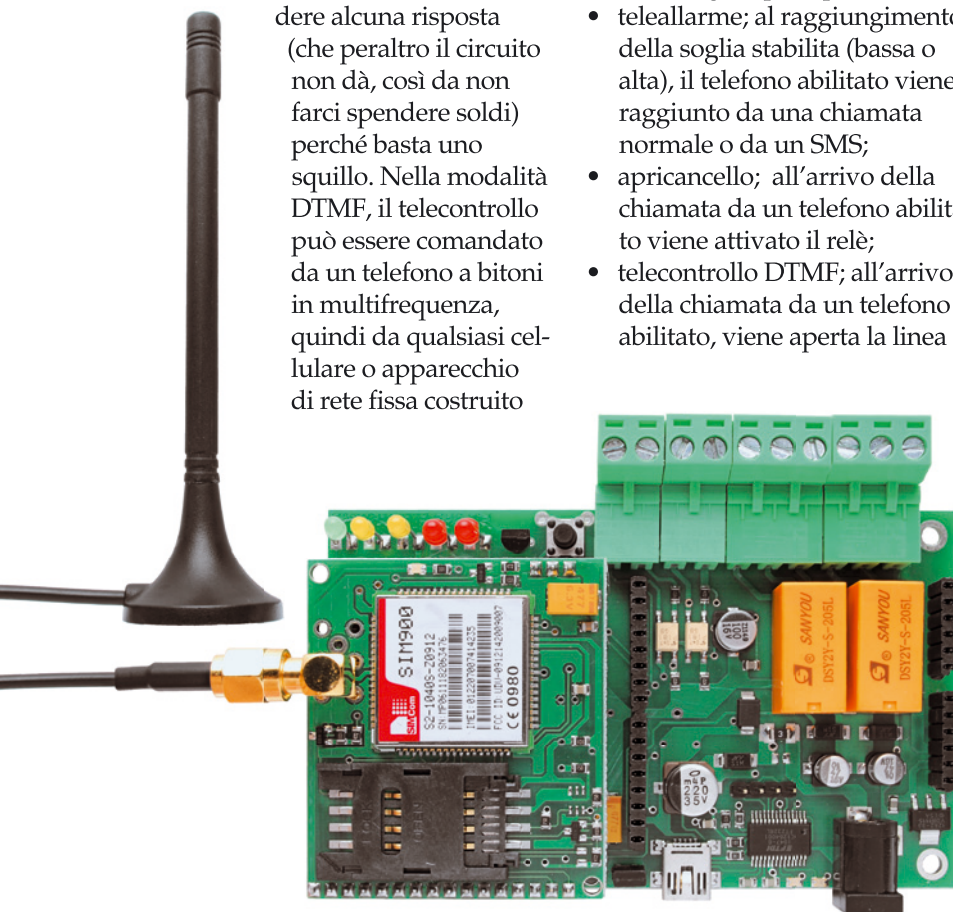
COMANDI E LORO FORMATO

Spieghiamo adesso i comandi previsti dal firmware caricato nell'ATmega 2560: parliamo del firmware inviato dal Sig. Daniele Denaro, vincitore del TiDiGino Contest, che trovate nel nostro sito alla pagina dedicata al Contest. I comandi SMS hanno il seguente formato: codice [parametro] [: password].

Ad esempio: **NUM +39333333333 : 123456** "abilita numero manager" imposta un numero di quelli abilitati a dare i comandi. **OU2 ON** "attiva relè 2". Ancora, **ST?** chiede lo stato dell'uscite e degli ingressi. Nei comandi da seriale, bisogna mettere davanti il numero chiamante che deve finire con cancelletto: **+39...# codice [parametro] [: password]**. Ad esempio, **+3933333333# PWD 567890 : 123456** cambia la password, mentre **+3933333333# OU1 OFF** disattiva il relè 1 e **+3933333333# AL?** chiede per quali numeri gli allarmi sono abilitati.

I numeri telefonici devono iniziare tutti con il prefisso nazionale (+39 per l'Italia), perché è così che il numero chiamante viene identificato dal modem.

I codici sono tutti formati da acronimi di 3 caratteri fissi. Gli spazi nel formato ci possono essere o mancare (es.: OU1OFF è ammesso). Inoltre maiuscolo o minuscolo non hanno importanza (case-insensitive). È possibile inserire più comandi (fino a 10) in un SMS separandoli con il



carattere “;” (ciò vale solo da SMS, non da seriale); ad esempio: **NUM +39333333333:123456;NUM +39222222222:123456;OU1 ON;OU2 OFF.**

Le password previste sono due: una per l'impostazione del telecontrollo ed una per caricare un numero di telefono nella lista degli abilitati alla funzione apricancello. Ambedue le password devono essere formate da un minimo di 4 e un massimo di 6 caratteri; quella per l'apricancello non può essere “000000”, che corrisponde alla password predefinita per tale funzione. Invece la predefinita per l'impostazione è “123456”.

Sono previste due liste di numeri telefonici (lunghe al massimo 16 caratteri compreso il prefisso di nazione): una per un massimo di 8 telefoni abilitati all'impostazione (telefoni manager) ed una per il comando dell'apricancello (contiene fino a 200 numeri).

Per ogni numero di telefono manager può essere abilitato un flag per l'avviso automatico di allarme sia via SMS che con chiamata diretta; in caso di allarme, alla connessione l'utente chiamato udrà una serie di toni per circa 15 s e successivamente la chiamata verrà terminata.

Per quanto riguarda la segnalazione via SMS, riceverà un messaggio il cui testo standard può essere modificato a piacimento anche ingresso per ingresso.

I comandi principali come l'inserimento dei numeri manager abilitati, la modifica della password o l'abilitazione alla chiamata d'allarme, devono essere completati dalla password, ma possono provenire da qualunque telefono. Esiste, comunque, una modalità di extra sicurezza (normalmente disabilitata) che richiede sempre l'invio da un telefono abilitato, ma esige che il comando contenga la password. Gli altri comandi di gestione “operativa” non hanno bisogno

della password ma devono essere inviati da telefoni abilitati. Per la sola funzione apricancello, è stata prevista una seconda password da aggiungere solo nei comandi di aggiunta di un numero alla lista di quelli abilitati al comando; tale password permette di inserire il numero del telefono da cui proviene il comando direttamente nella lista. Con essa è possibile utilizzare i soli comandi di inserimento e cancellazione dalla lista del proprio numero.

Per disabilitare questa funzionalità basta porre tale password a “000000”.

L'inserimento o cancellazione di telefoni nella lista per apricancello rimane comunque sempre possibile da un telefono manager, fornendo come parametro del comando il numero da abilitare. Avere due password permette di lasciare quella di gestione nelle mani del gestore del sistema e assegnare quella di inserimento dei numeri dell'apricancello limitata a tale funzione, così da rendere l'apparato più sicuro.

Il comando da DTMF, ricordiamo che può essere effettuato solo dagli 8 numeri della lista manager; i comandi sono i seguenti:

1. Out1 On
2. Out2 On
3. Out1 Off (inattivo se Out1 monostabile)
4. Out2 Off (inattivo se Out2 monostabile)
5. Legge stato Out1
6. Legge stato Out2
7. Legge stato In1
8. Legge Stato In2

Per fare un po' d'ordine, va detto che i comandi si dividono in due grandi categorie: comandi di amministrazione e di setting; i primi possono giungere da qualunque numero ma devono contenere la password o, se è attiva l'extra sicurezza, oltre ad avere la password

devono provenire da telefoni abilitati.

I comandi di amministrazione riguardano le impostazioni valide per tutte le funzionalità del telecontrollo. Sono comandi di amministrazione:

- Cambia password manager;
- Cambia password per inserimento numero per apricancello (se “000000” disabilita inserimento);
- Aggiungi telefono abilitato manager;
- Cancella telefono abilitato manager;
- Set allarme SMS su telefono nnn;
- Reset allarme SMS su telefono nnn;
- Set allarme VOC su telefono nnn;
- Reset allarme VOC su telefono nnn;
- Aggiungi numero abilitato per apricancello;
- Cancella numero abilitato per apricancello.

Esistono poi i seguenti comandi di amministrazione che non richiedono password, ma devono giungere solo da numeri manager:

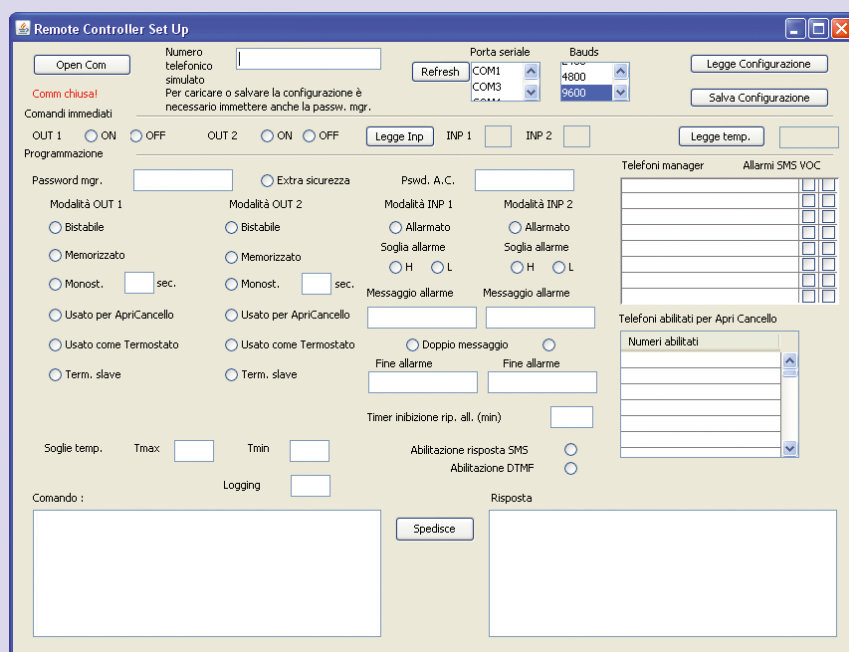
- Cancella tutti i numeri abilitati per apricancello;
- Modifica velocità seriale;
- Abilita o disabilita la risposta tramite SMS;
- Lista telefoni manager;
- Lista telefoni abilitati per apricancello;
- Lista completa del setting del telecontrollo;
- Inserimento numero apricancello.

Quanto ai comandi di setting, riguardano l'utilizzo e la definizione dei parametri di funzionamento della singola modalità; devono essere impartiti solo dai telefoni manager e non devono contenere la password.

TiDiGino Contest: il progetto vincente

Come potete vedere sul nostro sito e sul nostro blog (<http://blog.elettronica.in>) sono giunte solamente tre soluzioni

al nostro TiDiGino Contest, tutte peraltro molto valide e complete, spiegate accuratamente in ogni dettaglio e tutte meritevoli



Schermata dell'applicazione in Java.

di vincere. Per questo motivo abbiamo deciso di premiare la più tempestiva, riservando agli altri partecipanti un premio di consolazione, non previsto inizialmente dal nostro Contest. Abbiamo dunque deciso di premiare la soluzione inviata dall'Ing. Daniele Denaro, il quale oltre al firmware ha preparato un'applicazione scritta in Java e utilizzabile in ambiente Microsoft Windows, per gestire il telecomando anche localmente via porta seriale.

Il firmware è ben articolato e consente di svolgere tutte le funzioni previste dal sistema, ossia il telecomando a due ingressi e due uscite, il telecomando DTMF, l'apricancello ed il termostato GSM. Trovate l'intero firmware commentato e tutto il materiale realizzato dal Sig. Denaro nella sezione del nostro sito dedicata al TiDiGino Contest.

Quanto all'applicazione Java, si avvia molto semplicemente facendo doppio clic sull'icona corrispondente e funziona, chiaramente, se nel computer è installato l'ambiente Java. Una volta avviata, presenta la finestra di lavoro, che contiene caselle e pulsanti per le impostazioni e i comandi del caso; tale finestra è utilissima, oltre che per testare il funzionamento del telecomando, per eseguire da PC quelle configurazioni preventive all'installazione, che fatte da cellulare inviando

Questi comandi sono a loro volta ripartiti in base alle funzionalità del telecomando. Vediamo per primi quelli della funzione di telecomando pura e semplice:

- Lettura stato in/out;
- Out 1 On/Off;
- Out 2 On/Off;
- Modalità out 1 MONO/BIST/MEM/NOMEM;
- Modalità out 2 MONO/BIST/MEM/NOMEM;
- Timer monostabile out 1 : secondi;
- Timer monostabile out 2 : secondi;
- Lettura setting out;
- Allarme abilitato inp 1 On/Off;
- Allarme abilitato inp 2 On/Off;
- Set soglia allarme inp 1 H/L;
- Set soglia allarme inp 2 H/L;
- Timer inibizione seconda chiamata allarme;
- Testo messaggio allarme inp 1;
- Testo messaggio allarme inp 2;
- Lettura setting allarmi.

Ci sono poi i comandi di setting dell'apricancello:

- Abilita out1;

- Abilita out2;
- Lettura out abilitati.

Quanto al funzionamento da termostato, i comandi di setting previsti sono:

- Lettura temperatura;
- Abilita out1;
- Abilita out2;
- Set soglia alta temperatura (riferimento riscaldamento);
- Set soglia bassa temperatura (riferimento raffreddamento);
- Riscaldamento On/Off (manuale);
- Raffreddamento On/Off (manuale);
- Abilita controllo esterno riscaldamento On/Off;
- Abilita controllo esterno raffreddamento On/Off;
- Lettura setting out come termostati;

FUNZIONAMENTO DEL TELECONTROLLO

Passiamo ora alla descrizione del funzionamento del sistema in base al firmware del Sig. Denaro: all'accensione (start-up) il LED giallo (LD3) si illumina e, suc-

cessivamente, all'accensione del cellulare il LED verde comincia a lampeggiare. Infine, quando parte il ciclo operativo che rileva i comandi, lampeggia anche il diodo giallo, segnalando ad ogni lampeggio il completamento di un ciclo operativo.

Se ciò non accade, resettate il circuito con P1 e verificate nuovamente. Nel caso sia stato fatto uno start-up con reset (premendo il pulsante P2) LD3 si mette a lampeggiare velocemente per 4 secondi, segnalando il ripristino dei valori predefiniti.

Il pulsante P2 è stato utilizzato per il reset delle impostazioni: se allo startup (all'accensione o dopo che è stato premuto il pulsante P1), viene rilevata la pressione di tale pulsante, il micro carica i valori predefiniti e cancella le impostazioni preesistenti. Questo reset può essere ordinato anche via SMS, con l'apposito comando (proveniente da uno degli 8 numeri manager) purché contenga la password e il parametro "CONFERMA". P2 non va confuso con P1, che

SMS costano tempo e sono peraltro piuttosto scomode. Dalla finestra di dialogo dell'applicazione è possibile anche verificare l'attuale configurazione; allo scopo è necessario inserire la password manager nell'apposita casella. La schermata è costituita da quattro sezioni:

- quella in alto è dedicata a stabilire la comunicazione e a leggere o salvare la configurazione;
- la zona sotto (Comandi immediati) contiene i comandi utili per verificare l'hardware;
- la zona Programmazione contiene tutte le variabili ed i flag (alcuni mutualmente esclusivi) di configurazione, che possono essere caricati, aggiornati e salvati;
- la parte in basso riguarda l'invio dei comandi, ossia dei comandi di configurazione e di tutti quelli previsti dal sistema; nella casella a sinistra si può scrivere il comando (rammentando che se si necessita di un numero telefonico abilitato, questo deve essere inserito correttamente nel campo predisposto al numero telefonico simulato) e in quella a destra; facendo clic sul pulsante "Spedisce", appare il messaggio di risposta del TiDiGino.

L'applicazione salva alcuni dati (seriale utilizzata, baud e numero fittizio) su un file di caratteristiche che sarà letto al successivo avvio.

è il reset del microcontrollore: quest'ultimo non azzerava le configurazioni, che restano in EEPROM; fa solo ripartire il programma. I relé di uscita possono funzionare in modo sia impulsivo che bistabile; in quest'ultima modalità, in caso di blackout al ritorno dell'alimentazione il relé viene ripristinato nello stato in cui si trovava prima della mancanza di tensione. Ciascuna delle uscite può essere assegnata alla funzione apricancello, funzionando in monostabile con tempo d'impulso pari a 100 ms. Al funzionamento da termostato può essere assegnato il relé 1 per il comando della caldaia e il relé 2 per la gestione del raffrescamento; entrambi i relé funzioneranno in modo bistabile ma senza ripristino in caso di blackout. Al raggiungimento della temperatura di riferimento, per il riscaldamento si attiva il rispettivo relé se quella misurata dal sensore del telecomando è inferiore, mentre si disattiva nel caso opposto; è prevista un'isteresi di $\pm 0,5$ °C. Il raffrescamento funziona all'op-

posto. Come previsto dal nostro telecomando GSM, i due relé possono essere comandati anche direttamente da un interruttore, timer o termostato esterno, collegato agli ingressi, fermo restando che se il comando proviene da IN1 viene comandato OUT1 e se arriva da IN2 si attiva OUT2.

Per le funzioni di telecomando, ogni ingresso può essere considerato attivo a livello alto o basso, in base all'impostazione; è inoltre possibile impostare un tempo di inibizione, che permette di ignorare segnali consecutivi per un certo intervallo.

Chiudiamo la descrizione del funzionamento del sistema con la connessione seriale: per come funziona Arduino, se non viene rimosso JP2, ad ogni apertura della comunicazione viene effettuato un reset del microcontrollore, dato che il ponticello connette il DTR al reset dell'ATmega. Quindi se comandate il circuito serialmente dovete rimuovere il ponticello. Nei sorgenti del firmware è stato predisposto un secondo collegamento seriale (attualmente non attivo) che potrebbe essere utilizzato per un collegamento Wi-Fi o Bluetooth.

Infine è presente un'attività di logging che attualmente è indirizzata sulla porta seriale USB (logway=1). In questo log viene registrata sinteticamente l'attività. Il log può essere disabilitato (logway=0). Notate che la lista degli 8 telefoni manager è duplicata in RAM, mentre quella dei 200 telefoni per Apricancello esiste solo in EEPROM per evitare la saturazione della RAM (con conseguenti problemi di stack). La differente velocità di accesso è assolutamente ininfluente nell'utilizzo del controller. In ogni caso la scrittura su EEPROM è condizionata alla effettiva variazione, per prolungarne la vita operativa. Ovvero un

byte è riscritto solo se è veramente cambiato.

APPLICAZIONE JAVA

L'autore del firmware ha anche realizzato un'applicazione Java in modalità grafica, per la gestione semplificata del sistema tramite collegamento USB: oltre a permettere l'utilizzo dei comandi per via seriale, essa utilizza un protocollo dedicato per la lettura ed il successivo caricamento della configurazione eventualmente modificata. L'applicazione può essere utilizzata solo in ambiente Windows, perché richiede una dll (necessaria al collegamento seriale) non compresa nelle distribuzioni Java. Trovate l'applicazione nel nostro sito, con allegati due file: TiDiGinoW32.jar (per Windows a 32 bit) e TiDiGinoW64.jar (Windows 64 bit); per poterla utilizzare occorre aver installato l'ambiente Java ed in questo caso quindi basta cliccare sul nome del file. Se, per qualche motivo, i file jar non fossero associati a Java, si può aprire una finestra console e lanciare il comando: `Java -jar TiDiGinoW32.jar` (o W64). ■



per il MATERIALE

Tutte le librerie, gli sketch ed i programmi sviluppati per questo dispositivo sono scaricabili gratuitamente dal sito e dal blog della rivista. Ricordiamo che il telecomando TiDiGino montato e collaudato (cod. FT924M) costa 138,00 Euro, IVA compresa. Il dispositivo comprende l'antenna GSM Quadriband ma non l'alimentatore da rete (disponibile separatamente, cod. AL08-12N, Euro 8,00).

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

Cerchi sicurezza e affidabilità?

Ecco le nostre soluzioni per creare facilmente il tuo impianto di videosorveglianza



cod. CCTVPROM10
€ 469,00

**Set 4 telecamere,
DVR e monitor LCD 7" integrato**

Sistema composto da:

- 1 pz DVR 4 canali H.264 (con monitor LCD a colori, HDD 500GB SATA, integrati e telecomando IR),
- 1 pz Mouse USB,
- 4 pz Telecamere CMOS a colori con LED IR,
- 4 pz Cavi alimentazione/video da 18 metri
- 1 pz Adattatore da rete 12Vdc.



cod. FR368
€ 499,00

Set 2 telecamere, DVR e monitor LCD 7"

Sistema composto da:

- 1 pz DVR 4 canali H.264 con telecomando
- 1 pz Hard Disk 320 GB SATA
- 1 pz Monitor TFT 7" con alimentatore
- 2 pz Telecamera a colori da esterno con IR
- 2 pz Adattatore BNC maschio / RCA femmina
- 1 pz Adattatore BNC maschio / RCA maschio
- 2 pz Cavo video con plug d'alimentazione 10m



cod. FR460
€ 530,00

Set 2 telecamere, DVR e monitor LCD 19"

Sistema composto da:

- 1 pz DVR 4 canali H.264 con telecomando
- 1 pz Hard Disk 500 GB SATA
- 1 pz Monitor LCD 19" TFT
- 2 pz Telecamera con IR da esterno
- 2 pz Cavo video con plug d'alimentazione (10m + 20m)
- 2 pz Adattatore BNC maschio / RCA femmina.



**Set 4 telecamere
e DVR con mouse**

Sistema composto da:

- 1 pz DVR 4 canali H.264 con telecomando e mouse USB
- 1 pz Hard Disk 500 GB SATA
- 4 pz Telecamera a colori da esterno con IR
- 4 pz Cavo video con plug d'alimentazione 10m



cod. FR484
€ 399,00



cod. CP297
€ 58,00

**TELECAMERA
CCD, 480 LINEE TV,
IP65 CON AUDIO E LED IR**



**TELECAMERA CCD, 650 LINEE TV,
IP66 CON LED IR**

€ 169,00
cod. CP481



**TELECAMERA DOME CCD
ANTIVANDALO, 420 LINEE TV,
IP66 CON LED IR**

€ 114,00
cod. CP482



cod. CP356
€ 64,00

**TELECAMERA
CCD, 480 LINEE TV
CON OBIETTIVO DA 4 mm**

ACCESSORI:

CAVO VIDEO 2 RCA + PLUG ALIMENTAZIONE - 10 METRI ■ cod. VP001-10 ■ € 7,00
CAVO VIDEO + PLUG ALIMENTAZIONE - 10 METRI ■ cod. VP002-10 ■ € 7,00

Pur essendo basato su un RX per dati, sfrutta la possibilità di prelevare il segnale all'uscita del demodulatore in modo da essere usato per ricevere segnali BF trasmessi da qualsiasi ricevitore operante sulla frequenza su cui è accordato. Dispone di amplificatore di potenza e regolazione del volume.



RICEVITORE AM PER MICROSPIA UHF

di DAVIDE SCULLINO

I moduli a radiofrequenza prodotti dalla Aurel per l'uso nei radiocomandi ed in generale nella comunicazione dati, possono essere utilizzati, adottando opportuni accorgimenti, anche per trasmettere e ricevere segnali audio, quindi per impieghi lineari; i TX, per esempio, possono trasmettere segnali vocali semplicemente mantenendone l'ingresso dati polarizzato con

un potenziale che permette di attivare lo stadio oscillatore RF, quindi sovrapponendo la componente di segnale BF. Quanto ai ricevitori, quelli della serie RF290A (quindi l'obsoleto a 300 MHz e gli RF290A5 a 433,92 MHz) e i BC-NBK dispongono di due uscite: una è quella corrispondente al demodulatore AM e l'altra quella il cui segnale viene squadrato da un comparatore; ebbene, usan-

CARATTERISTICHE TECNICHE

- Frequenza di lavoro: 433,92 MHz
- Sensibilità RF: 106 dBm
- Potenza di uscita audio: 1 W su 8 Ω
- Tensione di alimentazione: 9÷12 Vcc
- Corrente assorbita a 1 W/ Ω : 330 mA
- Risposta in frequenza: 100Hz÷5 kHz

do quest'ultima e amplificandone il segnale con un amplificatore di piccola potenza, nasce il ricevitore descritto in queste pagine, che è stato pensato per utilizzare i più recenti moduli RF Aurel siglati AC-RX2.

In pratica questo nuovo circuito sostituisce il ricevitore che già avevamo proposto anni fa (FT98RX) nel fascicolo n° 2 del lontano 1995 e nasce per una precisa ragione: l'RF290A5-433 che equipaggiava tale progetto, tra breve non sarà più disponibile; ecco perché abbiamo riprogettato il ricevitore usando l'AC-RX2, che è uno dei più recenti ricevitori ibridi Aurel.

La frequenza di lavoro della microspia resta invariata, in quanto questo nuovo ricevitore opera anch'esso sui 433,92 MHz; ciò permette la piena compatibilità con il trasmettitore originale, che

si basa sul TX433SAW, il quale opera sui 433,92 MHz. Comunque nulla vieta di impiegare il ricevitore per altri scopi o abbinandolo a trasmettitori diversi dal nostro; l'unica condizione è che operino sulla stessa frequenza d'accordo e che modulino la portante RF in ampiezza, altrimenti la ricezione è impossibile.

Vediamo adesso com'è fatto il circuito del ricevitore, partendo dalla premessa che per ascoltare a distanza quanto captato dal microfono della trasmittente bisogna disporre di un ricevitore AM capace di sintonizzarsi sui 433,92 MHz.

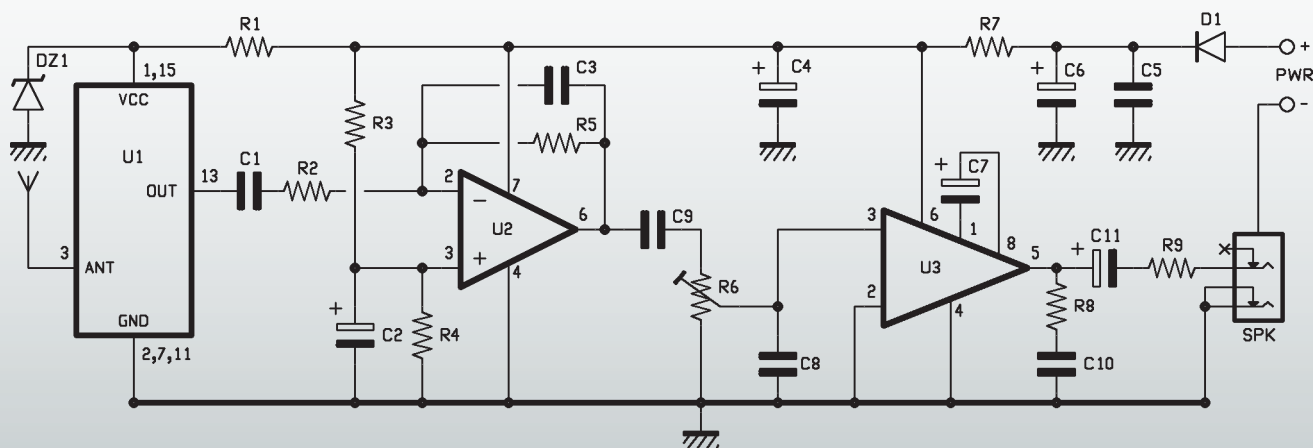
SCHEMA ELETTRICO

In buona sostanza, la nostra unità ricevente è composta da tre blocchi: il primo è un ricevitore ibrido per dati di tipo AC-RX2; il secondo è il preamplificatore audio ed il terzo un finale, sempre audio, capace di erogare circa un watt di potenza, quindi quanto basta per pilotare un piccolo altoparlante generico o una cuffia. Il ricevitore ibrido è un super-rigenerativo SMD molto sensibile (presenta una sensibilità di ben -106 dBm) a basse emissioni

spurie, accordato a 433,92 MHz e provvisto di demodulatore AM e squadratore del segnale di uscita; come accennato, per la nostra applicazione preleviamo il segnale dal piedino 13 (uscita del demodulatore d'ampiezza) e non dal 14 (come si userebbe nelle applicazioni dati) in modo da prelevare il segnale di bassa frequenza che ha modulato il trasmettitore e perciò l'audio originario.

Questo segnale lo mandiamo all'operazionale U2, che lo amplifica e lo filtra sopprimendo i disturbi impulsivi ad alta frequenza che sfuggono al ricevitore o che esso capta nell'etere, inevitabili in quanto il difetto della modulazione d'ampiezza è che dal demodulatore passano i disturbi radioelettrici, udibili sotto forma di crepitii. Ad essi va aggiunto il rumore elettrico di fondo. U2 è il "sempreverde" LM741, storico operazionale che in questo caso viene utilizzato come amplificatore non-invertente alimentato a tensione singola; per questa ragione vedete che il piedino 3 è alimentato con metà del potenziale dell'alimentazione mediante il partitore resistivo formato da R3 ed R4 (C2 filtra i

[schema ELETTRICO]



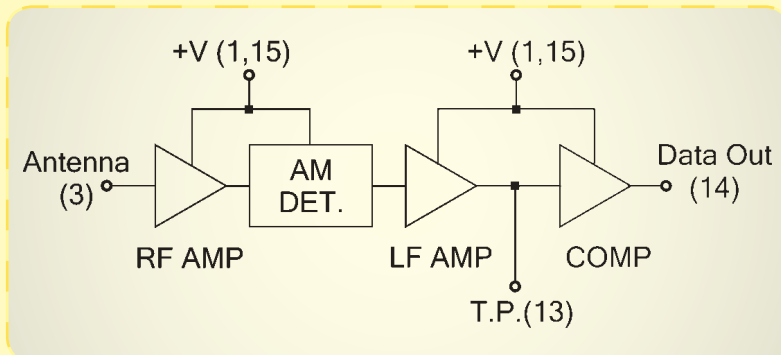
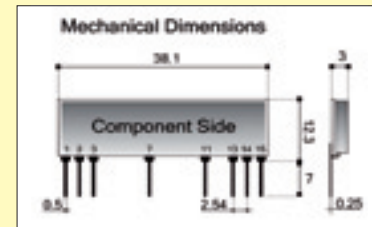
Il modulo AC-RX2

disturbi introdotti dalla linea di alimentazione) che ha lo scopo di mantenere, a riposo, il piedino (6) allo stesso livello. A ciò concorre il condensatore C1, che in continua si presenta come un circuito aperto e riduce il guadagno dell'operazionale ad 1; in presenza di segnale entro la banda audio, considerata trascurabile la reattanza del predetto C1, il guadagno dell'U2 vale $G=(R5+R2)/R2$, ossia circa tre volte. Note che l'azione di filtro è esercitata dal condensatore C3, il quale al crescere della frequenza riduce il guadagno dell'operazionale.

Notate ancora che l'aver portato a riposo la tensione di uscita a metà di quella di alimentazione, consente l'oscillazione del segnale uscente dal pin 6 tra valori più alti o più bassi, seguendo l'andamento dei segnali alternati, che oscillano intorno allo zero. In questo caso il segnale non è alternato ma conserva lo stesso andamento e, grazie al condensatore C9, si presenta ai capi del trimmer di regolazione del volume (R6) come se lo fosse. L'ultimo stadio del nostro circuito è U3, un amplificatore anch'esso classico: si tratta dell'LM386, che contiene un completo amplificatore monolitico capace di ottenere fino ad 1 watt di potenza, usato nel nostro caso come mini-finale di bassa frequenza; alla sua uscita possiamo collegare una cuffia a bassa impedenza (32 ohm, per esempio) o, cortocircuitando la resistenza R9, un altoparlante da 8 ohm, 1 watt. L'intero circuito funziona con la tensione continua fornita ai punti + e - PWR: tipicamente sono richiesti 9 volt, ma si può arrivare fino a 12 V; il diodo D1 protegge il circuito dall'inversione di polarità, mentre i condensatori C5 e C6 filtrano l'alimentazione

Il modulo AC-RX2 è un ricevitore superrigenerativo a basso assorbimento, alta immunità ai disturbi di alimentazione e bassa radiazione in antenna, dotato di bobina variabile e accordato sulla frequenza dei 433,92 MHz. Come si vede nello schema in basso, il modulo comprende un amplificatore RF, un rivelatore AM, un amplificatore a bassa frequenza ed un comparatore di tensione che ha lo scopo di squadrare perfettamente il segnale

di uscita. Fortunatamente prima di questo stadio esiste un Test Point dove è possibile prelevare il segnale BF prima che questo venga reso digitale dal comparatore.



CARATTERISTICHE	MIN	TYP	MAX	UNIT
V _S Alimentazione	4,5	5	5,5	Vdc
I _S Corrente Assorbita		2,5	3	mA
F _W Frequenza di ricezione		433,92		MHz
S _i Sensibilità RF		-106		dBm
B _W Banda passante RF a -3dB		±2		MHz
S _o Onda quadra in uscita			3	KHz
S _L Radiazione in antenna		-65	-60	dBm
H _o Livello alto d'uscita	V _S - 1,2			V
L _o Livello basso d'uscita			GND +0,4	V
T _{ON} Tempo di accensione			2	s
T _{OP} Temperatura di lavoro	-20		+80	°C

da disturbi ed eventuale ripple (tipico quando l'alimentazione è fornita da un alimentatore da rete). La resistenza R7 e il condensatore C4 operano da ulteriore filtro contro i disturbi, specie quelli impulsivi, proteggendo i circuiti audio dal rischio di auto-oscillazione. La polarizzazione del modulo radio è assicurata dal diodo Zener DZ1, che, aiutato dalla resistenza zavorra R1, fissa il potenziale sui piedini 1 e 15 dell'ibrido a 5,1 V.

REALIZZAZIONE E COLLAUDO

Passiamo adesso alla parte pratica del circuito, la prima fase della

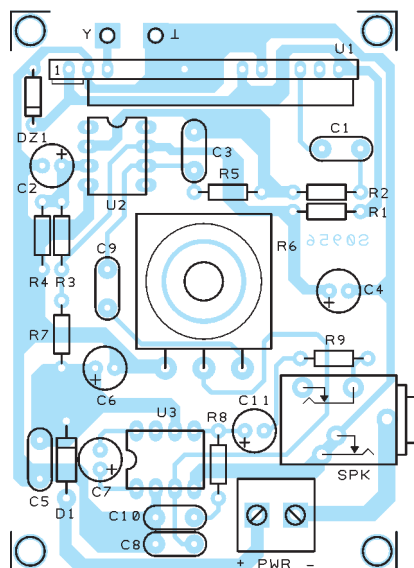
quale riguarda lo stampato, di cui trovate la traccia lato rame sul nostro sito www.elettronica.in.it. Ottenuta la basetta, montatevi i componenti iniziando con le resistenze e con il diodo Zener (attenzione al verso di inserimento: la fascetta colorata indica il terminale di catodo) e proseguendo con gli zoccoli a 4+4 pin per l'operazionale e l'LM386; per non avere problemi quando inserirete gli integrati, fate coincidere la tacca di riferimento ricavata nello zoccolo con il segno visibile nella serigrafia (piano di montaggio). Montate poi il trimmer (che pote-

Elenco Componenti:

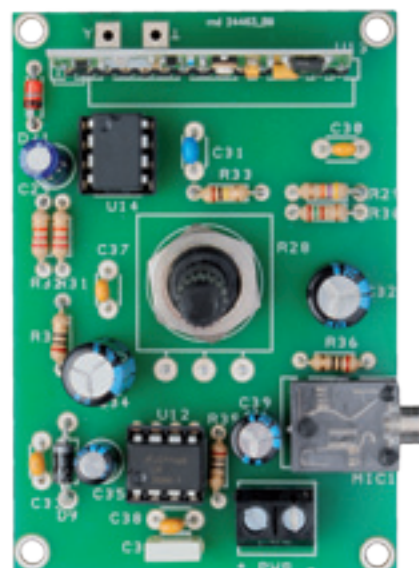
R1: 150 ohm
 R2: 47 Kohm
 R3, R4: 22 Kohm
 R5: 100 Kohm
 R6: 47 Kohm potenziometro
 R7÷R9: 10 ohm
 C1: 100 nF multistrato
 C2: 10 μ F 16 V elettrolitico
 C3: 2,2 nF ceramico
 C4: 470 μ F 16 V elettrolitico
 C5: 100 nF multistrato
 C6: 470 μ F 16 V elettrolitico
 C7: 1 μ F 16 V elettrolitico
 C8: 1 nF 63 VL poliestere
 C9, C10: 100 nF multistrato
 C11: 220 μ F 16 V elettrolitico
 D1: 1N4007
 DZ1: Zener 5,1V 500 mW
 U1: AC-RX2
 U2: 741
 U3: LM386

Varie:

- Morsetto 2 poli (2 pz.)



- Zoccolo 4+4
- Contenitore in plastica con portatile
- Filo di rame rigido



- Manopola potenziometro
- Presa jack da stampato
- Auricolare mono
- Circuito stampato

te sostituire con un potenziometro lineare o logaritmico da montare in modo passante come mostrano le foto del prototipo che vedete in queste pagine) e quindi i condensatori, avendo cura di rispettare la polarità di quelli elettrolitici.

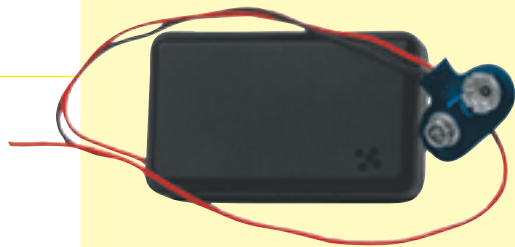
Montate quindi l'ibrido AC-RX2, infilandone i piedini nei rispettivi fori e tenendolo leggermente sollevato dal circuito stampato; non preoccupatevi del verso di inserimento, perché preparando lo stampato secondo la nostra traccia il modulo entra nei fori soltanto in un modo, quello giusto.

Per completare il circuito montate nelle piazzole + e - PWR una morsettiera bipolare a passo 5 mm e, infine, una presa jack da 3,5 mm stereo in corrispondenza dei fori siglati SPK; notate che della presa, i due contatti an-

teriori (canali sinistro e destro) sono tra loro cortocircuitati, in modo che se usate una cuffia stereo potete udire l'audio in entrambi i padiglioni. Se volete collegare un altoparlante, potete farlo sia mediante uno spinotto jack, sia saldandone i fili di collegamento direttamente ai contatti di massa e negativo del condensatore elettrolitico C11. Per utilizzare il circuito, inserite l'LM741 e l'LM386 nei propri zoccoli, avendo cura di posizionare correttamente il riferimento marcato sul suo corpo. Al punto ANT dovete collegare l'antenna ricevente del circuito: potete usare uno spezzone di filo elettrico lungo 18 cm o, meglio, connettere un BNC (il conduttore centrale alla piazzola che porta al piedino 3 dell'ibrido U1 e l'esterno alla massa dello stampato presa in prossimità delle

piazzole ANT) in cui inserirete un'antenna caricata in gomma, del tipo per UHF.

Per il collaudo prevedete un alimentatore da 9÷12 V capace di erogare 250÷330 mA (il circuito funziona eventualmente a batterie di stilo NiCd o NiMH o, ma con una limitata autonomia, con una pila piatta da 9 V) che collegherete con il positivo al +PWR e con il negativo alla massa (-PWR) del nostro circuito. Alimentate il trasmettitore e portate il perno del potenziometro di volume del ricevitore a metà corsa; a questo punto dovrete udire un fruscio di fondo nella cuffia o altoparlante che avete collegato all'uscita audio. Una volta attivato il trasmettitore, il fruscio deve smettere e in cuffia o altoparlante non dovrete udire alcunché. Parlando in prossimità del microfono del trasmettitore



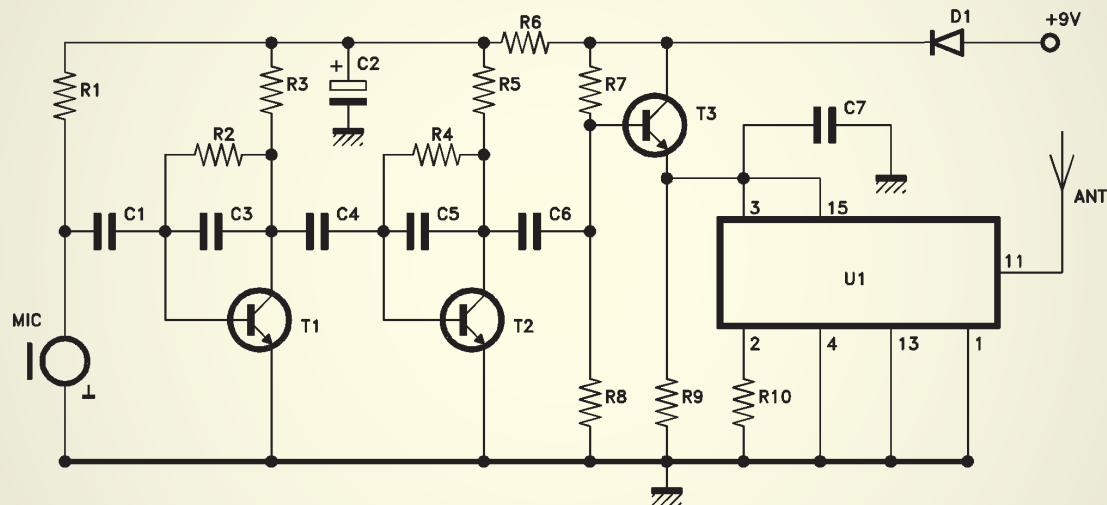
Il ricevitore qui descritto nasce per essere abbinato alla microspia FT98TX, anch'essa presentata sul fascicolo n. 2 di Elettronica In (settembre 1995) e basata sul modulo Aurel TX433SAW; nato per trasmettere segnali digitali, questo dispositivo può anche essere modulato con

Il trasmettitore adatto

segnali analogici giocando sulla tensione di alimentazione del modulo forzato in trasmissione o, più semplicemente, sulla polarizzazione del piedino di accensione dell'oscillatore.

Con tale tecnica è possibile modulare in ampiezza il segnale RF, potendo di fatto trasmettere segnali in banda audio. Nello specifico, il segnale audio captato da una capsula microfonica electret-condenser ed amplificato quanto basta mediante uno stadio a due transistor modula, mediante il transistor T3, l'alimentazione

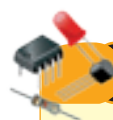
del modulo TX; quest'ultimo viene quindi acceso ma lavora ad una potenza ridotta. Alla tensione di alimentazione dei piedini 3 e 15 si sovrappone il segnale audio, il che determina la variazione della condizione di accensione del modulo TX e quindi della potenza RF irradiata dall'antenna; si ottiene, perciò, la modulazione di ampiezza voluta. Ciò consente, in abbinamento al ricevitore descritto in queste pagine, di realizzare un sistema via radio capace di coprire un raggio di circa 300 metri in campo aperto.



dovreste udire nel ricevitore quello che dite; ricordate di tenere sufficientemente distanti microfono e altoparlante, altrimenti si innescherà il feedback acustico e udrete il caratteristico fischio. Per evitare ciò, se avete i due circuiti molto vicini

preferite la cuffia all'altoparlante. Se necessario, agite sul controllo del volume (R6). Queste indicazioni valgono per l'abbinamento alla nostra microspia; se usate il dispositivo per ricevere il segnale di un altro trasmettitore, che magari diffonde audio

prelevato da una linea BF, non serve stare attenti al feedback acustico, ma solo regolare il livello audio nel TX.



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono di facile reperibilità. È in ogni caso disponibile il kit completo sia del ricevitore descritto in queste pagine (FT98RX, Euro 27,00) che del trasmettitore presentato a suo tempo (FT98TX, Euro 25,00). Entrambi i dispositivi comprendono il contenitore plastico ed 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>



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

**TRONCHESINO DI
PRECISIONE**



cod. VT100 € 2,50

**PINZA A BECCHI
SEMITONDI RICURVI**



cod. VT055 € 2,50

**SET 15 CACCIAVITI
ASSORTITI**



cod. VTSET15 € 11,00

**MINI TRAPANO PER
LAVORI DI PRECISIONE**



cod. VTHD01 € 7,00

**PINZA SPELAFILI
AUTOMATICA CON
TAGLIERINA**



cod. VTSTRIP5 € 7,00

**LENTE D'INGRANDIMENTO
5 DIOTTRIE CON
LAMPADA DA 22W**

cod. VTLAMP2WN
€ 39,00



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

LAMPADE UVA da 8 e 15W

cod. UVA8 € 5,00

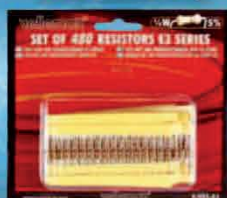
cod. UVA15 € 7,00

**STAZIONE
SALDANTE 48 W**

cod. VTSS4N € 16,00

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

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



cod. KRES-E3 € 7,00

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



cod. KRES-E12 € 7,20

**SET 120 DIODI
ASSORTITI**



cod. KDIODE1 € 7,80

**SET 100
TRANSISTOR**



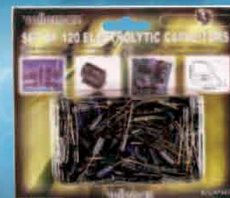
cod. K/TRANS1 € 13,00

**SET CONDENSATORI
CERAMICI**

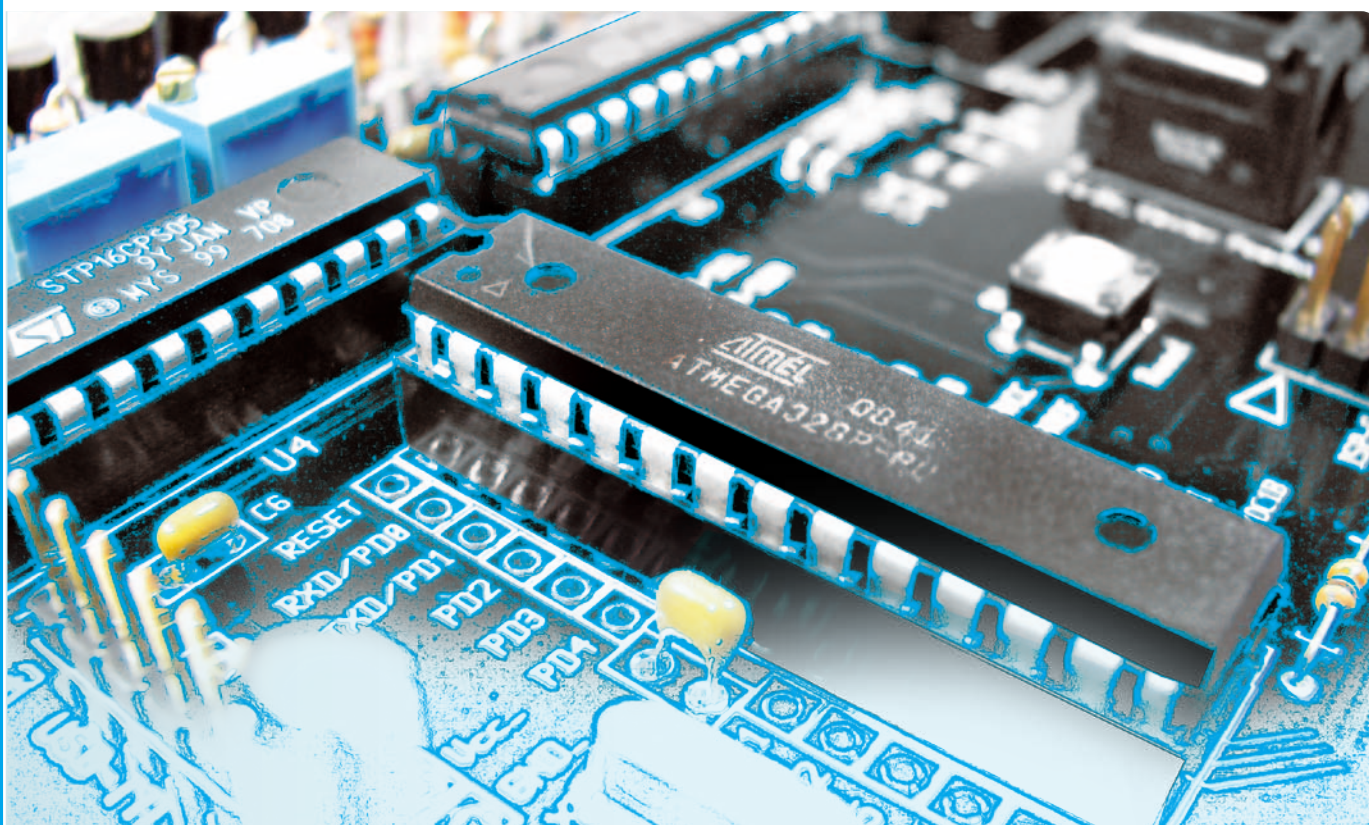


cod. KCAP1 € 11,00

**SET CONDENSATORI
ELETTROLITICI**



cod. KCAP2 € 11,00



ATMEL® OPEN SOURCE

Impariamo ad utilizzare i microcontrollori Atmel e il loro ambiente di sviluppo, partendo da nozioni propedeutiche all'apprendimento della programmazione e dell'uso dell'ATmega16. Prima puntata.

di OSVALDO SANDOLETTI

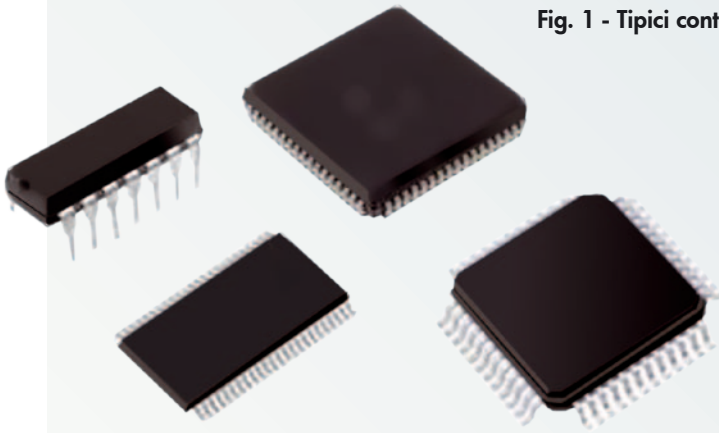
Iniziamo la stesura di una serie di articoli il cui scopo è iniziare i lettori interessati alla conoscenza e programmazione dei componenti elettronici più interessanti e versatili -i microcontrollori- che grazie alle loro caratteristiche e soprattutto al costo in continuo calo, stanno diventando l'elemento principe di un numero sempre maggiore di applicazioni. Premettiamo che tale corso non si focalizzerà su complicati aspetti teorici legati all'architettura dei dispositivi o ad avanzate tecniche di programmazione, di cui esiste abbondanza di documentazione sia cartacea che on-line:

l'obiettivo principale è mettervi in condizione, entro le prime cinque puntate, di poter lavorare praticamente con i microcontrollori per poterne apprezzare le potenzialità e verificare se può "valere la pena" di approfondire i numerosi argomenti che comunque tratteremo. Passiamo pertanto ad una breve descrizione necessaria a "rompere il ghiaccio".

CHE COS'È UN MICROCONTROLLORE

Con il termine microcontrollore (anche detto *single-chip microcomputer* e abbreviato in μC oppure, informalmente, "micro") si intende

Fig. 1 - Tipici contenitori dei microcontrollori.



un circuito integrato (che esternamente si presenta in uno dei contenitori o *packages* -in inglese- mostrati in Fig. 1) che ha la peculiarità di poter essere predisposto, dopo essere stato *istruito* opportunamente, ad eseguire molteplici funzioni.

Per rendere meglio l'idea, possiamo paragonare il microcontrollore alla cucina di un ristorante: in essa è presente un cuoco che già possiede le capacità e gli strumenti per creare diversi tipi di pietanze, ma che non può mettersi al lavoro fin quando non gli viene presentata la ricetta dei piatti scelti dai clienti. Allo stesso modo, il microcontrollore necessita di ricevere opportune istruzioni che gli comandino passo per passo le operazioni da svolgere; tali istruzioni, ovviamente, non sono le fasi di esecuzione della ricetta, ma sequenze di numeri binari e sono contenute non in un libro, ma in un apposito dispositivo detto *memoria di programma*, in quanto l'insieme

delle istruzioni è identificato -appunto- con il nome di *programma* (o *firmware*). Da qui deriva la versatilità del microcontrollore: andando a cambiare il contenuto della memoria, possiamo fare eseguire allo stesso dispositivo "ricette" diverse senza cambiare o modificare i componenti della scheda (il cosiddetto *hardware*), microcontrollore compreso.

A questo punto dovrebbe esservi abbastanza chiaro che se si vuole sostituire il microcontrollore guasto di una scheda non è sufficiente procurarsene uno uguale, ma bisogna anche scrivervi nella memoria lo stesso programma di quello "vecchio".

Passiamo ora ad analizzare meglio l'operato del nostro μC , dicendo che per ogni istruzione depositata in memoria esso deve compiere una serie di azioni per eseguire materialmente ciò che l'istruzione stessa chiede.

Un esempio chiarirà meglio il concetto: supponiamo che l'istruzione ricevuta sia relativa ad un'operazione di somma; allora il microcontrollore deve leggere i due operandi da un ingresso, elaborare i numeri per produrre la somma e mandare il risultato ad un'uscita per renderlo disponibile all'utilizzatore. In seguito, verrà letta la prossima istruzione (che può essere relativa ad una qualsiasi altra operazione) ed il ciclo si ripete.

Questa sequenza di operazioni (lettura istruzione, ingresso dati, elaborazione, uscita risultato) è simile quasi per tutti i tipi di istruzione

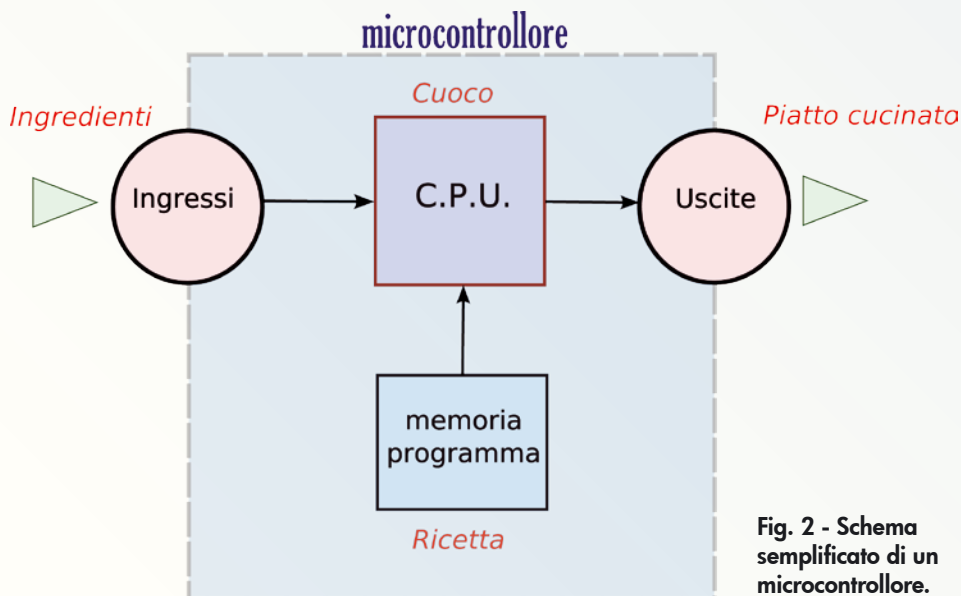
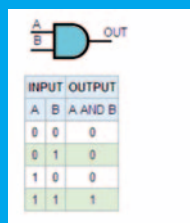


Fig. 2 - Schema semplificato di un microcontrollore.

