

160 PAGINE!

Mensile di progettazione elettronica, attualità scientifica, novità tecnologiche.

€ 6,00

Elettronica In

oltre l'elettronica

www.elettronica.in.it

Anno XVII - n. 157
Giugno 2011



come funziona il
**DIGITALE
TERRESTRE**



Costruiamo una stampante 3D



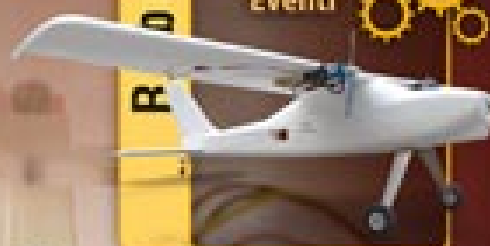
Pubblichiamo su
WEB i dati meteo



DemoNetShield

Robot Area

- Realizziamo un UAV bimotore
- News ed Eventi



- AUSSIE ZERO, il caricabatterie
- Contatore Geiger palmare
- Sistema RFID per TAG scrivibili
- Rilevamento per deflessione, quali sensori?
- Corso NI Multisim
- Corso HW/SW Arduino

• NOVITÀ •
Corso **iphone**



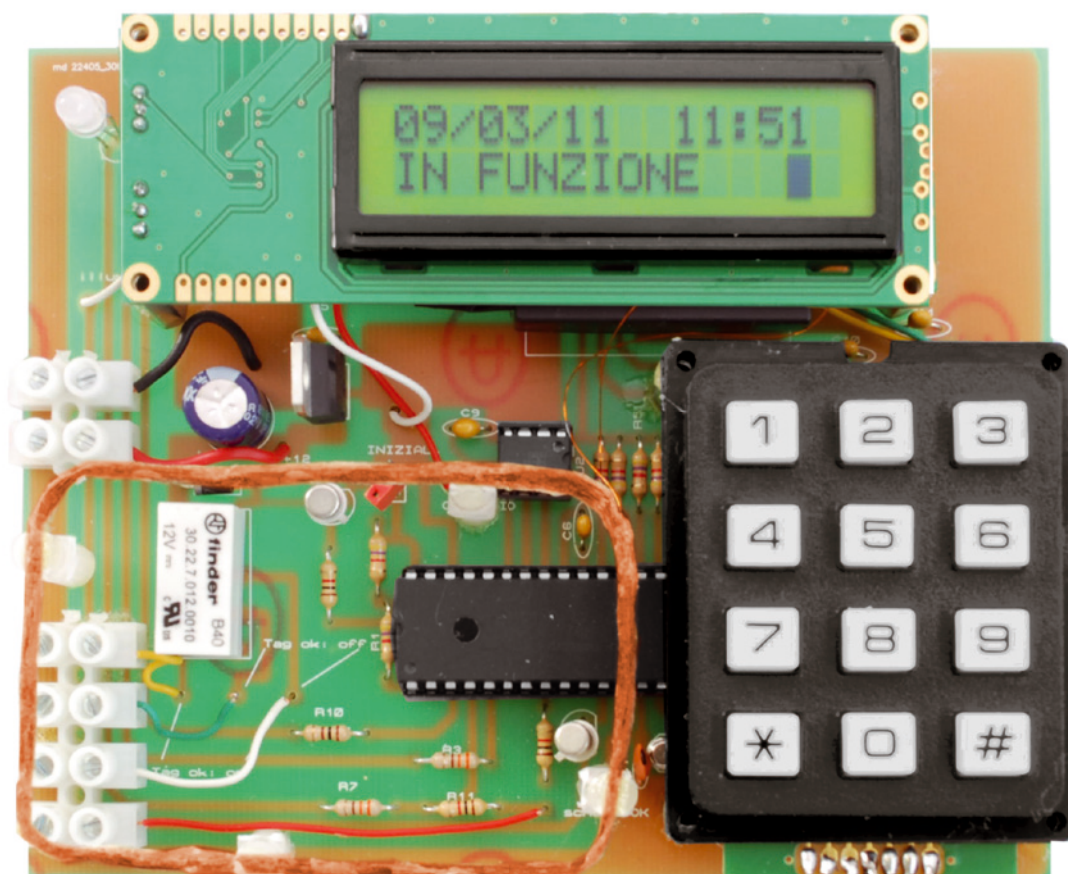
SISTEMA RFID PER TAG SCRIVIBILI

..... dell'ing. MICHELE BARONE

Nella prima puntata di questo articolo (pubblicata nel numero precedente di Elettronica In) vi abbiamo esposto tutta la teoria necessaria per progettare e costruire un circuito per la lettura-scrittura dei tag, basato sul modulo ID-20 WR della Innovation. Ora è venuto il momento di vedere come funziona e in che modo viene realizzato in pratica. Partiamo con la descrizione del

software di gestione, ovvero il firmware caricato nel microcontrollore che governa il sistema e che consente di realizzare tutte le funzioni descritte. Il firmware non è particolarmente complesso, tuttavia è abbastanza corposo, dato che occupa quasi tutti i 16 k messi a disposizione dalla memoria di programma del PIC 18F442 che abbiamo previsto per governare l'unità.

Dopo aver introdotto le basi della tecnologia RFID e analizzato le potenzialità del modulo di lettura scrittura ID-20 accoppiato ad un tag EM4469, descriviamo il progetto completo di un sistema che non esitiamo a definire "professionale".
Ultima puntata.



Elenco Componenti:

R1, R2: 4,7 kohm
R4, R5, R6: 4,7 kohm
R8, R9, R10, R13: 4,7 kohm
R3, R7: 330 ohm
R7: 330 ohm
R11, R14, R15: 1 kohm

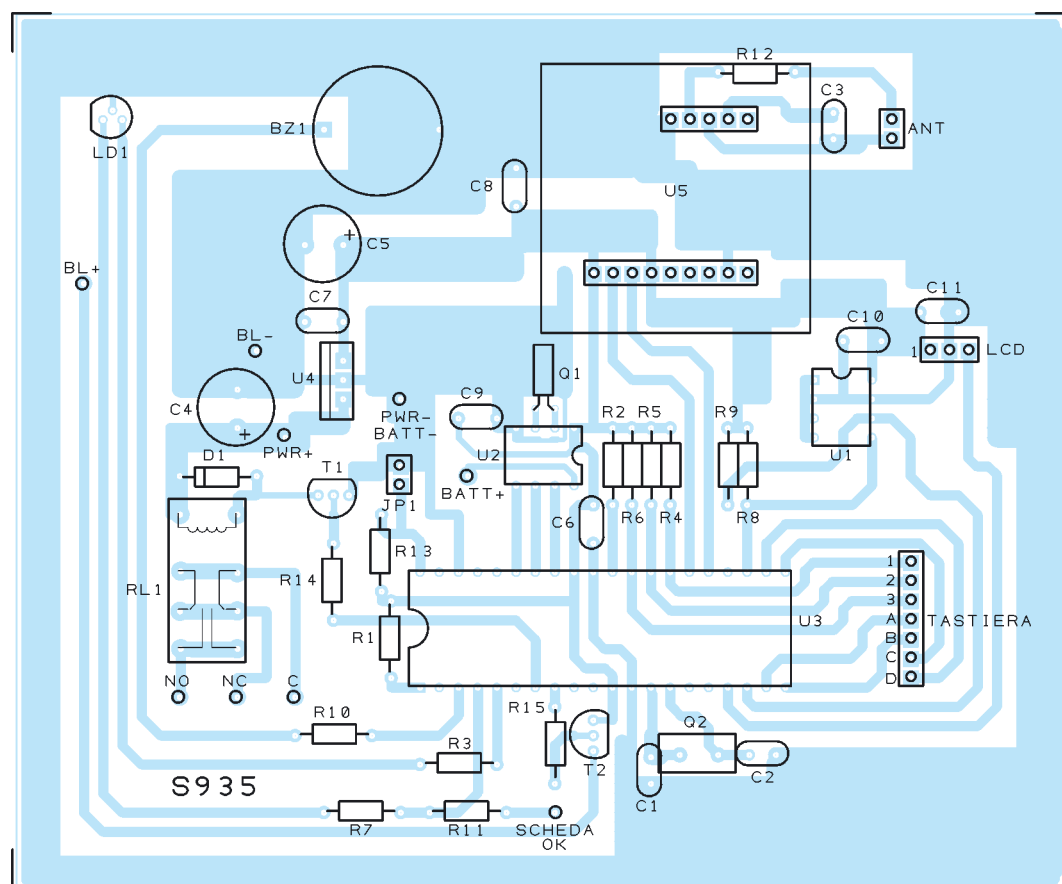
R12: 100 ohm
C1; C2: 22 pF ceramico
C3: 10 nF multistrato
C4, C5: 470 μ F 25 VL elettrolitico
C6÷C11: 100 nF multistrato
U1: 24LC256

U2: DS1302
U3: PIC18F442 (MF935)
U4: 7805
U5: ID-20
T1, T2: 2N2222
D1: 1N4007

Cominciamo col vedere il macro flow chart, per poi dettagliare l'algoritmo delle funzioni principali. In esso, in verde sono riportati i blocchi funzionali che saranno ulteriormente dettagliati. Il ciclo principale del processore inizia con una verifica del tempo passato dall'ultimo refresh dello schermo LCD. L'orario viene aggiornato una volta al minuto, quindi è inutile mandare ripetutamente l'informazione al display tutte le volte che si entra nel ciclo.

Il display perciò viene "rinfrescato" una volta al minuto. Poi si passa a verificare se è stato premuto un "tasto significativo" sulla tastiera: se si è premuto il tasto 0, si attiva la funzione di lettura scheda senza addebito. Se è stato premuto il tasto asterisco, viene chiesta l'introduzione della password e dopo la verifica positiva si accede alla navigazione nei menu di scelta delle funzioni di programmazione, che sono: *creazione nuova scheda, aggiornamento scheda, cancellazione*

scheda, cambio pin e regolazione orologio. Una volta completata la funzione si può scegliere se proseguire o uscire dalle attività di programmazione. Usciti dalla programmazione si attiva la funzione di lettura delle schede. Pur non essendo stata evidenziata nel flow chart, è prevista una funzione che resettava il PIC qualora, entrati nel menu di programmazione, non si premeva alcun tasto per più di 30 secondi; in questo modo si evita di lasciare "aperto" il dispositivo in



LD1: L-59EGW
 Q1: Quarzo 32.768 kHz
 Q2: Quarzo 4 MHz
 RL1: Relé doppio scambio 12V/1A
 LCD: LCDSER16X2NV
 BZ1: Buzzer da cs

Varie
 - Batteria CR1532 3 V Litio
 - Zoccolo 4+4 (2 pz.)
 - Zoccolo 20+20
 - Vite 10 mm 3 MA (8 pz.)
 - Distanziale esagonale M3 (8 pz.)

- Pin strip femmina 5 vie
 - Pin strip femmina 9 vie
 - Pin strip femmina 15 vie
 - Tastiera a matrice 4x3
 - Circuito stampato

modalità di programmazione. Se viene letta una scheda ed è valida, il sistema ne confronta il codice (numero scheda assegnato dal sistema, da non confondersi con l'identificativo univoco del chip in essa contenuto) con il codice della lettura immediatamente precedente. Questo accorgimento evita di "scaricare" ripetutamente il credito presente sulle schede a credito; ciò perché avvicinando una scheda al lettore, le letture avvengono consecutivamente al ritmo di circa due al secondo,

perciò mantenendo la scheda sul lettore, si corre il rischio di scaricare più crediti del necessario. In assenza di scheda, il lettore legge un codice composto da tutti zeri; quindi, grazie al confronto del codice corrente con quello precedente, per scaricare due crediti occorre necessariamente un passaggio per una situazione di assenza della scheda, ovvero bisogna avvicinare, allontanare e poi avvicinare nuovamente il tag. Se una scheda viene riconosciuta, si procede col verificarne

la data di scadenza e qualora si tratti di una scheda a credito, si verifica anche la consistenza del credito. A valle di tutte le verifiche, se esse danno esito positivo si procede con l'attivazione del segnale "scheda ok" ed eventualmente all'eccitazione del relé, per un periodo di 0,5 secondi. In caso contrario (scheda scaduta o credito esaurito) la situazione viene segnalata sul display e ovviamente non si attivano le uscite. A questo punto il ciclo del processore termina e riparte.

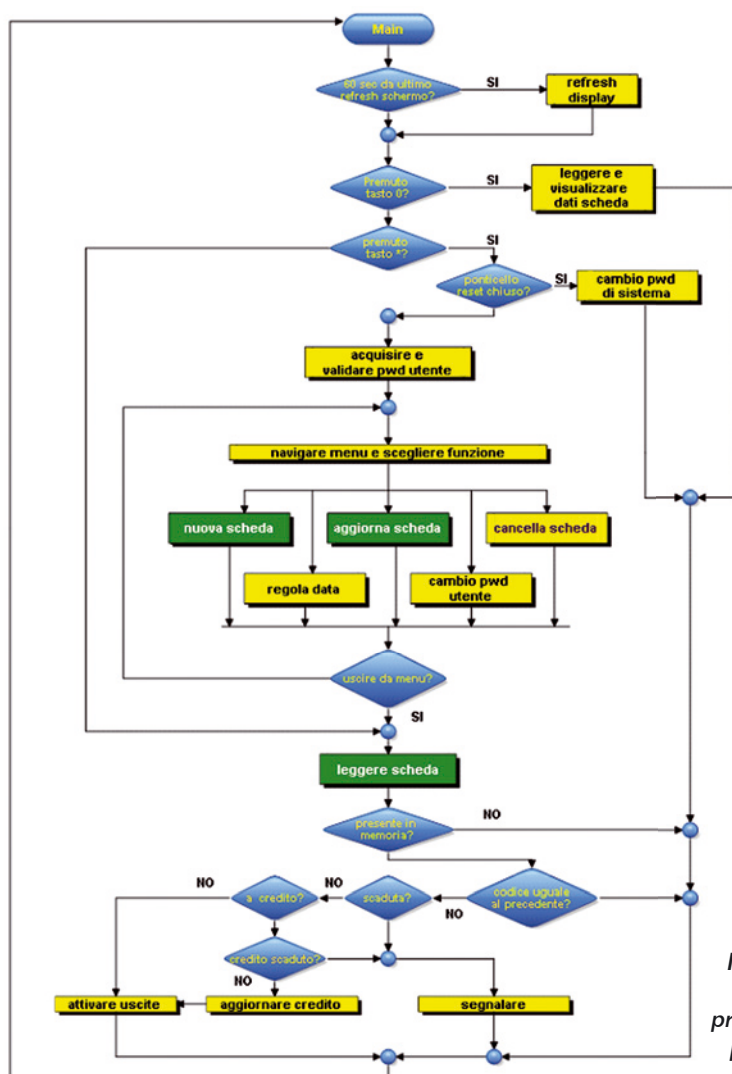


Fig. 1
Il flow-chart
del ciclo
principale del
processore.

LA PROGETTAZIONE LOGICA DEL SISTEMA DI RICONOSCIMENTO

Prima di passare all'analisi dei flow-chart delle principali funzioni, per comprenderne la logica dobbiamo innanzitutto definire le modalità di utilizzo della memoria interna delle schede e quella della memoria RAM del dispositivo. Una volta chiariti i legami logici tra la scheda e la memoria RAM interna, evidenzieremo la necessità di alcune operazioni di inizializzazione del sistema, alla sua prima accensione; poi sarà immediato comprendere le logiche delle funzioni (scrittura lettura ecc.) del sistema. Definiamo la modalità di utilizzo della memoria interna alla scheda: nella puntata precedente

abbiamo detto che la EM4469 (Fig. 2) ha 11 blocchi (a partire dal blocco 5) da 32 bit (4 byte) a disposizione dell'utente, per un totale di 44 caratteri memorizzabili. Le informazioni che vogliamo memorizzare sulla scheda sono: codice scheda (2 byte, che permettono fino a 65.535 codici scheda), tipo di scheda (flat o a crediti, che occupa 1 byte), data di attivazione (3 byte), data di scadenza (3 byte), credito (per le schede a credito, che occupa 2 byte) e flag per l'eventuale attivazione del relé (1 byte).

Le posizioni di memoria vengono così utilizzate:

- blocco 5: due byte del codice, due byte (eventuale) credito;
- blocco 6: un byte per il tipo di

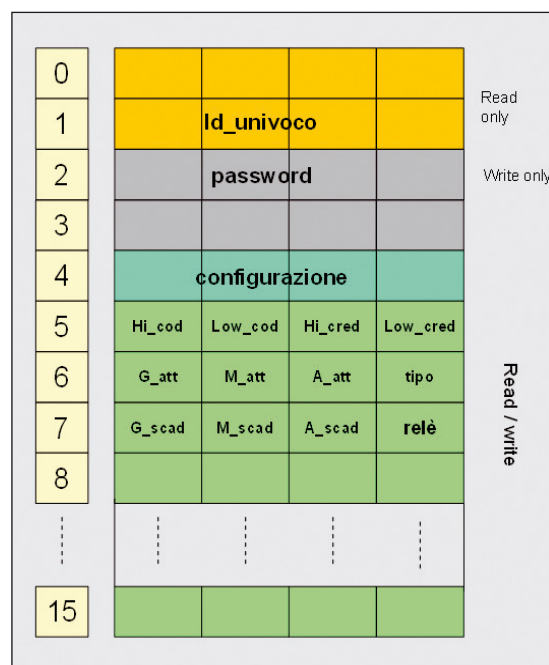


Fig. 2 - Utilizzo della memoria della EM4469.

scheda, tre byte per la data di attivazione;

- blocco 7: un byte per il flag di attivazione relé, 3 byte per la data di scadenza.

Ricordiamo inoltre che ogni scheda è caratterizzata da un identificativo univoco, memorizzato nel blocco 1 (32 bit) che è di sola lettura. Inoltre nel blocco 0 (sempre di sola lettura) 10 bit sono destinati al codice del costruttore; quest'ultimo ha a disposizione 32 bit (più di 4 miliardi!) di codici univoci da assegnare ai tag. La relazione logica che definiamo tra la scheda e la memoria interna del dispositivo è semplice; poi vedremo come implementarla, dal punto di vista tecnico. In fase di programmazione, si attribuisce alla scheda un codice numerico progressivo, a partire da 1, che viene memorizzato in uno dei blocchi disponibili all'utente. Poi si legge l'identificativo univoco della scheda e lo si memorizza, nella RAM del sistema, a partire dalla posizione corrispondente al codice assegnato. Si osservi, infatti, che l'identificativo univoco

è lungo 32 bit e la RAM indirizza su 8 bit. Così nelle posizioni da 1 a 4 della RAM interna ci saranno i 4 byte dell'identificativo della scheda n° 1, nelle posizioni da 5 a 9 l'identificativo della scheda 2 e così via. A questo punto, dopo la programmazione sarà semplice riconoscere le schede: letto il codice e l'identificativo univoco, si verificherà se nella RAM, nella posizione individuata dal codice, è memorizzato quello specifico identificativo univoco.

Si osservi che questo meccanismo introduce un livello di sicurezza elevatissimo: supponiamo, infatti, di voler clonare una scheda e ipotizziamo di conoscere le password di lettura dei blocchi, così siamo in grado di leggere la scheda e di poter replicare su un'altra tutti i dati presenti nell'area "read /write". Non potremo mai, però, modificare l'identificativo univoco, che nel tag clonato sarà diverso dall'id di quello originale. Provando a leggere la scheda clonata, la RAM non troverà—nella posizione del codice scheda— l'id univoco della scheda originale, quindi il tag clonato non verrà riconosciuto. Entriamo nel dettaglio tecnico della gestione della RAM: in essa (256 kbit, suddivisi in 32.000 byte indirizzabili singolarmente)

dobbiamo memorizzare, nelle fasi di scrittura delle schede, gli identificativi univoci dei tag che si programmano (4 byte). In fase di scrittura, il sistema calcola il codice della scheda incrementando di 1 l'ultimo codice assegnato e individuando la prima posizione libera in memoria, successiva all'ultima posizione occupata. Perciò in RAM dovremo sempre conservare queste due informazioni: ultimo indirizzo utilizzato (2 byte) e ultimo codice scheda (2 byte). Ci occorrono 4 byte; la prima posizione libera per l'id univoco della scheda numero 1 potrebbe essere la n° 5 (considerando che le prime 4 posizioni conterranno sempre i dati su ultima posizione e ultimo codice). Tuttavia si è scelto di lasciare libere alcune posizioni iniziali della RAM, per eventuali usi futuri. Si è scelto di memorizzare l'ultimo indirizzo usato nelle posizioni 300 e 301, e l'ultimo codice scheda nelle posizioni 302 e 303. Perciò i 4 byte dell'identificativo della prima scheda memorizzata (scheda n° 1) troveranno posto nelle posizioni 304, 305, 306 e 307; l'identificativo della scheda n° 2 starà nelle posizioni 308, 309, 310 e 311 e così via. Quindi si può dire che, detto UI l'ultimo indirizzo usato per

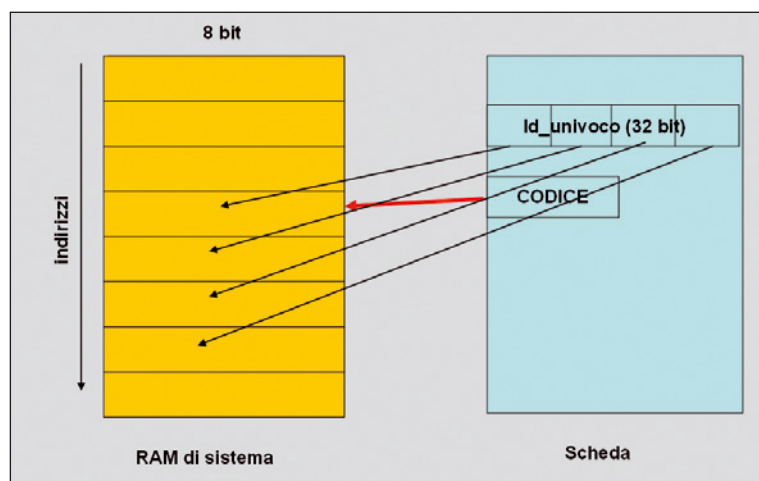


Fig. 3
Relazione
tra codice
scheda,
id_univoco
e RAM di
sistema.

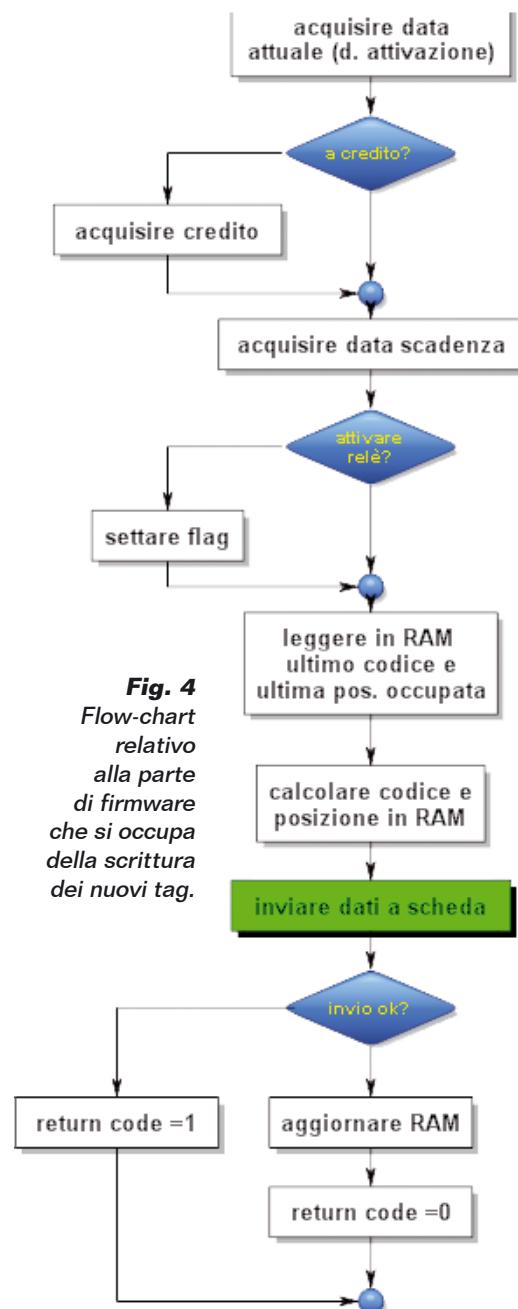
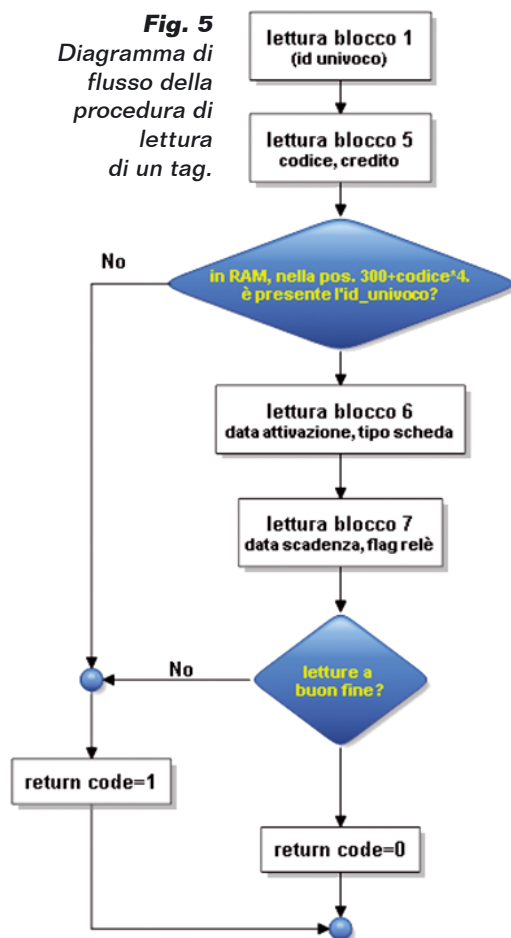


Fig. 4
Flow-chart
relativo
alla parte
di firmware
che si occupa
della scrittura
dei nuovi tag.

il primo byte dell'identificativo dell'ultima scheda memorizzata, l'indirizzo dove memorizzare un nuovo identificativo sarà dato da UI+4. Adottando questo rigido meccanismo di assegnazione di codici e indirizzi, si semplifica la successiva verifica dell'esistenza di una scheda in memoria. Supponiamo -infatti- di ricevere dal lettore RFID l'indicazione di una scheda con codice N: basterà ve-

Fig. 5
Diagramma di
flusso della
procedura di
lettura
di un tag.



rificare che nella posizione "300 + N*4" (nei quattro byte a partire da questa posizione) sia memorizzato l'identificativo univoco della stessa scheda.

Affinché il meccanismo funzioni, dobbiamo inizializzare la RAM alla prima accensione (quella immediatamente successiva alla programmazione del PIC) ed essere certi che il primo codice scheda da utilizzare sia il numero 1 e che la prima posizione di memoria libera sia la 304. Perciò vengono impartite al compilatore alcune istruzioni per la memorizzazione di questi due dati, nelle posizioni da 300 a 303 della RAM. Inoltre si provvede a memorizzare nella EEPROM del PIC il PIN predefinito per l'accesso alle funzioni di programmazione, la password di sistema ed un flag che darà modo al processore di capire, quando si alimen-

ta il circuito, se è la prima volta che si effettua l'accensione. Ci sarà perciò un ramo di codice che viene eseguito, fuori dal ciclo principale, solo alla prima accensione: sarà sufficiente, in questo ramo di codice, inserire un'istruzione che cambia il valore del flag "prima accensione" per garantirci che alle accensioni successive il ramo non sarà percorso. Per inciso (ma ne parleremo in seguito più in dettaglio) questo ramo viene percorso sia alla prima accensione sia ogni volta che si esegue un "reset di inizializzazione" del sistema, in occasione del cambiamento della password di sistema. È facile verificare che i 32.000 registri da 8 bit, escludendo i primi 300 che lasciamo liberi e i 4 destinati a contenere ultimo indirizzo e ultimo codice, permettono di registrare $(32.000-304)/4=7.924$ schede.

SCRITTURA DI UN NUOVO TAG

La scrittura di una nuova scheda avviene comunicando con essa attraverso il modulo ID-20 e secondo i protocolli di comunicazione definiti nella prima puntata. Per ogni blocco da inviare, si carica in un array la sequenza di byte (riga di comando) da spedire all'ID-20. Per la scrittura si impiega la password di sistema. Per una scheda "vergine" i 4 byte della password sono tutti impostati a zero, perciò dovremo (in sequenza):

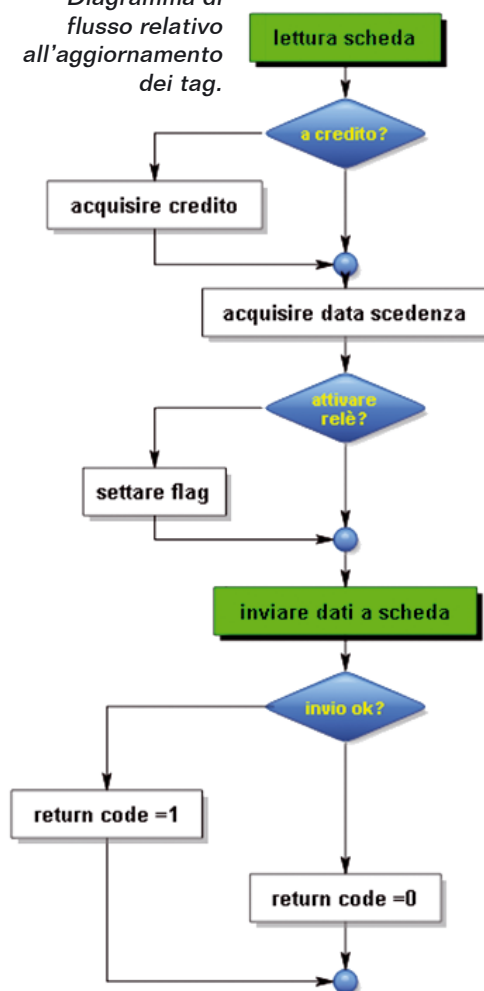
1. effettuare un'operazione di cambio password: da "000000" alla password di sistema (quella che è inizializzata a "111111" nelle EEPROM del PIC); quest'operazione consiste in una scrittura nel blocco 2;
2. informare la scheda dell'intenzione di proteggere con password sia la lettura che la scrittura; si devono settare

a "1" i bit n° 18 e n° 20 della "configuration word" della scheda, al blocco n° 4;

3. scrivere nei blocchi 5, 6 e 7 le informazioni che caratterizzano la scheda (codice, tipo, date, credito ecc.), secondo quanto detto in precedenza;
4. dopo aver registrato i dati sulla scheda, sarebbe opportuna, a rigore, una riletture di tutti i dati per verificare il buon esito della scrittura, tuttavia ci limiteremo a rileggere solo il blocco 5 (codice della scheda e credito).

In sintesi, il flow chart è quello illustrato nella Fig. 4. Attraverso

Fig. 6
Diagramma di
flusso relativo
all'aggiornamento
dei tag.



il display e la tastiera, si guida l'utente nell'immissione dei dati che caratterizzano la scheda (tipo, eventuale credito, data di scadenza, flag di attivazione relé). Si procede con la lettura in RAM dell'ultimo indirizzo utilizzato e dell'ultimo codice scheda, per calcolare nuovo indirizzo e nuovo codice. Poi si effettua una lettura dell'id univoco della scheda e si inviano i dati alla scheda (compiendo le operazioni da 1 a 4) descritte in precedenza. Se l'operazione di scrittura è andata a buon fine, si può finalmente memorizzare in RAM l'id univoco della scheda, all'indirizzo appena calcolato, e aggiornare in RAM il contenuto dei registri che contengono l'ultimo codice scheda e l'ultimo indirizzo utilizzato.

LETTURA DI UN TAG

La lettura di un tag avviene inviando al modulo ID-20 la riga di comando che "interroga" il blocco 1 -nel quale è contenuto l'identificativo univoco- e successivamente il blocco 5, nel quale è contenuto il codice (numero) della scheda. A partire dal codice si calcola la posizione in RAM, dove è presente l'id univoco per la scheda caratterizzata da quel codice e si effettua una lettura, sempre nella RAM, per verificare che il dato e l'id della scheda siano effettivamente gli stessi. A questo punto, se il tag è stato riconosciuto, si possono leggere i rimanenti blocchi (6 e 7) che permettono di recuperare le informazioni necessarie al trattamento della scheda (date, tipo di scheda, ecc.). Il tutto è definito dal flow-chart mostrato nella Fig. 5.

AGGIORNAMENTO DI UN TAG

L'aggiornamento consiste nella modifica della data di scadenza e/o della possibilità di attivare il relé. Inoltre, per le schede a credi-



L'RFId apre la porta al gatto...

Chi ha un gatto o un piccolo cane conosce l'utilità delle botole (comunemente dette "gattaiole") da applicare alla base di porte di ingresso e porte-finestre dei balconi per consentire agli animali domestici di uscire per le proprie necessità senza incomodare il padrone. Queste aperture, composte da uno sportello a molla, sono abbastanza piccole (e poste in una zona lontana dalla serratura) in modo da impedire l'entrata dei ladri, però grandi quanto basta a lasciar entrare animali come volpi



o procioni o anche cani e gatti estranei. Per risolvere il problema dell'intrusione di animali che non sono il cane o il gatto di casa, viene in aiuto la tecnologia RFId, proponendo sportelli dotati di un lettore situato nella porta, che legge il tag inserito nel collare del nostro animale e agisce su un'elettroserratura che sblocca l'apertura della botola. Con questo sistema gli animali domestici hanno assoluta libertà di movimento e i loro proprietari vivono tranquilli, sapendo che in casa entrerà soltanto il loro cane o gatto.

to, è possibile modificare anche il credito residuo. L'aggiornamento consiste in una preliminare lettura della scheda, che permette di guidare nelle successive richieste dei dati da aggiornare. Raccolti i dati (inseriti sempre con la tastiera, guidati da opportune scritte sul display) si procede all'invio alla scheda, tenendo presente che in caso di aggiornamento non si effettueranno le operazioni 1 e 2 (cambio password e settaggio configuration word) che sono necessarie solo alla "prima scrittura". Il tutto è descritto nel flow-chart illustrato nella Fig. 6.

CANCELLAZIONE DI UN TAG

La cancellazione avviene inserendo nel sistema, attraverso la tastiera, il codice (numero) della scheda da cancellare. Non è necessaria la presenza fisica della scheda. Questa caratteristi-

ca consente di cancellare anche tag di cui non si ha la materiale disponibilità (perse, rubate, o in carico a utenti che si desidera cancellare dal sistema). Il sistema verifica che effettivamente nella posizione "300+ codice*4" sia presente un dato diverso dal valore iniziale della RAM, che è costituito da byte settati a FF. Se la scheda è censita, è sufficiente riportare nuovamente ad FF il valore dei 4 registri che contengono l'id univoco. Da questo momento la scheda non sarà più riconosciuta. Sarà tuttavia possibile riutilizzarla, per memorizzare nuovi codici scheda. Il sistema non prevede il riutilizzo delle posizioni di memoria che vengono "liberate" dalla cancellazione di schede. Perciò, precisando l'affermazione fatta in precedenza, possiamo dire che si gestiscono complessi-

```

Hserout [b0, b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12, b13, b14, b15]
HSerIn 100,scrittura_ko,[Wait($02), d0, ret_cod, d1, d2, d3]
return_code=0
goto fine
scrittura_ko:
Return_code=1
fine:

```

Fig. 7 - Modalità di scrittura ad elevato rischio di KO.

```

i=0
Ciclo:
i=i+1
If i=4 then
    goto scrittura_ko
endif
Hserout [b0, b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12, b13, b14, b15]
HSerIn 100,ciclo,[Wait($02), d0, ret_cod, d1, d2, d3]
return_code=0
goto fine
scrittura_ko:
Return_code=1
fine:

```

Fig. 8 - Modalità di scrittura "ridondata", a basso rischio KO.

vamente fino a 7.924 schede, fra attive e cancellate.

LA PASSWORD DI SISTEMA

Più volte si è parlato della password di sistema, con la quale si compiono le operazioni di scrittura e lettura delle schede. Alla prima accensione la password è costituita da una sequenza di 6 "uno". È consigliabile cambiare questa password, per personalizzare il sistema e ridurre la probabilità di successo di tentativi di "aggiornamento fraudolento" dei dati delle schede. Conviene effettuare l'operazione subito dopo la prima accensione e prima di cominciare a memorizzare le schede; infatti il cambio della password di sistema rende assolutamente illeggibili (e quindi irricognoscibili) tutte le schede "scritte" con la vecchia password (è impossibile sia leggerle che scriverle). Al contrario delle schede "cancellate", che -si veda il paragrafo sulla cancellazione- possono essere riutilizzate per memorizzare nuovi codici, le schede scritte con una password

diversa da quella in uso nel sistema sono totalmente inutilizzabili, non essendo possibile effettuare l'operazione di "cambio password" sulla scheda (punto 1 della fase di scrittura). Quell'operazione, infatti, è eseguibile solo su schede vergini, per le quali è noto il codice di partenza ("000000").

Per cambiare la password di sistema si deve collegare a massa la porta 40 del PIC e accedere alle funzioni di menu (pressione del tasto asterisco). Con la porta 40 a potenziale zero, anziché il menu standard viene proposta la richiesta di inserimento della nuova password di sistema.

CONSIDERAZIONI SULLA SICUREZZA

A questo punto possiamo tornare sul discorso "sicurezza" e spiegare il motivo che spinge ad utilizzare la password nell'accesso alla lettura e scrittura dei blocchi della scheda: in precedenza abbiamo visto che è praticamente impossibile clonare una scheda, perché anche la conoscenza della password di sistema non è sufficiente; quindi si potrebbe pensare che questa password è inutile. In effetti, se la clonazione è impossibile, non è altrettanto difficile effettuare un *aggiornamento illecito* dei dati. Se scrivessimo i dati caratteristici della scheda senza utilizzare la pwd di sistema, sarebbe facile dall'esterno, conoscendo le logiche di memorizzazione (corrispondenza dati - blocchi di memoria della scheda) caricare

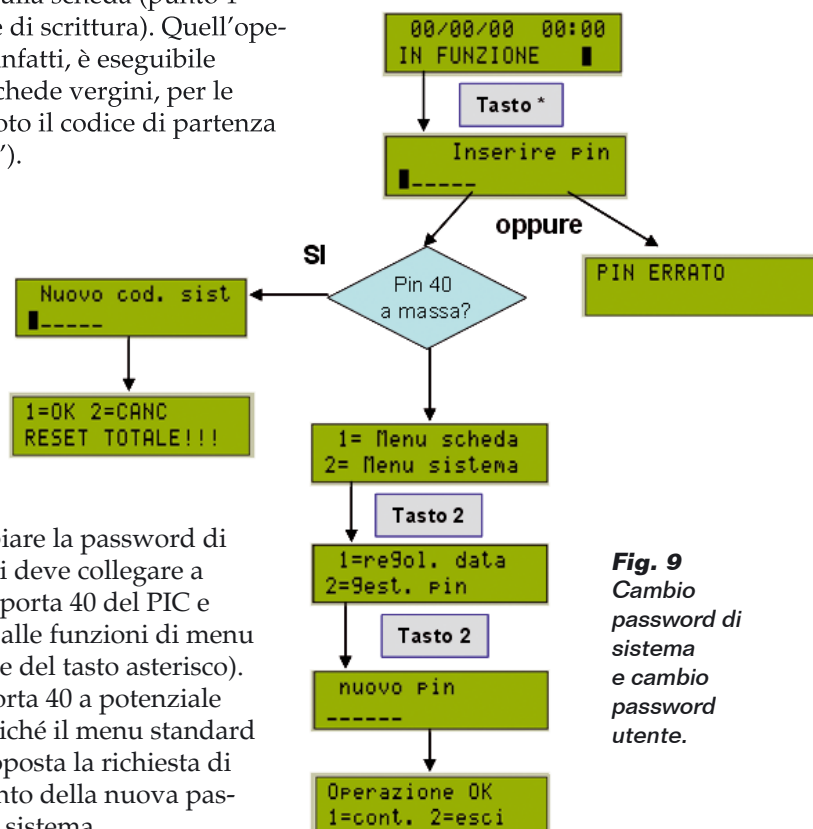


Fig. 9
Cambio
password di
sistema
e cambio
password
utente.

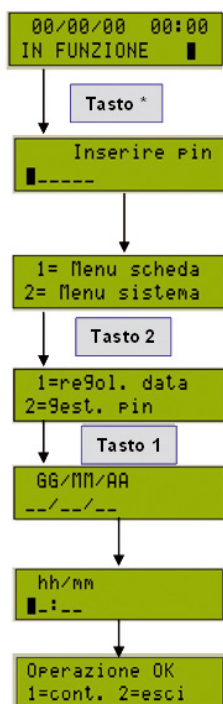


Fig. 10 - Regolazione data e ora.

su una scheda "ufficiale" i desiderati valori di data di scadenza, credito, attivazione relé. Ecco perciò che torna utile la pwd di sistema: anche conoscendo le logiche di memorizzazione (posizione delle informazioni nei blocchi), non è possibile caricare le informazioni se non si conosce la password. Però è bene sottolineare che la password di lettura/scrittura non protegge al 100 % dai possibili aggiornamenti illeciti, dal momento che la password non è leggibile dalla scheda ma tuttavia viaggia *in chiaro* nella comunicazione tra modulo RFID e scheda. Dalla lettura dei precedenti paragrafi e in particolare dall'analisi dello schema a blocchi di una scheda, si intuisce che non è facile ma nemmeno troppo difficile costruire una tag "a componenti discreti", in grado di emulare un tag reale e di colloquiare con un modulo di lettura/scrittura per carpire i dati che vengono emessi dal modulo, tra i quali è compresa la password di lettura/scrittura.

In generale, si può dire che per ottenere la massima sicurezza in sistemi RFID, è consigliabile memorizzare i dati significativi (nel nostro caso credito e scadenza) non solo sulla scheda ma anche nella RAM di sistema, che dovrebbe costituire il riferimento per la valutazione dei criteri di validità della scheda. I sistemi high-end -di cui si è parlato nella puntata introduttiva all'RFID- superano questo problema adottando evolute tecniche di crittografia.

RESTANTI BLOCCHI DI CODICE

Abbiamo descritto i flow-chart relativi alle principali funzioni del software, tuttavia il programma di gestione comprende numerose altre componenti: acquisizione di dati da tastiera, gestione del display, acquisizione del PIN, cambio PIN, acquisizione data di sistema, acquisizione data di scadenza, acquisizione del credito, modifica della password di sistema, verifica della scadenza, attualizzazione del credito, gestione dei menu; sono le funzioni evidenziate in giallo nei diagrammi di flusso. In alcuni casi si tratta di algoritmi abbastanza semplici, mentre in altri (ad esempio acquisizione dei dati da tastiera, gestione dell'orologio) si parla di routine che meriterebbero maggiore approfondimento se non addirittura un articolo specifico, come nel caso del Real Time Clock realizzato col DS 1302. Tuttavia, per quanto interessanti, le funzioni descritte non sono strettamente connesse all'impiego della tecnologia RFID. Pertanto i citati blocchi funzionali non saranno analizzati nel dettaglio. Su una parte del codice è invece utile soffermarsi fino al dettaglio: quella relativa all'invio/ricezione dei dati alla/dalla

scheda. In linea di principio, la richiesta di scrittura (o lettura) potrebbe esaurirsi in poche righe di codice: si invia un array di byte attraverso la porta USART del PIC e si resta in attesa di una risposta, attendendo un tempo ragionevole (100÷300 ms) oltre il quale è lecito ritenere che l'operazione non sia andata a buon fine. Tuttavia si è sperimentalmente osservato che con l'adozione di questo semplice meccanismo, i casi di mancata ricezione della risposta sono molto più frequenti di

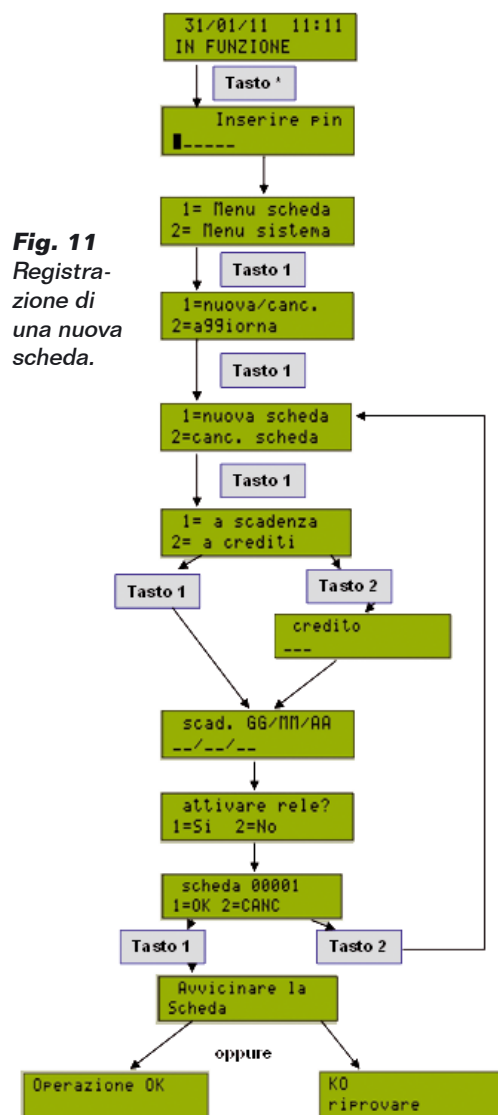


Fig. 11
Registrazione di una nuova scheda.

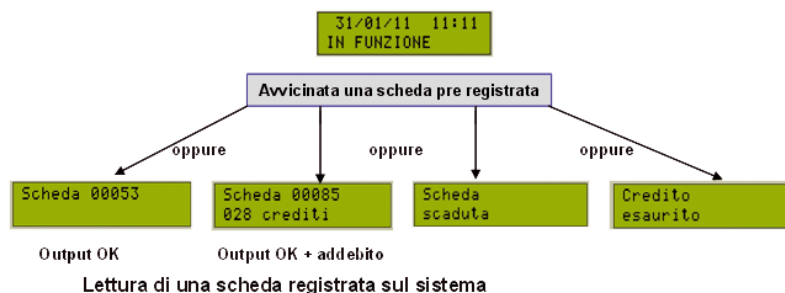
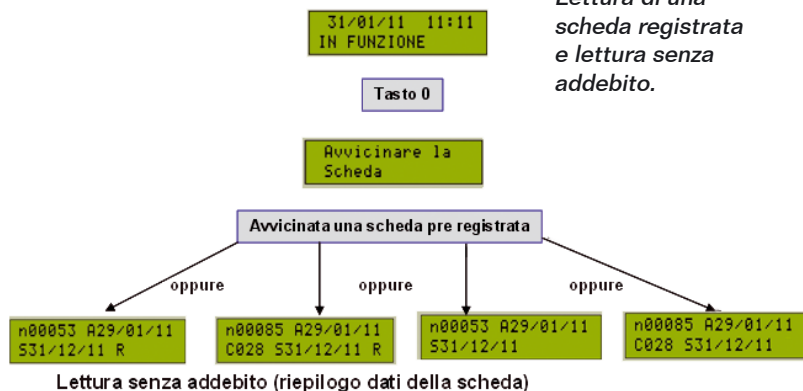


Fig. 12
Letture di una
scheda registrata
e lettura senza
addebito.



quanto sarebbe lecito attendersi. Probabilmente intervengono errori di trasmissione che impediscono il regolare trattamento del comando; si ricordi che nella riga di comando è presente un carattere di controllo (che è l'XOR dei caratteri precedenti nella riga) che ha proprio lo scopo di impedire che errori di trasmissione possano indurre comandi diversi da quelli voluti. Se l'XOR di un certo numero di caratteri ricevuti al modulo ID-20 non coincide con l'XOR "calcolato" all'atto dell'invio, qualcosa non ha funzionato nella trasmissione e il comando viene bloccato (situazione di KO: Fig. 7). Per superare il problema si è ragionato in termini probabilistici (Fig. 8): difficilmente un certo comando "fallirà" tre volte consecutive. Si invia il comando al modulo ID-20 e se dopo 100 ms non si è ricevuto l'ack, si rinvia il comando. Si esce dal ciclo di invii appena si riceve l'ack o se falliscono tre invii consecutivi (nel qual caso fallisce

tutta l'operazione, che in genere prevede la lettura o scrittura di più blocchi in sequenza). Con questo accorgimento, le "mancate letture" sono praticamente impossibili (anche perché il dispositivo è in costante lettura e una mancata lettura si traduce in un tempo di lettura maggiore) e le mancate scritture (KO di scrittura o aggiornamento) sono estremamente rare.

MONTAGGIO DEL CIRCUITO

Bene, spiegata la teoria, vediamo la pratica: il montaggio del lettore non presenta particolari difficoltà, perché il circuito stampato è a singola faccia e non sono utilizzati componenti a montaggio superficiale. L'unica nota di rilievo riguarda la connessione del quarzo per il Real Time Clock DS1302, che andrà montato dal lato saldature, direttamente sui piedini dell'integrato, e poi sarà ripiegato in direzione della piastra. Attenzione a individuare correttamente "righe" e "colonne" della tastiera,

da collegare al PCB con corti spezzoni di cavo flessibile. La tastiera ed il display LCD andranno montati su distanziatori. Conviene fissare i distanziatori ai componenti e successivamente incollare gli estremi liberi dei distanziatori sul PCB, con colla cianoacrilato, nella posizione riportata nella foto di presentazione del dispositivo. L'antenna può essere autocostruita avvolgendo 40 spire di filo di rame smaltato (diametro 0,16 mm) a formare un rettangolo di cm 7,2 x cm 5,2. Una volta completato l'avvolgimento, si consiglia di ricoprirlo con comune smalto per unghie, che essiccandosi conferirà all'antenna la necessaria rigidità.

IMPIEGO PRATICO

Una volta montato il circuito e

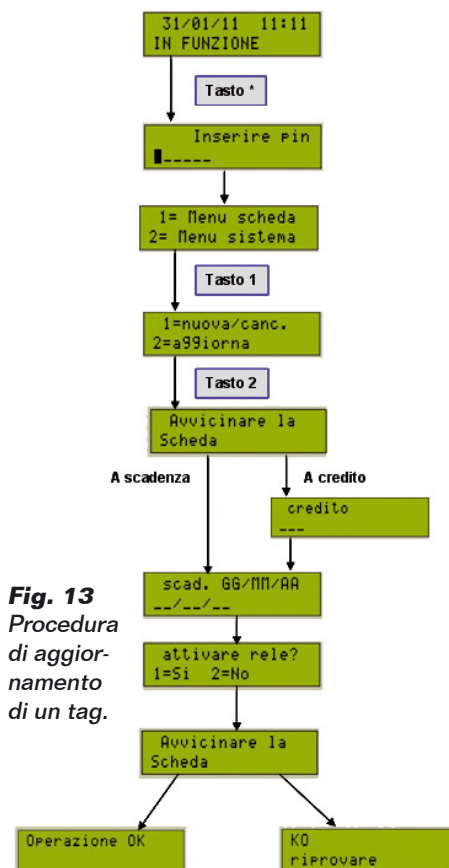


Fig. 13
Procedura di aggiornamento di un tag.

programmato il PIC, il dispositivo è pronto all'uso. Il tasto **asterisco** controlla l'accesso alle funzioni di programmazione, mentre il **cancellotto** funge da tasto di ritorno al menu precedente, quando previsto.

Nelle figure dalla 9 in poi riportiamo le principali schermate che il display propone nelle seguenti fasi d'uso: inizializzazione, scrittura, lettura, aggiornamento, cancellazione, lettura dati senza addebito.

Dopo l'accensione, si consiglia innanzitutto di cambiare la password di sistema (quella predefinita è 111111). Cortocircuitando a massa il piedino 40 del PIC e premendo il tasto **asterisco**, dopo l'inserimento della password utente (quella predefinita è 123456) si ha la possibilità di inserire la nuova password di sistema. Tolto il cortocircuito e ripremendo il tasto **asterisco**, si può rientrare nel menu di sistema e modificare anche la password utente (Fig. 9).

Il menu di gestione del sistema permette di regolare la data e l'ora (Fig. 10), per consentire la corretta gestione delle date di scadenza.

Per registrare una scheda (Fig. 11), premendo **asterisco** e inserendo la password, si potrà accedere ai menu di scrittura, che permettono di scegliere il tipo di scheda. Per schede a scadenza verrà chiesto l'inserimento della data di scadenza e la modalità d'uso del relé; per schede a credito sarà possibile inserire il credito iniziale.

L'accensione del LED rosso segnala la fase di registrazione (invio dati alla scheda).

La lettura delle schede avviene avvicinando la scheda tag al lettore; se il sistema la riconosce tra quelle preventivamen-

te "registrate" nella sua RAM, viene visualizzato il numero di scheda e sono avviate le operazioni di output (segnale positivo all'uscita "scheda ok" ed eventuale chiusura contatti relé, per 0,5 secondi). In caso di schede a credito, viene visualizzato anche il credito residuo. Infine, sul display, all'occorrenza, vengono visualizzate le situazioni di "scheda scaduta" o "credito esaurito".

L'aggiornamento avviene (sempre dopo aver acceduto alle funzioni di programmazione tramite il tasto **asterisco**) avvicinando la scheda, che viene riconosciuta e di conseguenza vengono proposti gli opportuni messaggi di acquisizione di data, modalità di uso del relé e eventuale credito. Il LED rosso segnala la registrazione. Nelle fasi di registrazione e aggiornamento, il sistema controlla l'effettiva presenza della scheda da registrare/aggiornare. In caso di anomalia (o se non si avvicina la scheda al dispositivo) viene segnalato l'errore corrispondente.

In modalità di funzionamento normale (lettura), premendo il tasto "zero" e avvicinando la scheda al lettore, viene mostrata la consistenza della scheda: numero, data di attivazione, data di scadenza, credito per le schede a credito.

Se la scheda deve attivare il relé, alle informazioni suddette si aggiungerà una R, alla fine della seconda riga. Questa lettura non decrementa l'eventuale credito presente sulla scheda ma allo stesso tempo non attiva le uscite (relé e segnale di "scheda ok").

La cancellazione richiede solo l'inserimento del numero di scheda da cancellare. Se si introduce il numero di una sche-

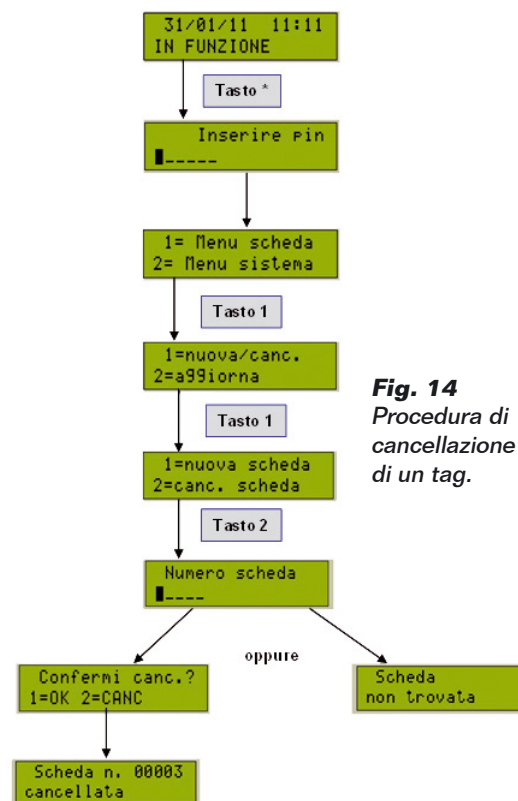


Fig. 14
Procedura di
cancellazione
di un tag.

da non presente in memoria (perché mai registrata o perché già cancellata precedentemente) il sistema informa dell'assenza della scheda in memoria. ■

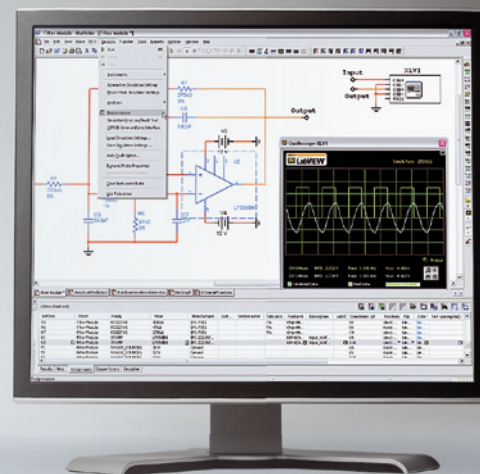
per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono di facile reperibilità. Il circuito stampato ed il firmware sono scaricabili gratuitamente dal sito della rivista (www.elettronica.in.it). Il modulo lettore/scrittore ID-20LFWR costa 85,00 Euro mentre le tessere riscrivibili TAG-EM4469 costano 4,50 Euro cadauna. Sono anche disponibili la tastiera (Cod. TST12, Euro 5,30), il display seriale (LCDSER16X2NV, Euro 20,00) ed il micro già programmato (MF935, Euro 18,00). Tutti i prezzi sono comprensivi di IVA.

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

Conoscere e usare NI Multisim

Progettazione più semplice,
migliore simulazione.



Parliamo del myDAQ, un dispositivo di acquisizione dati portatile dotato di propri strumenti di misura, capace di interagire con LabVIEW e Multisim, che offre agli studenti e agli appassionati di elettronica la possibilità di prototipare sistemi e di analizzare circuiti elettronici senza bisogno di un laboratorio. Quinta puntata.

di Filippo Spadaro

Il myDAQ è un dispositivo di acquisizione dati portatile (DAQ), a basso costo, che utilizza strumenti software sviluppati in tecnologia LabVIEW, consentendo di misurare ed analizzare i segnali elettrici provenienti dal mondo reale. È utilizzabile in combinazione con LabVIEW e con Multisim, ma anche in modalità autonoma (sempre collegato al PC), come strumento di misura ed analisi di circuiti elettronici, dato che è corredato di vari strumenti virtuali tra cui un tester, un alimentatore in continua, un generatore di segnali ed un oscilloscopio.

Combinando il myDAQ con LabVIEW è possibile interfacciare agevolmente trasduttori e attuatori per acquisire misure da elaborare via software e programmare sistemi di controllo ed attuazione di semplici processi gestiti dal computer.

Il suo abbinamento con Multisim offre inoltre l'opportunità di poter confrontare e valutare lo scostamento tra la risposta attesa da un circuito simulato al computer ed il funzionamento di un prototipo reale, realizzato su breadboard o assemblato su circuito stampato, il che permette di prototipare i dispositivi elettronici.

Prima di utilizzare il myDAQ è necessario installare sotto il sistema operativo Windows a 32 bit i driver NI ELVISmx e NI DAQmx dal DVD fornito con il dispositivo, all'interno del quale è anche presente una copia degli applicativi NI LabVIEW e NI Multisim, entrambi forniti con licenza Student Edition. Successivamente, ogni volta che il myDAQ verrà collegato mediante il cavo USB al computer, sarà riconosciuto e pronto per l'uso. Sul sito National Instruments sono disponibili una guida per l'installazione (software e hardware) e la spiegazione di come effettuare le misure usando gli strumenti software di NI ELVISmx e NI LabVIEW.

La confezione del dispositivo comprende, oltre al myDAQ e al DVD di installazione, anche un cavo USB per collegarlo al PC, due sonde per DMM (multimetro digitale), un cavo audio (3,5 mm), un connettore con terminali a vite ed un cacciavite. La scatola è un contenitore riutilizzabile, realizzato in scomparti, che permette di tenere in ordine il myDAQ insieme ai cavi, al cacciavite, al DVD in dotazione e ad eventuali componenti elettronici e accessori aggiuntivi, oltre ad offrire uno spazio



Fig. 1 - Vista laterale del myDAQ con il connettore con 20 terminali a vite e i connettori audio IN/OUT.

sagomato per ospitare una breadboard (non in dotazione) e di altezza adeguata per poter tenere un circuito elettronico montato su di essa.

CARATTERISTICHE HARDWARE

Per funzionare, il myDAQ deve essere collegato al PC tramite una porta USB e fornisce tensione sufficiente per alimentare circuiti semplici e sensori. Dispone di tre alimentatori che forniscono le seguenti tensioni:

- **+15 V** e **-15 V** possono essere utilizzati per alimentare componenti analogici di potenza, come amplificatori operazionali e regolatori lineari;
- **+5 V** può essere utilizzato per alimentare componenti digitali, come ad esempio i dispositivi logici.

Queste alimentazioni, insieme alle relative masse, analogica **AGND** e digitale **DGND**, sono prelevabili dal connettore con 20 terminali a vite presente lungo uno dei lati del dispositivo (**Fig. 1**).

Il myDAQ offre otto linee digitali di ingresso e uscita, **DIO**, le quali sono compatibili TTL a bassa tensione (LVTTTL) 3,3 V e tollerano anche 5 V, mentre in uscita non sono compatibili con livelli logici CMOS a 5V. Ciascuna linea digitale è configurabile individualmente in ingresso o in uscita, può essere temporizzata via software per interfacciare circuiti digitali sia TTL a bassa tensione, sia TTL a 5 V.

Due canali analogici differenziali di ingresso, **AI**, permettono al myDAQ di acquisire segnali dall'esterno. Ciascuno di questi canali può essere configurato come ingresso di tensione differenziale general-purpose ad alta impedenza o come ingresso audio. I due ingressi analogici sono multiplexati verso un unico convertitore analogico-digitale (ADC) con velocità di acquisizione pari a 200 kS/s (kSamples/secondo, ossia migliaia di campioni al secondo) per canale. In modalità di

impiego generale, si possono acquisire segnali nell'intervallo tra ± 10 V.

Gli ingressi analogici sono utilizzati da alcuni degli strumenti a disposizione del myDAQ: l'oscilloscopio, l'analizzatore di segnale dinamico e il Bode analyzer.

Due canali di uscita analogici, **AO**, possono essere configurati come uscita general-purpose di tensione o uscita audio. Entrambi i canali hanno un convertitore digitale-analogico (DAC) dedicato, in modo che sia possibile aggiornare simultaneamente tali uscite. In modalità di impiego generale (general-purpose) è possibile generare segnali fino a ± 10 V. In modalità audio, i due canali rappresentano le uscite stereo left e right.

Il dispositivo presenta anche due connettori audio per l'ingresso stereo e l'uscita stereo, rispettivamente **AUDIO IN** e **AUDIO OUT**, interfacciabili con cavo audio da 3,5 mm.

La potenza totale disponibile in uscita al myDAQ, considerando tutte le uscite impiegate, sia quelle analogiche che quelle digitali, sia di alimentazione che di segnale, è limitata a 500 mW e tipicamente il dispositivo eroga 100 mW.

NI Instrument ELVISmx Launcher

Analizziamo gli strumenti di misura di cui dispone il myDAQ, ciascuno dei quali è corredato di una propria interfaccia software in ambiente Windows che può essere avviata usando la comoda **NI ELVISmx Instrument Launcher** (**Fig. 2**) che li raggruppa tutti. Per avviarla, selezionare **Start > Tutti i programmi > National Instruments > NI ELVISmx for NI ELVIS & NI myDAQ > NI Instrument ELVISmx Launcher**.

MULTIMETRO DIGITALE



Il primo strumento di misura che consideriamo è il multimetro digitale, **DMM**, che permette di effettuare misure di tensione e corrente AC e DC, resistenza, tensione di diodo e continuità, i cui connettori (**Fig. 3**) sono posti su uno dei lati più corti del myDAQ e il cui utilizzo è uguale a quello di un qualunque multimetro. L'interfaccia del DMM è intuitiva e permette

Fig. 2 - NI ELVISmx Instrument Launcher



Tabella 1

di regolare le impostazioni di misura, oltre a illustrare la modalità di collegamento dei cavi ai relativi connettori. Lo strumento ha una risoluzione (intesa come numero di cifre significative mostrate sul display) di 3 cifre per la parte intera e 5 per i decimali. Agendo sull'impostazione **Device** è possibile scegliere **Simulate NI myDAQ** per effettuare misure in simulazione sugli schemi elettrici di Multisim, dopo aver collegato con dei fili i suoi connettori al circuito, oppure scegliere **Dev1 (NI myDAQ)** per analizzare un circuito elettronico reale mediante le sonde in dotazione. Sono possibili le opzioni di misura illustrate, con i relativi simboli, nella **Tabella 1**.

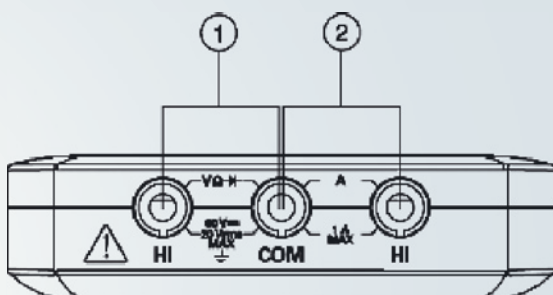
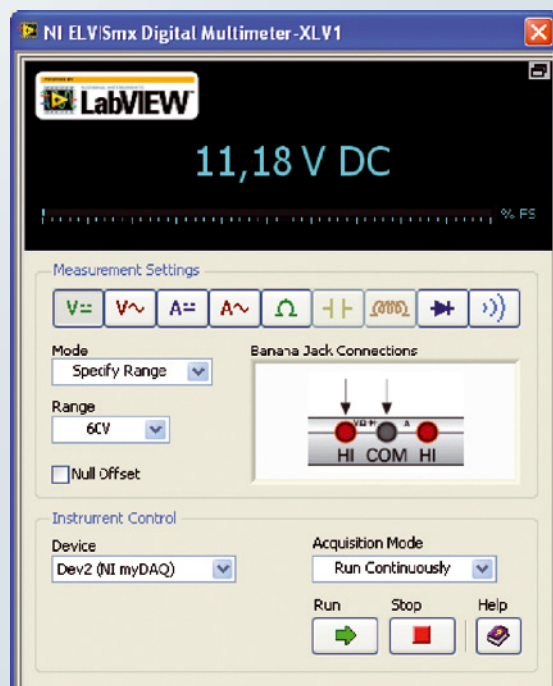


Fig. 3 - Vista laterale del myDAQ con (1) i connettori per le misure di tensione/resistenza/tensione di diodo/continuità e (2) i connettori per le misure di corrente.

Fig. 4 - Interfaccia software del DMM.



	Tensione DC , misura la componente continua della tensione, fino a 60V; è possibile eseguire la misura in modalità Auto o impostando il range manualmente su 60 V, 20 V, 2 V e 200 mV.
	Tensione AC , misura la componente alternata della tensione, in valore efficace fino a 20 Vrms; misura in modalità Auto o impostando il range manualmente su 20 V, 2 V e 200 mV.
	Corrente DC , misura la componente continua della corrente, fino a 1 A (è presente un fusibile interno di protezione 1,25A, il manuale del myDAQ riporta la procedura di test e di sostituzione); modalità di misura Auto o manuale, impostando il range su 1 A, 200 mA, e 20 mA.
	Corrente AC , misura la componente alternata della corrente, in valore efficace fino a 1 Arms; misura in modalità Auto o manualmente su 20 mA, 200 mA, 1 A.
	Resistenza misurata impostando il range manualmente su 20 MΩ, 2 MΩ, 200 kΩ, 20 kΩ, 2 kΩ e 200 Ω oppure in modalità Auto.
	Capacità .
	Induttanza .
	Diodo , misura la caduta di tensione su un diodo; la massima tensione di soglia per una misura valida è di 2V.
	Continuità , quando si misura la continuità, il computer emette un segnale acustico; la soglia ohmica per misure di continuità è 15 Ω.

Per avviare il DMM è necessario dare il **Run** e al termine della misura è importante cliccare su **Stop**, specialmente nel caso si stiano eseguendo misure di corrente che possano far saltare il fusibile di protezione del dispositivo.

ESERCITAZIONE 1

Si vuole montare su breadboard il semplice circuito resistivo con un LED riportato nella **Fig. 5**, allo scopo di effettuare alcune misure elettriche. Gli obiettivi sono:

- fare pratica con NI myDAQ Design simulando un circuito elettronico in ambiente Multisim 11;
- effettuare misure elettriche, mediante il DMM del myDAQ, sul circuito montato su breadboard;
- confrontare i dati acquisiti in simulazione con le misure reali.

I componenti necessari per approntare il circuito dell'esercitazione sono i seguenti: R1: 100 Ω 5% (¼ W), R2: 560 Ω 5% (¼ W), LED: $V_f = 1,83$ V, $I_f = 20$ mA. La strumentazione richiesta è: Alimentazione +15V e GND, DMM. È possibile simulare un circuito che utilizza il

Fig. 5 - Circuito proposto per l'esercitazione 1 e la misura di tensione ai capi del resistore R_2 sfruttando il DMM del myDAQ in simulazione.

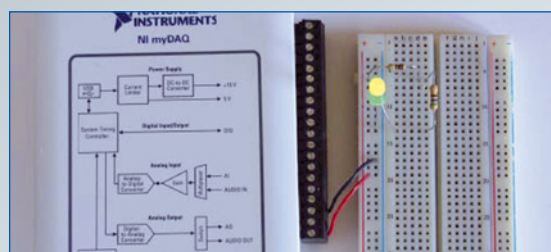
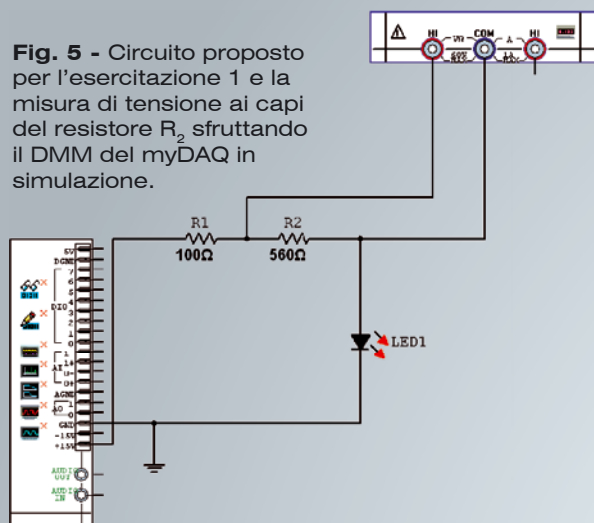


Fig. 6 - Circuito proposto per l'esercitazione 1, montato su breadboard e alimentato dal myDAQ a +15V.

myDAQ aprendo un file nuovo in Multisim 11, selezionando **New\NI myDAQ Design**. Pertanto, anche non disponendo del myDAQ è comunque possibile simularlo in ambiente Multisim, utilizzando la strumentazione elettronica di cui dispone, a patto di installare i relativi driver scaricabili via Internet dal sito NI (www.ni.com).

Nel disegnare lo schema elettrico di Fig. 5 non si dimentichi di collegare il connettore GND del myDAQ a terra, altrimenti la simulazione non funziona. Si passa quindi a rilevare le varie misure elettriche avviando il DMM direttamente da Multisim 11, con un doppio clic sul relativo pulsante di avvio  riportato accanto ai connettori di misura.

Si eseguono varie misure di tensione ai capi dei resistori, avviando la simulazione in ambiente Multisim, come nel caso riportato nella Fig. 5, dove viene rilevata la tensione ai capi di R_2 , che vale 11,18 V. Allo stesso modo si procede ad effettuare misure sul circuito montato su breadboard (Fig. 6), confrontandole poi con le precedenti.

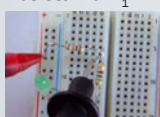
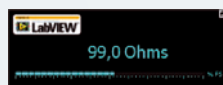
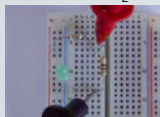
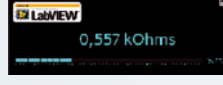
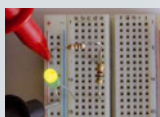
Si può provare che il principio di Kirchhoff delle tensioni, applicato al circuito è verificato: $V_E = V_{R1} + V_{R2} + V_{LED} \Rightarrow 14,36 = 1,84 + 10,29 + 2,23$

Inoltre, la potenza elettrica dissipata su ognuno dei resistori è pari a:

$$P_{R1} = 10,29 \cdot 1,84 \text{ m} = 18,9 \text{ mW}$$

$$P_{R2} = 10,29 \cdot 18,48 \text{ m} = 190,1$$

Alcune delle misure rilevate sul circuito dell'esercitazione 1.

Tipo di misura	Impostazioni	Valori rilevati
Tensione di soglia del diodo LED 	 selezionare diodo e scegliere misura in modalità Auto o con range 2V; del resto il LED adoperato nel circuito reale ha una tensione diretta V_f inferiore al soglia limite dello strumento. Il LED scelto in simulazione ha una V_f prossima a quella del dispositivo reale adoperato.	Misura in simulazione:  Misura nel circuito reale: 
Resistenza R_1 	 selezionare resistenza e scegliere misura in modalità Auto o con range 200ohm; la misura viene effettuata a circuito non alimentato.	Misura in simulazione: il myDAQ non è abilitato a misure di resistenza sullo schematico (si può comunque utilizzare il DMM di Multisim); il resistore ha comunque un valore di resistenza a noi noto: 100 Ω Misura nel circuito reale: 
Resistenza R_2 	 selezionare resistenza e scegliere misura in modalità Auto o con range 2kohm; la misura viene effettuata a circuito non alimentato.	Misura in simulazione: valore noto: 560 Ω Misura nel circuito reale: 
Tensione di alimentazione 	 selezionare tensione DC e scegliere misura in modalità Auto o con range 20V; la misura va effettuata a circuito alimentato. La differenza tra la tensione misurata in simulazione e quella sul circuito reale è dovuta all'alimentatore ± 15 V del myDAQ che può fornire una tensione compresa tra un massimo di 15,3 V (-15,3 V per la sorgente negativa) in condizione di carico assente e un minimo di 14,0 V (-14,0 V per la sorgente negativa) a pieno carico.	Misura in simulazione:  Misura nel circuito reale: 

mW, il che ci consente di utilizzare per l'esercitazione resistori da $\frac{1}{4}$ di watt.

OSCILLOSCOPIO



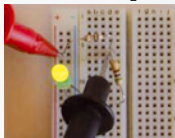
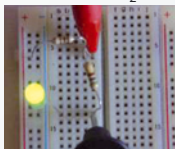
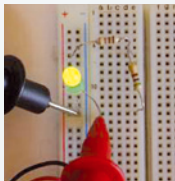
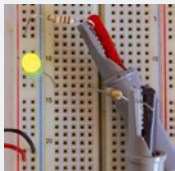
Analizziamo un altro importante strumento disponibile con il myDAQ: l'oscilloscopio (Scope), il quale visualizza le tensioni disponibili su due ingressi (**Fig. 7**), indicati con Channel 0 (CH 0) e Channel 1 (CH 1), per l'analisi nel tempo. Lo Scope del myDAQ offre le funzionalità standard di un tipico oscilloscopio da laboratorio. Come ingressi è possibile utilizzare i canali AI (AI 0 e AI 1) o i canali AudioInput

(AudioInput Left e AudioInput Right), ma non una combinazione di entrambi. Dopo aver scelto gli ingressi, nel menu a discesa Source, per abilitare l'acquisizione dei dati su ciascun canale è necessario spuntare la relativa casella **Enabled**; in caso contrario l'acquisizione dei dati su quel canale è disabilitata. La massima velocità di campionamento (o massima frequenza di campionamento, se preferite) disponibile per AI e AudioInput è pari a 200 kS/s (kilo samples per second), valida nel caso sia configurato un solo canale, sia quando sono configurati entrambi i canali di input. La banda passante, ossia l'intervallo di

frequenze in cui l'oscilloscopio del myDAQ funziona correttamente, è limitata a quella audio, compresa tra 0,2÷20 kHz (per bande più estese è necessario utilizzare i sistemi NI ELVIS I, NI ELVIS II e NI ELVIS II+).

L'oscilloscopio dispone di:

- due manopole **Volts/Div** (una per canale) con regolazione a scatti, per la sensibilità dell'asse y, e valori compresi tra 10 mV e 20 V;
- due manopole **Vertical Position (Div)** (una per canale), per regolare con continuità la posizione dell'asse y;
- una manopola per **Time/Div** per la regolazione della base temporale, lungo l'asse x, con andamento a scatti per valori compresi tra 5 μ s e 200 ms (range valido sia per segnali acquisiti dai connettori sia AI che AudioInput);
- il comando di **Trigger** consente la visualizzazione stabile dell'immagine: sono supportati la posizione

Tipo di misura	Impostazioni	Valori rilevati
Tensione su R_1 	 selezionare tensione DC e scegliere misura in modalità Auto o con range 2V; la misura va effettuata a circuito alimentato. La differenza tra la tensione misurata in simulazione e quella sul circuito reale è dovuta alle tolleranze dei resistori (5%) e al punto di lavoro I-V del LED che può non corrispondere con la simulazione.	Misura in simulazione:  Misura nel circuito reale: 
Tensione su R_2 		Misura in simulazione:  Misura nel circuito reale: 
Tensione sul diodo LED 	 selezionare tensione DC e scegliere misura in modalità Auto o con range 20V; la misura di tensione è effettuata a circuito alimentato, inserendo i puntali tra anodo e catodo del diodo. Si nota una differenza del punto di lavoro del diodo LED, tra quanto simulato e la realtà, dipendente dalla reale caratteristica volt-amperometrica del LED utilizzato.	Misura in simulazione:  Misura nel circuito reale: 
Corrente 	 selezionare corrente DC e scegliere misura in modalità Auto o con range 20mA; la misura va effettuata ponendo l'amperometro in serie al circuito. Per non rischiare di bruciare il fusibile è bene effettuare l'inserzione dell'amperometro a circuito non alimentato, ad esempio usando delle sonde con morsetti a coccodrillo (come in figura) per agganciare i componenti elettronici nel punto in cui si è aperto il circuito per l'inserzione in serie, quindi collegare l'alimentazione e avviare il Run sul DDM.	Misura in simulazione:  Misura nel circuito reale: 



Immediate che avvia l'acquisizione immediatamente e **Edge** in cui è presente un controllo di livello **Level** con cui è possibile specificare una soglia di trigger (in Volt), e un controllo **Horizontal Position**, variabile tra 0÷100%, che specifica la posizione del trigger come percentuale del tempo totale di acquisizione (il valore predefinito è 50%); il Digital trigger non è supportato per il myDAQ ed è disabilitato;

- **Coupling**: i canali AI supportano soltanto l'accoppiamento in DC (viene misurato l'intero segnale) mentre i canali AudioInput soltanto l'accoppiamento in AC (viene rimosso l'offset in continua dal segnale);
- **Probe** seleziona il valore della sonda per il relativo canale. Il valore di questo controllo è sempre 1x (per il myDAQ non è possibile impostare sonde con attenuazioni 10x, opzione disponibile invece per i NI ELVIS);
- opzione **Autoscale**, che permette una regolazione automatica via software della scala di tensione sul display in base alla tensione picco-picco del segnale AC, per la visualizzazione ottimale del segnale stesso.

GENERATORE DI FUNZIONI

Il myDAQ dispone di un generatore di funzioni FGGEN (Fig. 8) che genera forme d'onda standard in uscita dal canale AO 0, nella banda di frequenze 0,2 ÷ 20 kHz; queste forme sono:

-  = sinusoidale
-  = triangolare
-  = quadra

Fig. 7 - Interfaccia dell'oscilloscopio NI ELVISmx durante l'analisi di un brano audio.

Analizziamo di seguito le possibili impostazioni:

- La manopola **Frequency** controlla la frequenza della forma d'onda di uscita, ma è anche possibile immettere il valore di frequenza desiderato nel controllo digitale sotto la manopola (il valore predefinito è 100 Hz).
- **Amplitude** regola l'ampiezza picco-picco della forma d'onda generata (il valore predefinito è pari a 1 Vpp); è possibile immettere un valore di ampiezza nel controllo digitale sotto la manopola.
- **DC Offset** regola l'offset in continua della forma d'onda (il valore predefinito è 0 V); è possibile immettere un valore di offset nel controllo digitale sotto la manopola.
- **Duty Cycle** è un controllo attivo solo per le onde quadre, che permette di realizzare onde rettangolari e impulsive, indicando su un periodo di oscillazione la percentuale di tempo in cui l'onda generata rimane al livello alto (il valore predefinito è 50%: onda quadra).
- **Modulation Type** è un controllo non abilitato per il myDAQ.
- **Sweep Settings** contiene le impostazioni per emettere una sequenza di una certa forma d'onda a varie frequenze in maniera automatica:
 - **Start Frequency** specifica la frequenza di partenza per la scansione di frequenza della forma d'onda scelta (il valore predefinito è 100 Hz)
 - **Stop Frequency** specifica la frequenza alla quale si ferma la scansione di frequenza della forma d'onda scelta (il valore predefinito è di 1000 Hz)
 - **Step** specifica i salti di frequenza tra una forma d'onda generata e quella successiva (il valore predefinito è 100 Hz)
 - **Step Interval** specifica l'intervallo di tempo tra la generazione di una forma d'onda e quella successiva (il valore predefinito è 1000 ms).
 -  Per avviare una generazione di segnali di tipo Sweep è necessario premere il pulsante Sweep.

BODE ANALYZER



Il Bode Analyzer produce i diagrammi di Bode del guadagno e della fase per l'analisi in frequenza. La

Fig. 8 - Generatore di funzioni FGEN.

relativa interfaccia software è riportata in figura 12. È possibile impostare la banda di frequenza operativa dello strumento, all'interno della banda di frequenza supportata: 1÷20 kHz e per il grafico del guadagno scegliere tra le scale di visualizzazione lineare (Linear) o logaritmica (dB), usando il comando **Mapping**; la fase invece è sempre riportata in gradi (deg). Per effettuare una misura con tale strumento è necessario collegare:

- all'ingresso del sistema da analizzare, un segnale sorgente di stimolo, fornito dal canale **AO 0** del myDAQ e un canale di misura dello stimolo **AI 0**;
- all'uscita del sistema da analizzare, un canale di misura della risposta allo stimolo **AI 1**.

Le opzioni **Stimulus Channel** e **Response Channel** non possono essere modificate per il myDAQ.

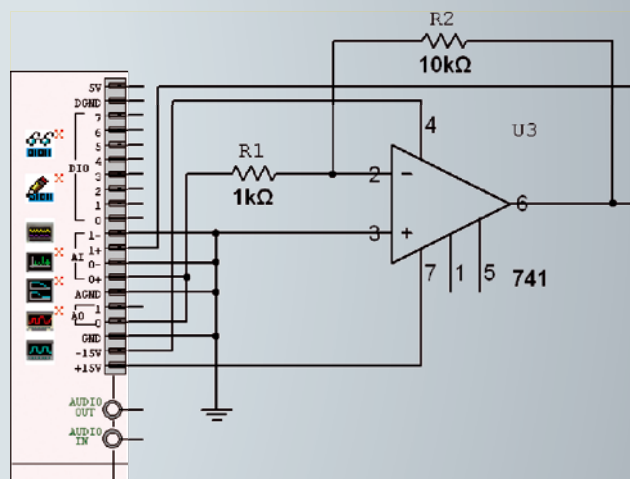
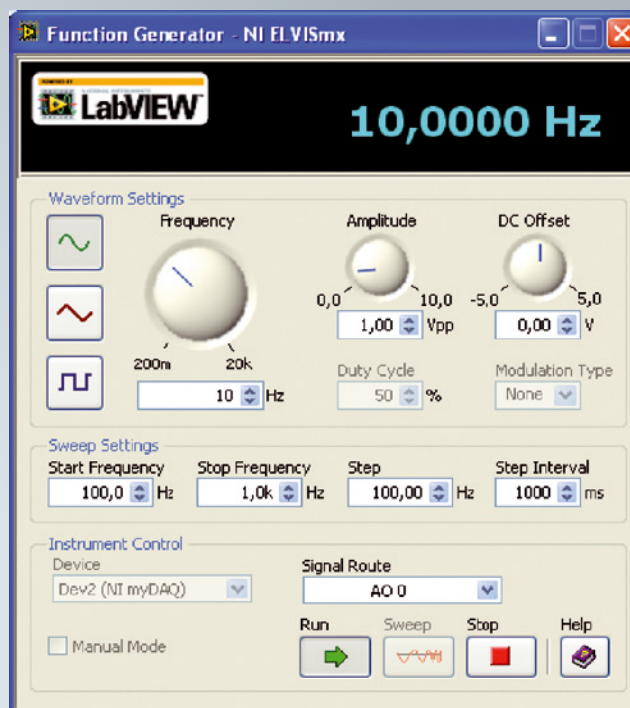
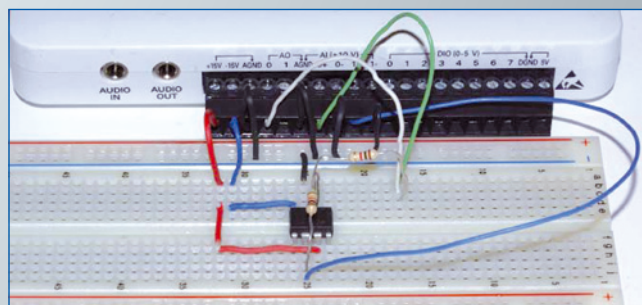
I grafici del guadagno e della fase vengono rilevati tra due frequenze definite dall'utente **Start Frequency** e **Stop Frequency**, tra cui il dispositivo fa variare il segnale di prova. La banda di frequenze supportata dal Bode Analyzer è compresa tra 1÷20 kHz. Il numero di **Steps** indica il dettaglio dell'analisi, espresso in numero di acquisizioni per decade (il valore predefinito è 5).

Peek Amplitude imposta il picco di ampiezza del segnale di uscita dal generatore di funzioni virtuale.

Una scelta opportuna dell'ampiezza del segnale permette di ottimizzare la risposta del sistema: circuiti passivi dovrebbero essere pilotati con segnali di elevata ampiezza, mentre circuiti con alto guadagno, come i circuiti attivi, andrebbero pilotati con segnali di ampiezza inferiore, per evitare di saturare l'uscita. Il valore predefinito è 2 V.

È possibile invertire i valori misurati del segnale di ingresso durante l'analisi di Bode, ponendo il comando **Op-Amp Signal Polarity** sull'opzione **Inverted**, ottenendo così l'azzeramento dello sfasamento per gli amplificatori invertenti. L'inversione non influenza i dati acquisiti (il valore predefinito è **Normal**).

Autoscale determina se i grafici del guadagno e della fase vengono riportati in scala automaticamente (il valore predefinito è attivato in automatico).

**Fig. 9** - Amplificatore invertente proposto per l'esercitazione 2 con generatore di funzioni e oscilloscopio inseriti.**Fig. 10** - Amplificatore invertente proposto per l'esercitazione 2, montato su breadboard, con inserimento di generatore di funzioni e oscilloscopio del myDAQ.

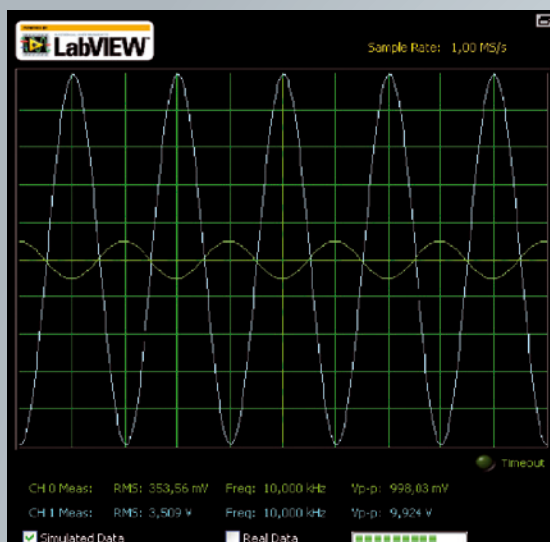
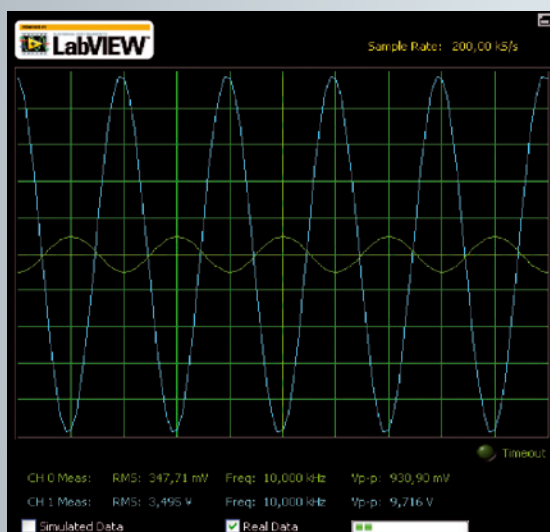
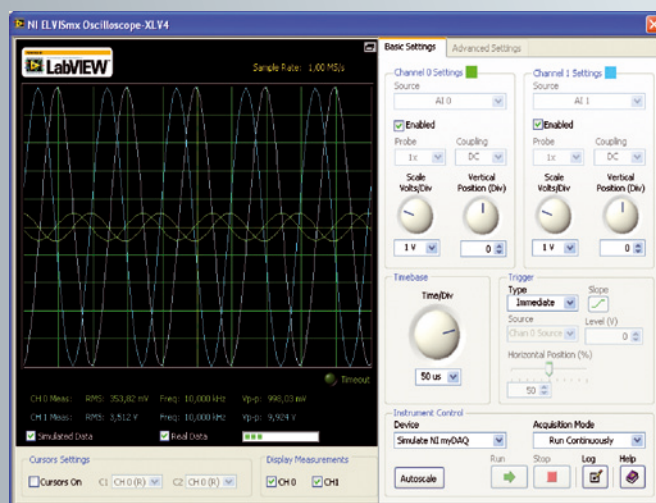


Fig. 11 - (a) Analisi in simulazione (Volts/Div 1V e Time/Div 50 μ s).