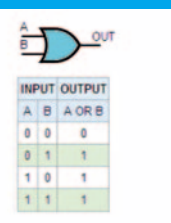
I numeri binari

C'era una volta l'elettronica analogica... Un tempo "lontano" (fino a quarant'anni fa) gli unici circuiti in grado di eseguire calcoli logici e aritmetici erano basati su componenti discreti (transistori) o Integrati (amplificatori operazionali) che lavoravano direttamente sui livelli di tensione: per fare un esempio, per sommare $3 + 5$ al circuito sommatore bisognava applicare sui due ingressi le tensioni di 3 V e 5 V e si andava a misurare l'uscita per conoscere il risultato (8 V). Successivamente sono state introdotte le *porte logiche*, cioè circuiti in grado di riconoscere ai propri ingressi unicamente due livelli di tensione per discriminare altrettanti *livelli logici*: zero e uno; queste porte erano in grado di eseguire solo operazioni elementari in logica binaria (AND, OR, NOT, ecc.). Nella Tabella 2 sono riportate le porte logiche (e le relative "tabelline della verità") corrispondenti alle tre operazioni logiche fondamentali (AND, OR e NOT). Tutte le altre si possono derivare da esse.

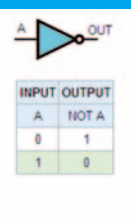
Porta AND



Porta OR



Porta NOT



Queste operazioni sono dette di logica *binaria*: leggono ingressi con soli due stati possibili e il risultato può avere ancora due soli stati.

Combinando in modo opportuno queste tre funzioni logiche fondamentali, si è in grado di eseguire virtualmente qualsiasi operazione aritmetica e logica: in questo senso si possono paragonare le porte logiche ai mattoni, che pur essendo "tutti uguali" permettono di costruire edifici di qualsiasi forma.

Inoltre il fatto di lavorare con circuiti che accettano solo livelli logici binari (i *bit*) non è limitante: se con un bit rappresentiamo solo due valori, con due bit abbiamo la possibilità di rappresentarne quattro: 00, 01, 10, 11. Con tre bit arriviamo ad 8 valori differenti e così via. La regola generale è che il numero di combinazioni rappresentabili vale $N = 2^n$ dove N è il numero di bit. Quindi l'aumento del numero di bit porta ad avere un aumento esponenziale del numero di combinazioni rappresentabili; nei circuiti a microprocessore si lavora soprattutto con numeri formati da 8 bit (256 combinazioni e quindi rappresentano, in decimale, i numeri da 0 a 255), ma si arriva anche a 32 bit (oltre

Decimale (base 10)	Esadecimale (base 16)	Binario (base 2)	Decimale (base 10)	Esadecimale (base 16)	Binario (base 2)
0	0	0000	8	8	1000
1	1	0001	9	9	1001
2	2	0010	10	A	1010
3	3	0011	11	B	1011
4	4	0100	12	C	1100
5	5	0101	13	D	1101
6	6	0110	14	E	1110
7	7	0111	15	F	1111

Tabella 1

4 miliardi di combinazioni). A seconda del numero di bit che li compongono, si sono stabiliti dei nomi per contraddistinguere i gruppi: *nibble* (4 bit), *byte* (8 bit), *word* (16 bit) e *double word* (32 bit).

Esistono regole precise per fare le conversioni tra i numeri binari (formati dai bit, 0 e 1) e i numeri decimali (le cui cifre vanno da 0 a 9, ai quali noi siamo abituati a pensare), che si possono rapidamente trovare in Internet. Purtroppo, però, la corrispondenza binario-decimale non è intuitiva: difficile capire "ad occhio" che il numero binario 10011001 corrisponde al numero decimale 77. Pertanto per esprimere rapidamente un numero binario è stata introdotta la numerazione esadecimale, che assegna ad ogni combinazione che può assumere un gruppo di 4 bit (il *nibble*) un simbolo diverso: poiché nel *nibble* i valori variano da 0000 a 1111 (16 combinazioni), utilizziamo, per rappresentarli, i numeri da 0 a 9 e le lettere da A a F come mostrato nella Tabella 1.

Quindi possiamo suddividere un numero binario arbitrariamente grande in gruppi da 4 bit e sostituire ad essi i simboli in accordo con la tabella. Ad esempio, trasformiamo il numero a 16 bit (word) '1101001000110100' in un numero esadecimale:

1101001000110100 □ 1101 | 0010 | 0011 | 0100 □ D234

Questa notazione è largamente utilizzata per esprimere numeri binari nella stesura dei programmi, in quanto è più semplice scrivere D234 piuttosto di 1101001000110100, e genera meno confusione ed errori anche di semplice battitura (è facile sbagliarsi e scrivere 1101010000110100 invece di 1101001000110100, mentre è più difficile scrivere D434 invece di D234). Un'ultima osservazione: poiché i simboli della notazione esadecimale comprendono anche i numeri, per non confondere un numero decimale con uno esadecimale (sono diversi) si possono far seguire questi ultimi da una 'H', oppure (in stile del linguaggio C) precederli con '0x'. Ad esempio 16 decimale vale 10 esadecimale e lo indichiamo come 10h o 10_h o 0x10; al numero decimale invece non si fa in genere seguire alcun suffisso e raramente si usa la lettera 'D' (16_D = 10_D).

che esso può eseguire. Lo schema semplificato di un microcontrollore può pertanto essere rappresentato come in Fig. 2, dove sono state anche riportate, in rosso, le voci dell'analogia culinaria.

Il micro è essenzialmente composto da blocchi di ingresso e uscita e da un'unità centrale di elaborazione (CPU, da *Central Process Unit*), detta anche *processore* oppure *core*, in grado di

interpretare ed eseguire le istruzioni che va a leggere da un'apposita memoria di programma, come vedremo meglio nel seguito.

LA CPU

È l'unità di elaborazione del microcontrollore: essa preleva le istruzioni dalla memoria di programma, le decodifica (per capire qual è l'azione da compiere) e le esegue, prelevando

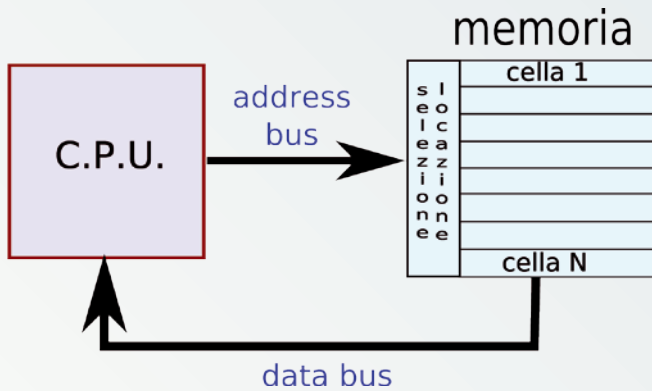


Fig. 3 - Collegamento tra CPU e memoria di programma.

(eventualmente) gli operandi dal blocco di ingresso e trasmettendo, infine, i risultati al blocco di uscita.

A questo punto chiariamo meglio in cosa consistono i dati (operandi ed istruzioni) con cui ha a che fare la CPU; questa è una **macchina digitale** in grado di leggere, interpretare ed elaborare sequenze di **numeri binari** (per chi avesse qualche perplessità sul significato del termine "binario" consigliamo di leggere il riquadro I **numeri binari**, in queste pagine) pertanto le uniche due lettere del suo alfabeto sono i valori "0" e "1" che possono assumere i bit costituenti le parole (siano esse *byte*, *word*, *double word*, ...) dei dati e delle istruzioni. Normalmente le CPU lavorano con parole il cui numero di bit è una potenza di due, da 4 in su (quindi 4, 8, 16, 32 bit), anche se esistono eccezioni (alcuni μ C della famiglia PIC di Microchip hanno istruzioni nel formato di 12 e 14 bit, ma i dati sono sempre "larghi" 8 bit). La CPU della stragrande maggioranza dei microcontrollori in commercio (tra cui il dispositivo ATMEGA16 di Atmel che utilizzeremo in questo corso) lavora con parole da 8 bit, sia per le istruzioni che per i dati.

LE ISTRUZIONI E LA MEMORIA DI PROGRAMMA

Come già accennato, la *memoria di programma* (esiste anche la memoria dati, ovvero la RAM, che però analizzeremo più avanti) è il contenitore in cui depositare tutte le istruzioni che compongono il programma da far eseguire al microcontrollore.

Possiamo paragonare questa memoria ad un mobile con tanti cassetti, identificati con una numerazione crescente, contenenti ciascuno un numero binario corrispondente ad un'istruzione del programma. Durante il funzionamento, la CPU del microcontrollore

va ad aprire questi cassetti (uno alla volta), ne estrae il numero binario che identifica una particolare istruzione, la esegue e quindi passa ad aprire il cassetto seguente per la lettura del numero binario (e quindi dell'istruzione) successivo. Come nelle fasi di preparazione di una ricetta di cucina, anche durante la normale esecuzione del programma le istruzioni devono essere eseguite in una precisa e rigorosa successione: pertanto la CPU si cura di aprire i cassetti in sequenza, ovvero dopo il cassetto 1 passa al 2, poi al 3 e così via.

Sperando che il concetto sia stato chiarito, formalizziamo alcune definizioni: parlando di memorie elettroniche, i cassetti si chiamano *celle* o *locazioni* e la loro selezione si chiama *indirizzamento*.

Nella Fig. 3 è schematizzata la connessione tra CPU e memoria di programma.

L'address bus (bus degli indirizzi) è formato da un certo insieme di fili che portano alla memoria il numero binario specificante la cella da selezionare: ad esempio mettendo 0101 (5 in decimale) si seleziona la quinta cella. Il valore contenuto nella cella viene quindi "riversato" sulle linee del *data bus* (bus dati).

In definitiva, in fase di lettura dell'istruzione la CPU imposta sui fili dell'*address bus* l'indirizzo della cella interessata e va poi a leggerne il valore sul *data bus*.

Una delle principali caratteristiche della memoria programma è la non volatilità, ovvero la capacità di mantenere i dati anche in caso di mancanza di alimentazione del dispositivo, per cui non è necessario riscriverla (programmarla) ad ogni accensione del sistema. Inoltre tale memoria è di sola lettura (*Read Only Memory*, o ROM) da parte della CPU, in quanto deve solo leggervi le istruzioni; può invece essere eventualmente riscritta (se del tipo EEPROM o Flash) dal programmatore (quindi da noi, tra poche lezioni...) per potervi caricare un nuovo programma.

IL SEGNALE DI RESET

In ogni microcontrollore esiste un ingresso di *reset* (che significa azzeramento), il cui scopo è quello di riportare la CPU ed i registri interni del micro nelle condizioni iniziali, indipendentemente dallo stato in cui si trovavano nell'istante di attivazione del reset stesso. Per *condizione iniziale* della CPU si intende, tra le

altre cose, che essa deve ripartire ad eseguire il codice dalla prima istruzione del programma, mentre i registri (dei quali spiegheremo più avanti le loro funzioni) vengono riportati ai valori definiti come "Initial value" nel data-sheet del microcontrollore, quindi in uno stato predefinito.

È dunque importante che il reset sia attivato almeno all'accensione del sistema, in modo da fare partire correttamente l'esecuzione del programma dalla prima istruzione; in caso contrario, sarebbe indefinito il punto in cui la CPU inizierebbe ad eseguirlo! Come tanti altri dispositivi, l'ATMEGA16 dispone di un meccanismo detto "brown-out" (talvolta è chiamato anche *smart reset*) che rileva quando la tensione d'alimentazione del micro è inferiore al livello di funzionamento normale (ad esempio perché sta ancora salendo dallo zero dopo l'accensione del sistema): in tal caso tiene attivato il reset fin quando il valore non è arrivato al livello stabilito, allorché lo rilascia e la CPU parte ed esegue regolarmente il programma dall'inizio.

Nel nostro micro il segnale di reset è comunque accessibile da un piedino esterno, in modo da dare all'utente la possibilità di attivarlo anche manualmente; potrebbe, ad esempio, essere utile per fare ripartire un programma che si è bloccato.

Nulla di strano: anche tutti i PC desktop hanno il famoso tastino di reset, che ben conoscono gli utilizzatori di alcuni sistemi operativi...

LE PERIFERICHE DI INGRESSO E USCITA

I blocchi di ingresso e uscita (detti *periferiche*) sono le porte che collegano il microcontrollore al mondo esterno: da esse vengono letti i dati e trasmessi i risultati di un'operazione. Le periferiche più semplici (dette porte parallele o *PIO, Parallel Input-Output*) collegano direttamente i piedini fisici (pin) del dispositivo ai bus (dati e indirizzi) della CPU, che può dunque leggerne lo stato (interpretato come bit: assenza di tensione -> legge 0, presenza di tensione positiva -> legge 1), oppure impostarlo (a 0 oppure a 1). Attraverso apposite istruzioni si può configurare un certo piedino per comportarsi arbitrariamente come ingresso o come uscita; da qui il termine di Input/Output (I/O pin).

Questi piedini vengono raggruppati per

Microprocessori e microcontrollori

La rivoluzione digitale che ha radicalmente cambiato in questi anni il nostro modo di vivere e comunicare inizia ufficialmente con la nascita del microprocessore, nel lontano (un'era fa, elettronicamente parlando) 1971. Con esso si apre la strada all'elettronica *programmata*, ovvero ai circuiti in grado di leggere programmi "pre-caricati" per eseguire calcoli anche notevolmente complessi e soprattutto in maniera flessibile. In quanto è il programma a decidere l'operazione da eseguire in ogni momento. Non ci volle molto ad intuire le potenzialità di questo componente, sul quale si concentrarono le risorse di diverse Case produttrici di semiconduttori (ed altre nacquero appositamente), ed iniziò subito la corsa al miglioramento continuo e all'abbattimento costi.

Nel corso del tempo, questo sviluppo si divise in due strade: da una parte la ricerca di un continuo aumento prestazionale (che seguono i microprocessori destinati per esempio al Personal Computer), e dall'altra una riduzione dei componenti necessari per realizzare un sistema completo, e quindi l'ottimizzazione dei costi e dello spazio. Un sistema richiede infatti, oltre al microprocessore stesso, una memoria di programma ed una memoria per i dati temporanei, oltre ad un insieme (variabile a seconda dell'applicazione) di periferiche di ingresso e uscita (I/O). Inoltre bisogna tenere in considerazione che questi componenti vanno interconnessi tra di loro con un certo numero di linee (che formano il *bus di sistema*), con la conseguente necessità di adottare circuiti stampati di ragguardevole dimensione e multistrato. Il miglioramento delle tecniche microelettroniche nello sviluppo del chip di silicio ha però consentito di inglobare all'interno di un *unico* dispositivo, oltre al microprocessore, anche le memorie ed alcuni dispositivi di I/O, dando origine al microcontrollore: un singolo circuito integrato contenente al suo interno un sistema a microprocessore completo. Esso, per funzionare richiede solamente di essere alimentato e naturalmente di essere programmato (o meglio bisogna programmare la sua memoria di programma interna, alla quale si può accedere con opportuni circuiti e segnali "dedicati").

A questo punto, la strada verso la miniaturizzazione e la diffusione di massa è stata splanata:

oggi giorno esistono numerose Case produttrici di microcontrollori (a differenza dei microprocessori, i quali sono ormai in un regime di oligopolio che guida il mercato degli elaboratori, tipo i PC) ciascuna delle quali propone una o più "famiglie" di dispositivi (come i *PIC* di Microchip e gli *ATMEGA* di Atmel) che coprono potenzialmente ogni esigenza di prezzo e prestazioni.

Di fatto, oggi possiamo facilmente trovare un microcontrollore dentro ad una sveglia elettronica, come pure nei bigliettiini di auguri musicali o nei termostati del riscaldamento; l'adozione di questi componenti richiede un costo minore di quello necessario per realizzare i circuiti corrispondenti con transistori o integrati dedicati (industrialmente, un piccolo microcontrollore "da termostato" costa poche decine di centesimi, ma anche comprandolo al dettaglio non si superano i 3 €).

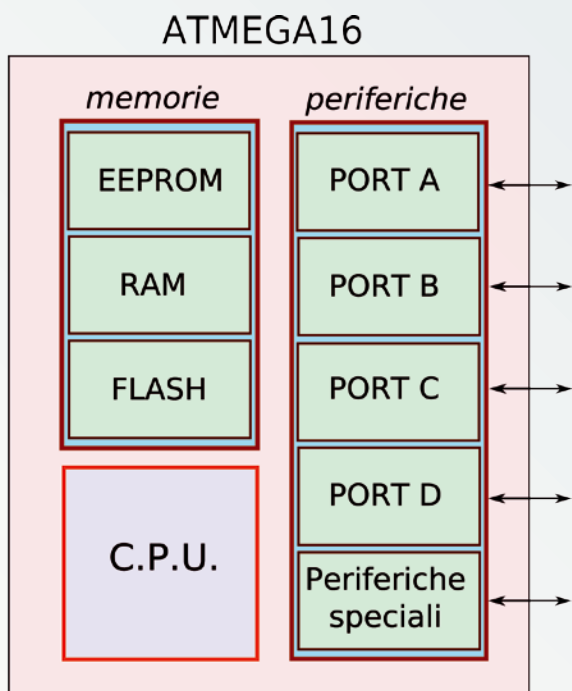


Fig. 4 - Schema a blocchi dell' Atmega16.

formare i cosiddetti I/O PORT (raramente tradotto in italiano in *PORTI*), in modo che il processore possa accedere contemporaneamente a tutti i piedini in esso contenuti (in modo, appunto, parallelo). Ad esempio, una CPU che lavora con parole da 8 bit (un byte) può leggere o scrivere in un "colpo solo" un Port formato da otto I/O pin. Avremo comunque modo di chiarire meglio questo concetto con gli esempi pratici delle prossime puntate. Come per la memoria, anche le periferiche sono indirizzate dalla CPU attraverso un numero binario sull'*address bus*, allo scopo di selezionare esattamente di volta in volta con quale si vuole operare.

Esistono anche periferiche più complesse, come la USART (Universal Synchronous Asynchronous Receiver Transmitter) o la I²C (Inter Integrated Circuit communication), per l'implementazione di protocolli seriali (i byte in arrivo dalla CPU vengono trasmessi all'esterno un bit per volta utilizzando un unico filo). Alcune periferiche possono anche non avere un collegamento diretto con i piedini esterni del microcontrollore (come l'oscillatore che fornisce le temporizzazioni alla CPU).

IL MICROCONTROLLORE ATMEGA16

Dopo una sommaria panoramica sui

microcontrollori, possiamo focalizzare la nostra attenzione sul dispositivo prescelto per lo sviluppo di questo corso: il chip **ATmega16** della Atmel. La sua scelta è stata dovuta principalmente ai seguenti fattori:

- fa parte di una "famiglia" di microcontrollori molto estesa e diffusa (la famiglia ATMEGA, della quale fanno parte anche i dispositivi utilizzati, tra l'altro, sulle schede *Arduino*), i cui membri si distinguono tra loro fondamentalmente per la diversa capacità di memoria, la presenza o meno di particolari periferiche ed il numero di piedini di I/O disponibili;
- ampia memoria programma (di tipo Flash e quindi riscrivibile) e numerose periferiche interne;
- ottima disponibilità di programmi e schede applicative in Internet;
- compilatore C e ambiente di sviluppo disponibili gratuitamente;
- sistemi di programmazione e *debug* a basso costo;
- disponibile in contenitore dual-in-line per facilità di montaggio e sostituzione anche sui prototipi.

Detto questo, proponiamo uno schema a blocchi semplificato ed una breve descrizione delle sue componenti; per approfondimenti è consigliabile scaricarne il data-sheet completo dal sito Atmel (www.atmel.com/dyn/resources/prod_documents/doc2466.pdf). In ogni caso, il funzionamento e l'utilizzo delle periferiche sarà descritto nel dettaglio durante le future applicazioni teoriche e pratiche.

CPU: è il motore del sistema in quanto legge ed esegue le istruzioni; essa è chiamata da Atmel "AVR-CPU" in quanto utilizzata nei microcontrollori che l'Atmel ha battezzato **AVR** (di cui fa parte la famiglia ATmega).

Flash: è la memoria di programma, può contenere fino a 16.384 Bytes, ed è cancellabile e riscrivibile (in fase di programmazione) fino a 10.000 volte.

RAM: è una *memoria dati*, ovvero un "serbatoio" che la CPU utilizza per riporre temporaneamente variabili e risultati delle operazioni. Pertanto, a differenza della memoria di programma è scrivibile dalla CPU durante l'esecuzione del programma stesso. Un'altra

differenza è la volatilità, intendendo con ciò che quando si toglie alimentazione al micro, la memoria perde tutti i dati.

Nel nostro micro la RAM complessiva è composta da 1.120 locazioni da 8 bit ciascuna (quindi 1.120 byte), ed è suddivisa in:

- 1.024 byte per utilizzo generico (chiamata "internal data SRAM memory");
- 32 byte (chiamati *registri* e numerati da R0 a R31) di cui si avvale la CPU per la memorizzazione temporanea di dati ed operandi;
- 64 byte per la gestione delle periferiche, anche questi chiamati *registri* (*I/O registers* per la precisione) e utilizzati per caricare i valori che impostano il funzionamento delle periferiche del micro, oppure per riportare alla CPU il loro stato (ad esempio la lettura di un port configurato come input).

EEPROM: è una memoria dati ma non volatile, dunque a differenza della RAM mantiene i dati memorizzati anche in assenza di alimentazione; tuttavia la CPU non può accedervi per leggere istruzioni (quindi non è utilizzabile come memoria di programma). Inoltre le procedure di lettura e scrittura non sono "immediate" ma richiedono sequenze di istruzioni ben precise, come vedremo più avanti. Va comunque ricordato che per l'esecuzione di un programma le memorie essenziali sono la Flash (per le istruzioni) e la RAM (per i dati temporanei): la EEPROM è una cosa "in più" che il nostro micro possiede e che può tornare utile per salvare particolari dati che non si vuole vadano persi spegnendo il sistema.

PORT A, B, C, D: il nostro microcontrollore dispone di quattro gruppi di piedini di ingresso/uscita (i PORT appunto) da 8 bit ciascuno; la CPU può accedervi specificando un determinato indirizzo (ogni port ha il proprio) per "comunicare" con l'esterno. Quasi tutti i pin dei port vengono inoltre condivisi con altre periferiche identificate nello schema come *speciali*, ovvero che svolgono funzioni particolari sui segnali in arrivo o verso il mondo esterno. Ogni microcontrollore, oltre ai classici I/O port paralleli, è equipaggiato con un proprio set di periferiche: di seguito elenchiamo brevemente quelle contenute nel

microcontrollore ATmega16.

- **Convertitore Analogico/Digitale (ADC):** permette di leggere una tensione continua analogica da alcuni pin (per esempio proveniente da un sensore) e convertirla in un numero binario interpretabile dalla CPU.
- **Timer/Counter:** vi sono 3 moduli indipendenti, utilizzabili come contatori di eventi o temporizzatori.
- **Oscillatore:** fornisce le temporizzazioni alla CPU ed alle periferiche (come i timer); può essere programmato per funzionare autonomamente (senza componenti aggiuntivi), prelevare il segnale da un pin o anche essere collegato ad un quarzo esterno per aumentare la precisione delle temporizzazioni (cosa utile, ad esempio, se si vuol fare un orologio digitale).
- **USART:** interfaccia per convertire i dati paralleli (come i byte, con cui lavora la CPU) in dati seriali (un bit per volta) utilizzati per colloquiare con alcuni dispositivi esterni con interfacciamento seriale (come mouse, memorie, modem).
- **I²C:** altra interfaccia seriale, che però supporta un altro protocollo (chiamato I²C) per il colloquio con dispositivi che lo utilizzano.
- **SPI:** interfaccia seriale sincrona, utilizzabile, tra l'altro, per programmare il microcontrollore.
- **JTAG:** interfaccia seriale utile per programmare ed eseguire il debug dei programmi.
- **Comparatore analogico:** confronta due tensioni V_1 e V_2 applicate su altrettanti pin ed ha in uscita un bit (collegabile anch'esso ad un pin e comunque leggibile dalla CPU) che indica quale delle due tensioni è maggiore (bit a 1 quando $V_1 > V_2$; bit a 0 quando $V_1 < V_2$).

Nel proseguimento del corso vedremo anche come mettere in funzione alcune di queste periferiche, ma per fare ciò è necessario imparare prima a programmare il microcontrollore, ovvero istruirlo ad eseguire le operazioni che desideriamo. Arriverci, dunque, alla prossima puntata, il cui scopo sarà proprio quello di fornire le basi per poter scrivere il nostro primo programma per l'ATmega16. ■



COLTIVA IL TUO HOBBY CON LE SCATOLE DI MONTAGGIO

di **Futura Elettronica**

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



cod. FT829K
€ 45,00

**SERRATURA
A TRANSPONDER
STAND-ALONE**



cod. VM188
€ 22,00

**TIMER DA
1 SECONDO A 60 ORE
(MONTATO)**



cod. K8004
€ 17,00

MODULATORE PWM CONTROLLATO IN DC



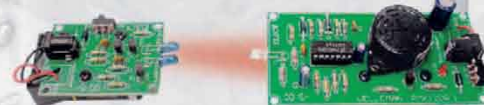
cod. K4001
€ 11,00

AMPLIFICATORE MONO 7W



cod. MK125
€ 9,00

INTERRUTTORE CREPUSCOLARE



cod. MK120
€ 12,50

BARRIERA INFRAROSSI TX/RX



cod. MK138
€ 9,00

TERMOSTATO



cod. MK127
€ 14,50

MINI ROBOT



cod. MK182
€ 18,00

**ECO DIGITALE
PER SISTEMI AUDIO**



cod. MK189
€ 7,00

**MONITOR DI
CARICA PER
BATTERIA AUTO**

Versione montata cod. VM189 - € 10,00



cod. FT908M
€ 98,00

**LOCALIZZATORE
TRAMITE RETE GSM**



cod. FT849K
€ 84,00

**TESTER EOBD
(UNITÀ DI DIAGNOSI
PER AUTO)**

Prezzi IVA inclusa.

Rileva l'asportazione o l'interruzione di uno dei pannelli costituenti un impianto fotovoltaico, dando l'allarme localmente o, tramite telecontrollo, in remoto.



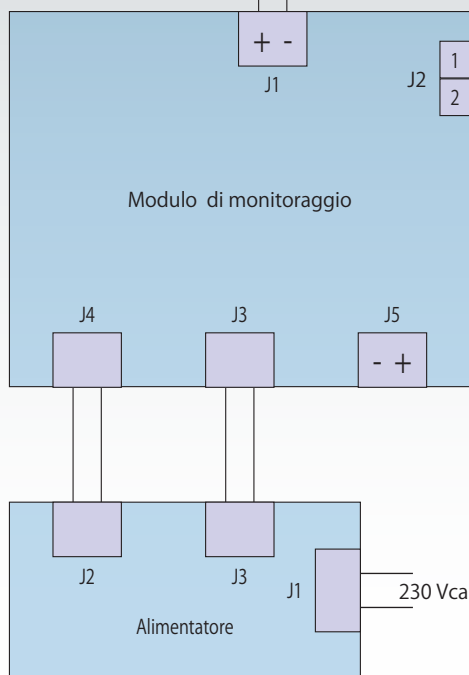
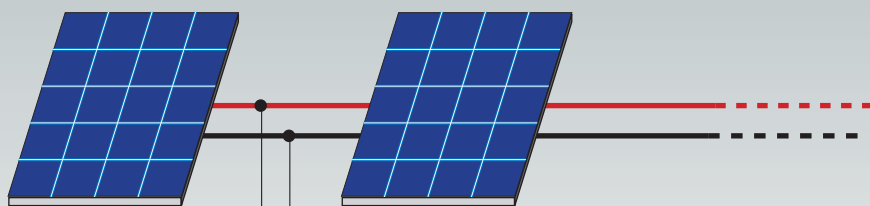
**STOP AI LADRI DI
PANNELLI SOLARI**

ANTIFURTO PER PANNELLI SOLARI

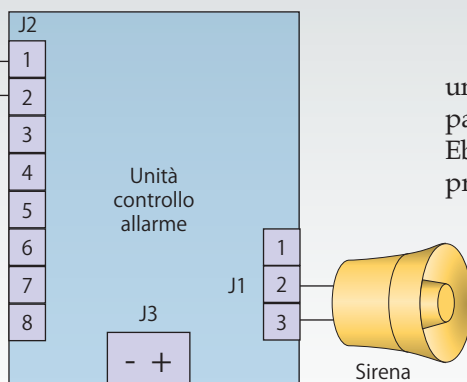
di ONOFRIO SPINELLA

L'aumento degli impianti per produrre energia a mezzo di pannelli fotovoltaici è da qualche tempo oggetto di attenzione da parte dei ladri: come se non bastassero i furti di cavi di rame (metallo divenuto quasi prezioso perché il suo prezzo è salito vertiginosamente negli ultimi anni) adesso sentiamo notizie anche di impianti fotovoltaici "alleggeriti" da furfanti d'ogni genere. Il fenomeno

si spiega con l'ancora elevato costo dei pannelli fotovoltaici (un elemento da 200 Wp costa anche più di 500 euro), che rappresentano una merce ben vendibile, ed affligge in special modo le installazioni non sorvegliate, cioè tipicamente quelle realizzate nei campi fuori dai grossi centri urbani e sui tetti di capannoni nelle zone industriali isolate dagli abitati e perciò poco sorvegliate nel week-end.



Per i possessori di vasti impianti si pone quindi il problema di proteggere il proprio investimento, per non vederlo "evaporare" al sole, o meglio, al buio, visto che i ladri agiscono la notte, quando si sentono meno disturbati. In commercio esistono antifurto per pannelli solari, ma spesso risultano costosi e funzionano



solo su impianti predisposti; per questa ragione ci è venuto in mente di progettare un sistema d'allarme dai costi contenuti, modulare (basta un modulo per ogni stringa FV, che è un insieme di pannelli in serie), gestito da microcontrollore ed interfacciabile con qualsiasi attuatore acustico, ma anche con il nostro telecomando 2IN/2OUT TDG133, che permette di ricevere SMS quando rileva un allarme.

COME FUNZIONA

Prima di analizzare lo schema elettrico dell'antifurto spieghiamo su quale principio si basa il sistema qui proposto; iniziamo dicendo che i pannelli fotovoltaici generano una tensione ai propri capi quando vengono colpiti dalla luce, mentre la notte, in pieno buio, non forniscono alcunché. Inoltre, per evitare di collegare i pannelli con la polarità invertita, ognuno ha un diodo in antiparallelo (catodo sul + e anodo sul -) cosicché in caso di collegamento al contrario, il pannello "sbagliato" viene bypassato nella stringa. Il diodo (o i diodi, se ce n'è più d'uno) di solito si trova dentro

Il nostro sistema

L'antifurto si compone di un modulo di monitoraggio che verifica la presenza dei pannelli e di un'unità di controllo allarme che ne legge i segnali e comanda gli attuatori d'allarme.

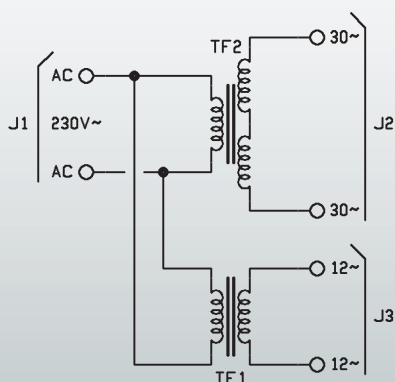
una scatola posta sul retro del pannello stesso (Fig. 1). Ebbene, per monitorare la presenza di un pannello FV, di giorno sarà sufficiente rilevare l'energia elettrica prodotta, poca o assai non ha importanza perché a noi bastano pochi volt ed alcuni mA per sapere che il pannello FV è ancora al suo posto; invece durante la notte, o meglio quando il pannello FV diventa inerte, non avendo una tensione da rilevare possiamo applicare al pannello una tensione continua e andare a rilevare (limitandone l'intensità) la corrente che ne deriva. Ciò è reso possibile proprio dalla presenza dei diodi di protezione (se non ci fossero dovreste aggiungerli...) che permettono lo scorrimento di corrente nel senso opposto a quello convenzionale; in questo modo, è possibile sapere se tutti i pannelli di una stringa FV sono a posto, oppure se ne manca anche uno solo, condizione nella quale la corrente s'interrompe.

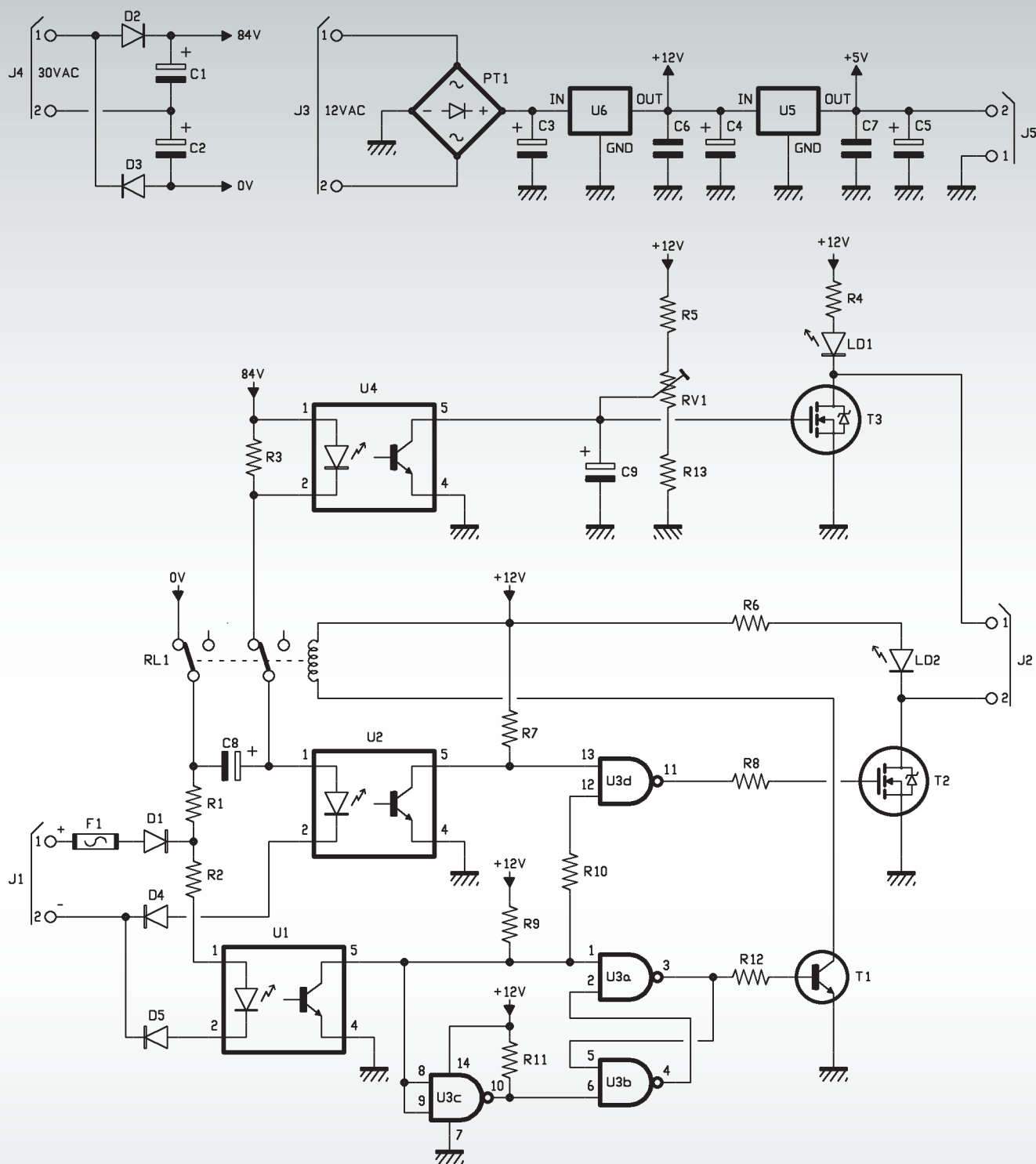
SCHEMA ELETTRICO

Il nostro sistema d'allarme è formato essenzialmente da due unità (anche se ne esiste in realtà una terza, perché abbiamo previsto uno stampato per alloggiare i due trasformatori di alimentazione), ciascuna delle quali prende posto su un proprio circuito stampato; le unità sono un modulo di monitoraggio ed un'unità di controllo allarme.

La prima si collega direttamente alla stringa di pannelli fotovoltaici

[schema elettrico
ALIMENTATORE]



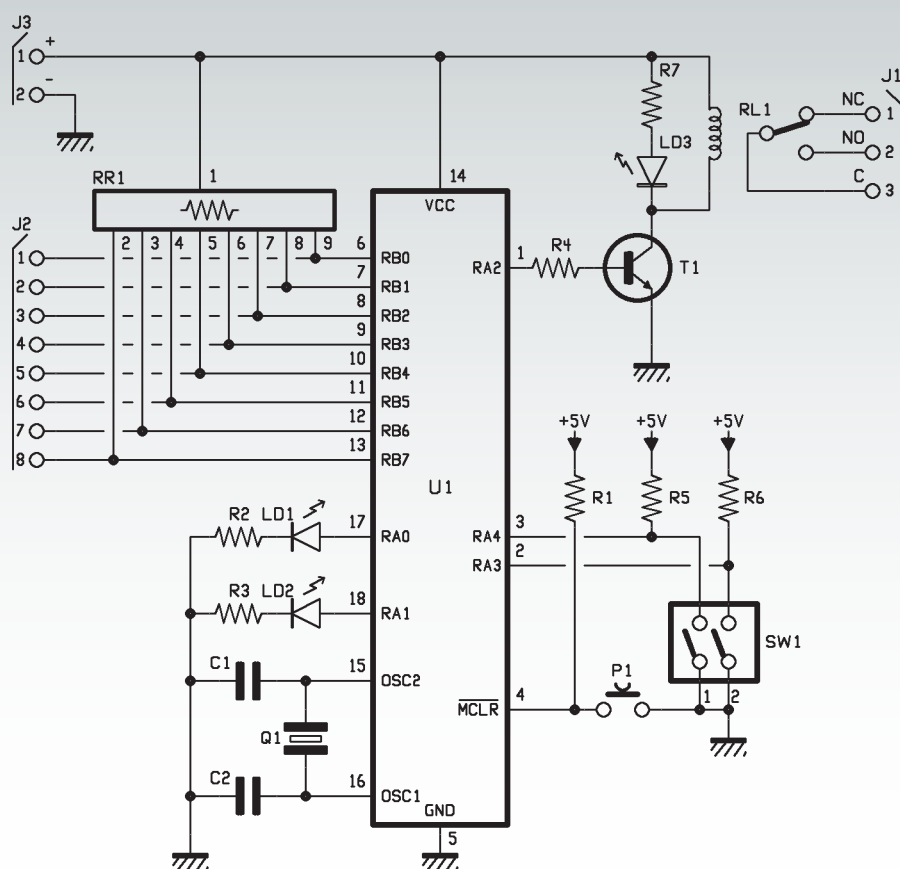


[schema elettrico **MODULO DI MONITORAGGIO**]

e dispone di un sensore che rileva la tensione diurna o il flusso di corrente inversa notturna. La seconda si interfaccia alla prima e mediante un microcontrollore gestisce gli allarmi, attivando degli attuatori mediante relé e fornendo le segnalazioni del caso.

Guardiamo per primo lo schema del modulo di monitoraggio, dove notiamo la morsetteria collegata ai pannelli solari, il cui polo positivo finisce su un fusibile (F1) e su un diodo 1N4007 (D1) posti in serie tra loro; il diodo è stato messo a protezione dall'inverso-

ne di polarità, mentre il fusibile protegge dall'eccesso di corrente, cosa utile perché in particolari condizioni di irraggiamento solare, soprattutto nel caso di batterie di pannelli messi in serie o in parallelo fra di loro, all'ingresso del circuito potrebbero giungere

[*schema elettrico* **UNITÀ CONTROLLO ALLARME**]

più di 450 Vcc, con correnti di 5 A ed oltre.

Per avere una certa protezione è opportuno posizionare il nostro sistema in prossimità della terminazione dei cavi FV; qualora si debbano coprire distanze di diversi metri, è opportuno interporre un fusibile (sempre da 100 mA) direttamente in serie ai morsetti della stringa FV, in modo da evitare che un cortocircuito nel collegamento possa fare danni. Andiamo avanti con l'analisi del circuito elettrico e immaginiamo, per comodità, che il sistema inizialmente sia in quiete: è notte e ai capi della stringa FV non è presente alcuna tensione; siccome in tale condizione RL1 risulterà diseccitato, nel contatto normalmente chiuso circolerà la corrente di test che viene fatta scorrere con polarità positiva verso il pin 1 di

U2 e raggiungerà il polo negativo della stringa, dove fluirà nei diodi di protezione dei pannelli finendo il suo percorso sull'altro contatto (NC) di RL1.

Notate che all'alimentazione inversa dei pannelli provvede il duplicatore di tensione formato da D2 e D3 e dagli elettrolitici C1 e C2, che funziona così: quando il trasformatore collegato a J4 fornisce tensione con polarità positiva sull'anodo del diodo D2, quest'ultimo conduce e lascia fluire in C1 la corrente che serve a caricarlo ad un valore pari a circa 1,41 volte ciò che il trasformatore eroga, quindi poco più di 42 volt; all'inversione di polarità (positivo sul + del C2) la corrente fluisce in C2 e da esso in D3, per uscire dal contatto 1 della morsettiera. In questo caso l'elettrolitico si carica a 42 V, negativi rispetto al

contatto 2 della J4. Ogni periodo dell'alternata applicata a tale morsetteria, C1 e C2 si caricano a circa 42 V, determinando tra l'elettrodo positivo del primo ed il negativo del secondo una differenza di potenziale che complessivamente vale 2×42 V, ossia 84 V.

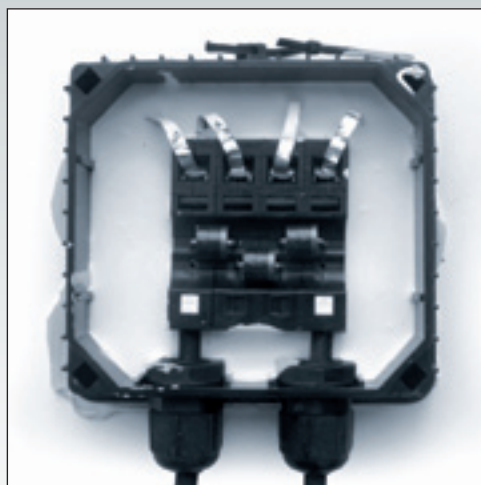
La corrente inviata al blocco di pannelli fotovoltaici fluisce anche in R1 ed R3 e nel LED interno al fotoaccoppiatore U2, cosicché il piedino 5 di quest'ultimo si porta a circa zero volt (dato che il fototransistor interno satura); invece U1 resta interdetto, perché D5 blocca la corrente in quanto viene polarizzato inversamente.

Notate che anche U4 rimane interdetto ed il suo pin 5 si trova a livello alto, per via di R3, che shunta una parte della corrente; di conseguenza, il gate del MOSFET T3 viene polarizzato ed il transistor conduce, facendo illuminare LD1 e portando il contatto 1 della morsettiiera J2 a livello logico basso.

Contemporaneamente, il pin 5 di U2 risulta a livello basso e mantiene a zero logico il piedino 13 di U3d (il pin 12 dello stesso è mantenuto a livello alto dall'interdizione dell'U1) cosicché l'uscita di quest'ultima è ad 1 e manda in conduzione T2, il cui drain porta a zero logico il contatto 2 di J2 e fa accendere il LED LD2.

Ora vediamo cosa accade quando fa giorno e i pannelli producono elettricità: in questo caso la polarità positiva sul contatto 1 della morsetteria J1 fa condurre il diodo D1, che polarizza il diodo emettitore del fotoaccoppiatore U1, il cui fototransistor di uscita satura e determina lo zero logico sul pin 5; ciò lascia andare il piedino 6 della U3b a livello alto, permettendo al piedino 4 di assumere uno stato logico che dipende da quello del piedino 3 della U3a. Quest'ultima fornisce

Fig. 1 - I diodi di protezione (in questo caso sono tre) si trovano nel retro del pannello solare.



in uscita il livello alto, dato che il suo piedino 1 viene forzato a livello basso; tale situazione determina lo zero logico all'uscita della U3a ed il blocco del transistor T1 nella condizione di interdizione. Quindi RL1 torna a riposo e gli 84 V non alimentano più il blocco dei

pannelli fotovoltaici. Notate che U3a e U3b formano un flip-flop che, quando il gruppo dei pannelli fotovoltaici non fornisce corrente, attiva il relé e chiude l'alimentazione di test notturna; infatti, grazie ad U3c, quando non viene generata tensione da parte dei pannelli, U3a si trova i piedini 1 e 2 ad 1 logico, in quanto U3c fornisce in uscita zero, stato che blocca ad 1 il piedino 4 della U3b. Stabilito ciò, vediamo cosa accade se interrompiamo il collegamento verso i pannelli fotovoltaici: in questo caso il circuito attiva RL1 come se fosse notte, ma siccome non circola corrente perché il circuito dei pannelli fotovoltaici è interdetto, lo stato sull'uscita di U3d passerà a zero logico e T1, non essendo più polarizzato, lascerà spegnere LD. Inoltre il contatto 2 di J2 si porterà a livello

basso mentre l'1 resterà ad 1 logico.

Tale condizione verrà rilevata dalla scheda unità di controllo allarme.

A questo punto i più smaliziati tra voi potrebbero osservare che basterebbe, per ingannare il nostro sistema, agire di notte quando la stringa non produce corrente, mettere in corto i terminali d'ingresso dell'allarme e simulare così la condizione notturna a tempo pieno, portando via i pannelli indisturbati.

Effettivamente questo si potrebbe fare, ma solo in teoria, perché in pratica ad impedirlo pensa U4, il quale, se ricordate, resta inattivo perché R3, shuntando la corrente, impedisce l'accensione del LED interno; ebbene, mettendo in cortocircuito l'ingresso del modulo in modalità notte, verrebbe meno

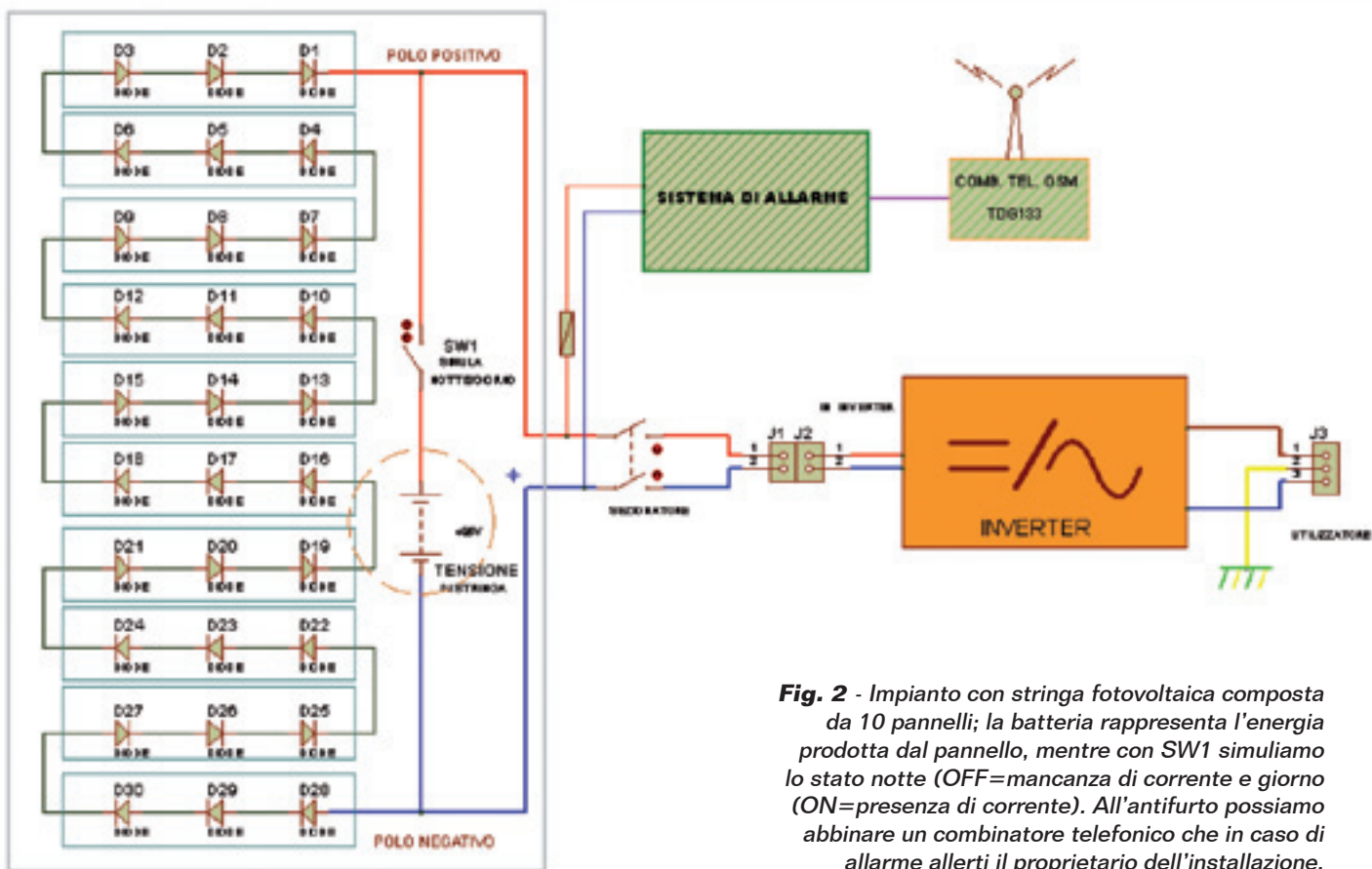


Fig. 2 - Impianto con stringa fotovoltaica composta da 10 pannelli; la batteria rappresenta l'energia prodotta dal pannello, mentre con SW1 simuliamo lo stato notte (OFF=manca di corrente e giorno (ON=presenza di corrente). All'antifurto possiamo abbinare un combinatore telefonico che in caso di allarme allerti il proprietario dell'installazione.

la caduta di tensione prodotta dai diodi presenti sui pannelli e ciò produrrebbe un aumento della corrente circolante, consentendo, nonostante la presenza di R3, di eccitare U4, il cui pin5 si porterà a livello basso lasciando spegnere LD1 e portando a livello basso il contatto 1 della morsettiera J2. Anche questa condizione farà scattare l'allarme.

Bene, detto ciò è opportuno fare una precisazione: la tensione che il circuito fornisce nel funzionamento notturno è di 84 volt perché tale ampiezza si rende necessaria per compensare la caduta di

tensione sui diodi di protezione dei pannelli fotovoltaici, la quale si attesta intorno agli 0,7 volt per diodo; ammettendo di avere a che fare con una stringa di 30 pannelli in serie aventi un diodo di protezione ciascuno, in tutto contiamo 31 diodi (compreso quello posto all'ingresso del modulo) cui si aggiunge il LED interno a U2, che per semplicità di calcolo ipotizziamo introduca una caduta di tensione di 1,2 V. Quindi il nostro alimentatore deve fornire una tensione di almeno:

$$V = 31 \times 0,7 + 1,2 = 22,9 \text{ V}$$

Ora occorre fare un'importantissima osservazione: immaginando di avere la solita stringa FV da 30 pannelli da 12 V nominali (anche 16 a vuoto) e supponendo che non vi sia alcun carico, la tensione fornita potrebbe superare i 460 volt; se in tali condizioni collegassimo il modulo di controllo, RL1 si troverebbe normalmente a riposo e prima che il sistema provveda a commutarlo, la tensione del FV per un brevissimo periodo si sommerebbe agli 84 V.

Tuttavia, con la circuizione adottata avremmo al massimo 7÷8 milliampere nei diodi d'in-

[piano di montaggio **ALIMENTATORE**]

Elenco Componenti:

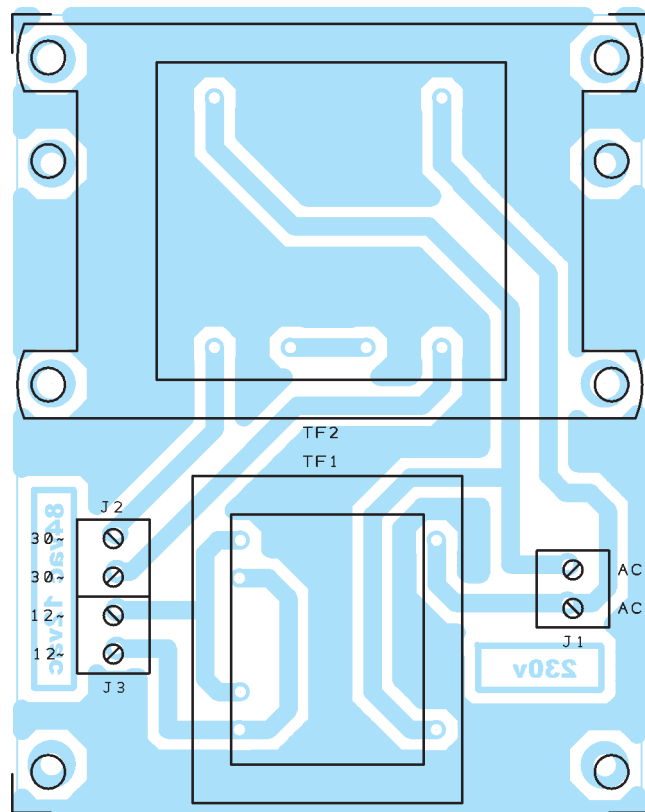
TF1: Trasformatore 230V/12V/5VA

TF2: Trasformatore 230V/15V/5VA

Varie:

- Morsetto 2 poli (2 pz.)