(b) Analisi reale (Volts/Div 1V e Time/Div 50 μ s).



(c) Sovrapposizione dei dati ottenuti dall'analisi reale con quelli della simulazione (Volts/Div 1V e Time/Div 50 μ s).

ESERCITAZIONE 2

Si vuole montare su breadboard e analizzare con myDAQ un amplificatore invertente con Op. Amp. LM741 e guadagno in tensione $G = 10$, schematicizzato in Multisim e riportato nella Fig. 9.

Obiettivi:

- fare pratica con lo FGEN e lo Scope del myDAQ esaminando un circuito elettronico in ambiente Multisim 11 in simulazione e successivamente analizzando il circuito reale montato su breadboard;
- confrontare il comportamento del circuito in simulazione con le misure reali;
- esaminare la risposta in frequenza del circuito con lo strumento Bode.

L'amplificatore operazionale LM741 viene alimentato ai pin 4 e 7 con una tensione duale ± 15 V fornita dal myDAQ. L'ingresso del sistema è collegato al canale AO 0, ossia al generatore di funzioni FGEN e al canale AI 0 dello Scope in modo da visualizzare sul display dell'oscilloscopio la forma d'onda di prova. L'uscita del sistema è collegata al canale AI 1 dell'oscilloscopio per analizzare la risposta del sistema al segnale di prova. La massa analogica AGND viene collegata insieme alla massa dell'alimentazione GND; anche i canali AI 0 e AI 1 dell'oscilloscopio vengono riferiti a massa (i rispettivi pin 0- e 1-). Per ottenere un'amplificazione in tensione pari a 10 si adoperano due resistori:

- R_1 da 1 k Ω collegato all'ingresso invertente dell'operazionale, pin 2;
- R_2 da 10 k Ω collegato in retroazione tra l'uscita dell'operazionale, pin 6, e l'ingresso invertente, pin 2.

È noto che il guadagno ideale in tensione per l'amplificatore invertente è pari al rapporto tra le resistenze:

$$G = -\frac{R_2}{R_1}$$

dove il segno meno indica uno sfasamento di -180° della risposta del sistema rispetto al segnale in ingresso. Dopo aver realizzato il circuito su breadboard (Fig. 10) e averlo collegato all'alimentazione e alla strumentazione del myDAQ è comodo avviare **FGEN** e **Scope** direttamente dall'ambiente Multisim 11 così

da poter confrontare il comportamento del circuito in simulazione con le misure reali. Per attivare gli strumenti si usano i pulsanti:



avvia il generatore di funzioni FGGEN;
avvia l'oscilloscopio Scope.

Si avvia dapprima l'analisi in simulazione, in risposta ad un segnale sinusoidale $V_{pp} = 1V$ e $f = 10\text{ kHz}$ emesso dal FGGEN ottenendo la visualizzazione di **Fig. 11a** con in verde l'onda in ingresso e in azzurro la risposta del sistema. Per fare ciò è necessario selezionare sia su FGGEN che su Scope come **Device** l'opzione **Simulate NI myDAQ** e avviare la simulazione da Multisim.

Il display contiene in basso alcune informazioni rilevate sui segnali nell'istante in cui i rispettivi grafici tagliano l'asse y delle tensioni:

- il valore efficace della tensione **RMS**;
- il valore picco picco **Vpp**, ossia la differenza tra il voltaggio più alto e quello più basso misurato.

Nel caso dei segnali sinusoidali alternati si ha il rapporto:

$$RMS = \frac{V_{pp}}{2\sqrt{2}}$$

Il grafico inoltre fornisce l'informazione sulla frequenza dei segnali visualizzati, accanto alla voce **Freq** (in alternativa sarebbe desumibile come inverso del periodo di oscillazione, a sua volta rilevabile sul grafico).

Si può notare come siano presenti nel segnale d'uscita un'amplificazione pari a circa 10 e uno sfasamento di -180° rispetto al segnale d'ingresso. Si passa successivamente ad analizzare la risposta nel tempo del circuito reale, dopo avere stoppato la simulazione precedente in Multisim. Si impostano adesso sia su FGGEN che su Scope le opzioni **Device** nella posizione **Dev1 NI myDAQ** e si avviano l'uno dopo l'altro i Run del generatore di funzione e dell'oscilloscopio. Si ottengono le forme d'onda riportate nella **Fig. 11b**.

Dopo avere acquisito le forme d'onda, sia in simulazione che in realtà, è possibile confrontare il comportamento del circuito nei due casi, spuntando ambedue le caselle **Simulated Data** e **Real Data** che si trovano all'interno del display. Come riportato in **Fig. 11c** il dispositivo reale segue abbastanza fedelmente quello

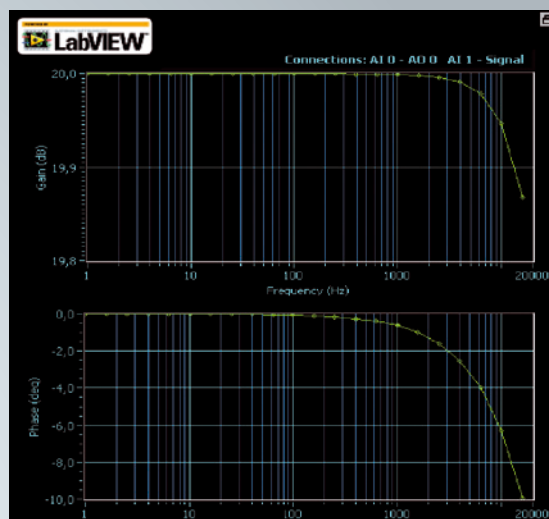
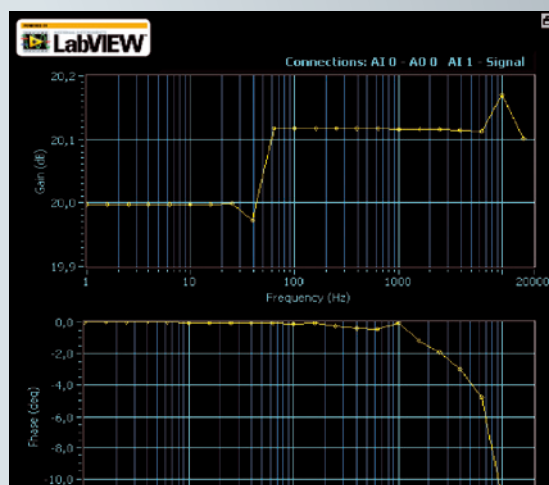


Fig. 12 - (a) Schermata del Bode Analyzer dell'analisi simulata dell'amplificatore invertente (Peak Amplitude: 1 V – banda: 1 ÷ 20 kHz – Steps: 5 per decade).



(b) Analisi reale (Peak Amplitude: 1 V – Op-Amp Signal Polarity: inverted – banda: 1 ÷ 20 kHz – Steps: 5 per decade).



(c) Sovrapposizione dei dati ottenuti dall'analisi reale (grafico di colore giallo) con quelli della simulazione (grafico di colore verde).

Fig. 13 - Digital Reader.



simulato in ambiente Multisim.

Si passa all'analisi di Bode, avviando il Bode Analyzer da Multisim, cliccando sul pulsante , così che siano possibili misure sia in simulazione che dal sistema reale montato su breadboard e collegato al myDAQ.

Si rilevano dapprima i diagrammi di Bode per lo schematico, avviando la simulazione in Multisim (Fig. 12a), dopo avere settato, nel Bode Analyzer, **Device:** Simulate NI myDAQ, **Start Frequency:** 1 kHz, **Stop Frequency:** 20 kHz, **Steps:** 5 per decade. Il guadagno per basse frequenze è pari a 20 dB e comincia a calare in maniera significativa dopo circa 1 kHz: ciò dipende dai limiti della risposta in frequenza del Op-Amp adoperato. Nel diagramma della fase (Fig. 12a) non viene riportato lo sfasamento iniziale di -180° tra uscita e ingresso (dovuto al segno meno nella formula del guadagno) e il diagramma della fase parte da 0° : si tratta di un limite della simulazione con gli strumenti del myDAQ in quanto, come si è già visto, con gli strumenti di simulazione propri di Multisim il problema non si pone.

Successivamente, si effettua l'analisi di Bode sul circuito reale (Fig. 12b) impostando prima un'inversione di polarità del segnale in ingresso all'amplificatore operazionale (Op-Amp), attraverso il comando **Op-Amp Signal polarità: Inverted**, così che venga azzerato lo sfasamento di -180° per l'amplificatore invertente e si possa eseguire un confronto con il diagramma della fase ottenuto in simulazione (Fig. 12a), che invece non tiene conto di tale sfasamento. L'accostamento è proposto nella

Fig. 14 - Digital Writer.

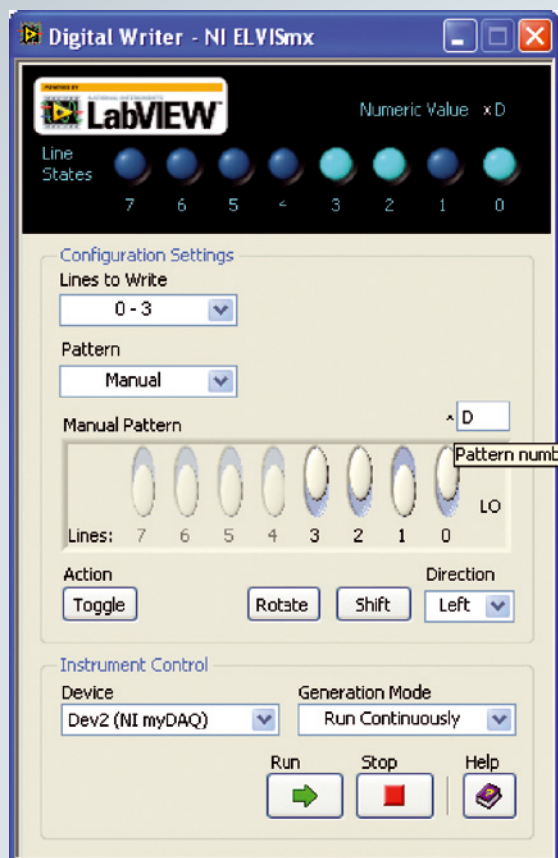


Fig. 12c avendo spuntato ambedue le caselle **Simulated Data** e **Real Data** che si trovano all'interno del display. È bene tenere presente che, essendo l'amplificatore invertente, uno sfasamento iniziale di -180° tra uscita e ingresso esiste a priori e va considerato, come del resto è evidente dai rilievi grafici prodotti con l'oscilloscopio (Fig. 11).

ALTRI DISPOSITIVI SUPPORTATI DA NI MYDAQ



L'analizzatore di segnale NI ELVISmx Dynamic Signal Analyzer (DSA) esegue l'analisi nel dominio della frequenza per una forma d'onda analogica rilevata sul connettore **AI** o sul connettore **AudioInput**.

Lo strumento può effettuare misurazioni in continuo o fare una singola scansione. È anche possibile applicare varie opzioni di filtraggio al segnale.

Parametri di misura:

- Sorgente di segnale: canali AI 0 e AI 1 oppure AudioInput Left e AudioInput Right;
- Gamma di tensioni: per i canali AI ± 10 V e ± 2 V, mentre per i canali AudioInput ± 2 V.



NI ELVISmx Arbitrary Waveform Generator è un generatore di segnali elettrici provenienti da forme d'onda programmate dall'utente. È possibile creare vari tipi di segnali utilizzando il **Waveform Editor** avviabile dall'interno dell'interfaccia del Arbitrary Waveform Generator e generare forme d'onda memorizzate.

Il dispositivo permette di generare simultaneamente due forme d'onda in uscita sui canali **AO** (AO 0 e AO 1) o sui canali **AudioOutput** (AudioOutput Left o AudioOutput Right), ma non su una combinazione di entrambi. È possibile scegliere di eseguire i segnali con continuità o una sola volta.

L'opzione **Trigger Source** è sempre disabilitata per il myDAQ.



NI ELVISmx Digital Reader (Fig. 13) legge e visualizza i dati digitali rilevati sulle otto linee digitali **DIO** del myDAQ poste in ingresso, mediante accensione/spengimento di LED sul proprio display e mediante un valore numerico (convertendo in esadecimale il valore binario ricevuto). L'acquisizione può essere continua o a singola lettura, impostando l'opzione **Acquisition Mode**.

Le linee possono essere raggruppate in lettura in due gruppi di quattro pin (0-3 e 4-7), oppure può essere posto in lettura l'intero port di otto pin (0-7).



NI ELVISmx Digital Writer (Fig. 14) aggiorna le otto linee digitali **DIO** del myDAQ poste in uscita con un dato digitale specificato dall'utente. È possibile creare manualmente un codice binario o selezionare modelli predefiniti, come: rotazione (**Rotate**), inversione della configurazione selezionata (**Toggle**), scorrimento (**Shift**). La scrittura può essere singola o continua, impostando l'opzione **Generation Mode**.

La configurazione in uscita del Digital Writer rimane presente fino a quando o viene generato un nuovo codice, o le linee **DIO** vengono configurate in lettura.

Le linee possono essere raggruppate in scrittura scegliendo un gruppo di quattro pin (0-3 o 4-7), oppure può essere posto in scrittura l'intero port di otto pin (0-7).

Dispositivi avviabili dal NI ELVISmx In-

strument Launcher ma non supportati dal myDAQ

L'offerta di strumenti di analisi circuitale si estende ad ulteriori strumenti che non sono supportati dal myDAQ, ma sono accessibili da ELVIS I, ELVIS II e ELVIS II+. In alternativa possono essere utilizzati in simulazione all'interno di Multisim 11.



NI ELVISmx Variable Power Supply (VPS) è un generatore di tensione variabile che consente ad un dispositivo NI ELVIS di erogare una tensione positiva nell'intervallo 0÷12 V sul connettore Supply+ e una tensione negativa tra 0÷-12 V sul connettore Supply-. Il VPS permette anche di erogare una sequenza di alimentazioni variabili (sweep) in un certo intervallo tra due valori di tensione pre-imposti dall'utente.



NI ELVISmx Impedance Analyzer (Imped) è un analizzatore di impedenza: misura e visualizza l'ampiezza e la fase di un circuito collegato agli ingressi DUT+ e DUT- di NI ELVIS. L'analizzatore misura anche l'impedenza in resistenza e reattanza.



NI ELVISmx Two-Wire Current-Voltage Analyzer (2-Wire) è un analizzatore a due fili corrente-tensione; le misure effettuate vengono riportate su un grafico V-I. Il 2-Wire è in grado di misurare segnali in corrente-tensione tra ± 10 V e ± 40 mA per un circuito collegato agli ingressi DUT+ e DUT- di NI ELVIS.



NI ELVISmx Three-Wire Current-Voltage Analyzer (3-Wire) è un analizzatore a tre fili corrente-tensione per transistor BJT NPN e PNP; le misure effettuate vengono riportate su un grafico V-I. Il 3-Wire è in grado di visualizzare segnali tra 0÷10 V per la tensione di collettore e ± 40 mA per la corrente di collettore.

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.

Un dispositivo basato su Arduino che permette di pubblicare, in un apposito sito Internet, i dati rilevati dalla stazione meteo La Crosse WS2355.

di BORIS LANDONI



PUBBLICHIAMO SUL WEB I DATI DELLE CENTRALI METEO

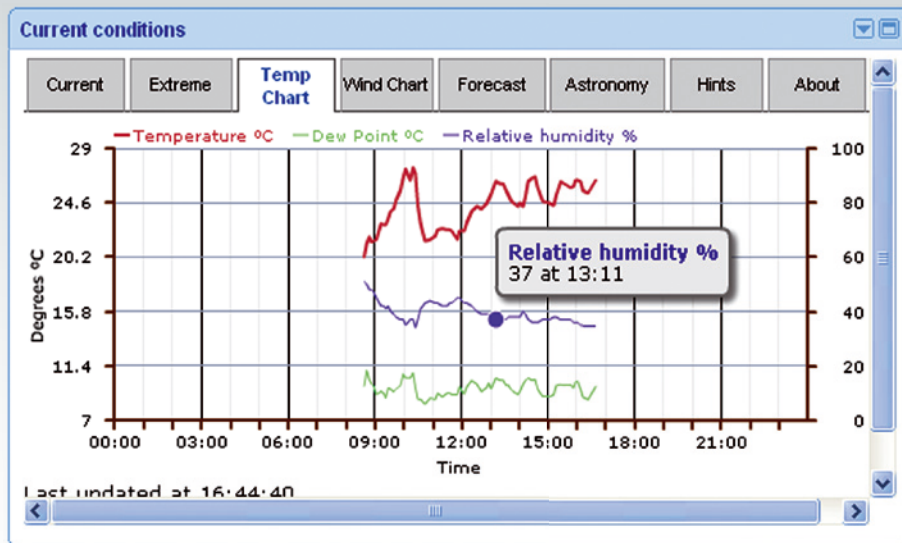
Molta gente ha installata in casa una stazione meteo, forse perché siamo un po' tutti dipendenti dalle previsioni del tempo (quelle in TV sono tra i più gettonati programmi dei vari palinsesti) o magari perché vogliamo semplicemente accertar-

ci del tempo che fa nella nostra casa di vacanza per decidere se concederci o meno un week-end di villeggiatura. Le centraline meteo sono comunque utili anche per valutare a distanza se sta per arrivare un temporale e quindi farci ritirare la tenda da sole

motorizzata sfruttando comandi quali i telecomandi GSM, oppure vengono utilizzate dalle stazioni balneari o in generale dalle località turistiche per informare i potenziali turisti delle condizioni del tempo locali. In tal senso, la possibilità di pubblicare

sul web le condizioni meteorologiche rilevate ha reso un notevole servizio ed ha destato molto interesse, soprattutto per la facilità con cui la procedura può essere automatizzata ed affidata persino ad un semplice microprocessore dotato di interfaccia

Fig. 1 - Widget per Google.



Ethernet, contando su siti web nati apposta per la pubblicazione remota di dati di vario genere, anche non necessariamente meteo.

Il progetto descritto in queste

pagine si muove proprio in tale direzione, in quanto consiste sostanzialmente in un server di rete in grado di leggere i dati da una stazione meteo dotata di interfaccia seriale e collegarsi

ad un apposito sito Internet per effettuarne l'upload; dato che non tutte le stazioni meteo sono uguali, il nostro circuito non può essere universale ma è indicato specificatamente per una serie di dispositivi. Dovendo scegliere, abbiamo optato per una stazione in particolare: la WS2355 della La Crosse. La pubblicazione dei dati avviene sul sito www.wunderground.com.

La scelta della stazione meteo WS2355 è stata dettata essenzialmente da una considerazione: ha una comoda interfaccia seriale che ben si presta al collegamento e al dialogo con un microcontrollore quale Arduino; si tratta infatti di un'interfaccia a livello TTL, i cui segnali sono compatibili (per temporizzazioni e protocollo)

The screenshot shows the "Weather Stations in Italy" page on the Weather Underground website. The page features a table with columns for Station ID, Neighborhood, City, Station Type, and Site. Below the table, there are sections for "PWS Network" and "Help Topics".

Station ID	Neighborhood	City	Station Type	Site
MESSIA2	Messina	Barcellona (ME)	wmr 918	>
9050205	Meteo Farneto	Castellano (RE)	Davis Vantage Pro2	>
ICASTELH1		Castelluccio Scivola	Vantage Pro Davis	>
ICATANIA2		Catania - S.A. LIBATTIATI - Sicily	Weatherlink Pro Vantage	>
IPALERMO12	Bagheria (Pa)	Rete stazioni MeteoPalermo	WMR 918	>
IVENETO87	Sossai	Sossai	LaCrosse WS2300	>
ITERNWV2	COLLESCPOLI 390m	-Terni		>
IVENETO15		Lamon	LACROSSE WS-2350 USB	>
IBARZO11	Valassina	Barzio	Davis Vantage Pro	>
ROMATOL1		Torfa	davis vantage pro2	>
ITERNET2	STRONCONE Palombara	-Terni	ws2300	>
ROMAITA2	Stazione Alessandrino	Ricciobelli - Roma	METEONETWORK DEA	>
IABRUZZO37		Canzano	WMR920N	>
IABRUZZO43	National Park Majella	Caramanico Terme (PE)	davis vantage pro2	>
IABRUZZO40	Tufo Dasso	Caroli	Deagostini-Meteonetwork	>
IABRUZZO11		Casiglione A Cessura	Oregon WMR200	>
IABRUZZO14		Chieti	oregon 925box	>
IABRUZZO12		Francavilla al Mare	La Crosse 2350	>
IABRUZZO24		Guardiglele	Matrix Ws2300	>
IABRUZZO9		Lucoli		>

PWS Network
More than 17,000 weather stations across the globe already send data to Weather Underground! Browse all stations contributing to our network.
Select a state: [dropdown] View
Select a country: [dropdown] View

Help Topics
If you require assistance, or would like to learn more about Personal Weather Stations, please select a related topic below:
• About Personal Weather Stations
• Software Configuration
• Frequently Asked Questions
• Rapid Fire Updates

Fig. 2 - La pagina di ricerca delle stazioni meteo.

con quelli degli UART di schede Arduino tradizionali (Duemilanove, Diecimila ecc.) prive di USB. Cambia qualcosa nei livelli, ma di ciò parleremo diffusamente fra breve.

Quanto al sito *www.wunderground.com*, l'abbiamo scelto perché consente a chiunque sia preventivamente registrato, di accedere al proprio account ed effettuare l'upload dei propri dati meteo, ma anche di monitorare dal web l'andamento di temperatura, umidità, pressione, vento ecc. dove ha installato la stazione meteo, quando non si trova sul posto.

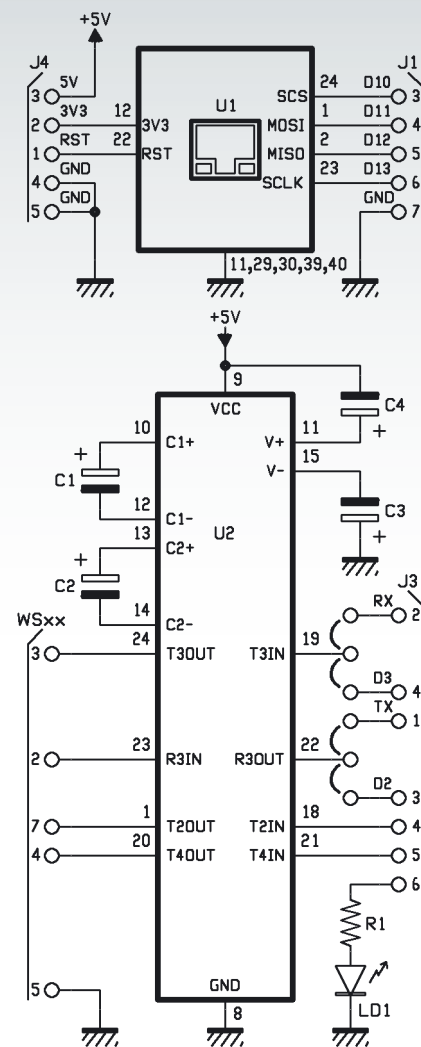
Inoltre in questo sito sono disponibili molti Widget che permettono di integrare i dati presenti in esso, nel proprio sito o blog

personale. Un esempio è il Widget per la home-page di Google, illustrato nella Fig. 1.

Perché abbiamo usato Arduino, è invece la domanda che probabilmente non ci farete: infatti Arduino è oggi il sistema più semplice per sviluppare ogni genere di applicazione, grazie ad un hardware semplice, diretto e versatile e ad una serie di routine e librerie software che consentono anche a un neofita di realizzare funzioni complesse con la massima semplicità. Nel caso specifico, Arduino ci permette di realizzare facilmente l'interfacciamento ethernet richiesto per navigare nel web e puntare al sito *www.wunderground.com*.

Quest'ultimo è un portale meteo nato con l'intento di fornire a chi

[schema ELETTRICO]



vi accede le informazioni meteorologiche ricevute dalla maggior quantità possibile di stazioni meteo di vario tipo, private e non; proprio la diffusione dei dati da parte dell'utente, contribuisce a rendere le previsioni più dettagliate e capillari possibili ed a fornirle in tempo reale.

I dati inviati dagli utenti al sito sono di pubblico accesso, quindi dalla home-page possiamo indicare l'area geografica che ci interessa e ci apparirà una schermata con elencate tutte le stazioni meteo corrispondenti (Fig. 2).

La scelta si effettua con i menu a tendina che si trovano nella zona di destra della pagina, ossia nel

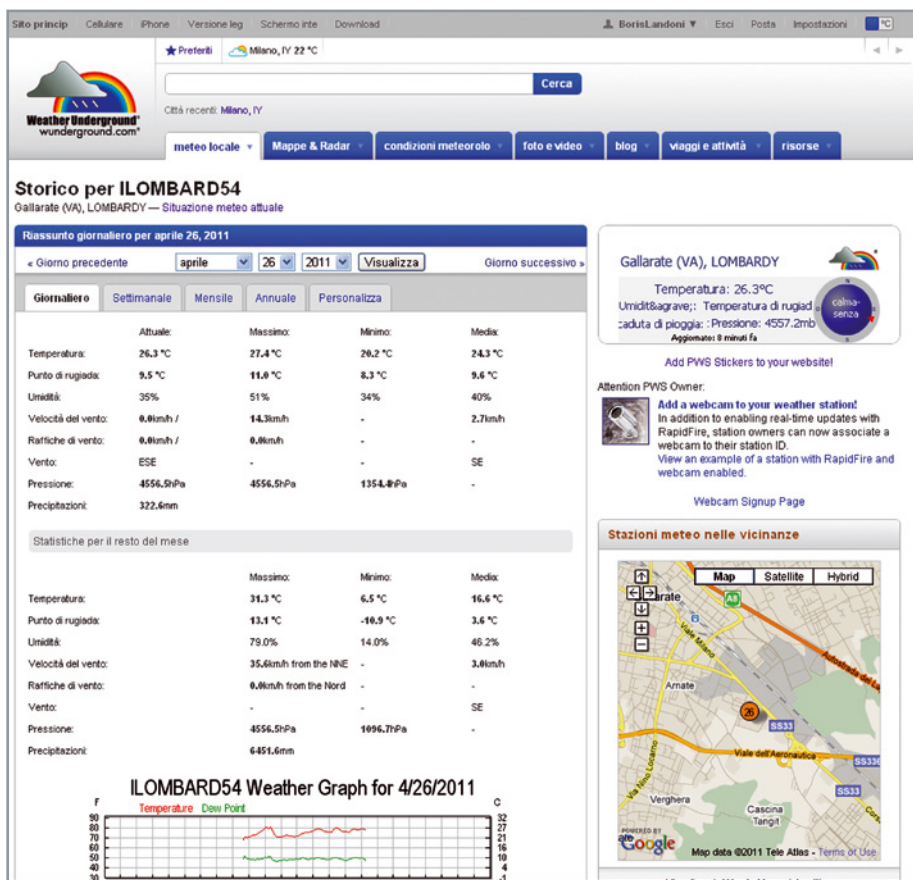
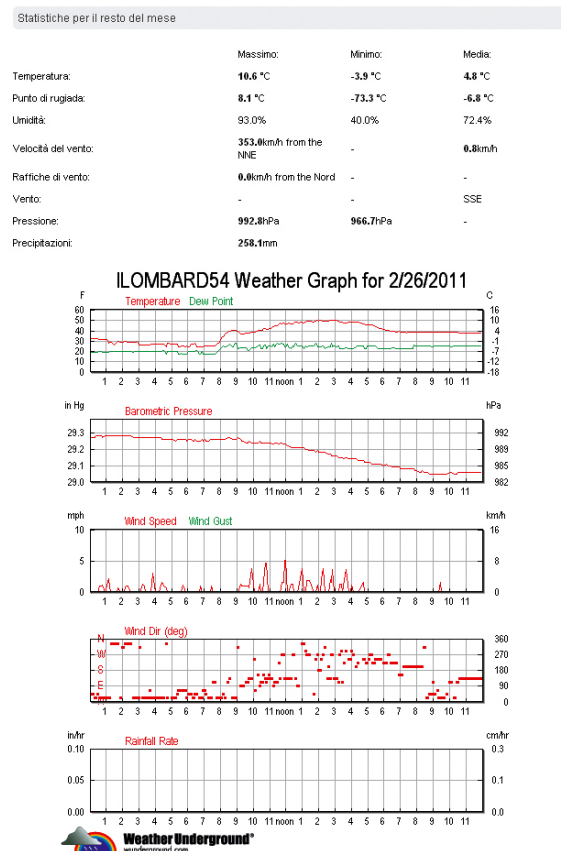


Fig. 3 - Pagina web che mostra i dati rilevati dalla centralina meteo selezionata nella pagina di ricerca.

Fig. 4 - Grafici dello storico dei dati meteo.



riquadro *PWS network*. La cosa interessante è che nella pagina web che riepiloga le stazioni meteo "iscritte" al servizio, potete vederne i dati, ovvero sapere di che tipo di stazione si tratta e in che località si trova.

Facendo clic sul nome di ognuna delle stazioni, appare una pagina come quella mostrata nella Fig. 3, dove vediamo tutte le informazioni che la centralina passa al sito mediante un computer o un dispositivo come quello descritto in queste pagine.

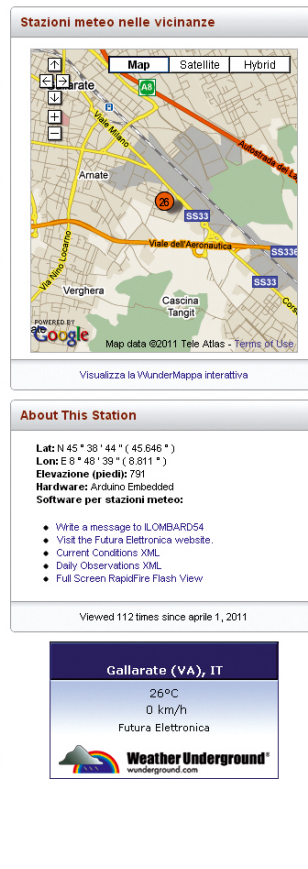
I dati possono essere in tempo reale, ovvero visualizzati in formato storico (cronologico); è anche possibile ottenere dei grafici, che trovate nella parte inferiore della stessa pagina web (Fig. 4).

Dal sito è anche possibile ricercare la situazione meteorologica per località: basta, nella home-page, fare clic nella casella *Inserisci località*, quindi scrivere il nome

del luogo desiderato (potete anche inserire il CAP, lo stato, la sigla dell'aeroporto più vicino); subito accederete alle condizioni meteo ottenute dal sito valutando quelle che gli vengono fornite più di recente dalle stazioni in zona. Per aiutare chi vuole collegarsi e inviare i propri dati, il sito mette a disposizione vari strumenti come (oltre ai Widget) software di supporto e utility per la comunicazione con le varie stazioni meteo esistenti in commercio. Tutto ciò potete trovarlo facendo clic, dalla home-page, sull'etichetta di menu *meteo locale* e scegliendo, nel menu a tendina che si apre, la voce *Stazioni meteo*. Vi troverete software per Windows, Mac e Linux; insomma, un supporto completo.

IL NOSTRO PROGETTO

Esistono diversi software che permettono la pubblicazione dei



dati meteo di una centralina, ma tutti funzionano su PC, ragion per cui potremmo utilizzarli solo a condizione di collegare la stazione meteo ad un computer e lasciare acceso quest'ultimo; a parte il costo dell'hardware, l'idea è abbastanza malsana, perché bisogna tenere da conto sia l'ingombro, sia -soprattutto- il consumo di elettricità dell'insieme. In tempi in cui l'energia dovremmo risparmiarla, proprio non sarebbe una buona cosa.

Il progetto di cui stiamo parlando invece opera in modo stand-alone e permette di pubblicare i dati della stazione meteo in maniera autonoma, "dimenticandosi" del PC: quindi niente consumo e nemmeno spazio occupato sulla scrivania.

Ciò, che fino a qualche anno fa era difficilmente proponibile, oggi è realtà, grazie sia a sistemi a microcontrollore di facile utilizzo come Arduino, sia a moduli ethernet come i Tibbo e i Wiznet, ad esempio; infatti Arduino può gestire molto facilmente il dialogo con la stazione meteo per acquisirne i dati e l'interfaccia ethernet provvede poi a trasferirli, effettuando la necessaria connessione ad Internet mediante ADSL, passando da un router (o da un hub, se ci si affaccia su una WAN o su Fastweb) puntando all'indirizzo IP del sito di pubblicazione e trasferendo le informazioni mediante il protocollo TCP/IP.

COME FUNZIONA?

Il sistema che vi descriviamo è composto da due schede: una è l'ormai famosa Arduino UNO e l'altra è uno shield che integra il modulo ethernet Wiznet, oltre ad un convertitore di livello logico TTL/RS232 di tipo MAX238. Il convertitore permette di adattare i livelli TTL di Arduino a quelli

della seriale della centralina meteo. Va tenuto presente che la porta di comunicazione della stazione meteo non è una semplice RS232, perché infatti non ha il riferimento di massa (anche se nello schema è stato collegato il pin 5, in realtà il cavo in dotazione con la WS2355 non utilizza tale riferimento) e quindi non è propriamente una linea sbilanciata, come la RS232. In quest'ultima, la comunicazione dovrebbe avvenire con riferimento ad una linea di massa (Ground) comune. In realtà nel corso degli anni ci sono state diverse variazioni al tema. Nel caso della stazione meteo La Crosse, affinché la comunicazione avvenga correttamente, la linea DTR deve presentare una tensione negativa mentre la RTS deve essere positiva. La mancanza della massa comune fa ipotizzare (ricordiamo che non esiste alcun documento ufficiale) che queste due linee siano prese come riferimento dei livelli logici utilizzati nella comunicazione. Si tratta di un sistema che, oltre a permettere l'utilizzo di collegamenti seriali relativamente lunghi, dà anche la possibilità di mettere d'accordo dispositivi con differenti livelli di tensione sulla seriale. Questa caratteristica è tipica delle linee di comunicazione seriale bilanciate, come ad esempio la RS422.

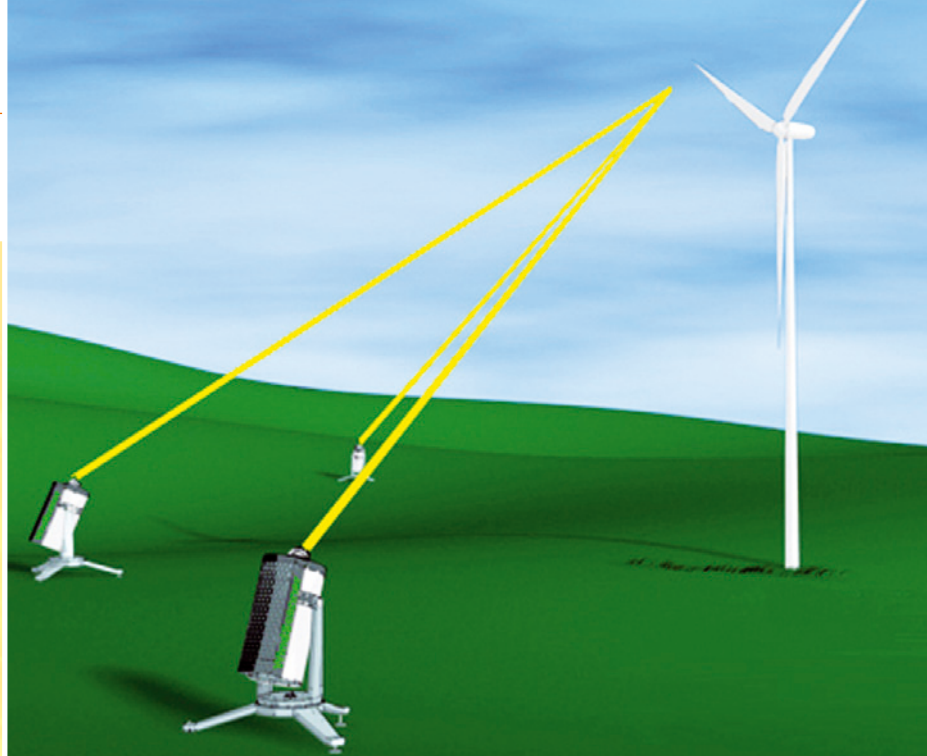
Per dialogare con la centralina La Crosse, la velocità di comunicazione della seriale del dispositivo usato (nel nostro caso, il modulo Arduino) deve essere impostata a 2.400 bps, con blocchi da 8 bit, senza parità e con un bit di stop (2400-8-N-1).

La mappa di memoria della stazione meteo, come molte altre informazioni non ufficiali sulla WS2355 (la LA Crosse è avida di informazioni), si può trovare sul sito <http://www.lavrsen.dk/foswiki/>

bin/view/Open2300/WebHome.
Giunti a questo punto, prima di descrivere lo sketch, ossia il programma che permette ad Arduino di interrogare la stazione meteo e pubblicare i dati sul web, vediamo com'è fatto lo shield ethernet, che abbiamo spiegato essere basato sul modulo Wiznet.

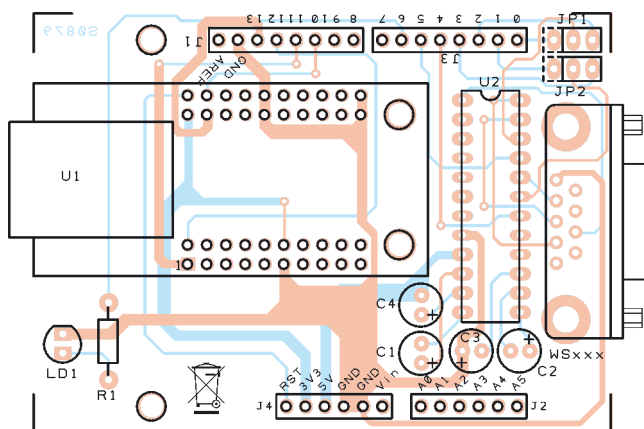
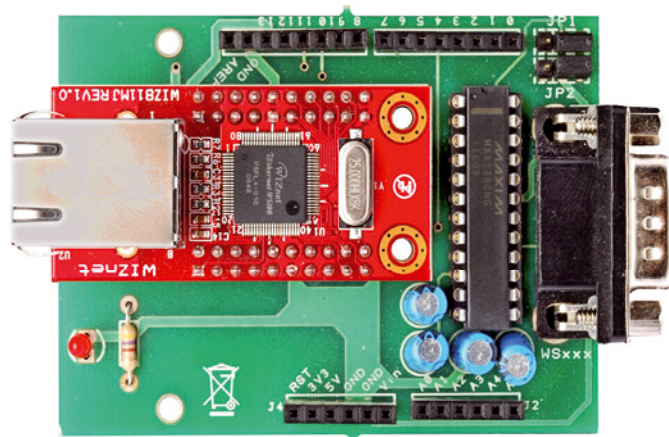
L'HARDWARE

Si tratta sostanzialmente di un'interfaccia o, se preferite, di un circuito stampato, che consente all'Arduino di dialogare con il modulo ethernet (siglato U1 nello schema elettrico dello shield, illustrato in queste pagine) collegato mediante i connettori J1, J3 e J4



Un laser che sente la forza del vento

Si chiama LIDAR ed è un anemometro basato su laser, che sfrutta lo stesso principio dei radar laser impiegati nella misura della velocità dei veicoli su strada. Il dispositivo permette di valutare con estrema accuratezza la forza e la direzione dei venti ed attualmente viene impiegato per orientare le pale delle turbine eoliche in modo che colgano il vento nella maniera ottimale. L'anemometro LIDAR è stato realizzato da alcuni ricercatori del Laboratorio nazionale danese per l'energia sostenibile (Risø Dtu) e permette un'analisi rapida delle condizioni meteorologiche; attraverso un sistema che consente di modificare rapidamente l'inclinazione e l'orientamento dei rotori, consente agli aerogeneratori di posizionarsi nel modo più ottimale rispetto alle condizioni del vento. Con questo sistema è possibile incrementare del 5 % l'efficienza energetica di ogni aerogeneratore, oltre ad allungarne sensibilmente la durata.



Elenco Componenti:

R1: 470 ohm

C1: 1 μ F 100 VL elettrolitico

C2: 1 μ F 100 VL elettrolitico

C3: 1 μ F 100 VL elettrolitico

C4: 1 μ F 100 VL elettrolitico

U1: WIZNET

U2: MAX238CNG

LD1: LED 3 mm rosso

Varie:

- Zoccolo 12+12

- Strip maschio 3 poli (2 pz.)

- Jumper (2 pz.)

- Strip femmina/maschio 6 poli (2 pz.)

- Strip femmina/maschio 8 poli (2 pz.)

- Strip femmina 10 poli (4 pz.)

- Connettore DB9 maschio

- Circuito stampato

delle Arduino Duemilanove e simili. La scheda si monta direttamente sull'Arduino e si connette ad esso tramite i pin-strip di cui è provvista, usando i connettori femmina previsti nell'Arduino stesso per le connessioni di I/O (connettore J1 di Arduino) e di alimentazione (connettore J4 di Arduino).

La scheda shield è sostanzialmente una basetta che ha sul lato superiore due connettori da 20 contatti (10 per fila) a passo 2x2,54 mm fatti per alloggiare il modulo ethernet e su quello inferiore una serie di pin-strip laterali per inserirsi nei connettori di Arduino.

Il modulo ethernet è il Wiznet WIZ811MJ, conforme IEEE 802.3/802.3, basato sul chip W5100 e dotato di connettore ethernet completo di filtri magnetici e di due LED per le segnalazioni di stato. Supporta

i protocolli hardware ethernet DLC e MAC e quelli internet TCP, IP ver.4, UDP, ICMP, ARP, PPPoE, IGMP. L'interfaccia di rete funziona in modalità sia full-duplex che half-duplex. La velocità di comunicazione permessa è 10/100 MHz e vengono supportati la negoziazione automatica ed il rilevamento automatico del cavo incrociato, oltre a quattro connessioni socket simultanee. L'interfaccia con cui il modulo comunica con il microcontrollore è di tipo SPI, con modalità di accesso al bus diretta/indiretta. Il tutto funziona con i 3,3 Vcc prelevati dalla linea 3V3 del connettore J4 che collega la nostra scheda all'Arduino; la massa è in comune. Dallo stesso J4 arriva l'eventuale segnale di reset. L'interfaccia di rete coinvolge anche il connettore J1, attraverso il quale la nostra scheda si collega all'Arduino per connettere

le linee dedicate al dialogo con il modulo ethernet e che costituiscono un completo bus SPI a quattro fili, dove identifichiamo la linea MISO (ingresso dati del micro) la MOSI (uscita dati del micro) la SCLK (clock della comunicazione) e l'SCS (Chip Select o Slave Select).

Quest'ultimo viene utilizzato dal microcontrollore, che funge da unità master della comunicazione, per scegliere con quale dispositivo slave vuole comunicare. In questo caso c'è una sola unità slave, ossia il modulo ethernet U1. Oltre al modulo Wiznet, lo shield che abbiamo realizzato monta un connettore DB-9 maschio per collegarsi alla stazione meteo ed un integrato adattatore di livelli (MAX238CNG) che ci serve ad interfacciare la stazione stessa con il modulo Arduino, ossia a traslare i livelli RS232 compatibili della centralina La Crosse

La stazione meteo La Crosse WS2355

con quelli TTL che Arduino può accettare. Il connettore DB-9 è siglato WSxx nello schema elettrico, mentre J3 è il pin-strip tramite cui la nostra scheda shield si connette ad Arduino.