- Circuito stampato



Elenco Componenti:

R1: 56 kohm 2 W
R2: 82 kohm 2 W
R3: 2,2 kohm
R4÷R6: 3,3 kohm
R7: 56 kohm
R8: 150 ohm 1%
R9, R10: 22 kohm 1%
R11: 10 kohm
R12: 150 ohm 1%
R13: 3,3 kohm

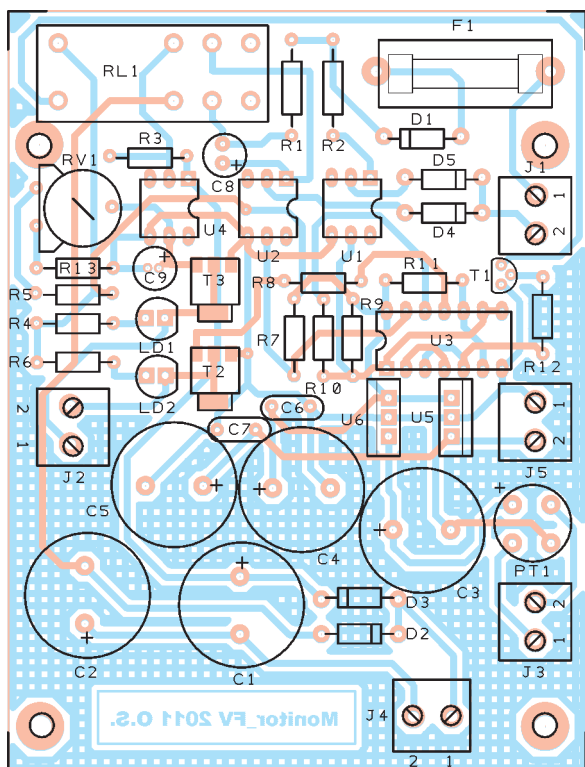
RV1: Trimmer 100 kohm MO
C1, C2: 1000 μ F 50 VL elettrolitico
C3: 2200 μ F 35 VL elettrolitico
C4: 470 μ F 35 VL elettrolitico
C5: 220 μ F 25 VL elettrolitico
C6, C7: 10 nF multistrato
C8: 10 μ F 100 VL elettrolitico
C9: 1 μ F 50 VL elettrolitico
U1, U2: 4N26
U3: 4011

U4: 4N26
U5: 7805
U6: 7812
T1: BC547
T2, T3: BSP100
D1÷D5: 1N4007
PT1: Ponte diodi W06
LD1, LD2: LED 5 mm
rosso

F1: Fusibile ritardato 100 mA
RL1: Relé 12V

Varie:

- Morsetto 2 poli (4 pz.)
- Zoccolo 3+3 (3 pz.)
- Zoccolo 7+7
- Porta fusibile da c.s.
- Circuito stampato



gresso dei fotoaccoppiatori, per un brevissimo intervallo di tempo; tale situazione non costituisce alcun pericolo, soprattutto perché RL1 commuta nel giro di pochi millisecondi, sconnettendo gli 84 V dal resto del circuito sensore. Comunque, se anche RL1 non commutasse, il circuito non riporterebbe danni e l'unità a microprocessore rileverebbe l'allarme, perché si avrebbe la morsettiera J2 con il contatto 1 a livello logico alto. Un'altra considerazione riguarda la lunghezza dei cavi di collega-

mento al gruppo di pannelli fotovoltaici: se è eccessiva, la resistenza elettrica che ne deriva potrebbe limitare la conduzione di U4 e quindi la corrente di collettore del fototransistor di tale optoaccoppiatore; perciò abbiamo previsto il trimmer RV1, che consente di regolare la polarizzazione di gate del T2. Concludiamo la descrizione dello schema elettrico del modulo di monitoraggio, dicendo che il circuito integra un completo alimentatore stabilizzato capace di fornire le tensioni di 12 e 5 Vcc

partendo dai 12 Vca ricevuti dalla morsettiera J3, cui va collegato il secondario di un trasformatore avente il primario da rete. Nel nostro circuito abbiamo previsto l'uso di due trasformatori distinti (vedere *schema elettrico alimentatore*) per assicurare il giusto isolamento tra la sezione di media tensione e quella di bassa tensione. L'alternata applicata a J3 viene raddrizzata dal ponte a diodi PT1, che fornisce all'elettrolitico C3 impulsi sinusoidali, da esso livellati fino ad ottenere circa 16 Vcc; U6 stabilizza tale tensione a

[piano di **UNITÀ CONTROLLO ALLARME**]

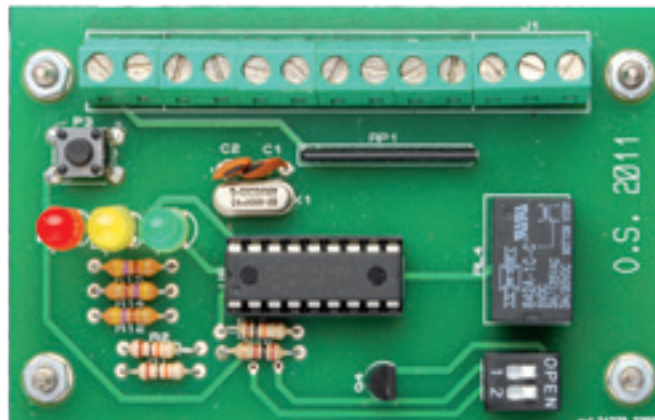
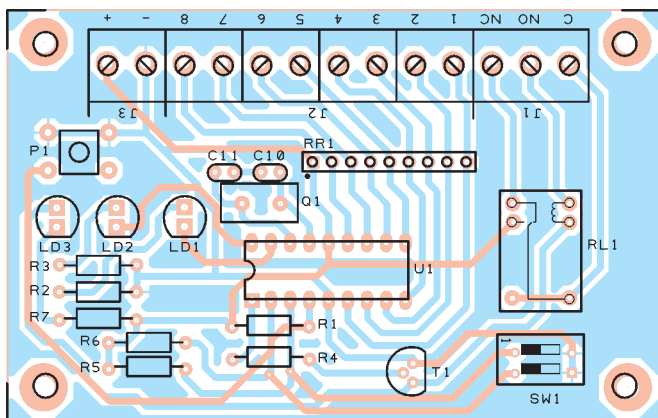
Elenco Componenti:

R1: 10 kohm
R2, R3: 470 ohm
R4: 150 ohm
R5, R6: 10 kohm
R7: 360 ohm

C1, C2: 22 pF ceramico
U1: PIC16F84A-04/P (MF969)
LD1: LED 5 mm verde
LD2: LED 5 mm giallo
LD3: LED 5 mm rosso

T1: BC547
SW1: Dip-Switch 2 vie
RL1: Relé 12V
P1: Microswitch
RR1: Rete resistiva 47 kohm 8+C
Q1: Quarzo 4 MHz

Varie:
- Morsetto 3 poli
- Morsetto 2 poli (5 pz.)
- Zoccolo 9+9
- Circuito stampato



12 V, fornendo l'occorrenza per alimentare i circuiti dei relé e dei LED, nonché i fotoaccoppiatori e la logica discreta della scheda. Con gli stessi 12 V viene alimentato il regolatore U5, finalizzato a ricavare i 5 Vcc che, tramite il connettore J5, alimentano l'unità di controllo allarme.

L'UNITÀ DI CONTROLLO ALLARME

Giunti a questo punto possiamo analizzare lo schema elettrico di quest'ultima unità, che è stata dimensionata immaginando di applicarla ad un impianto fino a 6 kW di potenza massima, avente da una a quattro stringhe FV variamente composte.

Il circuito è affidato ad un microcontrollore PIC16F84 (il cui programma sorgente, scritto in PIC-Basic, è riportato nel **Listato 1**); di esso usiamo gli otto bit del port B come ingressi per monitorare lo

stato delle due linee di uscita di un massimo di quattro moduli di monitoraggio (quindi di quattro stringhe di moduli fotovoltaici). In condizioni normali, tutte le linee di RB devono risultare a zero, mentre averne una a livello alto corrisponde ad un allarme. Il rilevamento di un allarme determina il livello logico alto sulla linea RA2 e l'innesco del relé, che permette di comandare lampeggianti, sirene ed altri attuatori (ad esempio delle lampade per illuminare la zona dove sono posati i pannelli e mettere in fuga i ladri) ma anche di dare il trigger ad un modulo di teleallarme via telefono fisso o cellulare.

Va subito precisato che per evitare falsi allarmi è stato previsto un certo ritardo dal momento in cui si verifica un cambio di stato sulle uscite 1 e 2 di J2 a quando il pin RA2 del micro si porta a

livello alto; questa condizione è la modalità di preallarme. Le linee RA3 ed RA4 vengono utilizzate per impostare, mediante SW1, tale ritardo (l'impostazione si fa tenendo conto della **Tabella 1**). Notate che variando il tempo varierà la velocità dei lampeggi di DL3: in stato di quiete DL3 lampeggerà in continuazione mentre DL4 rimarrà spento. In caso di anomalia rilevata, prima di attivare il relé di allarme il circuito attiva la modalità di preallarme, segnalata dall'accensione del DL4 per tutta la durata del tempo di preallarme impostato per mezzo di SW1. Qualora l'evento scatenante fosse un falso allarme di breve durata, il circuito ritornerà in quiete e DL3 riprenderà a lampeggiare indicandoci che tutto è regolare, ma DL4, che lampeggiava durante il preallarme, rimarrà acceso fisso per avvisarci dell'evento. Qualora la condizione di allarme si protrasse oltre il tempo preimpostato, RA2 si porterà nello stato alto per un tempo di circa tre secondi, sufficiente

Tabella 1 - Ritardo di attivazione dell'allarme.

Stato/tempo	30 secondi	1 minuto	2 minuti	4 minuti
dip 1	Off	On	Off	On
dip 2	Off	Off	On	On

a far partire la chiamata di un combinatore telefonico, attivare un avviso acustico temporizzato e altro ancora. A questo punto, se lo stato di allarme smette il circuito riprende il monitoraggio, altrimenti il ciclo si ripete sino a che permane lo stato di allarme. La memoria di allarme o preallarme si può resettare premendo il pulsante P1, che permette di far ripartire il micro cancellando ogni evento. DL4 si illumina se per caso, durante il normale funzionamento, venisse a mancare momentaneamente l'alimentazione al circuito.

Per finire diciamo che oltre a proteggere un impianto fotovoltaico da possibili furti, il nostro sistema si presta anche a segnalare anomalie come un cavo allentato, un contatto incerto, un filo in corto oppure interrotto; infatti osservando lo stato di DL1 e DL2, oltre ad individuare il settore anomalo possiamo diagnosticare altri problemi.

MONTAGGIO E COLLAUDO

Il montaggio non presenta particolari difficoltà e si conduce dopo aver realizzato i circuiti stampati occorrenti, le cui tracce sono disponibili nel nostro sito web www.elettronica.in.it. I piani di montaggio illustrati in queste pagine indicano come montare i pochi componenti che servono. Per l'installazione, ricordiamo, nell'unità di controllo allarme, di chiudere a massa i morsetti della morsettiera J2 che non sono utilizzati, ovvero quelli che non collegate ad alcuna stringa fotovoltaica. Quanto ai collegamenti tra l'impianto (cioè il modulo di monitoraggio) e l'impianto fotovoltaico, potete riferirvi alla **Fig. 2**. Ricordatevi di prestare la massima attenzione, dato che le tensioni prelevabili ai capi delle stringhe possono essere molto

elevate, tali da folgorare; indossate guanti in gomma e non toccate mai i cavi o eventuali cacciaviti con le mani.

I collegamenti ai pannelli devono essere possibilmente non troppo lunghi (cercate di rimanere in qualche metro); nel caso, collegate i circuiti vicino all'inverter e collegatevi lì.

Qualunque cavo utilizzate, ricordate che deve avere un isolamento pari almeno alla massima tensione prelevabile da una stringa, altrimenti potete incorrere in grossi problemi. Ricordatevi di

installare un interruttore bipolare tra il circuito dei pannelli e il modulo, che vi consenta di disconnettere il tutto in caso di interventi. Notate anche che molti inverter commerciali usati per lo scambio sul posto sono muniti di un dispositivo che verifica l'isolamento delle stringhe FV rispetto alla terra; perciò i terminali delle stringhe devono essere ben isolati da terra. In caso contrario l'inverter non si avvia e segnala la condizione di "errore d'isolamento".

Per la taratura del circuito è



Su che tetto li metto?

Non tutti i tetti permettono di ottenere la stessa resa da un impianto fotovoltaico, quindi prima di installare un impianto è importante sapere quanto convenga. Da oggi esiste SEES (Solar Energy from Existing Structures) cioè uno strumento sviluppato all'università di Göteborg, che permette di valutare se il tetto di un edificio è più o meno adatto a sfruttare al meglio l'energia del sole mediante pannelli fotovoltaici. Il suo funzionamento si basa su un sistema cartografico digitale (GIS, ossia Geographical Information System) che raccoglie ed analizza i dati geografici per valutare la posizione di un edificio rispetto agli altri; un simulatore interno proietta la luce di un sole virtuale sul quartiere ricostruito in 3D e simula le ombre create dagli edifici circostanti e dalle piante, calcolando quanto potrebbero influire sull'efficienza energetica nel corso dell'anno e addirittura mese per mese. In pratica lo strumento calcola l'entità della radiazione solare totale che potrebbe essere ricevuta da ciascuna parte del tetto e quindi l'effettivo irraggiamento (espresso in kWh/m²) cui verrebbero sottoposti i pannelli.

Listato 1

```

*****
* NAME      : UNIT_F84.BAS                               *
* AUTHOR    : [O. SPINELLA]                               *
* NOTICE   : COPYRIGHT (C) 2011 [MIKROITEK]              *
*           : ALL RIGHTS RESERVED                         *
* DATE      : 23/06/2011                                   *
* VERSION   : 1.2                                          *
* NOTES     : UNITA' DI CONTROLLO ALLARME FOTO VOLTAICO  *
*           :                                             *
*****
@ DEVICE PIC16F84,XT_OSC,WDT_OFF,PWRT_ON,PROTECT_OFF

TRISA = %11111000 'SETTA TUTTI I BIT DELLA PORTA A COME USCITE E PORTA.4 INGRESSO
TRISB = %11111111 'SETTA TUTTI I BIT DELLA PORTA B COME INGRESSI
'DEFINE BUTTON_PAUSE 50 'STABILISCE IL TEMPO MINIMO PERCHÉ UN PULSANTE
                        'VENGA RITENUTO PREMUTO

LED_RUN      VAR PORTA.0  'LED PROGRAMMA GIRA REGOLARE VELOCITA DEI LAMPEGGI
                        'IN FUNZIONE DEL RITARDO IMPOSTATO
LED_PRE_A    VAR PORTA.1  'USCITA LED DI PREALLARME E MEMO

OUT           VAR PORTA.2  'USCITA COMANDO RELE ALLARME
P1            VAR PORTA.3  'INGRESSO PULSANTE
P2            VAR PORTA.4

LINEA_A VAR PORTB.0      'INGRESSO DI MONITORAGGIO CANALE A
LINEA_B VAR PORTB.1      'INGRESSO DI MONITORAGGIO CANALE A
LINEA_C      VAR PORTB.2      '        "        "        "        "
"           B
LINEA_D VAR PORTB.3      '        "        "        "        "
LINEA_E VAR PORTB.4      '        "        "        "        "
LINEA_F VAR PORTB.5      '        "        "        "        "
LINEA_G VAR PORTB.6      '        "        "        "        "
LINEA_H VAR PORTB.7      '        "        "        "        "
"           C
"           D
"           D

PORTA=0
FLASH VAR word
S_TEMPO VAR WORD
attesa var word
S_TEMPO=500          'VALE MEZZO SECONDO CON QUARZO DA 4 MHZ

MAIN_LOOP:
if p1=1 and p2=1 then flash=60
if p1=0 then flash=120
if p2=0 then flash=240
if p1=0 and p2=0 then flash=480
OUT=0
PAUSE FLASH
TOGGLE LED_RUN
PAUSE FLASH

if portb <> 0 then gosub allarme 'se almeno uno degli ingressi è basso allora
                                'va in preallarme

TOGGLE LED_RUN

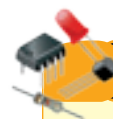
GOTO MAIN_LOOP      ' RICOMINCIA IL CICLO
                        *****
ALLARME:      'ROUTINE GESTIONE ALLARME E TEMPO DI PREALLARME
LED_PRE_A=1
LED_RUN=1
FOR ATTESA = 0 TO FLASH
toggle led_pre_a
PAUSE S_TEMPO
NEXT ATTESA
led_pre_A=1
if portb=0 then goto main_loop 'se gli ingressi si sono ripristinati fine pre allarme
OUT=1                          ' fa partire la chiamata
pause 3000
RETURN

END

```

previsto il trimmer RV1, che va ruotato sino a far illuminare il DL1; la sua regolazione dipende dal numero dei pannelli componenti la stringa FV e si può effettuare simulando l'asportazione di uno o più pannelli per trovare la soglia di intervento giusta. A riguardo, diciamo che da prove effettuate dopo aver tarato il trimmer, se si elimina un solo pannello dall'impianto il sistema va in preallarme; va comunque detto che questo tipo di prova si può fare solamente durante la notte, in quanto di giorno il circuito dell'unità di monitoraggio rileva la tensione prodotta dai pannelli.

I due circuiti e i relativi trasformatori possono trovare posto in una scatola a tenuta stagna del tipo usato negli impianti elettrici, possibilmente dotata di coperchio in plexiglass trasparente, oppure in una qualunque scatola purché sia di plastica. Montate la scheda con il microcontrollore in modo che i LED possano sporgere da opportuni fori che avrete fatto allo scopo su un lato della scatola. ■



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono di facile reperibilità. I master dei tre circuiti stampati ed il firmware con cui programmare il PIC possono essere scaricati gratuitamente dal sito della rivista. Il micro già programmato (cod. MF969) è disponibile al prezzo di 8,00 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>

Diventa il nuovo

AGENTE SEGRETO

Sistemi di registrazione discreti
ideali per scattare foto e realizzare
filmati senza dare nell'occhio.



€74,00
cod. TDOC01



Orologio digitale
da tavolo
con telecamera
e DVR integrati

Telecamera



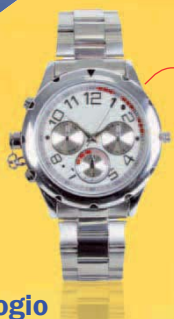
€89,00

cod. CAMCOLVC11



Orologio
analogico da tavolo
con telecamera
e DVR integrati

Prezzi IVA inclusa



€88,00

cod. CAMCOLVC10

Orologio
con telecamera
e DVR integrati



Telecamera

€36,00

cod. CAMCOLVC9

Penna con telecamera
e DVR integrati



Telecamera

€32,00

cod. CAMCOLVC8

Portachiavi
con telecamera e DVR

**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.

IN VOLO CON L'ELETTRONICA

Quello dell'aviazione civile è il settore dei trasporti in cui più si spende e si innova, ed è stata sicuramente la sinergia fra crescita tecnologica e strategie commerciali a regalarci voli sempre più economici, senza pregiudicare la nostra sicurezza. Dai primi aerei con ali in legno e tela, ai modernissimi Boeing 787 Dreamliner e Airbus A320 NEO, di acqua sotto i ponti, o meglio "d'aria sotto le ali", ne è passata molta; ma le novità più consistenti in fatto di aeromobili si devono all'elettronica e sono state sviluppate negli ultimi decenni. Così nel settore aeronautico è nata l'avionica, ossia l'elettronica di bordo, che comprende numerosi sistemi, i più

interessanti dei quali riguardano l'assistenza al volo (dal GPS ai sistemi di volo e atterraggio strumentale) ed il controllo di volo. In queste pagine vi spiegheremo come l'elettronica ha cambiato gli aerei, in special modo nei comandi.

Il primo importante automatismo adottato negli aerei civili è stato il pilota automatico, nato per sollevare i piloti dall'onere di compiere centinaia di manovre ripetitive durante i lunghi viaggi; inizialmente basato su sensori (anemometri, variometri, bussole, altimetri tradizionali) e attuatori meccanici, oggi è un complesso sottosistema elettronico che collabora con il computer di bordo in maniera

di DAVIDE SCULLINO

Dai fratelli Wright ad oggi, scopriamo come si sono evoluti i sistemi di bordo grazie a circuiti elettronici che assistono i piloti per garantirci voli sempre più sicuri e confortevoli, ma anche più ecologici.

più o meno decisiva. Nel pilota automatico si impostano parametri come l'altitudine da mantenere, la velocità di crociera desiderata, il rateo di salita o di discesa (*climb/descent ratio*, ossia di quanto l'aereo debba salire o scendere in un certo

lasso di tempo) e la rotta da seguire, quindi il computer interviene sui comandi per soddisfare le richieste; ad esempio, per il rateo di salita gestisce automaticamente la manetta dei motori (*automanetta*) azionandola con servomotori simili a quelli da mo-

dellismo e l'elevatore. La rotta viene controllata agendo sul timone e sugli alettoni. In molti aerei è previsto l'auto-YAW (yaw= imbardata), ossia un automatismo che coordina il movimento degli alettoni con quello del timone, in virata; questo sistema



Strumenti di ieri e di oggi

Salendo a bordo di un moderno velivolo e guardando qualche foto degli storici aerei del secondo dopoguerra, appare subito una differenza: quegli strumenti a lancetta e la bussola, i manometri ecc. sono stati tutti rimpiazzati da grandi LCD che riportano tutti i dati forniti dal computer di bordo. Un tempo si volava guardando l'orizzonte artificiale per valutare il pitch (beccheggio) ed il rollio, controllando la rotta sulla bussola e verificando nell'altimetro la propria altitudine; altri preziosi strumenti come l'indicatore di velocità (airspeed o anemometro, che dir si voglia) ed il variometro informavano, rispettivamente, sulla velocità relativa e sulla variazione di quota, ossia il rateo di salita o discesa. La bussola saprete certo come funziona: un ago magnetizzato punta il Nord terrestre; l'orizzonte artificiale è un po' come la bolla del muratore, cioè contiene una sfera sospesa con due linee ortogonali che seguono il livello del mare ed indicano quindi se l'aereo si sta inclinando lateralmente longitudinalmente. L'altimetro funziona rilevando la pressione atmosferica statica, mentre l'indicatore di velocità sfrutta il tipico tubo di Pitot, ossia una sonda che prende l'aria frontalmente e in base alla pressione che essa esercita su una membrana comanda un indicatore a lancetta; siccome la pressione varia, a parità di velocità, con la consistenza dell'aria, per evitare che l'indicazione sia falsata dalla

quota di volo il tubo di Pitot riporta un'apertura in basso, dove entra aria alla pressione atmosferica, indipendente dalla quota. Tale pressione viene sottratta a quella dovuta alla velocità e si ottiene così un'indicazione che non dipende dalla quota.

Oggi è cambiato un po' tutto: l'orizzonte artificiale viene rilevato con accelerometri MEMS, mentre il variometro e l'altimetro si basano su trasduttori piezoelettrici di pressione; il tutto può essere fatto con un accelerometro (l'accelerometro misura l'accelerazione di un corpo lungo una direzione) a tre assi. Si usano anche giroscopi, che misurando la rotazione di un corpo su di un asse, permettono di valutare le variazioni di rotta rispetto alle impostazioni; la combinazione di più giroscopi permette di ottenere indicazioni su tutti gli assi.

Per la velocità potrebbe provvedere il GPS, ma al momento il segnale non è sufficientemente preciso, quindi si usano ancora i tubi di Pitot, seppure la pressione agisca non più su un sistema pneumatico ma piuttosto su trasduttori piezoelettrici.

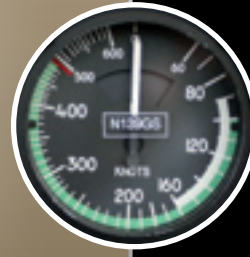
Comunque, per ragioni di sicurezza nei cockpit rimangono alcuni comandi tradizionali: si tratta dell'orizzonte artificiale e dell'altimetro, oltre all'indicatore del carburante e ad altro che, in caso di avaria del computer e di conseguente oscuramento della strumentazione LCD, possa fornire le indicazioni essenziali.



può essere usato anche senza inserire l'autopilota ed è realizzato comandando timone ed elevatori mediante sistemi elettroidraulici o elettromeccanici.

L'autopilota serve solo dopo il decollo e prima dell'atterraggio, anche se i moderni velivoli dotati di GLS molto precisi potrebbero atterrare con esso. Il compito dell'autopilota è oggi più facilitato che mai, grazie alla

disponibilità a bordo di piattaforme integrate (ADIRS) in grado di rilevare e trasmettere al computer le variazioni di rotta, di inclinazione, ma anche di velocità ed accelerazione: parliamo di giroscopi, accelerometri a tre assi, che consentono di percepire la rotazione (virata) il rollio (*roll*) e l'inclinazione del muso (*pitch*) con grande precisione ed affidabilità. Per quanto riguarda il controllo dei motori, nei velivoli con propulsione a turbopropellerica o turbogetto e turbofan, da decenni viene svolto elettronicamente mediante i FADEC (*Full Authority Digital Engine*): si tratta di unità a microprocessore che governano in modo ottimale i parametri di esercizio, la combustione, il raffreddamento e la lubrificazione, regolando anche il numero di giri delle turbine sulla base delle



richieste della manetta governata dal pilota o dell'automanetta del pilota automatico. Sono un po' come le centraline elettroniche delle automobili.

TANTI SENSORI, MENO TIMORI

Un altro grande passo avanti verso l'affidabilità è stato fatto con l'installazione di unità di diagnostica che riportano in cabina (*cockpit*, se preferite gli inglesi...) le condizioni dei molteplici parametri che in un moderno *liner* vanno tenuti sotto controllo; proprio il computer ha permesso di far volare grandi aerei con due soli piloti (*comandante e primo ufficiale*, che siede normalmente a destra del comandante) risparmiando l'ingegnere di volo, ossia quel membro dell'equipaggio che aveva il compito di leggere la miriade di strumenti e spie collocati a lato della parte posteriore o laterale della cabina (nonché controllare l'attività dei motori) e che è sparito con l'entrata in servizio, ad esempio, dell'A300, del Boeing 767 e dell'MD-11. I moderni velivoli hanno sensori nelle superfici di controllo per rilevare gli sforzi, nelle stive e in cabina per registrare la pressione atmosferica, all'esterno per leggere la temperatura dell'aria e valutare il rischio di congelamento delle superfici di controllo, nelle ruote dei carrelli per rilevare la pressione delle gomme, nei motori per monitorare ogni genere di parametro (temperatura, giri, avarie di vario genere). I dati di gran parte di essi confluiscono su pannelli multifunzione che ormai sostituiscono lancette e indicatori meccanici, i quali rimangono come backup di sicurezza per alcuni parametri vitali quali velocità, altitudine, orizzonte artificiale, livello del combustibile nei serbatoi, pressione dell'olio dei motori. Oggi

nelle cabine dei moderni aerei ci sono grandi display LCD, spesso touch-screen, che riepilogano dati come direzione, rotta, velocità, rateo di salita, giri dei motori, posizione rispetto all'orizzonte in rollio e salita/discesa e che sono funzionali interfacce bidirezionali tra pilota e aereo.

Tra i sensori possiamo considerare anche l'allarme di stallo, adottato da diversi decenni sugli aerei di linea e collegato ad un avvisatore acustico o ad un attuatore a vibrazione montato nel volantino del pilota o anche del primo ufficiale (nel DC-10 e nell'MD-11) in modo da percepirne la vibrazione. I rilevatori di stallo sono basati, ad esempio, su una piccola paletta montata un po' sotto e dietro al bordo d'entrata dell'ala; per piccoli angoli di attacco (*l'angolo di attacco* è quello che la linea longitudinale dell'ala forma con la direzione dell'aria) la linea di ristagno dell'aria si trova davanti alla paletta, che perciò viene spinta indietro, mentre ad alti angoli di attacco (rischio di stallo) la linea di ristagno si sposta sotto e dietro, sotto l'ala, superando la paletta ed attivando l'interruttore che comanda l'avvisatore di stallo in cabina. Un secondo tipo di sensore si basa sulla pressione esercitata dall'aria sulla superficie alare ed allo scopo viene posizionato appena sotto il bordo d'entrata dell'ala; per piccoli angoli di attacco, il bordo d'entrata è una zona di bassa velocità e alta pressione, mentre quando l'angolo d'attacco diventa tale da far rischiare lo stallo, diventa una zona di alta velocità e bassa pressione. In questo caso, attraverso un'apposita aper-

tura viene risucchiata aria, che comanda un sensore di pressione collegato all'allarme in cabina. Altri sensori importanti sono quelli posizionati sui carrelli per rilevare l'atterraggio; sono stati implementati dopo un incidente occorso ad un Boeing 757, in cui in un motore si attivò il *reverse* con il conseguente avvistamento in volo. Oggi, il *reverse*, ossia il sistema (basato su convogliatori che aprendosi dirigono il flusso di gas dei turbogetti in avanti) che inverte la spinta dei motori dopo l'atterraggio per dare, insieme agli aerofreni, la prima energetica frenata (senza essi, la corsa di atterraggio sarebbe eccessiva e richiederebbe piste lunghissime) si può attivare solo dopo che un sensore elettromeccanico rileva l'appoggio a terra del carrello principale.

DAI TIRANTI AI FILI ELETTRICI

Fino a qualche decennio fa, i comandi del pilota venivano trasmessi alle superfici di controllo dell'aereo mediante tiranti e cavi metallici; oggi, nella gran parte degli aerei i comandi dalla cabina di pilotaggio si trasmettono



no mediante cavi elettrici. Normalmente il controllo del volo si ottiene mediante comandi primari, che sono il volantino per salire (cabrare) o scendere (picchiare) e per rollare in senso orario o antiorario, ovvero la "manetta", per comandare i motori. Nei piccoli aerei e anche in quelli di medie dimensioni costruiti dalle origini dell'aviazione ad una trentina di anni fa, il pilota, agendo su volantino e pedaliera, muoveva le superfici di controllo mediante tiranti rigidi o cavi in acciaio; col crescere di peso e dimensioni, ma anche della velocità di crociera, sale considerevolmente lo sforzo richiesto al pilota, che per contrastare la resistenza dell'aria (cioè la spinta originata sulle superfici di controllo per reazione al movimento nell'aria) deve esercitare uno sforzo consistente. In un primo tempo il problema è stato risolto amplificando la forza esercitata dal pilota mediante leve vantaggiose, che però implicavano l'aumento dell'escursione dei comandi; poi sono arrivati servosistemi meccanici come ad esempio le servoalette, capaci di muovere le superfici di controllo per effetto dell'aria, montate nei piani di coda di aerei come



l'MD-80 della McDonnell Douglas, tutt'oggi uno dei preferiti dai piloti di linea. Il principio della servoalette è semplice: posta all'estremità dell'elevatore, per esempio, quando il pilota aziona il volantino essa si muove nella direzione opposta a quella richiesta, in modo che la forza esercitata dall'aria spinga nel verso desiderato la superficie di controllo; per esempio, sollevando la servoalette l'elevatore viene spinto verso il basso. La cosa facilita di molto il compito del pilota, perché la servoalette è piccola rispetto alla superficie di controllo da azionare, perciò azionabile con minimo sforzo.

Col crescere delle dimensioni, anche questo sistema si è rivelato poco efficiente: un Boeing 747 difficilmente si sarebbe potuto controllare così, senza contare che realizzare i tiranti per arrivare dalla cabina alla coda di un gigante del genere, lungo oltre 70 metri, sarebbe stato problematico; quindi per trasmettere i comandi si è fatto ricorso a circuiti idraulici dotati di servovalvole, ossia valvole comandate dal volantino e dai pedali per aprire il circuito dell'olio in pressione in modo che, tramite pistoni idraulici, muovesse le superfici di controllo. Questo sistema è stato adottato per anni su aerei di linea (o *liner*, come vengono chiamati usualmente) di medie e grandi dimensioni, quali ad esempio l'Airbus A300, McDonnell Douglas DC-10, MD-11 eccetera. Sebbene funzionale, il comando idraulico è stato da tempo messo in discussione qualche decennio fa, quando i progettisti di aerei si sono posti il problema di risparmiare peso ed hanno pensato di farlo, oltre che sostituendo parti di metallo con analoghe in materiali compositi (cominciò l'Airbus con la deriva dell'A300, che impiegava bulloni di acciaio alternati ad altri di composito), alleggerendo gli

Gli aerei volano sfruttando quel fenomeno detto "portanza", che si genera sulle ali per effetto della differenza della velocità dell'aria che scorre sotto (intradosso) rispetto a quella che passa sopra (extradosso); a produrre la portanza concorre anche il fatto che le ali sono normalmente più alte verso la parte anteriore rispetto alla posteriore e che perciò l'aria investe l'intradosso tendendo a sollevare l'ala. Naturalmente tutto ciò accade in movimento, quindi serve un motore che faccia muovere l'aereo in avanti.

Una volta sollevato da terra, il velivolo vola sospeso sulle ali, ma occorre un contrappeso che permetta di dirigerlo su o giù: parliamo non di un contrappeso meccanico ma di un sistema aerodinamico, che consiste nello stabilizzatore; quest'ultimo è formato da due piccole ali, simmetriche e centrate rispetto a quelle di sostentamento, la cui inclinazione determina uno sforzo sull'asse trasversale delle ali facendo orientare il muso in su o in giù. Negli aerei di linea lo stabilizzatore si posiziona finemente mediante il comando TRIM, che si regola a seconda della distribuzione del carico davanti e dietro alle ali, della velocità e quota di volo. Ma chi decide se far cabrare (tirare su il muso e quindi far salire l'aereo) o picchiare (puntare il muso in basso e scendere) è

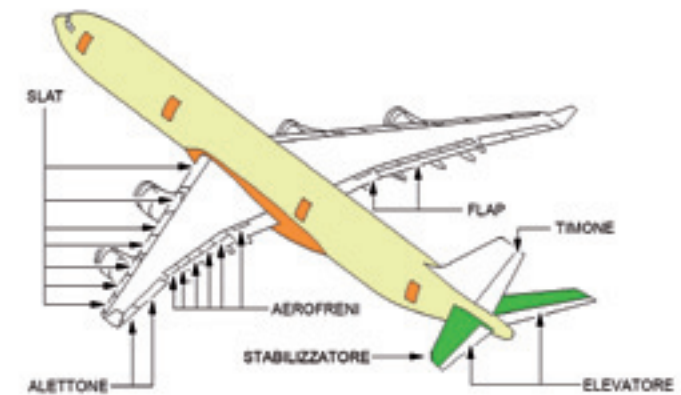
azionamenti dei comandi che partono dalla cabina. Sembra strano, ma tiranti e tubazioni che corrono avanti e indietro per un aereo di 50÷70 metri pesano tonnellate! Il primo passo in questa direzione è stato fatto con il *Fly-By-Wire* (abbreviato, FBW) un sistema derivato dall'aviazione militare (il primo aereo ad adottarlo fu l'F-16) introdotto negli aerei civili dall'Airbus, che consiste nel sostituire i comandi meccanici con quelli elettrici. Il sistema fu la scommessa che l'allora giovanissima (nacque nel 1970) Airbus fece per recuperare il divario di competenze esistente fra essa e costruttori storici come Boeing, Lockheed e McDonnell Douglas e per guadagnare quote di mercato; permise di realizzare aerei più leggeri, economici e facili da pilotare, peraltro con costi di formazione ridotti. Ciò invogliò molte compagnie aeree ad orientarsi verso le sue macchine.

l'elevatore (detto anche timone di profondità o elevone) che è la parte posteriore dello stabilizzatore, resa mobile in modo da poterla far ruotare verso l'alto o verso il basso su una cerniera; l'elevatore viene comandato dalla barra o volantino: tirando verso il pilota l'elevatore flette verso l'alto e l'aereo sale di quota, mentre spingendo in avanti l'elevatore si abbassa e la coda del velivolo viene spinta verso l'alto facendo "mettere giù" il muso e quindi scendendo di quota.

Altro problema: deciso come far salire o scendere l'aereo, c'è il problema di dove dirigerlo, cioè farlo virare a destra o sinistra; a terra ciò viene fatto con il carrello girevole (quello anteriore negli aerei di linea con carrello triclo, o quello posteriore nei piccoli velivoli tipo Cessna o Piper, dotati di carrello bicipede) mentre in volo deve pensarci ancora il piano di coda, ovvero la parte che contiene stabilizzatore ed elevatore: nel centro di

questo sistema c'è un'aletta verticale passante per la linea mediana longitudinale del velivolo, che serve a tenere dritto l'aereo, impedendogli di virare e rollare (ossia inclinarsi su un fianco). La parte terminale di questa "pinna" (detta tecnicamente deriva) è mobile e incernierata in modo da poter sporgere a sinistra o a destra; quest'aletta oscillante è il rudder o timone di direzione (o semplicemente timone) e funziona così: spostandosi verso destra l'aria spinge la coda in direzione opposta e punta il muso dell'aereo a destra, mentre verso sinistra spinge la coda a destra e fa virare a sinistra. Il timone si comanda con due pedali, resi solidali da una barra e disposti ai piedi del pilota: spingendo il pedale sinistro l'aereo vira a sinistra, mentre verso destra si ottiene la virata a destra.

Il movimento del velivolo sul terzo asse, ossia l'inclinazione in senso orario o antiorario, si ottiene con delle



alette, chiamate alettoni, incernierate nelle parti posteriori delle ali e funzionanti come elevatore e timone e comandate dalla barra o volantino; in quest'ultimo caso, girando il volante a sinistra si solleva l'alettone dell'ala destra e si abbassa quello della sinistra, facendo inclinare l'aereo allo stesso modo, mentre girando il volantino a destra succede l'opposto (l'alettone sinistro si abbassa e quello destro si alza) facendo rollare verso destra. A terra, quindi in manovra e

in rullaggio, il comando degli alettoni normalmente agisce anche sul carrello. La presenza dell'elettronica ha permesso di realizzare la funzione auto-YAW, ossia il coordinamento degli alettoni con il timone, che normalmente il pilota effettua da sé; tale manovra si rende necessaria perché in virata l'ala interna alla traiettoria curva perde portanza e l'aereo tende a piegarsi su un lato (scivolata d'ala) quindi usando gli alettoni si raddrizza.

Dopo le prime incertezze, altri costruttori hanno seguito l'esempio dell'Airbus, tant'è che oggi molti aerei non hanno più tiranti, amplificatori di sforzo o servoalette e le superfici di controllo vengono governate mediante segnali elettrici inviati tramite cavi ad attuatori elettroidraulici o elettromeccanici (ad esempio Airbus A318 ed A319), i quali azionano tali superfici di controllo. La sostituzione dei comandi meccanici con fili elettrici ha permesso di alleggerire i velivoli e ridurre i consumi, mantenendo un rapporto accettabile tra rischi e benefici.

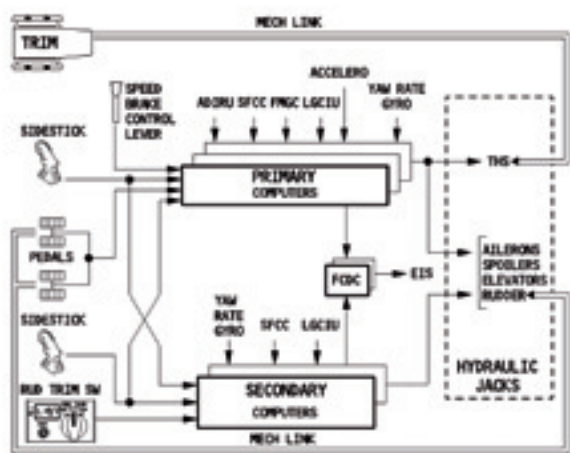
FBW: UN SISTEMA, DIVERSI SVILUPPI

Il Fly-By-Wire è stato sviluppato in vario modo, anche se viene tuttora visto con occhio critico soprattutto per come è stato sviluppato dall'Airbus. In realtà sostituire i meccanismi con attuatori elettrici

comandati dalla cabina di pilotaggio non presenta più rischi di quelli insiti nei sistemi meccanici e idraulici, anche perché, per garantire la massima sicurezza, i circuiti elettrici sono sdoppiati ed anche triplicati (per esempio sulle più recenti versioni dell'Airbus A340). Sostanzialmente un FBW funziona così: nei comandi sono inseriti potenziometri, encoder o sensori magnetici per rilevarne il movimento; i segnali ricavati da essi vengono amplificati in corrente ed inviati a dei servomotori posti sugli azionamenti delle superfici di controllo o alle elettrovalvole. L'unico vero difetto del Fly-By-Wire, è l'assenza del *force-feedback* (ritorno di forza) ossia il fatto che il pilota, mancando un collegamento meccanico tra i comandi e le superfici di controllo, non sente qual è l'entità dello sforzo compiuto da queste ultime, il che porta, in caso di condizioni atmosferiche

particolarmente avverse o volo a velocità eccessiva, a compiere manovre che possono compromettere strutturalmente l'aereo e danneggiare, ad esempio, il piano di coda o il timone. Per ovviare a questo inconveniente sono state escogitate varie soluzioni: la prima la pensò l'Airbus per i primi A320 e consisteva in un sistema di molle inserito nella cloche per irrigidirla man mano che la si sposta verso il fine-corsa; indubbiamente una tecnica assai discutibile. In un secondo tempo il *force-feedback* è stato rilevato mediante sensori elettrici di sforzo (ad esempio trasduttori di pressione piezoelettrici) posti sulle superfici di controllo, trasmettendo i segnali corrispondenti direttamente sui comandi (dove contrastano, mediante masse centrifughe o attuatori elettromagnetici, la forza esercitata dal pilota per trasmettere la sensazione dello sforzo compiuto:

Fly-By-Wire: Airbus e Boeing a confronto



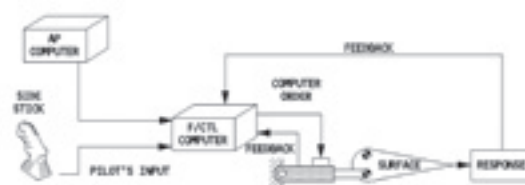
Schema dei comandi FBW dell'Airbus A340.

La vera rivoluzione portata dall'elettronica nell'aeronautica è il Fly-By-Wire, che consiste nel trasmettere i comandi dagli organi in cabina alle superfici di controllo mediante segnali elettrici. Osannato dai suoi fautori e deprecato da chi lo vede come l'inizio della fine dell'arte di pilotare, è stato un passaggio epocale e quasi obbligato, in quanto ha permesso di ridurre il peso degli aerei, soprattutto di quelli di grandi dimensioni, nei quali far correre tiranti (duplicati per garantire la ridondanza)

e condotti idraulici per diverse decine di metri dalla cabina alle ali ed al timone, significa sprecare carico utile. Sebbene per realizzare un Fly-By-Wire basti disporre dei sensori nei comandi in cabina e inviare i segnali elettrici ricavati ad attuatori posti nelle vicinanze delle superfici

di controllo, per azionarle, i costruttori si sono spinti oltre. In particolare, l'Airbus (prima ad adottarlo nel suo A320) partendo dalla considerazione che un sistema elettrico può facilmente interagire con un computer, ha sviluppato in realtà un governo computerizzato dei propri velivoli; in pratica negli Airbus 318, 319, 320, 321, 330, 340 e 380, al posto del canonico volantino c'è un joystick a lato del sedile di ogni pilota (detto, non a caso, *sidestick*) che comanda elevatore (avanti e indietro) ed alettoni (sinistra e destra) mentre rimane la pedaliera che gestisce il timone di direzione. Questi comandi forniscono segnali elettrici letti da un computer chiamato FCDC, cioè *Flight Control Data Concentrator* (per elevatori ed alettoni provvede un altro

computer, chiamato ELAC) il quale può agire in tre modi, a seconda delle impostazioni. Il computer, ampiamente ridondato (ci sono computer di backup...) normalmente elabora le richieste del pilota e, sulla base dei dati desunti dall'unità di acquisizione dei parametri di volo (ADIRU, *Air Data Inertial Reference Unit*) e dell'involucro di volo consigliato, all'occorrenza corregge le richieste, comandando l'aereo come ritiene opportuno; inoltre coordina il movimento degli alettoni e del timone in virata (auto-YAW) ed aggiusta il TRIM in funzione della distribuzione del carico, della quota e velocità. In sostanza, se il pilota chiede una cabrata di 40° e il computer ritiene che all'attuale velocità ciò possa portare allo stallo o compromet-



Schema di principio dei comandi dei moderni Airbus.

un sistema come quello della console per videogiochi Wii o inviandoli al computer di bordo e facendoli mediare da esso prima di mandarli ai comandi. Quest'ultimo è il caso tipico degli Airbus, dove il computer verifica anche l'eseguitività dei comandi: ad esempio limita l'escursione del timone al crescere della velocità (nell'A340, da $\pm 31,6^\circ$ fino

a 150 nodi a $\pm 3,5^\circ$ da 350 nodi in su). Purtroppo, se il computer va in avaria il pilota non riesce a percepire lo sforzo cui sottopone le superfici di controllo. Tale rischio non sussisterebbe nei FBW puri, dove servocomandi azionano le superfici di controllo e sensori rilevano il loro ritorno in

posizione di riposo, trasmettendo indipendentemente al pilota segnali corrispondenti allo sforzo. Il Fly-By-Wire è stato sviluppato essenzialmente secondo due filosofie; quella Airbus vede un coinvolgimento attivo totale del computer, che non solo governa la trasmissione dei comandi, ma

tere strutturalmente l'aereo, imposta l'angolo di salita al massimo consentito. Questa metodica, chiamata anche *Normal Law*, è quella vigente durante i voli in condizioni normali; esiste poi il livello a protezione ridotta, chiamato *Alternate Law*, in cui il pilota può escludere alcune protezioni e che si inserisce automaticamente quando i sensori determinanti ai fini del controllo automatico forniscono segnali inattendibili o si guastano: tipico è il caso dei tubi di Pitot, senza il cui segnale (condizione *unreliable IAS*) l'ADIRU si blocca ed il computer di bordo abbandona il velivolo nelle mani del pilota, lasciandogli il controllo diretto (*Direct Law*). Diversi Airbus A330 e A340 hanno manifestato problemi del genere e in qualche caso si è sfiorata la tragedia. Note che le ADIRU sono ben tre ed ognuna si sostituisce alle precedenti se queste vanno in avaria. In caso (improbabile) di guasto totale del computer, i comandi primari divengono inefficaci; per evitare il peggio sono stati previsti comandi di backup, che fino ai primi modelli dell'A340 erano il TRIM ed il timone, azionati meccanicamente tranne quando si è in *Normal Law*. Nei più

recenti modelli Airbus, ossia A340-600 ed A380, tutti i comandi principali (timone e TRIM sono meccanici, mentre alettoni ed elevatore hanno un circuito elettrico di scorta) hanno un backup, ossia possono saltare il computer in avaria. In teoria, per atterrare bastano TRIM e timone, almeno in condizioni meteo favorevoli.

I dubbi sollevati dal sistema Airbus sono tanti, primo fra tutti quello che riguarda la formazione dei piloti - secondo alcuni - non preparati per affrontare situazioni del genere: l'incidente dell'AF447 lo ha messo in luce ed il rapporto conclusivo del BEA (*Bureau d'Enquetes et de Analyses*) suggerisce di inserire, nel training dei piloti, la simulazione del volo in condizioni atmosferiche difficili senza l'aiuto del computer e con l'avaria delle ADIRU, che nell'incidente si sono disattivate una dopo l'altra. Senza contare che l'aiuto del computer permette di pilotare aerei diversi con una sola abilitazione; questo è stato uno dei punti di forza della politica commerciale Airbus: costi ridotti dei velivoli (anche perché A330 e A340 sono l'A300 allungato e con ali e motori diversi), il che ha ridotto i costi di progettazio-



ne, (mentre i Boeing sono stati tutti progettati da zero) e minori spese di formazione, dato che l'aereo è reso più semplice e perché le sue prestazioni aerodinamiche possono essere corrette dal computer. Il dubbio è cosa accade quando il pilota deve fare i conti con le differenze. Più cauto è stato l'approccio della Boeing, che ha esordito nel suo sviluppo del FBW con il 777, uno dei modelli più diffusi al mondo grazie ai bassi consumi dei suoi due (le norme ETOP consentono l'impiego in voli transoceanici con 2 soli motori) giganteschi motori General Electric GE90, che lo rendono più economico del diretto concorrente: il quadrimotore A340-600 (l'Airbus lavora alacremente al suo nuovo bimotore A350 per riprendere la quota di mercato persa).

Il sistema Boeing prevede la possibilità di assistere il pilota e correggerne i comandi, ma può essere completamente disinserito: insomma, mentre negli Airbus i comandi entrano nel computer e da lì escono, nei Boeing FBW i comandi sono semplicemente assistiti ed il computer può essere completamente escluso. Dunque, il Fly-By-Wire Boeing è una sorta di autopilota evoluto, nulla più. Inoltre la Boeing, come tutti gli altri costruttori che hanno sposato la filosofia del Fly-By-Wire, ha mantenuto il tradizionale volantino, che per molti piloti è più maneggevole nelle forti turbolenze, se non altro perché si può afferrare con due mani ed è centrato rispetto al sedile. Ancora, nel FBW della Boeing la condizione di *unreliable IAS* viene semplicemente segnalata al pilota, mentre nell'Airbus disinserisce il comando computerizzato dell'aereo e porta in *Direct LAW* o, peggio, prima di farlo può indurre il computer ad alterare l'assetto di volo.

la subordina alla loro effettiva eseguibilità, valutata in base ai parametri caricati nel software. Il governo in volo è affidato ad un computer più due di riserva, che difficilmente vanno tutti insieme in avaria, anche se alcuni incidenti riguardanti Airbus A330 (passati in sordina o quasi...) lasciano qualche dubbio ed il più eclatante di questi, accaduto all'Air France AF447 nel 2009, pone tanti interrogativi.

Il computer principale dei moderni Airbus media i comandi del pilota e può correggerli, rendendo di fatto più sicuro e facile il governo dell'aereo e sollevando il pilota da operazioni ripetitive che possono stancarlo, specie nei lunghi voli

notturni. Il grado di intervento del computer può essere ridotto fino al pieno controllo manuale, anche se in questa modalità in realtà i comandi passano sempre dal computer, che però non li filtra. Per scongiurare il pericolo che in caso di totale avaria del computer i comandi si interrompano e l'aereo diventi ingovernabile, nei primi A320, nonché in tutta la serie A330 e negli A340-200, 300, 500, è stato previsto un backup per il TRIM dell'elevatore ed il timone: questi comandi sono trasmessi meccanicamente dalla rotella del TRIM e dalla pedaliera alle rispettive superfici di controllo. Nei più recenti A340-600 e nel gigantesco A380, il backup elettrico è stato esteso a

tutti i comandi principali, ossia timone, elevatore ed alettoni; così se saltano tutti i computer c'è sempre la linea diretta, anche se manca il force-feedback ed il pilota deve evitare manovre brusche. Anche la Boeing ha intrapreso la strada del Fly-By-Wire, ma con maggior cautela: infatti, pur avendo adottato il sistema nei B737 NG (dove c'è il backup meccanico), B777, nel 787 Dreamliner e nel nuovissimo B747-8 (efficientissimo sul piano energetico, tanto da consumare circa il 10 % meno del diretto concorrente, l'A380) lascia al computer il ruolo di assistente e non impernia su di esso il governo dell'aereo, che resta al pilota. Per esempio, nel 777 viene adottato un

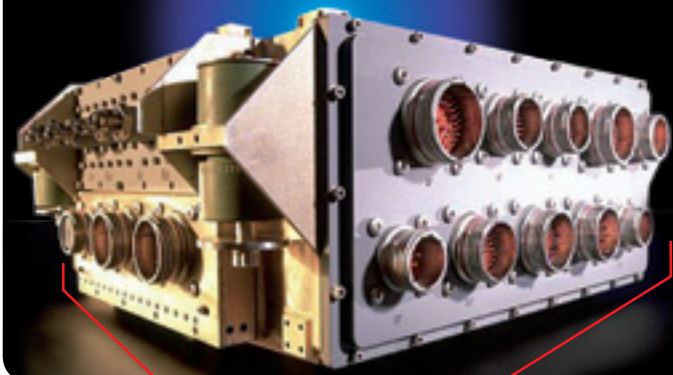
Il motore? Ci pensa il FADEC

I motori sono una parte fondamentale degli aerei: li portano in volo, garantiscono la velocità di crociera e mantengono la spinta che serve ad evitare lo stallo. Sono però delicati ed esigenti, perciò non basta "dare manetta" ma, come in auto, bisogna controllare la pressione dell'olio, il numero di giri, la tempera-

tura ecc. Fare tutto questo, soprattutto nei plurimotori, è un compito troppo gravoso per il pilota, tant'è che rientrava tra le mansioni dell'ingegnere di volo. Perciò si è pensato di affidare il controllo e l'ottimizzazione del funzionamento dei motori all'elettronica; così sono nati i primi sistemi di gestione che impostano i

parametri sulla base della richiesta di potenza del pilota e riportano in cabina eventuali segnali d'allarme. Il primo sistema del genere è stato un controllo automatico analogico, che comandava gli attuatori necessari alla variazione dei parametri di funzionamento del motore; è stato usato negli anni '60 del secolo scorso sul motore Rolls Royce Olympus 593 installato sul Concorde. Per ovviare alla sensibilità alle interferenze tipica del controllo analogico, venne progettato (da NASA e Pratt & Whitney, tra i più importanti produttori di motori a getto) il sistema che ancor oggi viene utilizzato: il FADEC. Si tratta di un computer che,

valutando densità dell'aria, posizione della manetta, temperatura dei gas di scarico, numero dei giri delle turbine ecc. (acquisiti ed elaborati decine di volte al secondo perché possono variare a seconda delle condizioni atmosferiche e di volo) regola i parametri del motore, tra cui il flusso di carburante e la posizione dei vani statorici e delle valvole di sfogo, mediante attuatori elettrici o servocomandi. Il FADEC controlla anche l'avviamento e regola i parametri per garantire la migliore efficienza del motore in ogni condizione di volo; inoltre protegge il motore prevenendo operazioni al di fuori dei suoi limiti costruttivi, per esempio limitando i giri



FBW che però comanda elettrovalvole e mantiene i circuiti idraulici (uno principale e due di scorta); il computer di bordo si interfaccia con ampi pannelli LCD touchscreen in cabina e può correggere i comandi del pilota, se ritiene che possano compromettere l'assetto (rischio di stallo) o la struttura del velivolo, solo che può essere escluso del tutto. Nel Boeing 787 è stata utilizzata una variante digitale di FBW a fibra ottica: in pratica i comandi non sono diretti ma si propagano dal computer mediante una rete locale ad alta affidabilità (senza collisioni dei pacchetti di dati); restano comunque i comandi di backup elettrico diretti. La sicurezza è quindi garantita. Questa fondamentale differenza

rende gli Airbus con FBW più delicati, perché se da un lato pilotarli stanca di meno ed espone a minori rischi nei riguardi degli errori dei piloti, quando il computer va in avaria può diventare molto difficile governarli; infatti il computer degli Airbus media le intenzioni del pilota, il quale in realtà nella gran parte del volo si limita a dire cosa vuole fare, poi il computer esegue. Ciò ha comportato da un lato una minore preparazione dei piloti ad affrontare situazioni di avaria che il costruttore ritiene "quasi impossibili" (ed è il quasi che preoccupa...) e dall'altro l'incapacità di governare certi aerei senza computer in condizioni di volo particolarmente critiche. A ciò va aggiunto che, per ragioni economiche, Airbus ha contato molto sull'intervento del computer per consentire ad un pilota, in possesso di un'unica abilitazione, di pilotare un'intera classe di velivoli: ad esempio A330 ed A340. Ciò perché il computer adatta il governo esercitato dal pilota all'aerodinamica e a tutte le altre caratteristiche in volo del velivolo, quindi quando manca l'aiuto del

computer il pilota deve fare i conti con la realtà fisica dell'aereo. Attualmente altri costruttori si stanno orientando verso il FBW: la brasiliana Embraer, terzo costruttore mondiale di aerei, lo adotta nei jet civili della serie ERJ, ed anche la canadese Bombardier (che opera anche nel settore dei trasporti ferroviari) lo usa nei più recenti CRJ. Se sapere di volare su un aereo comandato elettronicamente vi inquieta perché temete che i circuiti di bordo possano rimanere senza corrente (per un guasto o perché finisce il carburante e quindi i generatori azionati dai motori non funzionano più) sappiate che c'è sempre l'unità ausiliaria APU: si tratta di un generatore elettrico (quello che sentite sibilar quando l'aereo è a terra ed ha i motori fermi) azionato da una piccola turbina, che prima dell'avviamento dei motori fornisce elettricità a tutti i servizi di bordo e che viene avviato per primo; ebbene, l'APU può fornire elettricità in emergenza. Se anche l'APU non funziona, i moderni aerei se la cavano comunque, perché da sotto la

o mantenendo la temperatura dei gas di scarico (EGT) nei limiti.

Registra inoltre, in un'apposita memoria a disposizione dei tecnici di manutenzione, i parametri di funzionamento e le eventuali anomalie.