Notate i jumper posti sulle linee T3IN ed R3OUT (rispettivamente uscita dei dati in formato TTL dal converter RS232/TTL ed ingresso del convertitore TTL/RS232): essi permettono di scegliere a quali linee del modulo Arduino far effettuare la comunicazione, nel senso che potendo usare, come canale seriale, sia TXD ed RXD, sia due qualsiasi linee di I/O digitali, vi diamo la possibilità di scegliere. A riguardo, rammentate che se chiudete i ponticelli su TX ed RX, usate la seriale hardware di Arduino, mentre se collegate D2 e D3 (linee di I/O digitali di Arduino) la comunicazione avverrà emulando una seriale con l'apposita libreria scaricabile dal nostro sito Internet. Ricordate altresì che i jumper di trasmissione e ricezione vanno chiusi solo in una direzione, nel senso che potete chiudere T3IN su RX o su D3 ma non su entrambi e quindi R3OUT solo su TX o D2. Concludiamo la descrizione del circuito con il diodo luminoso LD1, utilizzato per comunicare lo stato della comunicazione seriale e pilotato da Arduino mediante il piedino 6 del connettore J3.

REALIZZAZIONE DELLO SHIELD

Passiamo adesso a due indicazioni riguardanti la costruzione dello shield, che nel nostro caso è la banale scheda a circuito stampato appena descritta, che ospita il connettore per il collegamento alla centralina meteo, il converter TTL/RS232 ed il modulo Wiznet. Per costruire lo shield bisogna innanzitutto preparare il circuito stampato corrispondente, per

fotoincisione, dopo aver ottenuto la pellicola da una buona stampa delle tracce lato rame (la basetta è a doppia faccia) scaricabili dal nostro sito www.elettronica.in.it. Sul c.s. dovete innanzitutto montare (sul lato superiore) i due connettori a 2x10 poli a passo 2x2,54 mm, che potete ricavare semplicemente da file di contatti

femmina a passo 2,54 mm. Fatto ciò, nella faccia inferiore dello stampato inserite e saldate le file di punte a passo 2,54 mm che occorrono a realizzare le connessioni con Arduino: due da sei contatti per J2 e J4 ed altrettante, ma da otto contatti, per J1 e J3. I pin strip da utilizzare devono essere lunghi almeno 20 mm e fanno

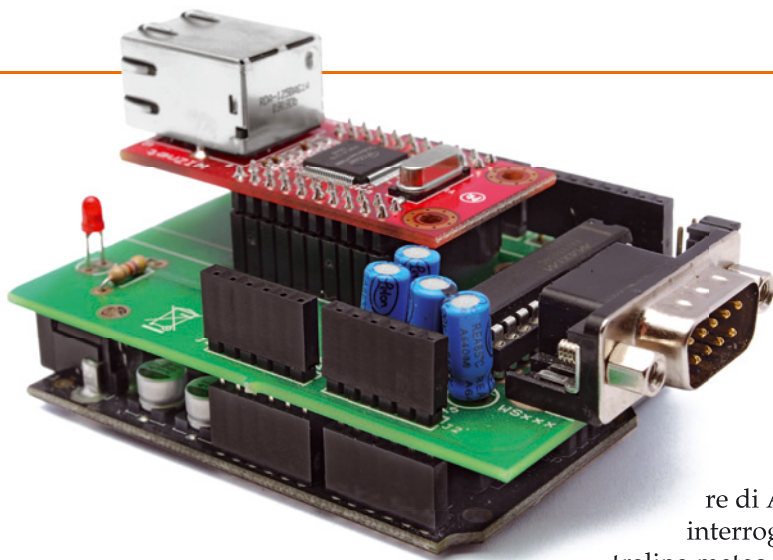
Si tratta di una stazione meteo professionale completa di pluviometro, anemometro, sensori di direzione del vento, temperatura e umidità, barometro, orologio radiocontrollato. I sensori esterni comunicano via radio (a 433 MHz) con la stazione, la quale si può interfacciare con un PC tramite il cavo seriale/USB ed il software in dotazione.



Le caratteristiche salienti di questa stazione sono:

- ora e data sincronizzate con il segnale della stazione oraria DCF77;
- Visualizzazione dei dati meteo con funzioni di allarme programmabili;
- registrazione di tutti i valori minimi e massimi con ora e data di registrazione;
- misura di temperatura interna ed esterna in °C o °F;
- visualizzazione dell'umidità relativa;
- visualizzazione dell'umidità interna e esterna;
- visualizzazione del giorno, data, ora (12/24h), mese e fuso orario;
- velocità del vento in mph, km/s, m/s, nodi o Beaufort;
- visualizzazione della direzione e del wind-chill;
- visualizzazione del punto di rugiada;
- previsioni meteo con icone (soleggiato, nuvoloso, piovoso);
- allarme temporali;
- porta COM per connessione al PC;

Oltre a ciò, la stazione permette di impostare i parametri diversamente per più utenti e visualizzarli distinti per utenza (sono possibili fino a 175 serie di dati); le registrazioni possono avvenire ad intervalli di misurazione impostabili a scelta dall'utente.



stagnati dal lato superiore dello stampato.

Ora, sempre sulla faccia superiore del c.s. sistemate il MAX238, che suggeriamo di montare su zoccolo, poi il connettore a vaschetta DB-9 (stagnatene bene le alette di fissaggio, altrimenti si potrà strappare) i jumper e i pochi altri componenti che servono. Fatto ciò, inserite il modulo Wiznet orientandolo come indicato nel piano di montaggio illustrato nelle pagine precedenti e sistemate lo shield su Arduino; non preoccupatevi per l'orientamento perché il montaggio è possibile solo nel verso giusto.

Ora potete dare tensione e collegare al sistema il cavo seriale che connette la stazione meteo. Prima di usare il tutto, però, dovete programmare Arduino caricandovi l'apposito sketch.

LO SKETCH

Spiegato questo, vediamo adesso come avviene la comunicazione con la centralina, iniziando col dire che la connessione seriale si avvale dei pin 2 e 3 del modulo Arduino.

Questi pin non fanno capo ad una seriale hardware, bensì sono due semplici I/O; per questa ragione abbiamo utilizzato la libreria NewSoftSerial, che permette di emulare un UART utilizzando dei generici contatti di I/O come linee seriali.

Ciclicamente, il microcontrollo-

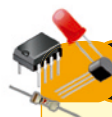
re di Arduino interroga la centralina meteo e salva in memoria i dati relativi alla data, all'ora, alla temperatura esterna, all'umidità esterna, alla velocità e direzione del vento, alla pioggia caduta nell'arco di un'ora e nelle 24 ore, alla pressione atmosferica ed al punto di rugiada (temperatura alla quale l'umidità condensa). La costante UPDATE_INTERVAL definisce ogni quanti secondi Arduino pubblicherà sul sito www.wunderground.com i dati rilevati.

Per poter pubblicare i dati della centralina in questo sito, come già accennato è necessario avere un account; una volta registrati, ci viene assegnato un ID che sarà necessario introdurre durante la connessione al sito per poter effettuare la pubblicazione. Nello sketch per Arduino è necessario inserire il proprio ID e la propria password, affinché la pubblicazione possa avvenire con successo; diversamente il nostro sistema punterà al sito ma non otterrà l'accesso.

Per la pubblicazione dei dati è sufficiente richiamare la pagina <http://weatherstation.wunderground.com/weatherstation/updateweatherstation.php>, passando i dati appesi all'URL. Per esempio, bisogna comporre una stringa del tipo: <http://weatherstation.wunderground.com/weatherstation/updateweatherstation.php?ID=KCASANFR5&PASSWORD=XXXXXX&dateutc=2000-01-01+10&windd>

ir=230&windspeedmph=12&windgustmph=12&tempf=70&rainin=0&baromin=29.1&dewptf=68.2&humidity=90&weather=&clouds=&softwaretype=vws%20versionxx&action=updateraw.

Come vedete, nella stringa vengono inseriti (per essere passati al sito) anche l'ID utente e la password. I dati della temperatura vanno espressi in gradi Fahrenheit e la velocità del vento deve essere in mph (miglia per ora). La conversione dei dati letti dalla centralina viene effettuata direttamente dallo sketch, così da risparmiarvi l'onere di procedere al calcolo manuale. L'uso del sito www.wunderground.com per la pubblicazione dei dati meteo è libero e soggetto solo alle regole del provider; esistono anche funzioni avanzate, tra le quali l'aggiunta di foto e delle immagini riprese da una webcam sul luogo dove si trova la stazione meteo e inviate tramite Internet. ■



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono di facile reperibilità. Il master del circuito stampato e lo sketch utilizzato possono essere scaricati gratuitamente dal sito della rivista (www.elettronica-in.it). Presso Futura Elettronica sono reperibili la scheda di networking Wiznet (Euro 35,00), la stazione meteo La Crosse WS2355 (euro 172,00), l'integrato MAX238NG (Euro 9,80) e la board Arduino Uno (Euro 32,00). Tutti i prezzi si intendono IVA compresa.

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

RILEVAMENTO PER DEFLESSIONE QUALI SENSORI?

di DARREN WENN

Le interfacce per *touch-screen* e *touch-panel* hanno rapidamente acquisito enorme importanza in moltissime applicazioni, come ad esempio nei dispositivi portatili quali l'iPhone e l'iPad di Apple.

Le interfacce tattili hanno avuto una lunga storia prima che la fase attuale fosse di accettazione diffusa nell'elettronica di consumo; una grande varietà di tecnologie del rilevamento, e una grande quantità di inventiva, sono state applicate alla soluzione del problema nel corso degli anni passati. L'odierna crescita esplosiva di prodotti *touch-based* ha portato i critici dei *consumer-product* e persino i consumatori stessi a pronunciarsi sulla superiorità della tecnologia capacitiva su quella resistiva. Il rilevamento tattile capacitivo ha diversi vantaggi: assorbe meno corrente, è più economico e semplice da applicare, meccanicamente affidabile e permette

grande libertà ai progettisti per quanto riguarda la forma e il posizionamento di pulsanti, slider ed altri formati di controllo ormai diventati onnipresenti. Infatti, gli sviluppatori di dispositivi *touch-sensing* possono orientarsi tra una pletora di possibilità d'approccio al rilevamento tattile capacitivo, tutte diverse nei dettagli della tecnica da

Gli schermi *touch-sensing* basano il loro funzionamento su tecniche resistive e capacitive. Facciamo il punto sullo stato della tecnologia e scopriamo i sensori tattili capacitivi con pannello frontale interamente in metallo.



MPLAB X: il nuovo ambiente di sviluppo open-source

È arrivato MPLAB® X, il nuovo ambiente di sviluppo integrato (l'IDE, se preferite) open-source con supporto cross-platform per sistemi operativi Linux, Mac OS e Windows. Il nuovo IDE MPLAB X di nuova generazione introduce un vasto spettro di dotazioni e di caratteristiche ad alte prestazioni quali la possibilità di gestire più progetti, una serie di tool con debugging simultaneo, un editor avanzato, grafico delle chiamate di funzione e completamento del codice. MPLAB X offre un ambiente unico capace di supportare un portafoglio completo di microcontroller a 8-, 16- e 32-bit comprendente tutti gli oltre 800 modelli di microcontroller PIC, di dsPIC e di dispositivi di memoria. La combinazione tra il potente IDE MPLAB X e il portafoglio di microcontroller "migration-friendly" PIC mette a disposizione una piattaforma di

sviluppo semplice, universale e flessibile per i progetti embedded di nuova generazione. Questi progetti richiedono un IDE in grado di assicurare una solida base per uno sviluppo ad alte prestazioni, user-friendly e flessibile oltre che una totale compatibilità con un ampio spettro di tool di progettazione. A questo si aggiunge un ampio portafoglio di microcontroller che permette di semplificare la migrazione, di ridurre la curva di apprendimento e di proteggere gli investimenti esistenti in termini di tool e codici. MPLAB X offre una singola interfaccia grafica unificata per i tool di Microchip e delle terze parti, compresi i debugger/programmatori MPLAB ICD 3, PICKit 3 e MPLAB REAL ICE. Queste le caratteristiche principali del nuovo MPLAB X:

- nuovo IDE MPLAB® X: sessioni di debugging multiple simultanee,

editor avanzato e completamento del codice;

- l'unico IDE universale con supporto per tutto il portafoglio di MCU a 8/16/32-bit;
- Interfaccia grafica unica per tool di Microchip e di terze parti.

MPLAB X si sviluppa sui vantaggi propri della piattaforma open-source NetBeans sponsorizzata da Oracle, la quale vanta una comunità attiva di utenti che contribuisce con un vasto range di migliorie e di plug-in. I clienti Microchip possono anche beneficiare dei plug-in e dei componenti software gratuiti NetBeans già disponibili, oltre che della possibilità di personalizzare l'IDE MPLAB X IDE per indirizzare particolari esigenze di sviluppo.

Tra le ulteriori caratteristiche del nuovo IDE MPLAB X si segnalano:

loro impiegata, e tutte sufficientemente diverse perché i loro rispettivi fornitori possano accampare potenziali diritti brevettabili su di esse.

In ogni caso, però, la natura dell'effetto di campo del rilevamento tattile capacitivo tuttora impone alcune limitazioni, a prescindere dal dettaglio di applicazione.

I sistemi tattili capacitivi standard normalmente non possono funzionare attraverso i metalli; e la tecnica richiede software speciali -in effetti, il *signal-processing* dei dati del sensore- al fine di mantenere l'operatività in ambienti con rumore irradiato o condotto. Questo è un punto tutt'altro che banale quando l'interfaccia tattile va oltre il contesto di un "cool-gadget" per applicazioni medicali o industriali *mission-critical*.

Alcune altre limitazioni delle tecnologie capacitive includono una contenuta capacità di rilevare correttamente un tocco in presenza di acqua o altri contaminanti, o indossando guanti

(specie grossi guanti protettivi) ed il fatto che la posizione di un pulsante è delineata solo graficamente, problema, questo, cui vanno incontro gli utilizzatori non vedenti che si affidano al linguaggio Braille.

RILEVAMENTO DELLA FLESSIONE DEL PANNELLO

Una nuova tecnica che adatta tecniche collaudate di circuiterie di rilevamento con una semplice quanto insolita costruzione fisica, supera tutte queste limitazioni senza compromettere il consumo di corrente o la semplicità di progettazione.

Con l'approccio *Metal Over Cap*, tutta la circuiteria è celata in sicurezza dietro il pannello -solitamente il pannello frontale del prodotto stesso- e tuttavia mantiene sensibilità al tatto e immunità ai rilevamenti di falsi input.

Nei sistemi capacitivi tradizionali, l'utilizzatore modifica il valore capacitivo ponendo il proprio dito direttamente in contatto con il sensore. Il dito

dell'utilizzatore diventa parte del condensatore di rilevamento, formando un secondo strato del condensatore oppure cambiando le proprietà dielettriche attorno ad un preesistente dispositivo a doppia piastra.

Il cambiamento di valore può essere rilevato con una varietà di tecniche circuitali: per esempio un metodo elementare è quello di usare il condensatore di rilevamento per variare la frequenza di un oscillatore; in questo caso la capacità così modificata sposta la frequenza di un valore rilevabile.

Contrariamente ai sistemi che utilizzano il contatto del dito direttamente come fosse uno strato capacitivo, il sistema *Metal Over Cap touch* utilizza un target conduttivo, sospeso sopra i sensori di tatto capacitivi, per formare un secondo strato del condensatore.

Quando un utilizzatore applica una pressione su un target (pannello), la deformazione risultante del target lo fa avvicinare al sensore capacitivo. Un

- utility di importazione per una rapida e semplice migrazione dei progetti dalla precedente piattaforma IDE MPLAB;
- menu contestuali e completamento codice attraverso un editor avanzato;
- finestre di osservazione (watch) configurabili;
- supporto simultaneo per più versioni di compilazione
- tool di collaborazione in team per l'osservazione dei bug e il controllo del codice sorgente.

Con l'IDE MPLAB X, Microchip consolida la propria posizione di fornitore di piattaforme di sviluppo embedded low-cost ideate per supportare il range completo di microcontroller PIC, di digital signal controller dsPIC e di dispositivi di memoria, ricchi di funzionalità e di



periferiche Per evitare effetti negativi sui progetti attivi, Microchip, in ogni caso, continuerà a supportare l'attuale ambiente MPLAB 8.

L'IDE MPLAB X può essere esaminato scaricandolo gratuitamente al link: <http://www.microchip.com/get/D413>.

In alternativa, è possibile fornire un feedback attraverso l'MPLAB X Users' Forum al link <http://www.microchip.com/get/M6RA>.

sistema del genere, nella pratica richiede un cambiamento nella spaziatura degli strati capacitivi e quindi di capacità -dato che quest'ultima varia in maniera inversamente proporzionale alla distanza tra gli strati- di un 6 % o più.

La variazione di capacità viene quindi misurata da un microcontrollore e trasformata in un segnale di tipo on/off corrispondente alla condizione di pulsante premuto o a riposo.

La Fig. 1a mostra la costruzione di base di un tipico sensore *metal over capacitive*, mentre la Fig. 1b raffigura la deformazione causata dalla pressione esercitata dall'utente. Infine, la Fig. 1c mostra una configurazione alternativa che impiega un target metallico collegato al retro di una pellicola di sostegno.

Il target in questa configurazione può essere sia un sottile foglio metallico collegato al retro della lamina di plastica, sia un rivestimento metallico o una copertura metallica applicata sul foglio plastico.

Mentre il target metallico esegue la stessa funzione elettrica di un sistema a fascia metallica, sono le caratteristiche fisiche della plastica che determinano la deviazione meccanica conseguente alla pressione dell'utilizzatore in questa configurazione.

La progettazione di un dispositivo *Metal Over Cap* e la scelta di materiali e dimensioni appro-

priate, implica il bilanciamento di un certo numero di fattori. Il materiale di cui è fatto dovrà essere sottile e abbastanza flessibile da deformarsi sotto il tocco o a seguito di una leggera pressione; tuttavia, la configurazione è sensibile a deformazioni fisiche anche molto piccole.

Questo requisito è prontamente soddisfatto dalla tecnica *Metal Over Cap*.

Il materiale dovrebbe essere flessibile e deformarsi solo all'interno del suo limite elastico, fino all'attesa massima pressione applicata, in modo da tornare alla sua posizione "piatta" non appena la pressione è rimossa. Tra le altre considerazioni da fare in proposito ci sono la dimensione dei pulsanti (o di altri oggetti di controllo) e la loro separazione, il sistema adesivo utilizzato per connettere lo spaziatore e la lamina target al PCB, oltre allo spessore dello strato separatore. La distanza tra i pulsanti è importante al fine di minimizzare movimenti sui sensori adiacenti a quello in uso.

La dimensione, la disposizione e la distanza dei bottoni sono im-

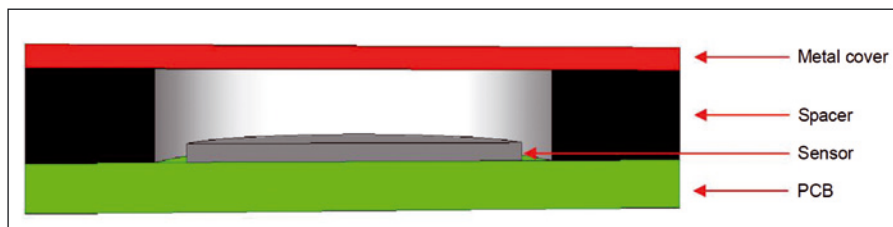


Fig. 1a

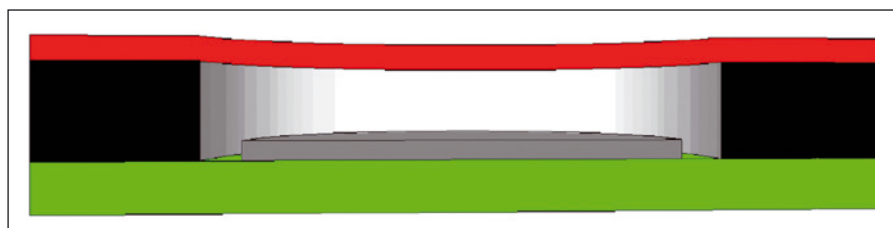


Fig. 1b

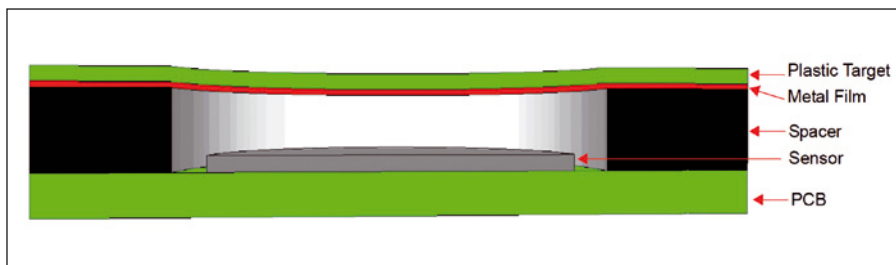


Fig. 1c

portanti al fine di minimizzare la deflessione del pannello sul sensore adiacente a quello che è stato premuto (che altrimenti viene eccitato anche se non vogliamo perché non gli abbiamo premuto sopra) tenendo bene a mente che in un'applicazione dove le mani potrebbero indossare guanti, il tocco potrebbe non essere sempre centrato sul target.

Tra i fattori che incidono sul come i pulsanti del dispositivo touch-sensing interagiscono tra di loro, c'è l'elasticità del materiale di cui è fatto il target; oltre all'adesione causata dalla colla utilizzata per far aderire il target allo spaziatore.

Se il target fosse troppo rigido e l'adesivo elastico, allora una forza applicata al pulsante A solleverebbe il target sopra il sensore B.

Un vincolo pratico nella progettazione di pannelli è che la spaziatura tra i pulsanti dovrebbe essere almeno tra un terzo e metà del diametro dei pulsanti stessi; visto il gran numero di materiali utilizzabili per realizzare sia per il pannello che per la lamina di separazione, i progettisti dovrebbero lavorare a stretto contatto con i produttori di adesivi, allo scopo di scegliere il sistema di fissaggio maggiormente adatto.

Lo stesso progetto della lamina isolante è un ulteriore fattore; il progettista è impegnato a raggiungere la flessibilità nella lamina o nel pannello, cercando, nel contempo, di evitare ogni

movimento nel PCB su cui sono montate le piastre capacitive fisse. Così come per mantenere la distanza richiesta, lo spaziatore deve essere rigido il più possibile; il materiale del PCB (FR4), o altri film plastici non deformabili possono assolvere molto bene a questo compito.

PROGETTO ELETTRICO

Il rilevamento di variazioni della capacità può impiegare un numero di tecniche simili a quelle usate in altre architetture di sensori capacitivi di rilevamento tattile.

Allo scopo Microchip offre due sistemi, ciascuno dei quali lavora molto bene con la tecnica *Metal Over Cap*.

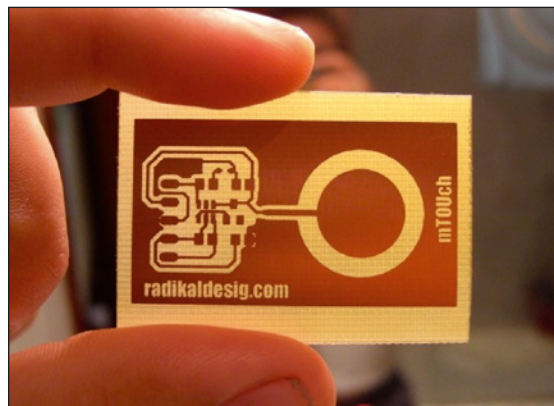
Nella tipologia di approccio *Capacitive Voltage Divider (CVD)*, il sensore capacitivo è connesso in parallelo con un condensatore *sample-and-hold* di un convertitore analogico/digitale; la variazione tra le capacità al tocco e al rilascio appare come una variazione di tensione all'input del ADC ed è convertita in un valore digitale.

La tecnica *Charge Time Measurement Unit (CTMU)* utilizza una sorgente di corrente costante e misura la capacità attraverso la sua curva di carica/scarica. In entrambi i casi, viene reso disponibile un sof-

tware completo da includere nella codifica dei microcontroller. Entrambe le tecniche di rilevamento rendono uno spostamento di 60÷100 conteggi in un risultato a 12-14 bit, per una variazione del 6 % della capacità. Questo è sufficiente per fornire un ben distinto rilevamento di tocco, anche se è richiesta attenzione nella progettazione del sistema per ridurre i rumori sia elettrici che meccanici. La messa a terra, AC e DC, della pellicola target, un'attenta analisi del PCB del rilevatore, un attento posizionamento delle connessioni e una buona pratica nella scelta dei condensatori di bypass, porteranno alla realizzazione di un progetto valido ed affidabile.

CONCLUSIONI

La combinazione di una pellicola target/pannello deformabile e il basso consumo e la semplicità del rilevamento tattile capacitivo creano una potente combinazione: le problematiche associate all'utilizzo in presenza di acqua e rumore vengono eliminate, l'effetto *proximity-trigger* viene sostituito mettendo in atto le esatte specifiche del progettista, e il sistema mantiene il basso assorbimento di corrente di funzionamento del rilevamento tattile capacitivo. ■



PUNTATA 2

Corso di programmazione iPhone



Dopo la prima puntata, introduttiva, è giunto il momento di vedere come avviene l'installazione dell'ambiente di sviluppo X-Code, che ci permetterà di scrivere le nostre applicazioni aventi per target iPhone e iPad. L'ambiente permette inoltre il debugging, il testing e tutto ciò che riguarda la gestione dei dispositivi dedicati allo sviluppo. È importante ricordare che per installare XCode dedicato allo

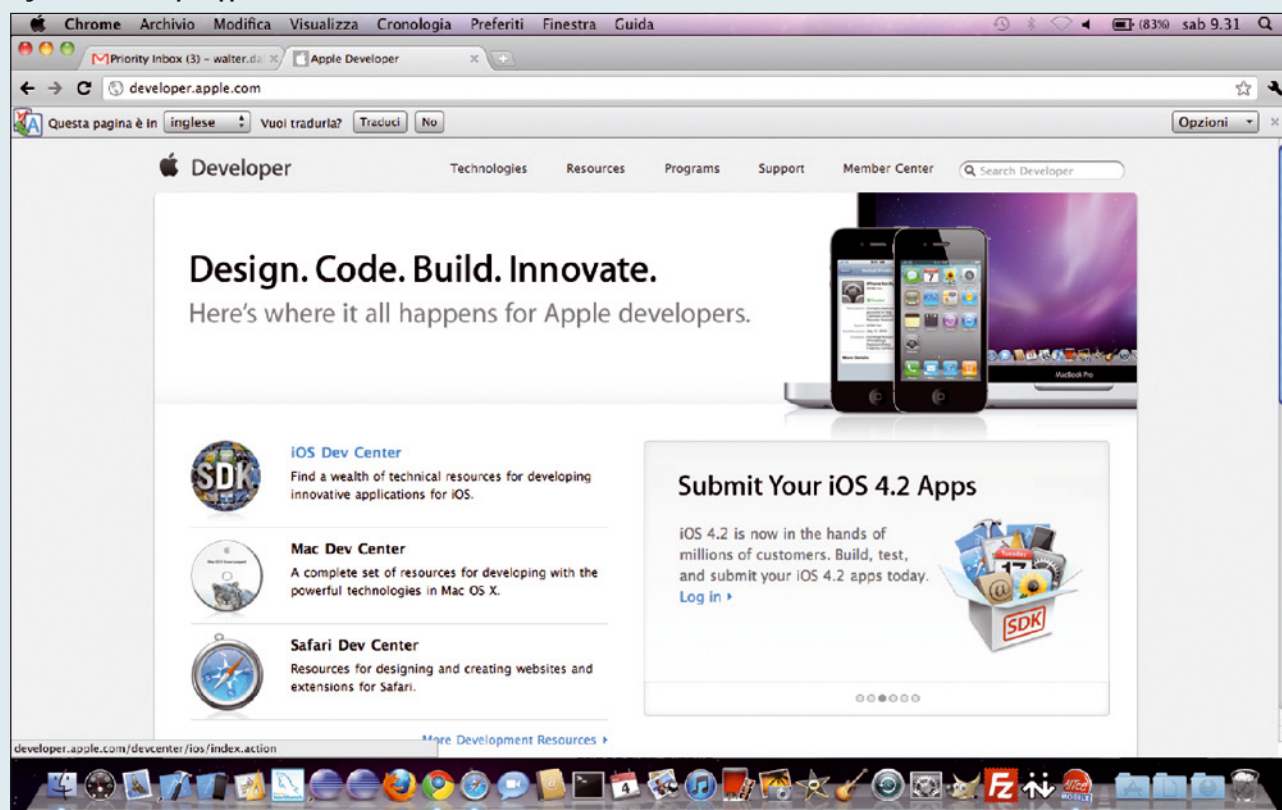
sviluppo è assolutamente necessario un computer Macintosh basato su processore Intel, altrimenti non sarà possibile configurare la SDK.

Installazione di X-Code

Per installare X-Code è necessario scaricare sul Mac l'immagine disco "dmg" direttamente dal sito degli sviluppatori Apple. In Fig. 1 è rappresentata la pagina web dedicata al sistema iOS, da dove è possibile

Installiamo ed impariamo ad utilizzare l'ambiente di sviluppo X-Code, grazie al quale possiamo scrivere, debuggare e testare le applicazioni per iPhone e iPad.

Fig. 1 - Il sito Developer Apple.



scaricare la SDK; l'URL di questo sito è <http://developer.apple.com>, nel caso in cui avessimo già un account di sviluppo per altre piattaforme Apple, potrà andare bene la firma già registrata. I tools di sviluppo sono riservati ai soli titolari di account validi, per cui la registrazione è obbligatoria; una volta completata la registrazione, si può accedere all'area riservata, dalla quale prelevare il pacchetto di Xcode, ovvero il tool necessario al lavoro su iOS, sia per iPhone che per iPad. Per iniziare la registrazione bisogna fare clic sul link in alto a destra, visibile già nella Fig. 1, identificato dall'etichetta *Member Center*. Una volta fatto clic, il sistema naviga verso la sezione di login (come si può vedere in Fig. 2) dedicata allo sviluppo. Nel caso in cui non fossimo già detentori dell'account, dobbiamo cliccare sul pulsante a destra della finestra, dove è scritto "Need to register? Just now"; una volta fatto questo, si apre un'altra pagina da dove è possibile iniziare la registrazione. L'acquisto della licenza di sviluppo è necessario per la sintesi di ogni nostro progetto

su dispositivo; a questo punto della trattazione basta una registrazione ridotta al solo account, quindi senza oneri di spesa, in quanto per iniziare è necessario solamente il simulatore fornito nella SDK.

Come possiamo vedere in Fig. 3, nella sezione dedicata agli sviluppatori possiamo leggere la documentazione, scaricare le versioni di sviluppo di iOS e molto altro ancora; lasciamo a voi la scelta di approfondire la conoscenza del portale apple-developers.

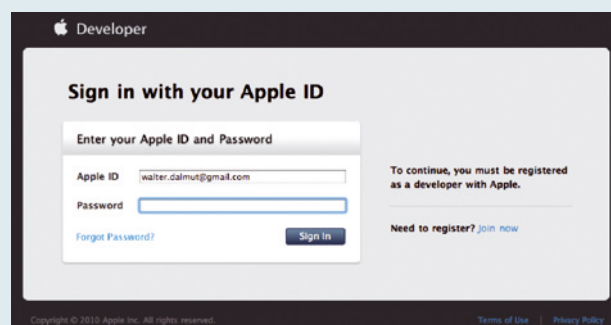
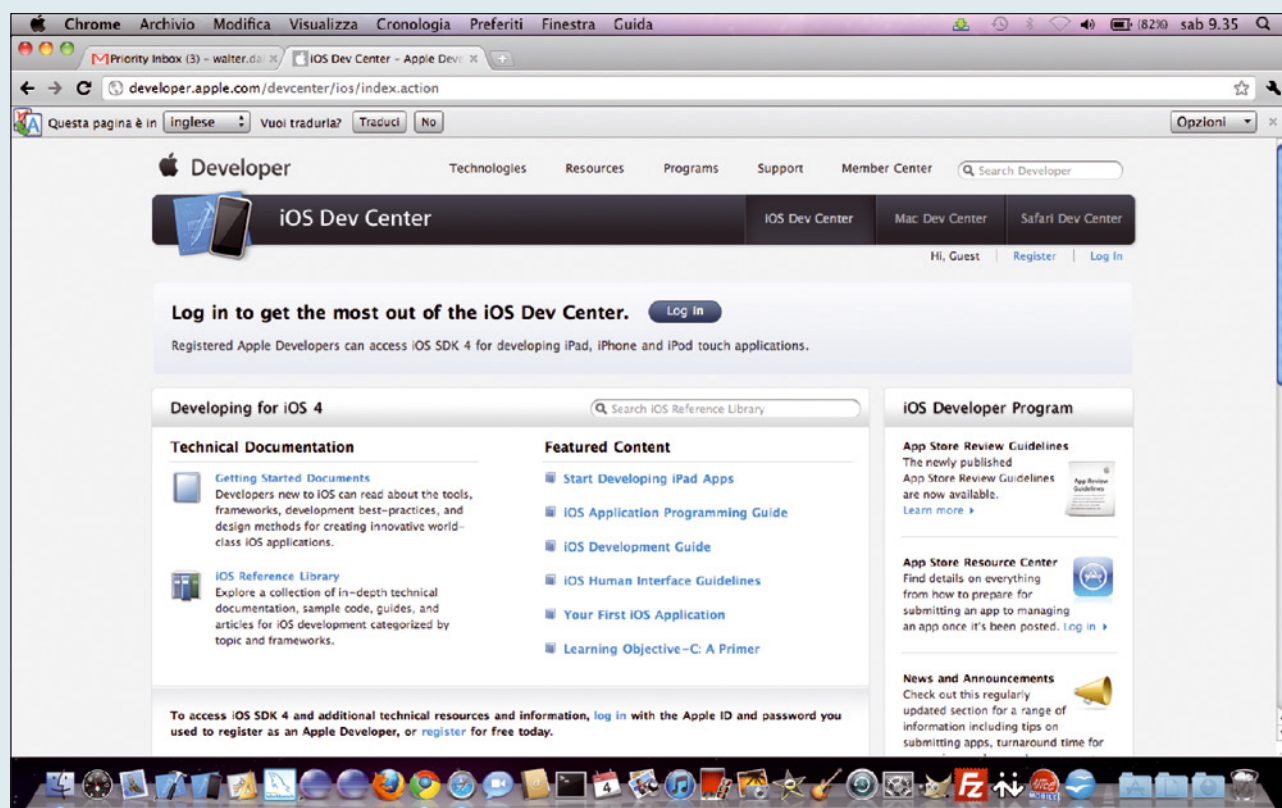


Fig. 2 - Finestra di login.

Fig. 3 - L'area riservata agli sviluppatori, da questa sezione possiamo scaricare Xcode.



L'installazione del pacchetto Xcode non richiede alcuna configurazione ad-hoc ed è possibile farla senza problematiche di sorta, semplicemente facendo sul file un doppio clic e seguendo i passi dell'installer.

Avviare X-Code

Ad installazione terminata, l'icona di Xcode non compare tra le applicazioni del computer, ma viene inserita in un altro punto del nostro file system; per questa motivazione bisogna individuarla tramite il finder del computer, come mostrato nella Fig. 4. La chiave di ricerca più proficua per questa operazione è "xcode.app". Una volta individuata l'icona del programma, facciamo doppio-clic sull'icona rappresentante il martello per avviare finalmente X-Code e poter iniziare la nostra avventura nel mondo iPhone. Nella Fig. 5 è rappresentata la finestra di introduzione di Xcode, a sinistra della quale è possibile vedere tre grossi pulsanti: il primo, chiamato "Create a new Xcode project", serve per creare un nuovo progetto; il secondo

pulsante ("Getting started with Xcode") è una guida introduttiva all'utilizzo di Xcode, una risorsa molto interessante che introduce i nuovi utenti all'utilizzo ed ai segreti di Xcode. Il terzo pulsante, infine, ci collega tramite il browser al sito di riferimento degli sviluppatori <http://developer.apple.com>. Nella sezione di destra è presente una lista delle ultime applicazioni su cui avete lavorato: se è la prima volta che

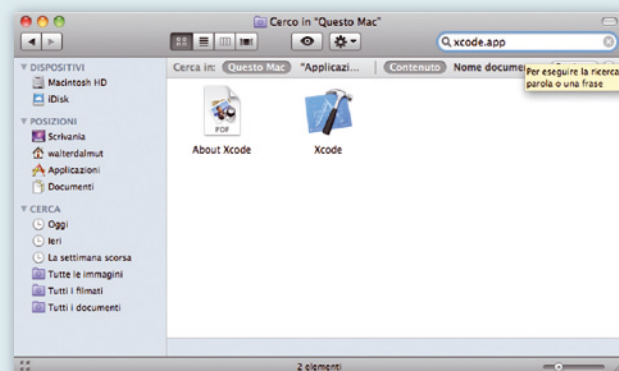


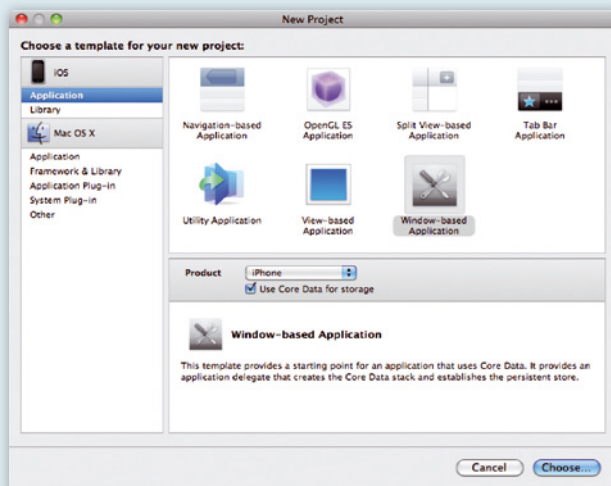
Fig. 4 - Cercare l'applicazione X-Code dopo aver effettuato l'installazione.

Fig. 5 - Intro di X-Code



installate Xcode, questa sarà vuota. In basso a sinistra, infine, c'è il pulsante "Open Other..." che serve ad aprire un progetto che non compare nel menu veloce sulla destra, in quanto non è utilizzato di recente. Il pulsante che ci interessa di più è sicuramente il primo, ossia "Create a new Xcode project", in quanto andremo subito a configurare un nuovo

Fig. 6 - Navigazione delle risorse disponibili in Xcode.



progetto; quindi facciamo clic su questo collegamento ed il sistema apre una seconda pagina dove possiamo scegliere dettagliatamente quale applicazione vogliamo realizzare. Nella Fig. 6 è rappresentata la schermata per la realizzazione di un App per iPhone; sulla sini-

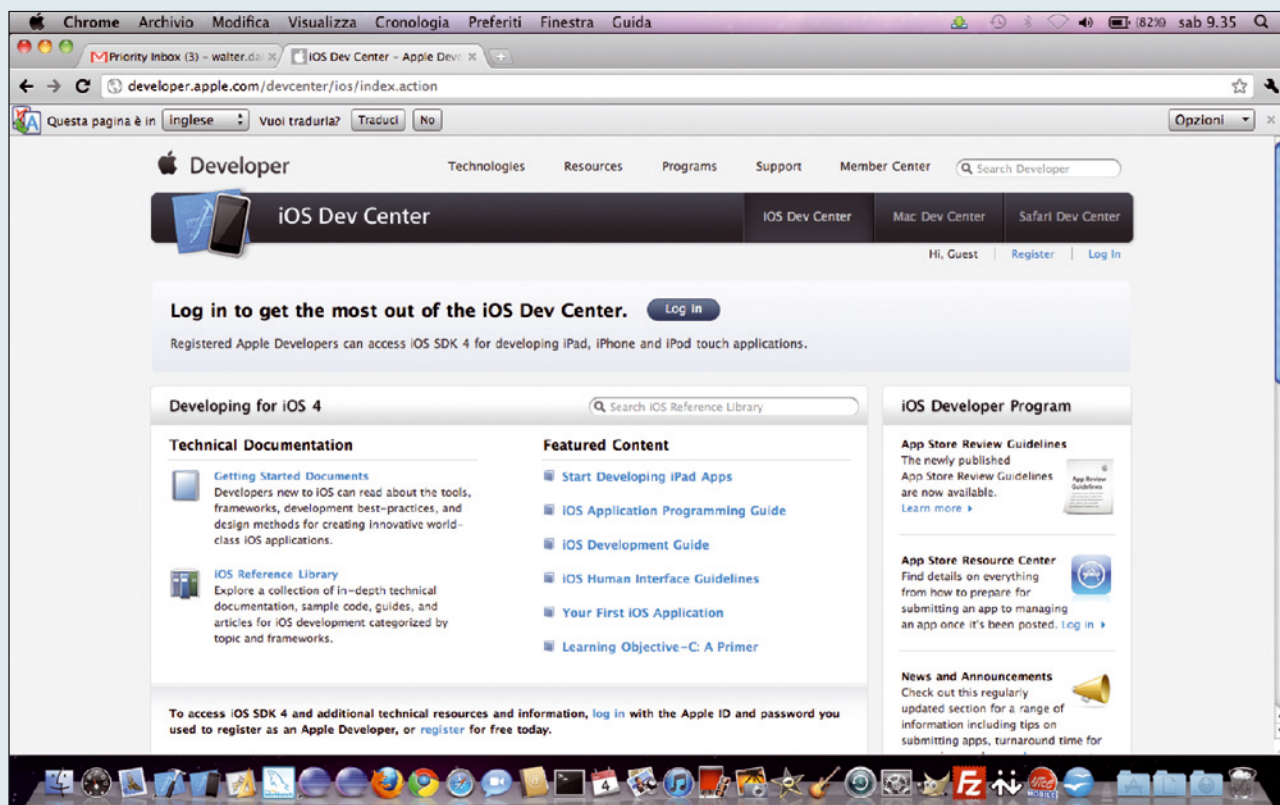


Fig. 7 - InterfaceBuilder: lo strumento per la creazione delle grafiche.

stra sono presentate due sezioni, la prima delle quali riguarda effettivamente lo sviluppo iPhone sotto l'etichetta iOS, mentre la seconda, con la dicitura Mac OS X, è dedicata allo sviluppo di applicazioni per computer Mac. In questo corso ci occuperemo solo della prima sezione e, nella fattispecie, della parte *Application*, dato che svilupperemo applicazioni stand-alone e non ci occuperemo, per il momento, dello sviluppo di librerie.

Nel riquadro grande sono presenti sette tipologie di modelli di App già pronte e disponibili per lo sviluppo dei nostri progetti; in basso, invece, si trova una descrizione di quello che andremo a realizzare, differente per ogni modello applicativo.

Tutti questi modelli sono disponibili sia per iPhone che per iPad, a parte alcuni che sono riservati ad uno solo dei due dispositivi, come la "Navigation Based Application" dedicata solo a iPhone, "Split View-Based Application" dedicata solo ad iPad, "Utility Application" solo per iPhone.

Tipicamente, quando si ha una buona conoscenza del sistema ci si dedica quasi esclusivamente alla "Window-based Application", in quanto è l'unica che permette un template universale, ovvero un singolo progetto per un'applicazione sia iPhone che iPad.

Xcode non è l'unico tool di sviluppo che ci è stato messo a disposizione dalla Apple: un altro prezioso IDE che utilizzeremo per la creazione delle nostre App è InterfaceBuilder, scopo del quale è semplificarci la vita nella creazione delle nostre "viste", ovvero la parte grafica della nostra applicazione, come mostra la Fig. 7. L'ultimo programma a nostra disposizione è il simulatore, visibile in Fig. 8. Questo prezioso strumento di sviluppo permette la simulazione delle nostre applicazioni senza obbligatoriamente sintetizzarle su un dispositivo fisico; con esso possiamo simulare un iPhone avente a bordo diverse versioni del sistema operativo o anche l'iPad, anche questo con diverse versioni dell'OS. Per ogni dispositivo possiamo simulare alcune condizioni di lavoro, come una telefonata in arrivo oppure un problema di memoria, ruotare il dispositivo, stoppare e riavviare l'applicazione, eseguire uno shake ed altro ancora. Ricordiamo, però,

che siamo sempre di fronte ad un simulatore e non ad un emulatore, quindi molte caratteristiche risultano diverse rispetto al *device* fisico, specialmente i tempi di esecuzione. Si pensi ad esempio ad un collegamento Internet, dove, a meno che non utilizzate una connessione GSM dal vostro Mac, la velocità di download e parsing delle pagine web è molto più elevata rispetto all'elaborazione sulla CPU ARM a bordo dell'iPhone, specie nei modelli più "anziani" come il 2G o il 3G.