Siccome il FADEC è un sistema "non aggirabile", nel senso che se si guasta non è possibile controllare i motori, nei velivoli moderni ne sono installati almeno due per propulsore, ad assicurare la necessaria ridondanza. In pratica ogni FADEC è composto di due unità complete (si chiamano ECU nei motori General Electric ed EEC nei Pratt & Whitney) che lavorano indipendentemente l'una dall'altra. Normalmente un canale controlla il motore, mentre l'altro verifica il corretto funzionamento del sistema; se va in avaria il canale attivo (o un suo sottosistema) il canale di riserva prende il controllo. Ad ogni avviamento del motore, circuiti di autodiagnosi controllano il corretto funzionamento del FADEC e dei suoi sensori.

fusoliera esce automaticamente un aerogeneratore a turbina (RAT, *Ram Air Turbine*) che sfrutta la forza dell'aria impressa dall'avanzamento del velivolo per generare elettricità, quanta ne basta a caricare delle batterie e far funzionare il Fly-By-Wire. La RAT è stata adottata anche in aerei tradizionali come il Boeing 767 (salvò la vita ai passeggeri del celebre *Gimli Glider*, un 767 della Air Canada che planò da oltre 12.000 metri dopo aver finito il combustibile) e l'MD-11, dove alimenta anche le pompe degli impianti idraulici. Il suo contributo, chiaramente, viene meno nell'approccio al suolo, dato che diminuendo la velocità dell'aereo cala la potenza eolica disponibile.

SENZA TIRANTI E SENZA FILI, SI CHIAMA FLY-BY-WIRELESS

Nel 2010 un istituto di ricerca portoghese ha sperimentato una variante, particolarmente azzardata, del FBW: partendo dalla considerazione che, seppure meno dei sistemi meccanici, i cavi elettrici pesano, i ricercatori hanno ipotizzato la possibilità di trasmettere i comandi dal *cockpit*

via radio. Il sistema è stato battezzato *Fly-By-Wireless* e si ispira alla rete a fibra ottica del Boeing 787, solo che impiega, per impartire i comandi alle superfici di governo e ricevere informazioni dai sensori di bordo, una rete wireless. Naturalmente l'aereo è per ora solo un modello telecomandato, dato che ci sono troppi interrogativi sull'applicabilità ai *liner* di questa tecnologia; infatti, mentre i fili del FBW rappresentano un mezzo di trasmissione sicuro, una rete wireless ha una limitata affidabilità, dovuta alla sensibilità dei link radio ai disturbi elettromagnetici (ad esempio quelli dovuti ai cellulari, che potrebbero bloccare o rallentare la trasmissione dei comandi) tale da facilitare attentati terroristici.

SISTEMI DI NAVIGAZIONE

L'elettronica non ha rivoluzionato solo il governo dell'aereo, ma anche il controllo del traffico: fino a qualche anno fa i velivoli avevano a bordo sistemi radio di identificazione e comunicazione abbastanza semplici e le conversazioni tra piloti e centri di controllo a terra (ma anche tra piloti di aerei in volo) avvenivano in VHF, nella gamma sopra la radiodiffusione (tipicamente in modulazione d'ampiezza, da 110 a 130 MHz) mentre l'identificazione veniva ottenuta con un transponder attivo, ossia un apparato (teorizzato per la prima volta nella Seconda Guerra Mondiale per sopperire all'incapacità del radar di distinguere gli aerei nemici da quelli alleati...) che, stimolato da un segnale radio, trasmette un codice contenente l'identificativo dell'aereo. Il transponder consente di realizzare il cosiddetto *Radar Anticollisione* o SSR (*Radar Secondario di Sorveglianza*) derivato dal sistema militare IFF. Il transponder, eccitato da un segnale a 1,03 GHz, risponde su

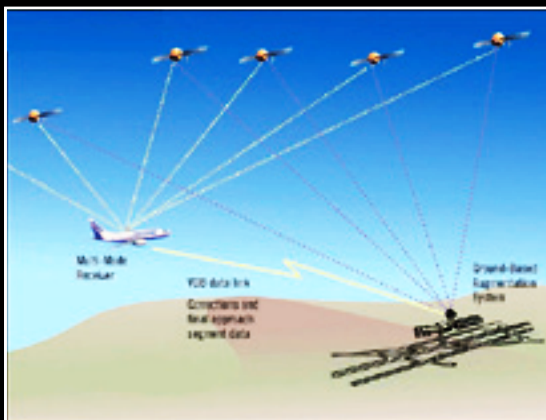
una portante a 1,090 GHz; la portata del sistema è fino a 200 miglia nautiche. L'interrogazione del radar è costituita da due *parentesi* dette P1 e P3, di durata di 0,8 μ s, la cui distanza determina il modo d'interrogazione: in modo A viene richiesta l'identità e la distanza tra P1 e P3 è di 8 μ s; in modo C viene chiesta la quota, e la distanza tra P1 e P3 è di 21 microsecondi.

Al radar anticollisione si è aggiunto il ricevitore VOR, capace di interagire con il pilota automatico o semplicemente di visualizzare, su un apposito display, la rotta e le deviazioni rispetto ad essa. Il VOR (*VHF Omnidirectional Range*) è un sistema di radioguida che serve a tracciare un corridoio virtuale per guidare gli aerei; in pratica indica la posizione radiale dei velivoli rispetto a una trasmittente che genera due segnali. Il suo funzionamento è molto semplice: una stazione di terra VOR, chiamata anche *radiofaro*, trasmette due segnali radio in VHF che vengono captati da un ricevitore a bordo dell'aereo; uno dei due è direzionale e l'altro cambia di fase ruotando di 360 gradi 30 volte al secondo, nel senso che idealmente viene trasmesso in una direzione che varia continuamente come se l'antenna emittente girasse come quella del radar. Ogni volta che il segnale "rotante" giunge dalla stessa direzione di quello fisso, lo sfasamento tra i due si annulla, mentre diventa 90° ad est, 180° a sud e 270° ad ovest. Conoscendo la velocità di rotazione della fase e la differenza di fase, il ricevitore del VOR riesce a determinare in ogni istante qual è l'angolazione della prua del velivolo rispetto al radiofaro. Spesso le stazioni di terra sono dotate anche del DME (*Distance Measuring Equipment*) che misura la distanza tra l'emittente e il ricevitore e permette di valutare se l'aereo sta viaggiando

GLS: atterraggi sicuri con Galileo

Quando le condizioni di visibilità impediscono l'atterraggio manuale in sicurezza, entra in gioco l'ILS, ossia il sistema di atterraggio strumentale; oggi, con la certificazione Safety Of Life del segnale EGNOS del progetto Galileo, si punta all'integrazione dell'ILS con la localizzazione satellitare. L'ILS si basa su un apparato di trasmissione a terra, operante in VHF tra 108,10 e 111,95 MHz ed in UHF tra 328,6 e 335,40 MHz, i cui segnali radio creano un sentiero che consente agli aerei di avvicinarsi alla pista con un angolo di discesa che permette di toccare terra esattamente all'inizio della pista. Al suolo, il sistema è formato da tre apparati denominati LOC, GP e MARKER. I marker, posti lungo il prolungamento dell'asse della pista, sotto la traiettoria prevista dell'aeromobile, emettono un segnale molto direttivo verso l'alto: l'antenna più lontana (*Outer Marker*) è posta tra 6 e 11 chilometri dall'inizio della pista, quella di mezzo (*Middle Marker*) tra 900 e 1.200 m, mentre l'*Inner Marker* si trova ad inizio pista. A bordo dell'aereo c'è un apparato composto da un ricevitore multifrequenza e da un quadrante che indica di quanto l'aereo si discosta lateralmente e orizzontalmente dal sentiero ideale di discesa. Il sorvolo dei marker viene evidenziato da segnali acustici e luminosi.

Esistono tre categorie di ILS (CAT I, CAT II e CAT III) che definiscono la precisione garantita dal sistema di terra; la terza si suddivide in tre sottocategorie (CATIIIA, CATIIIB e CATIIIC) l'ultima delle quali consente l'atterraggio anche in condizioni di visibilità nulla.



Oggi si sta sperimentando un sistema di avvicinamento basato sulla localizzazione satellitare e che viene chiamato GLS (*GPS Landing System*); lo scopo è consentire l'atterraggio strumentale dove manca e non è economicamente conveniente installare l'ILS, ma anche sfruttare la localizzazione satellitare (grazie al fatto che il sistema Galileo permette di garantire l'autenticità della fonte del segnale di localizzazione e la sua qualità) su cui si basa anche l'ADS-B.

Da qualche mese il segnale del sistema di correzione europeo dell'errore di posizione (EGNOS) è stato certificato come ILS CAT I e viene testato nell'atterraggio strumentale anche in Italia. Il funzionamento del GLS è molto semplice: alcuni ricevitori GPS vengono installati in prossimità delle piste, in posizioni determinate con la massima precisione; ricevendo il segnale valutano il posizionamento che esso fornisce e lo confrontano con quello reale, determinando lo scarto e quindi il fattore di correzione da applicare ai dati desunti dal GPS. Il segnale d'errore viene trasmesso mediante una portante VHF a tutti gli aerei in prossimità dell'aeroporto, i cui apparati di avvicinamento strumentale lo utilizzano per correggere i dati forniti dal ricevitore GPS di bordo, ottenendo così una precisione inferiore al metro, che consente atterraggi anche in condizioni di visibilità nulla. Un sistema del genere è più economico dell'ILS, perché richiede apparecchiature di terra e di bordo meno costose; è inoltre più affidabile e richiede una minore manutenzione. In un futuro non lontano, il GLS potrebbe permettere agli aerei di atterrare in maniera completamente automatica, come avviene con gli UAV.

verso la stazione emittente o si sta allontanando da essa. I VOR funzionano entro un raggio che va dai 46 ai 240 chilometri ed operano su frequenze, valide in tutto il mondo, comprese fra i 108 e i 117,95 MHz, con ampiezza del canale di 50 kHz. I due segnali emessi dai radiofari per codificare la direzione sono a 30 Hz: quello fisso è modulato in AM e contiene il nome della stazione (in codice Morse); l'altro è in FM, modulato su una sottoportante di 9.960 Hz. Quando il ricevitore sull'aereo aggancia il VOR, la parte di segnale modulata in FM viene discriminata e confrontata con quella in AM per desumere la differenza di fase, la quale viene poi miscelata con un'onda prodotta localmente e il segnale risultante, amplificato, pilota i puntatori della *girobrussola*, ossia dello strumento che mostra la prua dell'aereo rispetto ai punti cardinali. Cambiando la fase locale, utilizzando la manopola conosciuta come *Omni-Bearing Selector* (OBS) il pilota può azzerare l'angolo di una stazione; ad esempio, se desidera volare a 90 gradi rispetto a una stazione, l'OBS fa in modo che il segnale locale dia -90 gradi di fase in modo che l'ago dell'indicatore segni zero (centrato) quando il velivolo sta volando a 90 gradi rispetto al radiofaro. Per seguire una direzione, basta fare in modo che lo sfasamento rimanga costante, ossia dirigere il velivolo in modo che l'indicatore punti sempre nella stessa direzione. Proseguendo con gli apparati elettronici di navigazione, non possiamo non parlare dell'ILS (*Instrumental Landing System*) che si basa su stazioni a terra in grado di emettere segnali direzionali agganciandosi ai quali, un apposito apparato di bordo, consente di allinearsi alla pista quando le condizioni di visibilità non permettono al pilota di farlo con certezza.



ADIRU: più di una piattaforma inerziale

Oggi, per risparmiare sui costi di tali sistemi e consentire un atterraggio sicuro in quegli aeroporti in cui manca l'ILS, si sta sperimentando l'avvicinamento assistito da satellite (GLS).

Recentemente è stata avviata anche la sperimentazione dell'ADS-B che potrà raggruppare le funzionalità di molti apparati sia a terra che a bordo; per comprendere tale sistema va ricordato che per controllare la posizione degli aerei in volo (ma anche di quelli in movimento lungo le piste degli aeroporti) vengono utilizzati radar che di solito hanno una portata compresa tra 20 e 100 miglia. I dati ricavati vengono visualizzati dai monitor dei centri di controllo del traffico aereo, i cui addetti verificano la distanza tra un velivolo e l'altro, controllano la quota di volo e instradano gli aeromobili verso l'aerovia cui sono destinati.

Con il perfezionamento del GPS civile e soprattutto dell'EGNOS (e del suo corrispondente americano, il WAAS) è stato possibile, installando a bordo un ricevitore, conoscere con esattezza la posizione e l'altezza dal suolo del velivolo; se prendiamo questi dati e li inviamo alle stazioni di terra, otteniamo le informazioni fornite dalla rete radar, ma più precise. Il sistema che ne deriva prende il nome di ADS-B (*Automatic Dependent Surveillance-Broadcast*) e può essere usato per trasmettere sia a terra, sia agli altri aerei in volo dotati di analoga apparecchiatura, dati come l'identificativo, la posizione, la rotta, il tipo di aeromobile e la compagnia, il numero del volo ed altro ancora. Un'interessante dimostrazione del funzionamento dell'ADS-B può essere seguita sul sito www.flightradar.com, il cui server riceve i segnali di ricevitori piazzati qua e là per il globo terrestre e ne decifra i dati, fornendo le posizioni sufficientemente

Negli aerei Fly-By-Wire prodotti dall'Airbus, in Normal Law il computer può governare il volo grazie ai dati che gli sono passati dall'ADIRS (*Air Data Inertial reference System*) e che vengono anche visualizzati sugli LCD dell'EFIS (*Electronic Flight Instrumentation System*) in cabina. Tali dati riguardano velocità, pressione atmosferica, temperatura esterna, ma anche l'assetto di volo. L'ADIRS è composto da tre ADIRU, ognuna delle quali è l'insieme di una piattaforma inerziale (IR, concettualmente simile all'ARDUIMU basata su Arduino che abbiamo usato nel nostro progetto di UAV pubblicato nei fascicoli 154 e 155) e di una ADR (*Air Data reference*): quest'ultima fornisce altitudine barometrica (presa di pressione statica), airspeed (tubo di Pitot), mach, AOA, temperatura, mentre la piattaforma inerziale (*Inertial Reference*) fornisce i dati di posizione,

assetto, vettore della velocità, accelerazioni (accelerometri) accelerazioni angolari (giroscopi). I segnali dei sensori analogici vengono convertiti in digitale dagli ADM per poter essere letti dalle ADIRU.



ADIRU dell'Airbus A330.

aggiornate degli aerei che stanno sorvolando una certa zona.

La necessità di sistemi di assistenza al volo basati su satellite nasce essenzialmente per fare fronte al continuo aumento del traffico aereo, ma anche perché gli attuali sistemi di controllo del traffico aereo non coprono tutto il globo (a causa della limitata portata dei radar o per l'assenza di valide infrastrutture): ne soffrono, ad esempio, i voli transoceanici. Inoltre il prezzo di un'apparecchiatura ADS-B di bordo è dell'ordine del migliaio di euro, quindi è possibile beneficiare della tecnologia anche su piccoli velivoli da turismo. Nel controllo a terra (la gestione del movimento degli aerei in aeroporto) l'ADS-B è utilissimo e si candida a sostituire il radar di controllo a terra usato per evitare collisioni in condizioni di scarsa visibilità (come l'incidente di Milano Linate, dove un MD-82 investì un piccolo jet privato, finendo poi in fiamme su un hangar, o il disastro di Tenerife, in cui un Boeing 747 in decollo finì su un altro in rullaggio) o congestione del traffico. I dati trasmessi dai velivoli dotati di trasmettitore ADS-B vengono

captati da una rete di stazioni riceventi di terra (molto più economiche di quelle radar) collegate ai centri di controllo; nel caso di voli transoceanici, il segnale viene captato da una rete di satelliti per radiocomunicazioni e ritrasmessi a terra. Lo stesso segnale radio può essere ricevuto dagli altri aerei nelle vicinanze i quali possono così visualizzare su un monitor la situazione del traffico. La stringa di dati viene trasmessa ogni due secondi sulla frequenza di 1,09 GHz utilizzando il Modo S extended; il segnale radio è modulato in PPM da una stringa di 112 bit alla velocità di 1 Mbps. Il segnale identificativo può essere codificato per evitare che un trasmettitore pirata possa generare segnali fasulli tali da creare problemi al controllo del traffico aereo.

Chiudiamo questa panoramica sull'elettronica di bordo con due particolari: l'illuminazione in cabina a LED, che nei B767 (ad esempio della NEOS) ha sostituito quella tradizionale consentendo di risparmiare elettricità e di variare la colorazione; le scatole nere, che oggi sono completamente digitali e solid-state.

Corri in edicola... per te il volume speciale “Primi passi con ARDUINO”

solo
€ 9,90



**Impara a realizzare i tuoi progetti interattivi
con la piattaforma Made in Italy più famosa al mondo.**



Realizziamo un apparato in grado di interfacciarsi ai sintetizzatori musicali dotati di porta MIDI e far loro suonare sequenze di note comandandoli da uno smartphone Android, mediante il protocollo OSC.

ANDROID ARDUINO SEQUENCER

di ANDREA BELLONI

In questo articolo descriviamo la realizzazione di un semplice MIDI sequencer implementato con Arduino e controllato, via Wi-Fi, da uno smartphone Android per il quale abbiamo sviluppato un'applicazione che gestisce l'interfaccia grafica e la comunicazione con Arduino. Nella Fig. 1 vedete uno schema a blocchi dell'intero sistema. Il MIDI (Musical Instrument Digital Interface) è un protocollo standard usato per consentire a strumenti musicali elettronici, computer e altri dispositivi elettronici, di scambiarsi messaggi relativi,

per esempio, all'altezza delle note da suonare e all'intensità. Un MIDI sequencer è un dispositivo che permette di registrare e riprodurre sequenze di messaggi per controllare, mediante il protocollo MIDI, strumenti musicali elettronici. Nell'utilizzo più tipico, nel sequencer vengono impostate sequenze di note di diversa altezza e intensità, che una volta inviate a uno o più strumenti musicali compongono un brano. Per realizzare il MIDI sequencer con Arduino abbiamo usato il MIDI Shield (Fig. 2) e scritto uno



Fig. 1

sketch per consentire ad Arduino di registrare le sequenze di note ed inviarle al MIDI Shield, oltre che gestire la comunicazione con lo smartphone Android. La gestione dell'interfaccia grafica del MIDI sequencer è stata implementata con un'apposita applicazione scritta per uno smartphone Android, che permette di scegliere il canale (diversi canali = diversi strumenti nel protocollo MIDI), di impostare la sequenza di note relative al canale e di gestire la comunica-

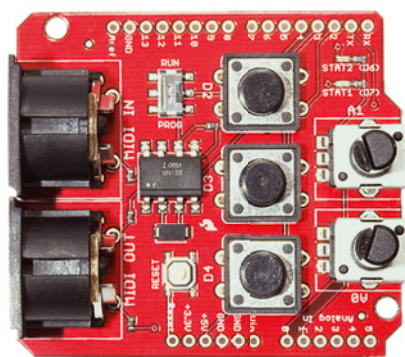


Fig. 2

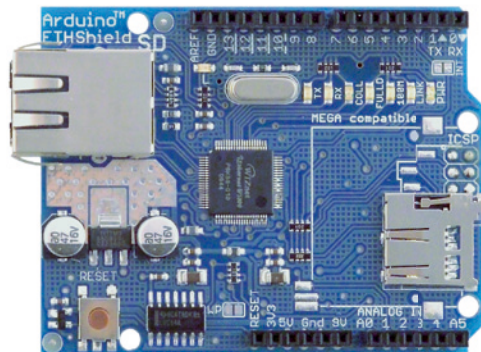


Fig. 3

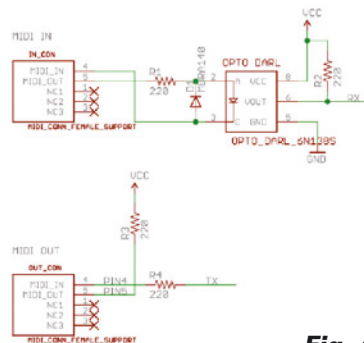


Fig. 4

zione con Arduino mediante lo scambio di messaggi OSC. Il Protocollo OSC (Open Sound Control) nasce con intenti simili a quelli del MIDI ma, essendo stato sviluppato più recentemente, consente di ottenere migliori prestazioni, maggiori risoluzioni e il controllo di una più ampia gamma di parametri musicali. Inoltre, a differenza del protocollo MIDI, che come livello fisico usa una connessione seriale, i messaggi OSC vengono trasportati tramite una connessione di rete (Ethernet, Wi-Fi...). Quest'ultimo è il motivo principale per il quale abbiamo usato OSC in questa parte del nostro sistema (scambio di messaggi tra Arduino e Android). La connessione tramite Arduino e lo smartphone dotato di Android è stata realizzata tramite un comune router Wi-Fi, con Arduino collegato ad una porta Ethernet del router usando l'Ethernet Shield (vedi Fig. 3) e lo smartphone connesso via Wi-Fi.

L'HARDWARE DEL SEQUENCER

I componenti principali usati per implementare il MIDI sequencer con Arduino sono due: uno hardware ed uno software (più specificatamente, firmware). Il componente hardware è il MIDI shield che implementa il lato fisico dell'interfaccia MIDI. Il componente software è lo sketch sviluppato con l'Arduino IDE che implementa il lato logico

dell'interfaccia MIDI. Come vedete nello schema in Fig. 4, dal punto di vista fisico lo shield MIDI non è altro che una connessione elettrica con l'interfaccia seriale asincrona di Arduino realizzata mediante una resistenza (R4) per la parte TX (MIDI OUT) e tramite un fotoaccoppiatore per la parte RX (MIDI IN); quest'ultimo serve ad evitare problemi di ground-loop e la propagazione di eventuali disturbi lungo la catena di dispositivi MIDI collegati in cascata tramite le porte MIDI IN e MIDI OUT. Per soddisfare la specifica MIDI, l'interfaccia seriale asincrona (UART) di Arduino deve essere impostata per trasmettere e ricevere dati a 31.250 bps, con otto bit di dati, nessun bit di parità e un bit di stop (8-N-1). Nella nostra applicazione, dell'interfaccia MIDI viene usata solo la parte relativa al MIDI OUT: Arduino genera le note della sequenza man mano che il tempo avanza e le invia tramite questa porta allo strumento musicale elettronico che le suona; questo può essere sia un sintetizzatore stand-alone con interfaccia MIDI, che un sintetizzatore software che gira in un computer dotato di interfaccia MIDI.

IL FIRMWARE

La parte dello sketch che implementa il livello logico dell'interfaccia MIDI si preoccupa di generare ed inviare i messaggi

Listato 1

del protocollo attraverso l'interfaccia; nel nostro caso, dei vari messaggi che il protocollo prevede, vengono generati ed inviati solo i messaggi di tipo Note On e Note Off, entrambi implementati tramite la funzione **noteOn(int cmd, int pitch, int velocity)** (vedi **Listato 1**). Al parametro **cmd** vengono assegnati i valori 0x90, 0x91, 0x80 e 0x81 a seconda del tipo di messaggio da inviare, secondo il seguente schema: la prima delle due cifre esadecimali indica il tipo di messaggio (9 per Note On e 8 per Note Off) la seconda cifra indica il canale (nel nostro caso vengono utilizzati solo due dei sedici canali disponibili). Il parametro **pitch** (in italiano altezza della nota) viene usato per comunicare la nota da suonare tramite un intero che codifica il numero di nota MIDI. Nel **Listato 1** è possibile vedere l'array **pitch[8]** inizializzato con i numeri che corrispondono alle otto note da C3 a C4 (nella notazione musicale inglese) senza i semitoni (solo i tasti bianchi della tastiera).

L'ultimo parametro della funzione **noteOn()** è l'intero **velocity**; quest'ultimo permette di impostare l'intensità della nota (nel nostro semplice sequencer ha un valore costante uguale a 60). Il nome del parametro deriva dal significato originale, che è legato alla codifica della velocità con cui viene premuto il tasto sulla tastiera dello strumento musicale. Nel **Listato 1** è possibile vedere l'impostazione tipica di uno sketch per Arduino (abbiamo trascurato l'inizializzazione) a partire dalla funzione **loop()** che viene eseguita in un ciclo implicito fino al prossimo reset o spegnimento, ed infine le funzioni definite dall'utente per svolgere i vari compiti.

Per gestire correttamente il tem-

```
void loop()
{
    now = millis();
    if (now >= nextNote) {
        nextNote = now + beat;
        playNote(stepNote, oldNote);
    }
    if (now >= nextOSC) {
        nextOSC = now + beat/2;
        readOSC();
    }
    if (now >= nextStep) {
        nextStep = now + beat/3;
        syncStep(stepPos);
    }
    if (now >= nextButtons) {
        nextButtons = now + beat;
        readButtons();
    }
    if (now >= nextPot) {
        nextPot = now + beat*2;
        readPot();
    }
    if (now >= nextLED) {
        nextLED = now + beat*2;
        setLED(play);
    }
}

void noteOn(int cmd, int pitch, int velocity) {
    Serial.print(cmd, BYTE);
    Serial.print(pitch, BYTE);
    Serial.print(velocity, BYTE);
}

void playNote(int note[2][8][8], int old[2][8])
{
    if(play) {
        for (int j = 0; j < noteCount; j++) {
            int vel = old[0][j];
            if (vel > 0) {
                noteOn(0x80, pitch[j], 0);    // Turn note off
            }
            vel = old[1][j];
            if (vel > 0) {
                noteOn(0x81, pitch[j], 0);    // Turn note off
            }
        }

        // Now we Play the notes of the step
        for (int j = 0; j < noteCount; j++) {
            int vel = note[0][stepPos][j];
            old[0][j] = vel; // Save the note, to turn it off for the next step
            if (vel > 0) {
                byte cmd = 0x90;
                noteOn(cmd, pitch[j], vel);
            }
            vel = note[1][stepPos][j];
            old[1][j] = vel; // Save the note, to turn it off for the next step
            if (vel > 0) {
                byte cmd = 0x91;
                noteOn(cmd, pitch[j], vel);
            }
        }

        stepPos++; // Next step calculation
        if (stepPos == stepCount)
        {
            stepPos = 0;
        }
    }
}

void readOSC()
{
}
```

(Continua)

Listato 1 - segue

```
int i, j, c;

if(server.available()){
  rcvMes = server.getMessage();
  byte *ip = rcvMes->getIpAddress();

  i = rcvMes->getInteger32(0);
  j = rcvMes->getInteger32(1);
  c = rcvMes->getInteger32(2);

  if (i == 16 && j == 16)
  {
    for (int l = 0; l < noteCount; l++)
    {
      for (int k = 0; k < stepCount; k++)
      {
        stepNote[0][k][l] = 0;
        stepNote[1][k][l] = 0;
      }
    }
  }
  else
  {
    if (stepNote[c][i][j] > 0)
    {
      stepNote[c][i][j] = 0;
    }
    else
    {
      stepNote[c][i][j] = 60;
    }
  }
}

void syncStep(int stepPos)
{
  long int tmp=(long int)stepPos;

  Z_OSCMessage mes;

  mes.setAddress(remoteIp, remotePort);
  mes.setZ_OSCMessage(oscAdr, "i", &tmp);

  client.send(&mes);
}
```

po (suonare le note al corretto istante temporale) e non perdere cicli di clock in attese inutili, è stata implementata una versione minimale di pseudo multitasking cooperativo. Questo è stato realizzato con l'uso della funzione **millis()** che ritorna il numero di millisecondi trascorsi dall'accensione di Arduino. Alla fine della funzione **setup()** delle variabili (**nextOSC**, **nextNote**, **nextStep**, ...) vengono inizializzate con la funzione **millis()**; nella funzione **loop()** queste variabili vengono usate ed incrementate secondo uno schema che permette di eseguire le funzioni utente negli istanti temporali coretti. Scorrendo il **Listato 1**, gli oggetti

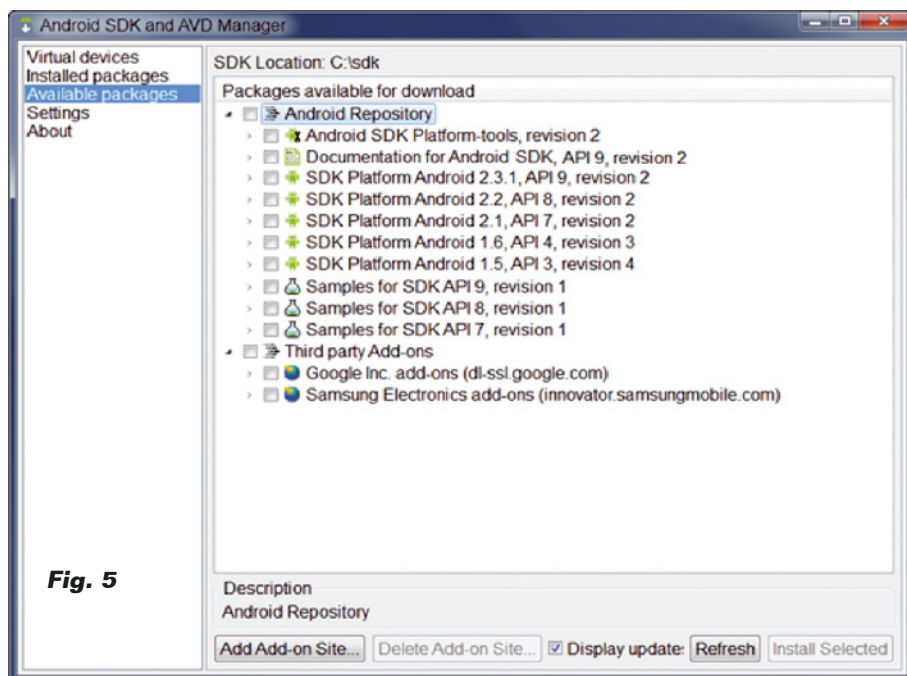


Fig. 5

rilevanti nell'implementazione del sequencer sono gli array **stepNote[2][8][8]**, **oldNote[2][8]** e **pitch[8]**, le funzioni **noteOn(int cmd, int pitch, int velocity)** e **playNote(int note[2][8][8], int old[2][8])**. L'array **stepNote[2][8][8]** mantiene lo stato delle note da suonare negli otto passi temporali (ripetuti ciclicamente) per ognuna delle otto note considerate e per ognuno dei due canali utilizzati. L'array **oldNote[2][8]** serve a memorizzare lo stato delle otto note suonate al passo temporale precedente all'attuale. La funzione **playNote(int note[2][8][8], int old[2][8])** è centrale al funzionamento del sequencer; inizia col verificare il contenuto dell'array **old[2][8]**, per ognuno dei due canali, in modo da fermare le note suonate al passo temporale precedente, poi, in base al contenuto dell'array **note[2][8][8]**, inizia a suonare le note del passo temporale attuale (**stepPos**) e ne salva lo stato nell'array **old[2][8]**. Infine, il passo temporale viene incrementato di uno ed eventualmente riportato a zero al raggiungimento di otto. Per suonare o fermare le note, viene usata la funzione **noteOn()**,

come descritto precedentemente. Gli ultimi oggetti interessanti da notare nello sketch del **Listato 1** sono le funzioni **readButtons()**, **readPot()** e **setLED(boolean play)**. Con la prima viene intercettata la pressione del bottone in alto nel MIDI Shield, che modifica il valore della variabile **boolean play** (da true a false e viceversa). La variabile **play** viene inizializzata a true e quando assume il valore false, Arduino smette di inviare le note alla porta MIDI OUT (vedi funzione



Fig. 6

`playNote()` all'inizio: `if (play) ...` e tramite la funzione `setLED()` accende i due LED presenti sullo Shield.

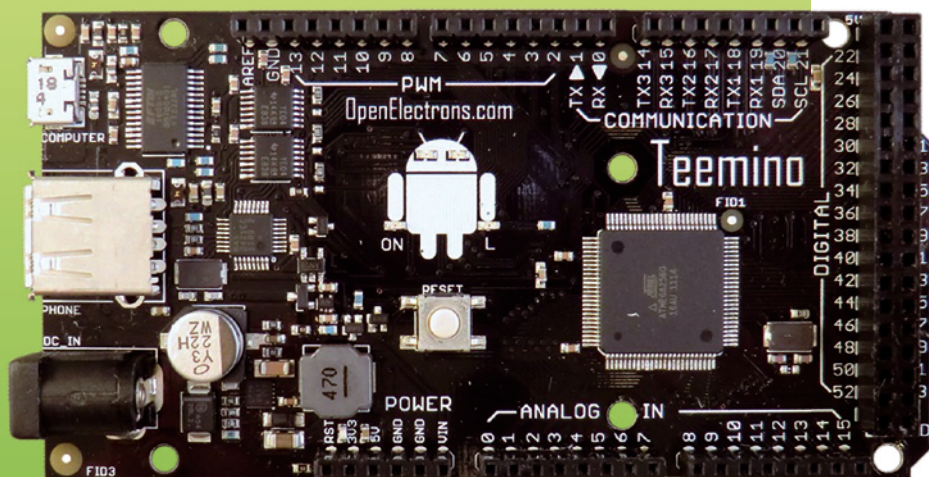
La funzione `readPot()` legge la porta di input analogica **A0**, alla quale è collegato il potenziometro in basso nel MIDI Shield. A seconda del valore letto, la variabile `beat` assume valori interi da un minimo di 90 ad un massimo di 330 (escluso). Per come viene usata nel loop principale dello sketch (nella funzione `loop()` appunto) la variabile `beat` imposta i BPM del sequencer, cioè il numero di battute al minuto.

INTRODUZIONE ALLA PROGRAMMAZIONE ANDROID IN PROCESSING

Prima di passare a spiegare come c'è fatta e in che modo funziona l'interfaccia utente su Android, facciamo un brevissimo tutorial sulla programmazione in Processing, linguaggio cui stiamo dedicando un apposito corso. L'ultima versione di Processing, la 1.5, prevede due modalità di sviluppo: una, quella classica, per desktop/web, e l'altra per Android. Quest'ultima è un novità importante, perché rende molto semplice sviluppare applicazioni per tutti quei dispositivi mobili (smartphone, tablet, ...) che usano il sistema operativo Android, perlomeno più semplice rispetto all'utilizzo dell'SDK nativo di Android.

Per iniziare a sviluppare in Processing per Android bisogna:

- scaricare l'Android SDK da qui (<http://developer.android.com/sdk/index.html>);
- installarlo seguendo le istruzioni visibili alla pagina web <http://developer.android.com/sdk/installing.html>;
- saltare il passo 3 ("Installing the ADT Plugin for Eclipse");
- al passo 4 ("Adding Platforms



Controllare Arduino con Android

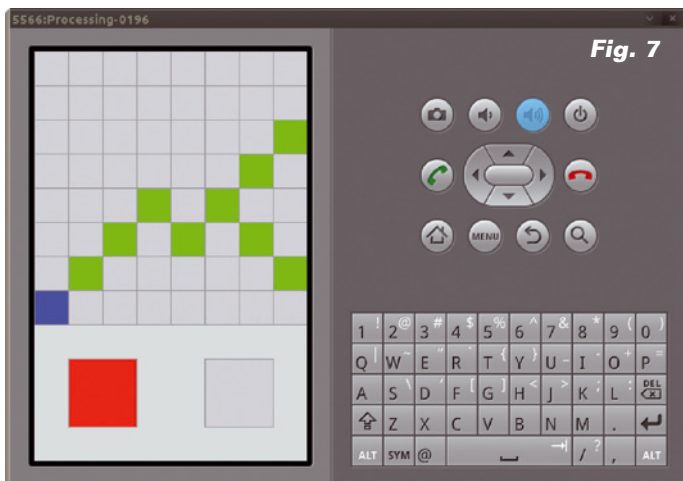
Arduino si avvicina sempre più al mondo Android, come dimostra la recente disponibilità di Teemino, una scheda di controllo basata su architettura Arduino ed equipaggiata con il microcontrollore ATMEGA ATmega 2560, dotato di host USB. Questa unità, che poi è un'Arduino Mega dotata di un firmware speciale, nasce per sviluppare applicazioni gestibili da smartphone equipaggiati con il sistema operativo di Google. Teemino può ospitare anche l'NXShield, che consente il controllo di quattro servomotori ed altrettanti sensori del sistema LEGO Mindstorms NXT. Le applicazioni si sviluppano usando la piattaforma ADK (Android open accessory Development Kit) accessibile da Android 3.1 o 2.3.4, mediante il supporto accessori Android, che consente ad hardware esterno USB di interagire con un dispositivo Android in una speciale modalità chiamata "accessorio". Come altri accessori gestibili da USB, Teemino risponde al protocollo Android Accessories. La particolarità di Teemino è che integra sia la normale connessione USB (con connettore mini-USB) per la programmazione da PC, sia un Host USB accessibile da un connettore USB-A, per dialogare con lo smartphone Android.

and Other Components") assicurarsi di selezionare (mettendogli il segno di spunta accanto, la voce "SDK Platform Android 2.1, API 7" sotto il percorso root "Android Repository";

- sotto il percorso Third party Add-ons, spuntare "Google APIs by Google Inc., Android

API 7" che potete vedere espandendo "Google Inc." (vedi Fig. 5).

Fatto ciò, scaricate l'ultima versione di Processing da www.processing.org; per installarla, non occorre alcuna particolare procedura: basta scompattare il file in una qualsiasi directory del disco



rigido del computer. Una volta eseguiti questi passi (per un maggior dettaglio vedere la relativa pagina wiki, <http://wiki.processing.org/w/Android>) per mandare in esecuzione l'ambiente di sviluppo di Processing basta un doppio clic sull'eseguibile. Quindi, se volete sviluppare per Android selezionate "Android" dal drop-down box della toolbar, che per impostazione predefinita è su "Standard" (vedi Fig. 6). Da questo momento è possibile scrivere programmi in Processing e mandarli in esecuzione, nell'emulatore o direttamente sul dispositivo (nel nostro caso si tratta di uno smartphone) collegato via USB, selezionando "Run in Emulator" o "Run on Device" dal menu "Sketch".

Per poter testare le applicazioni sul dispositivo, una volta collegato via USB al computer, occorre:

- attivare "Debug USB" da *Impostazioni* -> *Applicazioni* -> *Sviluppo* (sul dispositivo);
- se il sistema operativo usato è Windows o Linux è necessario un driver aggiuntivo come spiegato in questa pagina <http://developer.android.com/guide/developing/device.html>, sotto "Setting up a Device for Development", punto 3.

Nel **Listato 2** è possibile esaminare lo sketch (anche in questo caso abbiamo trascurato l'inizializzazione) per Processing in modalità Android, scritto per implementare l'interfaccia utente del sequencer e la comunicazione con Arduino. Anche qui possiamo vedere la struttura tipica di uno sketch Processing, simile a quella dello sketch per Android. All'inizio troviamo la funzione **setup()** eseguita una sola volta all'avvio dell'applicazione; segue la funzione **draw()** che, analogamente alla funzione **loop()** in Arduino, implica un loop infinito delle istruzioni in essa contenute. Infine troviamo le funzioni definite dall'utente. Nel **Listato 2** si possono vedere anche delle funzioni che sono peculiari della modalità Android di Processing (se lo sketch in questione fosse per Processing, in modalità desktop "Standard" non

Listato 2

```
void setup() {
  size(screenWidth, screenHeight, A2D);
  orientation(PORTRAIT);

  rectSize = screenWidth/8;

  stroke(128);
  fill(192);

  int x, y, i, j;
  for (x = 0, i = 0; i < 8; x += rectSize, i++) {
    for (y = 0, j = 0; j < 8; y += rectSize, j++) {
      rect(x, y, rectSize, rectSize);
      stepNote[0][i][j] = 0;
      stepNote[1][i][j] = 0;
    }
  }

  x = rectSize;
  y = rectSize * 9;
  rect(x, y, rectSize * 2, rectSize * 2);
  x = rectSize * 5;
  y = rectSize * 9;
  rect(x, y, rectSize * 2, rectSize * 2);
}

void onResume() {
  super.onResume();

  oscP5 = new OscP5(this, 10023);

  remoteAddr = new NetAddress("192.168.1.177", 8888);

  OscMessage myMessage = new OscMessage("/monoid/grid");

  myMessage.add(16);
  myMessage.add(16);
  myMessage.add(0);

  oscP5.send(myMessage, remoteAddr);
}

void onPause() {
  oscP5.dispose();
  super.onPause();
}

void draw() {
  int ch = 0;
  int x, y, i, j;

  if (button0) {
    x = rectSize;
    y = rectSize * 9;
    fill(color(255, 0, 0));
    rect(x, y, rectSize * 2, rectSize * 2);
    x = rectSize * 5;
    y = rectSize * 9;
    fill(192);
    rect(x, y, rectSize * 2, rectSize * 2);

    ch = 0;
  }
  else if (button1) {
    x = rectSize;
    y = rectSize * 9;
    fill(192);
    rect(x, y, rectSize * 2, rectSize * 2);
    x = rectSize * 5;
```

(Continua)

ci sarebbero); queste sono **onResume()**, **onPause()** e **surfaceTouchEvent(MotionEvent event)**. Le funzioni **onResume()** e **onPause()** entrano in azione quando, nell'interattività tipica di uno smartpho-

Listato 2 - segue

ne Android, si entra e si esce dall'applicazione; nel nostro caso contengono l'inizializzazione e la chiusura della connessione in rete con Arduino. La funzione **surfaceTouchEvent(MotionEvent event)** serve a gestire gli eventi di interazione dell'utente con la superficie touch del dispositivo; nella nostra applicazione viene usata per conoscere le coordinate x e y relative al "tap" dell'utente sullo schermo e a gestire il relativo evento.

Anche nell'applicazione per Android abbiamo l'array **stepNote[2][8][8]** che tiene conto dello stato delle otto possibili note da suonare negli otto passi temporali per i due canali MIDI utilizzati. Nella funzione **setup()** l'applicazione crea una griglia di 8x8 quadrati sullo schermo dello smartphone (vedi Fig. 7) che diventano altrettante aree sensibili per impostare (con un tap) e visualizzare (mediante il cambiamento di colore) le note da suonare negli 8 passi. Sotto la griglia dei quadrati relativi alle note vengono posizionati altri due quadrati utilizzati per la selezione del canale MIDI. La gestione degli eventi legati ai tap sui quadrati delle note e su quelli dei canali avviene all'interno della funzione **surfaceTouchEvent(MotionEvent event)** dove viene cambiato il valore dell'elemento dell'array **stepNote[][][]** corrispondente al quadrato "tappato" e al canale scelto. Ogni cambiamento di stato dell'array **stepNote[][][]** viene comunicato ad Arduino con un messaggio OSC che riflette questo cambiamento nell'array corrispondente. Nella funzione **draw()** lo schermo del dispositivo Android viene aggiornato costantemente per riflettere le modifiche effettuate dall'utente e per tenere conto del tempo che scorre ciclicamente attraverso gli

```

y = rectSize * 9;
fill(color(255, 0, 0));
rect(x, y, rectSize * 2, rectSize * 2);

ch = 1;
}

for (x = 0, i = 0; i < 8; x += rectSize, i++) {
  for (y = 0, j = 0; j < 8; y += rectSize, j++) {
    if (stepNote[ch][i][j] > 0) {
      if (step == (i)) {
        r = 0;
        g = 0;
        b = 255;
      }
      else {
        r = 0;
        g = 255;
        b = 0;
      }
      fill(color(r, g, b));
      rect(x, y, rectSize, rectSize);
    }
    else {
      fill(192);
      rect(x, y, rectSize, rectSize);
    }
  }
}

if (mclick) {
  x = mx/rectSize;
  y = my/rectSize;

  print(x + " " + y);

  mclick = false;
}

void oscEvent(OscMessage theOscMessage) {
  if (theOscMessage.checkAddrPattern("/monoid/step")==true) {
    /* check if the typetag is the right one. */
    if (theOscMessage.checkTypetag("i")) {
      /* parse theOscMessage and extract the values from the osc message arguments. */
      int s = theOscMessage.get(0).intValue();
      //print("### received an osc message /monoid/step with typetag i.");
      //println(" values: " + s);

      if (s == 16) {
        for (int i = 0; i < 8; i++) {
          for (int j = 0; j < 8; j++) {
            stepNote[0][i][j] = 0;
            stepNote[1][i][j] = 0;
          }
        }
      }
      else {
        step = s;
      }
    }
    else {
      println("### received an osc message. with address pattern "+theOscMessage.addrPattern());
    }
  }
}

```

(Continua)

otto passi del sequencer. Come feedback visivo del tempo che avanza, passo dopo passo tutte le celle attive della colonna relativa

al passo attuale cambiano colore dal verde al blu, per poi ritornare verdi al passo successivo. La variabile globale **step** contiene il

Listato 2 - segue

```
public boolean surfaceTouchEvent(MotionEvent event) {
    if (event.getAction() == 1) {
        mx = (int)(event.getX());
        my = (int)(event.getY());
        mclick = true;

        int ch = 0;

        if (button0) {
            ch = 0;
        }
        else if (button1) {
            ch = 1;
        }

        int i = mx/rectSize;
        int j = my/rectSize;

        if (i < 8 && j < 8) {
            if (stepNote[ch][i][j] > 0) {
                stepNote[ch][i][j] = 0;
            }
            else {
                stepNote[ch][i][j] = 60;
            }

            OscMessage myMessage = new OscMessage("/monoid/grid");

            myMessage.add(i);
            myMessage.add(j);
            myMessage.add(ch);

            oscP5.send(myMessage, remoteAddr);
        }
        else {
            if ((i == 1 || i == 2) && (j == 9 || j == 10)) {
                button0 = true;
                button1 = false;
            }
            else if ((i == 5 || i == 6) && (j == 9 || j == 10)) {
                button0 = false;
                button1 = true;
            }
        }
    }
}

// if you want the variables for motionX/motionY, mouseX/mouseY etc.
// to work properly, you'll need to call super.surfaceTouchEvent().
return super.surfaceTouchEvent(event);
}
```

valore del passo attuale e viene aggiornata tramite un messaggio OSC inviato da Arduino per sincronizzare l'intero sistema.

COMUNICAZIONE TRA ARDUINO E ANDROID TRAMITE OSC

Vediamo innanzitutto come utilizzare l'Ethernet Shield con Arduino: come detto nell'introduzione, per consentire ad Arduino di scambiare informazioni con l'applicazione che gira sullo smartphone Android abbiamo usato l'Ethernet Shield. Questo è costituito principalmente dal chip WIZnet W5100 (Fig. 8) che

implementa sia il MAC e il PHY della connessione Ethernet, che lo stack TCP/IP con i protocolli TCP, UDP, ICMP, IPv4 ARP, IGMP, PPPoE, Ethernet. L'Ethernet shield comunica con Arduino tramite la porta SPI e può essere facilmente utilizzato nelle applicazioni per Arduino grazie alla libreria fornita con l'ambiente di sviluppo e ben documentata in questa pagina <http://arduino.cc/en/Reference/Ethernet>.

Detto ciò, possiamo vedere come si fa ad inviare e ricevere messaggi OSC con Arduino: lo scambio di messaggi tra Arduino e Android

Il protocollo MIDI costituisce un metodo efficiente e standardizzato per veicolare informazioni relative all'esecuzione musicale tramite dati digitali. Le informazioni MIDI vengono trasmesse mediante messaggi che possono essere pensati come istruzioni inviate ad un sintetizzatore musicale per suonare un brano. Ci sono diversi tipi di dispositivi che usano il protocollo MIDI (dispositivi portatili, tastiere musicali, computer...) accomunati dalla possibilità di parlare e capire il "linguaggio" MIDI. Questo linguaggio descrive l'esecuzione della musica in un modo molto simile ad uno spartito: ci sono messaggi MIDI che descrivono quali note devono essere suonate e quanto a lungo; lo stesso per il tempo, quali strumenti devono essere suonati e a che volume. I dati MIDI possono essere trasmessi per via elettronica tra strumenti musicali MIDI-compatibili o memorizzati in file standard MIDI, per essere suonati in seguito. Ci sono tre parti importanti nella specifica del protocollo MIDI: quella relativa al livello fisico, quella riguardante i messaggi e quella relativa ai file standard MIDI. Per la prima, l'interfaccia MIDI è una seriale asincrona a 31,25 kbps con 10 bit trasmessi per ogni byte (un bit di start, 8 bit di dati e un bit di stop). I connettori tipicamente usati sono i DIN a 5 pin e in uno strumento MIDI generalmente si trovano tre di questi connettori, chiamati MIDI-IN, MIDI-OUT e MIDI-THRU. Nelle implementazioni moderne si possono trovare altri tipi di

serve a informare Arduino sui cambiamenti di stato provocati dall'interazione dell'utente con il dispositivo Android e ad inviare a quest'ultimo un messaggio di sincronizzazione; tale scambio avviene tramite il protocollo OSC, che a sua volta viene veicolato dal protocollo UDP.

Nella nostra applicazione, dal lato Arduino, per gestire i messaggi OSC abbiamo usato la libreria Z_OSC. Come è possibile vedere, nel Listato 1 la libreria viene importata con l'istruzione **#include <Z_OSC.h>** (subito dopo aver importato le librerie SPI e Ethernet)

interfacce usate come trasporto per il protocollo MIDI (tipicamente USB, FireWire ed Ethernet). La parte della specifica relativa ai messaggi MIDI è quella più importante. Originariamente pensati per connettere due strumenti musicali tramite cavi e connettori DIN i messaggi MIDI vengono ora utilizzati anche nei computer e negli smartphone per generare musica che può essere suonata dagli stessi dispositivi o trasportati ad altri dispositivi MIDI-compatibili. Ci sono diversi gruppi di messaggi MIDI per diverse applicazioni, qui ne prenderemo in considerazione solo alcuni. Un messaggio MIDI è costituito da un byte di stato generalmente seguito da uno o due byte di dati. I messaggi MIDI sono classificati in "Channel

Messages" e "System Messages". I "Channel Messages" sono quelli che si applicano ad uno specifico canale e hanno il numero di canale incluso nel byte di stato. I "System Messages" non riguardano uno specifico canale e quindi non hanno il numero del canale nel byte di stato. I "Channel Messages" possono essere ulteriormente classificati come "Channel Voice Messages" o "Channel Mode Messages". I "Channel Voice Messages" trasportano dati relativi all'esecuzione musicale, che comprendono la maggior parte del traffico in un tipico flusso di dati MIDI. I "Channel Mode Messages" influenzano il modo in cui uno strumento musicale risponde ai "Channel Voice Messages". Nella tabella qui sotto

vengono riportati, come esempio, i "Channel Voice Messages". L'ultima parte della specifica riguarda gli Standard MIDI File (SMF) che vengono usati per collezionare e memorizzare nel file system di un computer (o di uno smartphone) sequenze di messaggi MIDI insieme alle informazioni relative al tempo. I MIDI file vengono normalmente creati usando programmi per computer che organizzano i messaggi MIDI in una o più tracce, le quali possono essere registrate ed editate indipendentemente. Ogni traccia può essere assegnata ad uno specifico canale MIDI e/o a uno specifico strumento. Tutte le piattaforme di computer normalmente usate possono mandare in esecuzione i file MIDI (*.mid).

Channel Voice Messages [nnnn = 0-15 (MIDI Channel Number 1-16)]		
Status D7---D0	Data Byte(s) D7---D0	Description
1000nnnn	0kkkkkkk 0vvvvvvv	Note Off event. Questo messaggio è inviato quando una nota termina. (kkkkkkk) è il numero del tasto (nota) dello strumento; (vvvvvvv) è la velocità.
1001nnnn	0kkkkkkk 0vvvvvvv	Note On event. Questo messaggio è inviato quando una nota inizia (start). (kkkkkkk) è il numero del tasto (nota); (vvvvvvv) è la velocità.
1010nnnn	0kkkkkkk 0vvvvvvv	Polyphonic Key Pressure (Aftertouch). Questo messaggio è spesso inviato premendo un tasto dopo averlo mandato a fondo. (kkkkkkk) è il numero del tasto (nota) dello strumento; (vvvvvvv) è la velocità.
1011nnnn	0ccccccc 0vvvvvvv	Control Change. Questo messaggio è inviato quando cambia un valore del controller. I controller sono dispositivi quali pedali e leve. I controller numero 120+127 sono riservati come "Channel Mode Messages". (ccccccc) è il numero del controller (0+119). (vvvvvvv) è il valore del controller (0+127).
1100nnnn	0ppppppp	Program Change. Questo messaggio viene inviato quando cambia il numero di patch. (ppppppp) è il numero del programma.
1101nnnn	0vvvvvvv	Channel Pressure (After-touch). Questo messaggio è spesso inviato premendo a fondo un tasto. Questo messaggio è diverso dall'Aftertouch polifonico. Utilizzare questo messaggio per inviare il valore massimo individuale di pressione (di tutti i tasti attualmente premuti). (vvvvvvv) è il valore di pressione.
1110nnnn	0lllllll 0mmmmmmm	Pitch Wheel Change. Ommmmmmm. Questo messaggio è inviato per indicare un cambio nella rotella del pitch; lo stato di quest'ultima è espresso con 14 bit: Center (nessun cambiamento del pitch) è 2000H. La sensibilità dipende dal trasmettitore. (lllllll) sono i 7 bit più significativi e (mmmmmmm) i 7 bit meno significativi.

e viene usata con la definizione e il successivo utilizzo degli oggetti **Z_OSCClient**, **Z_OSCServer** e **Z_OSCMessage**. Il server viene inizializzato nella funzione **setup()** fornendo come parametro la porta UDP sulla quale mettersi in ascolto per ricevere messaggi. In seguito, sempre nella funzione **setup()**, viene inviato un messaggio (indirizzo: `"/monoid/step"`, un solo argomento di tipo intero: `"i"`) ad Android con il metodo **send** dell'oggetto **client**. Questo messaggio viene usato dall'applicazione Arduino per la sincronizzazione con Android, ma

in questo caso (solo nel **setup()**), con valore dell'argomento impostato a 16, assume il significato di reset (azzeramento) dell'array che mantiene lo stato nell'applicazione Android. Le funzioni nello sketch per Arduino (**Listato 1**) che si occupano della ricezione e dell'invio dei messaggi OSC verso Android sono **readOSC()** e **syncStep(int stepPos)**. La prima si occupa di ricevere i messaggi contenenti il passo temporale, la nota e il canale impostati dall'utente sullo smartphone, oltre che di aggiornare di conseguenza l'array **stepNote[c][i][j]**, dove *c* è il canale,

i il passo temporale e *j* la nota. La funzione **syncStep()** invia all'applicazione lato Android messaggi per mantenere sincronizzati gli eventi di invio delle note verso la porta MIDI da parte di Arduino e di aggiornamento della visualizzazione da parte di Android.

COME INVIARE E RICEVERE MESSAGGI OSC CON ANDROID

La gestione dei messaggi OSC nell'applicazione lato Android viene realizzata usando la libreria **oscP5**, che viene importata all'inizio della sketch (vedi **Listato 2**) e inizializzata nella funzione **onRe-**

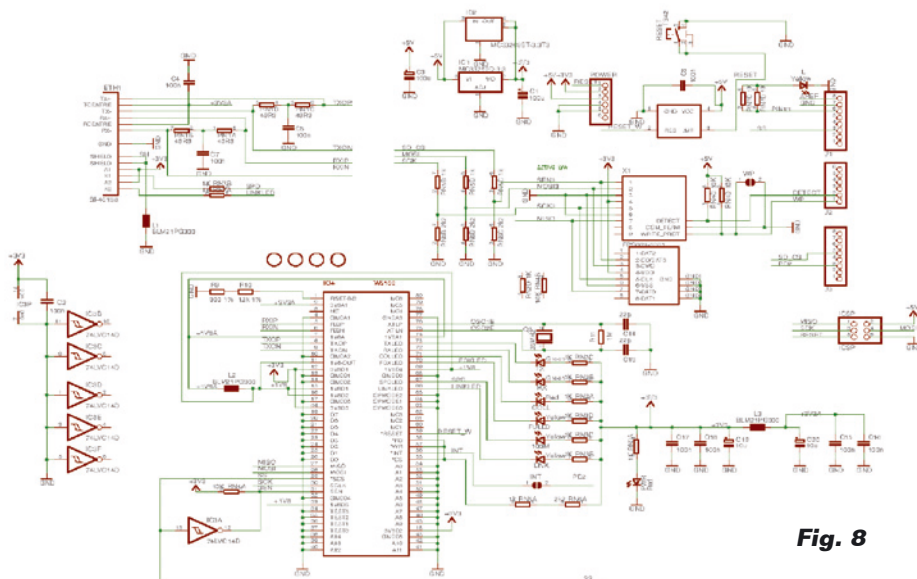


Fig. 8

sume() usando come parametro la porta UDP per la ricezione dei messaggi. Sempre nella funzione **onResume()**, viene inviato verso Arduino il messaggio che normalmente serve a comunicare il cambiamento dello stato del sequencer (indirizzo: `"/monoid/grid"`, tre argomenti di tipo intero) ma solo in questa fase iniziale, con argomenti 16, 16 e 0, provoca un reset (azzeramento) dell'array con lo stato del sequencer (**stepNote[]**).