Creazione di un progetto Xcode

Dalla schermata di selezione del template di Xcode, che abbiamo già visto nella Fig. 6, selezioniamo la "View-based Application". Inseriamo il nome del nuovo progetto (possiamo cominciare con il classico Hello World) e proseguiamo. Una volta fatto ciò, si apre una nuova finestra visibile nella Fig. 9. La finestra di Xcode 4, ovvero l'ultimo nato in casa Apple, è divisa in due sezioni per la scrittura del codice sorgente ed in tre per quello che riguarda l'integrazione di Interface Builder, necessario allo sviluppo della grafica. Nell'area a sinistra dell'IDE è presente il project manager, e nella parte centrale trovate l'editor dove scrivere il nostro codice.

Ma prima di addentrarci nel mondo iPhone dobbiamo fare una digressione sul linguaggio ObjectiveC, per fissare i concetti espressi nella prima lezione utilizzando la giusta sintassi.

Il linguaggio ObjectiveC

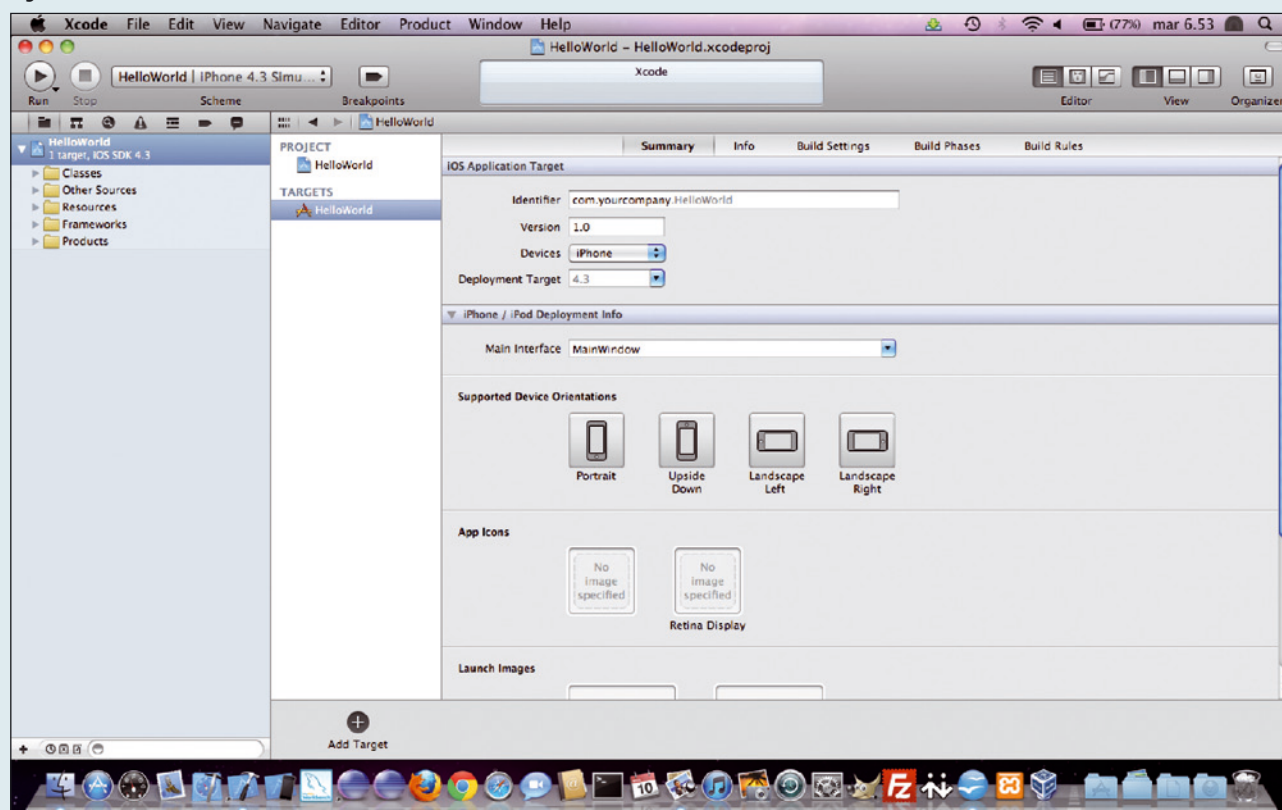
L'objectiveC, come in tutti i linguaggi di programmazione, necessita di un "entry-point", ovvero un punto da dove la nostra applicazione inizierà ad eseguire il suo codice. Nel caso dell'objective, questo entry point si trova all'interno del file chiamato *main.m*, presente nella cartella *Other Sources*, all'interno delle risorse di progetto.

Una prima differenza essenziale con il C è

Fig. 6 - Il simulatore iPhone.



Fig. 9 - Main window Xcode.



l'estensione del file: l'estensione ".m" (main.m), indica al compilatore che il codice è di tipo ObjectiveC e quindi dovrà essere trattato con le dovute considerazioni. All'interno di questo file troviamo subito un primo assaggio di ObjectiveC.

Prima di analizzare il codice ed iniziare a capirne i fondamenti, analizziamo la struttura del file, aiutandoci con il **Listato 1**. È importante sottolineare che per ObjectiveC, come per il C, è assolutamente necessaria la dichiarazione esplicita dei "prototipi", o meglio delle firme dei metodi per tutto quello che vogliamo utilizzare nel codice. Per fare questo si creano due file per ogni risorsa da aggiungere: uno di intestazione chiamato "header" che ha estensione ".h", ed un secondo che invece implementa realmente le dichiarazioni espresse nel "header", avente estensione ".c" per il linguaggio C ed ".m" per l'ObjectiveC.

Ritornando al **Listato 1**, vediamo sulla prima riga la prima direttiva del preprocessore: la chiave "import" serve per includere un altro

file, infatti, come notiamo, questo ha come estensione ".h".

La direttiva "import" dell'objectiveC è decisamente più efficiente della sua controparte in C, cioè la direttiva "include", che serve a importare nel file corrente delle definizioni. Infatti in C, per evitare il problema delle definizioni multiple, è necessario introdurre opportune direttive del preprocessore, che condizionano il processo di compilazione in modo da prevenire errori o sviste dello sviluppatore.

Nel **Listato 2** è riportato un esempio di file header in C; le direttive "ifndef" ed "endif" che trovate in esso sono "di compilazione condizionale": in pratica eseguono una direttiva di

Listato 1 - Introduzione ai files objectiveC.

```
#import <UIKit/UIKit.h>

int main(int argc, char *argv[]) {
    ... //Codice objective
}
```

controllo su una costante, che nel nostro caso si chiama “__ESEMPIO_H__”. Se questa non esiste, la si crea tramite la direttiva “define” e poi viene processato anche il contenuto, quindi tramite questo metodo anche se provassimo ad includere tante volte questo file “h”, non si verificherebbero problemi di compilazione, visto che dopo la prima volta il file verrebbe scartato dal nostro compilatore in quanto già processato.

In ObjectiveC non è necessario utilizzare tale tipo di sintassi, perché infatti possiamo utilizzare la direttiva del preprocessore “#import” ed essa si occuperà di risolvere il problema della multi-inclusione.

Ritorniamo al nostro codice del **Listato 1**, alla seconda riga del quale troviamo il tipico “entry-point” C, con i due parametri per ottenere informazioni aggiuntive che potrebbero arrivare dalla console che avvia il nostro programma. Per chi non avesse mai visto la struttura proposta alla riga 2, nella variabile “argc” sono indicati il numero di parametri passati al nostro applicativo, mentre la variabile “argv” è un puntatore ad un vettore di caratteri, quindi un puntatore di puntatore, che contiene le stringhe passate dalla console verso la nostra applicazione.

Un esempio pratico si può fare quando passiamo diversi parametri ad un programma di tipo “salutami -name elettronica in -lingua inglese”, dove *salutami* è il nome del programma, la variabile *argc* avrà valore 5, mentre la variabile *argv* sarà costituita da un array di cinque elementi ognuno dei quali rappresenta una stringa: “salutami”, “-name”, “elettronica in”, “-lingua” ed “inglese”.

Passiamo adesso ad analizzare il **Listato 3**, che integra il **Listato 1** aggiungendo la parte mancante di codice della funzione *main*. Come

Listato 2 - Esempio di dichiarazione di un file header C.

```
//esempio.h
#ifndef __ESEMPIO_H__
#define __ESEMPIO_H__

    void saluta(void);
    int testFunction(char *string);

#endif
```

Listato 3 - Corpo della funzione main.c.

```
NSAutoreleasePool *pool = [[NSAutoreleasePool alloc]
init];
int retVal = UIApplication(argc, argv, nil, nil);
[pool release];
return retVal;
```

è possibile vedere nella riga 1 del **Listato 3**, abbiamo una variabile puntatore di tipo “NSAutoreleasePool”.

Il fatto che la variabile sia un puntatore si nota dalla presenza dell’asterisco prima del nome della variabile “pool”, esattamente come in C. Con l’istruzione *NSAutoreleasePool *pool* si vuole dichiarare quindi una variabile che sarà di tipo “NSAutoreleasePool”. Un altro esempio potrebbe essere *NSString *stringa*, che identifica una variabile di nome “stringa” di tipo “NSString”. L’istruzione successiva è *[[NSAutoreleasePool alloc] init]*: questa espressione si può scomporre in due parti, una prima *[NSAutoreleasePool alloc]* ed una seconda *[pool init]*. Le parentesi quadre in ObjectiveC sono il modo sintattico con cui si invocano i metodi sugli oggetti. Il tipo *NSAutoreleasePool* è quindi il nostro primo oggetto in ObjectiveC; non lo abbiamo scritto noi ma ora sappiamo che è un oggetto. Il metodo “alloc” non è propriamente un costruttore (come quelli descritti nella prima puntata), perché serve solo per allocare in memoria per un certo tipo di oggetto. Il metodo “alloc” è un parente stretto della “malloc” del C; con questa procedura abbiamo creato fisicamente l’oggetto, ma non l’abbiamo ancora inizializzato (operazione che un costruttore di un linguaggio OOP farebbe automaticamente). Per inizializzarlo correttamente utilizzeremo il metodo “init”.

Nella seconda riga, invece, è presente una linea di codice C (*int retVal = UIApplication(argc, argv, nil, nil)*); con essa stiamo richiamando una funzione *UIApplication(...)* con i parametri necessari ed il suo valore di ritorno di tipo intero, assegnando questo valore alla variabile *retVal*. Questa funzione è definita all’interno del framework “Foundation” di *cocoa touch* e serve per inizializzare e lanciare l’applicazione. Alla riga 3 stiamo invece chiamando un metodo su un oggetto, infatti sono presenti le parentesi

Listato 4 - Esempio di prenotazione e successivamente liberazione della memoria.

```

NSObject *stringa = [[NSObject alloc] init];
//A questo punto il reference counter è diventato 1,
//questo è gestito dalla creazione dell'oggetto.
[stringa retain];
//Ora il reference counter è 2
[stringa retain];
//Reference Counter = 3
[stringa release];
//Reference Counter = 2
//L'oggetto non va via dalla memoria, esistono ancora 2 riferimenti.
[stringa release];
//Reference counter è 1
[stringa release];
//Reference counter zero ora viene rimosso dalla memoria

```

quadre e l'oggetto *NSAutoreleasePool* che abbiamo dichiarato in precedenza. Il metodo *release* è il parente della *free* del C e possiamo dire, per adesso, che serve a liberare la memoria di lavoro (vedremo la gestione della memoria più nel dettaglio, nel successivo paragrafo). Sull'ultima riga del **Listato 3** restituivamo al sistema operativo il codice di uscita della nostra applicazione allo stesso modo del C.

Gestione della memoria per iPhone

Nella prima puntata abbiamo già detto che per iPhone è necessaria la gestione della memoria da parte del programmatore; questo si fa su tutti gli oggetti tramite il metodo *release*, il nostro pseudo-distruttore. Nel C, quando chiamiamo la funzione "*free*" su un tipo di dato, questo viene immediatamente rimosso dalla memoria. Qualunque altro puntatore che mantiene il riferimento con questa porzione di memoria liberata, nel momento in cui tenterà di fare una qualunque operazione di "*read*" o "*write*" manderà in crash l'applicazione, in quanto cercherà di fare riferimento ad una porzione di memoria che non appartiene più allo spazio dell'applicazione. In ObjectiveC questo problema è stato risolto, dato che quando invochiamo il metodo "*release*" su un oggetto, questo non viene immediatamente rimosso dalla memoria, ma si diminuisce un "*reference counter*" e solo quando questo numero si riduce a zero, l'oggetto viene effettivamente rimosso dalla memoria.

Questo metodo ci fornisce una maggiore sicurezza, a patto di rilasciare sempre i riferimenti agli oggetti quando abbiamo terminato di utilizzarli, altrimenti questi resteranno sempre in memoria causando un problema di "*memory leakage*". Per capire questo concetto facciamo un esempio pratico: in ObjectiveC, per aumentare il numero di riferimenti si deve utilizzare il metodo "*retain*", mentre per ridurre il conteggio si usa il metodo "*release*". Durante i nostri esempi non utilizzeremo molto il metodo "*retain*", in quanto esistono metodologie che invocano per noi questo metodo, mentre faremo un largo uso del metodo "*release*".

Nel **Listato 4** abbiamo fatto uso di un oggetto di nome "*NSObject*" per due motivazioni: la prima è che attualmente non abbiamo ancora scritto il nostro primo oggetto; la seconda è che vogliamo sollevare una questione riguardante lo spazio dei nomi di un'applicazione. Il problema che viene introdotto è molto noto e si risolve con i "*namespaces*", ovvero lo spazio dei nomi, un modo, questo, per evitare di creare classi o funzioni con lo stesso nome, il che renderebbe impossibile la compilazione in quanto abbiamo detto che l'ObjectiveC non ammette "*overload*" dei metodi ed il C non lo ammette per le funzioni. In aggiunta, non si possono avere classi con nomi identici, altrimenti il compilatore non saprebbe quale utilizzare. Per risolvere questo problema in ObjectiveC, così come nei linguaggi che non hanno un sistema "*namespace*" ufficiale, si introduce un prefisso prima del nome come prefisso della classe. Questa soluzione è utilizzata dalla Apple stessa, infatti nel nome "*NSAutoreleasePool*" il prefisso *NS* è lo spazio di nomi scelto: infatti è l'abbreviazione di *NextStep* (quando Steve Jobs uscì dalla Apple nel 1985, fondò una società chiamata proprio *Next*, che sviluppò un sistema operativo chiamato *NextStep* che in seguito

Listato 5 - Costrutto fondamentale if-else.

```

if (1 == 1) {
    printf("Sono uguali");
} else {
    printf("Sono diversi");
}

```

Listato 6 - Costrutto fondamentale while.

```
while(i < 5)
{
    printf("%d", i);
    ++i;
}
```

divenne la base di quello che è l'attuale sistema operativo utilizzato dalla Apple, MacOS X). Vi consigliamo di utilizzare sempre un vostro spazio nomi, per non incorrere in problemi con codici Apple o comunque di terzi parti. In questo corso utilizzeremo il prefisso EI come contrazione di Elettronica In.

Iniziare a lavorare in ObjectiveC

Per scrivere realmente un po' di codice ObjectiveC, dobbiamo ancora vedere i costrutti base del linguaggio, che sono essenzialmente quelli del C, e le strutture dati base che vengono fornite direttamente dalla Apple.

Iniziamo comunque con la base di ObjectiveC, in modo da poter poi realizzare un'applicazione reale. I costrutti base *if*, *for*, *while*, sono esattamente quelli del C, con in più la possibilità di dichiarare le variabili direttamente nei cicli. Una particolarità dell'Objective C è che non è utilizzato il "NULL" bensì il "nil".

Nel **Listato 5** vediamo il costrutto "if", che, come possiamo vedere, è lo stesso del C. Nell'esempio specifico viene controllata l'uguaglianza delle espressioni. I costrutti *while* e *do/while* sono sempre quelli del C. Nel **Listato 6** vediamo un esempio di uso del *while*. Il costrutto *for*, il cui esempio applicativo è visibile nel **Listato 7**, è l'unico che oltre a mantenere l'eguaglianza con il C permette anche soluzioni di scorrimento diverse; analizzeremo queste ultime quando e se sarà necessario. Comunque è sempre possibile utilizzare il metodo standard C. In questo esempio è stata dichiarata la variabile "i" all'interno del "for"; fate attenzione a questa dichiarazione, in quanto non è ANSI C ma è possibile utilizzarla in ObjectiveC senza alcun problema. I tipi base del C sono stati ridichiarati in ObjectiveC, anche se è sempre possibile utilizzare quelli standard. Come ricorderete, in C le stringhe non esistono,

infatti ci sono i vettori di caratteri e quindi vale lo stesso discorso anche per ObjectiveC; però la Apple ha messo a nostra disposizione la classe *NSString*, semplificandoci notevolmente la vita. Vediamo subito un esempio di una stringa costante:

```
NSString *stringa = @"Io sono un oggetto";
```

Notiamo che, a differenza del C, prima dei doppi apici c'è il simbolo speciale "@", che serve proprio a indicare che è un riferimento ad un oggetto; infatti, come possiamo vedere, stiamo dichiarando un puntatore a oggetto "NSString". Quando utilizziamo la classe "NSString" dobbiamo fare attenzione, perché essa è una classe immutabile. Il termine "immutabile" indica che tale classe non può essere modificata, in quanto è sintetizzata come un vettore di caratteri e non è più possibile scegliere di aumentare o ridurre lo spazio una volta che tale vettore è stato allocato. Per questa motivazione, se stiamo lavorando con stringhe mutabili, possiamo utilizzare la classe *NSMutableString*, che permette di modificare la struttura delle stringhe riducendo i problemi di memoria, in quanto le stringhe non vengono continuamente riallocate. Anche per gli array abbiamo delle classi già pronte all'uso (*NSArray*, *NSMutableArray*, *NSSet*, *NSMutableSet*) che permettono di registrare al loro interno i nostri oggetti. Nel **Listato 8**

Listato 7 - Costrutto fondamentale for.

```
for(int i=0; i<10; i++)
{
    printf("%d", i);
}
```

Listato 8 - Esempio completo in ObjectiveC.

```
for (int i=0; i<[lista count]; i++) {
    NSLog(
        @"Indice: %d - Valore: %@",
        i,
        [lista objectAtIndex: i]
    );
}
```

Listato 9 - Primo esempio reale su iPhone.

```
#import <UIKit/UIKit.h>
#import "helloworldViewController.h"

int main(int argc, char *argv[]) {

    NSString *stringa = @"Io sono un oggetto";
    NSArray *lista = [[NSArray alloc] initWithObjects: stringa, nil];

    for (int i=0; i<[lista count]; i++) {
        NSLog(@"Indice: %d - Valore: %@",
              i,
              [lista objectAtIndex: i]
        );
    }

    NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];
    int retVal = UIApplicationMain(argc, argv, nil, nil);
    [pool release];
    return retVal;
}
```

possiamo vedere un primo esempio completo in ObjectiveC. Prima di tutto abbiamo creato una stringa, che è `NSString *stringa = @"Io sono un oggetto"`. Dopo creiamo un array di elementi chiamato `lista`.

Come abbiamo già spiegato, dobbiamo allocare lo spazio in memoria di tipo `"NSArray"` tramite la direttiva `"[NSArray alloc]"` e poi chiamiamo il nostro costruttore `"initWithObjects"`. Questo costruttore necessita di una serie di oggetti separati da virgola e conclusi con il terminatore `"nil"`. Dopo aver allocato l'array, o meglio un oggetto `"NSArray"` che contiene l'array reale, cicliamo gli elementi al suo interno tramite il costrutto `for`. In questa dichiarazione stiamo invocando il metodo `"count"` sull'oggetto `"lista"`, che restituisce il numero di elementi registrati: in questo caso, solo uno. Per ogni elemento nella lista utilizziamo la funzione `NSLog`, che scrive nella console iPhone un nostro messaggio, contenente l'indice ed il valore dell'elemento immagazzinato nella

posizione corrente della lista. La struttura di `NSLog` è la seguente: `NSLog(NSString *, ...)` dove i dots indicano un numero variabile di parametri. La prima parte deve essere una stringa e quindi deve iniziare con il carattere `"@"`, in quanto è un puntatore, e poi un numero variabile di parametri come descritto in precedenza. Il contenuto del primo parametro è: `"Indice: %d - Valore: %@"`. Il primo tipo che vogliamo rappresentare è un normale numero decimale (`%d`) mentre il secondo (`%@`) indica che vogliamo rappresentare un

oggetto come se fosse una stringa. Prendiamo adesso il nostro progetto `"HelloWorld"`, che utilizzeremo ancora nella prossima puntata, apriamo il file `"main.m"` posto sotto il gruppo chiamato `"Other Sources"` e scriviamo al suo interno la stringa `NSString *stringa = @"Io sono un oggetto"`. Il file `"main.m"` dovrebbe apparire come il **Listato 9**. Per eseguire questo codice dobbiamo fare clic sul pulsante `Run` in alto a sinistra nella barra di controllo dell'TDE, come visibile in **Fig. 10**. Questa operazione effettuerà la compilazione ed avvierà la console nella quale potremmo visualizzare i messaggi stampati dalla funzione `NSLog`. Il risultato di questa operazione è visibile in **Fig. 11**. Dopo aver visto il messaggio possiamo chiudere il simulatore facendo la combinazione `MELA+Q`. Per questa lezione abbiamo terminato, nella prossima vedremo i concetti del framework MVC (Model View Controller) pesantemente utilizzato nello sviluppo delle applicazioni per iPhone.



Fig. 10 - La console Xcode dopo che abbiamo avviato la nostra applicazione.

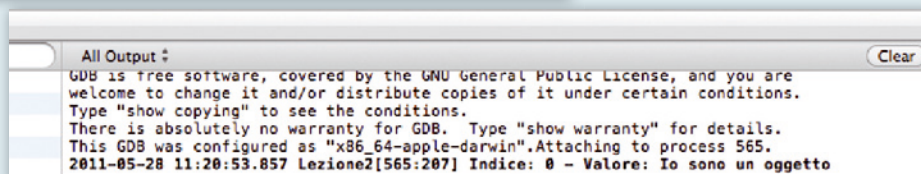


Fig. 10 - Barra di controllo di Xcode.

Realizziamo l'alimentatore da rete per caricare nella maniera ottimale le quattro batterie al piombo della nostra auto elettrica.

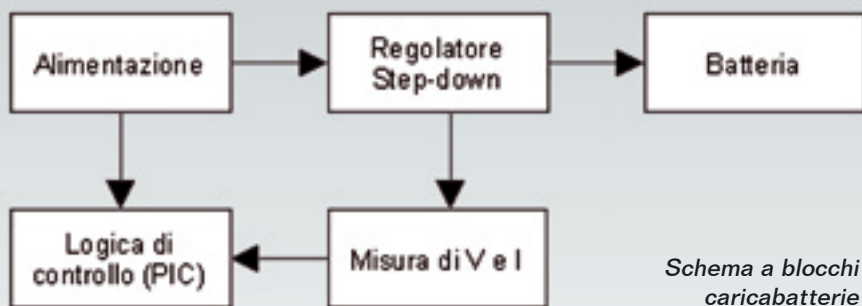
AUSSIE ZERO.

IL CARICABATTERIE

..... di FABIO FIORAVANZO

Due mesi fa abbiamo iniziato a descrivere la nostra vettura ad energia solare, cominciando dalla meccanica e dall'impianto elettrico (dando particolare risalto ai motori di trazione ed al sistema di controllo della velocità) e proseguendo, il mese scorso, con un articolo propedeutico alla comprensione delle problematiche riguardanti l'alimentazione dei veicoli elettrici. Finalmente possiamo proporvi il primo progetto pratico che riguarda l'elettronica di bordo della nostra Aussie Zero. E

cosa poteva essere, se non un caricabatterie? Ma non uno qualsiasi, bensì un caricabatterie specifico per batterie al piombo e con la possibilità di caricare anche pacchi batterie fino a 4 elementi in serie. Per il nostro caricabatterie non abbiamo adottato complicati schemi switching che lavorano sulla trasformazione della tensione alternata (aumento della tensione di ingresso e della sua frequenza per diminuire le dimensioni del trasformatore); questo perché una circuitazione siffatta avrebbe compor-



tato non pochi problemi nella realizzazione del trasformatore, che va progettato e fatto ad hoc per l'applicazione specifica, precludendo di fatto la realizzazione di questa tipologia di circuito da parte dello sperimentatore o appassionato di elettronica. Abbiamo quindi optato per uno schema con trasformatore e regolatore di tipo step-down, che consente di realizzare facilmente il circuito con pezzi comunemente reperibili in commercio. Le batterie di Aussie Zero sono quattro in serie da 12V - 60 Ah, cosicché la tensione totale risultante è di 48V. Fare un caricabatterie che fosse in grado di caricare completamente accumulatori di questa taglia, avrebbe comportato l'utilizzo di componenti dalle dimensioni importanti, in primis il trasformatore. Ci siamo quindi "accontentati", si fa per dire, di un circuito in grado di erogare al massimo 6 A a 48 V, capace cioè di operare la cosiddetta ricarica lenta di un pacco batterie come quello della nostra vettura elettrica, ma nel contempo in grado di caricare al meglio, anche se nuove o completamente scariche, batterie da qualche decina di Ah. Inoltre, abbiamo pensato di rendere ancor più flessibile il nostro caricabatterie, permettendo di selezionare la tensione della batteria fra 12, 24, 36, 48 V. In questo modo si possono ricaricare anche accumulatori da 24 V o serie di batterie. Dulcis in fundo, abbiamo aggiunto la funzione di mantenimento,

in grado cioè di dare una debole corrente di ricarica per mantenere in forma batterie in stoccaggio o in utilizzo non continuativo. In tal modo abbiamo ottenuto uno strumento dalla massima flessibilità, in grado di ricaricare o "mantenere", ad esempio, la batteria della moto durante il periodo invernale, di veicoli elettrici per bambini, o di piccoli macchinari.

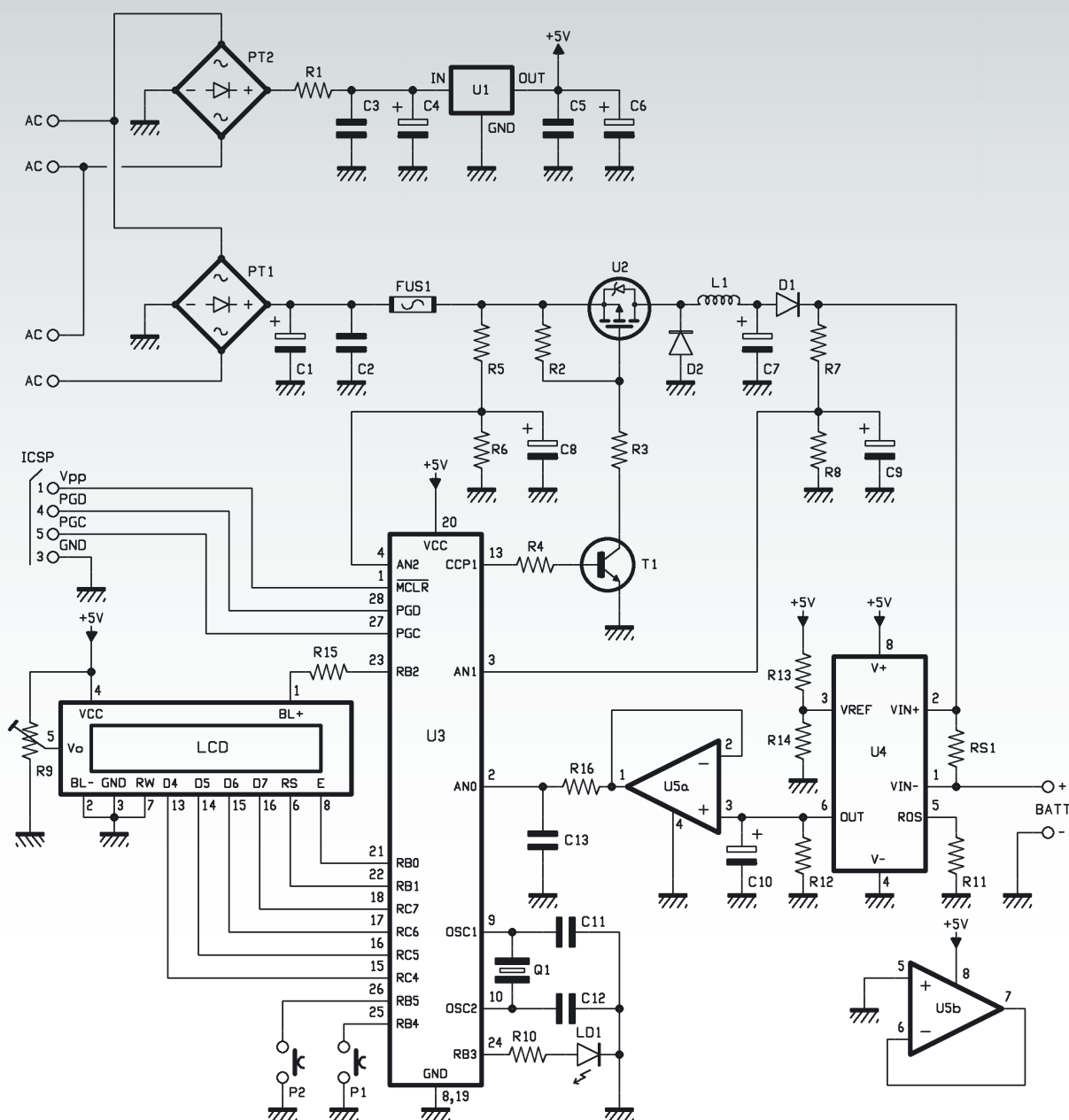
SCHEMA ELETTRICO

Lo schema elettrico si può idealmente suddividere in quattro parti: il blocco di alimentazione, la parte di regolazione della tensione, la sezione logica di controllo e quella di misura. Partiamo dall'alimentazione: date le caratteristiche richieste al sistema, abbiamo utilizzato un bel trasformatore toroidale da 230 V - 300 VA della Velleman, avente due secondari da 21 V capaci di erogare 7,14 A ciascuno. I due secondari sono collegati in serie ed i loro estremi liberi sono applicati agli ingressi di un ponte raddrizzatore da 600 V e 8 A di picco; all'uscita (+ e -) di questo ponte è stato posto un condensatore di livellamento da 470uF, ai capi del quale otteniamo circa 58 V conti-

nui. Per tale ragione l'elettrolitico C1 è da 100 V, dato che è buona norma dimensionare i condensatori di filtro per una tensione di lavoro circa doppia di quella nominale, per mettersi al riparo dagli sbalzi di tensione, ma anche dalla tolleranza (abbastanza larga) su tensione e capacità che gli elettrolitici presentano. Da uno dei due secondari si deriva la tensione necessaria a far lavorare la parte logica del circuito: i 5 V occorrenti al PIC e agli altri integrati utilizzati, sono ottenuti da un classico regolatore lineare 7805 (U1) contornato da condensatori di filtro e livellamento della tensione. Poiché i 21 V in ingresso sarebbero comunque alti per il nostro regolatore (ricordiamo che nei regolatori lineari la potenza dissipata è pari al prodotto della corrente erogata per la differenza tra tensione in ingresso ed in uscita) abbiamo introdotto la resistenza R1, che allevia il lavoro del regolatore facendo cadere su di sé parte della tensione in ingresso. Vediamo adesso il regolatore per la carica della batteria, che è del tipo step-down o buck: è costituito dai componenti U2, D2, L1 e C7. Per sua natura, il buck può solamente erogare tensioni inferiori a quella in ingresso; nel nostro caso, adottiamo la classica configurazione dove l'interruttore di potenza è costituito da un MOSFET di tipo P, in quanto se si usasse un canale N occorrerebbe un particolare circuito a pompa di carica per pilotare il gate con una tensione superiore a quella

Tensione di alimentazione	230 Vca
Protezione	Fusibile da 10 A
Tensione batteria	12, 24, 36, 48 V selezionabile
Corrente di ricarica	0,2÷ 6 A selezionabile
Controlli di sicurezza	Sovraccarica, sovratensione, cortocircuito
Tipologie di ricarica	Normale, a tempo, di mantenimento

Caratteristiche principali del caricabatterie.



da fornire alle batterie, cosa che complicherebbe inutilmente il circuito. Il componente utilizzato è della IRF e presenta in particolare, una $R_{DS(ON)}$ di soli 117 mΩ, una massima corrente di drain di ben 23 A ed una tensione V_{DS} di 100 V. Anche per i diodi la scelta è stata oculata: nei circuiti switching è molto importante il tempo di recovery (**trr**) di questi componenti, pertanto ci siamo orientati verso un modello di diodo ultrafast

della ST che ha, un **trr** di appena 17nsec! Inoltre nell'alimentatore buck, il diodo deve avere una V_{RRM} ed una I_F superiori ai valori di ingresso. Ma anche qui non abbiamo di che preoccuparci, avendo il diodo da noi scelto $V_{RRM}=100$ V e $I_F=8$ A. Il secondo diodo dopo l'induttanza è necessario per evitare contro alimentazioni del circuito da parte della batteria da caricare. Per il dimensionamento della bobina

si è fatto ricorso alle "classiche" formule dello step-down:

$$L > V_{OUT} * (V_{IN} - V_{DS} - V_{OUT}) / (2 * V_{IN} * f_{PWM} * I) \approx 208 \mu H$$

Come da formula, il valore effettivo dell'induttanza deve essere maggiore di quello calcolato. La scelta è caduta su un'induttanza della Murata da 220 μH, e $I_{DC-MAX}=5,5$ A. Lavorando ad alta frequenza (il nostro switching opera a 10 kHz), assumono importanza



Milano trionfa alla Shell Eco Marathon

Si è conclusa bene per i team italiani l'edizione 2011 della Shell Eco-marathon Europe, che ha visto 187 squadre sfidarsi a suon di km percorsi sul circuito EuroSpeedway di Lausitz, in Germania, dal 26 al 28 maggio. Quest'anno sono stati stabiliti nuovi record, uno dei quali nella classe E-mobility che ha debuttato in questa edizione. Il team mecc-Sun del Politecnico di Milano ha corso con un Prototipo alimentato a pannelli solari, stabilendo il nuovo record nella categoria prototipi: il veicolo ha percorso ben 1.108 km consumando solo 1 kWh di energia. Il vincitore della 27° edizione della Shell Eco-marathon nella classe tradizionale è stato il team Microjoule dell'istituto francese La Joliverie, che ha percorso 3.688,2 Km per litro con un veicolo alimentato a benzina. A Lausitz è stato premiato anche quest'anno il veicolo che ha fatto registrare le minori emissioni di CO₂: appartiene al team tedesco proTRon, che nella categoria UrbanConcept è sceso sotto la soglia dei 2 g/km (1,972 g/Km, per l'esattezza).

le componenti parassite dei vari elementi circuitali, in particolare del condensatore (C7) che deve essere scelto con bassa ESR (resistenza serie equivalente), al fine di ridurre la dissipazione termica e non pregiudicare le performance dell'alimentatore.

Analizziamo adesso la parte logica, che è incentrata su un microcontrollore PIC della serie 18F, operante alla frequenza di 20

MHz; il suo compito è gestire il display 16 caratteri per 2 colonne e i due pulsanti attraverso i quali si visualizzano e regolano le varie funzioni del caricabatterie, ma soprattutto comandare il regolatore step-down, in base alle tensioni e correnti che legge dai suoi ingressi analogici, al fine di avere in uscita i valori esatti per la ricarica della batteria.

Le tensioni a monte e a valle del-

lo step-down sono lette tramite partitori resistivi utili a riportare agli ingressi del PIC tensioni al di sotto del massimo sopportabile dall'integrato. Notate che sono state usate resistenze con tolleranza all'1 % per garantire una buona accuratezza delle misure. Gli elettrolitici in parallelo agli ingressi analogici livellano la tensione letta, evitando che il PIC possa leggere valori non corretti. La misura della corrente è sempre un'operazione difficoltosa, specie quando, come nel nostro caso, dev'essere molto precisa. Ci siamo quindi affidati ad un integrato dedicato appositamente alla misura di correnti continue, l'INA170 della Burr Brown (U4); si tratta di un "High-Side, Bidirectional Current Shunt Monitor". Grazie allo shunt RS1, esso effettua una precisa misura della corrente, fornendo una tensione proporzionale ad essa. Scegliendo opportunamente il valore delle resistenze di contorno dell'integrato è possibile scegliere il range di correnti da misurare, se sono solo positive o sia positive che negative, e la tensione proporzionale in uscita.

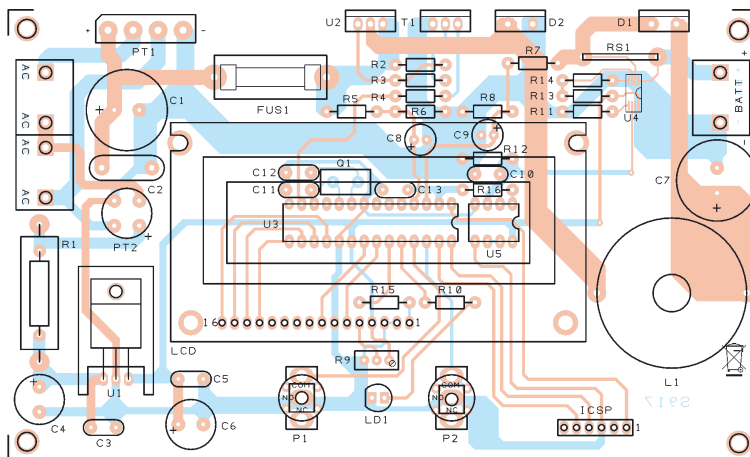
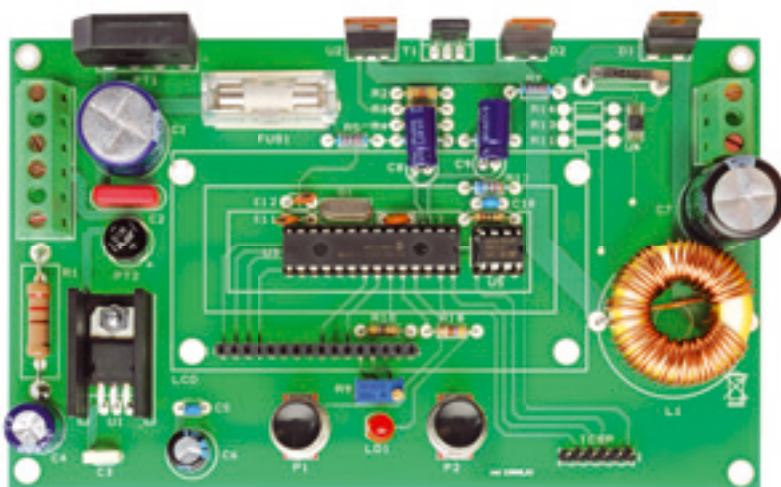
Per una descrizione più approfondita dell'integrato vi rimandiamo al progetto del multimetro digitale presentato nel fascicolo 154 di Elettronica In. Nel nostro caso, dovendo misurare solo correnti positive, abbiamo sfruttato l'intero range di variazione e pertanto il pin 3 dell'integrato (Reference Voltage Input) non è collegato, né tantomeno il pin 5 (Offset Resistor).

La resistenza di uscita (R12) è stata dimensionata in modo da sfruttare al meglio tutta la scala del convertitore analogico del PIC. Senza offset, la funzione di trasferimento dell'integrato è:

$$I_{OUT} = gm * V \text{ con } gm = 1.000 \mu A/V$$

Elenco Componenti:

R1: 220 ohm 3 W
 R2: 4,7 kohm
 R3: 10 kohm
 R4: 1 kohm
 R5: 12,1 kohm 1%
 R6: 1 kohm 1%
 R7: 12,1 kohm 1%
 R8: 1 kohm 1%
 R9: Trimmer 10 kohm multigiro
 R10: 470 ohm
 R11: -
 R12: 68,1 kohm 1%
 R13: -
 R14: -
 R15: 47 ohm
 R16: 10 ohm
 RS1: 0,01 ohm 1%
 C1: 470 μ F 100 VL elettrolitico
 C2, C3: 100 nF multistrato 100 VL
 C4: 470 μ F 35 VL elettrolitico
 C5: 100 nF multistrato
 C6: 470 μ F 16 VL elettrolitico
 C7: 220 μ F 100 VL elettrolitico
 C8, C9: 100 μ F 25 VL elettrolitico
 C10: 100 nF multistrato
 C11, C12: 15 pF ceramico
 C13: 220 pF ceramico
 D1: STTH802
 D2: STTH802
 LD1: LED 5 mm rosso
 T1: BD681
 U1: 7805
 U2: IRF9540N
 U3: PIC18F2620-I/P (MF917)
 U4: INA170
 U5: LM358N
 L1: Induttanza 220 μ H 5,5 A
 FUS1: Fusibile 10A
 PT1: Ponte diodi KBU805
 PT2: Ponte diodi W06M



Il piano di montaggio è illustrato ridotto al 65 %.

P1: Pulsante 8701
 P1: Pulsante 8701
 LCD: CDL4162
 Q1: Quarzo 20 MHz

Varie:
 - Zoccolo 4+4

- Zoccolo 14+14
 - Morsetto 2 poli passo 10 mm (3 pz.)
 - Portafusibile da c.s.
 - Pin strip maschio 6 poli
 - Pin strip maschio 16 poli
 - Pin strip femmina 16 poli

- Dissipatore ML26
 - Vite 10 mm 3 MA
 - Dado 3 MA
 - Cappuccio copri pulsante CV3 (2 pz.)
 - Circuito stampato

Per avere misure affidabili, il costruttore consiglia di non superare i 500 mV tra gli ingressi V_{IN+} e V_{IN-} .

Abbiamo quindi scelto uno shunt (resistenza RS1 nello schema) da 0,01 Ω , in modo che alla massima corrente di lavoro (6 A) abbiamo una ΔV di 60 mV.

La massima tensione di uscita dev'essere inferiore al massimo livello tollerabile dagli ingressi

del PIC. Ciò si ottiene tarando opportunamente il valore della resistenza di uscita R12. Con i valori da noi scelti si ha:

$$V_{out} = I_{out} * RS1 * R12 * g_m = 6 * 0,01 * 68K * 1000 = 4,08V$$

C10 forma, insieme ad R12, un filtro passa basso per il segnale di uscita.

Con i valori scelti, la frequenza di taglio è 23 Hz. Poiché la resistenza di uscita dell'INA170 è più

elevata di quanto richiesto da Microchip per gli ingressi analogici dei PIC, abbiamo introdotto un buffer (U5) che grazie alla sua elevata resistenza di ingresso e bassa resistenza di uscita, soddisfa le nostre esigenze.

All'uscita dell'amplificatore operazione è presente un ulteriore filtro passa basso che elimina eventuali disturbi che dovessero presentarsi.

Tabella 1

BATTERIE AL PIOMBO					
Vnominale [V]	Icarica	Vmin scar. [V]	Vmax car. [V]	Imant.	Vallarme
2	0,1C÷0,4C	1,6	2,4	0,1C	2,45

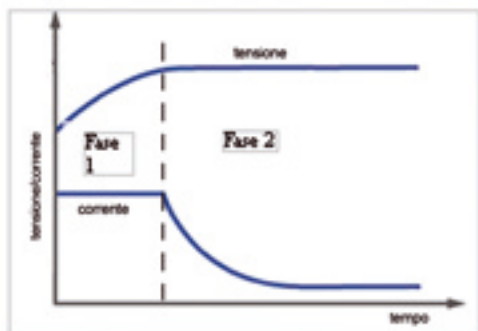


Fig. 1 - Fasi di carica di una batteria al Piombo.

REALIZZAZIONE PRATICA

Per costruire l'alimentatore partiamo dal circuito stampato, del quale potete scaricare le tracce rame dal nostro sito www.elettro-nicain.it. Rispettate fedelmente la traccia perché le piste di potenza sono dimensionate per le correnti in gioco e cambiarle potrebbe compromettere il funzionamento dell'intero circuito. Una volta in possesso dello stampato, iniziate a montarvi i componenti; l'unica criticità riguarda U4, che essendo in case di tipo SMD necessita di qualche accortezza. Aiutatevi con uno stuzzicadente per tener fermo l'integrato, quindi saldate un pin laterale, e poi quello diametralmente opposto e via via gli altri. Fate attenzione a non insistere troppo con il saldatore, pena il rischio di danneggiare irreparabilmente l'integrato. Gli altri componenti sono tutti di tipo tradizionale, per cui vale la solita regola di partire da quelli a più basso profilo e poi passare a quelli più grandi. Il regolatore U1 va montato accoppiato ad un dissipatore ad "U". Il display va dotato di uno strip maschio a passo 2,54 mm a pin lunghi, che servirà ad inserirlo in uno strip femmina che monterete sullo

stampato; quattro colonnette distanziali da 15 mm avvitate nei fori appositamente previsti, sosterranno l'LCD.