Per la gestione dei messaggi in arrivo, la libreria **oscP5** implementa un gestore dell'evento di nome **oscEvent(OscMessage theOscMessage)**. Con quest'ultimo viene gestito il messaggio con indirizzo `"/monoid/step"` inviato da Arduino per la sincronizzazione; se l'argomento ricevuto col messaggio è uguale a 16, l'array **stepNote[]** viene azzerato (condizione di reset) altrimenti viene usato per impostare la variabile globale **step** con il passo temporale corretto. Nel gestore dell'evento **surfaceTouchEvent(MotionEvent event)** vengono generati ed inviati verso Arduino, i messaggi con indirizzo `"/monoid/grid"` che

contengono le informazioni sul cambiamento di stato del sequencer (modifica del valore di un elemento dell'array **stepNote[]**) causato dall'interazione dell'utente con un tap su una cella della griglia.

ESEMPIO DI FUNZIONAMENTO

Affinché l'Android Arduino Sequencer funzioni, il sistema nel quale viene installato deve avere alcuni requisiti minimi (un router Wi-Fi, un computer con sistema operativo Windows, Linux o Mac OSX e uno smartphone con Android) ed effettuare alcune impostazioni. Nella Fig. 9 possiamo vedere una foto di come si presenta il sistema composto da Arduino con l'Ethernet Shield e il MIDI Shield a bordo. Arduino va collegato al computer tramite il cavo USB con il duplice scopo di alimentarlo e programmarlo. Dall'Ethernet Shield parte il cavo Ethernet per la connessione verso il router. La porta MIDI OUT del MIDI Shield va collegata con un cavo DIN a 5 poli ad un sintetizzatore audio con una porta MIDI IN o (come nel nostro caso) ad un computer, dotato di interfaccia

MIDI USB, sul quale gira un sintetizzatore software.

Nello sketch relativo ad Arduino (**Listato 1**) il vettore **ip[]** va impostato con un indirizzo compatibile con la configurazione del vostro router (nel nostro caso: **ip[] = {192, 168, 1, 177}**). Sempre nello sketch Arduino, il vettore **remotelp[]** va impostato con l'indirizzo che il router Wi-Fi avrà assegnato in automatico al vostro smartphone Android (nel nostro caso: **remotelp[] = {192, 168, 1, 18}**). Nello sketch relativo a Processing in modalità Android (**Listato 2**) nella funzione **onResume()** usate l'indirizzo IP impostato per Arduino come parametro per la funzione **NetAddress()** (nel nostro caso: **NetAddress("192.168.1.177", 8888)**).

Dopo aver configurato gli indirizzi di rete come descritto sopra, si deve eseguire l'Upload dello sketch nell'Arduino dal relativo IDE e caricare l'applicazione Android nello smartphone, scegliendo la voce "Run on Device" dal menu "Sketch" dell'IDE Processing in modalità Android.

Una volta effettuati tutti i collegamenti e le configurazioni descritte, il sistema è pronto per l'uso. Nello schermo dello smartphone possiamo scegliere i due canali a disposizione e per ognuno di essi impostare la sequenza di note desiderata. I due canali (1 e 2)

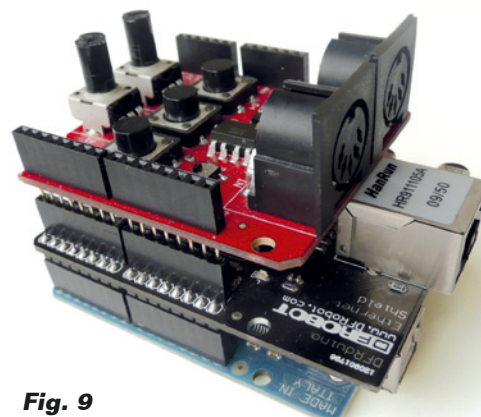


Fig. 9

Open Sound Control (OSC) è un protocollo, sviluppato presso il Center for New Music and Audio Technology (CNMAT) di Berkeley, per la comunicazione tra computer, strumenti musicali elettronici e altri dispositivi multimediali, ma ottimizzato per le moderne tecnologie di rete. L'intento degli sviluppatori era di realizzare un protocollo utilizzabile per la musica al computer interattiva, che potesse sfruttare le esistenti tecnologie di rete ad alta velocità, tipo l'ethernet. L'OSC è un protocollo di rete indipendente dal trasporto, infatti OSC prevede un formato di messaggi binari e i dati nel formato OSC posso-

no essere veicolati da qualsiasi tecnologia di rete. La maggior parte delle implementazioni OSC attuali usano i principali protocolli di Internet (UDP e TCP/IP) con connessioni di rete ethernet o wireless, ma ci sono anche implementazioni per connessioni seriali. Un altro aspetto importante del protocollo OSC è la maggiore flessibilità nel significato dei messaggi rispetto al modello MIDI fatto di note, canali e controller continui. In OSC si usano nomi dei parametri simbolici, invece di parametri numerici. I parametri e i messaggi sono organizzati in uno spazio di indirizzi gerarchico: per esempio "/voices/3/osc/14/freq"

potrebbe essere un nome di un messaggio per impostare la frequenza del quattordicesimo oscillatore nella terza voce di sintesi additiva. Gli argomenti nei messaggi OSC possono essere stringhe, numeri in virgola mobile a 32 bit, numeri interi a 32 bit e "blob" arbitrari di dati binari. Nella tabella qui sotto vengono riportate le principali caratteristiche del protocollo OSC a confronto con quelle del protocollo MIDI, tratte da "Open Sound Control: an enabling technology for musical networking", di Matthew Wright, Center for New Music and Audio Technology (CNMAT), University of California at Berkeley.

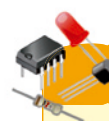
	OSC	MIDI
Esempio di messaggio	/wii/ir/x 0.1503 /stylus/pressure 0.014 /camera/look-at 5. 12. 17. /play-note 15 0.9	144 60 64 (MIDI Note-on) 128 60 64 (MIDI Note-off)
Tipo di messaggi	definito dall'utente Human-readable	predefinito Byte-encoded
Aggiornamento di parametri multipli	✓ via Bundle	-
Time-tagging	✓ via Bundle	-
Trasporto hardware indipendente	✓	-
Numero di canali	illimitato	16
Formati dei dati	Integer, Double Precision Floating Point, Strings	1-byte integers 0+255
Struttura dei messaggi	definita dall'utente	predefinita
Compatibilità con microcontrollori	✓	-
Indipendenza dallo stato della macchina	✓	-
Aree di applicazione	musica, video, robotica ecc.	musica
Accuratezza del clock	picosecond (via NTP / IEEE 1588 Sync)	20.833 microseconds
Data-rate	> 800 Mbps	31.25 bps

vanno associati, nel sintetizzatore software o stand-alone utilizzato, a due strumenti a scelta.

CONCLUSIONI

In queste pagine abbiamo visto come implementare un semplice sequencer MIDI utilizzando un Arduino dotato di MIDI Shield e Ethernet Shield e come controllarlo con uno smartphone Android. La comunicazione tra Arduino e Android avviene mediante lo scambio di messaggi OSC (Open Sound Control) attraverso un router, sfruttando la connessione Ethernet per Arduino e quella WiFi per lo smartphone Android. Un'applicazione sviluppata con l'IDE di Arduino (vedi **Listato 1**)

si occupa della gestione dell'interfaccia MIDI e dello scambio di messaggi OSC via rete, mentre per lo smartphone Android è stata scritta un'applicazione con l'IDE Processing (vedi **Listato 2**) che gestisce l'interfaccia utente e la comunicazione con Arduino. Successivamente alla stesura di questo articolo sono state rilasciate nuove versioni dei vari ambienti di sviluppo utilizzati; l'impiego delle nuove versioni implica una serie di modifiche: nel caso di una versione più recente di Processing in modalità Android è necessario fare uso di Processing 2.0 alpha mentre, nel caso di Arduino 1.0, va cambiato il sorgente dello sketch.



per il MATERIALE

I sorgenti descritti nell'articolo sono scaricabili gratuitamente dal sito della rivista. Lo shield MIDI è disponibile in kit al prezzo di 20,50 Euro (cod. MIDI-SHIELD); lo shield comprende la base con il chip già montato nonché tutti gli altri componenti ad eccezione degli strip. Lo shield Ethernet (cod. SHIELDETHERNET) è disponibile al prezzo di 42,00 Euro. I prezzi si intendono IVA compresa.

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

NUOVE TELECAMERE CON TECNOLOGIA **Effio™**

Qualità e Prestazioni senza precedenti

TELECAMERA IR VARIFOCAL a COLORI con DSP SONY EFFIO-E 700 LINEE TV

- Sensore da 1/3" Exview CCD;
- Sensibilità 0 lux (IR On);
- 976x582 pixel (PAL);
- Ottica varifocal manuale 2,8-11mm;
- 42 LED IR (portata ~40m);
- Alimentazione 12Vdc;
- Dimensioni 200x17x83mm;
- Da esterno (IP66);
- Completa di staffa.

✓ **Alta definizione**
Immagini con risoluzione orizzontale superiore a 600 linee TV.



✓ **Elevato rapporto S/N**
Immagini sempre nitide grazie alla riduzione del disturbo ottenuta tramite speciali filtri digitali.



✓ **Fedele riproduzione dei colori**
Colori sempre brillanti e intensi con qualsiasi tonalità di luce.



cod. CAMCOLBUL27

€ 110,00

TELECAMERA IR a COLORI con DSP SONY EFFIO 600 LINEE TV

- Sensore da 1/3" CCD HR;
- Sensibilità 0,1 lux (F2.0);
- 753x582 pixel (PAL);
- Ottica f=3,6mm / F2.0;
- Angolo di ripresa 92,6°;
- 21 LED IR (portata ~15m);
- Alimentazione 12Vdc;
- Dimensioni 140x56x80mm;
- Da esterno (IP67);
- Completa di staffa.



cod. CAMCOLD19

€ 105,00

TELECAMERA DOME a COLORI con DSP SONY EFFIO 600 LINEE TV

- Sensore da 1/3" CCD HR;
- Sensibilità 0,05 lux;
- 753x582 pixel (PAL);
- Ottica f=3,8mm / F1.5;
- Angolo di ripresa 85°;
- Alimentazione 12Vdc;
- Dimensioni Ø125x85mm.

cod. FR475

€ 190,00



**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.

Un anno di Robotica

Il 2011 è stato sicuramente un anno importante per la Robotica, sia per quanto riguarda la ricerca - portata avanti da numerose Università in tutto il mondo (compresi molti soggetti italiani) - sia per quanto riguarda le applicazioni pratiche e le competizioni e manifestazioni di settore; anche se - come dicevamo il mese scorso - manca ancora quella "killer application" in grado di portare le applicazioni robotiche nel nostro quotidiano. Probabilmente, al contrario di altri settori che hanno avuto un boom improvviso e fortemente pervasivo (come i sistemi di comunicazione personale, dai cellulari agli smartphone), le applicazioni robotiche entreranno nella nostra vita molto più lentamente, un passettino alla volta.



Per quanto riguarda il nostro mondo, ovvero quello della Robotica Educativa, ludica e didattica, a livello più generale si è andata diffondendo sempre di più la convinzione che la filosofia dell'open-source, la collaborazione tra gli utenti e tra i centri di ricerca, sia il modo migliore per riuscire ad ottenere dei risultati significativi, vista la complessità della materia. Da questo punto di vista l'attività del sito DIYDRONES nel campo degli UAV e quella del progetto iCub nel settore dei robot umanoidi sono tra le più interessanti; soprattutto la prima - che abbiamo seguito da vicino quest'anno - ha portato a risultati a dir poco sorprendenti.

Nel settore educativo il lavoro di molti soggetti - dalla Rete per la RoboCup Jr. Italia a Scuola di Robotica a tante altre Associazioni - ha contribuito ad aumentare l'interesse di studenti, insegnanti e persone comuni per questo settore che, come ha recentemente detto il Prof. Giovanni Marcianò della RoboCup Jr. "dovrebbe essere finalizzato ad una corretta cultura d'uso della tecnologia, ad imparare ad utilizzare un oggetto con cui puoi, appunto, grazie alle potenze della tecnologia ed alla semplicità delle interfacce, programmare senza grandi corsi specialistici. Questa cultura d'uso va poi a riverberarsi sul tuo essere cittadino in un mondo in cui, domani, troverai questi oggetti al tuo servizio". Da questo punto di vista, un altro importante player è sceso in campo: Arduino. Un sistema che grazie alla sua forte vocazione robotica ed alla semplificazione introdotta nell'apprendimento della programmazione, ma anche alla sua comunità così vasta, potrebbe rappresentare la vera sorpresa dei prossimi anni.

IN QUESTO NUMERO

108 ■ NEWS ED EVENTI

Si è chiusa la terza edizione di Robotica: ecco una breve panoramica della manifestazione che, alla terza edizione, sembra finalmente aver trovato una precisa identità ed un ruolo di tutto rispetto nell'ambito degli eventi internazionali dedicati alla robotica umanoide e di servizio.

112 ■ REMOTE CONTROL

ROVER CONTROLLATO DA IPHONE O IPAD

Sperimentiamo un interessante sistema di controllo remoto mediante il quale è possibile pilotare a distanza qualsiasi Robot o Rover mediante iPhone o iPad; su questi dispositivi viene anche visualizzato quanto ripreso dalla telecamera montata sul mezzo remoto.



Tutte le novità di **ROBOTICA** 2011

di Davide Scullino

Ecco cosa ci ha proposto la manifestazione dedicata alla robotica umanoide e di servizio.



Buona la terza: quest'anno, alla terza edizione, appunto, una manifestazione di robotica sembra finalmente aver preso piede nel nostro Paese, creando le premesse, se gli organizzatori proseguiranno sulla strada intrapresa, per il consolidamento e la crescita di un evento nazionale che rappresenti un significativo punto di incontro e di confronto tra il mondo della ricerca, le start-up italiane, i pochi produttori nazionali, le altrettanto scarse realtà internazionali ed un pubblico di addetti ai lavori e di utilizzatori, ma anche di curiosi, studenti e, più in generale, di appassionati di tecnologia. È l'impressione che abbiamo avuto quest'anno visitando gli spazi destinati a Robotica 2011, l'unica rassegna italiana dedicata ai robot umanoidi e di ser-

vizio, conclusasi lo scorso 19 novembre. Organizzata all'interno di Hitech Expo, l'evento di Fieramilano-Rho dedicato all'Hi-Tech (meccatronica, automazione industriale, nuove frontiere dell'illuminazione) ed alle energie rinnovabili, Robotica 2011 sembra finalmente aver fatto il passo giusto verso un futuro in cui la robotica al servizio dell'uomo dovrebbe uscire dalla sua fase embrionale per proporre al pubblico le prime, grandi novità concretamente recepibili dal mercato, a dimostrare che tra non molto i robot entreranno nella nostra vita quotidiana e ci aiuteranno nei compiti più complessi, ma anche nei più semplici e scontati, laddove non riuscissimo più a svolgerli.

Oltre a vantare nel proprio parco espositori alcuni tra i più grandi player in-



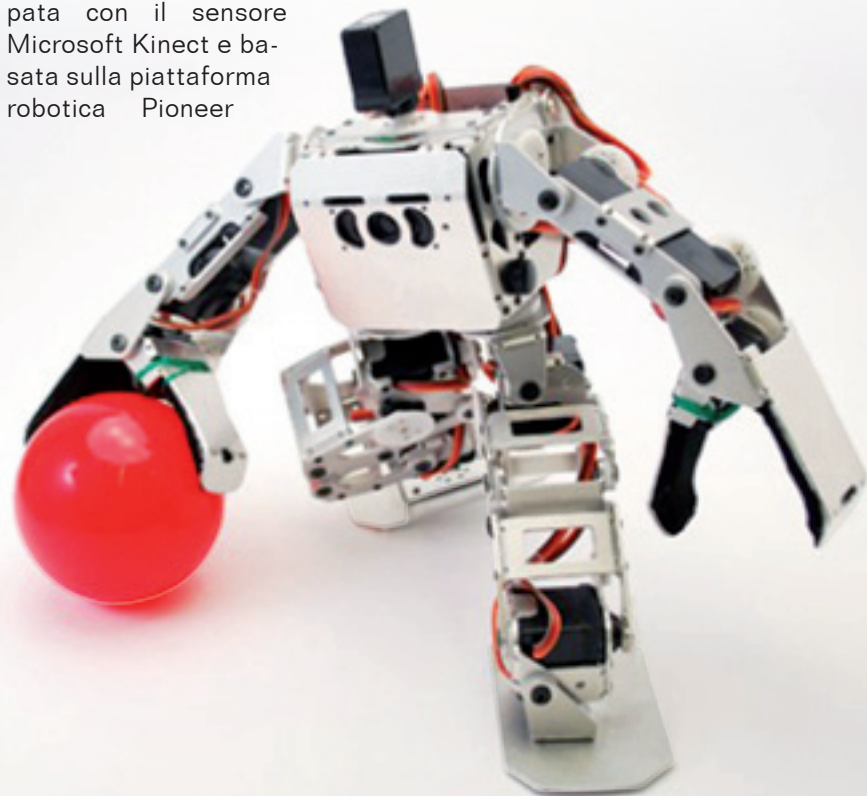
ternazionali del settore (da Kuka a IT+Robotics, da Comau a STMicroelectronics, da LEGO Education ad Aldebaran), Robotica 2011 è stata anche il palcoscenico privilegiato per aziende come Nuzoo Robotics, che ha presentato in anteprima mondiale il suo I-Do Robot, il primo service robot per la telepresenza (si chiamano così gli automi che permettono di controllare a distanza un luogo, spostandosi al suo interno e trasmettendo le immagini riprese da una telecamera, i suoni captati da microfoni, oltre a tanti altri parametri rilevati da sensori) interamente progettato e realizzato in Italia. Questo automa ha attratto particolarmente il pubblico, che ha ammiccato all'idea di mettersi in casa un robot che permette di telefonare, guardare TV e video in alta definizione, ascoltare musica, sorvegliare a distanza la propria casa, l'ufficio o il magazzino, oppure fare una visita virtuale agli anziani genitori quando non è possibile andarci di persona; capace di muoversi autonomamente evitando persone e ostacoli, I-Do Robot è anche in grado di ricaricarsi da solo raggiungendo autonomamente la propria base.

Grande spazio hanno trovato i maggiori centri di ricerca universitaria italiani; tra questi, l'AILab (il Laboratorio di Intelligenza Artificiale e Robotica del Politecnico di Milano), il Dipartimento di Ingegneria Chimica, Gestionale, Informatica e Meccanica dell'Università di Palermo, lo IAS-Lab (Dipartimento di Ingegneria dell'Informazione dell'Università di Padova) ed il Dipartimento di Ingegneria dell'Università degli studi di Bergamo, che insieme ad ST ha realizzato interessanti sperimentazioni.

La squadra dell'AILab ha presentato l'ultima evoluzione della sua carrozzina robotica ad autonomia condivisa Roby WheelChair, oltre ad un robot equilibrista per telepresenza e riconoscimento e ad un "robogame" di ultima generazione chiamato Jedi Trainer, consistente in un quadricottero robotico in grado di volare autonomamente sparando ad un giocatore -aspirante guerriero Jedi- che deve



difendersi usando una spada laser. Il Laboratorio IAS-Lab ha presentato Robovie-X, robot umanoide per le scuole derivato dal cinque volte campione del mondo di calcio tra robot prodotto dalla giapponese VStone, e Robovie-Nano, fratello minore del modello X (in pratica una versione semplificata dotata di soli 13 servomotori). Inoltre ha proposto Demo People Tracking, un'applicazione di ricerca persone sviluppata con il sensore Microsoft Kinect e basata sulla piattaforma robotica Pioneer



AT-X. Camminando per Robotica 2011, si è respirata un'aria nuova, sia per le novità proposte, sia per l'attenzione che il pubblico ha riservato alle realizzazioni, dalle più tecniche a quelle più coreografiche e più facili da metabolizzare, come, ad esempio, i robot musicisti (capaci di suonare le note che un computer gli passa via interfaccia MIDI) che si sono esibiti per il piacere e l'ammirazione di tutti i visitatori; l'automa pianista proposto da Teotronica seduto di fronte ad una tastiera elettronica che ha suonato grazie alle sue 19 dita azionate da attuatori elettromagnetici, per quattro giornate ha fatto sentire i visitatori come in un piano bar. Vestito da artista, come l'abbiamo trovato nel suo stand, questo robot certo non sfuggirebbe in feste ed eventi di vario genere, dove attrarrebbe l'attenzione sicuramente più di un pianista in carne ed ossa, anche se suonasse motivi un po' demodé. Chissà se, come il celebre Humprey Bogart in Casablanca, avremmo potuto domandargli: "play it again, Sam"... ehm... "Teo"...

Altro robot musicista che ha polarizzato l'attenzione del pubblico è stato il

batterista: accompagnato da un animatore, ha eseguito numerosi "pezzi" intrattenendo il pubblico, che lo riprendeva con l'iPhone, per poi pubblicarne il video su YouTube. Alla manifestazione si sono fatti notare anche i piccoli NAO, da anni protagonisti della Robocup (i campionati mondiali di calcio riservati ai robot), che si sono esibiti a più riprese ballando sulle note del Bolero di Ravel, grazie alla sofisticata tecnologia implementata da Aldebaran Robotics.

A Robotica 2011 non poteva, naturalmente, mancare Arduino, protagonista dell'elettronica open-source e indubbiamente il progetto per eccellenza tra quelli che hanno avvicinato l'elettronica e l'automazione al pubblico "digiuno" di queste materie.

Nella giornata di venerdì 18 novembre, organizzato da Scuola di Robotica, si è tenuto l'Arduino Day, un evento nell'evento, con la partecipazione di Massimo Banzi che ha presentato le ultime novità del team di Ivrea. Da parte nostra abbiamo presentato il prototipo del nostro UAV realizzato con board basate sulla piattaforma Arduino, progetto della cui evoluzione ci siamo occupati sulle pagine della rivista nel corso



di quest'anno. Presso il nostro stand il pubblico ha potuto vedere in funzione e toccare con mano alcuni degli altri progetti robotici e di automazione proposti negli ultimi mesi, dalla stampante 3D, alla CNC, al quadricottero.

Grande successo ha riscosso anche la nuova Area Inventors, che ha inteso offrire agli inventori l'opportunità di presentare agli operatori del settore le realizzazioni frutto del loro ingegno. Interessante anche la sessione sui sistemi robotici per l'utenza debole (disabili e ammalati) organizzata dal Siri (Associazione Italiana di Robotica e Automazione), all'interno della quale si è svolta la consegna del Premio Antonio D'Auria 2010, riservato a progetti e prototipi di dispositivi meccatronici innovativi di ausilio ai disabili motori, assegnato ad Andrea Bonarini del Politecnico di Milano e ad Andrea Calanca dell'Università di Verona. Alla rassegna non è mancata la Scuola Superiore Sant'Anna di Pisa, che ha presentato due progetti: Robot Companions for Citizens, finalizzato all'integrazione dei robot nella nostra società in modo da migliorare la qualità della vita delle persone, e HydroNet, sostenuto dalla Commissione Europea, che utilizza robot marini sensorizzati per il monitoraggio dell'ambiente.

Robotica 2011 non è stata solo un'esibizione interattiva, ma ha visto avvicinarsi convegni tematici; nella sua ultima giornata ha fatto il punto sulla robotica educativa nella scuola italiana con un evento che ha visto la partecipazione di numerosi insegnanti e ricercatori tra i quali numerosi esponenti della Rete di scuole per la RoboCup Junior Italia, manifestazione nata per promuovere esperienze di robotica educativa nella scuola dell'obbligo.

I convegni hanno visto la partecipazione, tra gli altri, del professor Yoshiyuki Sankai del Cybernics Laboratory (una delle massime autorità nel campo della robotica) intervenuto nel corso del convegno sulla robotica riabilitativa per presentare il suo esoscheletro HAL, ossia un robot che ridà la capacità di camminare alle persone che non pos-



sono più farlo. HAL (Hybrid Assistive Limb) è una sorta di armatura che le persone con difficoltà motorie possono indossare assicurandola agli arti ed al tronco mediante fasce di tela: rilevando con appositi sensori lo sforzo compiuto dai muscoli, fa compiere i movimenti. In Giappone vengono già usati 250 esemplari in 110 tra ospedali e centri di cura e in Europa arriveranno a gennaio, in Svezia e Austria. Molto interessante e con grande partecipazione di pubblico è stato anche l'intervento a distanza di Paolo Dario (Scuola Superiore Sant'Anna di Pisa e responsabile dell'Istituto di Biorobotica) che ha presenziato in teleconferenza ad un affollatissimo seminario in cui ha illustrato il progetto Robot Companions for Citizens. □



COSTRUIAMO UN ROVER CONTROLLATO DA REMOTO MEDIANTE iPad

di MARCO MAGAGNIN

Sperimentiamo un interessante sistema di controllo remoto mediante il quale è possibile pilotare a distanza qualsiasi Robot o Rover mediante iPhone o iPad; su questi dispositivi viene anche visualizzato quanto ripreso dalla telecamera montata sul mezzo remoto.



Durante lo sviluppo di quello che abbiamo definito come “sistema UAV”, più di un lettore ci ha scritto per chiederci di presentare un veicolo terrestre in grado di essere pilotato in una determinata area inviando immagini in diretta dell'ambiente circostante.

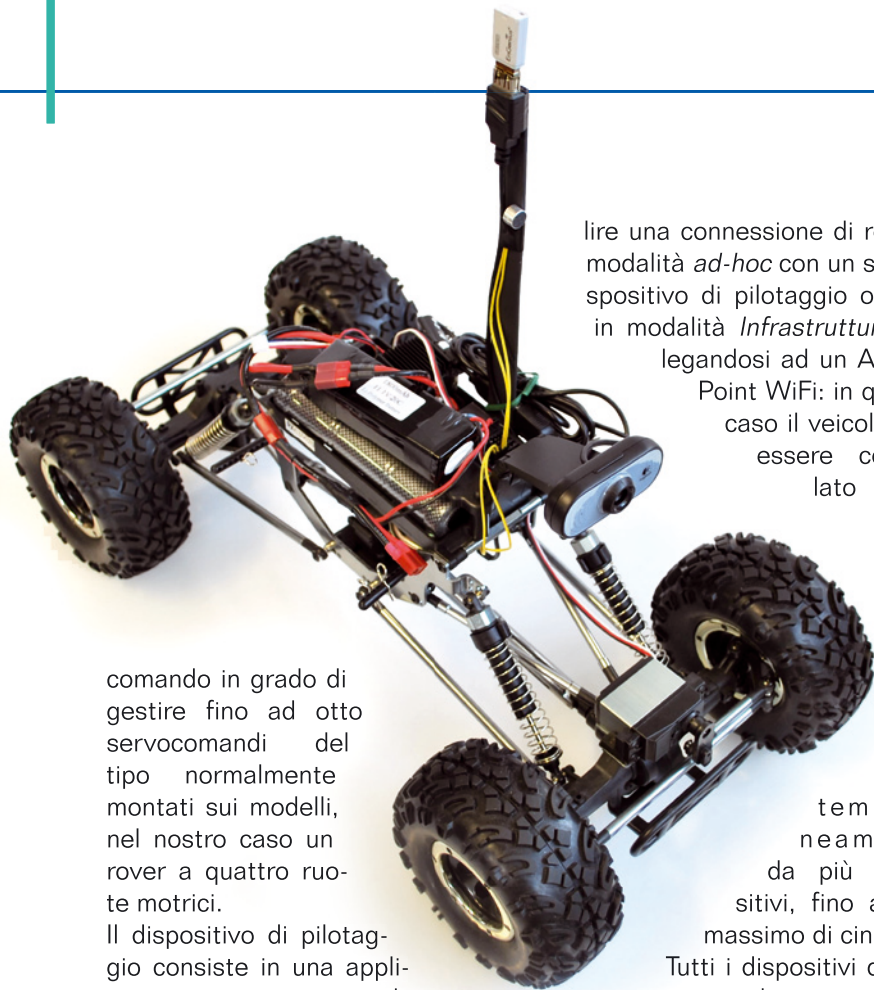
Qualcuno si è spinto oltre richiedendo anche la presenza a bordo di sensori di rilevamento nonché la possibilità di attivare ulteriori dispositivi di bordo nel caso di particolari situazioni.

Ovviamente la comunicazione deve avvenire collegandosi alla rete dati in modalità wireless ed il sistema di controllo deve risiedere su un dispositivo facilmente trasportabile, come un telefono o un tablet. Incuriositi e stimolati da queste richieste e nel tentativo di trovare una soluzione a

queste esigenze, abbiamo cercato sul mercato uno o più dispositivi che risolvessero nel modo più semplice questo tipo di problematiche, ovvero un sistema elettronico che fosse facilmente integrabile in un rover o un robot commerciale per realizzare le funzionalità richieste.

Il sistema che presentiamo si chiama WIRC ed è un radiocomando su rete WiFi prodotto da Dension, una società specializzata nello sviluppo di sistemi multimediali di qualità, destinati alla installazione a bordo di mezzi mobili, auto, autobus, treni e navi. Il sistema in questione si comporta come una ricevente per radio-





comando in grado di gestire fino ad otto servocomandi del tipo normalmente montati sui modelli, nel nostro caso un rover a quattro ruote motrici.

Il dispositivo di pilotaggio consiste in una applicazione interattiva sviluppata per Ipad, Ipad e iPhone. È prevista per il futuro la disponibilità dell'applicazione anche in ambiente Android.

Una volta connessa al sistema di guida l'applicazione presenta sullo schermo l'immagine ripresa dalla telecamera mentre la grafica in sovrapposizione rappresenta i due joystick e gli altri comandi in modalità touch screen.

Vediamo, dunque, quali sono i componenti che fanno parte del set e come vanno utilizzati.

La **Figura 2** rappresenta l'insieme WIRC con un modulo centrale dotato di due porte USB alle quali sono collegati un dongle WiFi per la connessione di rete ed una Webcam, sempre USB.

Sul corpo del modulo centrale sono presenti i connettori per il collegamento di un massimo di otto servocomandi, delle alimentazioni per le sezioni WiFi ed RC nonché i pin per il collegamento di quattro entrate e quattro uscite digitali.

La sezione WiFi permette di stabi-

lire una connessione di rete in modalità *ad-hoc* con un solo dispositivo di pilotaggio oppure in modalità *Infrastruttura* collegandosi ad un Access Point WiFi: in questo caso il veicolo può essere controllato con-

tempora-
neamente
da più dispo-
sitivi, fino ad un
massimo di cinque.

Tutti i dispositivi di pilotaggio vedono contemporaneamente le immagini e lo stato dei sensori ma solo uno alla volta può pilotare il veicolo.

Sulla connessione di rete passa il canale video che permette di ricevere le immagini riprese dalla telecamera sui dispositivi di pilotaggio ed il protocollo di comunicazione bidirezionale che permette l'invio dei comandi dai dispositivi di pilotaggio verso i servocomandi e gli attuatori digitali e, nella direzione opposta, dai sensori verso i dispositivi di pilotaggio.

Sempre per mezzo della connesio-

ne WiFi si eseguono le impostazioni e configurazioni del sistema sia per quanto riguarda la connessione di rete che della parte di comando. Per provare il WIRC abbiamo deciso di installarlo su un modello di mezzo terrestre del tipo ogni-terreno. Abbiamo scelto un rover radiocomandato in scala 1:10, semiriproduzione di una Jeep Wrangler, per le sue caratteristiche di robustezza, di costruzione meccanica ... personalizzabile, di spazio a disposizione per l'installazione di dispositivi aggiuntivi e, non ultimo, di costo abbastanza contenuto. Le caratteristiche del modello sono:

- Scala: 1/10, che significa una lunghezza di 47 cm per una larghezza di 25, per un peso totale di 2,0 - 2,5 Kg;
- Telaio e giunti di trasmissione in metallo assemblato con viti e con ancoraggi aggiuntivi facilmente utilizzabili per il montaggio di dispositivi supplementari;
- Trazione a quattro ruote motrici con assali rigidi indipendenti e sospensioni a corsa lunga che permettono movimenti di torsione accentuati per il superamento di ostacoli anche importanti;

Fig. 1



- Possibilità di aggiungere un servo con braccio per ottenere il funzionamento a quattro ruote sterzanti;
- Quattro ruote da sterrato del diametro di 126mm e larghe 56mm;
- Motore RC-540 con riduttore a ingranaggi 1:48;
- Regolatore di tensione con BEC e retromarcia;
- Batteria ricaricabile Ni-MH 7.2V - 1800mAh;
- Carrozzeria semiriproduzione già pretagliata e verniciata.

Nella confezione è anche incluso un radiocomando a 2.4 GHz ed il caricabatteria per la batteria Ni-MH.

Per la nostra applicazione la trasmettente e la ricevente RC sono superflui in quanto vengono sostituiti dal sistema WIRC mentre dovremo procurarci una ulteriore batteria 3 celle LiPo da 11,1 V e relativo caricabatteria per la alimentazione del WIRC stesso (Fig.1).

La confezione del WIRC contiene:

- Il modulo centrale WIRC;
- Una WebCam USB;
- Un dongle WiFi USB;
- Un cavo a T per l'alimentazione.

Il modulo centrale del WIRC racchiude l'elettronica di funzionamento basata su processore ARM9 266 con 16 MB di memoria SDRAM e 4 MB di memoria FLASH. Il firmware preinstallato è aggiornabile per mezzo della stessa App di controllo. Nel contenitore del modulo sono ricavate le aperture per la esposizione dei connettori di collegamento dei vari dispositivi esterni che, come visibile nello schema elettrico (Fig.2) sono i seguenti:

- Un connettore per l'alimentazione esterna;

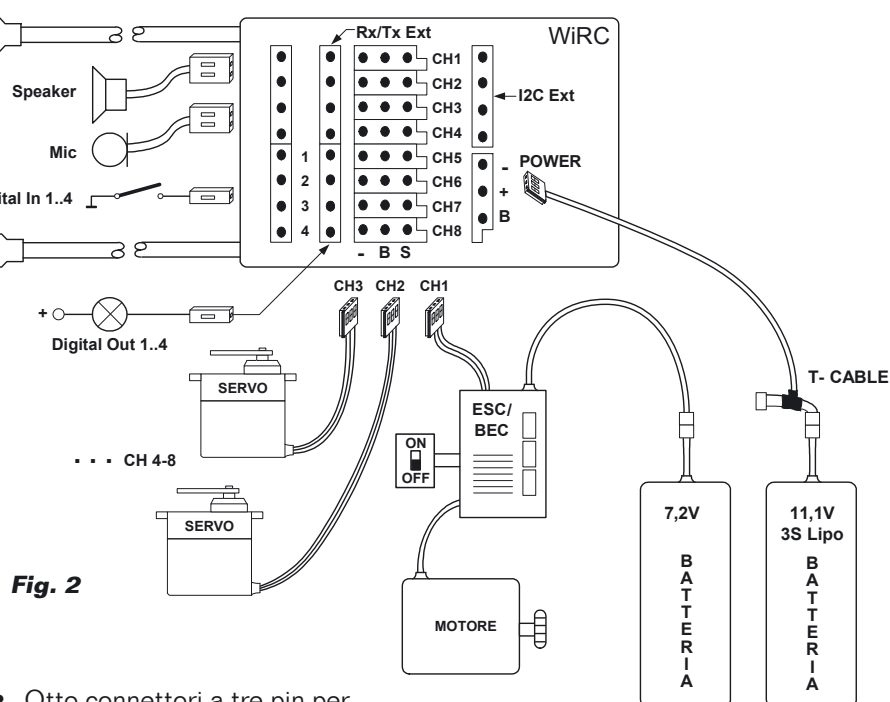


Fig. 2

- Otto connettori a tre pin per altrettanti servocomandi RC di tipo classico;
- Due cavi con uscite USB ai quali si collegano la Webcam e il dongle WiFi;
- Un connettore per il collegamento di un altoparlante ed un connettore per il collegamento di un microfono;
- Quattro pin per collegare altrettanti ingressi di tipo digitale, per esempio, provenienti da sensori di presenza all'infrarosso, sensori di fumo o di luce ecc.;
- Quattro pin per altrettante uscite digitali a cui collegare utenze come sirene, luci, eccetera.

Per installare i componenti del sistema WIRC sul nostro modello abbiamo operato come segue.

Come prima operazione è necessario rimuovere la carrozzeria sfilando le quattro spinette di blocco sui supporti della carrozzeria stessa e liberandola dai supporti.

Se installata, staccate la batteria dal regolatore di giri e, per precauzione, scollegate dal motore il cavo positivo di alimentazione proveniente dal regolatore di giri; questo per evitare messe in moto indesiderate in fase di prima confi-

gurazione del sistema.

Staccate i cavi di regolatore e servo dalla ricevente RC e toglietela dal suo supporto, rimuovendo delicatamente il nastro biadesivo di fissaggio (Fig. 3).

Posizionate il modulo WIRC al posto della ricevente RC fissandola con del nastro biadesivo.

Per ottenere un corretto funzionamento del WIRC sul nostro veicolo è necessario predisporre una doppia alimentazione in quanto il modulo WIRC funziona bene con una tensione vicina ai 12 volt mentre il motore del rover richiede una batteria da 7,2 volt.

Per questo abbiamo utilizzato una batteria LiPo a tre celle (11,1 volt),

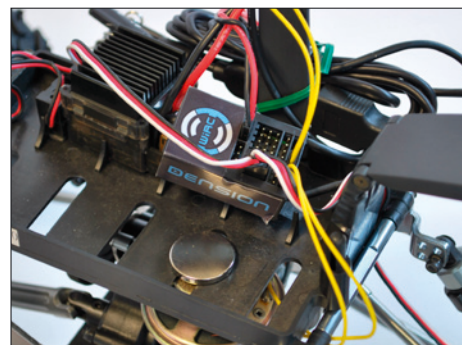


Fig. 3



Fig. 4



Fig. 5

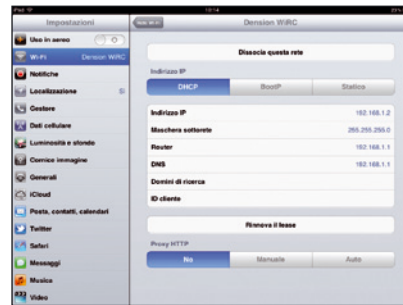


Fig. 6

collegata al connettore a T incluso nella confezione, a sua volta collegato al connettore Power del modulo WIRC.

Il secondo ramo del connettore a T va lasciato libero.

La batteria originale da 7,2 volt verrà collegata a suo tempo direttamente al cavo di alimentazione del regolatore di giri.

Sempre riguardo all'alimentazione, il modulo WIRC non dispone di un interruttore di accensione, per attivarlo è necessario collegare entrambe le alimentazioni, sia quella a 11,1 volt del connettore Power che quella di alimentazione dei servo proveniente dal regolatore BEC.

Verificate che l'interruttore sul regolatore di tensione sia in posizione OFF.

Collegate il cavo segnale proveniente dal regolatore di giri al Canale 1 del modulo WIRC facendo attenzione alla polarità.

Collegate il cavo segnale proveniente dal servo dello sterzo al Canale 2 del modulo WIRC.

Inserite la spina USB della WebCam in una delle prolunghe USB del modulo WIRC ed inserite il dongle WiFi nella seconda prolunga.

Come anticipato il sistema WIRC si gestisce utilizzando una applicazione interattiva sviluppata per Ipad, Ipad e iPhone ed in preparazione per dispositivi Android.

L'applicazione si scarica gratuitamente all'indirizzo:

<http://itunes.apple.com/us/app/wifi-rc/id412503866>.

Una volta scaricata e installata l'applicazione è necessario confi-

gurarla inizialmente per l'utilizzo con il modulo WIRC.

La prima operazione è di accendere il modulo WIRC collegando le due batterie ai rispettivi connettori. Verificate ancora di avere scollegato il cavo positivo di alimentazione del motore per evitare partenze ... indesiderate.

Alla prima accensione il modulo WIRC si configura come un Access Point WiFi con un suo SSID. Nelle operazioni successive è possibile collegarsi con il tablet Apple direttamente a questo Access Point oppure configurare il modulo WIRC come nodo di una nostra rete WiFi e collegarsi ad esso dal tablet collegato alla nostra stessa rete.

Per un primo utilizzo lasciamo il modulo WIRC configurato come Access Point.

A questo punto, sul tablet Apple, andando nelle funzionalità di visualizzazione delle connessioni WiFi disponibili dovremmo trovare anche quella espressa dal modulo WIRC; ovviamente all'inizio la connessione è non protetta quindi possiamo connetterci semplicemente scegliendo l'AP di WIRC dalla lista delle connessioni disponibili (**Fig. 4, 5, 6**).

Fatto questo, apriamo l'applicazione WIRC che presenterà la videata visibile in **Figura 7**.

Tochiamo il pulsante "Connect": l'applicazione apre la videata di controllo composta dalla immagine ripresa dalla telecamera con sovrapposti i controlli di tipo touch screen per la guida e la configurazione del modulo (**Fig.8**).

Ecco una breve descrizione dei controlli disponibili:

- In alto a sinistra il pulsante Settings, che permette di configurare tutte le opzioni del modulo WIRC e che analizzeremo approfonditamente in seguito;
- In alto a destra una serie di pulsanti, il primo dei quali, partendo da sinistra, è il pulsante di Start, che permette di abilitare e disabilitare il controllo del rover dall'applicazione. A questo riguardo va sottolineato il fatto che il modulo WIRC può essere controllato da diversi tablet, fino ad un massimo di cinque. Chiaramente tutti vedono contemporaneamente le immagini riprese dalla telecamera, ma solo uno alla volta può impartire comandi. Vedremo nella descrizione delle funzioni "Settings" come gestire le priorità sul controllo del modulo e di conseguenza del rover;
- Il secondo pulsante da sinistra è il pulsante di controllo della WebCam che permette di passare da una WebCam all'altra nel caso ne siano installate più di una. Il numero della WebCam collegata è il numerino a destra a fianco del pulsante inserito in un esagono azzurro;
- Ancora a sinistra, "Mode" permette di selezionare i diversi modi di pilotaggio che possono essere configurati con modalità diverse; per esempio nel modo 1 si può decidere di pilotare il rover utilizzando i joystick virtuali mentre nel modo 2 si può decidere di pilotare il rover muovendo

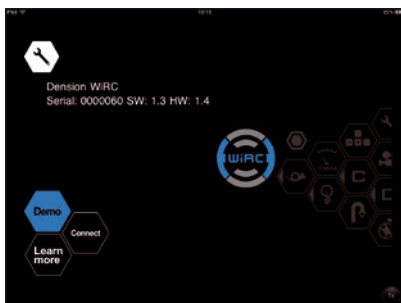


Fig. 7



Fig. 8



Fig. 9

do semplicemente il tablet ed utilizzando i sensori interni allo stesso come input per lo sterzo ed il comando dell'acceleratore. Il modo impostato è indicato dal numero nel piccolo esagono azzurro al di sotto del pulsante "Mode" stesso;

- Il pulsante "Trim" permette di alternare tra Trim e Dual Rate la regolazione associata ai due assi, verticale ed orizzontale, attorno al pulsante esagonale che rappresenta il joystick. La regolazione del trim permette di regolare la posizione dei servo e dell'acceleratore rispetto al centro del joystick. Nel caso dello sterzo, effettuate la regolazione fino ad avere le ruote centrate rispetto al centro del joystick. Per il comando dell'acceleratore la posizione di centro corrisponde al motore fermo; joystick avanti innesca la marcia avanti del rover mentre joystick indietro corrisponde al moto all'indietro del rover. Per regolare lo stick dell'acceleratore posizionate il rover su un supporto che lo sollevi dal suolo in modo che nessun asse in movimento possa toccare le parti che lo sostengono. Togliete le mani e ricollegate il positivo di alimentazione del motore. Poi potete regolare la posizione del Trim fino a trovare il punto che corrisponde a motore fermo. Fate molta attenzione perché la coppia erogata dal motore demoltiplicato è molto forte e non si ferma con le... dita. Quindi attenzione! La regolazione del

Dual Rate permette di regolare la sensibilità dei comandi impartiti rispetto alla posizione del joystick, in modo da ottenere una modalità di pilotaggio più o meno reattiva.

- Il pulsante "Video Rec" permette di attivare e disattivare la registrazione su file delle immagini riprese dalla telecamera;
- I due pulsanti in basso a sinistra permettono di richiamare delle manovre precedentemente memorizzate in fase di settings;
- Ciascuno dei quattro pulsanti in basso a destra permette di verificare lo stato ON/OFF del corrispondente ingresso digitale (triangolino rosso o verde nell'angolo sinistro di ciascun esagono) e di attivare e disattivare il corrispondente output digitale (pulsante che varia tra bianco e rosso, Fig.9).

Torniamo ora alle funzionalità di configurazione.

Nel sottomenu "Basic Settings" possiamo cambiare il nome

della trasmittente, che normalmente coincide con il nome del tablet ma, soprattutto, possiamo cambiare la "Transmitter Priority" nel caso si voglia pilotare il rover da diversi tablet; il livello di priorità permette di stabilire una gerarchia tra i diversi dispositivi: un valore di priorità alto permette di mantenere il controllo del rover anche se altri device con valore di priorità inferiore tentano di acquisirne il controllo. Un device con priorità alta può sempre decidere di cedere volontariamente il controllo ad uno con priorità inferiore. Questa funzionalità è molto utile, ad esempio,





Fig. 10

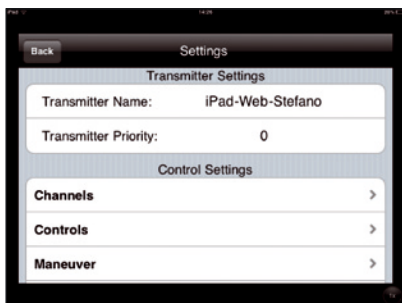


Fig. 11

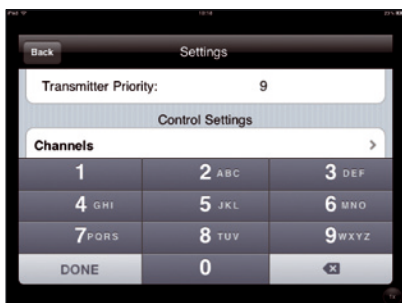


Fig. 12

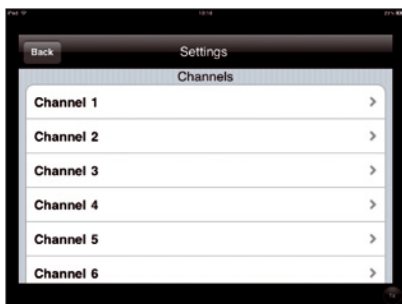


Fig. 13

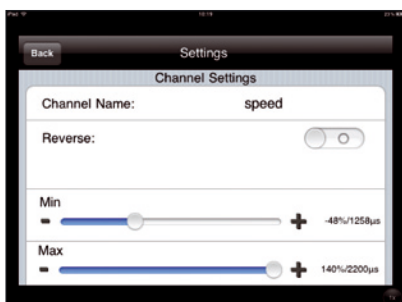


Fig. 14

durante le manifestazioni, per evitare che altri prendano accidentalmente il controllo del nostro mezzo. Per prendere il controllo del rover basta premere il pulsante "Start", se si è l'unico dispositivo di controllo o chi controlla attualmente il rover ha una priorità inferiore; per passare il controllo ad un altro tablet è sufficiente strisciare il centro dello schermo; appare la lista dei tablet collegati dove è possibile scegliere a chi passare il comando (Fig.10, 11, 12).

Il canale video disponibile per tutti i tablet impegna la banda della stessa connessione wireless e quindi più è alto il numero dei dispositivi collegati, più lenta potrebbe essere la risposta del modulo WIRC ai comandi di pilotaggio (Fig.13, 14, 15, 16, 17, 18, 19, 20).

Conviene spendere due parole in più sulla configurazione dei canali, che rispecchia molto la logica delle trasmissioni per radiocomando ma con la possibilità in più di impostare diverse modalità di pilotaggio configurando opportunamente i diversi "Modi" disponibili. Come abbiamo anticipato, possiamo decidere in Modo 1 di pilotare il rover con i joystick, in Modo 2 di utilizzare lo spostamento del tablet e in Modo 3 di configurare un modo misto eventualmente utilizzando dei pulsanti. Partiamo dalla configurazione dei canali e quindi dal sottomenu "Channel Settings" dove troviamo, per ciascun canale, i seguenti parametri:

- Numero del canale - non modificabile;
- Nome del canale: può essere impostato a piacere, come "Sterzo" o "Acceleratore";
- Reverse: cambia la direzione di funzionamento del servo ad es. destra/sinistra per lo sterzo;
- Test Settings: permette di aggiustare la corsa ed il centro del servo, tenendolo premuto si configura la posizione del Trim, mentre in posizione rilasciata permette di aggiustare la corsa (Min/Max);
- Min/Max: Permette di aggiustare le estremità della corsa dei servo, ad esempio per regolare il raggio massimo di sterzata. Una nota particolare merita il canale dell'acceleratore dove i valori di minimo e massimo dell'escursione si riferiscono alla massima velocità in marcia avanti e alla massima velocità in retromarcia; è ovviamente possibile regolare opportunamente i parametri in modo da ottenere, per esempio, una velocità massima in retromarcia facilmente gestibile;
- Trimming: permette di aggiustare la posizione centrale del servo, per esempio marcia dritta per lo sterzo e posizione di motore fermo per l'acceleratore;
- Failsafe: questa configurazione è particolarmente importante perché permette di impostare i comportamenti che il modulo WIRC deve tenere in caso di perdita dei segnali di controllo dalla



Fig. 15

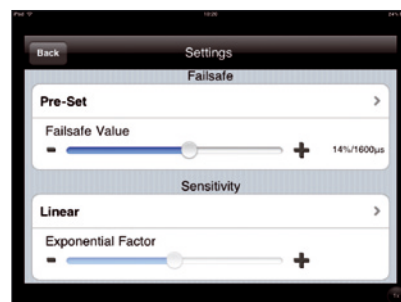


Fig. 16

trasmittente (tablet). Le opzioni disponibili sono: "Disable Output" che lascia il controllo libero, "Last good" che mantiene il comportamento impostato coerentemente con l'ultimo segnale valido ricevuto oppure "Pre Set" che permette di impostare una posizione predefinita da assumere in caso di perdita segnale; per il comando dell'acceleratore vi raccomandiamo di impostare "Pre Set" con un valore che corrisponda a motore fermo. Il valore di "Pre Set" si configura utilizzando lo strumento a slitta corrispondente;

- Sensitivity: permette di impostare la sensibilità associata al canale, ovvero con quale reattività il comando viene trasmesso al rover. La sensibilità può essere lineare con il comando imparito oppure seguire una curva esponenziale; valori negativi del Fattore Esponenziale significano che la reattività ai comandi è maggiore vicino al centro del comando e minore alle estremità mentre viceversa a valori positivi del Fattore Esponenziale corrisponde una reattività minore al centro del comando e maggiore alle estremità;
- Sensitivity mix / Value mix: le miscele permettono di configurare le dipendenze tra le uscite dei canali. Per esempio se si desidera variare la reattività dello sterzo all'aumentare della velocità si può configurare una

miscele con una proporzionalità esponenziale negativa tra il canale del regolatore ed il canale dello sterzo in modo da ridurre proporzionalmente l'intervento sullo sterzo all'aumentare del comando dell'acceleratore;

- Value: abilita le miscele tra i canali;
- Repeat: imposta la frequenza di ripetizione del segnale PPM verso i servo. Può essere impostato per le coppie di canali 1-2, 3-4 e 5-6. I canali 7-8 hanno una impostazione fissa a 50 Hz.

Output digitali.

- Invert output: inverte il significato del segnale digitale ad esempio un impulso di ON presenta il valore 0 invece di 1;
- State On Range: permette di impostare l'intervallo di valori del segnale di output all'interno del quale viene emesso il valore ON mentre all'esterno dello stesso viene emesso il valore OFF.

Impostazione dei controlli

Con le funzioni di questo sottomenu è possibile assegnare un elemento di controllo a ciascun canale o uscita digitale (joystick, giroscopio o pulsante, Fig. 21, 22).

Chiaramente sul display verranno presentati solo quei joystick ai quali è stato assegnato almeno un canale:

- Gyro X: selezionando questo parametro il comando si attiva ruotando il tablet a destra o a sini-

stra;

- Gyro Z: selezionando questo parametro il comando si attiva ruotando il tablet in avanti e indietro;
- Mode 1-4: associa il tipo di controllo alla modalità di pilotaggio;



Fig. 22

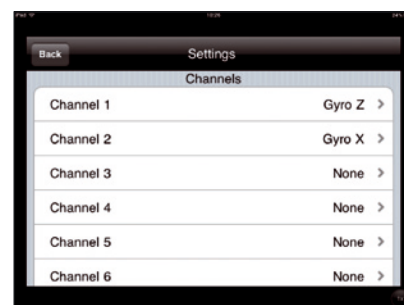


Fig. 21



Fig. 20



Fig. 17



Fig. 18

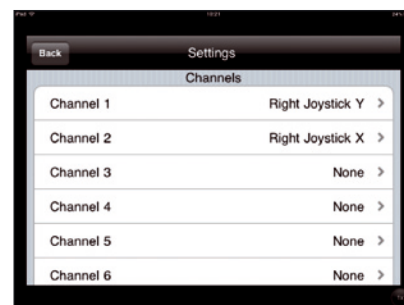


Fig. 19

- Channels 1-8: associa il tipo di controllo al canale;
- Digital outputs: associa il tipo di controllo all'uscita digitale;
- Hold position: l'asse relativo al joystick selezionato non torna alla posizione di zero una volta rilasciato ma mantiene la posizione impostata;
- Buttons: permette di impostare il comportamento dei pulsanti: "Trigger" attiva il comando associato al pulsante solo per il tempo durante il quale il pulsante viene mantenuto premuto, "Toggle": il pulsante cambia stato (ON/OFF) ad ogni pressione, OFF/ON: associa il valore che assume il pulsante in ciascuno stato, premuto o rilasciato.

Infine due ultime note per quanto

riguarda la possibilità di aggiornare il firmware caricato sul modulo WIRC e la possibilità di riportare la configurazione come all'uscita dalla fabbrica.

Quando si avvia l'applicazione con il tablet collegato ad una rete Internet, quest'ultima verifica la presenza di aggiornamenti, proponendone l'installazione.

Nel caso del firmware del modulo WIRC, dopo avere accettato l'aggiornamento, questo parte automaticamente e richiede circa tre minuti durante i quali è necessario non scollegare le batterie e mantenere collegato il modulo per tutto il tempo.

Alla fine del processo di aggiornamento è necessario riconnettere il modello alle batterie.

Nel caso si desideri resettare il

modulo per riportarlo alla configurazione di fabbrica, è necessario seguire la seguente procedura. Connettersi al sito:

<http://wirc.dension.com> e, dalla sezione downloads, scaricare il file WRC_Reset_Settings.den.

Copiate il file su una chiavetta USB. Inserite la chiavetta USB sul connettore USB al posto del dongle WiFi.

Date tensione al modulo WIRC e attendete circa 30 secondi per il completamento del processo. Togliete la chiave USB e ricolocate il dongle WiFi.

Ripetete la configurazione come per il modulo nuovo.

Il sistema WIRC di Dension è reperibile in Italia presso Futura Elettronica (www.futurashop.it) al prezzo di 132,00 Euro IVA compresa. □



RADIOCOMANDO SU RETE Wi-Fi

cod. WIRC

€132,00



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

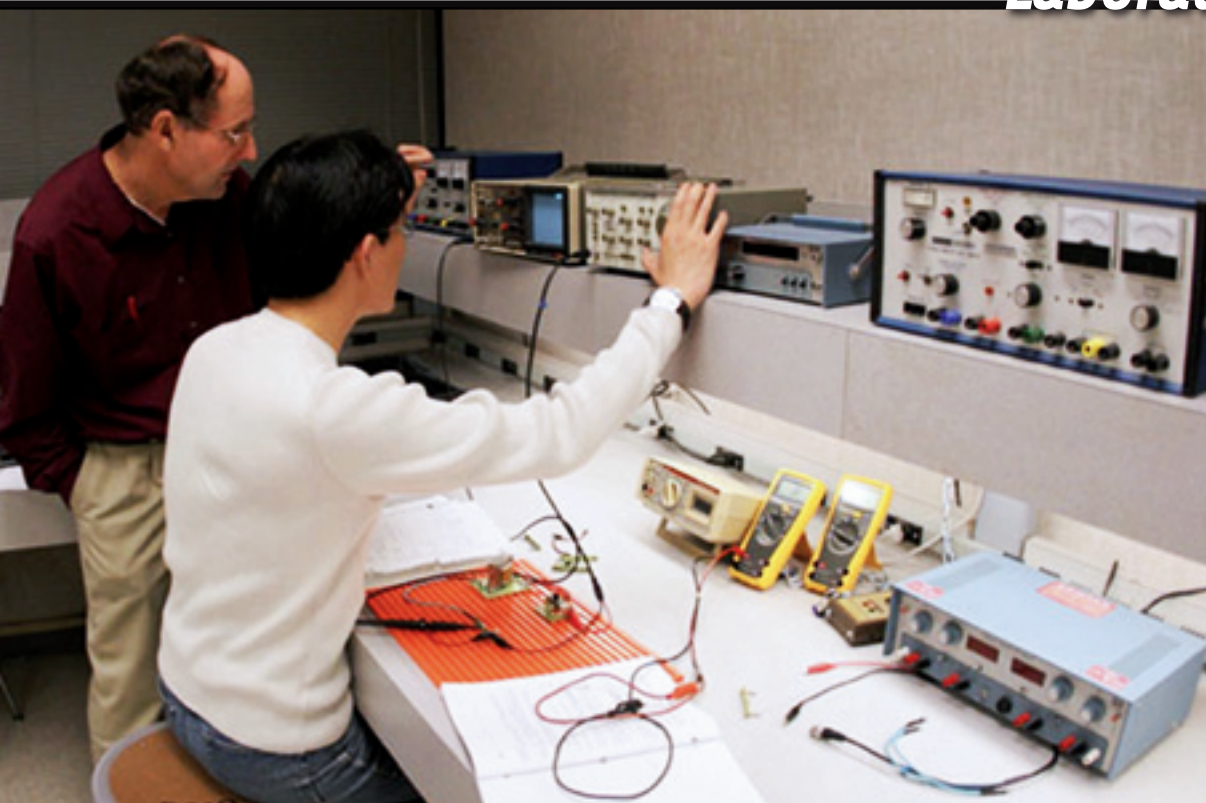





Via Adige, 11
21013 Gallarate (VA)
Tel. 0331/799775 • Fax. 0331/792287

Maggiori informazioni su questo prodotto e su altri dispositivi
per robotica sono disponibili sul sito www.futurashop.it
tramite il quale è anche possibile effettuare acquisti on-line.

Prezzi IVA inclusa.



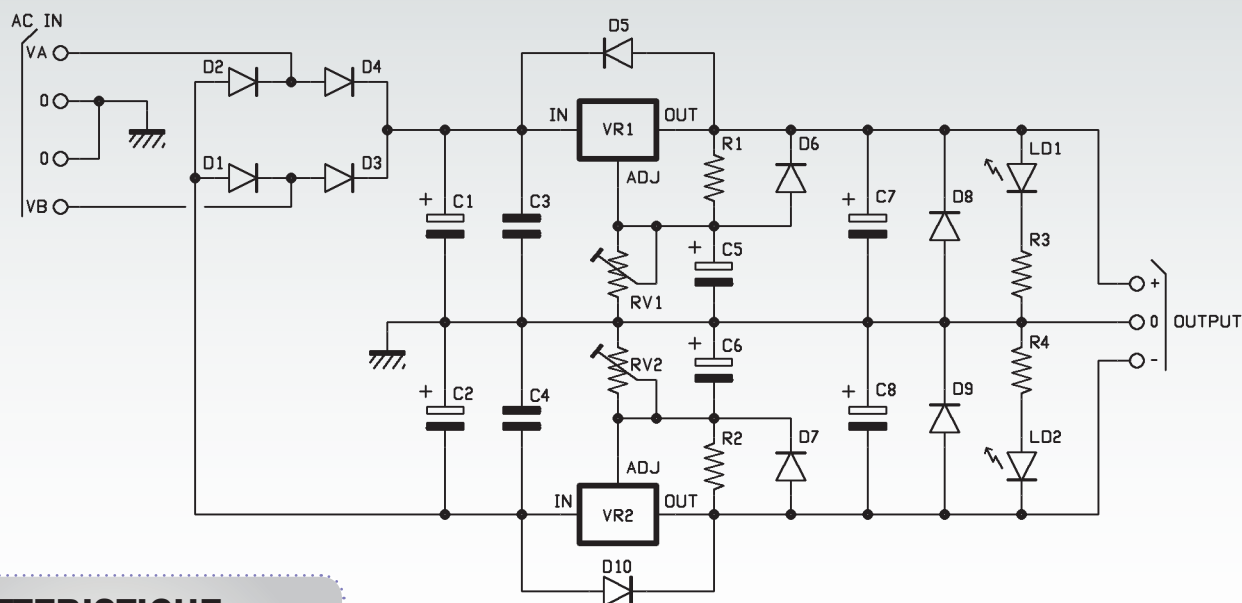
Alimentato con un trasformatore a presa centrale, può erogare una tensione continua simmetrica o asimmetrica rispetto a massa regolabile tra $\pm 1,25$ e ± 18 V ed una corrente di 1 ampere. Ideale per alimentare amplificatori audio e per strumentazione.

ALIMENTATORE DUALE REGOLABILE

di DAVIDE SCULLINO



I circuiti per l'amplificazione dell'audio e in generale quelli degli strumenti di misura e dei sensori che restituiscono segnali analogici, normalmente funzionano a tensione duale rispetto a massa, ovvero richiedono un'alimentazione che consiste in due tensioni di uguale valore assoluto ma di segno opposto tra loro; la ragione di ciò è che amplificatori e buffer devono poter erogare tensioni sia positive che negative rispetto a massa. Questo è il caso, ad esempio, del trattamento di componenti BF sinusoidali alternate, che oscillano tra valori positivi e negativi e perciò richiedono all'amplificatore di poter far assumere alla sua uscita tensioni bidirezionali, pena inaccettabili distorsioni. Nel caso



CARATTERISTICHE TECNICHE

- Tensione d'uscita: $\pm 1,25 \div \pm 24$ Vcc
- Corrente nominale d'uscita: ± 1 A
- Corrente massima d'uscita: $\pm 1,5$ A
- Massima tensione d'ingresso: 2×24 Vca
- Regolazione della tensione
- Protezione dal cortocircuito

dei finali di potenza audio, poi, il problema diventa ancor più rilevante perché l'uscita deve poter fornire corrente bidirezionale, allo scopo di far compiere alle membrane degli altoparlanti movimenti in avanti e indietro. In molti casi l'effetto dell'alimentazione duale si simula ponendo in serie all'uscita un condensatore che, scaricandosi, fornisce la corrente che l'amplificatore erogherebbe se alimentato a tensione simmetrica, ma tale soluzione è accettabile per i piccoli amplificatori e non particolarmente per quelli di potenza, sebbene sia stata molto usata nei decenni passati. Per fornire ad un circuito una tensione duale, esistono varie soluzioni; qui ve ne proponiamo una basata su una configurazione lineare e capace di fornire grandezze stabilizzate. L'alimentatore duale lineare si basa su un trasformatore che

ha il secondario a presa centrale (capace, cioè, di fornire tra i propri estremi in ogni istante due tensioni di uguale ampiezza ma di fase opposta rispetto al capo centrale) il quale alimenta un doppio raddrizzatore a ponte ed eventualmente un regolatore di tensione. L'alimentatore qui descritto può essere usato sia per alimentare gli stadi di un amplificatore, mixer, equalizzatore, nonché uno strumento di misura, un campionario, un D/A converter o l'interfaccia verso i sensori di un apparato medicale; ma nulla vieta di impiegarlo per costruirsi un piccolo alimentatore duale regolabile da laboratorio.

SCHEMA ELETTRICO

Il nostro alimentatore duale è di tipo stabilizzato, nel senso che oltre a fornire una tensione simmetrica, permette di stabilizzarla al variare del carico, erogando fino ad un massimo di 1 A per ciascun ramo (ossia 1 ampere dall'uscita positiva ed altrettanto da quella negativa); inoltre ci consente di regolare i valori delle tensioni di uscita, tra un minimo di 1,25 ed un massimo di 18 V, ossia $+1,25 \div 18$ V sul ramo positivo e $-1,25 \div 18$ V su quello

negativo. Il circuito si basa su un ponte a diodi discreto e due regolatori di tensione "adjustable", ossia a tensione d'uscita regolabile: si tratta dell'LM317 e dell'LM337, che sono tra loro complementari in quanto svolgono identica funzione, solo che il primo è pensato per regolare tensioni positive ed il secondo per stabilizzare tensioni negative. Questi regolatori offrono una perfetta stabilizzazione della tensione nei limiti della corrente erogabile, implementando una protezione dal cortocircuito e dal sovraccarico, il che significa che in ogni condizione non erogano più di 1,5 ampere, anche se l'utilizzatore chiedesse di più.

Vediamo subito come funziona l'alimentatore, supponendo di alimentarne l'ingresso (ossia i punti AC IN) con il secondario a presa centrale di un trasformatore da rete, collegato con la presa sui contatti 0 (massa) e gli estremi indifferentemente sui punti VA e VB; trattandosi di alimentazione in corrente alternata, è irrilevante quale capo del secondario sia su VA e quale su VB.

Il ponte raddrizzatore di Graetz formato dai diodi D1, D2, D3 e

D4, raddrizza le tensioni sinusoidali fornite dal trasformatore ed ottiene impulsi unidirezionali positivi rispetto a massa sui catodi di D3 e D4 e negativi sugli anodi di D1 e D2. Grazie alla presenza dei condensatori C1 e C3, gli impulsi positivi vengono filtrati e diventano una componente continua ben livellata; analogamente accade sul ramo negativo, per effetto di C2 e C4. Dunque, ci troviamo con due tensioni teoricamente identiche, ma di verso opposto, fra il terminale IN del regolatore VR1 e massa e fra l'IN del VR2 e massa; ognuna delle tensioni vale, in assenza di carico, 1,414 volte il valore efficace della componente alternata fornita dal relativo mezzo secondario, il tutto diminuito della caduta sui due diodi del ponte che di volta in volta conducono. Chiamando V_{eff} la tensione di ciascuna metà del secondario e V_d la caduta diretta sui diodi (ognuno trattiene tipicamente 0,6 V) vediamo che il valore assoluto (cioè trascurando il segno) della tensione continua ai capi dei condensatori di filtro (V_{cc}) vale:

$$V_{cc} = V_{eff} \times 1,414 - 2 \times 0,6 \text{ V}$$

Supponiamo di alimentare il circuito con un trasformatore avente il secondario da 12+12 volt efficaci e vediamo che all'ingresso dei regolatori, rispetto a massa, abbiamo:

$$V_{cc} = 12 \times 1,414 - 1,2 \text{ V} = (16,968 - 1,2) \text{ V} = 15,768 \text{ V}$$

Questo è il valore della tensione positiva presente all'ingresso dell'LM317 e di quella negativa all'input dell'LM337. Bene, vediamo adesso cosa fanno i due regolatori, partendo da quello positivo, ossia VR1, usato per fornire una tensione V_{out} sta-

bilizzata funzione di V_{in} , secondo la relazione seguente:

$$V_{out} = V_{ref}(1 + RV1/R1) + I_{adj} \times RV1$$

dove per V_{ref} s'intende la tensione che normalmente è presente tra il piedino ADJ e massa e che ammonta ad 1,25 volt.

La relazione sopra riportata può essere approssimata riconducendola a quella semplificata:

$$V_{out} = 1,25 \text{ V}(1 + RV1/R1)$$

trascurando il termine $I_{adj} \times RV1$, perché la corrente I_{adj} è trascurabile rispetto a quella di uscita I_{out} (infatti I_{adj} è dell'ordine di poche decine di microampere).

Tabella 1

TENSIONE STABILIZZATA D'USCITA	TENSIONE ALTERNATA D'INGRESSO AC IN
da $\pm 1,25 \text{ A} \div 5 \text{ V}$	2 X 9 VCA
da $\pm 5 \text{ A} \div 9 \text{ V}$	2 X 12 VCA
da $\pm 9 \text{ A} \div 12 \text{ V}$	2 X 15 VCA
da $\pm 12 \text{ A} \div 15 \text{ V}$	2 X 18 VCA
da $\pm 15 \text{ A} \div 24 \text{ V}$	2 X 22 VCA

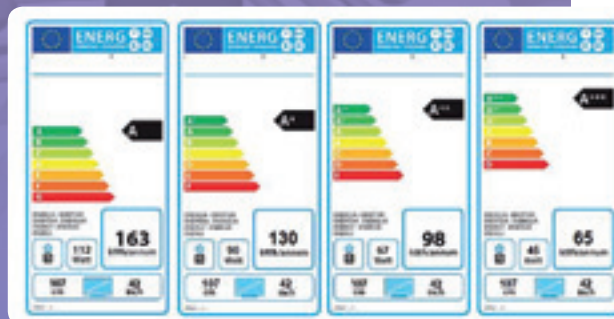
Tensioni del trasformatore con secondario a presa centrale correlate con le tensioni continue d'uscita per un'erogazione di corrente pari ad 1 ampere.

Nel nostro caso, essendo RV1 un trimmer, è evidente che possiamo variare la tensione che il regolatore presenta tra V_{out} e massa; la variazione è possibile tra 1,25 V (che corrispondono all'avere il trimmer cortocircuitato, ossia a resistenza zero) e 18 V (trimmer tutto inserito). Analogo

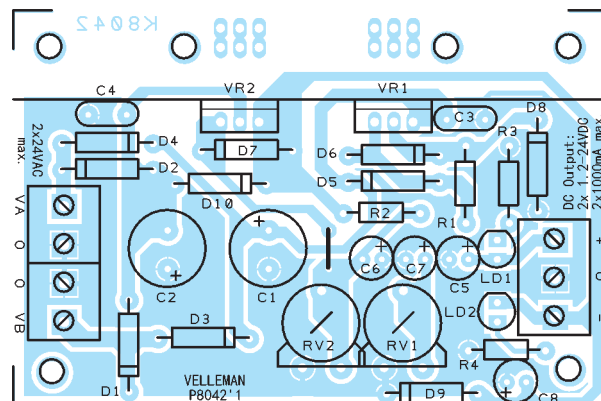
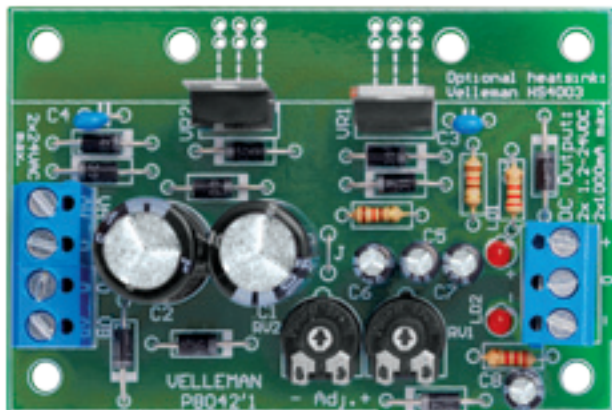
La nuova etichetta energetica

Per gli elettrodomestici sono arrivate le nuove etichette indicanti la classe energetica: seguendo la legislazione europea, vengono introdotte le nuove classi energetiche aggiuntive A+, A++ e A+++, che sono più congrue con la maggiore efficienza energetica degli elettrodomestici moderni. In tutti gli stati UE l'etichetta sarà uniforme e riporterà frecce di colori diversi per distinguere le diverse classi di efficienza energetica: se un prodotto avrà la freccia color verde scuro vorrà dire che si tratta di un prodotto ad alta

efficienza energetica, mentre un elettrodomestico con una freccia rossa indica che il prodotto è a bassa efficienza. Le etichette inoltre specificano il consumo energetico annuale in kWh. Per aiutare i consumatori ad orientarsi nella lettura delle nuove etichette, il WWF si è fatto promotore, sul sito www.topten.eu, di un'iniziativa che ha portato alla realizzazione di una guida comparativa tra i 10 migliori prodotti per capire quali, a parità di prestazioni e caratteristiche, consumano meno energia elettrica.



[piano di MONTAGGIO]



Elenco Componenti:

R1, R2: 120 ohm
R3, R4: 2,2 kohm
RV1: Trimmer 2,2 kohm MO
RV2: Trimmer 2,2 kohm MO
C1: 1.000 μ F 35 VL elettrolitico
C2: 1.000 μ F 35 VL elettrolitico

C3: 100 nF multistrato
C4: 100 nF multistrato
C5÷C8: 10 μ F 63 VL elettrolitico
D1÷D10: 1N4007
VR1: LM317
VR2: LM337

LD1, LD2: LED 3 mm Rosso

Varie:

- Morsettiera 2 poli passo 5,08 mm (2 pz.)
- Morsettiera 3 poli passo 5,08 mm
- Circuito stampato

è il discorso per la sezione negativa, presieduta dall'LM337, solo che le tensioni sono di valore opposto; vale la formula semplificata:

$$V_{out} = 1,25V(1 + RV2/R2)$$

In questo caso, quando il trimmer RV2 è cortocircuitato la tensione d'uscita è pari a -Vref, mentre con il trimmer tutto inserito, dal regolatore escono -18 volt.

Naturalmente il discorso vale a condizione che la tensione applicata all'ingresso dei regolatori sia sufficiente, ovvero superi quella di uscita Vout della caduta uscita/ingresso di VR1 e VR2; va poi considerato che a pieno carico la tensione filtrata dagli elettrolitici si abbassa di circa il 20 % rispetto al valore a vuoto, pertanto per tensione in ingresso ai regolatori intendiamo quella sotto carico. Tutto ciò, perché sia garantita la tensione calcolata con i partitori, ossia impostata

dai trimmer; altrimenti il regolatore duale non potrà fornire sempre tensioni stabilizzate. Volendo farci un'idea, diciamo che LM317 ed LM337 per funzionare correttamente devono presentare una caduta (drop-out) tra uscita ed ingresso dell'ordine di 3 V, quindi, ad esempio, se vogliamo 1,25 V in uscita è necessaria una Vin minima di 1,25 V + 3 V = 4,25 V. Sapendo che un 20 % della tensione a vuoto si perde sotto carico, dobbiamo stare un po' più in alto ed imporre una Vin di almeno 5,2 volt. Per comodità riportiamo nella **Tabella 1** le tensioni suggerite per il secondario del trasformatore a presa centrale al fine di ottenere i più comuni valori di tensione stabilizzata all'uscita del circuito; tenete comunque presente che all'ingresso di ciascuno dei regolatori non devono giungere mai più di 40 volt e che i condensatori elettrolitici C1 e C2 sono stati previsti per una tensione di lavoro di 35 volt, ragion per cui è bene non

applicare, ai contatti AC IN, oltre 24+24 Vca efficaci.

È bene attenersi alle indicazioni perché un trasformatore che eroga una tensione troppo bassa non garantisce la tensione stabilizzata, mentre uno che fornisce troppo, costringe i regolatori, a parità di corrente erogata in uscita, a dissipare troppa potenza e quindi impone l'uso di grossi dissipatori; infatti sia l'LM317 che l'LM337 dissipano parte dell'energia fornita dal trasformatore e la potenza persa (Pd) su di essi è pari al prodotto della corrente assorbita (circa uguale a quella erogata in uscita) per la caduta tra i piedini OUT e IN. Ad esempio, erogando 1 ampere, se la tensione ai capi dei condensatori di filtro è 20 V (trasformatore da 2x15 Vca) e i trimmer vengono impostati per ottenere in uscita 10 volt, su ciascun regolatore cadono 10 V. Quindi la perdita di potenza in ciascuno dei VR1 e VR2 è pari a:

$$Pd = 1A \times 10 V = 10 W$$

Bene, detto ciò completiamo la descrizione del circuito analizzando i componenti che restano; partiamo dai diodi D5 e D10, usati per far rifluire la corrente direttamente agli ingressi dei regolatori quando viene a mancare l'alimentazione alternata ai punti AC IN, ovvero allo spegnimento dell'alimentatore: scopo di questi diodi è impedire che quando C1 e C3 per il ramo positivo e C2 e C4 per quello negativo si scaricano, C7 e C8 mantengano la tensione alle uscite dei regolatori maggiore di quella agli ingressi. Questa situazione si verifica assai facilmente se alle uscite non c'è carico e potrebbe danneggiare gli stadi di uscita dei regolatori, almeno quando la tensione scelta in uscita è relativamente alta. Il diodo D6 per il ramo positivo ed il D7 per quello negativo hanno una funzione simile: proteggono i piedini ADJ dalla scarica degli elettrolitici C5 e C6, facendo fluire la corrente direttamente sui condensatori di uscita e quindi bypassando il circuito ADJ-OUT.

Notate che C5 e C6 sono stati inseriti per filtrare dai disturbi e dal ripple (ondulazione residua che i filtri capacitivi non riescono ad eliminare e che potrebbe

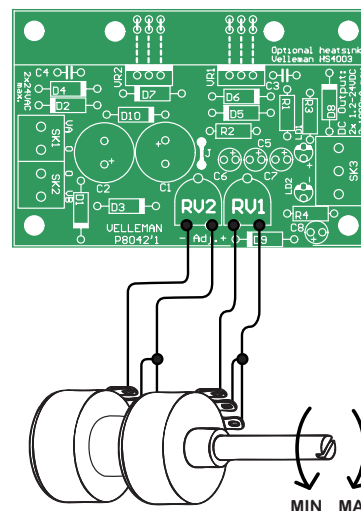
ripercuotersi in uscita) la tensione portata come riferimento al piedino ADJ di ciascuno dei regolatori.

I diodi D8 e D9, invece, servono per cortocircuitare tensioni inverse impulsive che possono presentarsi sulle linee di uscita alimentando particolari carichi. Infine, i diodi luminosi LD1 ed LD2 indicano, accendendosi, la presenza delle tensioni rispettivamente positiva e negativa alle uscite dell'alimentatore.

REALIZZAZIONE PRATICA

Bene, spendiamo adesso qualche parola sulla costruzione del circuito, che si presenta molto semplice dato che richiede l'uso di componenti (e pochi) a montaggio tradizionale; la prima cosa da fare è preparare lo stampato per fotoincisione, dopo averne scaricato la traccia dal nostro sito web www.elettronica.in.

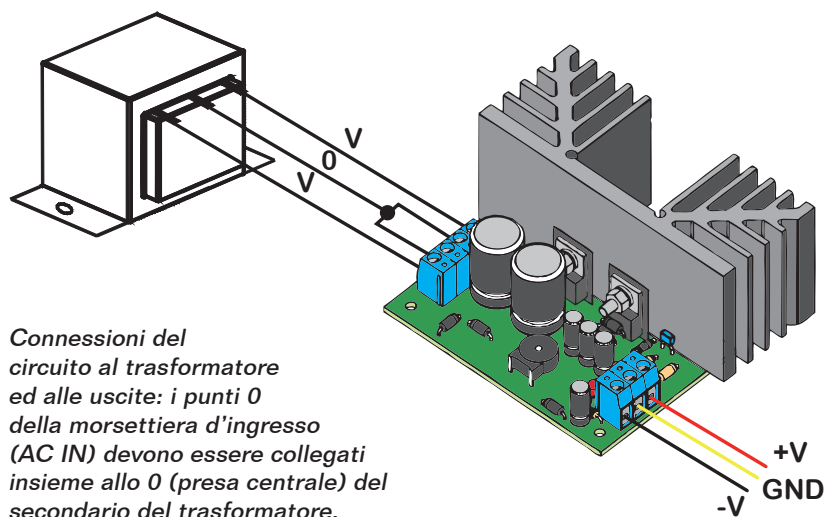
Incisa e forata la basetta, iniziate ad inserirvi e saldarvi resistenze e diodi al silicio (fate attenzione alla polarità indicata per questi ultimi nel piano di montaggio che vedete in queste pagine...) e poi i due trimmer, quindi sistemate i condensatori, dando la precedenza a quelli non polarizzati e rispettando la pola-



Per variare simultaneamente le due tensioni potete sostituire i due trimmer con un doppio potenziometro lineare di pari valore, collegandolo come mostrano in questo disegno, ossia a doppio reostato.

rità degli elettrolitici; fatto ciò, disponete i due LED orientandoli come mostrato in queste pagine, ossia con il catodo (lato smussato) che guarda verso le resistenze R1 ed R3.

Ora non vi resta che montare i regolatori LM317 ed LM337 tenendoli in piedi, con il corpo sollevato di 7÷8 mm dalla superficie dello stampato; per consentire agli integrati di smaltire il calore durante il funzionamento, montateli ciascuno a ridosso di un radiatore di calore avente resistenza termica di 8°C/W, serrando con un bullone da 3 MA e interponendo della pasta al silicone per agevolare il trasferimento del calore dal corpo degli integrati al metallo dei radiatori. Potete anche adottare un dissipatore unico per VR1 e VR2, che dovrà avere resistenza termica non superiore ai 4°C/W; in questo caso dovrete isolare elettricamente gli integrati dal metallo del dissipatore, mediante l'interposizione (tra la zona metallica di ciascuno e il dissipatore stesso) di un foglietto di mica spalmato per bene di pasta al



Connessioni del circuito al trasformatore ed alle uscite: i punti 0 della morsettiera d'ingresso (AC IN) devono essere collegati insieme allo 0 (presa centrale) del secondario del trasformatore.

33^a FIERA REGIONALE DELL'ELETTRONICA SCANDIANO (REGGIO EMILIA) 18/19 FEBBRAIO 2012 CENTRO FIERISTICO



**MERCATINO DELLE PULCI
RADIOAMATORIALI
TELEFONIA - COMPONENTISTICA
COMPUTER - HI-FI CAR - RADIANTISMO
CB e OM - VIDEOREGISTRAZIONE**

**ORARI: Sabato 18 Febbraio 9-18,30
Domenica 19 Febbraio 9-18**

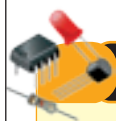


**INGRESSO euro 7 50 RIDOTTO euro 5
GRATUITO ragazzi fino a 14 anni**

**INFO: 0522/764302-857436
www.fierasandiano.it**

silicone su entrambe le facce.

In alternativa si possono usare isolatori in teflon grigio, che trasferiscono bene il calore anche senza essere spalmati di pasta al silicone. I dissipatori possono essere fissati al circuito stampato utilizzando i fori appositamente previsti, ma anche stagnati nelle piazzole di ancoraggio che si trovano dietro ai regolatori. Notate che gli LM317 ed LM337 possono erogare fino ad un massimo di 1,5 ampere, tuttavia perché ciò sia possibile bisogna sovradimensionare i dissipatori, scegliendone uno da non oltre 5°C/W per regolatore o uno unico da non più di 2,5°C/W per entrambi i VR1 e VR2. Una volta completato il montaggio, potete saldare delle morsettiere a passo 5 mm in corrispondenza delle piazzole AC IN e delle OUTPUT, in modo da agevolare le connessioni. Il circuito può essere sistemato all'interno del mobile dell'unità che deve alimentare o in un proprio contenitore, come nel caso in cui con esso vogliate realizzare un alimentatore da laboratorio. Volendo è possibile riunire i due trimmer sostituendoli con un doppio potenziometro, che collegherete con una sezione a VR1 e l'altra a VR2; in pratica ricordate di unire cursore ed estremo di sinistra di ciascun potenziometro connettendoli con un filo alla massa e di portare i contatti estremi rimasti liberi, ciascuno ad uno degli ADJ dei regolatori. Notate che essendoci due trimmer distinti, ognuno dei regolatori può essere impostato per fornire una tensione diversa da quella dell'altro; ciò torna utile in quelle applicazioni in cui servano due tensioni, una positiva e l'altra negativa, non simmetriche. ■



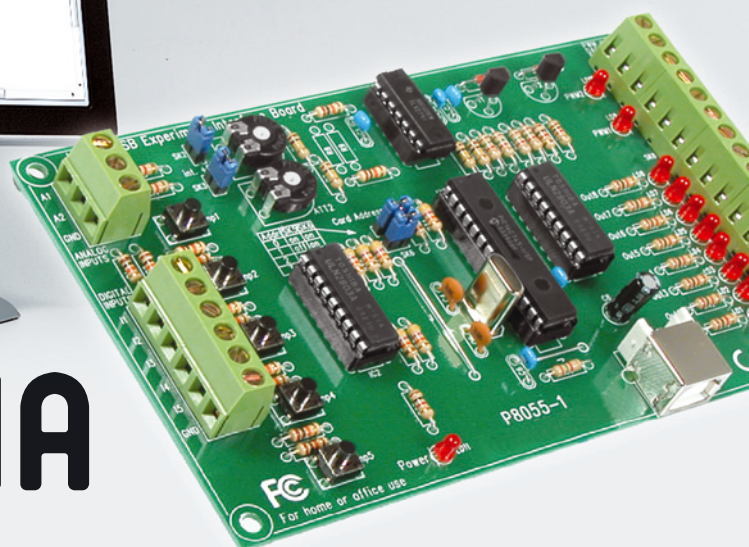
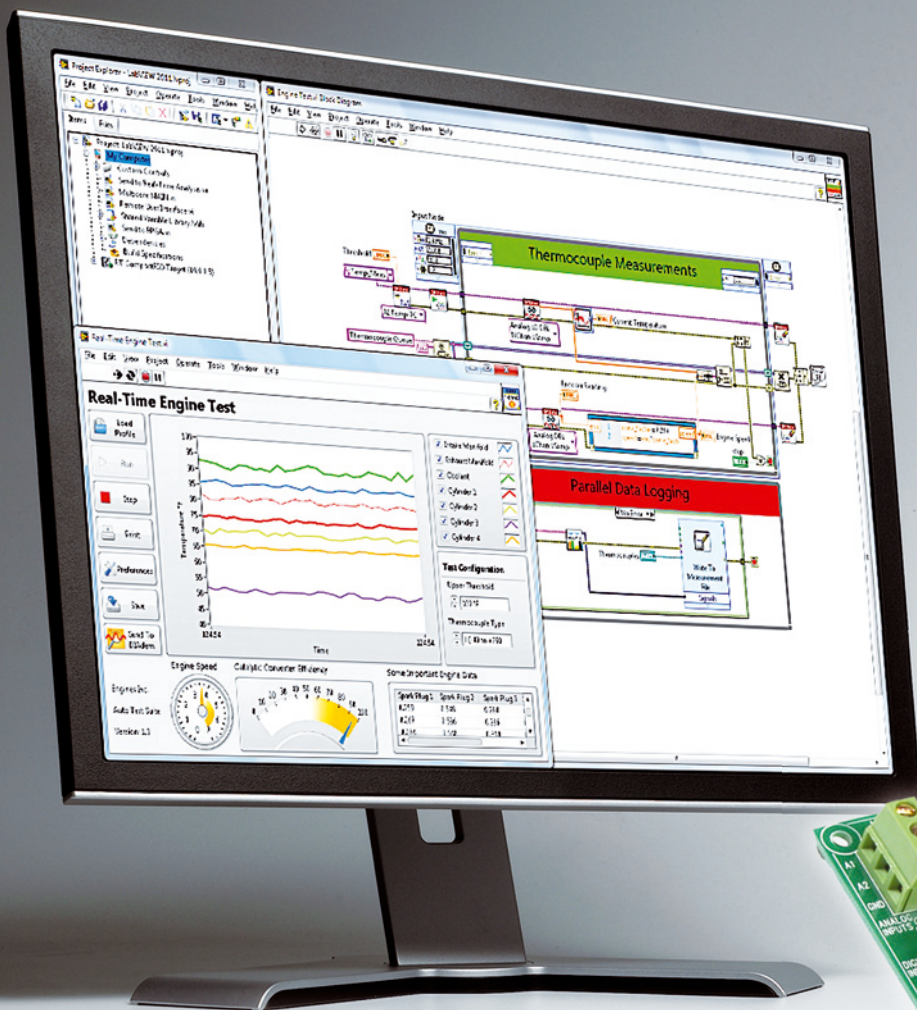
per il MATERIALE

L'alimentatore descritto in queste pagine è disponibile in scatola di montaggio prodotta da Velleman. Il kit (cod. K8042, Euro 11,00 IVA compresa), il trasformatore e il dissipatore termico sono acquistabili presso Futura Elettronica. Il master del circuito stampato può essere scaricato direttamente dal sito della rivista (www.elettronicaonline.it).

Il materiale va richiesto a:

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

Sfruttiamo le potenzialità di LabVIEW per realizzare un'interfaccia grafica con cui controllare la scheda a relé Velleman K8055; un'occasione per imparare ad usare l'ambiente di sviluppo per la gestione di dispositivi non NI, sfruttando periferiche presenti nei PC come l'interfaccia USB.



INTERFACCIA LABVIEW PER K8055

di FRANCESCO FICILI

Nelle ultime puntate del corso dedicato a LabVIEW (pubblicato nei fascicoli dal n° 144 al n° 152) abbiamo visto come interfacciare una scheda di acquisizione dati (per la precisione si trattava di una NI USB-6008, un multifunction DAQ di produzione National Instruments) al nostro PC, sfruttando la potenza della programmazione grafica e i driver forniti dalla NI, come il pacchetto DAQmx; quest'ultimo incorpora il tool DAQAssistant, che permette, con semplici comandi parametrici all'interno di un wizard, di generare del codice

per l'interfacciamento a basso livello della scheda, svincolando il programmatore dalla gestione di queste problematiche, che costituiscono un consumo non indifferente di tempo (e quindi un calo della produttività) nelle fasi di sviluppo di un prodotto software. Come ci si comporta, però, quando a dover essere interfacciato non è un dispositivo di produzione NI, ma una generica scheda elettronica? Naturalmente l'ambiente di sviluppo dispone delle risorse necessarie anche per la risoluzione di questa tipologia di problematiche. In generale

La scheda Velleman K8055

La K8055 è una scheda di interfaccia I/O ad interfaccia USB. Si basa su un PIC16C745, dotato di transceiver USB interno e non necessita quindi di altri circuiti per la realizzazione dell'interfaccia (come i popolari convertitori USB seriali, spesso usati in schede di questo tipo). La scheda K8055 dispone delle seguenti risorse hardware:

- 5 ingressi digitali (con pulsanti di test);
- 2 ingressi analogici con opzioni di attenuazione e amplificazione (disponibile mediante jumper una tensione di test di +5V);

- 8 uscite digitali open-collector (valori massimi: 50V/100mA) con LED di indicazione dell'attività di ciascuna;
- 2 uscite analogiche (da 0 a 5 V, impedenza di uscita 1,5 k Ω);
- 2 uscite PWM (open-collector, da 0% a 100%, 100mA/40V);
- Tempo di conversione medio: 20ms per comando;
- Alimentazione: da porta USB (assorbimento, 70 mA circa).

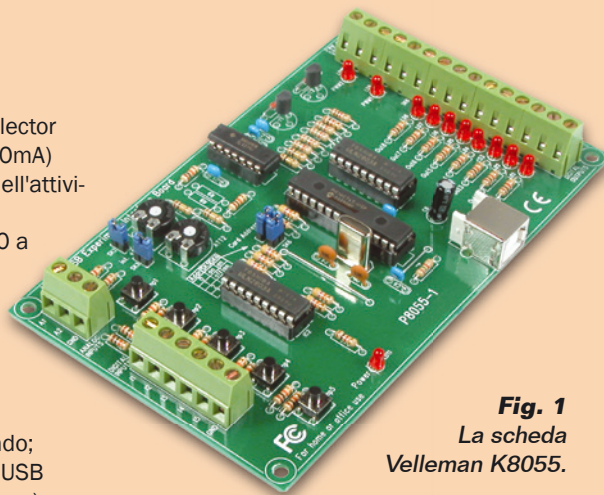
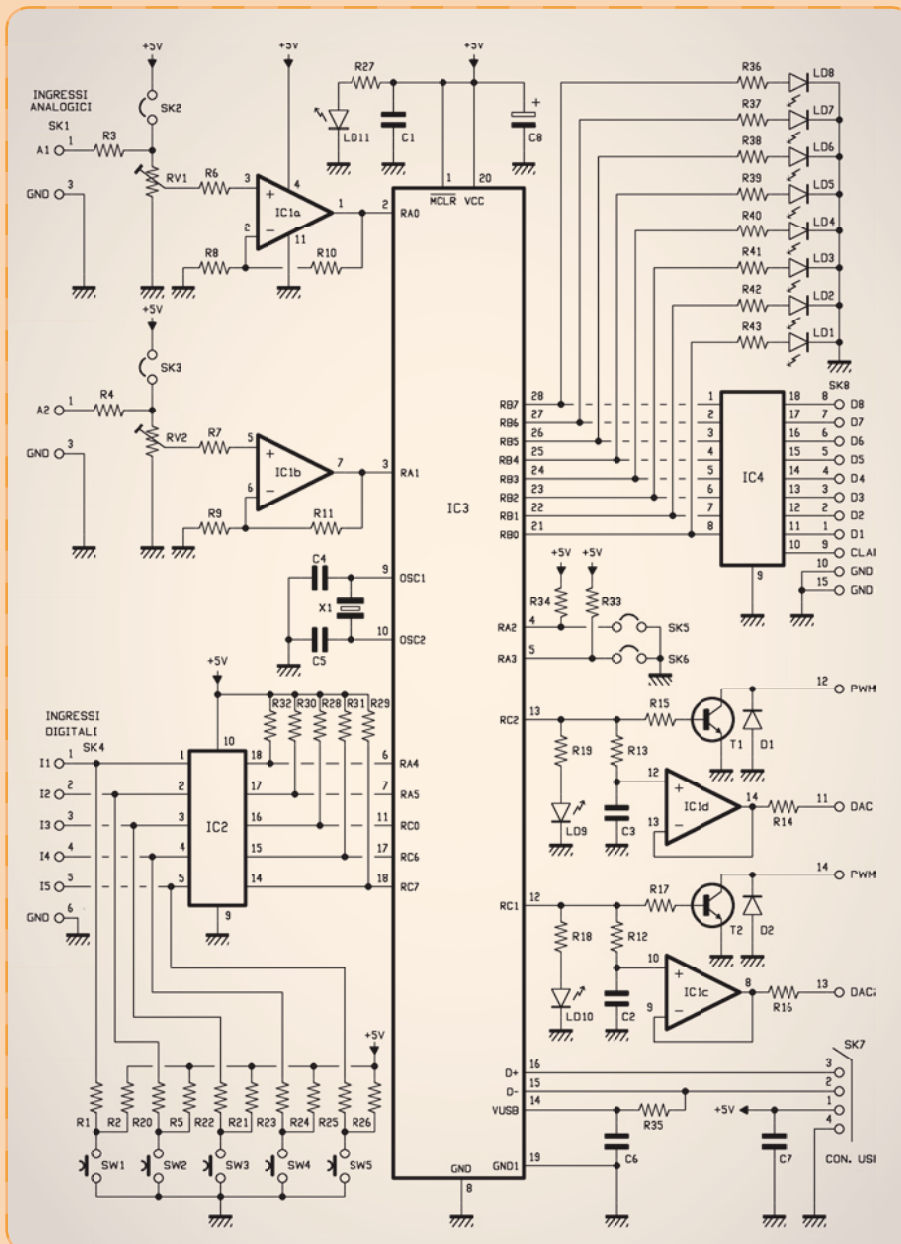


Fig. 1
La scheda Velleman K8055.

In **Fig. 2** è riportato lo schema



elettrico del dispositivo, il cui cuore è il PIC16C745, interfacciato dal lato delle uscite digitali attraverso un ULN2003, un array di darlington che permettono di pilotare carichi come relè o interruttori a stato solido; dal lato degli ingressi digitali, un secondo ULN2003 permette di leggere tensioni anche elevate senza rischiare di danneggiare gli ingressi digitali del PIC. Gli ingressi analogici sono invece bufferizzati tramite un TLV274, doppio amplificatore operazionale. Gli amplificatori sono collegati in configurazione non invertente ed è possibile regolare i guadagni tramite i due trimmer (ATT1 e ATT2). Si può anche aumentare il guadagno tramite l'inserimento di due resistenze R8 ed R9, di valore adatto.

La scheda dispone inoltre di due uscite PWM (utilizzabili per controllare ad esempio motori elettrici) e di due uscite analogiche (ricavate filtrando il segnale PWM con un filtro RC del primo ordine) bufferizzate tramite voltage follower. L'interfaccia di comunicazione è una USB 1.1, e la classe scelta è la HID (Human Interface Device), quindi il PC riconosce la scheda senza che venga installato alcun driver aggiuntivo: la K8055 viene riconosciuta come un dispositivo di interfaccia uomo-macchina (come ad esempio un mouse, o una tastiera USB). La gestione successiva è affidata alle DLL fornite dalla Velleman, necessarie all'interfacciamento software.

Fig. 2 - Schema elettrico della scheda Velleman K8055.

LabVIEW dispone sia di blocchi di gestione per i protocolli più comuni (Ethernet, USB, Seriale, ecc.) che anche dei blocchi per il caricamento di DLL generiche che possono essere fornite dal produttore della scheda da gestire, come supporto software per l'uso dell'hardware.

Qui vogliamo esplorare questo tipo di situazione, sviluppando un'interfaccia grafica per la gestione della K8055, una scheda di acquisizione/controllo I/O dotata di interfaccia USB, prodotta dalla Velleman.

LA DLL FORNITA CON LA K8055

A supporto della scheda K8055, la Velleman fornisce una ricca libreria di funzioni (sotto forma di DLL) che può essere scaricata liberamente e utilizzata per realizzare interfacce di gestione della scheda. Le funzioni disponibili sono elencate di seguito.

- **OpenDevice:** apre il collegamento con la scheda e carica i driver necessari a comunicare tramite la porta USB; questa procedura deve essere eseguita prima di ogni tentativo di comunicare con la scheda e si può anche utilizzare per selezionare la scheda attiva in modo da leggere e scrivere i dati. Tutte le routine di comunicazione dopo questa chiamata di funzione, sono indirizzate a tale scheda fino a quando non viene selezionata l'altra scheda da parte di questa chiamata di funzione.
- **CloseDevice:** scarica le routine di comunicazione per la scheda e scarica il driver necessario a comunicare tramite la porta USB. Questa è l'ultima azione del programma applicativo prima dell'interruzione.
- **ReadAnalogChannel:** il convertitore analogico/digitale selezionato legge la relativa

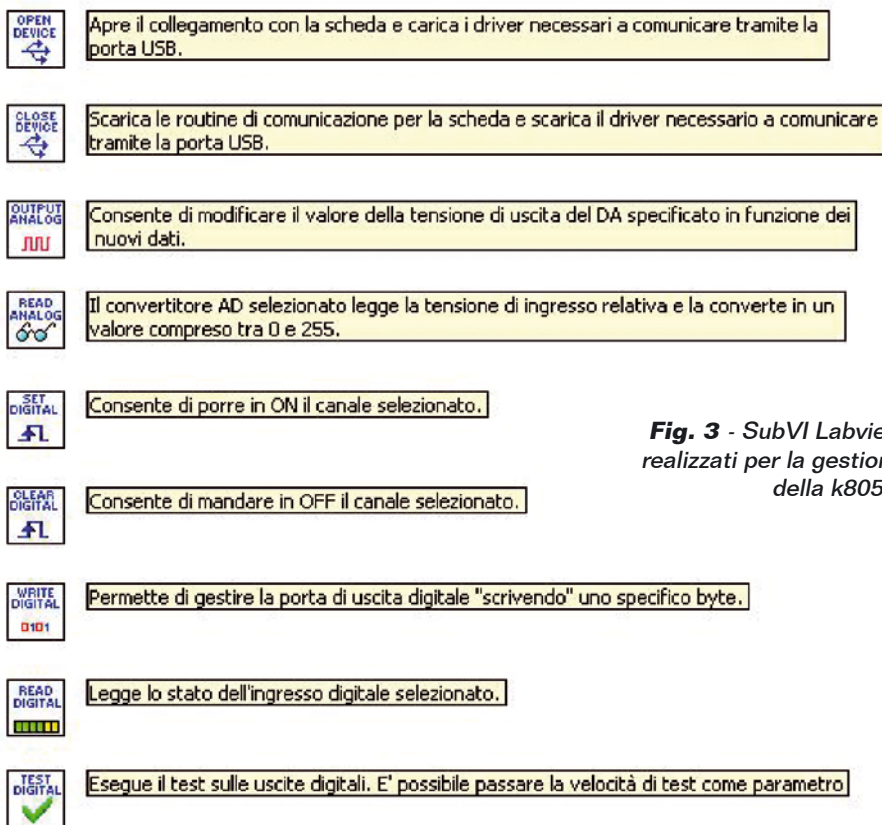


Fig. 3 - SubVI Labview realizzati per la gestione della k8055.

- tensione di ingresso e la converte in un valore compreso tra 0 e 255.
- **ReadAllAnalog:** lo stato di entrambi i convertitori A/D viene letto in un insieme di interi lunghi.
- **OutputAnalogChannel:** consente di modificare il valore della tensione di uscita del convertitore D/A specificato, in funzione dei nuovi dati. Il valore 0 corrisponde alla minima tensione di uscita (0 V) mentre 255 coincide con la massima tensione di uscita (+5 V). Per ricavare il valore di "Dati" necessario ad ottenere una determinata tensione di uscita, è possibile utilizzare la seguente formula: $\text{Dati}/255 \times 5V$.
- **OutputAllAnalog:** entrambe le uscite dei due convertitori D/A vengono modificate in funzione dei nuovi dati. Come nel caso precedente, il valore 0 coincide con la minima tensione (0 volt) mentre a 255 corrisponde la massima tensione di uscita (+5 V). Per determinare i valori da utilizzare in modo da ottenere specifiche tensioni di uscita, può essere impiegata la seguente formula: $\text{Dati}/255 \times 5V$.
- **ClearAnalogChannel:** consente di azzerare l'uscita del canale D/A selezionato.
- **ClearAllAnalog:** con questa procedura, entrambi i canali D/A vengono impostati per la minima tensione di uscita (0 V).
- **SetAnalogChannel:** l'uscita DA selezionata viene configurata per ottenere la massima tensione di uscita (+5 V).
- **SetAllAnalog:** entrambe le uscite D/A vengono settate per ottenere il massimo valore di tensione (+5 V).
- **WriteAllDigital:** lo stato delle uscite digitali viene aggiornato con lo stato dei corrispondenti bit nel parametro dei dati. Nel byte il valore più significativo (quello più a

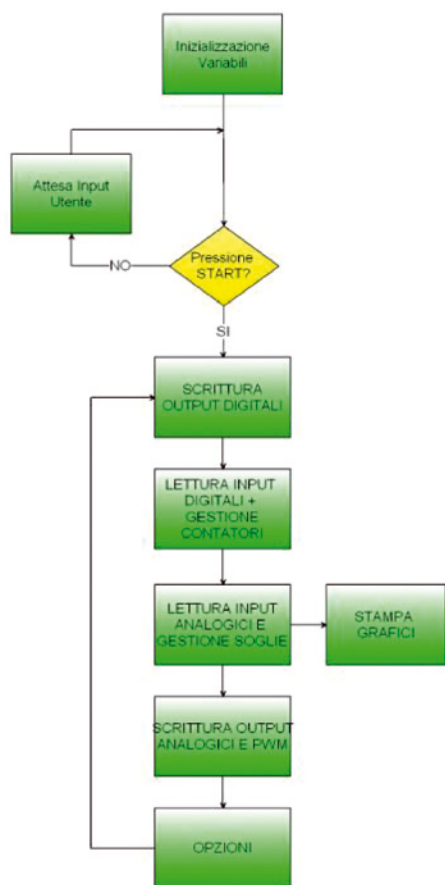


Fig. 4 - Diagramma di flusso semplificato del codice LabVIEW scritto per la realizzazione dell'interfaccia.

impulsi a 16 bit selezionato. Il contatore numero 1 conta gli impulsi che giungono all'ingresso I1, mentre il contatore numero 2 conta gli impulsi che giungono all'ingresso I2.

- **SetCounterDebounceTime:** riguarda la durata dell'antirimbato. Per consentire una precisa lettura degli input quando si utilizzano interruttori meccanici o relè, l'impulso deve rimanere stabile per almeno 2 ms (valore predefinito); il tempo è uguale sia per il fronte di salita che per quello di discesa. Con il valore predefinito, la massima velocità di conteggio è 200 impulsi circa, mentre con un valore di 0 ms la velocità di conteggio è al massimo 2.000 impulsi. Con questa funzione è possibile impostare il tempo dell'antirimbato tra 0 e 5.000 millisecondi.

Partendo da queste funzioni, è stata realizzata una serie di SubVI LabVIEW che possono essere utilizzati all'interno delle nostre interfacce scritte in linguaggio-G. Non sono state replicate tutte le funzioni messe a disposizione dalla DLL, in quanto, partendo dalle funzioni base ed aggiungendo pochi blocchi esterni, siamo in grado di replicare tutte le altre funzioni della libreria. I subVI realizzati sono visibili nella Fig. 3.

In quest'articolo non ci addenteremo tanto sulla realizzazione dei SubVI, ma ci concentreremo su come questi ultimi vengono utilizzati, in connessione con altri SubVI LabVIEW, per la realizzazione del software di controllo della periferica K8055.

L'INTERFACCIA SOFTWARE

Passiamo adesso alla descrizione dell'interfaccia vera e propria.

Ciò che ci proponiamo di realizzare è un software che permetta di gestire gli otto canali di uscita digitali, i due canali analogici di ingresso, i cinque input digitali e, naturalmente, i due canali analogici di uscita con i relativi canali PWM.

Vogliamo poi realizzare una serie di funzioni aggiuntive, in modo da accrescere le funzionalità dell'interfaccia, aggiungendo:

- stampa di un grafico nel tempo (chart) dei due canali analogici, con possibilità di cancellazione dell'acquisizione in tempo reale e visualizzazione selettiva delle tracce (solo canale 1, solo canale 2, entrambi i canali);
- impostazione di cinque soglie di intervento sui canali analogici con comando on-off dei canali digitali;
- gestione di cinque contatori associati agli ingressi digitali, con regolazione del tempo di debounce ed azzeramento;
- gestione del "digital test" sugli output digitali.

Nella Fig. 4 è riportato il flow-chart semplificato del software di gestione della K8055. Come si vede, dopo una fase iniziale di configurazione, il programma passa in esecuzione (condizionata alla pressione del tasto START, in modo da dare il tempo all'utente di collegare la scheda e controllare che i collegamenti siano corretti) e procede in maniera sequenziale ad eseguire i vari "task". Naturalmente la scelta dell'esecuzione sequenziale non è l'unica possibile, specialmente visto che stiamo programmando in LabVIEW, ambiente di sviluppo che è nativamente orientato e supporta facilmente il multitasking. Tuttavia, per ragioni didattiche e pratiche, nel caso specifico l'approccio sequenziale è risul-

sinistra) corrisponde all'uscita 8; quello meno significativo all'uscita numero 1. Se il bit è posto ad 1, l'uscita è attiva (alta) mentre in caso contrario (0) l'uscita è a massa.

- **ClearDigitalChannel:** consente di mandare in OFF il canale selezionato.
- **ClearAllDigital:** consente di porre in OFF tutte le uscite.
- **SetDigitalChannel:** consente di porre in ON il canale selezionato.
- **SetAllDigital:** consente di porre in ON tutte le uscite digitali.
- **ReadDigitalChannel:** legge lo stato dell'ingresso digitale selezionato.
- **ReadAllDigital:** legge lo stato di tutti gli ingressi digitali.
- **ResetCounter:** consente di resettare il contatore degli impulsi.
- **ReadCounter:** la funzione legge lo stato del contatore di

tato più conveniente. Inoltre la scheda in esame presenta tempi di risposta non eccessivamente veloci (circa 20 ms a comando, dettati dal fatto che la classe utilizzata è la HID, quindi una classe non particolarmente veloce), quindi la gestione sequenziale non penalizza eccessivamente l'esecuzione ed i tempi di risposta del software.

Va comunque detto che è possibile realizzare applicazioni più performanti.

Lo sviluppo del software, come in quasi tutti i programmi LabVIEW, comprende una parte di sviluppo dell'interfaccia grafica ed una componente di sviluppo di codice grafico; in quest'articolo non ci soffermeremo troppo a descrivere l'interfaccia grafica, in quanto durante il corso presentato nei mesi scorsi i lettori dovrebbero aver raggiunto un certo livello di confidenza con quest'aspetto, mentre ci soffermeremo maggiormente sui dettagli della programmazione.