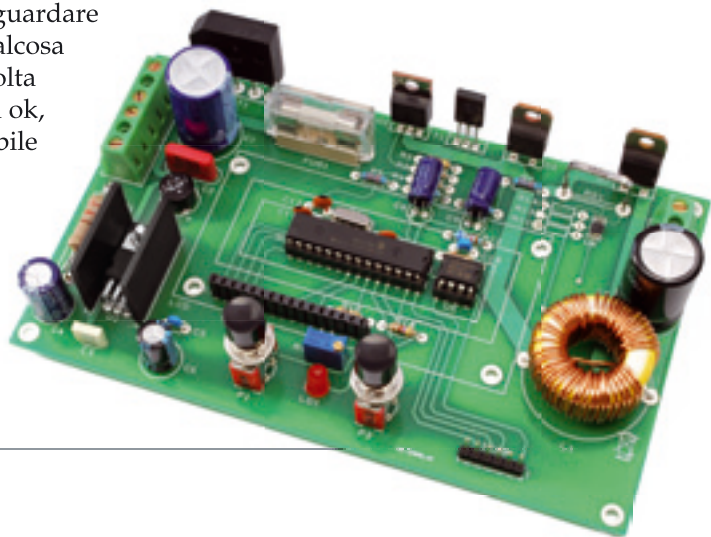
Tutti i componenti di potenza sono stati posizionati da un lato, in modo che possano essere fissati ad un dissipatore di dimensioni pari a 150x60x25 mm, avente resistenza termica di circa 3 °C/W. Per fissare i semiconduttori ad esso, sono stati utilizzati appositi kit di isolamento, al fine di evitare cortocircuiti tra componenti a potenziale diverso. I condensatori e l'induttanza saranno gli ultimi componenti a prendere posto sullo stampato.

Una volta finito l'assemblaggio, si può collegare il trasformatore alla scheda. Un doveroso controllo alle tensioni sia della parte di potenza che di quella logica. Se tutto è ok si possono montare i due integrati U3 e U4. Il micro si programma facilmente grazie al connettore ICSP previsto in basso a destra nello stampato, inserendovi il firmware scaricabile dal nostro sito. Una volta programmato il PIC e testato il funzionamento della parte logica (display, pulsanti, LED di stato), si può inserire il fusibile F1 e verificare il funzionamento della parte di potenza. Magari inizialmente è meglio inserire un fusibile da 1A, in modo da salvaguardare il circuito nel caso qualcosa andasse storto; una volta verificato che tutto sia ok, si può montare il fusibile previsto.

FUNZIONAMENTO

Passiamo ora a descrivere le modalità operative del circuito. Come detto, il PIC gesti-

sce sia l'interfaccia verso l'utente, sia il convertitore step-down, comandandolo con l'opportuno segnale PWM generato dai calcoli seguiti alla lettura delle tensioni e corrente circolante nella sezione di potenza del circuito. Come già scritto in altre occasioni, la migliore modalità di carica delle batterie al piombo prevede una prima fase in cui il caricabatterie fornisce una corrente costante pari a 0,4 C, fintantoché la tensione della batteria non arriva al suo valore massimo (circa 2,4 V per cella), seguita poi da una fase a tensione costante in cui la corrente decresce; una volta arrivati al di sotto di 0,1 C, la batteria viene considerata carica. Abbiamo scelto queste soglie perché se è vero che le batterie al piombo sopportano anche cariche rapide con correnti oltre C, a lungo andare ciò può portare ad un precoce invecchiamento degli elementi ed al danneggiamento permanente delle piastre a causa delle elevate temperature che si vengono a creare. Ricordiamo che la corrente di carica si indica riferendosi alla capacità della batteria "C" che si misura in Ah; quindi per una batteria da 5 Ah, caricarla a 0,1 C significa darle una corrente di 500 mA. Nella **Tabella 1** sono indicati alcuni dati tipici di una cella al piombo, e nel grafico vengono evidenziate le fasi di carica.



Una volta acceso il circuito, dopo la schermata iniziale, in cui è indicata la versione del SW, viene visualizzata la schermata in cui si sceglie il tipo di batteria. Con il pulsante sinistro si esegue lo scroll tra i valori 12, 24, 36, 48 V, con il pulsante destro si conferma la scelta. Si passa poi alla selezione della funzione desiderata tra:

- carica normale;
- carica di mantenimento;
- carica a tempo;
- setup.

La carica normale segue il ciclo descritto poc'anzi, con le due fasi a corrente e poi a tensione costante; si consiglia di impostare una corrente di ricarica pari a 0,4 C. La carica di mantenimento, è quella che consente il rimessaggio e "sostentamento" di batterie che, pur non utilizzate, devono essere mantenute cariche per evitare la solfatazione. Il circuito lavora a corrente costante da impostare a cura dell'utente a 0,1 C. Poiché in questa modalità il caricabatterie può rimanere in funzione anche per diversi mesi, la retroilluminazione del display si disattiva automaticamente. Essa può essere attivata/disattivata premendo il pulsante destro. Premendo invece il pulsante sinistro si esce da questa modalità, e si torna al menu principale (Fig. 2). La funzione di carica a tempo consente di caricare la batteria con le stesse modalità della carica normale, però il sistema controlla anche il tempo di carica, e la interrompe non appena superato il tempo che è stato impostato. Questa funzione è utile per definire a proprio piacimento il tempo di ricarica. Infine, il menu *Setup* consente di impostare alcuni parametri del caricabatterie. In esso (Fig. 3) abbiamo le seguenti voci:

- Safety Time;

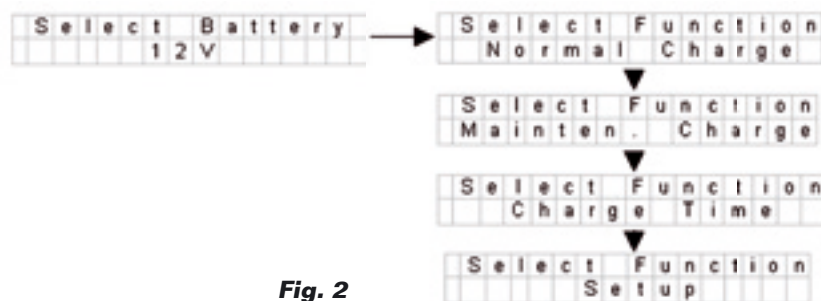


Fig. 2



Fig. 3



Fig. 4

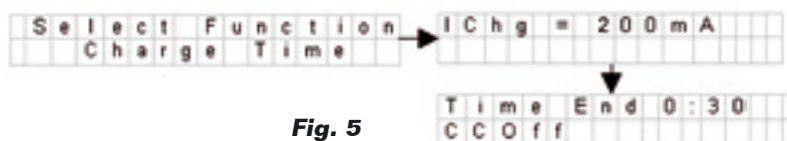


Fig. 5

- Capacity Cut-Off;
- Lighting.

Come sempre, si effettua il cambiamento con il tasto sinistro e si conferma con il destro.

Il Safety Time, se attivato, permette un controllo sul tempo di ricarica (tempo che viene impostato successivamente) oltre il quale il processo di ricarica viene comunque interrotto anche se non sono state raggiunte le condizioni di fine carica. Tale funzione è utile nel caso ci si trovi a ricaricare una batteria in pessime condizioni o che magari è danneggiata internamente o solfata. In questi casi potrebbe accadere che non vengano mai raggiunte le condizioni di fine carica, con la conseguenza che il caricabatterie continuerebbe a lavorare inutilmente scaldando la

batteria e dissipando energia.

La funzione Capacity Cut-Off permette invece di impostare un livello di capacità oltre il quale la carica viene interrotta. Il nostro sistema infatti, oltre a tensione, corrente e tempo di ricarica, misura anche i mAh forniti alla batteria; grazie a questa funzione si può decidere esattamente quanto caricare l'accumulatore. Se la capacità impostata è "0" mAh (valore predefinito), la funzione non è attiva. Con il pulsante sinistro si incrementa il valore di capacità da un minimo di 100 mAh fino ad un massimo di 60 Ah, dopodiché si riparte da 0 mAh. La regolazione avviene a step di 100 mAh fino a 5 Ah, e poi a step di 1 Ah fino a 60 Ah.

Infine, il menu *Lighting* consente di avere attiva o meno la retroilluminazione del display.

12:34 24.5V 6.5A
CHARGE 3535mAh

Fig. 6 - Lo stato del display durante la carica delle batterie.

12:34 24.5V 6.5A
END TIME 3535mAh

Fig. 7

Mainten. Charge
12.4V 1.3A

Fig. 8

Torniamo ora alle funzioni di ricarica: dopo aver selezionato quale delle tre tipologie di ricarica si desidera, si passa ad impostare la corrente di carica. In genere per le batterie al piombo il range va da un massimo di 2 C per una carica rapida, fino a 0,1÷0,4 C per una carica lenta o di mantenimento. Il menu consente di selezionare una corrente di carica da 200 mA fino a 6A. Fino a 2 A, gli step di incremento sono pari a 200 mA; oltre, gli step di incremento sono pari a 200 mA. Arrivati a 6 A, si riparte da 100 mA. La pressione continua del tasto sinistro incrementa automaticamente il valore (Fig. 4). Una volta impostata la corrente di carica, si fa un primo controllo

della presenza o meno della batteria e del corretto collegamento ai morsetti di uscita. Se il sistema rileva una tensione troppo bassa per la tipologia di batteria che è stata impostata (per batterie a 12 V abbiamo considerato un valore pari a 8 V), viene segnalata l'anomalia sul display: "No battery!". Viceversa, se si rileva una tensione troppo alta (magari perché erroneamente abbiamo impostato una tensione di 24V, ma la nostra batteria è da 48V), allora sul display appare: "VBatt too high!". In entrambi i casi una pressione del pulsante destro riporterà il sistema alla schermata iniziale. Se invece il controllo ha dato esito positivo, si passa all'impostazione del tempo di ricarica. Questo menu (Fig. 5) appare solo se in precedenza era stata attivata la funzione Safety Time o la ricarica a tempo. La grandezza da impostare è "Time End" nel caso di carica a tempo e "Safe Time" se è stata attivata la funzione Safety Time. Nel caso in cui fosse stato attivato anche il controllo di capacità, nella seconda riga appare "CCOff". Il tempo può essere impostato da un minimo di 30 minuti fino ad un massimo di 20 ore e 30 minuti, dopodiché si riparte dal tempo minimo. Gli step di incremento sono di 30 minuti e, come per gli altri menu, la pressione prolungata del pulsante sinistro consente l'incremento automatico del tempo. Terminata anche quest'ultima regolazione, confermando con il tasto destro si dà inizio al vero e proprio processo di ricarica. Il display mostra (Fig. 6) sulla prima riga il tempo di ricarica: fino ad 1 ora vengono visualizzati minuti e secondi, poi ore e minuti. Seguono la tensione della batteria e la corrente di carica. Nella seconda riga viene indicata la tipologia di carica che il sistema sta facendo:

Tabella 2

CAPACITÀ BATTERIA (Ah)	TEMPO (h)
2	1,5
8	6
20	20
32	36

- "Charge" per carica normale;
- "Time Charge" per carica a tempo.

Infine vengono visualizzati i mAh forniti alla batteria fino ad una capacità di 5 Ah, raggiunta la quale si passa a visualizzare gli Ah. In qualsiasi momento si può interrompere il ciclo, premendo il tasto sinistro, tornando così al menu principale.

Una volta raggiunte le condizioni impostate inizialmente, il processo di carica termina, il timer si ferma e nella seconda riga del display viene indicato secondo quale modalità il caricabatterie ha terminato la carica (Fig. 7):

- "End Char" se la carica è terminata per raggiungimento delle condizioni "standard";
- "End Time" se la carica è terminata per raggiungimento del tempo impostato;
- "TCOff" se la carica è terminata per superamento del tempo limite impostato;
- "CCOff" se la carica è terminata per superamento della capacità limite impostata.

Tale condizione permane per 30 minuti dopodiché, se non è stato premuto alcun pulsante e non è stata scollegata la batteria, il circuito passa automaticamente alla carica di mantenimento.

Nella carica di mantenimento, la schermata del display indica solo la tensione e la corrente di ricarica della batteria (Fig. 8). Il LED di stato lampeggia ogni 5 secondi, ad indicare che il caricabatterie sta funzionando correttamente. Con una pressione del pulsante destro si può attivare/disattivare



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono di facile reperibilità. Il master del circuito stampato ed il firmware per programmare il PIC possono essere scaricati gratuitamente dal sito della rivista (www.elettronica.in.it). Il microcontrollore è anche disponibile già programmato (cod. MF917) al prezzo di 18,00 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>

la retroilluminazione del display, mentre premendo il pulsante sinistro si esce dal ciclo, tornando al menu principale.

Durante il processo di carica (qualunque dei tre sia stato selezionato), oltre a monitorare costantemente tensione e corrente, il programma effettua una serie di controlli al fine di evitare danneggiamenti della batteria e del caricabatterie. Se in uscita si trova una tensione minore di quella minima per quella data batteria (abbiamo fissato 6 V per una batteria da 12 V), ma la corrente è inferiore ad 1A per un certo periodo di tempo, significa che la batteria è stata sconnessa ed appare il messaggio: "No Battery!". Se invece la tensione in uscita supera il livello massimo consentito (per le celle al piombo è 2,45V),

la batteria stessa rischia di andare in ebollizione e di danneggiarsi irreparabilmente. In questo caso appare il messaggio: "Vout too High!". Infine, se il programma rileva una tensione in uscita inferiore alla minima (i soliti 8 V per una batteria da 12 V), ed una corrente superiore ad 1 A, allora ci troviamo in presenza di un cortocircuito ai morsetti di uscita ed appare il messaggio: "Short Circuit!".

Viene anche effettuato un controllo sulla tensione di alimentazione e se questa cala al disotto dell'80 % del suo valore nominale, viene mostrato il messaggio: "Valim tool low!". Se invece la tensione di alimentazione eccede il 110 % del valore nominale, il display mostra il messaggio di avviso: "Valim too high!".

In tutti i casi, la carica viene immediatamente interrotta. Nella **Tabella 2** riportiamo i tempi indicativi per ricaricare all'80 % batterie di diverse capacità.

Il programma è stato studiato in modo da memorizzare i parametri impostati, cosicché al successivo utilizzo del caricabatterie, si troveranno già le impostazioni fatte in precedenza.

Raccomandiamo sempre di collegare la batteria da caricare, prima di accendere il caricabatterie e di scollegare la batteria dopo aver spento il caricabatterie, in quanto, come ben sapete, il funzionamento dei circuiti switching prevede che ci sia almeno un piccolo carico in uscita per effettuare la corretta regolazione della tensione. ■

Nuovo Multisim 11

Un approccio integrato alla progettazione di circuiti



- Realizza un prototipo virtuale via software prima di raggiungere il laboratorio
- Confronta all'istante i dati simulati con le misure reali
- Ricevi feedback istantanei dagli strumenti virtuali integrati
- Utilizza le funzionalità integrate per la didattica
- Scegli l'opzione con oscilloscopio da 100MS/s
- Sistema da laboratorio compatto, robusto e versatile
- 12 strumenti integrati in un'unica piattaforma

>> Scarica le risorse gratuite su ni.com/academic/ii/eep

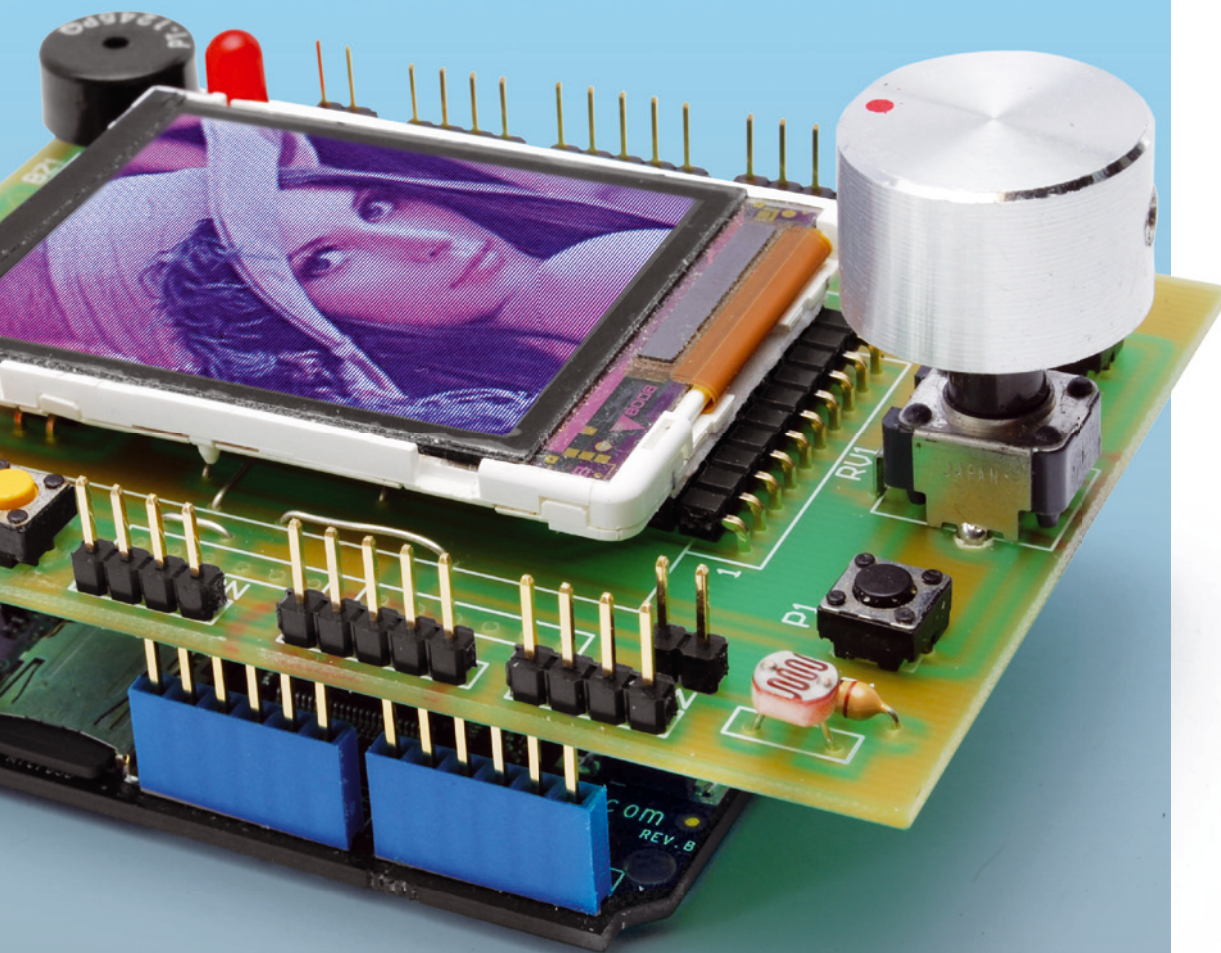
02 41.309.1

IRS Ingegneria
Ricerca
Sistemi

Distributore unico per l'Italia
Tel.: +39 049 8705156 • info@irsweb.it

**NATIONAL
INSTRUMENTS**

NATIONAL INSTRUMENTS 17020 S.W. 27th Ave., Austin, Texas 78741 • Tel. 512 851 2000 • Fax 512 851 2001
E-Mail: info@ni.com • www.ni.com • Italy: Tel. 02 41 309 111 • Fax 02 41 309 112
©2010 National Instruments. Tutti i diritti sono riservati. LabVIEW, Multisim, National Instruments, NI e i suoi nomi sono
registri di National Instruments. Altri prodotti e nomi possono essere marchi commerciali della rispettiva società. 1000



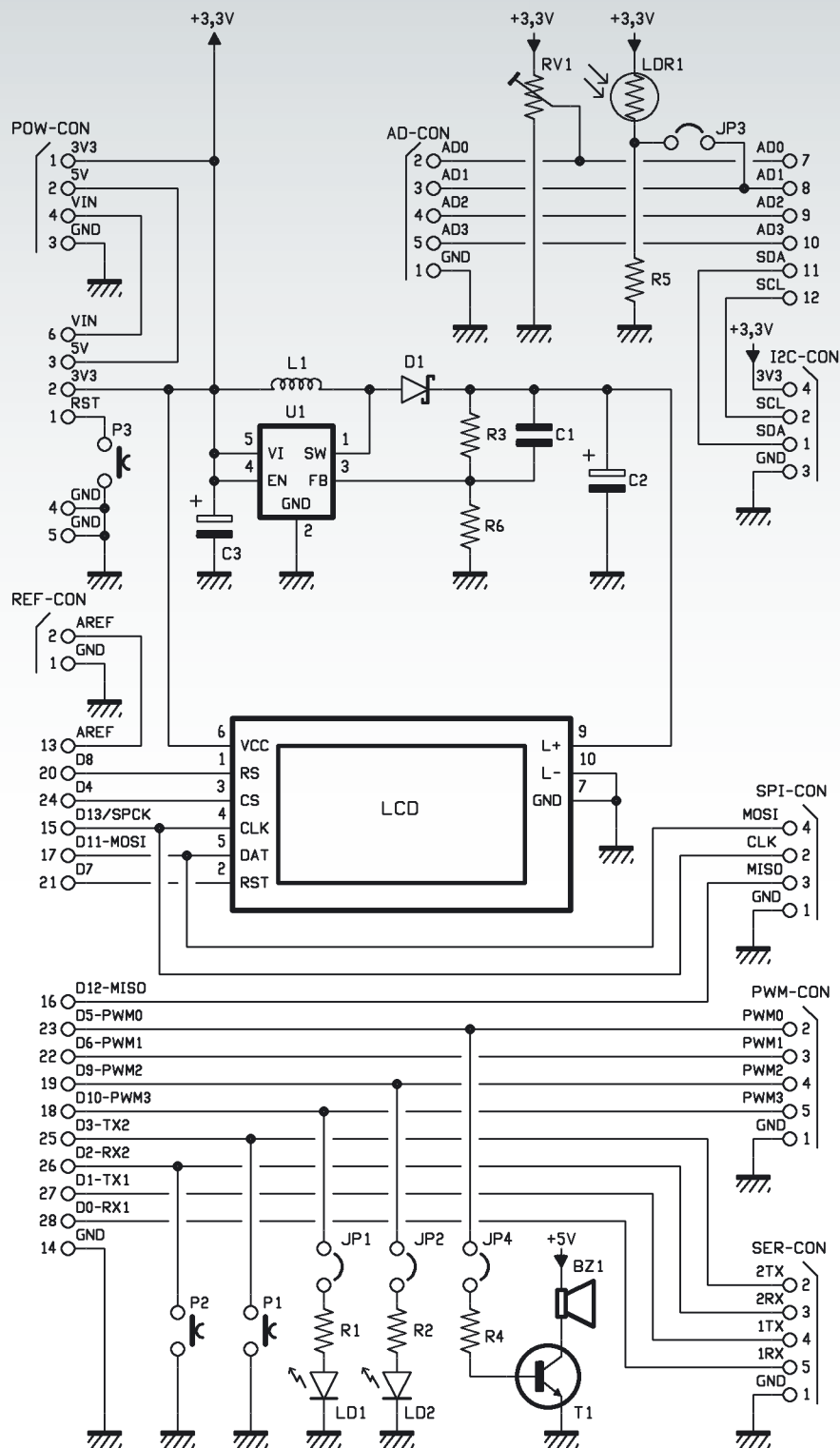
Shield dotato di display a colori e input/output generici per sviluppare su NetduinoPlus, la nuova piattaforma compatibile con Arduino, dotata di interfaccia LAN, supporto SD e basata su un microcontrollore a core ARM che permette di utilizzare la tecnologia .NET Micro Framework.

DEMONETSHIELD

di ANDREA CHIAPPORI

Avete già avuto modo di conoscere Arduino e .Net Micro Framework dai corsi che gli abbiamo dedicato: si tratta di due importanti realtà apparentemente slegate l'una dall'altra, ma in queste pagine vi faremo vedere come effettivamente possono concorrere alla realizzazione di applicazioni molto potenti, con la massima semplicità. Allo scopo vi presenteremo una nuova piattaforma di sviluppo di "casa" Arduino, chiamata

Netduino Plus, ed uno shield che abbiamo appositamente realizzato per essa e che ci servirà a testarne a fondo le periferiche a disposizione, sfruttando sia la versatilità di questo nuovo hardware, sia la potenza di un tool di sviluppo come VisualStudio 2010, scaricabile gratuitamente dal sito Microsoft nella versione Express. Grazie al nostro shield, doteremo Netduino Plus di un display LCD a colori, di un potenziometro, due pulsanti



mini, un cicalino piezoelettrico e di una fotoresistenza, in modo da dare sfogo alla nostra fantasia e alle nostre creazioni. Uno schema logico della soluzione

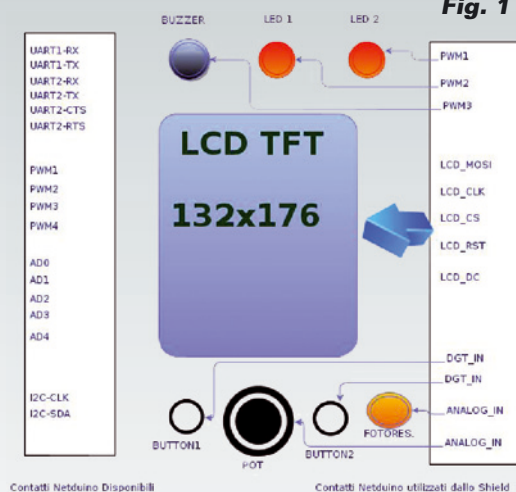
proposta è raffigurato in **Fig. 1**, dove possiamo già farci una prima idea dei pin utilizzati per il nostro shield e di quelli nativi della board Netduino Plus, co-

dunque disponibili all'utente. Prima di addentrarci nel nostro progetto può essere necessaria una breve panoramica su Netduino Plus, una scheda che nasce dall'esperienza dell'ormai consolidato Arduino e che ci permette, sfruttando anche la tecnologia .NET Micro Framework di Microsoft, di scrivere e testare in tempo reale la nostra applicazione direttamente su di essa, fornendoci anche la possibilità di inserire breakpoint (come visibile in **Fig. 2**) nel nostro codice e valutare lo stato delle variabili.

Questo è reso possibile dai nuovi microcontrollori Atmel AT91SAM7X, che integrano un core Arm7 a 32 bit, una memoria Flash da 512 kB e una serie di periferiche hardware, tra cui un'interfaccia ethernet 10/100 Mbps, un connettore per SD-card e la gestione interna del protocollo USB, che lo rendono un oggetto davvero appetibile. A ciò va aggiunto il notevole rapporto qualità prezzo, che non guasta mai, soprattutto considerando che le schede di sviluppo basate su processore Arm possono raggiungere costi veramente elevati.

In particolare, la scheda Netduino Plus utilizza un AT91SAM7X512, tra le cui caratteristiche spicca l'elevata frequenza di clock, di molto superiore a quella dei più usati e moderni microcontrollori a 8 bit. Esistono, in verità, tre versioni di Netduino, ognuna con diverse caratteristiche: Netduino-Mini, Netduino-Classic, e Netduino-Plus. Il nostro shield potrà essere applicato sia a Netduino-Classic sia a Netduino-Plus, ma avendo mantenuto la compatibilità con le precedenti Arduino, altri shield costruiti e pensati per esso potranno essere

Fig. 1



Contatti Netduino Disponibili

Contatti Netduino utilizzati dallo Shield

riutilizzati anche su Netduino Plus; analogamente, il nostro shield potrà essere utilizzato anche su un comune Arduino. In Fig. 3 è possibile vedere come si presenta la scheda Netduino Plus che utilizzeremo per testare il nostro DemoNet shield. Per maggiori dettagli rimandiamo al sito ufficiale di Netduino (<http://netduino.com/>) da dove potete raggiungere la sezione download e in cui potrete scaricare tutto l'occorrente per scrivere la vostra prima applicazione.

L' HARDWARE

Iniziamo col dire che un possessore di Netduino-Plus ha già a disposizione abbastanza elementi per testare il funzionamento dei suoi moduli hardware, perché in meno di 8 cm quadrati di scheda trova un LED, un pulsante, un connettore di rete, un supporto per micro SD in cui memorizzare e leggere dati e, ovviamente, la classica striscia di contatti femmina compatibile con Arduino, da cui possiamo colloquiare con l'esterno tramite protocollo seriale, I²C e SPI.

La programmazione avviene tramite un cavetto USB dal quale il modulo riceve anche l'alimentazione, per cui non servono nemmeno particolari programmatori dedicati. Nonostante tutto questo, dopo aver iniziato a scrivere le prime applicazioni, si inizia a sentire la mancanza di un feedback che diventerà quasi indispensabile:

Il "cuore" di Netduino

Netduino Plus è equipaggiato con un microcontrollore Atmel della serie SAM7X, che è basato su un processore ARM7 (ARM7TDMI) a 32 bit. Presenta al suo interno un set completo di periferiche tra cui spicca una porta Ethernet 10/100 Mbps che lo eleva rispetto ai microcontrollori della serie SAM7S e una buona quantità di spazio in memoria.

Tabella 1 - Caratteristiche di Netduino Plus

AT91SAM7X512	
Moduli	
Funzione	Quantità
Ethernet 10/100	1
USB 2.0 full speed	1
CAN 2.0	1
USART/UART	2
SPI	2
I2C	1
PWM 16 bit	4
Timer 16 bit	4
ADC 10 bit	8
RTT 32 bit	1
WDT 12 bit	1
Memoria	
Internal FLASH	512 KB
Internal SRAM	128 KB
Tensione di alimentazione	
3,3V e 1,8V	
Clock	
48 kHz	

Come tensione di alimentazione si potrebbe applicare solo 3,3 V, perché il modulo integra un regolatore di tensione interno con cui può ricavare gli 1,8 V. Riguardo al pinout, è dotato di cento pin, ma purtroppo il passo piuttosto stretto (0,50 mm) rende difficile l'utilizzo (anche se non impossibile) in campo hobbistico. Risulta così necessario ricorrere a piattaforme già pronte e di basso costo come appunto Netduino o Netduino Plus.

Altra nota interessante di questa famiglia di microcontrollori è la presenza di un bootloader integrato in una sezione riservata della Flash, che permette il caricamento del firmware tramite protocollo USB, utilizzando un tool ad interfaccia grafica dedicato e fornito direttamente dall'Atmel; si chiama SAM-BA (SAM - Boot Assistance) e può essere scaricato liberamente dal sito.

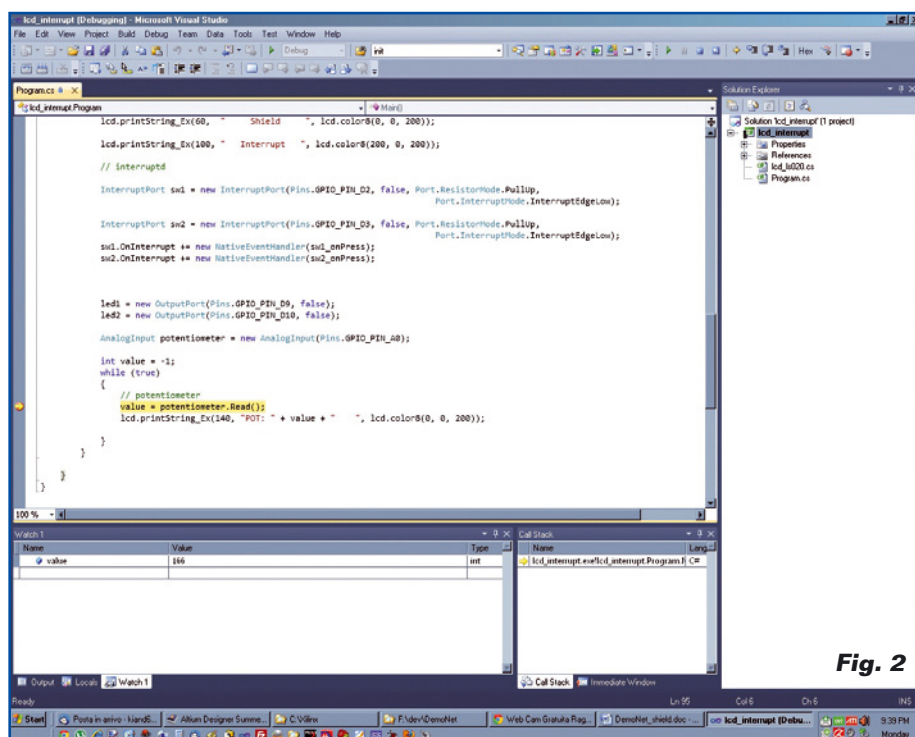


Fig. 2

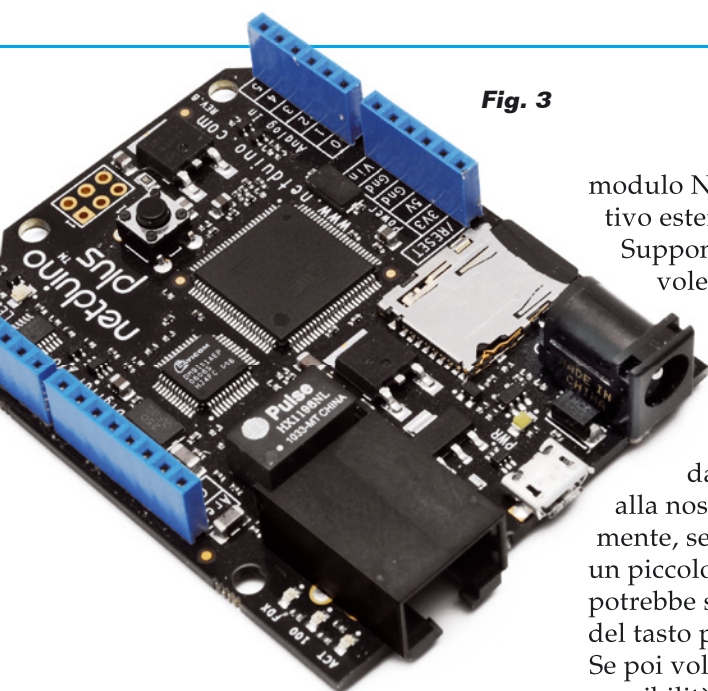


Fig. 3

modulo Netduino o un dispositivo esterno.

Supponiamo, ad esempio, di voler realizzare un piccolo server web; potrà allora essere necessario avere l'informazione a schermo del numero IP assegnato dal DHCP del router alla nostra board. Analogamente, se volessimo realizzare un piccolo lettore di file di testo, potrebbe servirci qualcosa più del tasto presente sulla scheda. Se poi volessimo aumentare le possibilità di segnalazione della nostra piattaforma, potrebbero servirci un paio di LED, oltre a quello blu già a disposizione, e magari un avvisatore acustico col quale potremmo anche generare una semplice melodia. È in questo caso che ci viene in aiuto il nostro DemoNet shield,

un display LCD su cui visualizzare le nostre prime scritte di test, piuttosto che informazioni utili al fine dell'utilizzo, o semplicemente messaggi di debug qualora il nostro scopo sia quello di testare un nuovo

che può rendere Netduino una piattaforma ancora più completa.

Lo schema elettrico dello shield si commenta da solo: la maggior parte dell'elettronica è dedicata alla generazione dei 10 V che servono per attivare la retroilluminazione interna del display LCD; una piccola parte è per poter controllare il cicalino piezoelettrico tramite un classico transistor BC547 (T1) così come il partitore per la fotoresistenza LDR1 utilizzata per testare l'ingresso analogico (nel nostro caso AD1). La manopolina è collegata ad un potenziometro lineare, il cui pin è collegato ad un altro ingresso analogico di Netduino, cioè AD0.

A questa dotazione è stata poi aggiunta una serie di connettori in modo da rendere sempre disponibili le funzionalità associate ai pin di Netduino.

Scendendo nei dettagli, il display utilizzato è un piccolo LCD TFT di casa SHARP, montato nei telefonini Siemens C65 e già impiegato nel Digital Frame (numero 153 di Elettronica In); ci limiteremo, pertanto, ad una descrizione delle funzioni della classe *display_ls020*, che ci per-

Tabella 2

Libreria DISPLAY_LS020	
Funzioni	Descrizione
init()	inizializza display
println(line, str, color)	scrive una stringa <i>str</i> di colore <i>color</i> sulla linea <i>line</i>
clearScreen()	colora di bianco tutto lo schermo
fillBox(x1, y1, x2, y2, color)	colora di colore <i>color</i> l'area indicata dalle coordinate sullo schermo
drawImage(pBuffer, width,height,posx, posy)	disegna l'immagine contenuta nel buffer nell'area impostata
progressBar(value, color)	disegna una progressBar di lunghezza <i>value</i> e di colore <i>color</i>
prepareImage(width,height,posx,posy)	imposta l'area dello schermo in cui verrà caricata una immagine
sendImageBuffer(bufImage)	disegna effettivamente l'immagine contenuta nel buffer
color16(R, G, B)	converte da formato R,G,B a 16bit
color8(R, G, B)	converte da formato R,G,B a 8bit
reverse()	effettua un reverse dei colori su tutto lo schermo
mode8bit()	imposta modalità 8 bit
mode16bit()	imposta modalità 16 bit
mode24bit()	imposta modalità 24 bit
drawLine()	disegna una linea
drawCircle()	disegna un cerchio
Variabili	
LcdFont8[]	buffer contenente i font 8x8



Fig. 4

mette di prenderne il controllo. Tali funzioni sono descritte nella **Tabella 2**.

Saremo così in grado di scrivere le nostre stringhe sul display e anche di visualizzare un'immagine di 132x176 pixel, memorizzata in precedenza in una micro SD-card. Tutto questo senza aumentare troppo le dimensioni della compatta board Netduino e rendendo disponibili su strisce di contatti maschi i vari pin già presenti su Netduino ed organizzati in connettori separati, come visibile nella **Fig 4**.

I due pulsantini sono collegati a due ingressi digitali, mentre i LED abbiamo scelto di collegarli ai pin che offrono anche un'uscita PWM, in modo da poter variare facilmente anche la loro luminosità. A tal proposito è d'obbligo una precisazione in merito ai due jumper (JP1 e JP2) posti in serie a queste uscite: nel caso in cui si volessero utilizzare i moduli PWM per pilotare altri dispositivi, come ad esempio servomotori, o in altri casi in cui un collegamento a massa tramite il LED (e rispettiva resistenza) potrebbe causare conflitti, sarà possibile collegare i diodi luminosi e continuare a utilizzare i pin del PWM dal connettore PWM-CON disponibile nel nostro shield.

A questo punto diventa indispensabile scendere più in dettaglio e mettere in correlazione le funzionalità dei pin di Netduino e la loro mappatura sullo Shield DemoNet; ci viene in aiuto la **Tabella 3**, in cui vediamo la corrispondenza tra i pin di Netduino e il loro utilizzo nella nostra scheda.

Bisogna poi considerare che i pin indicati come input analogici sono quelli collegati internamente ai pin del microcontrollore Atmel e che hanno

Tabella 3

Power Pin		
Pin	Funzione	Shield
/RESET	RESET	RST
3V3	POWER	-
5V	POWER	-
GND	GND	-
GND	GND	-
Vin	POWER	-
Analog Input		
Pin	Funzione	Shield
A0	AN - INPUT	CON4 - ADO (POT)
A1	AN - INPUT	CON4 - AD1 (FOTORESISTENZA)
A2	AN - INPUT	CON4 - AD2
A3	AN - INPUT	CON4 - AD3
A4	AN - INPUT	CON5 - SDA
A5	AN - INPUT	CON5 - SCL
Digital I/O		
Pin	Funzione	Shield
0	DIGITAL I/O UART1 - RX	CON1-RX1
1	DIGITAL I/O UART1 - TX	CON1-TX1
2	DIGITAL I/O UART2 - RX	CON1-RX2 BUTTON1
3	DIGITAL I/O UART2 - TX	CON1-TX2 BUTTON2
4	DIGITAL I/O	LCD-CS
5	DIGITAL I/O PWM	CON2-PWM0
6	DIGITAL I/O PWM	CON2-PWM1
7	DIGITAL I/O UART2 - RTS	LCD-RST
8	DIGITAL I/O UART2 - CTS	LCD-DC
9	DIGITAL I/O PWM	CON2 - PWM2 LED1
10	DIGITAL I/O PWM	CON2 - PWM3 LED2
11	DIGITAL I/O SPI - MOSI	CON3 - MOSI LCD - MOSI
12	DIGITAL I/O SPI - MISO	CON3 - MISO
13	DIGITAL I/O SPI - CLK	CON3 - CLK LCD - CLK
14	GND	CON3 - GND
15	AREF	CON4 - AREF

la possibilità di funzionare in modalità analogica (ingressi del modulo ADC), ma nulla ci vieta di usarli anche come semplici I/O digitali.

IL SOFTWARE

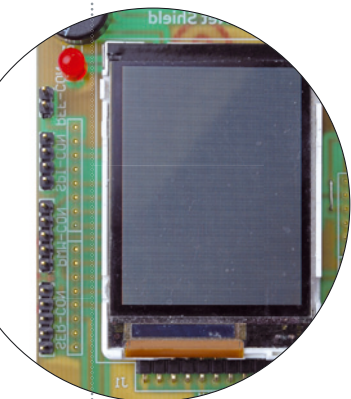
Analizzato l'hardware possiamo iniziare a scrivere le nostre prime applicazioni che utilizzeranno lo shield appena costruito, iniziando a sfruttare i vantaggi della tecnologia .NET Micro Framework.

La presentazione di questo shield può facilmente diventare un pretesto per esaminare più in profondità le possibilità che ci vengono offerte dalla piattaforma Netduino.

Descriveremo, pertanto, alcune applicazioni che utilizzano il display, il potenziometro e gli altri I/O presenti nello shield, ma anche l'interfaccia ethernet, la lettura scrittura su card SD e il modulo PWM.