Iniziamo quindi la descrizione del codice grafico che implementa i vari task descritti precedentemente nel flow-chart di Fig. 4. La prima operazione fondamentale da eseguire (dopo l'inizializzazione della variabili, che in questa sede tralasciamo) è la connessione del dispositivo, da fare prima di qualsiasi altra sulla scheda K8055 ed illustrata nella Fig. 5. Come si vede, viene usata una flat sequence per separare due frammenti di codice. Nel primo si genera un ciclo di attesa software (naturalmente temporizzato, per non appesantire troppo il sistema operativo, con un tick di 50 ms), vincolato alla pressione del tasto START. Abbiamo già visto altre volte questa procedura, che usiamo generalmente per fornire all'utente un "entry point" dell'applica-

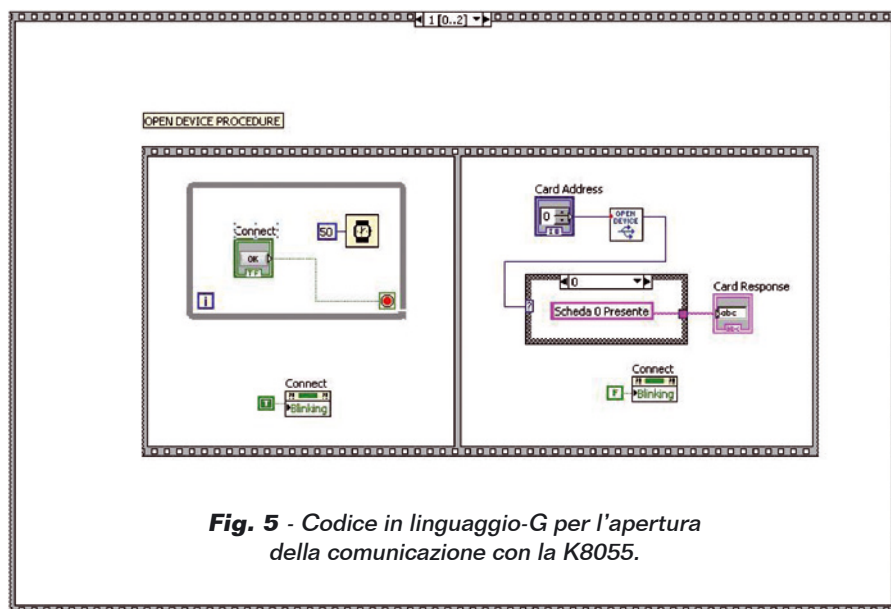


Fig. 5 - Codice in linguaggio-G per l'apertura della comunicazione con la K8055.

zione. In questo caso abbiamo anche aggiunto un property node collegato alla proprietà blinking del tasto START. Inserendo una costante pari a *true* nel primo frame della flat sequence, l'effetto ottenuto sarà il lampeggio del pulsante START (con periodo di circa 1 secondo) finché l'utente non vi farà clic. Fatto ciò, si passa al frame successivo, dove la stessa proprietà viene messa a *false*. Analizziamo ora il frammento successivo; ma prima di andare

avanti bisogna premettere che le K8055 sono dotate di due jumper collegati ad altrettanti piedini del PIC16C745. Questi jumper servono a configurare un indirizzo per le schede, in modo da poter

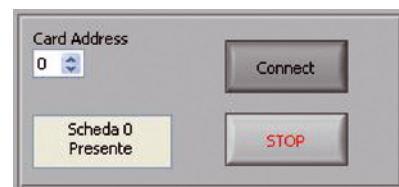
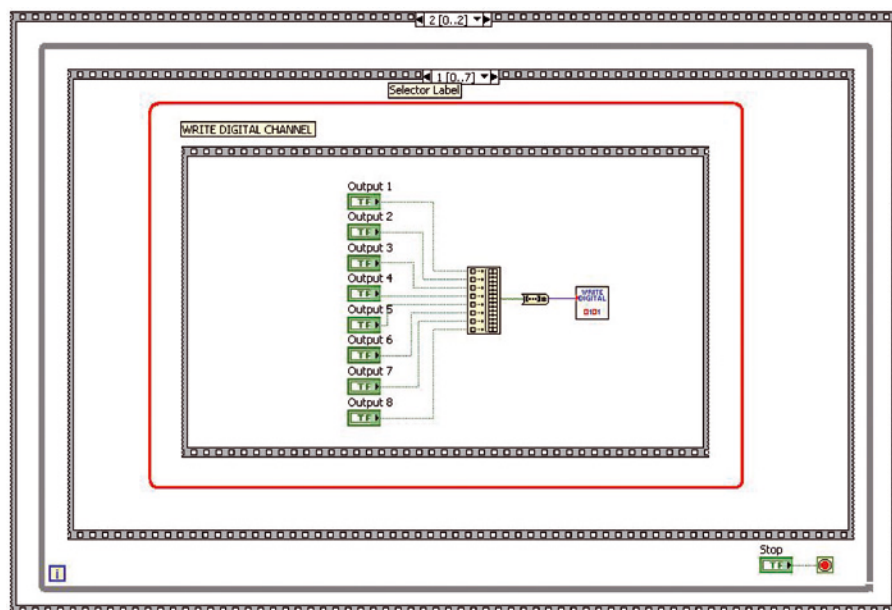


Fig. 6 - Pannello frontale corrispondente alla sezione di apertura dell'applicativo.

Fig. 7 - Scrittura delle linee digitali.



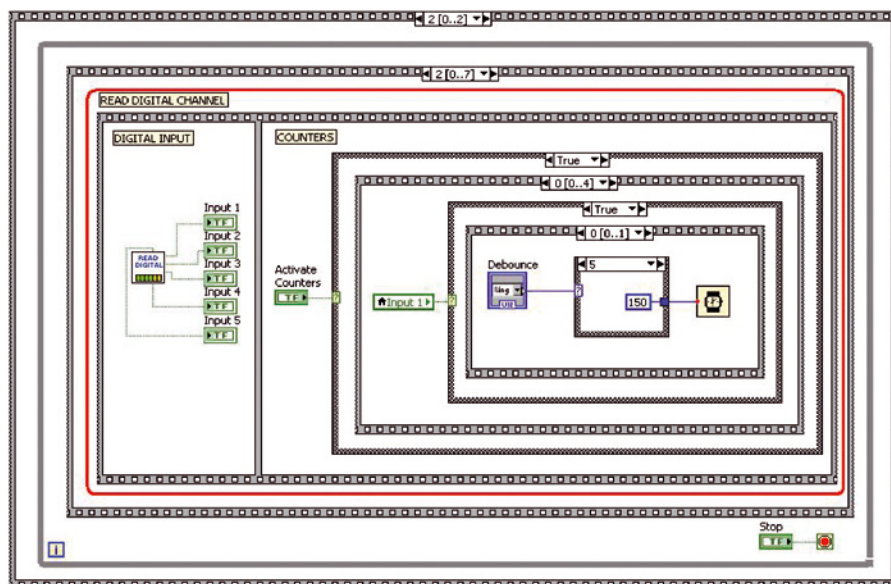


Fig. 8 - Lettura input digitali e gestione dei contatori digitali.

collegare fino a quattro unità sullo stesso bus. Per sfruttare questa proprietà, il subVI "open device" è stato dotato di un input numerico che serve per passare un parametro pari all'indirizzo della scheda (da 0 a 3). Quindi il codice grafico non fa altro che prendere l'input da un controllo di tipo *numeric* e lanciare il subVI "open device", passando tale parametro. Il risultato dell'operazione viene presentato all'utente tramite un indicatore di tipo *stringa*.

In Fig. 6 è riportata la relativa sezione del pannello frontale. Passiamo ora all'analisi dei task successivi. Il primo che viene eseguito dopo l'attivazione della comunicazione è la scrittura delle otto linee digitali; questa operazione viene gestita tramite otto pulsanti ed il subVI "Write Digital". Tale blocco sfrutta la fun-

zione *WriteAllDigital* contenuta nella DLL fornita a corredo della K8055. Il codice è molto semplice: dopo aver raccolto gli input degli 8 pulsanti in un array di booleana, questo viene convertito in una parola di 8-bit e passato al SubVI. In Fig. 7 è riportato il frammento di linguaggio-G che realizza le operazioni descritte. Successivamente si passa alla lettura dei 5 canali digitali ed alla gestione dei relativi contatori. In Fig. 8 è riportato il frammento di codice in linguaggio-G che si occupa di questo task. La lettura degli input digitali è relativamente semplice, dato che il corrispondente subVI, "read Digital", esce direttamente con cinque variabili booleane, che possono essere collegate ad altrettanti LED virtuali sul pannello di controllo. Più complicata è la gestione dei contatori, per la descrizione della

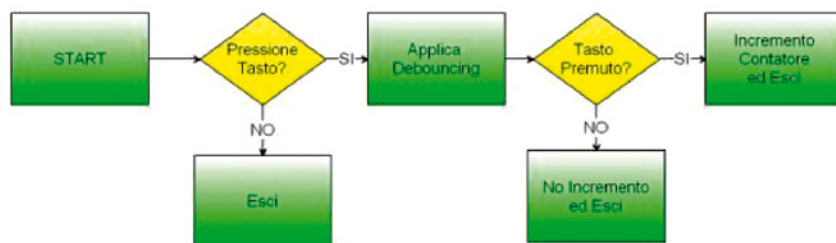


Fig. 9 - Diagramma di flusso del codice di gestione dei contatori digitali.



Fig. 10 - La Digital I/O tab.

quale ci avvaliamo del diagramma di flusso visibile nella Fig. 9. Come si vede, alla variazione di una delle cinque linee digitali di ingresso (nello schema sono stati genericamente indicati i tasti fisici della scheda, utilizzati nella fase di debug, ma possono essere anche dei segnali digitali provenienti da un'altra fonte, collegati alle morsettiere della K8055), si avvia la procedura di debouncing (con debounce time impostabile dall'utente), ed allo scadere del ritardo per l'antirimbombo, si controlla nuovamente la linea; se la variazione persiste si incrementa il contatore, altrimenti si esce senza incrementare. In Fig. 10 è rappresentata la sezione del pannello frontale che si occupa della gestione degli output e degli input digitali (denominata Digital I/O tab). Passiamo adesso alla gestione dei due canali analogici di ingresso: dai dati analogici acquisiti dal convertitore del PIC16C745, vengono generati degli indicatori a barra ed un grafico; inoltre viene gestito un algoritmo che permette di impostare delle soglie di intervento sui valori delle letture analogiche, il cui stato permette di comandare automaticamente

Fig. 11 - Lettura canali analogici.

fino a 5 delle 8 linee digitali di uscita disponibili. Naturalmente le applicazioni sono molteplici: ad esempio potrebbe essere collegato un sensore di luminosità su uno dei due ingressi analogici, in modo da comandare l'apertura o la chiusura delle tapparelle motorizzate di un appartamento. Oppure sarebbe possibile comandare l'accensione o lo spegnimento di un impianto di irrigazione in funzione di determinate condizioni di umidità o di temperatura ambientale.

Vediamo quindi come viene gestita la fase di acquisizione dei dati analogici, con la relativa presentazione dei dati. In Fig. 11 è rappresentato il codice in linguaggio-G che realizza l'acquisizione dei canali analogici e la stampa dei dati sul grafico e sulle bar. Il subVI "Read Analog" è di semplice utilizzo, infatti restituisce direttamente due valori di tipo numeric (u8), che possono essere passati direttamente alle bar (nel frammento di codice i wire passano anche dentro una struttura *case* comandata da un radio button, per la selezione delle scale, tra bit (0÷255) e volt (0÷5). Prima di essere inviati al grafico (un waveform chart) i dati devono essere preventivamente racchiusi in un cluster, usando il blocco bundle (*cluster, class e variant/bundle*), poiché il grafico è di tipo multiplot. Il risultato ottenuto è visibile in Fig. 12 (esempio di dati acquisiti e stampati su grafico) ed in Fig. 13 (bar verticali con indicatori digitali). Il grafico di Fig. 12 è stato dotato delle sue funzioni base per il controllo degli stili dei plot e della graph palette per la gestione degli zoom (questi oggetti possono essere facilmente attivati cliccando con il tasto destro del mouse sull'oggetto grafico e selezionando le relative opzioni

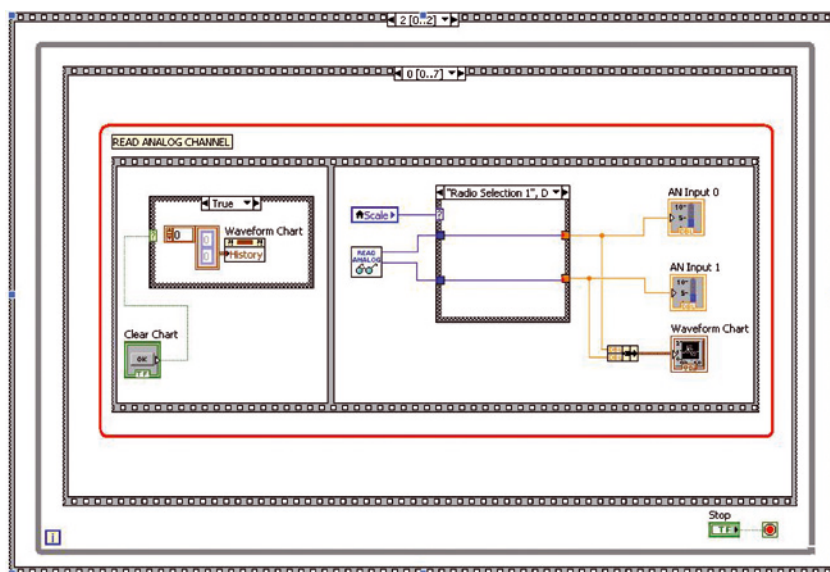
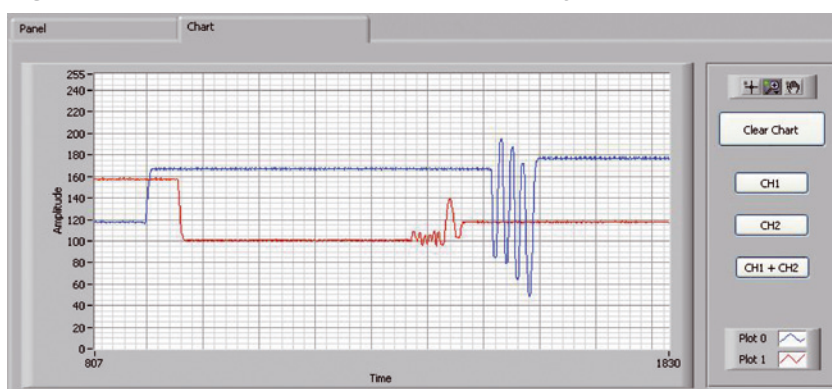


Fig. 12 - Grafico rappresentante i due canali analogici.



sotto *Visibile Items*). Inoltre sono stati aggiunti alcuni altri controlli sulla destra: in particolare, abbiamo dotato il grafico del pulsante "clear" e di una modalità di visualizzazione che permette di visualizzare selettivamente solo una delle due tracce o di generare un multiplot. Come si vede, anche in questo caso, che poteva apparire almeno inizialmente più ostico dei precedenti, la versatilità e l'efficienza di LabVIEW ci vengono in aiuto. Nella Fig. 14 vediamo il codice grafico per la gestione delle soglie di intervento: viene effettuato un confronto tra il canale analogico selezionato e la soglia impostata tramite un indicatore digitale (Threshold 1). L'esito del confronto (positivo o negativo che sia) va a comandare una

struttura di tipo *case*, nella quale, tramite un altro *case*, viene posta a ON o OFF l'uscita digitale indicata dal controllo di tipo ring I/O. Passiamo, infine, alla gestio-

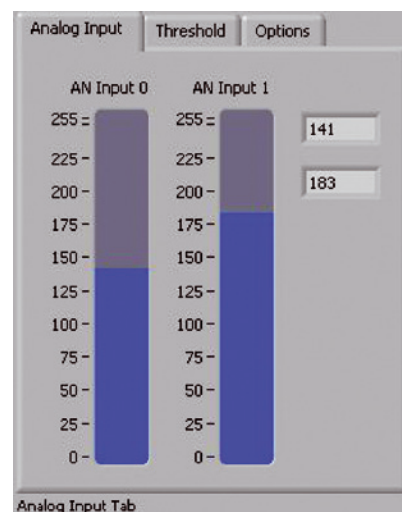


Fig. 13 - Analog input tab.

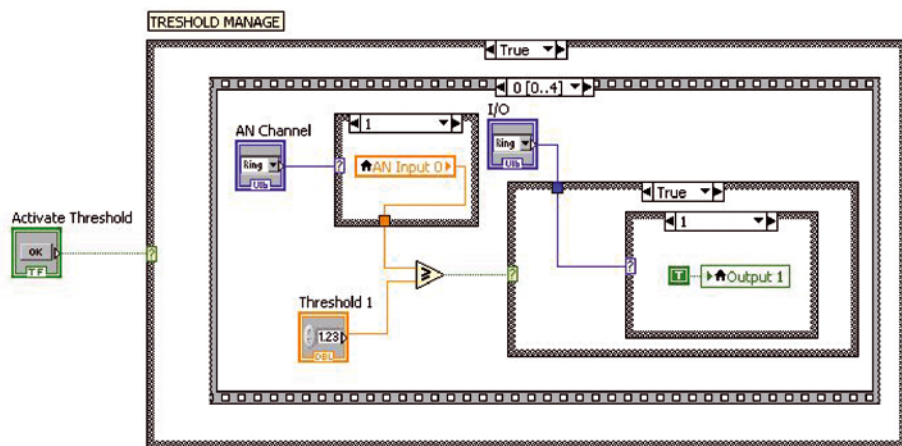


Fig. 14
Gestione soglie di intervento.

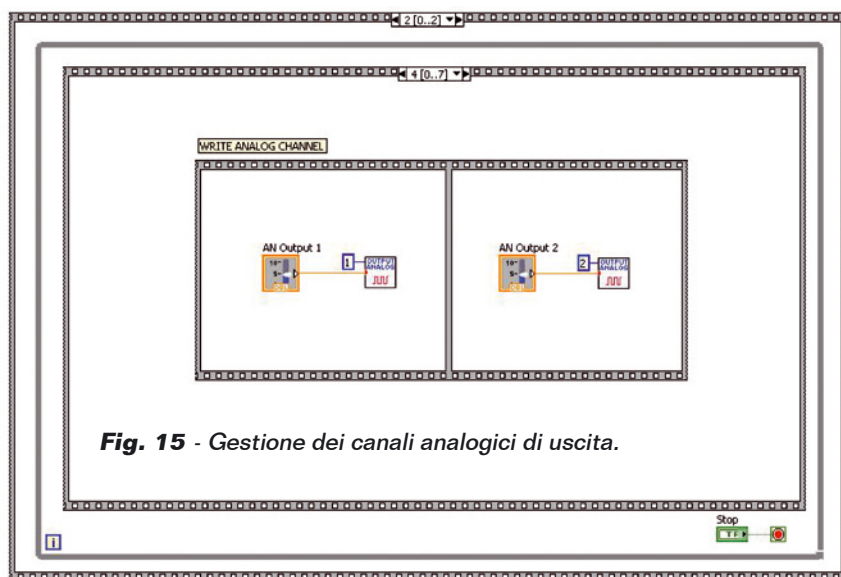


Fig. 15 - Gestione dei canali analogici di uscita.

ne degli output analogici. Come detto in precedenza, la K8055, dispone di due uscite analogiche PWM (per controllo di motori elettrici o di azionamenti) e di due uscite analogiche ricavate filtrando (con un semplice filtro RC del primo ordine ed un voltage follower) i segnali PWM stessi. Quindi i canali sono tra di loro vincolati e non possono essere gestiti separatamente.

Per la gestione degli output analogici è stato creato il blocco "analog output", che ha un funzionamento elementare: prende in ingresso due valori numerici (il primo corrisponde al canale analogico del PWM, mentre il secondo è il valore, rappresentato per mezzo di un byte) del duty-cycle del PWM.

In Fig. 15 è riportato il frammento di linguaggio-G che implementa la gestione dei canali: notate che per il controllo del valore del duty-cycle sono state utilizzate due "vertical pointer slide" (*numeric/vertical pointer slide*). In Fig. 16 è invece riportata la sezione relativa del front panel.

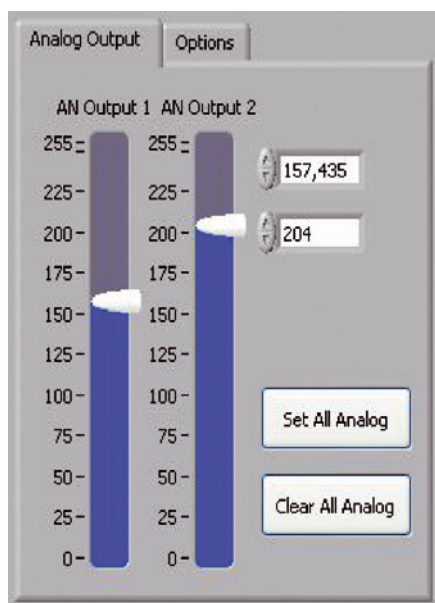


Fig. 16 - Analog output tab.

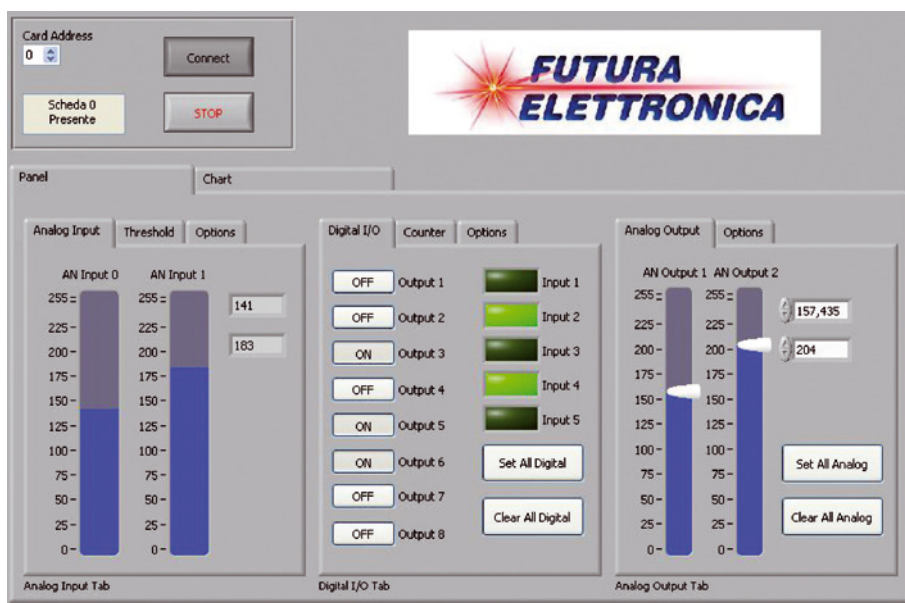


Fig. 17 - Vista complessiva del front panel del software durante l'esecuzione - Tab Panel.

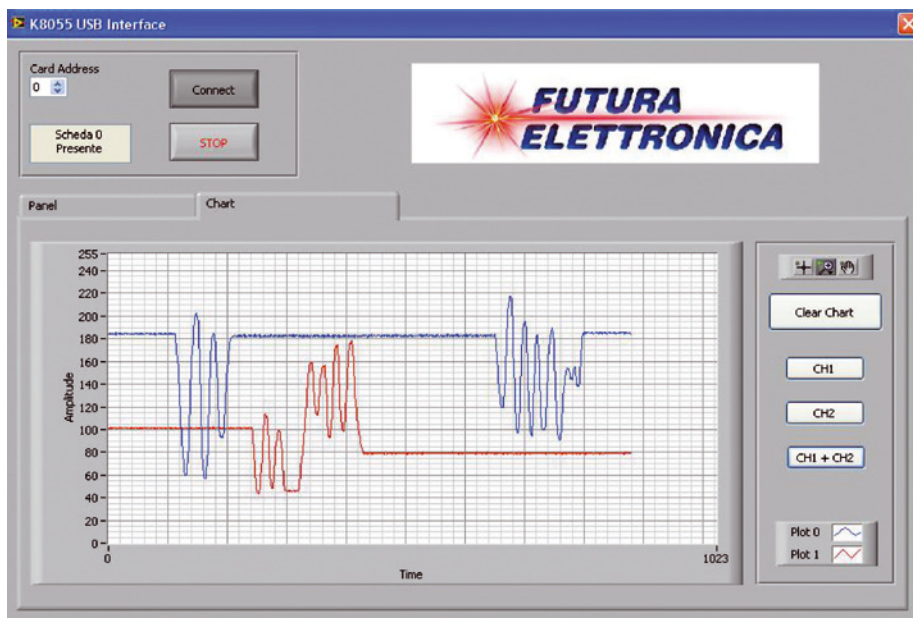


Fig. 18 - Vista complessiva del front panel del software durante l'esecuzione – Tab Chart.

In Fig. 17 vedete un'immagine del software in esecuzione, con aperto il tab Panel, mentre in Fig. 18 trovate la finestra del software con aperto il tab Chart. Come si vede, vengono utilizzati un tab per separare il grafico dal resto dei comandi e degli indicatori, e poi tre tab più piccole per separare la parte di analog input, digital I/O ed analog output.

Concludiamo questa trattazione con uno sguardo al codice nel complesso: come si vede in Fig. 19, che mostra una sezione del frame principale contenente tutto il block diagram, il codice risulta piuttosto compatto; ciò a tutto vantaggio della leggibilità. Invece è abitudine dei programmatori alle prime esperienze in LabVIEW, di sviluppare il codice

in tutte le direzioni disponibili (quindi sia in orizzontale che in verticale); ciò porta alla realizzazione di block diagram di dimensioni titaniche, veri e propri enigmi per i programmatori chiamati a revisionarli o modificarli e comunque a lavorare su programmi sviluppati da altri. Nei casi in cui, per dimensioni dell'applicazione, non sia possibile racchiudere tutto il codice in una singola schermata, è importante cercare di sviluppare il codice grafico in una sola direzione (preferibilmente quella orizzontale, da sinistra a destra).

Anche l'uso delle sequenze è un parametro da tenere in considerazione: nei casi in cui il codice abbia uno sviluppo organico o venga forzata l'esecuzione sequenziale, se ne può fare un buon uso ed incrementare la leggibilità del programma che si sta realizzando, ma bisogna fare attenzione a non abusare dello strumento, che, se usato male, può portare ad un eccessivo frazionamento del codice o ad un decremento delle prestazioni. ■

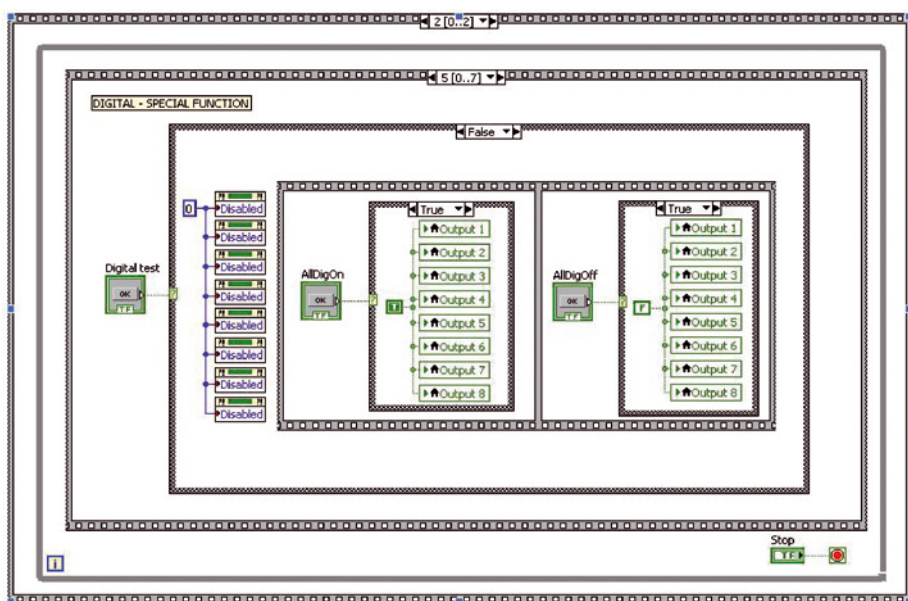
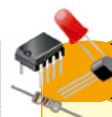


Fig. 19 - Vista di un'altra sezione del block diagram.



per il MATERIALE

Il codice sorgente dell'interfaccia software presentata in questo articolo può essere scaricato dal sito della rivista (www.elettronica.in.it). Il sistema di sviluppo LabVIEW 2010 Student Edition è reperibile presso Futura Elettronica al prezzo di 39,95 Euro (cod. LABVIEWSE) così come la scheda a relé prodotta da Velleman disponibile in kit al prezzo di 38,00 Euro (cod. K8055) e nella versione già montata (cod. VM110) al prezzo di 56,00 Euro.

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

Videoispezioni?

Mai state più portatili di così!



SNAKE CAMERA CON MONITOR LCD

Sistema per videoispezione composto da una telecamera CMOS a colori con tubo orientabile da 84,5 cm e monitor LCD TFT a colori da 2,36".

- Angolo di ripresa 50°;
- Illuminatore con intensità regolabile;
- Dimensioni telecamera Ø17 x 79 mm.

cod. CAMCOLI3
€ 120,00

SNAKE CAMERA USB

Telecamera CMOS a colori, con tubo orientabile da 58 cm, ideale per videoispezioni.

- Angolo di ripresa 54°;
- Messa a fuoco 6 cm;
- Illuminatore integrato (2 LED);
- Alimentazione tramite porta USB (5 Vdc);
- Dimensioni telecamera Ø14 x 55 mm.

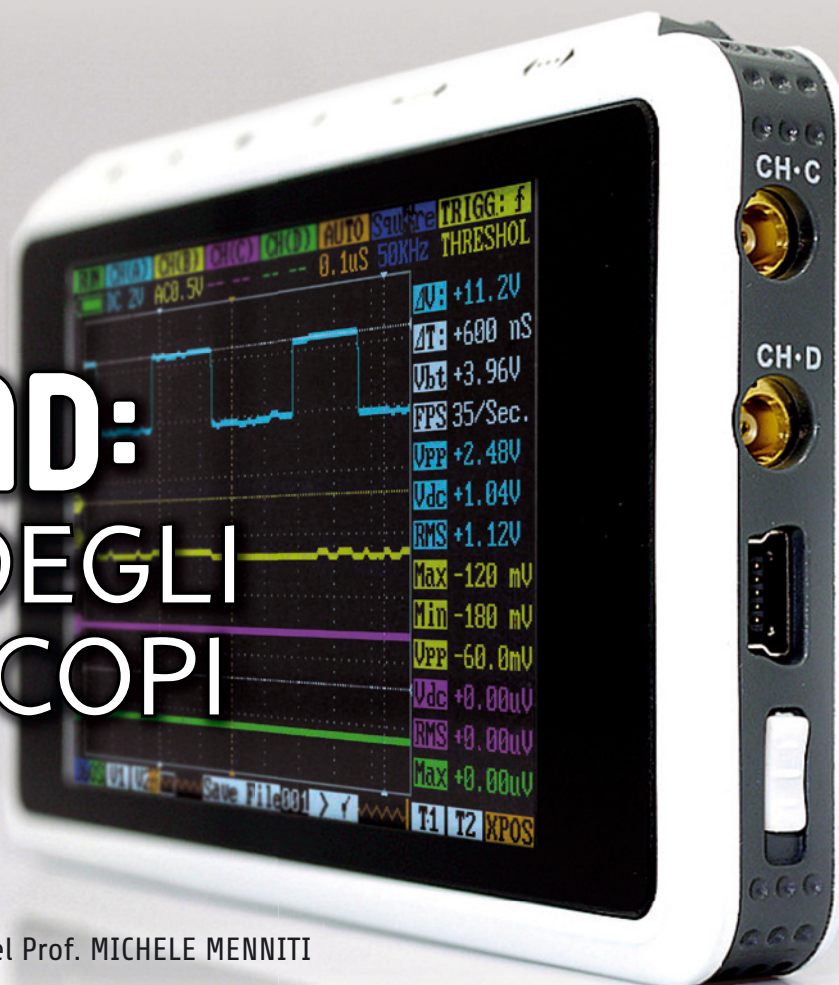
cod. CAMCOLI2
€ 67,00



collegamento
a PC via USB

Prezzi IVA inclusa.

DSO QUAD: L'iPhone DEGLI OSCILLOSCOPI DIGITALI



..... del Prof. MICHELE MENNITI

Il primo problema nasce subito e fa venire il sospetto che sia stato studiato ad arte dal produttore, la cinese SEEED Studio, per far mettere all'utente, tra i preferiti del proprio browser, la pagina web www.seeedstudio.com/depot/dso-quad-4-channel-digital-storage-oscilloscope-p-736.html?cPath=174.

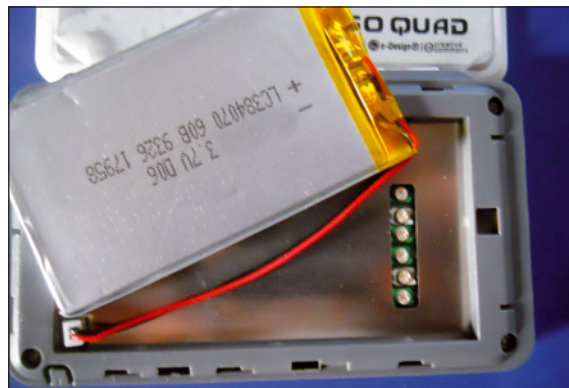


Fig. 1 - Contenuto della confezione.

Fig. 2 - Apertura del cover.



Fig. 3 - Collegamento della batteria.



Infatti ci si trova la batteria tra le mani e nessuna indicazione su come si debba installare; dopo averle pensate tutte, convinti (a buona ragione) di poter guastare lo strumento ancor prima di iniziare ad usarlo, siamo andati sul sito, abbiamo rapidamente localizzato la pagina del DSO QUAD e puntato, in essa, i link

•---NEW! Download manual here--- e •How to connect battery to you DSO Quad.

Quest'ultimo contiene proprio le istruzioni per montare la batteria, operazione che, in definitiva, è piuttosto semplice.

In pratica bisogna girare il DSO QUAD per vedere il cover posteriore con in basso a destra la scritta "DSO QUAD" e poi spingere il cover stesso da destra verso sinistra, proprio come un normale cellulare con chiusura a slitta (Fig. 2); tolto il coperchio

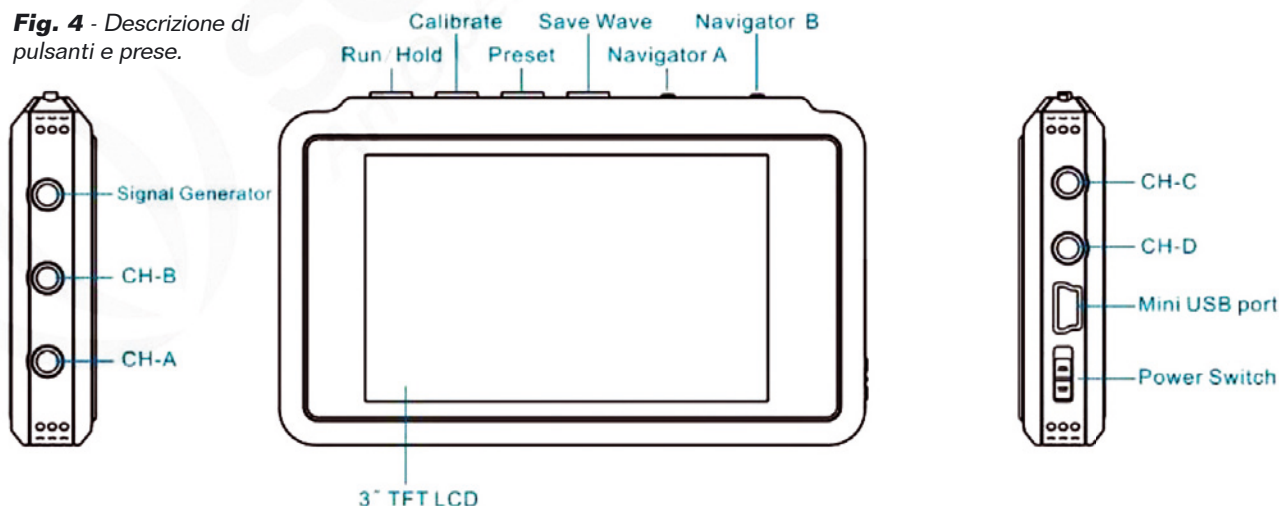
bisogna collegare la batteria al connettore tipo JST visibile in basso a sinistra. Nonostante la connessione sia obbligata dalla guida del connettore, più di qualcuno deve aver collegato la batteria all'inverso: ecco perché la SEEED Studio si è presa la briga di pubblicare diverse foto esplicative... Comunque sia, il connettore ha un verso d'inserzione obbligato: nella versione da noi provata (e ad oggi non ci risultano cambiamenti) il polo positivo (filo rosso) va in basso, verso il bordo inferiore del case; il corretto inserimento sarà confermato da un "clic" che si percepirà appena inserito il connettore (Fig. 3).

La batteria si ricarica tramite USB, prendendo la corrente dal PC. Prima di richiudere il cover (da maneggiare con cura perché è facile farlo uscire dalle guide) sarà

bene notare, sul fondo dell'alloggiamento della batteria, la fila di sei compensatori che, in base alle prime misurazioni, potrebbero servire per la taratura delle tracce dei due canali analogici (A e B) e delle sonde 10:1.

Bene, se avete dato uno sguardo attento alla caratteristiche tecniche (pagina accanto) vi renderete conto di quante qualità sono state racchiuse in così poco spazio. E le buone notizie non sono ancora terminate, perché il DSO QUAD è un prodotto open-source: SEEED Studio ha deciso di seguire la strada che ha intrapreso quando ha "sfruttato" questa filosofia per creare i propri Arduino-compatibili. Visto il successo, ha pensato bene di mettere questo gioiello a disposizione della comunità internazionale, che non si è fatta attendere: infatti il firmware è periodicamente aggiornato e le

Fig. 4 - Descrizione di pulsanti e prese.



applicazioni sono state potenziate con nuove funzionalità. Il link Internet www.seeed-studio.com/forum/viewtopic.php?f=22&t=1929&start=90 è un Topic riguardante proprio gli aggiornamenti.

USARE IL DSO QUAD

Naturalmente abbiamo provato in campo lo strumento e di seguito vi esponiamo i principali comandi e le impostazioni, affinché possiate valutare l'utilizzabilità pratica dello strumento.

Tutte le funzionalità del DSO QUAD sono gestibili mediante una serie di pulsanti posti sul bordo superiore dello strumento (Fig. 4).

Il fatto che alcuni pulsanti funzionino normalmente a pressione mentre due di essi operino in tre direzioni (sinistra, destra e in basso) e svolgano svariate funzioni, i primi tempi creerà un po' di confusione, ma una volta fatta l'abitudine vi accorgete che i comandi sono piuttosto intuitivi. Guardando il DSO QUAD di fronte e partendo da sinistra, i pulsanti si chiamano rispettivamente: *Run/Hold*, *Calibrate*, *Preset*, *Save Wave*, *Navigator A*, *Navigator B*. Sul lato sinistro, dall'alto verso il basso, si vedono: wave out (uscita del generatore di forme d'onda integrato) e CH-B e CH-A (ingressi dei due canali analogici). Sul lato destro, sempre dall'alto verso il basso, troviamo: CH-C e CH-D (ingressi dei due canali digitali), presa USB (collegamento al PC e caricabatteria) e interruttore d'accensione. Di seguito, vediamo le funzionalità di base della pulsantiera.

- **Run/Hold (▶ ||)**: ad ogni pressione del tasto si passa dal fermo immagine alla modalità normale e viceversa.
- **Calibrate (■)**: premuto per 2

CARATTERISTICHE TECNICHE

- **Leggerissimo**: pesa 99 g con la batteria.
- **Display LCD** da 3" a colori con risoluzione 400x240.
- **Quattro canali**, di cui due analogici a 72MS/s (CH-A e CH-B) e due digitali (CH-C e CH-D); 144 MS/s con la configurazione a singolo canale.
- **Firmware aggiornabile ed open-source**.
- **Possibilità di memorizzare e rivedere le forme d'onda**.
- **Ben quattro locazioni programmabili per le applicazioni degli utenti**.
- **Larghezza di Banda dei canali analogici**: >5MHz (*).
- **Sample Rate** (massimo tasso di campionamento): da 1kS/s a 144MS/s, 8 bit.
- **Sample memory depth** (profondità della memoria di campionamento): 4.096 punti per ciascun canale.
- **Scala verticale**: da 50mV/div a 10V/div (con sonda 1:1) in otto portate.
- **Risoluzione verticale**: 8 bit.
- **Base dei tempi**: da 0,1 µs a 1 s/div in ventidue portate.
- **Regolazione posizione orizzontale e verticale di ogni singola traccia**.
- **Retroilluminazione e volume audio regolabili**.
- **Accoppiamento d'ingresso**: AC/DC.
- **Impedenza d'ingresso**: >800 kΩ.
- **Max segnale di ingresso**: 80 Vpp con sonda 1:1.
- **Tipi di Trigger**: Auto, Normale, Singolo, Scan, Nessuno.
- **Sorgenti di Trigger**: CH-A, CH-B, Esterno.
- **Trigger**: <TH, >TH, <TL, >TL, v (Falling), ^ (Rising), <Vt, >Vt.
- **Generatore interno di forme d'onda a frequenza fissa**:
 - quadre da 10 Hz a 8 MHz in 20 portate (è prevista l'implementazione della regolazione del duty-cycle dall'1 al 99 %);
 - sinusoidali da 10 Hz a 20 kHz in 11 portate;
 - triangolari da 10 Hz a 20 kHz in 11 portate;
 - a dente di sega da 10 Hz a 20 kHz in 11 portate.
- **Memorizzazione**: memoria USB interna da 2 Mb per file .BMP e .DAT.
- **Misurazioni automatiche**: Vbt (tensione batteria), FPS (frame/secondo), Vpp (tensione picco-picco), Vdc (tensione continua), RMS (tensione RMS), Max (tensione del picco massimo), Min (tensione del picco minimo), FRQ (frequenza), CIR (periodo), DUT (duty cycle), TH e TL (periodo HIGH e LOW dell'impulso visualizzato);
- **Misurazioni manuali**: ΔV (intervallo di tensione tra due punti preimpostabili V1 e V2) e ΔT (intervallo di periodo tra due punti preimpostabili T1 e T2);
- **Modalità di campionamento**: in tempo reale;
- **Alimentazione**: tramite batteria LiPo integrata;
- **Dimensioni esterne**: 98x60x14,5 mm.

(*) SEEED Studio dichiara che l'hardware è progettato per arrivare a 12 MHz, valore che abbiamo verificato in campo.

secondi dà accesso alla modalità di calibrazione dello strumento.

- **Preset (●)**: la semplice pressione di questo pulsante permette di memorizzare tutte le impostazioni attuali del DSO QUAD, comodissima da usare quando si arriva a determinare qual è la migliore combinazione di tutte le funzionalità del menu.
- **Save Wave (▲)**: premendo

questo pulsante è possibile memorizzare la schermata attuale nella memoria USB interna, con possibilità di richiamarla successivamente o trasferirla sul PC.

- **Navigator A**: viene utilizzato per selezionare o cambiare il sottomenu del menu attivo; la pressione del pulsante serve per lo spostamento tra i sottomenu (l'operazione di pressione d'ora in poi sarà definita

Tabella 1 - Il Gruppo di Menu 1.

con ▼), mentre lo spostamento sinistra/destra (-...+) serve per variare i valori del sottomenu attivo. Per esempio, se il menu attivo (lampeggiante) è "CH(A)", con lo spostamento -/+ tale valore viene cambiato in "HIDE", cioè il canale da attivo passa a nascosto e non si vede più la traccia. Se invece in posizione "CH(A)" il pulsante viene premuto, si seleziona il sottomenu "DC" (o "AC") e con lo spostamento -/+ si alternano i due valori. Premendo ancora il pulsante, si seleziona il sottomenu relativo all'ampiezza del canale (volt/divisione) che con lo spostamento -/+ si può impostare da 50mV/div a 10V/div in otto passaggi; questo pulsante servirà anche nelle operazioni di calibrazione.

- **Navigator B:** questo pulsante serve per spostarsi tra i vari menu dello strumento; tali menu sono divisi in due gruppi: premendo il pulsante (▼) si passa dal Gruppo 1 al Gruppo 2 e viceversa; con lo spostamento sinistra/destra (<...>), invece, ci si sposta tra i menu appartenenti ad uno stesso Gruppo; anche questo pulsante, come il precedente, servirà nelle operazioni di calibrazione.

GRUPPI DI MENU

Le voci di menu del DSO QUAD sono organizzate in due gruppi,

denominati 1 e 2. Il Gruppo 1 è costituito dalle voci di scelta delle misurazioni da effettuare ed è riepilogato nella **Tabella 1**.

Per spostarsi tra i menu del Gruppo 1 si usa il pulsante Navigator B (</>); invece, per entrare nei sottomenu relativi e spostarsi tra di essi lo stesso pulsante va premuto (▼). Il Gruppo 2, invece, serve per mostrare i risultati delle misure, ed è costituito da tutte le voci programmabili, localizzate sulla destra del display, a partire da quella immediatamente successiva a "Vol" (campo di regolazione del Volume); in **Fig. 5** vedete una possibile impostazione dei valori. Per spostarsi tra i campi del Gruppo 2 occorre usare il solito pulsante Navigator B (</>). Ad ognuno dei campi del Gruppo 2, semplicemente usando il pulsante Navigator A (-/+), può essere assegnata una tipologia di misura a scelta tra Vbt (tensione batteria), FPS (frame/secondo), Vpp (tensione picco-picco), Vdc (tensione continua), RMS (tensione RMS), Max (tensione del picco massimo), Min (tensione del picco minimo), FRQ (frequenza), CIR (periodo), DUT (duty cycle), TH o TL (periodo HIGH e LOW dell'impulso visualizzato). I valori Vbt e FPS sono comuni a tutte le

Gruppo Menu 1	Opzioni
CHA Control	CH(A) / HIDE
CHB Control	CH(B) / HIDE
CHC Control	CH(C) / HIDE
CHD Control	CH(D) / HIDE / (A+B) / (A-B) / (C&D) / (C D) / REC_A / REC_B / REC_C / REC_D
Sync. Mode	AUTO / NORM / SINGL / NONE / SCAN
Output Mode	Sine / Triang / Saw / Square
Trigg	Modalità di Trigger Applicata al canale selezionato, di cui lo sfondo assume il colore
ΔV	Diff. tra V1 & V2 in volt Calcolata sul canale selezionato, di cui lo sfondo assume il colore
ΔT	Diff. tra T1 & T2 in ms Calcolata sul canale selezionato, di cui lo sfondo assume il colore
Storage Menu	È attivato dall'apposito pulsante ed è visibile solo quando è selezionato.
B.L.	Regol. retroilluminazione: 10%+100%
Vol	Regol. Volume audio: 10%+100%

impostazioni e sono mostrati su sfondo bianco; tutti gli altri, invece, si riferiscono ad un canale, in base al colore di sfondo che hanno (CH-A=ciano; CH-B=giallo; CH-C=magenta e CH-D=verde). A titolo di esempio descriviamo, dall'alto verso il basso, i valori impostati visibili in **Fig. 5**:

- Vbt (prima voce che si incontra) ossia tensione della batteria (sfondo bianco);
- Frq (frequenza del segnale), Vdc (valore di tensione continua) e Dut (Duty-Cycle) del CH-A (ciano);
- Frq (frequenza del segnale), Vdc (valore di tensione continua), Dut (Duty-Cycle) del CH-B (giallo);
- Frq = frequenza del segnale del canale CH-C (magenta);
- Frq = Frequenza del segnale del canale CH-D (verde).

Come già detto, queste nove voci sono completamente personalizzabili: ad esempio, per effettuare ogni misura possibile del solo segnale CH-A si possono tranquillamente selezionare tutti i tipi di misurazioni assegnandole al colore di questo canale (ciano).

CANALI E IMPOSTAZIONI

CH(A) e CH(B):

Servono per gestire i due canali analogici; si possono nascondere



Fig. 5 - Il Gruppo di Menu 2.

i relativi segnali impostandoli su HIDE.

CH(C):

Gestisce il primo canale digitale ed anch'esso può essere settato su HIDE, se non serve.

CH(D):

Questo canale ha, in aggiunta alla normale funzionalità, la particolarità di mostrare il risultato di operazioni eseguite sui segnali in ingresso delle quattro tracce; infatti è dotato delle seguenti funzioni:

- (A+B); la traccia D mostra la somma tra le tracce A e B;
- (A-B); la traccia D mostra la differenza tra le tracce A e B;
- (C&D); la traccia D mostra l'AND tra le tracce C e D;
- (C | D); la traccia D mostra l'OR tra le tracce C e D;
- REC_A (opp. B/C/D): quando si richiama una schermata precedentemente memorizzata, la traccia D mostra la traccia selezionata;
- HIDE; in questo caso la traccia D viene nascosta.

Modalità Sincronismo

Permette di selezionare il metodo di sincronizzazione, che può assumere i seguenti valori:

- **AUTO**=visualizzazione continua, dove la sincronizzazione avviene quando il segnale è triggerato; questa impostazione è utile per l'uso generale;
- **NORM**=il segnale viene visualizzato solo quando è triggerato, altrimenti il display resta vuoto; utile per rilevare segnali periodici;
- **SINGL**=il segnale viene visualizzato solo quando è triggerato e la lettura va in auto-hold; utilissima per la rilevazioni di segnali pulsanti sporadici;
- **NONE**=visualizzazione continua non sincronizzata del segnale (viene ignorato il trigger); si usa per segnali non sincronizzabili;

- **SCAN**: visualizzazione continua del segnale da sinistra a destra sullo schermo; si usa nelle operazioni di monitoraggio di un segnale;
- **FFT**: (Fast Fourier Transform), serve per calcolare la Trasformata di Fourier veloce; da notare che questa funzione è in fase di implementazione, ma la relativa voce di menu è già presente

Generatore di Frequenze

Il DSO QUAD è dotato di un buon Generatore di Frequenze interno, in grado di fornire, sul relativo connettore, segnali ad onda quadra (20 valori fissi da 10 Hz a 8 MHz), sinusoidale, triangolare e dente di sega, con frequenze tra 10 Hz e 20 kHz (11 valori fissi).

Trigger

Numerose sono le modalità di trigger, delle quali è dotato lo strumento; eccole riepilogate:

- **<TL**: livello LOW dell'impulso con larghezza inferiore al ΔT ;
- **>TL**: livello LOW dell'impulso con larghezza superiore al ΔT ;
- **<TH**: livello HIGH dell'impulso con larghezza inferiore al ΔT ;
- **>TH**: livello HIGH dell'impulso con larghezza superiore al ΔT ;
- **v** (Falling): fronte di discesa della soglia;
- **^** (Rising): fronte di salita della soglia;
- **<Vt**: tensione inferiore alla soglia;
- **>Vt**: tensione superiore alla soglia.

In **Fig. 6** è possibile vedere tutte le linee di riferimento del DSO QUAD: la soglia di trigger, orizzontale (di colore giallo in questo caso) al centro del display, la XPOS verticale (di colore arancione), sempre al centro del display,



poi le due linee bianche orizzontali V1 e V2 e le due linee bianche verticali T1 e T2.

Misura Intervallo Tensione

Due coppie di riferimenti manuali, costituite dalle linee appena elencate, permettono di misurare intervalli di Tensione (ΔV) o di Tempo (ΔT).

Memorizzazione

Il DSO QUAD è dotato di una memoria USB da 2 Mb, visibile su un PC mediante il semplice collegamento con un cavo USB, e quindi accessibile come una comune pen drive.

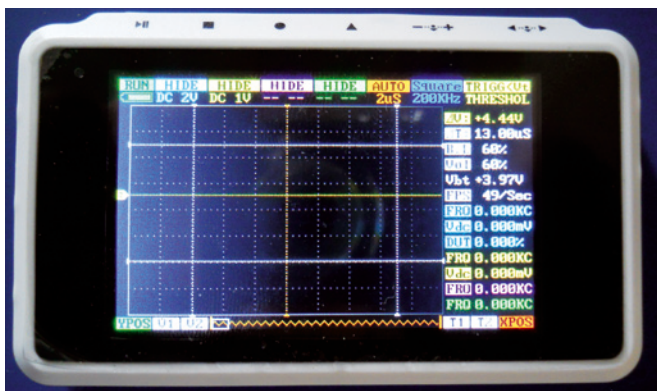
È possibile salvare file nei seguenti formati:

- **.DAT**: effettua una cattura della schermata attuale;
- **.BUF**: permette la memorizzazione separata delle quattro tracce;
- **.CSV**: salva la schermata, sotto forma di un elenco di 4.096 punti (12 bit) in formato compatibile con Excel;
- **.BMP**: salva la schermata sotto forma di immagine, richiamabile con un qualsiasi programma di grafica dopo il trasferimento sul PC.

Retroilluminazione

Il DSO QUAD è dotato di un efficiente sistema di retroilluminazione, regolabile dal 10 al 100 % in dieci passi; naturalmente la regolazione va fatta per ottenere

Fig. 6 - Le linee di riferimento del DSO QUAD.



la migliore visibilità possibile in base alle condizioni di luce in cui è usato il DSO QUAD, tenendo conto del fatto che maggiore è questo valore, maggiore sarà il consumo della batteria durante l'uso dello strumento.

Volume audio

La voce **Vol** indica il volume audio dello strumento, anche in questo caso regolabile in dieci passi dal 10 al 100 %.

SONDE E LETTURA DEI SEGNALE

Il DSO QUAD viene fornito con due sonde 1:1 più due semplici sonde che terminano con altrettanti gancetti di colore rosso (segnale) e nero (massa); queste ultime potrebbero non fare un'ottima presa sullo spinotto MCX, quindi bisogna inserirle con attenzione e poi usare abbastanza forza per estrarle. Per ottenere le migliori prestazioni potrebbe essere necessario calibrare le sonde; la verifica della calibrazione è peraltro raccomandata dal costruttore. In realtà, nei DSO QUAD testati in laboratorio tale operazione non è risultata necessaria, nemmeno dopo l'aggiornamento del firmware.

Per la calibrazione della sonda è prevista una semplice prova descritta nel manuale d'uso. Nel compiere le misure è consigliabile nascondere, con la funzione **HIDE**, le tracce inutilizzate: il display è da 3", visibile e chiaro,

ma è meglio evitare di avere linee inutili.

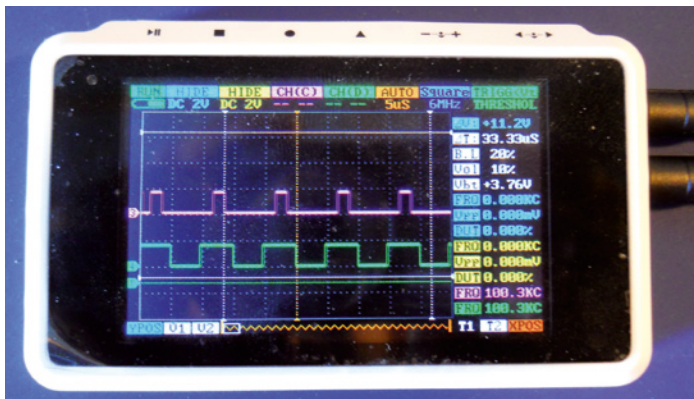
In tutte le prove effettuate, il DSO QUAD ha risposto sempre benissimo: ad esempio, quando gli abbiamo applicato quattro diversi segnali in contemporanea. Anche la visualizzazione dei segnali sopra i 5 MHz massimi dichiarati è stata soddisfacente. Non da meno è stato il generatore di funzioni incorporato.

Prima di concludere il discorso sulle prove, bisogna dedicare qualche riga al funzionamento delle due tracce digitali, cioè **C** (magenta) e **D** (verde); esse permettono di mostrare esclusivamente segnali di tipo digitale (onde quadre, per capirci...) dato che i loro ingressi sono dotati di un circuito squadratore, che, nei limiti del possibile, "digitalizza" i segnali in arrivo.

Il segnale fornito a questi ingressi deve avere un'ampiezza minima di 2 Vpp e viene "trattato" e mostrato come onda quadra, quindi se è analogico viene squadrato ed appare rettangolare. In **Fig. 7** le tracce **C** (magenta) e **D** (verde) stanno ricevendo in ingresso, rispettivamente, un segnale sinusoidale ed un segnale ad onda quadra con frequenza di 100 kHz ed ampiezza di 2 Vpp.

Non esiste, allo stato attuale, la possibilità di regolare i V/div per queste due tracce; quando esse vengono usate, i segnali corrispondenti occupano un solo qua-

Fig. 7 - Segnali da 100 kHz sulle tracce **C** e **D**.



dratino della griglia di riferimento. La frequenza massima alla quale i segnali si vedono correttamente è circa 300 kHz.

FIRMWARE: VERSIONI E AGGIORNAMENTI

Chiudiamo con l'aggiornamento del firmware: come qualsiasi strumento di fascia medio-alta, il DSO Quad prevede la possibilità di aggiornare il firmware; riteniamo che questa funzione verrà usata piuttosto spesso dagli utenti, visto che, come detto all'inizio, il firmware, per volere di SEEED Studio, è open-source.

Dalla comparsa del DSO QUAD sul mercato, il firmware e le varie applicazioni sono stati aggiornati diverse volte, con la correzione di alcuni bug, ma soprattutto con l'implementazione di molte nuove funzioni.

Durante le nostre prove abbiamo avuto modo di aggiornare il firmware dello strumento per ben due volte, arrivando alla versione 2.52; altre novità sono state annunciate e vedranno la luce a breve.

L'operazione di aggiornamento del firmware è molto semplice, ma deve essere effettuata con le dovute cautele, altrimenti si potrebbe facilmente rendere lo strumento inservibile.

Lo strumento può essere acquistato presso Futura Elettronica (www.futurashop.it) al prezzo di 289,00 Euro IVA compresa (codice DSOQUAD).

Al via l'In-Orbit Test di Galileo

È tempo di esami per Galileo. Per i primi due satelliti del sistema di navigazione "made in EU" si sono aperte le operazioni di sperimentazione dei payload di navigazione. Dopo la fase di lancio e messa in orbita (LEOP), il 3 novembre scorso il controllo dei satelliti è passato dal CNES di Tolosa in Francia, al Centro di Controllo Galileo a Oberpfaffenhofen in Germania. Quest'ultimo è gestito dal Centro Aerospaziale Tedesco DLR e avrà il comando e il controllo dei satelliti durante i 12 anni della loro vita operativa.

I primi due satelliti della costellazione Galileo sono stati lanciati grazie al vettore Soyuz il 21 ottobre scorso, dallo spazioporto di Kourou, nella Guyana Francese. Tre ore e 49 minuti dopo il lancio, la Fregat-MT li ha portati ad uno stadio superiore della loro orbita prevista, a 23.222 km, dove sono stati rilasciati contemporaneamente. Successivamente un team congiunto CNES e ESOC dell'ESA si è attivato per iniziare a rendere operativi i due satelliti. I primi segnali sono stati rilevati contemporaneamente e questo ha confermato il loro buon funzionamento.

Come dei paracadutisti che saltano giù da un aereo, i due satelliti sono stati lasciati cadere nello spazio. Una volta che le ruote di reazione sono state pronte a sorreggerli, 70 minuti dopo la fase di separazione, hanno cercato il sole per ricaricare le batterie, grazie alla presenza dei pannelli solari. È stata Fregat a trasportarli duran-

te tutta la loro corsa verso lo spazio. Si è proceduto quindi a testare i propulsori. La coppia di satelliti ha dovuto infatti manovrare in solitudine gli ultimi 100 km delle orbite previste e lo ha fatto con successo. In seguito i satelliti si sono stabiliti nella loro normale modalità: con i pannelli solari rivolti verso il sole e l'antenna di navigazione puntata verso la terra. Raggiunta questa fase il Centro di Tolosa ha ordinato che le orbite subissero un'inclinazione di 56 gradi rispetto all'equatore.

Il passaggio a Oberpfaffenhofen, avvenuto subito dopo il LEOP, è stato completato alle 22:00 CET del 3 novembre scorso. Dei controllori a terra hanno iniziato la messa in piattaforma, verificando che tutti i sottosistemi principali funzionassero come previsto. Nei prossimi giorni si procederà all'accensione del payload di navigazione e questo segnerà l'inizio dell'In-orbit Test di Galileo. Il controllo rigoroso dei segnali di navigazione viene condotto dalla stazione di terra dell'ESA a Redu, in Belgio. In particolare un'antenna, con un diametro di 20 m, misurerà la forma precisa dei segnali di navigazione con un grado molto elevato di accuratezza. Una volta effettuato

il check-out al payload di navigazione, un secondo Centro di Controllo Galileo, quello di Fucino in Italia, gestito dalla società Telespazio, supervisionerà tutti i servizi di navigazione. I soggetti che partecipano alle procedure di testing - ESA-ESOC, CNES, DLR e Telespazio - collaborano con Spaceopal, una consociata di DLR e Telespazio, che segue tutte le attività di Galileo.

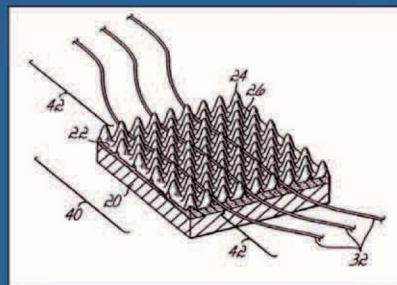
www.esa.eu



L'elettronica indossabile ha il suo filo conduttore

Uno studio dell'Istituto di nanoscienze del Cnr e dell'Università di Bologna ha permesso di realizzare transistor da un semplice filo di cotone. Annodati o tessuti, potranno consentire di avere tessuti "intelligenti" per monitorare battito cardiaco o temperatura corporea, utili a lavoratori a rischio, pazienti e atleti. Tessuti "intelligenti" che permetteranno di ascoltare musica o monitorare il battito cardiaco, ma allo stesso tempo semplici da realizzare e comodi da indossare. I ricercatori Annalisa Bonfiglio e Giorgio Mattana (Istituto nanoscienze del Consiglio nazionale delle ricerche e Università di Cagliari) e Beatrice Fraboni (Università di Bologna) hanno dimostrato che da un semplice filo di cotone si possono creare dei transistor, aprendo la strada alla creazione di indumenti "intelligenti" e "sensibili" e alla possibilità di creare interi

circuiti in tessuto. Lo studio è stato pubblicato sulla rivista internazionale "Organic Electronics". Il cotone, ideale per realizzare capi "intelligenti" grazie alle sue proprietà di flessibilità, economicità e comodità, presenta il limite di non essere conduttivo. "Col nostro lavoro di ricerca siamo riusciti a ottenere fibre conduttive da semplici fili di



cotone, preservando comfort ed elasticità tipici di tale tessuto", prosegue Bonfiglio. "Abbiamo raggiunto questo obiettivo sviluppando una tecnica innovativa di rivestimento dei fili di cotone con un finissimo strato di nanoparticelle d'oro e di polimeri conduttivi e semiconduttivi", spiega Beatrice Fraboni. Le potenzialità applicative sono molteplici. "Questi transistor miglioreranno la qualità e la realizzazione dei biosensori indossabili", conclude Annalisa Bonfiglio. "In futuro potremo avere tessuti 'intelligenti' semplici e comodi, in grado di monitorare il battito cardiaco o rilevare la temperatura corporea, utili alla realizzazione di capi per i lavoratori a rischio, per monitorare i pazienti o per atleti che praticano sport estremi".

www.cnr.it

Lancio riuscito: Curiosity e Leonardo partiti per Marte

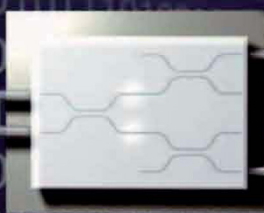
Tutto secondo le previsioni: il vettore Atlas V-541, è stato lanciato alle 16:02 (ora italiana) del 26 novembre, dalla base dell'Aeronautica degli Stati Uniti di Cape Canaveral, dando inizio alla missione Mars Science Laboratory (MSL). A bordo il preziosissimo carico: Curiosity, il rover targato NASA, e un microchip con la scansione del "Codice del Volo" di Leonardo Da Vinci. Curiosity ha le dimensioni di un'automobile – è lungo circa 3 metri e pesa 800 chilogrammi – ed è un gigante rispetto ai suoi due "fratelli minori", gli altri due rover già attivi sul pianeta rosso: Spirit e Opportunity pesano infatti poco più di 170 chili e non raggiungono il metro e mezzo. I suoi strumenti analizzeranno se vi sono state le condizioni favorevoli per lo sviluppo di forme di vita microbiche che hanno consentito la permanenza di loro tracce nelle rocce del Pianeta Rosso.

MSL potrà raccogliere campioni, frantumarli e analizzarli in una camera di analisi chimica che porterà a bordo. Una serie di strumenti scientifici, poi, gli permetterà di identificare materiali organici come proteine, aminoacidi e altri componenti della vita basata sul carbonio. Il laboratorio scientifico spaziale, inoltre, potrà identificare gas connessi ad attività biologiche. Curiosity volerà a bordo del vettore ed effettuerà il suo viaggio in compagnia di un testo simbolico, considerato il fondamento della storia del volo: il "Codice del Volo" di Leonardo da Vinci. Un microchip, collocato all'interno del rover,

contiene infatti la copia scannerizzata del prezioso documento e dell'autoritratto di Leonardo. Il "Codice del Volo" è il quaderno, risalente al 1505 e custodito alla Biblioteca Reale di Torino, su cui il genio toscano ha disegnato e descritto non solo il volo degli uccelli, ma anche la sua Macchina Volante. L'arrivo di Curiosity sulla superficie di Marte è previsto per il mese di agosto 2012.

www.asi.it

La porta della luce



Si chiama Cnot (Integrated photonic quantum gates for polarization qubits) ed è il cuore dell'informazione quantistica, la porta logica del computer del futuro che per funzionare userà i fotoni, cioè la luce invece degli elettroni. A realizzarla una collaborazione fra l'Istituto di fotonica e nanotecnologie del Consiglio nazionale delle ricerche (Ifn-Cnr), il gruppo di ottica quantistica della Sapienza e il Politecnico di Milano. Il dispositivo, di cui è stato depositato il brevetto, è costituito da un chip di vetro delle dimensioni di un paio di centimetri sul quale viene implementato un circuito integrato che guida la luce rendendo possibile il passaggio dei quanti di informazione quantistica (qu-bit). La fabbricazione della porta logica nel vetro è stata realizzata grazie ad una tecnica innovativa che utilizza impulsi laser di brevissima durata (circa 100 milionesimi di miliardesimi di secondo) come "penna ottica" per scrivere direttamente nel chip i circuiti ottici necessari per l'elaborazione dei qubit.

"Questa sofisticata tecnologia – afferma Roberto Osellame, primo ricercatore dell'Ifn-Cnr – permette di realizzare circuiti ottici a sviluppo tridimensionale, non ottenibili con altre tecnologie, che consentono di implementare architetture innovative e di integrare in un singolo dispositivo sistemi di complessità sempre maggiore". Viene così realizzato un componente essenziale e miniaturizzato dell'hardware dei futuri computer quantistici, che saranno caratterizzati dalla capacità di effettuare con grande velocità di calcolo operazioni di complessità inaccessibile ai computer classici.

Lo studio, pubblicato sulla rivista Nature Communications, apre prospettive promettenti non solo nel campo dell'informazione e della computazione quantistica ma anche in quello della simulazione quantistica: questi dispositivi infatti possono essere utilizzati per simulare in laboratorio il comportamento di determinati fenomeni fisici difficilmente accessibili alla sperimentazione diretta.

www.cnr.it





Processing

Introduzione a Processing

di MATTEO FIORAVANTI

Scopriamo l'Object Oriented Programming, che rappresenta un modo più semplice di costruire le nostre interfacce. Quarta puntata.

Nella scorsa puntata abbiamo imparato a disegnare dei controlli grafici: indicatori, pulsanti, LED, manopole. Per vedere all'opera tali elementi, abbiamo costruito una semplice interfaccia grafica per un programma che controlla Arduino. In questa quarta puntata risolveremo la problematica di dover costruire interfacce grafiche complesse che richiedano, magari,

l'uso di molti controlli e indicatori. Infatti l'interfaccia costruita nella terza puntata funzionava come un unico programma, cioè il codice relativo alla gestione dei controlli e indicatori era fuso all'interno del codice principale del programma, il quale controllava completamente l'interfaccia. In questa sede risolveremo altresì il problema di richiamare all'interno del

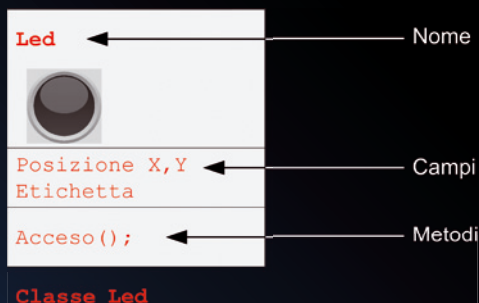


Fig. 1

programma principale i veri LED, interruttori, manopole, nelle quantità che ci occorrono, senza preoccuparci del codice occorrente ad essi, ma interessandoci unicamente al loro comportamento in termini di ingressi ed uscite (come del resto accadrebbe con i controlli reali).

Conosciamo già l'uso delle *funzioni* in Processing; ricordiamo che una funzione è definita all'interno del programma, viene richiamata a seconda dei casi e ne gestiamo ingressi ed uscite. Per il problema che affrontiamo, ci serve qualcosa di simile. Supponiamo di dover inserire nella nostra interfaccia tre LED, ognuno dei quali avrà un suo significato e sarà collegato a certe variabili, proprio come accadrebbe in un sistema reale.

Alla luce di quanto spiegato nella scorsa puntata, la prima cosa che ci viene in mente è piazzare tre immagini dei LED e modificarle a seconda delle variabili che dovrebbero rappresentare.

L'interfaccia siffatta sarà certamente funzionante, ma il relativo codice potrebbe diventare ingestibile nel momento in cui ci trovassimo a dover aggiungere altri LED ai precedenti, oppure laddove dovessimo essere costretti a complicare l'architettura del programma.

Pensiamo invece a quanto sarebbe più comodo se potessimo richiamare una specie di *costruttore* di LED, che ci permettesse di costruire i componenti led1, led2 led3 (cioè i nostri tre LED virtuali) e così via, e se ogni diodo luminoso potesse essere gestito in completa autonomia, cioè avesse un codice a sè stante e direttamente controllabile.

Potremmo, infatti, attribuire lo stato di *Accesso* a led1, mentre cambiamo led2 da *Accesso* a *Spento* e così via.

Se avessimo questa possibilità, ci basterebbe concentrarci unicamente sul programma principale ed ogni LED si com-

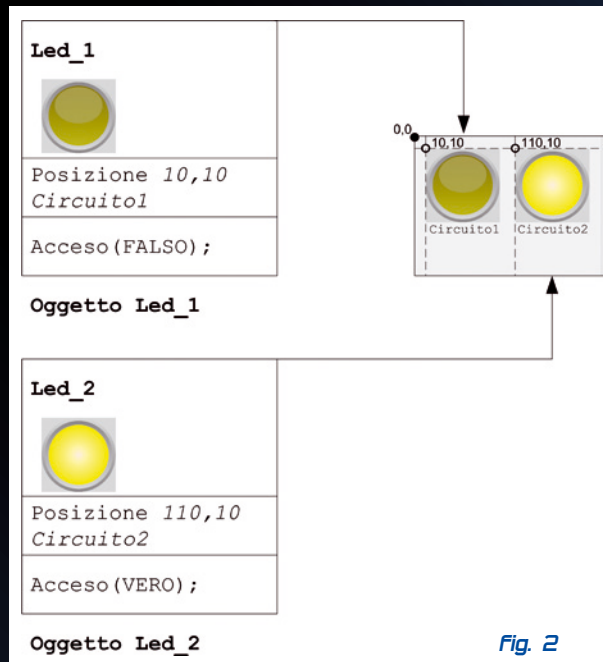


Fig. 2

porterebbe proprio come se fosse reale. Fare questo è possibile, ma dobbiamo introdurre una tecnica di programmazione nuova e che risolva in maniera estremamente semplice questi casi; tale tecnica si chiama programmazione orientata agli oggetti, ovvero, abbreviando, OOP (acronimo di Object Oriented Programming). Processing permette di implementare questa tecnica un modo del tutto naturale. Per spiegare l'OOP partiremo introducendo il concetto di Classe e di Oggetto, poi trasformeremo i controlli descritti nella lezione precedente a loro volta in Classi ed Oggetti e, con quanto appreso, scriveremo il codice di un'interfaccia che simulerà un sistema reale.

IL CODICE DI QUESTA PUNTATA

Qui di seguito elenchiamo i programmi di esempio di questa puntata del corso.

1.LEZIONE 4_1 - INTRODUZIONE ALLA PROGRAMMAZIONE AD OGGETTI.

Questo codice permette di costruire un'interfaccia con tre oggetti appartenenti alla Classe Led; serve, inoltre, per familiarizzare con i concetti di Classe e di Oggetto.

2.LEZIONE 4_2 - INTERFACCIA GRAFICA COMPLESSA. Questo codice permette di costruire un'interfaccia grafica complessa, nella quale tutti i controlli ed indicatori della terza lezione sono diventati Classi ed Oggetti.

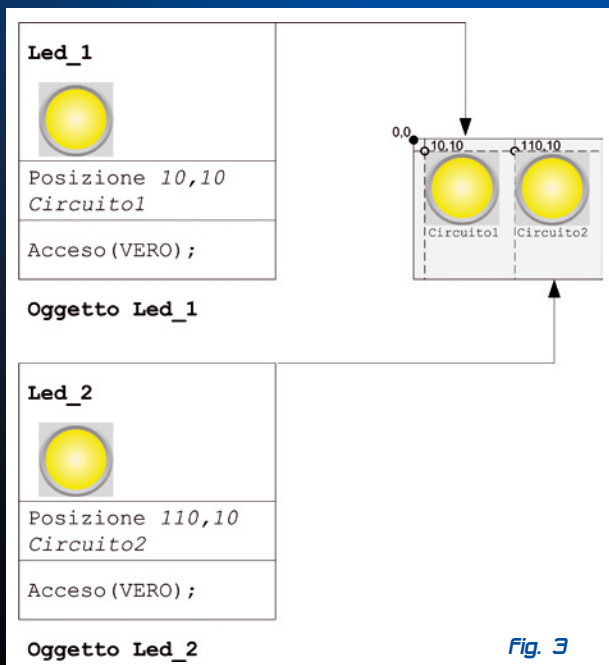


Fig. 3

LEZIONE 4_1 - INTRODUZIONE ALLA PROGRAMMAZIONE AD OGGETTI

Partiamo ora da una minima astrazione, definendo il LED come una *Classe*; ciò significa innanzitutto che alla *Classe* dobbiamo dare un *Nome*, quindi bisogna definire dei *Campi* e dei *Metodi*. Tutto questo è rappresentato in Fig. 1.

Il *Nome* della *Classe* è quello che identificherà l'*Oggetto* da essa costruito, mentre i *Campi* sono gli attributi (ad esempio la posizione X ed Y dell'oggetto e l'etichetta) mentre i *Metodi* sono i comportamenti che possono essere eseguiti. Nel caso del LED, abbiamo definito come metodo **Acceso()**: in pratica, **Acceso(true)** accenderà il diodo luminoso, mentre **Acceso(false)** lo spegnerà. Quindi, i *Campi* di una *Classe* sono degli attributi, mentre i *Metodi* sono le azioni, ovvero i comportamenti che possono essere intrapresi.

La *Classe* è un insieme generico; di essa si definiscono i singoli *Oggetti*, come ci mostra la Fig. 2. Infatti, se definiamo come *Nome* **Led_1**, come *Posizione* 10,10 e per *Etichetta* **Circuito1**, avremo definito un *Oggetto* che si chiama **Led_1**, appartiene alla *Classe* **Led**, si trova nella posizione 10,10 e mostra come etichetta "Circuito1". Ora, se volessimo accendere questo oggetto, dovremmo intervenire sul metodo **Acceso**, scrivendo quanto segue:

```
Led_1.Acceso(true);
```

Se lo volessimo spegnere, opereremmo sempre attraverso lo stesso metodo, ma scrivendo:

```
Led_1.Acceso(false);
```

Volendo inserire nel nostro ipotetico programma un secondo LED, basterà definire un *Oggetto* che, verosimilmente, chiameremo **Led_2**, sempre appartenente alla *Classe* **Led**, ma avente attributi differenti, una diversa posizione X ed Y (110,10) ed una differente Etichetta (**Circuito2**). Quindi abbiamo creato, o meglio Processing ha *costruito*, un nuovo *Oggetto* chiamato **Led_2**, diverso da **Led_1**, ma avente comunque gli stessi *Metodi*. Infatti, se volessimo accendere **Led_2** opereremmo in questo modo:

```
Led_2.Acceso(true);
```

Supponiamo, ora, di avere all'interno del programma la situazione rappresentata in Fig. 2, ossia **Led_1** spento e **Led_2** acceso. Operando in termini di *Metodi*, possiamo accendere **Led_1** e passare alla situazione di Fig. 3, scrivendo:

```
Led_1.Acceso(true)
```

Chiaramente quanto riportato fin qui è un esempio che dovrebbe aiutarci a capire il concetto di *Classe* ed *Oggetto*.

Scendiamo ora sempre più sul caso pratico, per vedere come tutto questo si applica. Consideriamo il codice di esempio (Lezione_4_1.pde): prima di tutto gli *Oggetti* devono essere dichiarati, proprio come succede con le variabili; infatti nel nostro codice abbiamo bisogno di tre LED (**ld1**, **ld2**, **ld3**), quindi dichiareremo tre oggetti "led" appartenenti alla *Classe* **Led**, come riportato in Fig. 4. Poi, dentro **void setup()** andremo a fare ciò che prende il nome di istanza degli oggetti precedentemente dichiarati; in pratica, ciò si fa attraverso il comando **new**. Perciò, la nuova istanza dell'oggetto **ld1** assumerà questo aspetto:

```
ld1 = new Led(10, 10, "Led 1");
```

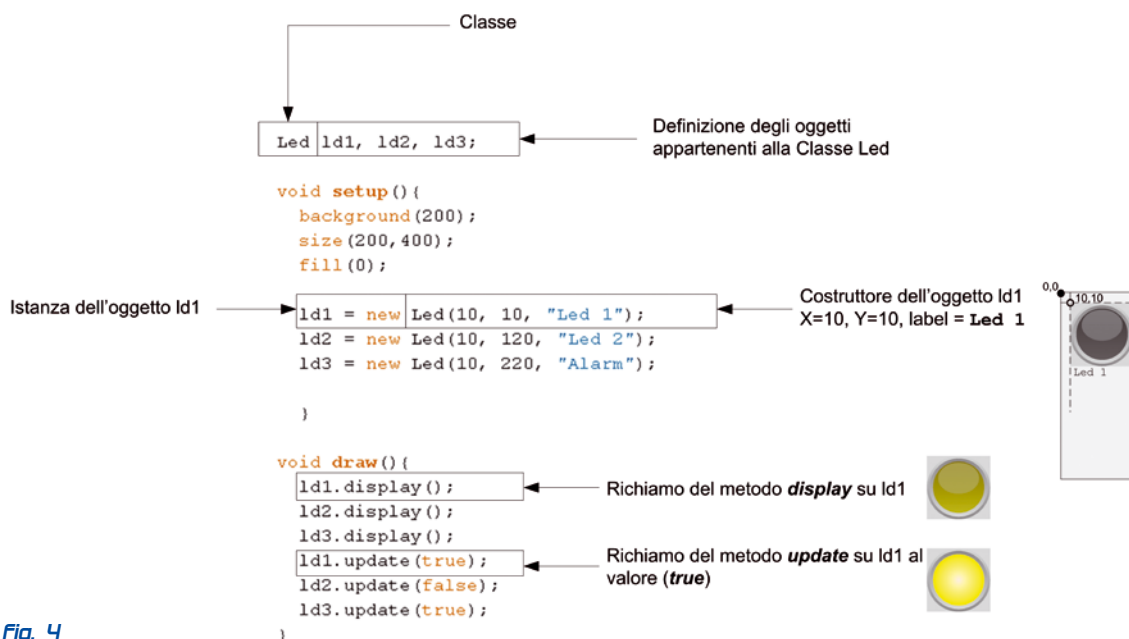



Fig. 4

Ciò significa che il nuovo oggetto `ld1` della *Classe* `Led` deve essere costruito con i seguenti campi: posizione `X = 10`, posizione `Y = 10`, Etichetta = `"Led 1"`. Procediamo allo stesso modo per gli altri *Oggetti*, come riportato in Fig. 4. Dentro il ciclo principale `void draw()`, si andrà ad agire sui *Metodi* di `Led`. Per la *Classe* `Led` abbiamo definito due *Metodi*: `.display()` e `.update()`. Il primo fa letteralmente apparire il LED (ciò significa che se non viene richiamato questo metodo, il diodo luminoso non compare nell'interfaccia) anche se è stato definito ed istanziato, mentre il secondo consente

di aggiornare dinamicamente lo stato del LED, facendolo passare da acceso a spento e viceversa.

Il codice del programma corrispondente è riportato di seguito nel Listato 1 ed è anche visibile in Fig. 4.

A questo punto abbiamo visto come si opera con le *Classi* e con gli *Oggetti*; rimane da capire come questi si programmano, ed è quanto faremo ora. Scrivere il codice per definire una *Classe* non è per nulla differente dallo scrivere codice per un programma o per una funzione; cambiano solo alcuni dettagli.

Prima di tutto la *Classe* (che ad esempio

Listato 1

```

Led ld1, ld2, ld3; //dichiarazione degli oggetti ld1, ld2, ld3 appartenenti alla Classe Led

void setup(){ //inizializzazione
  background(200); //sfondo
  size(200,400); //dimensione dell'area grafica del programma
  fill(0);

  ld1 = new Led(10, 10, "Led 1"); //istanza dell'oggetto ld1 e sua costruzione
  ld2 = new Led(10, 120, "Led 2"); //istanza dell'oggetto ld2 e sua costruzione
  ld3 = new Led(10, 230, "Alarm"); //istanza dell'oggetto ld3 e sua costruzione
}

void draw(){
  ld1.display(); //richiamo del metodo .display() dell'oggetto ld1, in pratica ld1 viene visualizzato
  ld2.display(); //identica cosa per ld2
  ld3.display(); //identica cosa per ld3
  ld1.update(true); //richiamo del metodo .update() per ld1, mettendolo a "true" si accende il led ld1,
                  //mettendolo a "false" si spegne
  ld2.update(false); //ld2 viene spento
  ld3.update(true); //ld3 viene acceso
}

```

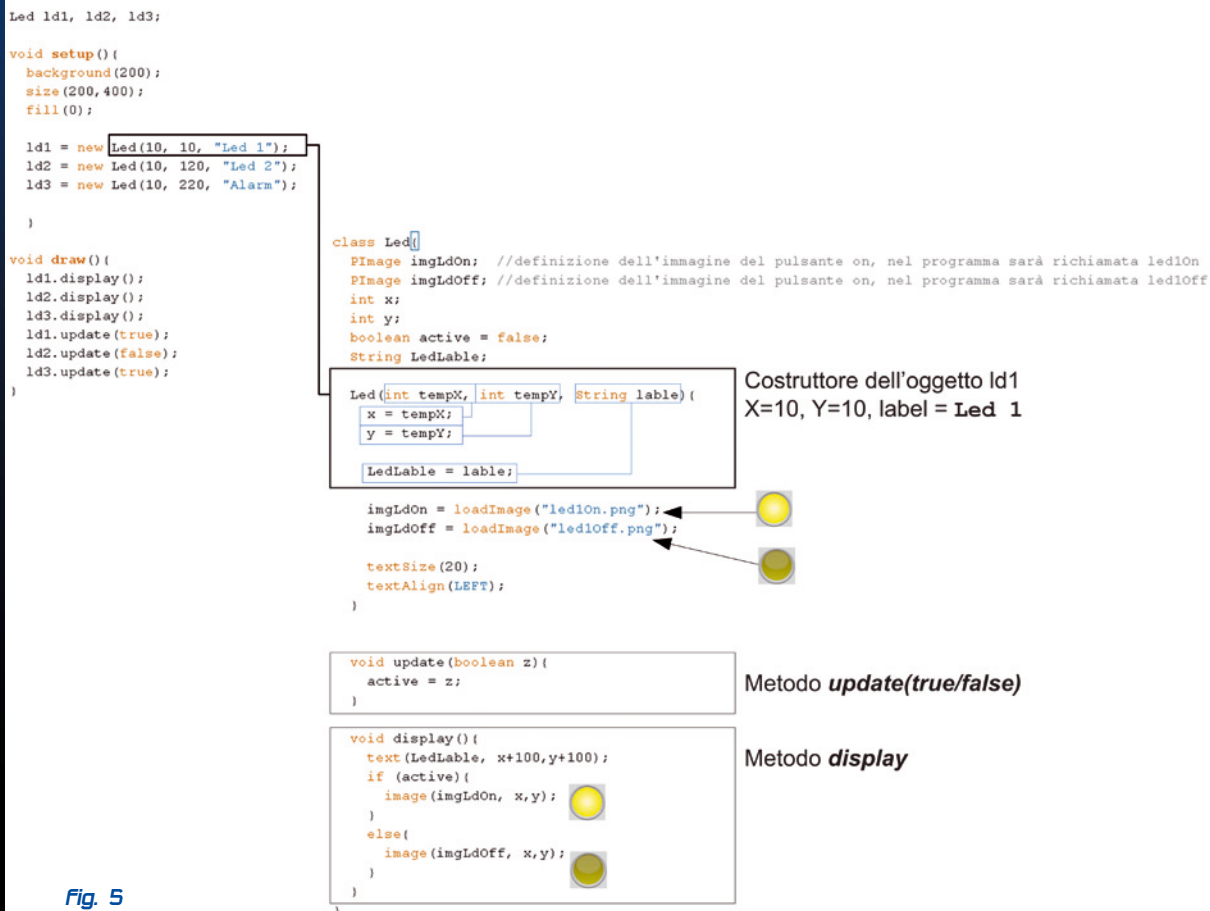


Fig. 5

può essere Led) è definita all'interno di due parentesi graffe:

```
class Led{}
```

Questo significa che tutto quanto serve alla *Classe* Led per funzionare è contenuto all'interno di tali parentesi.

Ci saranno delle variabili locali e inoltre verranno definiti il costruttore dell'interfaccia e i metodi. Nel Listato 2 è riportato il codice commentato per la *Classe* Led e lo stesso è visibile nella Fig. 5, che mostra come questo codice interagisce con il programma principale.

Un aspetto importante su cui porre l'attenzione è che la classe, per funzionare, potrebbe necessitare di sue variabili interne; tali variabili sono definite come in qualsiasi altro programma di Processing e soprattutto sono delle variabili locali. Quando viene istanziato un nuovo oggetto, abbiamo visto come questo debba essere definito nei suoi *Campi*; questi ultimi saranno poi passati alle variabili locali della *Classe*, così come mostrato dalla Fig.

5, attraverso quello che viene definito il *Costruttore dell'Oggetto*. Per la *Classe* in questione, quest'ultimo assume la seguente forma:

```
Led(int tempX, int tempY, String label) {}
```

Il *Costruttore* è in definitiva il ponte che consente di trasferire i *Campi*, definiti quando l'*Oggetto* viene istanziato, alle variabili locali che servono poi al programma per costruire l'*Oggetto*.

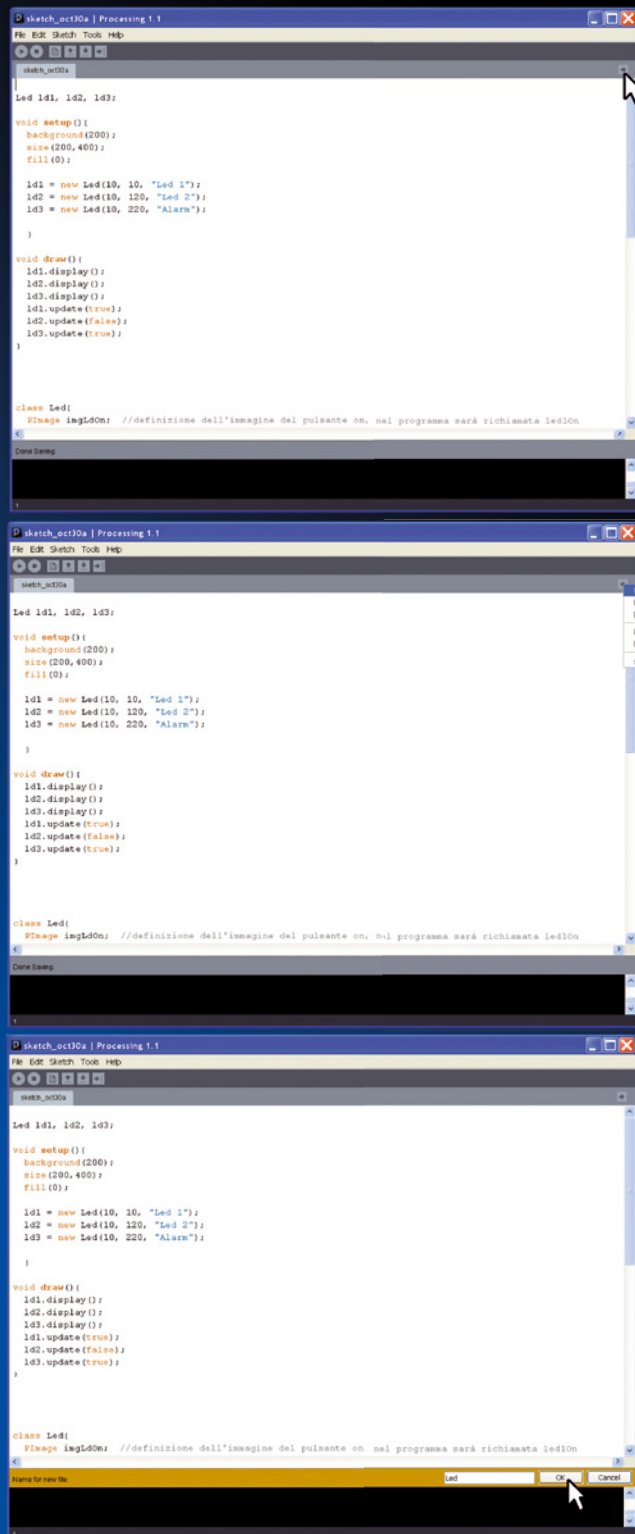
Riassumendo, i *Campi* vengono definiti nel programma principale quando si istanzia il nuovo *Oggetto*; nella *Classe*, i *Campi* sono trasferiti alle variabili locali attraverso il *Costruttore dell'Oggetto*.

La Fig. 5 rappresenta anche, in maniera schematica, i due *Metodi*, i quali all'interno della *Classe* sono sostanzialmente delle funzioni; ad esempio:

```
void display() {}
```

Quindi il *Metodo*, in realtà, è una funzione, che però viene richiamata dall'*Oggetto*

Fig. 6



proprietario, nel programma principale:

```
...
ld1.display();
...
```

Nel Listato 2 (codice riguardante la clas-

se Led) la parte riguardante la *Classe* può essere scritta in fondo al programma principale, cioè praticamente fuori da `void draw()`, analogamente a quanto si fa per le funzioni. Però esiste un modo che ci consente di rendere il codice ancora più leggibile: si tratta di sfruttare una funzionalità dell'editor di Processing, che è quella che ci permette di aprire dei nuovi Tab di editor e che ancora non abbiamo affrontato. La Fig. 6 ci illustra come ciò viene fatto e comunque la procedura è la seguente: innanzi tutto occorre andare con il cursore, nell'angolo in alto a destra dell'editor, sopra alla rappresentazione della freccia verso destra, e fare clic con il tasto sinistro del mouse. Allora

si aprirà un menu a tendina e bisognerà selezionare **New Tab**; a questo punto comparirà una barra gialla sotto l'editor, contenente la richiesta di inserimento del nome per il nuovo file "New name for file". Nella casella di testo bisogna scrivere il nome del nuovo Tab, che noi con "grande originalità" chiameremo *Led*. Di fianco al Tab indicante il nome del file, ne sarà comparso un altro con il nome di "Led"; facendovi clic, si aprirà una nuova pagina di editor. Ora non rimane altro da fare che copiare ed incollare tutto il codice relativo alla *Classe Led*, che avevamo scritto nel programma principale, ed incollarla nel nuovo Tab *Led* (Fig. 7). A questo corrisponde effettivamente un nuovo file, che si trova nella stessa cartella del programma principale. Il codice, quindi, rimane semplicemente quello di Fig. 8, ma soprattutto, la *Classe Led* che abbiamo creato possiamo trasportarla in altri programmi, spostando il file *Led.pde* nelle cartelle dei programmi che andremo a creare; inoltre, dentro ai programmi potremo creare tutti gli oggetti "led" che vorremo, ricordandoci di definirli all'inizio del programma, di istanziarli e di usarli con i relativi metodi. Questa tecnica consente di semplificare notevolmente il codice e di riutilizzarlo tutte le volte che vorremo nei programmi che andremo a creare. Dal punto di vista della creazione di interfacce grafiche, questo permette, una volta che

Listato 2

```

class Led{
  PImage imgLdOn; //definizione dell'immagine del pulsante on, nel programma sarà richiamata led10n
  PImage imgLdOff; //definizione dell'immagine del pulsante on, nel programma sarà richiamata led10ff
  //dichiarazioni delle variabili funzionali all'esecuzione della classe
  int x;
  int y;
  boolean active = false;
  String LedLabel;

  //costruttore
  Led(int tempX, int tempY, String lable){ //la Classe Led necessita di tre campi che devono essere definiti
                                          // quando si istanzia un nuovo oggetto
                                          //la posizione X, Y ed una stringa che definisce l'etichetta
    x = tempX; //il campo tempX viene passato alla variabile globale della Classe X
    y = tempY; //il campo tempY viene passato alla variabile globale della Classe Y

    LedLabel = lable; //il campo lable viene passato alla variabile globale della Classe LedLabel

    imgLdOn = loadImage("led10n.png");//vengono caricate le immagini .png rappresentative del led acceso e
    spento
    imgLdOff = loadImage("led10ff.png");

    textSize(20); //viene definita dimensione del carattere
    textAlign(LEFT); //viene definito l'allineamento del testo, a sinistra
    //questi parametri servono per la rappresentazione dell'etichetta
  }

  //METODO update() -----
  void update(boolean z){ //il metodo si aspetta che sia passato un parametro true/false
    active = z; //il valore del parametro è trasferito alla variabile globale "active", della Classe
  }
  //-----

  //METODO display() -----
  void display(){
    text(LedLabel, x+100,y+100);
    if (active){ // a seconda del valore della variabile "active" il metodo display() mostra un'immagine
      diversa del led
      image(imgLdOn, x,y); //immagine del led acceso, se active == true
    }
    else{
      image(imgLdOff, x,y); //immagine del led spento, se active == false
    }
  }
  //-----
}

```

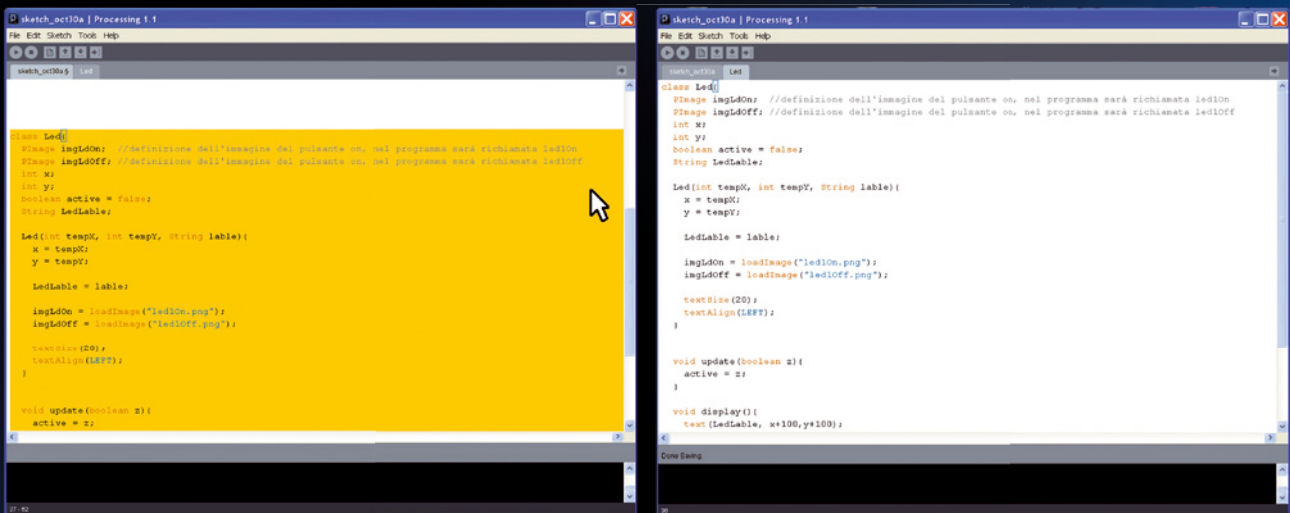
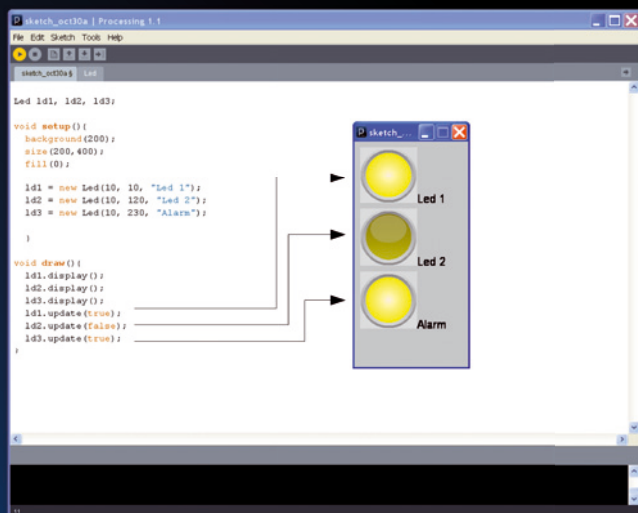


Fig. 7



avremo creato i nostri controlli, di concentrarci unicamente sul programma.

LEZIONE 4_2 - INTERFACCIA GRAFICA COMPLESSA

Il secondo codice presentato in questa lezione ci serve per prendere dimestichezza con i controlli di tipo Led, Knob, Button ed Indicator, trasformati però in Classi e quindi in *Oggetti*. Supponiamo di voler realizzare l'interfaccia di Fig. 9, dove si trovano tre interruttori, tre LED, una manopola e un indicatore analogico. Stabiliamo a questo punto una funzionalità puramente di esempio: l'interruttore denominato "Start" attiva un ipotetico sistema che incrementa la sua velocità e questa viene indicata dalla lancetta dell'indicatore analogico; il livello di questa velocità è settato tramite la manopola "Level" e quando raggiunge una soglia, si deve accendere il LED denominato "Alarm". Questa funzionalità è

Fig. 8

rappresentata in Fig. 10. Aggiungiamo un'ulteriore complessità, cioè facciamo in modo che i due interruttori siglati "Circuit 1" e "Circuit 2" attivino, rispettivamente, "Led 1" e "Led 2" (Fig. 11). L'interfaccia qui rappresentata è in definitiva un simulatore: non serve a nulla di reale, ma ci consente di apprendere una maniera più efficace di programmare questo genere di applicazioni. Se vogliamo che la nostra interfaccia controlli realmente qualcosa, possiamo riprendere le tecniche illustrate nella lezione precedente e magari collegare il tutto ad Arduino. Il codice contenuto nella cartella Lezione_4_2 contiene sia il programma principale che simula l'interfaccia, sia tutte le classi usate in questo esempio: Led, Button, Indicator e Knob. Il codice del programma è strutturato come nell'esempio precedente, quindi nella fase iniziale in cui si dichiarano gli oggetti, nella fase di istanza degli oggetti e nel vero e proprio ciclo principale del programma. Le figure 12, 13, 14 e 15 riassumono le *Classi* utilizzate, ciascuna con i *Campi* e *Metodi* caratteristici.

Brevemente, tutti i *Campi* sono rappresentati dalle posizioni X, Y e dall'etichetta. Tutte le *Classi* qui usate hanno il metodo `.display()`, che ne consente la visualizzazione.

La *Classe* Led ha il *Metodo* `.update()`, il quale, come spiegato, se gli viene passato true accende il LED, mentre lo spegne se riceve false.

La *Classe* Button ha il *Metodo* `.active()`, in-



Fig. 9

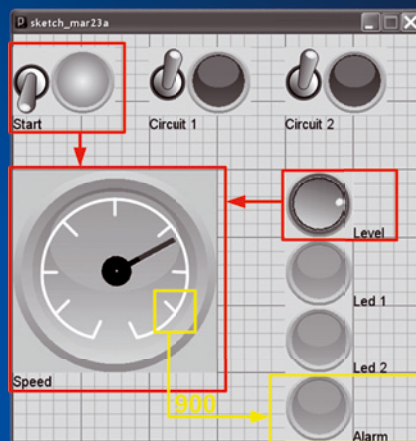


Fig. 10

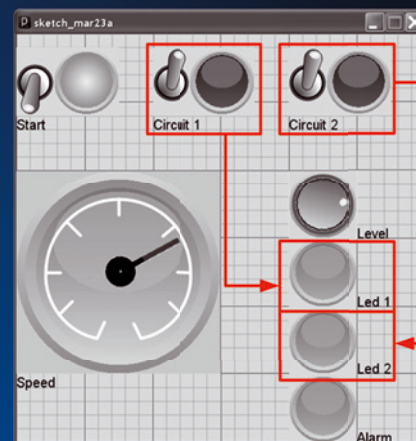


Fig. 11

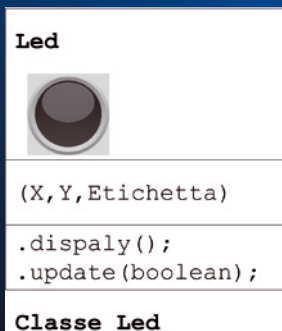


Fig. 12

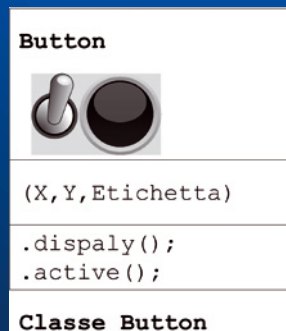


Fig. 13

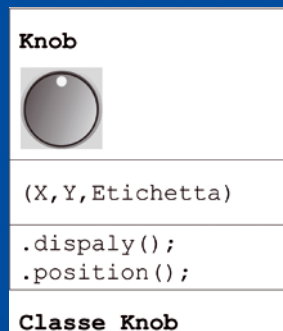


Fig. 14

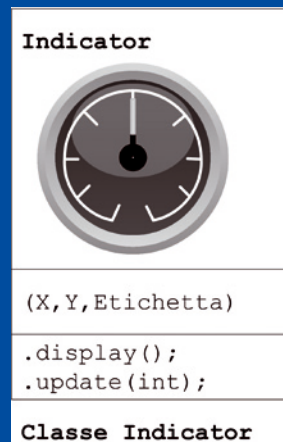


Fig. 15

vece, restituisce un booleano (true/false) a seconda della posizione dell'interruttore. La *Classe Knob* ha il *Metodo* `.position()`, il quale restituisce un valore intero compreso tra 0 e 1024, proporzionale alla posizione della manopola. La *Classe Indicator* ha il *Metodo* `.update()` che si aspetta un valore intero compreso tra 0 e 1024; chiaramente il valore 0 corrisponderà all'inizio della scala, mentre 1024 equivarrà alla

posizione di fondo scala. Il codice del programma di interfaccia è contenuto nel Listato 3. Non ci dilunghiamo, in questa lezione, sul codice specifico delle varie *Classi* qui presentate, in quanto è sostanzialmente lo stesso esposto nella terza puntata del nostro corso, ma scritto sotto forma di *Classe*. Quindi, a livello sostanziale non cambia nulla: è solo differente la forma-

Listato 3

```
//definizione degli OGGETTI
Button bt1, bt2, bt3; //definizione degli oggetti della classe Button
Indicator ind1; //definizione degli oggetti della classe Indicator
Knob knb1; //definizione degli oggetti della classe Knob
Led ld1, ld2, ld3; //definizione degli oggetti della classe Led
//-----

//dichiarazione VARIABILI
PImage griglia; //definizione della variabile griglia di tipo PImage
int i=0; //definizione della variabile i che servirà come contatore
//-----

void setup(){
  background(200); //colore di sfondo
  size(600,600); //dimensione dell'area di disegno

  bt1 = new Button(0,20,"Start"); //istanza dell'oggetto bt1 e sua costruzione
  bt2 = new Button(200,20,"Circuit 1"); //istanza dell'oggetto bt2 e sua costruzione
  bt3 = new Button(400,20,"Circuit 2"); //istanza dell'oggetto bt3 e sua costruzione
  ind1 = new Indicator(0,200,"Speed"); //istanza dell'oggetto ind1 e sua costruzione
  knb1 = new Knob(400,200,"Level"); //istanza dell'oggetto knb1 e sua costruzione
  ld1 = new Led(400, 300, "Led 1"); //istanza dell'oggetto ld1 e sua costruzione
  ld2 = new Led(400, 400, "Led 2"); //istanza dell'oggetto ld2 e sua costruzione
  ld3 = new Led(400, 500, "Alarm"); //istanza dell'oggetto ld3 e sua costruzione

  griglia = loadImage("griglia_600_600.png"); //si carica l'immagine di una griglia per facilitare il posizionamento degli elementi dell'interfaccia
}

void draw(){
  image(griglia,0,0); //rappresentazione dell'immagine griglia

  bt1.display(); //rappresentazione di bt1
  bt2.display(); //rappresentazione di bt2
  bt3.display(); //rappresentazione di bt3
  ind1.display(); //rappresentazione di ind1
  knb1.display(); //rappresentazione di knb1
  ld1.display(); //rappresentazione di ld1
  ld2.display(); //rappresentazione di ld2
```

(Continua)

Listato 3 – segue

```
ld3.display(); //rappresentazione di ld3

//funzionamento dell'interfaccia

//se il bt1 "Start" è attivo
//l'indicatore crescerà fino a raggiungere il valore impostato dalla manopola knob1 "Level"
//se riduciamo il valore della manopola, il valore rappresentato dall'indicatore si ridurrà

if(bt1.active()){ //se il l'interruttore bt1 "Start" è in posizione attiva

  println(knob1.position()); //sul terminale viene rappresentato il valore trasmesso dalla posizione della
  manopola knob1

  if(i<knob1.position()){ //se la posizione della manopola è superiore alla posizione dell'indicatore
    i +=1; //incremento la variabile i
    ind1.update(i); // aggiorno la posizione dell'indicatore ad i
  }
  if(i>knob1.position()){ //se la posizione della manopola è inferiore a quella dell'indicatore
    i -=1; //decremento i
    ind1.update(i); //aggiorno la posizione dell'indicatore ad i
  }

  if(i>900){ //se la variabile i oltrepassa 900
    ld3.update(true); //aggiorno lo stato del led3 "Alarm" ad acceso
  }else{
    ld3.update(false); //altrimenti lo stato di led3 è spento
  }

}

if(bt2.active()){ //se l'interruttore bt2 è acceso
  ld1.update(true); //aggiorno lo stato di ld1 a "true", accendo il led
}
else{
  ld1.update(false); //altrimenti lo stato del led è spento
}

if(bt3.active()){ //se l'interruttore bt3 è acceso
  ld2.update(true); //aggiorno lo stato di ld2 a "true", accendo il led
}
else{
  ld2.update(false); //altrimenti lo stato del led è spento
}

}
```

lizzazione, come spiegato in riferimento al codice della *Classe Led*. Comunque vi invitiamo ad analizzare il codice delle varie *Classi* e ad apportare tutte le modifiche che ritenete opportune a seconda delle vostre necessità. Una piccola nota: la Fig. 16 mostra l'interfaccia completa e funzionante, ma dobbiamo notare che in essa è presente una griglia di sfondo, richiamata da:

```
griglia = loadImage("griglia_600_600.png");
```

Praticamente, abbiamo creato un'immagine rappresentante un quadrato di 600x600 pixel, con una griglia interna spaziata di 20 pixel, la quale ci consente di avere un riferimento quando piazziamo i controlli e gli indicatori sull'interfaccia, permettendoci in maniera

più rapida di allinearli. Una volta completata l'interfaccia, basterà eliminare (commentare) la riga di codice precedente, che richiama la griglia, per far sparire questa immagine. ■



Fig. 16

La Bachecca 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 (campionatura).
 - Accumulatore al gel a tracolla con multiuscite multitensioni ad uso contemporaneo accessorio 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 accessorio.
 - 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 polycarbonato anche a pezzi unici senza costi di attrezzatura.

Vendo:

- Kit per la stampa in proprio di mascherine in polycarbonato composto da 2 fogli formato A4 di speciale policalcarbonato 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.argo@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:

- Ricetrasmittitore 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;
 - Ricetrasmittitore 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. Appareto 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.elettronica.in.it nella sezione "Annunci".

CREA LA GIUSTA ATMOSFERA...

...CON GLI STRIP A LED

A casa, in negozio o nella tua piscina
potrai ricreare l'ambientazione ideale per ogni occasione,
personalizzando con un SOLO gesto l'illuminazione.
Tutto questo semplicemente utilizzando strip a LED, un controller
ed un modulo di alimentazione!

STRIP

cod. R300RGB-IP67

cod. CHLS4RGB

cod. CHLS11RGB

cod. CHLS16RGBW

€189.00

€95.00

€128.00

€130.00

RGB
12V/78W
IP67
5m

RGB
12V/36W
IP61
5m

RGB
12V/36W
IP68
5m

RGB + Bianco
12V/48W
IP61
5m

CONTROLLER

cod. CHLSC2

cod. CHLSC1

cod. CHLS16C

€23.50

€57.00

€89.00

12Vdc/50W
Telecomando IR

12Vdc/180W
24Vdc/360W
Telecomando IR

12Vdc/240W
24Vdc/480W
DMX
Telecomando IR

ALIMENTATORE

cod. PSIN10012

€29.00

12Vdc
100W



www.futurashop.it



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

Prezzi IVA inclusa.

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

Telecamere IP

Telecamera dome IP

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



cod. CP479

€ 74,00

(con interfaccia RJ-45)

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

cod. CP439

€ 105,00

(con interfaccia RJ-45)



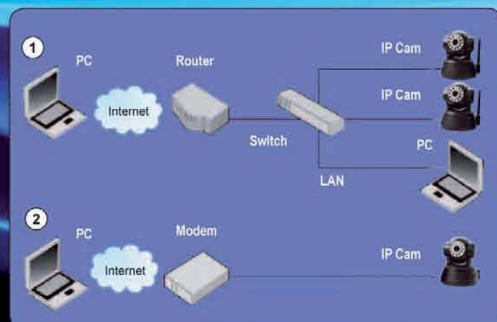
Telecamera IP motorizzata a colori PAN (270°) / TILT (120°)

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

cod. CP439W

€ 129,00

(con interfaccia Wi-Fi / RJ-45)



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

Telecamera IP da esterno con Wi-Fi e IR

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



cod. CP391W

€ 139,00

(con interfaccia Wi-Fi / RJ-45)

Versione con interfaccia RJ-45
(cod. CP391 • € 118,00)



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

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



cod. CAMIP4N

€ 290,00

cod. CP360

€ 98,00

(con interfaccia RJ-45)



cod. CP360W

€ 120,00

(con interfaccia Wi-Fi / RJ-45)



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

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

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.

ARDUINO

lo strumento
per stimolare il tuo
ingegno e la tua
creatività

Tre proposte per iniziare a
sperimentare con la piattaforma
Made in Italy più famosa al mondo.

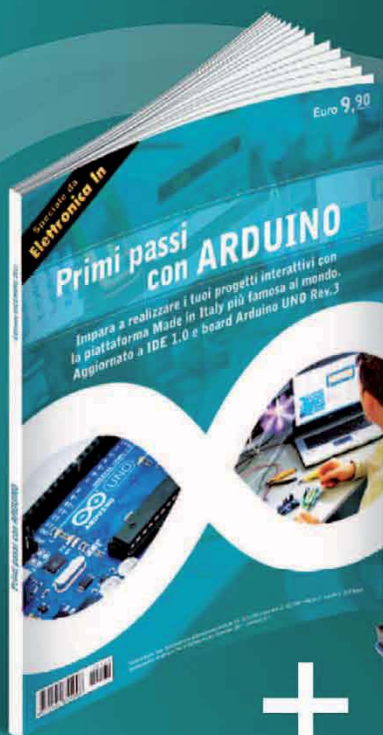


Starter kit con Arduino UNO R3

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

cod. ARDUINOKITBOOK

€ 52,00



cod. ARDUINOKITBOOK1

solo
€ 29,90
risparmi
€ 4,50!



+

**Volume "Primi passi con ARDUINO"
e scheda Arduino UNO R3**



Arduino UNO R3
è la più recente
versione della
scheda di
sperimentazione
basata su
Atmega328.

cod. ARDUINOUNOREV3

€ 24,50

**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.