Per rendere l'inizio più semplice, abbiamo creato tanti semplici progetti in VisualStudio, scaricabili dal sito della nostra rivista, ognuno dei quali si riferisce ad un particolare controllo. Non vogliono essere programmi a sé stanti, ma piuttosto una base di partenza e di spunto per applicazioni future e per prendere familiarità con questo nuovo shield, insomma piccoli esempi che possono essere utilizzati sia

[piano di MONTAGGIO]

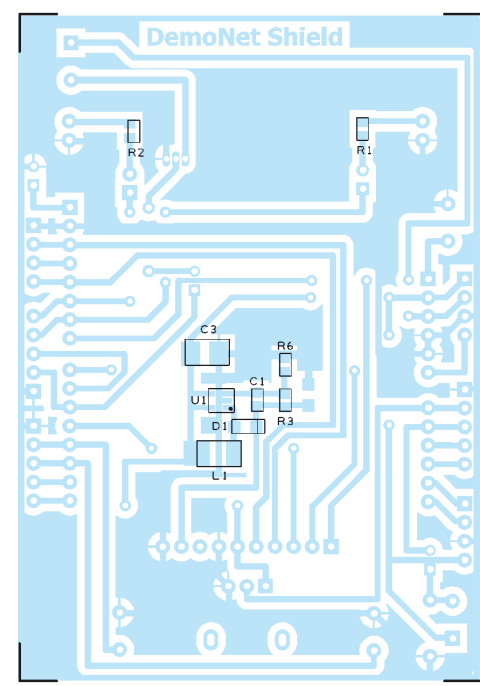
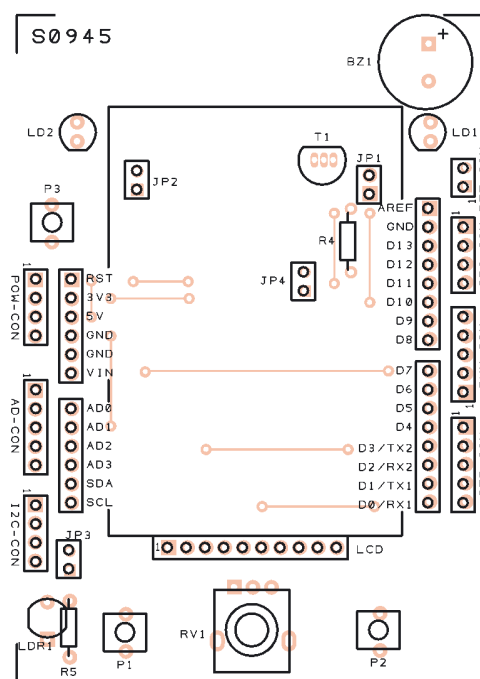
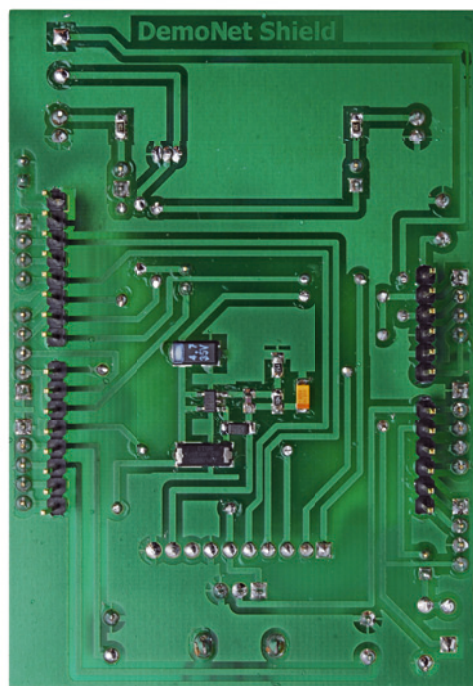
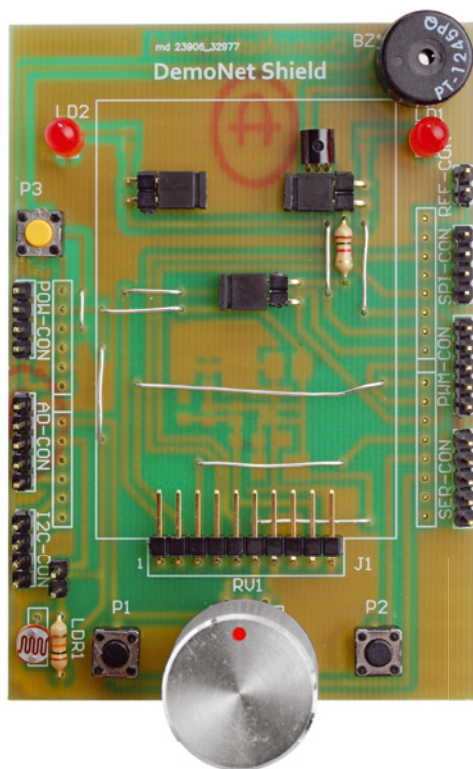


Elenco Componenti:

R1, R2: 330 ohm (0805)
 R3: 100 kohm (0805)
 R4: 1 kohm
 R5: 10 kohm
 R6: 15 kohm (0805)
 RV1: Potenzimetro lineare 10 kohm
 C1: 100 pF ceramico (0805)
 C2: 1 μ F ceramico (0805)
 C3: 4,7 μ F 6,3 VL tantalio
 LD1: LED 5 mm rosso
 LD2: LED 5 mm rosso
 P1÷P3: Microswitch
 BZ1: Cicalino piezoelettrico
 LDR1: Fotoresistenza
 T1: BC547
 D1: MBRS140TRPBF
 L1: Induttanza 10 μ H
 U1: TPS61040
 LCD: LS020

Varie:

- Strip maschio 5 vie (4 pz.)
- Strip maschio 4 vie (3 pz.)
- Strip maschio 10 vie
- Strip maschio 6 vie
- Strip maschio 2 vie
- Strip maschio 16 vie
- Strip femmina 10 vie
- Jumper (2 pz.)
- Circuito stampato



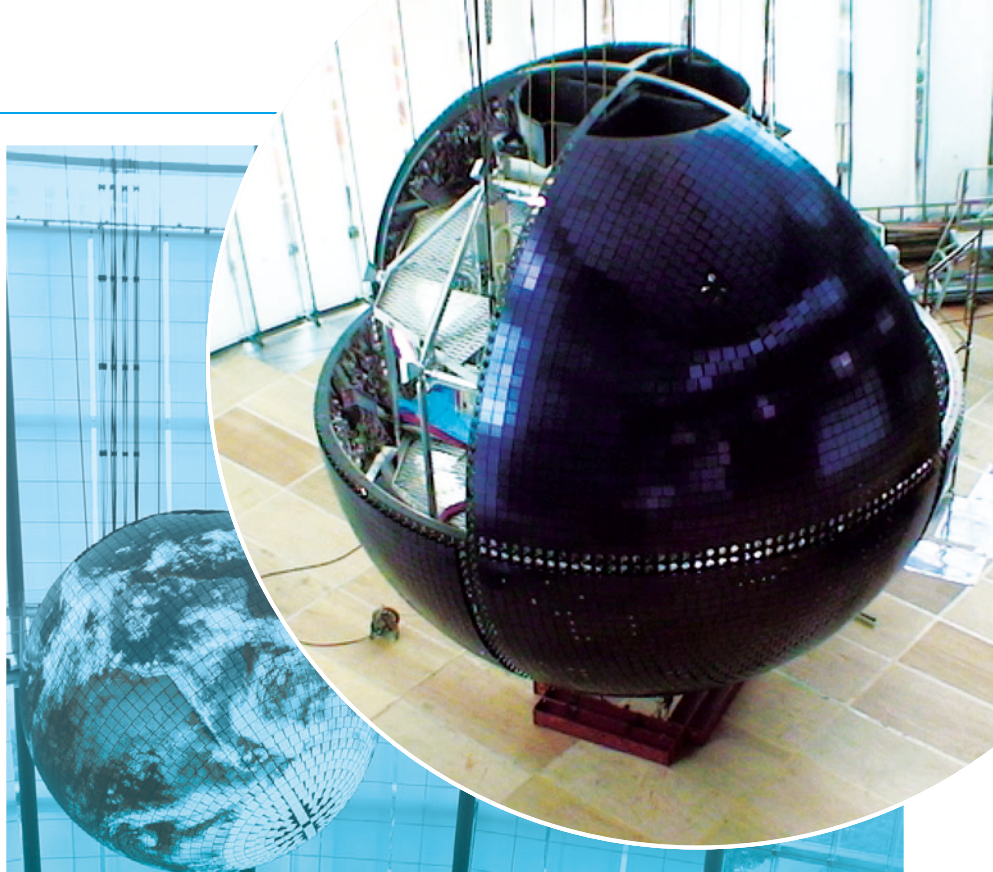
come test funzionali della nostra board sia come base di partenza per una applicazione più complessa ed operativa. Questi sono classificati in base

al loro contenuto nella **Tabella 4**. Grazie all'utilizzo di un potente IDE come Visual Studio e al linguaggio C#, non dobbiamo più

preoccuparci di quali e come settare i vari registri interni del microcontrollore Atmel, magari sfogliando e risfogliando il data-sheet, perchè ora tutto viene

gestito a più alto livello direttamente dal framework Microsoft. In questo modo, se volessimo impostare un determinato valore o regolare l'intensità di un LED o la velocità di un servo motore potremmo, con poche righe di codice, leggere lo stato del potenziometro, scrivere la lettura su display ed impostare di conseguenza il modulo PWM attivo su una delle nostre uscite a disposizione a cui sono collegati i LED ed il cicalino BZ1 per generare toni audio e magari attivare qualche uscita in base allo stato di luminosità esterna, controllato dalla fotoresistenza LDR1.

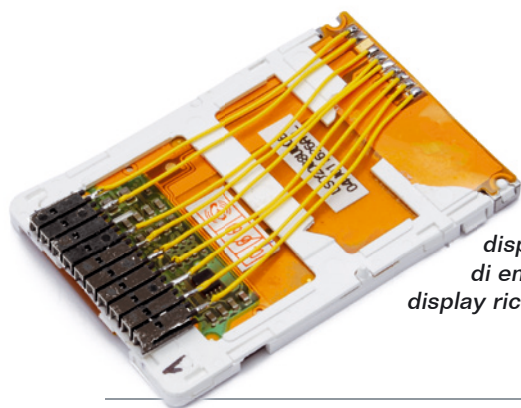
In particolare nel **Listato 1** potete notare come la lettura del potenziometro e il settaggio del PWM avvengano in singole righe di codice, rendendolo più chiaro e leggibile. L'unica accortezza sta nel configurare i vari moduli (come PWM e AnalogInput) in modo che siano associati al pin corretto, come riassunto in **Tabella 3**. È possibile anche utilizzare la funzione *Debug.Print()* offerta da .NET MF per ottenere informazioni che ci potrebbero servire e che verranno visualizzate a run-time nella output-console di Visual Studio. L'inizializzazione del modulo SPI per pilotare il display, inserita all'interno della classe *DisplayLs020* risulta un poco più



La sfera OLED ci mostra il mondo

È della Mitsubishi il più grande display OLED al mondo: Geo Cosmos, così si chiama, oltre che essere decisamente voluminoso è a forma di sfera ed ha un diametro di ben sei metri. Esposto al Museo Nazionale della Scienza e dell'Innovazione di Tokyo, è appeso a 18 metri di altezza dal suolo. Il globo, che rappresenta la Terra, è una sfera di alluminio ricoperta da 10.362 pannelli OLED, ciascuno dei quali misura 96x96 millimetri. Il grande display è stato realizzato in occasione del 10° anniversario del Museo, per sostituire il precedente grande schermo a LED; grazie alla sua eccellente risoluzione di oltre 10 milioni di pixel, mostrerà le immagini della Terra (come ad esempio lo spostamento delle nubi) scattate da un satellite meteorologico.

Questo avveniristico visualizzatore dimostra le ampie prospettive della tecnologia Mitsubishi Electric Diamond Vision OLED scalabile, che permette di realizzare schermi di tutte le dimensioni. Il sistema OLED, grazie alla struttura e composizione dei diodi organici, permette di comporre qualsiasi forma e realizzare pixel di ogni dimensione. Con una luminosità che può arrivare fino a 1.200 candele/m² e un contrasto due volte superiore rispetto ai valori tradizionali, il display Mitsubishi è ideale per le installazioni in aree molto luminose.



Per montare il display nello shield bisogna collegare il connettore di cui dispone ad un pin-strip femmina mediante spezzoni di filo da saldare sui contatti di entrambi, punto a punto. Nell'effettuare le saldature sui pad del connettore del display ricordate di esercitare poca pressione e tenere la punta del saldatore (che deve essere da non più di 20 W) lo stretto necessario a fondere lo stagno.

complessa, ma questo è dovuto essenzialmente alle varie configurazioni possibili legate al protocollo. Ad esempio, per inizializzare il display dovremo per prima cosa attivare il modulo SPI del microcontrollore e per questo occorrerà creare una configurazione con la linea di codice seguente:

```
SPI.Configuration lcdSpiConfig =
new SPI.Configuration(
    Pins.GPIO_NONE, false, 0, 0,
    false, false, 40000, SPI.SPI_module.
    SPI1);
```

Successivamente, dovete creare un nuovo oggetto della classe .NET SPI, utilizzando la seguente linea di codice:

```
lcdSpi = new SPI(lcdSpiConfig);
```

Fatto ciò, possiamo utilizzarlo direttamente dopo aver preventivamente caricato il buffer *spiBuffer* con i dati necessari, come nella riga seguente:

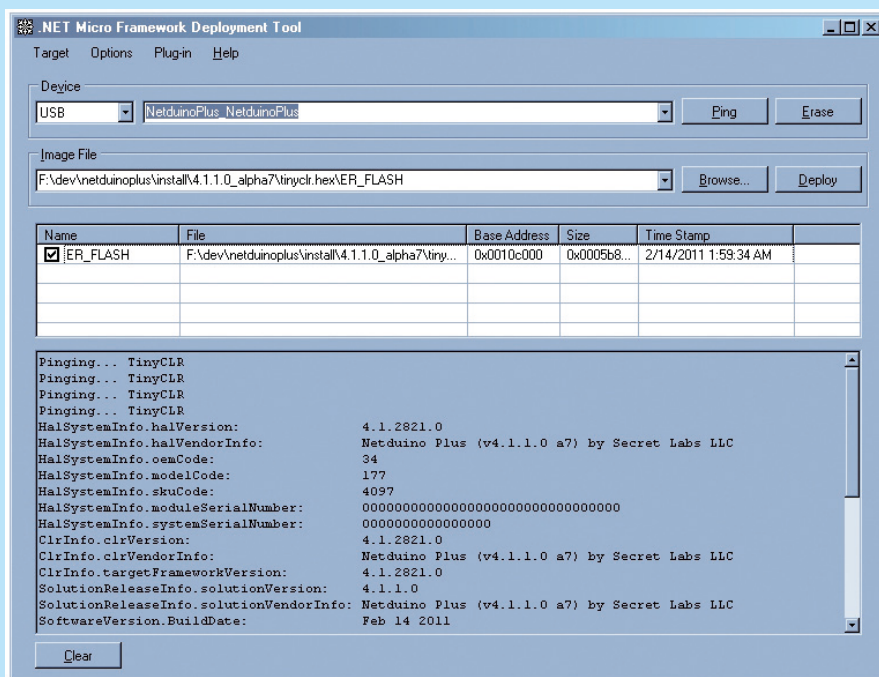
```
lcdSpi.Write(spiBuffer);
```

In particolare, le funzioni che

gestiscono il display sono racchiuse nella classe *DisplayLs020* contenuta nel file *lcd_Is020.cs* e sono visualizzate anche nella **Tabella 2**. Non ci soffermeremo sui dettagli, ma è interessante notare come la riga di codice

usata per inizializzare il protocollo SPI sia la stessa usata nel corso di NET Micro Framework del fascicolo numero 152 di questa rivista, anche se in quel caso si trattava di una scheda differente.

È questo un altro grande vantaggio di .NET che ci permette di utilizzare lo stesso codice su schede diverse e fornite con microcontrollori diversi, senza perdere tempo e fare sforzi inutili. Purtroppo, però, tutto ha un prezzo e in questo caso lo paghiamo nel fatto di dover dotare il microprocessore, (nel nostro caso AT91SAM7) del firmware necessario per tradurre le funzioni di alto livello che scriviamo da Visual Studio, in istruzioni native per il particolare core utilizzato. Oltre a limitare lo spazio per il nostro codice, ciò può rappresentare un over-head in termini di velocità; e infatti, rimanendo sul



Usando .NET MF e programmando in C# ad alto livello, la gestione di livello più basso è delegata ad un firmware particolare scritto specificatamente per il microcontrollore Atmel e per la piattaforma hardware sul quale andrà montato. Questo firmware è già

presente al momento dell'acquisto ed è il responsabile di un lavoro "oscuro" che nasconde al programmatore tutte le procedure per scrivere e leggere dai registri interni del microcontrollore al fine di permettere il corretto funzionamento delle istruzioni impartite

Tabella 4

Progetti di esempio per Netduino		
Directory	Descrizione	Moduli usati
Lcd	prime stringhe sul display	LCD
Lcd_IO	iterazione con I/O disponibili	LCD, Potenzziometro, fotoresistenza, pulsanti
Lcd_Interrupt	utilizzo interrupt sui pulsanti P1 e P2	LCD, Interrupt, pulsanti
Lcd_pwm	regolazione luminosità tramite PWM	LCD, Pwm, led
Lcd_Menu	semplice menu su display	LCD, potenziometro
Lcd_Progress	progressBar su display	LCD, potenziometro
Lcd_fat	scansione directory SD	LCD, SD
Lcd_readwrite_sd	lettura e scrittura di file di testo su SD	LCD, SD
Lcd_image	visualizzazione di una immagine in Flash	LCD, SD
Lcd_image_sd	visualizzazione di immagini da SD	LCD, SD
Lcd_timer	lampeggio led utilizzando un timer impostabile	LCD, timer
Lcd_web	piccolo webs-server	LCD, Ethernet, SD

Firmware Netduino e aggiornamento

in linguaggio C# dal Visual Studio. Tuttavia è soggetto a continui aggiornamenti che nascono sia dallo sforzo che la comunità Netduino fa per risolvere bug esistenti e aggiungere nuove caratteristiche, sia dalla necessità di mantenere compatibilità con gli aggiornamenti delle librerie di Microsoft per il .NET Micro Framework. Diventa, perciò, necessario aggiornare il proprio firmware periodicamente. Ma niente paura, perchè con Netduino è possibile farlo semplicemente tramite USB e grazie ad un tool fornito da Microsoft (o dal sito di Netduino) gratuitamente, chiamato MFDeploy.

Basterà scaricare il firmware in una directory del vostro PC, trascinare i file nominati ER_FLASH e ER_CONFIG nella lista di MFDeploy come illustrato qui a sinistra, e premere il pulsante Deploy.

Tutto questo ci offre anche un altro vantaggio: quello di poter decidere se riservare o meno spazio in memoria di programma e in RAM per gestire periferiche (come, ad esempio, la rete, la seriale, o la SD-

Card) quando non ci serve utilizzarle. Essendo Netduino un progetto open-source, vengono forniti, dal sito ufficiale, tutti i sorgenti (in codice C) così che possiamo riscrivere un nostro firmware personalizzato, invece di usare sempre quello predefinito, che consuma risorse per gestire sempre tutte le periferiche. Nella tabella qui sotto viene evidenziata la quantità di spazio a disposizione in cui scrivere il codice, in base al firmware caricato su Netduino. Normalmente Netduino Plus viene fornito con un firmware che contiene codice open-source per supportare la rete, si tratta di uno stack TCP/IP chiamato lwIP (light-weight IP) che è pensato appositamente per questo tipo di applicazioni embedded, ma il cui codice è comunque oneroso in termini di

spazio. Per i più esperti c'è anche la possibilità di cancellare completamente il firmware con cui viene fornito Netduino (seguendo una particolare procedura ben descritta sul forum del sito di Netduino) per utilizzare la scheda come un vero e proprio tool di sviluppo Arm a basso costo.

A questo punto dovremo compilare le nostre applicazioni scritte in linguaggio C, utilizzando una tool-chain per Arm di terze parti, e scaricarle sul microcontrollore tramite l'accoppiata SAM-BA e cavetto USB o direttamente tramite un programmatore JTAG. Questo ci costringe ad una programmazione a basso livello, ma nello stesso tempo ci permette di avere a disposizione gli interi 512 kB di memoria di programma e i 128 kB di RAM.

Memoria (Flash e RAM) a disposizione

Firmware	Internal Flash	Internal SRAM
.NET Originale	64 kB	28 kB
USB 2.0 full speed	128 kB	60 kB
Nessun Firmware	512 kB	128 kB

display (anche se possibile) non è conveniente disegnare immagini o font utilizzando funzioni di classe che scrivono un singolo pixel alla volta, ma è meglio caricare tutto su un buffer in memoria e successivamente inviarlo al display tramite la funzione sendImageBuffer(), altrimenti noteremo fastidiosi rallentamenti nella visualizzazione dell'immagine perchè, pur avendo un'elevata frequenza di clock (48 kHz), le continue chiamate delle funzioni .NET verso quelle native rappresentano un collo di bottiglia, anche aumentando la velocità del modulo SPI in fase di inizializzazione.

A queste due limitazioni c'è, tuttavia, un rimedio, come spiegato più in dettaglio nel riquadro in alto in questa pagina, relativo all'aggiornamento del firmware. Prima di procedere ed a proposito di aggiornamento, ancora una nota importante sulla

gestione del protocollo SPI: nel caso la versione del vostro Netduino fosse precedente alla 4.1.1.0, occorre un aggiornamento del firmware, perchè nelle versioni precedenti l'inizializzazione del protocollo SPI (usato

per comunicare col display) presentava alcuni bug che ne impedivano il corretto funzionamento.

Detto ciò, continuiamo la presentazione del software soffermandoci sulla lettura/scrittura

Listato 1

```
public static void Main()
{
    lcd = new DisplayLs020();
    lcd.init();
    lcd.clearScreen();
    lcd.printString(20, "    DemoNet    ", lcd.color8(200, 0, 0));

    PWM pwmLed1 = new PWM(Pins.GPIO_PIN_D9);
    AnalogInput potentiometer = new AnalogInput(Pins.GPIO_PIN_A0);

    Debug.Print("Start Program\n");

    uint value = -1;
    while (true)
    {
        //potentiometer
        value = potentiometer.Read();
        lcd.printString_Ex(140, "POT: " + value + "    ", lcd.color8(0, 0, 200));

        // pwm
        pwmLed1.SetPulse(1024, (uint)value);
    }
}
```

Listato 2

```
// write on sd
FileStream stream = new FileStream(@"\SD\info.txt", FileMode.OpenOrCreate);
byte[] buff = Encoding.UTF8.GetBytes("ciao mondo from Netduino");
stream.Write(buff, 0, buff.Length);
stream.Close();

// read from sd
FileStream sr = new FileStream(@"\SD\info.txt", FileMode.Open);
byte[] buff = new byte[10];
sr.Read(buff, 0, 4);
Debug.Print(buff);
sr.Close();
```

visualizzato su display per dirci dove potremo visualizzare la nostra pagina.

Per un'analisi più approfondita di questo codice rimandiamo al listato presente nella directory del sito di Elettronica In, mentre possiamo rivolgere più attenzione alla facilità con cui risulta possibile con Netduino e .NET, attivare un timer, che ad ogni intervallo prestabilito chiami una determinata funzione e la esegua lasciando inalterato il normale flusso delle operazioni scritte nel main. Questo è visualizzabile nel **Listato 3**: il timer `timerWatchDog` viene inizializzato nella funzione `initTimer()` in modo che ogni 1000 ms venga chiamata la funzione `WatchDog()`. Questa, nel nostro caso, cambia lo stato di un LED e lo fa mentre il codice scritto nel main può continuare la propria esecuzione. Inutile aggiungere, a questo punto, come diventi possibile grazie a Netduino e a .NET Microframework, utilizzare con la stessa semplicità anche la gestione dei thread, così come associare un interrupt software ai pulsanti del nostro shield.

CONCLUSIONI

Con questo articolo abbiamo cercato di fornire una prima base per prendere il controllo di queste nuove piattaforme hardware basate su ARM e su .NET MF, che stanno trovando molti consensi tra hobbisti ed esperti. In particolare, sul forum di Netduino troverete numerosi aggiornamenti su come sfruttare ogni particolarità di questa scheda in continua evoluzione. Noi ci fermiamo qua, anche se le possibilità e gli sviluppi futuri sono innumerevoli; a voi continuare con le vostre applicazioni con Netduino. ■

Listato 3

```
private Timer timerWatchDog;

public void WatchDog(Object stateInfo)
{
    ledState = 1 - ledState;
    if (ledState == 1) {
        led.Write(true);
    } else {
        led.Write(false);
    }

    timerCounter++;
}

public void initTimer()
{
    TimerCallback watchDogDelegate = new TimerCallback(this.WatchDog);
    timerWatchDog = new Timer(watchDogDelegate, null, 100, 1000);
}

public static void Main()
{
    lcd = new DisplayLs020();
    lcd.init();
    lcd.clearScreen();
    lcd.printString_Ex(80, "    Timer    ", lcd.color8(0, 100, 200));
    Program thisProg = new Program();

    thisProg.initTimer();

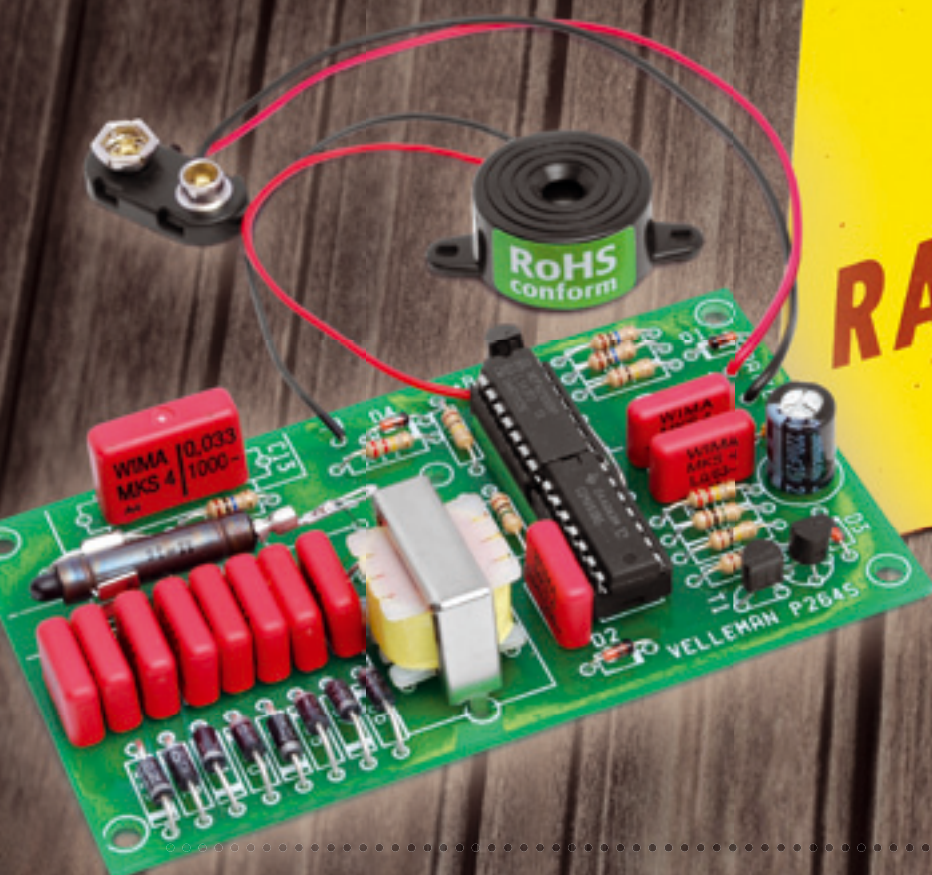
    while(true)
    {
        // do something
    }
}
```

di una card SD, che è permessa grazie all'utilizzo di un modulo SPI interno al microcontrollore ed indipendente da quello usato per il display; in questo modo non dovremmo preoccuparci di nulla, se non di aprire lo stream del file come faremmo in un normale programma C#. Un piccolo spezzone di codice con semplice esempio è visualizzabile nel **Listato 2**. A questo punto possiamo anche collegare Netduino ad un router e provare le funzionalità di rete che

vengono offerte dal .NET Micro Framework.

Tramite la classe `MyHttpHandler` potremo scrivere la nostra pagina HTML (a questo punto anche leggerla da un file memorizzato sulla card SD) e, con il metodo `Start()` della classe `HttpServer`, renderla disponibile sul web. Supponendo di aver configurato il nostro netduino con DHCP, possiamo utilizzare la stessa classe `httpServer` per farci restituire l'indirizzo IP con cui si è registrato e che verrà

CONTATORE GEIGER PALMARE

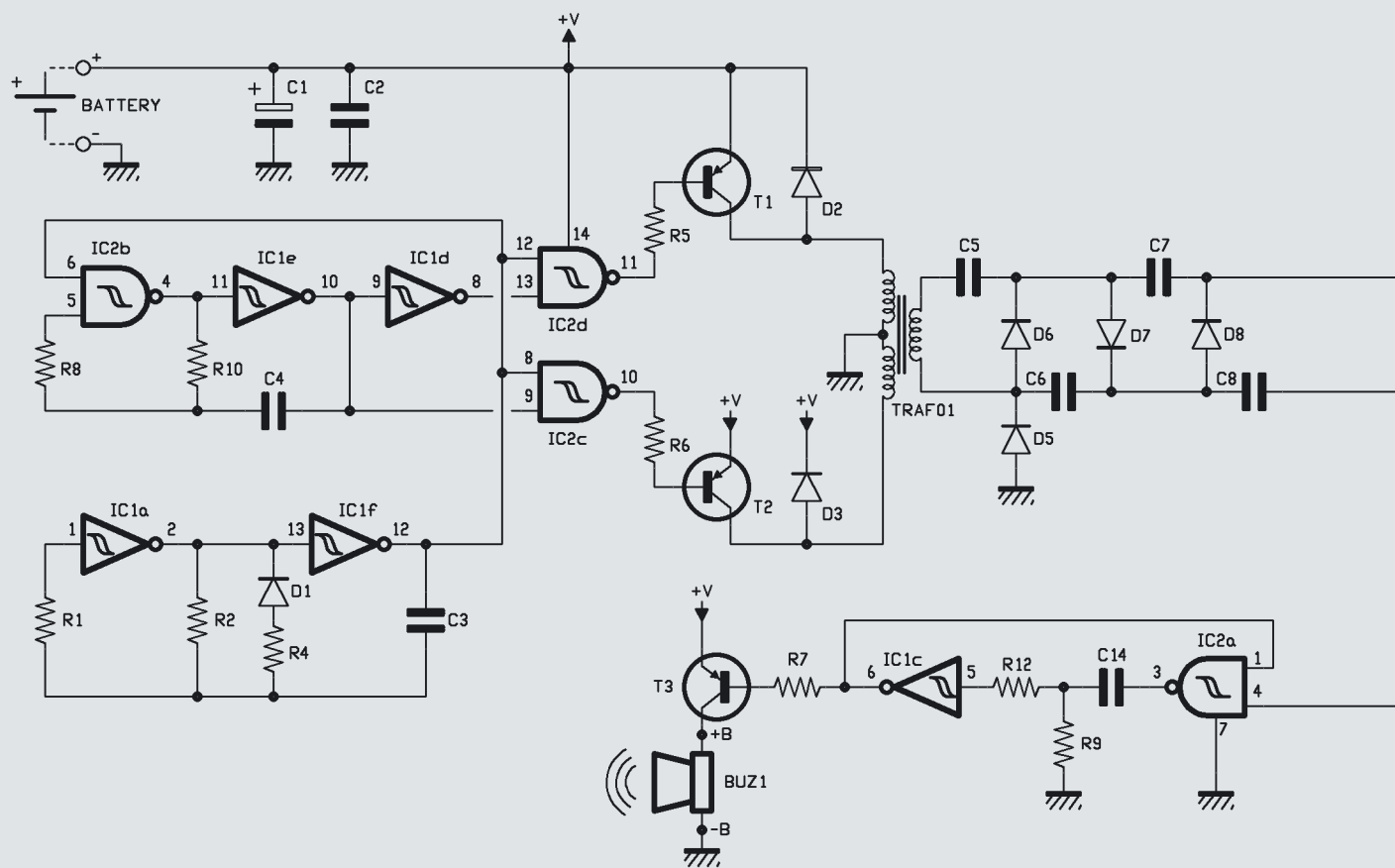


Rilevatore di radioattività utilizzabile sia per valutare la contaminazione di merci ed alimenti, sia per rilevare le radiazioni nell'ambiente.

di DAVIDE SCULLINO

Gli eventi del Giappone hanno bruscamente ridestato l'attenzione del grande pubblico verso uno strumento che sembrava ormai nell'oblio o confinato nei film d'azione: il contatore Geiger. Purtroppo questa riscoperta di un apparato tanto interessante dal punto di vista fisico ed elettronico coincide con eventi infausti e con la crescente paura che anche qui in occidente abbiamo di venire in contatto con materiali, apparecchiature, manufatti e cibi contaminati provenienti dal Giappone e dalle nazioni limitrofe. Contatori Geiger

ce n'è di vario genere, da quelli trasportabili a tracolla ai palmari, tutti comunque accomunati da due caratteristiche: costano tanto e sono difficilmente reperibili nei momenti in cui si verifica qualche incidente nucleare. Per questa ragione abbiamo pensato di proporvi, dopo l'articolo del mese scorso in cui descrivevamo l'impiego di un contatore reperibile nel surplus militare (ad esempio nelle mostre-mercato di elettronica e in quelle di articoli militari) in abbinamento ad un programma per Personal Computer, un progetto



pratico, facilmente realizzabile, di contatore Geiger palmare; si tratta di un dispositivo reperibile anche in kit di montaggio, il cui costo è nettamente inferiore alla metà del più economico Geiger reperibile in commercio in Italia o all'estero.

Il nostro contatore è di tipo tradizionale, basato su un moderno tubo Geiger-Muller miniatura e permette di segnalare all'operatore la radioattività mediante il solito sistema dei suoni, ovvero emettendo il classico "toc" che aumenta di frequenza al crescere dell'entità delle radiazioni; come quello descritto nel fascicolo del mese scorso (il famoso SV-500), dispone dunque di un'uscita audio collegabile ad un cicalino o ad una cuffia, ma anche alla scheda audio di un computer. Questo significa che potete tranquillamente

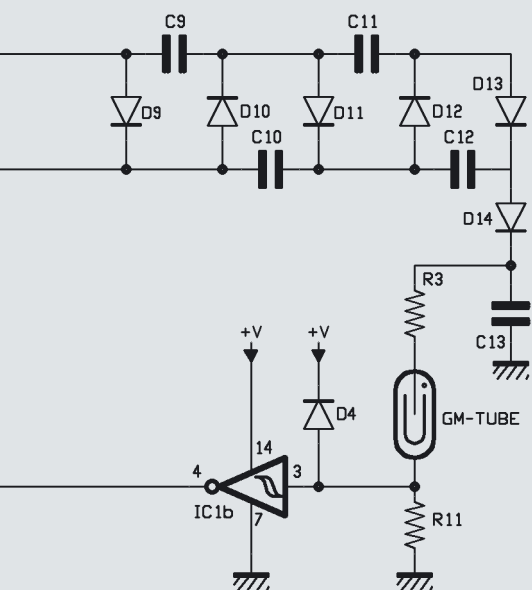
abbinarlo ad un notebook nel quale sia installato il software CDV Counter (scaricabile da diversi siti Internet) e realizzare misure più precise basate su lunghi intervalli di osservazione. Per quanto riguarda il tipo di misurazione effettuabile, diciamo che rileva bene le radiazioni alfa e beta e gamma.

SCHEMA ELETTRICO

Vediamo subito in cosa consiste il nostro contatore, dando uno sguardo allo schema elettrico che lo descrive: il circuito si alimenta a batterie (tipicamente con una pila da 9 volt o una batteria da 12 V) è composto da un elevatore di tensione necessario a polarizzare il tubo Geiger-Muller, dal tubo, da un resistore di carico e da un rivelatore degli impulsi, oltre che da un monostabile ed uno stadio

amplificatore di potenza che pilota un cicalino.

L'elevatore di tensione è composto a sua volta da un inverter CC/CA seguito da una catena di duplicatori di tensione; l'inverter consta del trasformatore TRAF01, il cui primario a presa centrale (la cui presa è collegata al negativo della linea di alimentazione in arrivo dalla batteria) viene pilotato da due transistor PNP, ciascuno dei quali è polarizzato in base da una forma d'onda rettangolare prodotta da un doppio generatore di forma d'onda quadra ed è protetto sul collettore da un diodo che taglia le tensioni inverse prodotte dagli avvolgimenti del trasformatore quando vengono privati dell'alimentazione. L'inverter funziona in push-pull, nel senso che quando un transistor è in saturazione



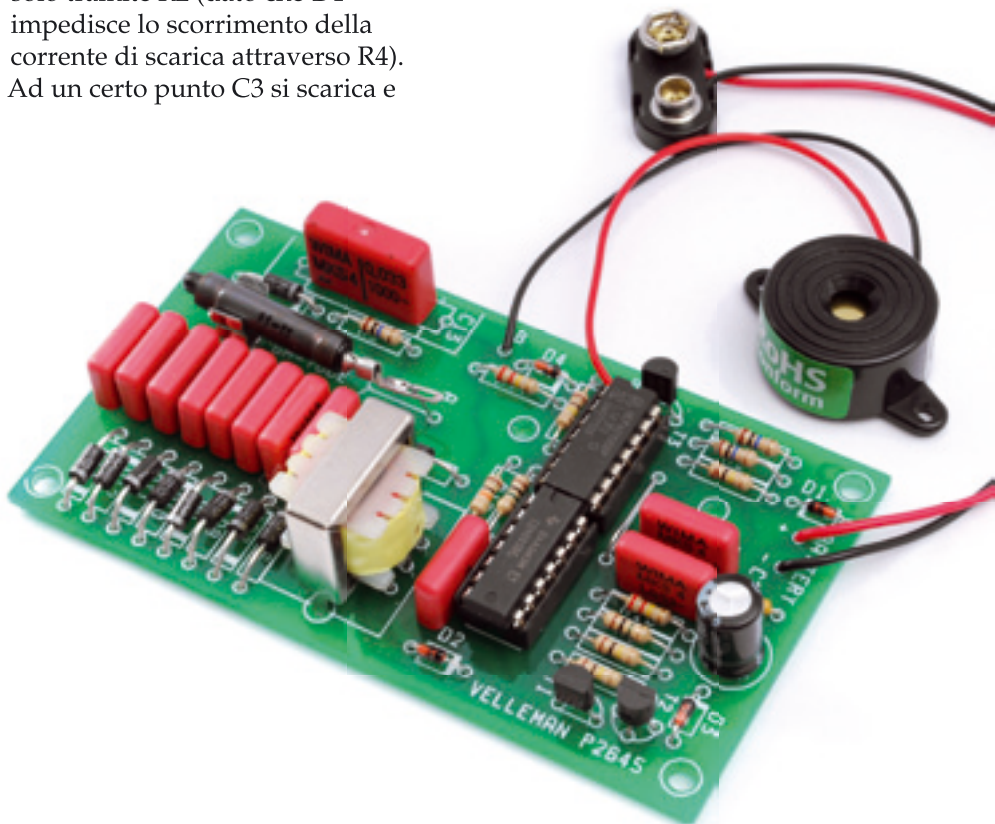
doppio multivibratore astabile, il primo dei quali è basato su IC1a ed IC1f (due NOT con ingresso a trigger di Schmitt) e produce un'onda rettangolare alla frequenza di circa 0,1 Hz con la quale modula l'attività dell'altro generatore. Il funzionamento si può spiegare così, partendo dalla condizione in cui il condensatore C3 è scarico: supponendo che l'ingresso della IC1f sia a zero logico, quello della IC1a si trova nella medesima condizione perché C3 è scarico e lascia transitare lo stato alto; allora il piedino 2 si trova a zero e questa condizione conferma il livello 1 all'uscita della IC1f. Per effetto di ciò C3 prende a caricarsi tramite la rete R2, R4, D1, e la tensione sul piedino 1 dell'IC1a scende fino a valere zero logico; a questo punto il piedino 2 passa a livello alto e forza a zero il 12, cosicché C3 prende ora a scaricarsi, ma solo tramite R2 (dato che D1 impedisce lo scorrimento della corrente di scarica attraverso R4). Ad un certo punto C3 si scarica e

prende a ricaricarsi con polarità opposta, quindi il piedino 1 della IC1a torna a trovarsi a livello alto ed il 2 scende nuovamente a zero logico, forzando il 12 a livello alto e ricominciando il ciclo. La cosa procede all'infinito determinando all'uscita della IC1f una forma d'onda rettangolare periodica, avente duty-cycle minore del 10 %, in quanto il fatto che C3 venga scaricato più lentamente di come si carica (per effetto del D1) rende il livello alto più breve di quello basso.

L'onda prodotta dal primo generatore abilita e disabilita le NAND IC2c e IC2d, nel senso che quando i piedini 8 e 12 di queste ultime sono a livello basso le loro uscite sono bloccate ad uno logico (e sia T1 che T2 restano interdetti) mentre quando vengono poste ad 1 logico lo stato delle uscite dipende da

ed alimenta la rispettiva metà del primario del trasformatore, l'altro risulta interdetto cosicché il mezzo primario corrispondente rimane isolato. Ciò permette di presentare ai capi del secondario una tensione di forma d'onda pressoché rettangolare, alternata, dell'ampiezza di circa 150 Vpp, ovvero con valore massimo di 75 V; dato che, considerando la caduta Vce di T1 e T2, ogni mezzo primario riceve circa 8,5 volt, ciò significa il rapporto spire riferito ad ogni mezzo primario è $75/8,5=8,82$.

Prima di vedere cosa accade alla tensione generata dal trasformatore, spieghiamo brevemente in che modo funziona il generatore d'onda rettangolare che pilota i transistor (T1 e T2) incaricati di far pulsare i mezzi primari del TRAF01; si tratta in realtà di un



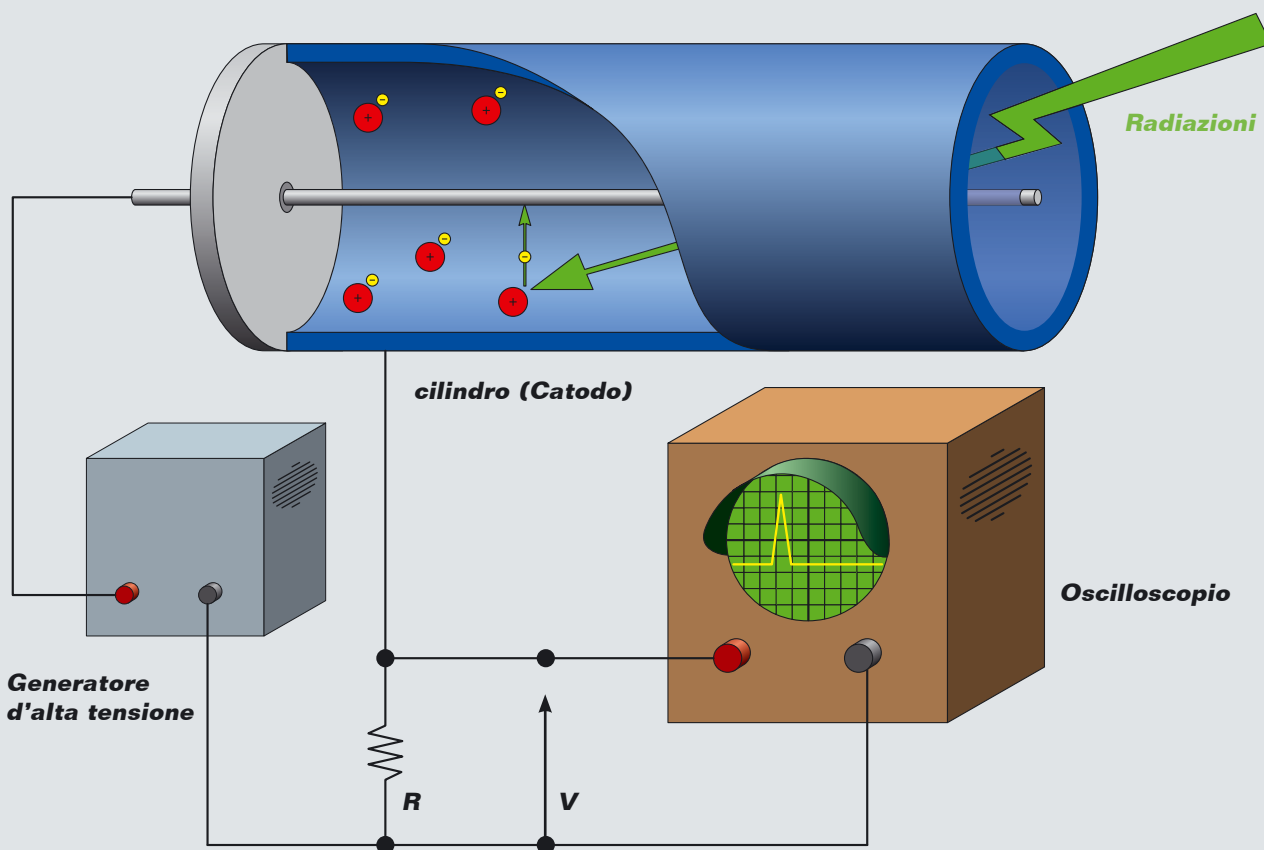
Il tubo Geiger-Muller

Più semplicemente detto tubo Geiger, è il più antico e funzionale rilevatore di radiazioni ionizzanti alfa e beta (quindi quelle emesse dai nuclidi radioattivi) ma anche gamma; strutturalmente è un cilindro metallico cavo e aperto alle estremità, in mezzo al quale, assialmente, passa un elettrodo composto da un filo metallico teso o una bacchetta metallica, sospeso mediante un efficace isolante elettrico in grado di evitare il contatto con il cilindro. Le estremità vengono chiuse a tenuta in modo da sigillare il cilindro, che viene riempito di gas a bassa pressione (ad esempio una miscela di argon e vapore di alcool alla pressione di 0,1 bar). Tra il cilindro ed il filo viene applicata una differenza

di potenziale dell'ordine di centinaia o anche migliaia di volt, positiva su filo rispetto al cilindro. Quando le radiazioni investono il tubo ed entrano dalle estremità, con energia sufficiente a ionizzare alcuni atomi del gas, si formano coppie di ioni (atomi privati di un elettrone strappato dall'urto della particella ionizzante costituente la radiazione) ed elettroni, che in assenza di tensione fra cilindro e filo si ricombinerebbero restituendo l'energia ricevuta sotto forma di onda elettromagnetica. Applicando tensione di ampiezza sufficiente fra cilindro e filo, gli elettroni staccati dai nuclei di gas vengono messi in circolazione ed originano una corrente elettrica, facilmente rilevabile ponendo in serie al

tubo una resistenza di valore adeguato. Dunque, ad ogni particella ionizzante dovuta alla radiazione, ai capi del resistore registriamo un impulso, che può essere udito sotto forma di crepitio in un auricolare o cuffia ad alta impedenza collegato in parallelo alla resistenza tramite un condensatore di disaccoppiamento o, meglio, un condensatore ed un trasformatore che protegge l'operatore da eventuali scariche elettriche dovute all'alta tensione di polarizzazione del tubo Geiger-Muller. I tubi Geiger-Muller esistono di vario genere e possono rilevare più tipi di radiazioni, derivanti da particelle beta o raggi gamma. Generalmente un buon contatore geiger con tubo separato dall'apparecchio ha il tubo rile-

vatore fatto di materiale attraversabile dalle particelle beta, che può essere coperto da un involucro impermeabile a dette particelle. In questo modo, con una finestra apribile e chiudibile il contatore potrà rilevare sia tutte le radiazioni fino al grado di penetrazione delle particelle beta (finestra aperta), sia solo le radiazioni con grado di penetrazione uguale o superiore ai raggi x e gamma (finestra chiusa). Alle dimensioni del tubo sono legate le prestazioni in fatto di sensibilità: più sono grandi, più il tubo può ricevere radiazioni ionizzanti e con maggiore costanza; se le dimensioni sono piccole (ad esempio meno di una pila ministilo) il tubo è adatto a misurare livelli molto alti di radioattività.



quello degli ingressi 9 e 13. La tensione prodotta dal primo astabile va anche a bloccare periodicamente il secondo generatore

d'onda rettangolare, costruito intorno ad IC1d, IC1e ed IC2b, il quale funziona analogamente al primo, con la sola differenza che

c'è una NAND a consentire o arrestare l'oscillazione. In pratica quando il piedino 6 della IC2b riceve lo zero logico l'uscita della

porta è bloccata a livello alto, mentre quando si trova sotto-posta all'uno logico, il piedino 4 commuta in funzione dei cicli di carica e scarica del C4. All'uscita della IC1e si trova una NOT che inverte il livello logico, cosicché IC2c e IC2d ricevono, ai piedini rispettivamente 9 e 13, stati logici differenti; ciò assicura che i transistor T1 e T2 vadano in conduzione sempre l'uno alternativamente all'altro. Questo secondo oscillatore genera impulsi a circa 2,5 kHz.

Bene, detto ciò vediamo cosa succede quando i "burst" di impulsi rettangolari prodotti dal generatore fanno funzionare il trasformatore e determinano una tensione alternata ai capi del secondario: la componente alternata viene resa continua dal primo duplicatore di tensione, realizzato dai condensatori C5 e C6 e dai diodi al silicio D6 e D7; il funzionamento del duplicatore si spiega dicendo che al primo impulso negativo ai capi del secondario del trasformatore la corrente fluisce in D6 e carica C5 con la tensione di picco diminuita della caduta sul diodo (la polarità della tensione sarà positiva sul catodo del diodo). Quando la tensione ai capi del secondario s'inverte di polarità, l'impulso si somma ai circa 75 V presenti ai capi del C5 e la somma di queste componenti carica C6 attraverso D7, con polarità positiva sul catodo di quest'ultimo. Essendo, il duplicatore, collegato a massa tramite il diodo D5, tra il catodo del diodo D7 e massa si registra una tensione pari a circa il doppio del picco sul secondario, ossia 150 V.

Il duplicatore è seguito da altri tre circuiti identici, che elevano ancor più la tensione, nel senso che ognuno la duplica; infatti, quando la tensione ai capi del se-

condario del trasformatore torna negativa, si riversa sul C7 sommata a quella presente ai capi del C6 ed alla successiva inversione C7, carico a $150+75$ V, si scarica su C8, sommata all'impulso positivo presente al secondario del TRAFO1. Insomma, C8 dopo quattro inversioni di polarità presenta una tensione pari al quadruplo dell'ampiezza degli impulsi forniti dal secondario del trasformatore, ossia circa 300 V. Alla fine della catena di quattro duplicatori, la tensione ammonta a circa 600 volt in continua, che il diodo D14 applica all'anodo

del tubo Geiger-Muller mediante la resistenza di limitazione R3, da ben 10 Mohm; il valore è sì elevato in quanto il tubo assorbe una corrente praticamente nulla, che assume un valore appena apprezzabile solamente quando esso viene investito da particelle radioattive, allorché il gas si ionizza ed avviene una scarica che rappresenta un passaggio di corrente elettrica. Ciò spiega perché possiamo alimentare il tubo Geiger-Muller con una catena di duplicatori di tensione, circuiti che certamente elevano la differenza di potenziale rispetto



Ore, minuti e radiazioni

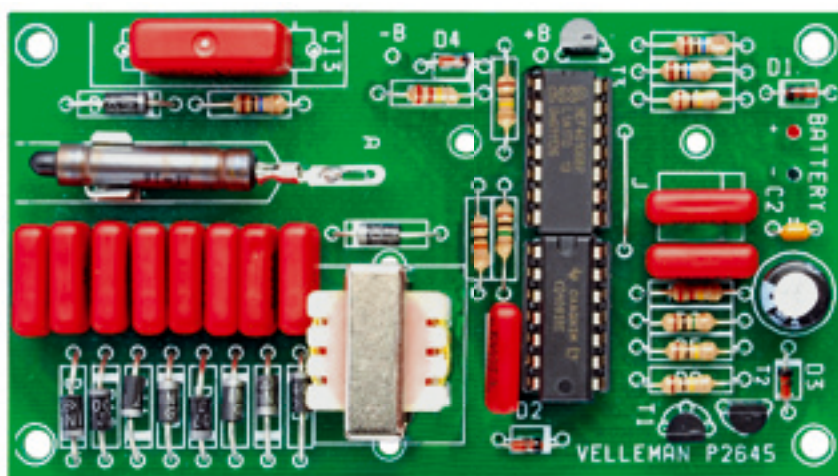
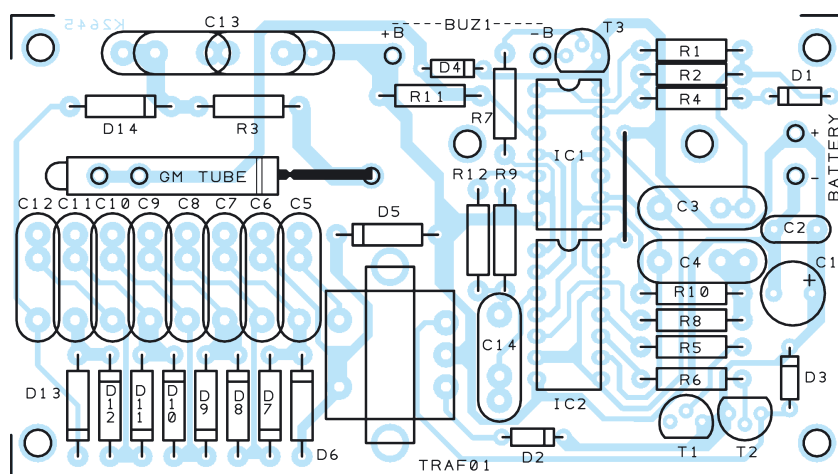
L'ansia indotta dall'incidente nucleare in Giappone ha fatto impennare le richieste di contatori Geiger. Il mercato ha risposto proponendone versioni anche molto originali e combinate con altri strumenti. Un esempio è l'orologio con contatore incorporato: si chiama PM1208M, è prodotto dalla Polimaster (www.polimaster.com), ed è un classico segnatempo per ore, minuti e secondi, con cassa metallica, subacqueo fino a 100 m, dotato di schermo retroilluminato e simile in tutto per tutto ad un comune orologio, solo che incorpora un contatore Geiger che permette di avere sempre sotto controllo il livello delle radiazioni. In caso venga superata la soglia di sicurezza, emette un segnale acustico; inoltre può memorizzare fino a 500 eventi e può essere collegato ad un computer (tramite un'interfaccia ad infrarossi) per scaricare i dati ed elaborarli con un apposito software. Sebbene non serva prettamente a rilevare la contaminazione di merci ed alimenti, questo prezioso segnatempo è l'ideale per chi deve viaggiare in zone a rischio come i Paesi in cui si sono verificati incidenti nucleari.

Elenco Componenti:

R1, R2, R3: 10 Mohm
 R4, R5, R6, R7: 100 kohm
 R8, R9: 1 Mohm
 R10, R11: 220 kohm
 R12: 10 kohm
 C1: 470 μ F 25 VL elettrolitico
 C2: 100 nF multistrato
 C3: 1 μ F 63 VL poliestere
 C4: 1 nF 63 VL poliestere
 C5, C6, C7, C8, C9, C10, C11, C12: 33 nF 400 VL poliestere
 C13: 33 nF 1000 VL poliestere
 C14: 47 nF 250 VL poliestere
 D1, D2, D3, D4: 1N4148
 D5, D6, D7, D8, D9, D10, D11, D12, D13, D14: 1N4007
 T1, T2, T3: BC557
 IC1: HEF40106BP
 IC2: CD4093BE
 BUZ1: Buzzer 9V-12V

Varie:

- Tubo Geiger-Muller
- Zoccolo 7+7 (2 pz.)
- Clip per batteria 9V
- Trasformatore
- Circuito stampato



a quanto ottenibile da comuni raddrizzatori a diodi, tuttavia permettono di prelevare una corrente dimezzata per ogni duplicazione di tensione effettuata. Il catodo del tubo termina a massa mediante R11, la quale chiude il circuito dell'alimentazione fornita dalla catena di duplicatori; ogni volta che si verifica un impulso di corrente causato dalla ricezione di una particella radioattiva, ai capi del resistore si sviluppa un picco di tensione che IC1b inverte di livello, fornendo zero logico e triggerando il monostabile formato da IC2a e IC1c. Il funzionamento di questo monostabile è il seguente: a ripro-

so (cioè quando il contatore non viene investito dalle radiazioni) l'uscita della IC1b è a livello alto e così il piedino 4 della NAND IC2a, la cui uscita, supponendo scarico C14 e quindi il piedino 5 della IC1c a livello basso, è allo stato logico zero. L'uscita della NOT (piedino 6) si trova a livello alto e così pure il piedino 1 della NAND. Quando viene rilevata una radiazione d'intensità sufficiente a determinare ai capi della R11 un impulso ampio almeno tre volt, l'uscita della IC1b commuta da uno a zero logico e forza in tale condizione il piedino 4 della IC2a, la cui uscita passa a livello alto; la tensione

corrispondente attraversa C14, inizialmente scarico, giungendo al pin 5 della NOT IC1c, la cui uscita commuta a zero logico e mantiene in tale condizione il piedino 1 della IC2a, facendo sì che l'uscita della NAND resti ad 1 logico anche se l'impulso fornito dal Geiger smette. Lo stato all'uscita della IC1c resta basso fin quando C14 non si è caricato (attraverso R9) abbastanza da lasciar tornare a zero logico il piedino 5 di tale porta, allorché torna ad assumere l'1 logico; per tutto il tempo in cui il piedino 6 resta a zero, tramite la resistenza R7 viene polarizzato in base il transistor NPN T3, il quale va in

saturazione ed alimenta il cicalino BUZ1, facendogli emettere una nota acustica (BUZ1 è del tipo con oscillatore integrato). Una volta carico C14, l'uscita della IC1c torna ad assumere il livello alto e così pure il piedino 1 della IC2a, cosicché se non vi sono nuovi impulsi all'uscita del tubo Geiger-Muller ed anche il pin 4 è ad uno logico, l'uscita della NAND torna a zero e scarica il condensatore C14.

Notate che nel monostabile manca il diodo di protezione in parallelo alla resistenza R9; ciò perché si sfrutta il fatto che all'ingresso dei moderni CMOS sono presenti diodi di protezione posti verso l'alimentazione e massa, quindi quando l'uscita della IC2b torna a zero logico e scarica C14, la tensione inversa derivante dalla corrente di scarica viene cimitata da detti diodi e non danneggia gli ingressi della IC1c.

Il monostabile serve a garantire impulsi sonori sul cicalino la cui durata è indipendente da quella dei picchi rilevati sul catodo del tubo, che per loro natura sarebbero irregolari. Gli impulsi prodotti dal circuito possono essere ascoltati anche con un altoparlante o una cuffia aventi impedenza di almeno 32 ohm, ovvero inviati all'ingresso di un circuito di conteggio o ad un computer nel quale giri un apposito software; nulla vieta di abbinare all'avvisatore acustico uno strumento a lancetta, che si può pilotare molto semplicemente ponendogli in serie una resistenza da 100 kohm ed un trimmer di pari valore e in parallelo un condensatore da 1 μ F. Lo strumento dovrà essere un microamperometro da 100÷500 microampere f.s.

REALIZZAZIONE PRATICA

Bene, giunti a questo punto possiamo spiegare in che modo

Tipicamente le radiazioni si esprimono in dosi, intendendo con dose la quantità di radiazione cui sono esposti o che viene assorbita dai materiali o corpi che vi sono esposti; la dose di esposizione, espressa in Roentgen (R) definisce la capacità ionizzante delle radiazioni elettromagnetiche come raggi X e gamma: i Roentgen è la dose di raggi X (con energia pari a 250 keV) che in una superficie d'aria di 1 cm², alla pressione di 760 mm di mercurio e alla temperatura di 0 °C, riesce a ionizzare 2 miliardi di atomi. L'intensità della dose di esposizione si esprime in R/ora e indica quanto l'ambiente sia contaminato dalle radiazioni.

Ai fini della valutazione del danno causato dalle radiazioni, assume importanza la dose assorbita, espressa in RAD (*Radiation Adsorbed Dose*): 1 RAD corrisponde a 0,01 joule/kg.

La dose assorbita non è comunque strettamente correlata col danno che un corpo può ricevere dall'esposizione alle radiazioni, quindi per collegare i RAD all'equivalente effetto, ad esempio, su un essere vivente, bisogna usare un fattore di conversione, una sorta di fattore di merito (di demerito, in questo caso) che possiamo chia-

mare Q e che indica l'efficienza della radiazione assorbita nel determinare alterazioni sul corpo. Questo fattore va moltiplicato per la dose assorbita e consente di ottenere l'equivalente di dose, ossia l'effetto pratico della dose assorbita, anche detto REM o Roentgen Equivalent Man. Per le radiazioni alfa e quelle costituite da protoni e neutroni, Q è pari a 10, mentre vale solo 1 per le radiazioni beta e gamma.

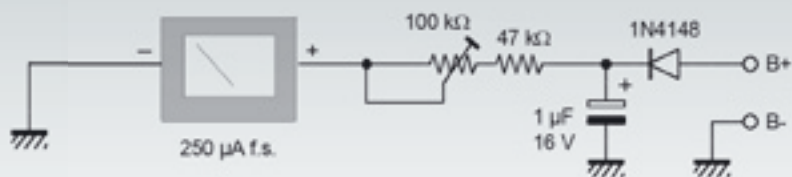
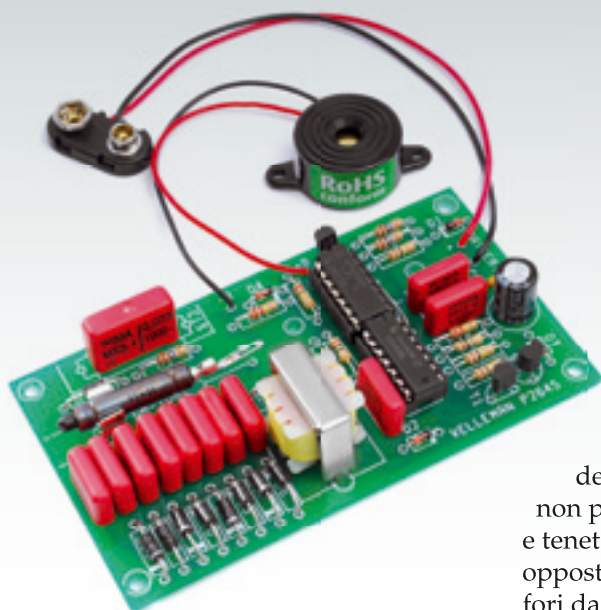
L'equivalente di dose in REM si ottiene dunque dal prodotto RA-DxQ, quindi vediamo che una dose assorbita di 1 RAD determina un equivalente di 1 REM se deriva da raggi beta o gamma e 10 REM se è causato da raggi alfa e simili. Ciò indica come a parità di dose assorbita, le radiazioni facciano danni differenti e spiega perché una dose assorbita di un tipo di radiazioni possa essere innocua mentre di un altro tipo faccia già danno.

L'equivalente di dose si misura in Sievert (SV) ricordando che un SV vale 100 REM.

Il contatore Geiger è in grado di misurare la dose di esposizione, ma non può dare indicazioni sull'equivalente di dose; è però utile a valutare l'entità delle radiazioni e soprattutto il livello di contaminazione.

costruire il contatore Geiger, a partire dal circuito stampato, facilmente ottenibile per fotoincisione partendo dalla traccia lato rame scaricabile dal nostro sito Internet www.elettronica.in.it; essendo a singola faccia, la basetta è di facile realizzazione. Una volta incisa e forata, è pronta per ospitare i componenti, che collegherete a partire dalle resistenze e dai diodi al silicio, quindi proseguendo con gli zoccoli per i due integrati ed il ponticello di interconnessione (ottenibile da un avanzo di terminale di resistenza); proseguite con i condensatori, prestando attenzione alla polarità dell'elettrolitico, e con i

transistor, che monterete come indicato nella disposizione componenti visibile in queste pagine. Il tubo Geiger-Muller miniatura va montato sul circuito stampato fermandolo con una clip da fusibile 5x20 che lo stringa per il catodo, e connettendolo con uno spezzone di filo in rame ed una clip di piccolo diametro sull'anodo. Il trasformatore elevatore va realizzato utilizzando un nucleo a doppia E in lamierino di ferro, delle dimensioni di circa 2,5x2x5 mm con colonna centrale di 6 o 7 mm; il primario va costruito avvolgendo 18+18 spire di filo in rame smaltato del diametro di 0,3 mm tutte in un verso e unendo la



Il circuito da collegare all'uscita del contatore (in parallelo al cicalino) per visualizzare la radioattività con uno strumentino a lancetta.

fine del primo avvolgimento con l'inizio del secondo a formare la presa centrale. Per il secondario, servono circa 160 spire di filo in rame smaltato del diametro di 0,2 mm. Alla fine degli avvolgimenti, bloccate il filo avvolto con del nastro adesivo o della vernice e scoprite, raschiandoli con un temperino, gli estremi per asportarne lo smalto, che altrimenti impedirebbe le saldature. Ricordate che il trasformatore va disposto con il primario rivolto agli integrati ed il secondario verso la fila di diodi del duplicatore. Per completare il circuito saldate nelle piazzole + e - BATTERY i fili di una presa volante per pile da 9 volt e in B+ e B-, rispettivamente positivo e negativo del cicalino piezo. Infine, inserite il 4093 ed il 40106 nei rispettivi zoccoli, ricordando che il secondo va vicino al T3 e che entrambi devono essere orientati (vedere il piano di montaggio pubblicato in queste pagine) con il riferimento verso il lato in cui si trova lo stesso transistor, ovvero con il lato del piedino 1 verso la parte di circuito contenente il trasformatore. L'intero circuito va inserito in un contenitore plastico dotato di contenitore per la pila, quest'ul-

tima da 9 volt; fate in modo che sotto la zona dei duplicatori di tensione non passino fili di alcun genere e tenete cicalino e pila dalla parte opposta. Prevedete uno o più fori dal lato del contenitore in cui fisserete il cicalino e magari, se volete portare all'esterno il segnale del contatore, disponete su un altro lato una presa jack da pannello da 4,5 mm mono collegata con l'elettrodo esterno alla massa dello stampato e con quello interno al positivo del cicalino (B+). Dato che ha una zona sottoposta all'alta tensione (oltre 600 V) evitate assolutamente di tenere in mano il circuito quando è alimentato; per le prove appoggiate ad un piano in materiale isolante o prima racchiudetelo in un idoneo contenitore plastico.

UTILIZZO DEL CONTATORE

Descritto il montaggio, vediamo come usare il contatore Geiger, partendo dalla considerazione che il tubo usato in questo progetto, come la gran parte dei Geiger-Muller, non è idoneo a rilevare radiazioni di bassissima entità come quelle naturalmente presenti e innocue per l'uomo; queste potrebbero far emettere sporadici "beep" al circuito, che non vanno, come frequenza, oltre i 10 al minuto. Per l'uso, considerate che in linea di massima il tubo qui usato emette un impulso in corrispondenza di ogni particella ionizzante rilevata e che, tolleranza (del 10 %) compresa,

tipicamente genera 2,5 impulsi al secondo per una radiazione di 1 millirem/ora. La radioattività ambientale di fondo, presente normalmente anche nelle zone non considerate a rischio, provoca un'esposizione dell'ordine di 10÷40 microrem/ora, corrispondenti a non più di 10 beep al minuto. Quindi se rilevate oltre 20 impulsi al minuto, potete considerare che l'oggetto analizzato o l'ambiente in cui vi trovate è ad un livello di radiazione che provoca un'esposizione oltre la soglia di non pericolosità. Per effettuare la misura su un oggetto che ritenete contaminato dovete avvicinarvi l'apparecchio, rivolto con il tubo Geiger-Muller verso di esso: se c'è radioattività, il cicalino emetterà un suono. ■



per il MATERIALE

Il contatore Geiger descritto in queste pagine è disponibile in scatola di montaggio, prodotto da Velleman e distribuito da Futura Elettronica. Il kit (cod. K2645, Euro 209,00 IVA compresa) comprende tutti i componenti, le minuterie, la basetta forata e serigrafata ed il tubo Geiger-Muller. Non è compresa la batteria.

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



FIRMWARE PER OROLOGIO A NIXIE

Molti lettori ci hanno richiesto un commento dettagliato alla soluzione del concorso "Scrivi il firmware" del mese scorso; eccola in queste pagine.

..... a cura della REDAZIONE

Si è da poco concluso il concorso (proposto nella nostra rubrica Elettronica Insieme) che chiedeva ai lettori di cimentarsi nella scrittura del firmware per l'orologio con nixie pubblicato nel fascicolo n° 145; nel nostro sito si trova già la soluzione al quesito, ma abbiamo ritenuto utile pubblicare la spiegazione dettagliata del programma del microcontrollore. La versione di firmware proposta dal risolutore del quiz apporta qualche leggera miglioria rispetto a quella originale implementata dalla Velleman: ad esempio è stata inserita la funzione antirimbombo sul pulsante di impostazioni, utile perché a volte risulta difficile impartire i comandi; altra variante è l'aumento del tempo a disposizione per impostare il formato dell'orario (12 o 24 ore). Possono essere considerate migliorie di poco

conto, perché solitamente una volta impostato l'orario questo non deve essere più variato, ma invece non è così, dato che il sistema non è dotato di batteria tampone e quindi in caso di blackout, al successivo avvio, quello che viene mostrato è l'ultimo orario visualizzato; pertanto sarebbe necessario sincronizzare nuovamente l'orario. Le migliorie apportate sono però state inserite a discapito della funzione di risparmio energetico presente nel firmware originale, pertanto non è possibile stabilire in quale fascia oraria le Nixie devono rimanere spente; ciò non dovrebbe essere un problema, perché in pratica un orologio deve essere visibile in qualsiasi momento. Ma lasciamo la parola al Sig. Alessandro, affinché ci spieghi in cosa consiste il suo firmware. ...il programma è stato strutturato in più funzioni

Listato 1

```

INCLUDE "modedefs.bas"

@ DEVICE MCLR_OFF
@ DEVICE INTOSCIO

OSCCON=%01110101'Imposto frequenza oscillatore ad 8MHz
OPTION_REG.7=1 'Abilito pull-up per utilizzo RAO
WPUA.0=1 'Abilito pull-up su RAO
OPTION_REG.6=0 'Interrupt su fronte di discesa
OPTION_REG.5=1 'Interrupt su transizione pin
OPTION_REG.4=1 'Incrementa Timer su interrupt
INTCON=%10000000'Disabilito interrupt RA2 all'avvio
PIE1=0 'Disabilita tutti gli altri interrupt
PIE2=0 'Disabilita tutti gli altri interrupt
ANSEL=0 'Disabilito le porte analogiche
ANSELH=0 'Disabilito le porte analogiche
TRISB=0 'PORTB uscite
TRISC=0 'PORTC uscite

'***** DICHIARAZIONE PIN *****
symbol DISP1 = PORTB.7
symbol DISP2 = PORTB.6
symbol DISP3 = PORTB.5
symbol DISP4 = PORTB.4
symbol ZERO = PORTC.7
symbol UNO = PORTC.6
symbol DUE = PORTC.5
symbol TRE = PORTC.4
symbol QUATTRO = PORTC.3
symbol CINQUE = PORTC.2
symbol SEI = PORTC.1
symbol SETTE = PORTC.0
symbol OTTO = PORTA.5
symbol NOVE = PORTA.4
symbol HZ = PORTA.2
symbol PUSH = PORTA.0
symbol NEON = PORTA.1
symbol RETRO = PORTA.3

INPUT PUSH
input HZ
INPUT RETRO
OUTPUT NEON

'***** DICHIARAZIONE VARIABILI *****
TMP VAR BYTE 'Variabile di comodo
TMP1 VAR BYTE 'Variabile di comodo
FREQ VAR BYTE 'Frequenza rilevata, usata come clock dell'orologio
SECONDI VAR BYTE 'Secondi dell'orologio
MINUTI VAR BYTE 'Minuti dell'orologio
ORE VAR BYTE 'Ore dell'orologio
CONTO VAR BYTE 'variabile di comodo per il conteggio dell'orologio
CONTO_FQ VAR BYTE 'variabile di comodo per il conteggio dell'orologio
PWMVAL VAR BYTE 'Valore del PWM
FUNZIONE VAR BYTE '0:Startup - 1:12/24hh - 3:Visualizza Orario
HH VAR BYTE '12:12hh - 24:24hh
NOW VAR BYTE 'Imposto i secondi attuali
DATO1 VAR BYTE 'Display a sinistra del separatore
DATO2 VAR BYTE 'Display a destra del separatore
CIFRA VAR BYTE 'Numero della Nixie da visualizzare
DISPLAY VAR BYTE 'Nixie da accendere

'***** IMPOSTAZIONE VARIABILI *****
PWMVAL=127 'Valore di default del PWM
SECONDI=0
MINUTI=0
ORE=0
TMP=0
TMP1=0
CONTO=0
CONTO_FQ=0
FUNZIONE=0

'***** LETTURA CONFIGURAZIONE IN EEPROM *****
eeprom 0,[24,0,0] 'Al primo avvio memorizzo la tipologia 24hh

```

per essere facilmente intuibile da tutti, pertanto in alcuni casi, data la semplicità del programma, ho sorvolato nell'inserire alcuni commenti, ma ogni singola variabile è stata commentata descrivendo in alcuni casi anche il valore che può assumere, così da rendere più leggibile il programma. Ma soprattutto, il nome di ogni variabile è stato scelto per collegarla intuitivamente, per quanto possibile, al compito che svolge. Stesso discorso per le funzioni (subroutine) in cui viene descritta l'esatta funzione che esse svolgono.

Riguardo, invece, alla logica del programma, è stata rispettata quella della Velleman, nel senso che all'avvio si effettua un controllo sulla frequenza di rete (funzione RILEVA_FREQUENZA), ovvero si verifica se si tratta di una rete a 50 o 60 Hz, allo scopo di stabilire la frequenza di clock dell'orologio. Se la frequenza è tra le due appena indicate, il sistema viene avviato regolarmente, ma se per ipotesi il sistema è stato alimentato con un alimentatore DC, il clock per la gestione dell'orologio viene a mancare, quindi i display visualizzano il valore "00", ad indicare che la frequenza è errata e il programma si arresta. Invece, nell'ipotesi che l'alimentazione fornita all'orologio sia in alternata e di frequenza corretta, il passo successivo compiuto dal firmware consiste nel leggere la EEPROM del PIC per prelevare l'ultimo orario visualizzato e il formato dell'orario (12 h oppure 24 h). Lette le impostazioni, viene data la possibilità di variare sia il formato dell'ora che l'orario impostato, lasciando un tempo sufficiente ad impostare il formato orario, che termi-

nerà nel momento in cui non verrà premuto il pulsante per 3 secondi. Ciò comporterà la visualizzazione dell'orario (hh.mm) che potrà essere variato in qualsiasi momento premendo l'unico pulsante disponibile, il quale non farà altro che incrementare l'orario a due velocità differenti, lentamente con funzione antirimbalo se si preme e rilascia il pulsante ad ogni incremento dei minuti, oppure a scorrimento veloce se il pulsante viene tenuto premuto.

LE SUBROUTINE

Qui di seguito sono elencate le funzioni utilizzate e di ognuna è riportata la spiegazione logica, per rendere di più facile comprensione il file sorgente mostrato nel Listato 1, privo delle subroutine *Accendi cifra e Incremento orario* (il firmware completo è comunque disponibile sul nostro sito).

• CONTA

Si tratta di una funzione che viene richiamata su interrupt, ovvero ogni qualvolta venga rilevato un ciclo di clock (prelevato dalla frequenza di rete), pertanto a seconda se si tratta di un 50Hz o 60Hz di rete si sa a priori ogni quanti cicli dovranno essere incrementati i secondi e di conseguenza i minuti e le ore che poi dovranno essere visualizzati a display. Se l'orario viene aggiornato, viene automaticamente salvato in EEPROM l'attuale orario.

• RILEVA_FREQUENZA

Permette di individuare la frequenza di clock della sorgente di alimentazione per permette la corretta gestione dell'orario. Viene utilizzata la funzione "COUNT

Listato 1 - seguito

```
read 0,HH           'Lettura formato ore dalla EEPROM
read 1,ORE           'Lettura ultima ora salvata in EEPROM
read 2,MINUTI        'Lettura ultimi minuti salvati in EEPROM

***** INIZIO PROGRAMMA *****

gosub RILEVA_FREQUENZA 'Rilevazione frequenza di rete per il clock orario
INTCON.4=1           'Abilita interrupt RA2
ON INTERRUPT GOTO CONTA 'Quando avviene interrupt salta a CONTA
GOSUB VIS_FREQUENZA   'Visualizzazione Frequenza di rete elettrica
FUNZIONE=FUNZIONE+1
NOW=0
GOSUB VIS_HH          'Visualizzazione se orario a 12 ore o 24 ore
FUNZIONE=FUNZIONE+2

***** M A I N *****
MAIN:
  GOSUB VIS_ORARIO      'Visualizzazione orario aggiornato
  GOSUB CONTROLLA_PULSANTE 'Controllo se il pulsante è premuto
  GOTO MAIN

***** C O N T A *****
'Funzione richiamata su interrupt che incrementa l'orario
DISABLE
CONTA:
  CONTO=CONTO+1
  IF CONTO>=125 THEN 'Contatore per recupero tempo di 8ms dovuto ad istruzione PWM
    CONTO=0
    CONTO_FQ=CONTO_FQ+1
    IF CONTO_FQ>=FREQ THEN
      CONTO_FQ=0
      SECONDI=SECONDI+1
      NOW=NOW+1
      IF NOW=200 THEN
        NOW=0
      ENDIF
      IF SECONDI=60 THEN
        SECONDI=0
        MINUTI=MINUTI+1
        IF MINUTI=60 THEN
          MINUTI=0
          ORE=ORE+1
          IF ORE=HH+1 THEN
            ORE=1
          ENDIF
        ENDIF
        write 1,ORE 'Memorizzazione ore attuali
        write 2,MINUTI 'Memorizzazione minuti attuali
      ENDIF
    ENDIF
  ENDIF
RESUME
ENABLE

***** R I L E V A _ F R E Q U E N Z A *****
'Rilevazione frequenza della rete elettrica
RILEVA_FREQUENZA:
  count HZ,1000,TMP
  SELECT CASE TMP
    CASE IS < 5 'Alimentazione in DC
      FREQ=0
    CASE IS <= 55 'AC 55Hz
      FREQ=50
    CASE IS > 55 'AC 60Hz
      FREQ=60
  END SELECT
RETURN

***** V I S _ F R E Q U E N Z A *****
'Visualizzazione frequenza di rete elettrica letta in precedenza
VIS_FREQUENZA:
  DAT01=FREQ
  DAT02=255
  GOSUB VISUALIZZA
```

Listato 1 - seguito

```

if freq<>0 then
  IF SECONDI<3 THEN
    GOTO VIS_FREQUENZA
  ENDIF
endif
RETURN

'***** V I S _ H H *****
'Visualizzazione se si tratta di orario a 12 ore o 24 ore
VIS_HH:
  DATO1=HH
  DATO2=255
  GOSUB VISUALIZZA
  IF now<3 THEN
    GOSUB CONTROLLA_PULSANTE
    GOTO VIS_HH
  ENDIF
RETURN

'***** V I S _ O R A R I O *****
'Visualizzazione orario attuale
VIS_ORARIO:
  DATO1=ORE
  DATO2=MINUTI
  GOSUB VISUALIZZA
RETURN

'***** V I S U A L I Z Z A *****
'Effettua un refresh del display con i nuovi valori
VISUALIZZA:
  IF DATO1<>255 THEN 'Se è 255 devo tenere spento il display
    TMP=DATO1/10 'Ottengo le decine per visualizzare sul primo NIXIE
    CIFRA=TMP
    DISPLAY=1
    GOSUB VIS_NIXIE
    TMP=TMP*10 'Ottengo le unità per visualizzare sul secondo NIXIE
    TMP=DATO1-TMP
    CIFRA=TMP
    DISPLAY=2
    GOSUB VIS_NIXIE
  ENDIF

  IF DATO2<>255 THEN 'Se è 255 devo tenere spento il display
    TMP=DATO2/10 'Ottengo le decine per visualizzare sul terzo NIXIE
    CIFRA=TMP
    DISPLAY=3
    GOSUB VIS_NIXIE
    TMP=TMP*10 'Ottengo le unità per visualizzare sul quarto NIXIE
    TMP=DATO2-TMP
    CIFRA=TMP
    DISPLAY=4
    GOSUB VIS_NIXIE
  ENDIF
RETURN

'***** C O N T R O L L A _ P U L S A N T E *****
'Verifica se è premuto il pulsante, se si esegue l'operazione del
'caso a seconda se è in avvio o nella visualizzazione dell'orario
CONTROLLA_PULSANTE:
  SELECT CASE FUNZIONE
    CASE 1 'Visualizza se 12hh o 24hh
      IF PUSH=1 THEN
        now=0
        if hh=12 then 'Se attualmente è impostato 12 ore allora
          imposto 24 ore
          hh=24
        else 'Se attualmente è impostato 24 ore allora
          imposto 12 ore
          hh=12
        endif
        WRITE 0,HH
        WHILE PUSH=1
          NOW=0
          DATO1=HH
          DATO2=255

```

HZ,1000,TMP” per poter contare in un secondo (1000ms) quante variazioni alto/basso saranno presenti sulla porta RA2 del PIC (variabile HZ), valore che verrà memorizzato in TMP, ovvero una variabile temporanea di comodo.

- **VIS_FREQUENZA**
Questa funzione si spiega da sola, ovvero dopo avere rilevata la frequenza di clock mediante la funzione RILEVA_FREQUENZA, questa viene visualizzata sulle Nixie.
- **VIS_HH**
Tale funzione consente di visualizzare il formato data impostato, ovvero quello che viene letto in fase di avvio dalle locazioni di memoria del pic.
- **VISUALIZZA**
Questa funzione viene richiamata da diverse subroutine e permette di fare un refresh di quanto visualizzato sulle Nixie, visualizzando ciò che viene passato dalla variabile DATO1 e DATO2. La variabile DATO1 rappresenta il contenuto delle prime due Nixie da sinistra, mentre DATO2 le altre due Nixie rimanenti. Dato che una coppia di Nixie può anche rimanere spenta, ad esempio quando si visualizza la frequenza di clock o il formato dell'orario, se la variabile assume valore 255, si disattiva la Nixie, ovvero si spegne, mentre se assume un valore differente, tale valore viene visualizzato a display richiamando la funzione VIS_NIXIE.
- **CONTROLLA_PULSANTE**
Viene controllato se il pulsante è premuto oppure no e se lo è, a seconda della funzione

Listato 1 - seguito

```

        GOSUB VISUALIZZA
        PAUSE 5
    WEND
endif

CASE 3 'Visualizzazione orario
    TMP1=0
    IF PUSH=1 THEN
        GOSUB INCREMENTA_ORARIO
        WHILE PUSH=1
            GOSUB VIS_ORARIO
            PAUSE 5
            TMP1=TMP1+1
            IF TMP1=30 THEN 'Velocità di scorrimento molto lenta (antirimbalo)
                TMP1=0
                GOSUB INCREMENTA_ORARIO
                WHILE PUSH=1
                    GOSUB VIS_ORARIO
                    PAUSE 5
                    TMP1=TMP1+1
                    IF TMP1>=2 THEN 'Aumento la velocità di scorrimento
                        TMP1=0
                        GOSUB INCREMENTA_ORARIO
                ENDIF
            WEND
        ENDIF
    WEND
ENDIF
END SELECT
RETURN

```

attualmente svolta dal programma, determina un'operazione ben precisa, ad esempio se si è in fase di avvio (variabile FUNZIONE=1) la pressione del pulsante deve determinare la variazione da formato 12h a 24h e viceversa. Se invece la fase di avvio è superata, ovvero ci si trova nella visualizzazione dell'orario (variabile FUNZIONE=3), la pressione del pulsante determina l'incremento dell'orario attuale, quindi viene richiamata la subroutine di incremento (INCREMENTO_ORARIO)

- **INCREMENTO_ORARIO**
Funzione (omessa nel listato) il cui significato è chiaro dal nome: permette di incrementare l'orario a seguito della pressione del pulsante. Ogni incremento di orario, comporta il salvataggio in EEPROM dell'orario impostato.
- **ACCENDI_CIFRA**
Consente di accendere il

Nixie desiderato, facendogli assumere il valore scelto, ad esempio se si vuole accendere la terza Nixie visualizzando il valore 5, prima di richiamare codesta funzione, bisognerà impostare la variabile DISPLAY=3 e la variabile CIFRA=5. Per poter visualizzare quanto scelto, viene richiamata l'istruzione PWM. Vi è da tenere presente una particolarità importante riguardo l'istruzione PWM che viene gestita via software in quanto il PIC non dispone di PWM hardware, questo comporta una pausa nell'esecuzione firmware per un tempo pari al numero di cicli di clock (riferiti al quarzo di sistema) scelti per l'istruzione PWM che nel nostro caso è 2 cicli, pertanto dato che ogni ciclo con quarzo 8 MHz occupa 4ms, la pausa sarà di 8 ms. Questo significa che ad ogni interrupt derivante dalla Frequenza di rete (un interrupt ogni 20 ms per 50 Hz) in realtà passano 8 ms in più

del previsto. Durante l'esecuzione dell'istruzione PWM, l'interrupt rimane attivo comportando un conteggio troppo veloce pari ad 8 ms, tempo che deve essere obbligatoriamente recuperato ad ogni interrupt. Pertanto dato che si hanno 50 interrupt al secondo (per frequenza di 50 Hz) e ognuno comporta un incremento di 8 ms si ha $50/0,008=6250$ ovvero $6250/50=125$.

Questo 125 è proprio il valore che è possibile visionare nella subroutine CONTA. Tale valore 125 potrebbe anche essere sostituito con un'istruzione di pausa da 8 ms, ma comporterebbe uno spegnimento temporaneo delle Nixie, per questo motivo viene utilizzato questo contatore che deve raggiungere 125 ad indicare l'effettivo raggiungimento dei 20 ms corrispondenti alla frequenza di 50 Hz. Anche questa funzione è stata omessa, per motivi di spazio, nel listato qui pubblicato. ■

La TV è cambiata e tra poco non ci saranno più trasmissioni in analogico; scopriamo come funziona il digitale terrestre con qualche nozione utile a comprendere il progetto di trasmettitore DVB che presenteremo prossimamente.



DIGITALE TERRESTRE COME FUNZIONA

di FRANCESCO DONI

È praticamente impossibile non aver notato quella rivoluzione nella TV chiamata Digitale Terrestre (o anche DTT o DVB) quantomeno perché accendendo il televisore il giorno dopo il cosiddetto "switch off", anche chi non ne sapeva nulla ha dovuto fare i conti con questa realtà. Ma perché siamo passati dalla televisione analogica, tradizionale, a quella digitale? Ve lo spiegheremo in questo articolo, cercando anche di entrare nel merito di

questa nuova tecnologia e di scoprirne i segreti. La principale ragione dell'adozione del digitale terrestre consiste nella possibilità di aumentare i canali disponibili a parità di banda radio occupata e di trasmettere un segnale di qualità migliore. Un maggior numero di canali significa poter offrire all'utenza più canali generalisti e a carattere locale, ma anche programmi monotematici dedicati agli argomenti più vari (vedi l'esempio, via satellite, di

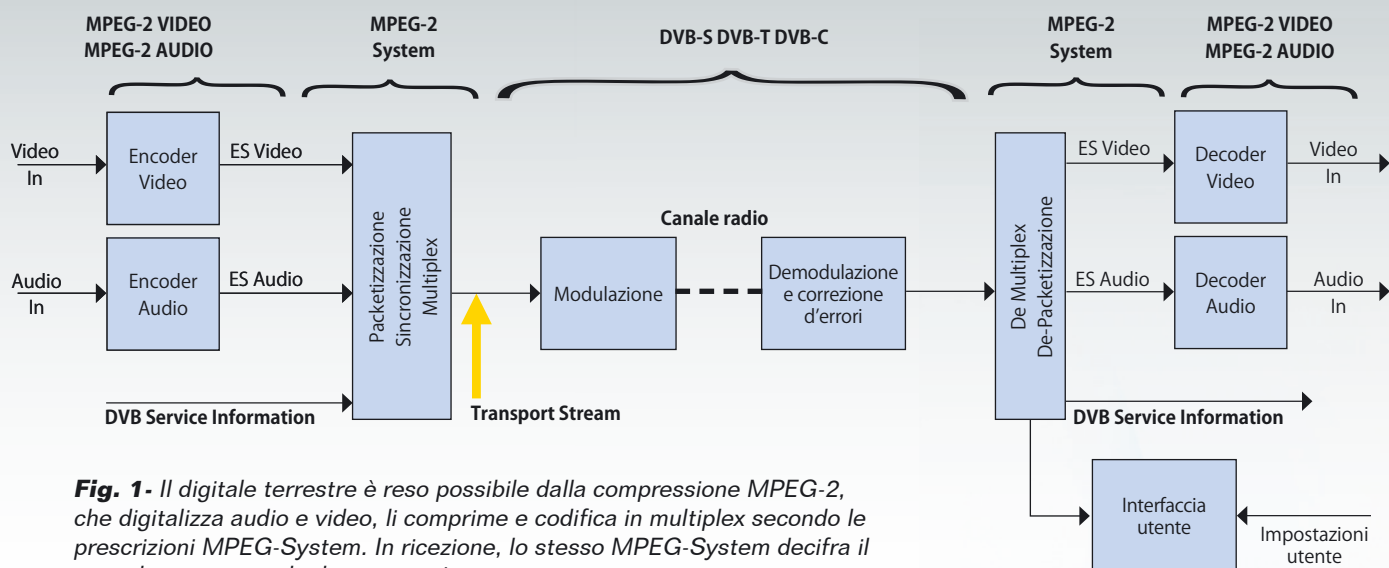


Fig. 1- Il digitale terrestre è reso possibile dalla compressione MPEG-2, che digitalizza audio e video, li comprime e codifica in multiplex secondo le prescrizioni MPEG-System. In ricezione, lo stesso MPEG-System decifra il segnale e permette la decompressione.

Sky) nonché servizi a pagamento. L'introduzione della tecnologia digitale nelle trasmissioni terrestri ha anche consentito una razionalizzazione delle frequenze rendendo disponibili, per la telefonia mobile e per i servizi annessi, importantissime porzioni di banda. E lo scenario di cui ci occupiamo in queste pagine potrebbe cambiare drasticamente tra qualche anno con l'affermarsi dell'IPTV. Ma torniamo a noi. Per capire come il DVB consenta di incrementare i canali disponibili, ricordiamo come funziona la televisione tradizionale: qui un canale è rappresentato da un'onda radio detta *portante*, modulata in ampiezza dal segnale video-composito che costituisce l'immagine e che è detto *modulante*; la stessa onda viene modulata in frequenza per trasmettere l'audio. Considerate le tolleranze dei circuiti accordati di sintonia degli apparecchi televisori e la normale deviazione di frequenza per deriva termica, i canali devono essere opportunamente spazati; inoltre ognuno deve avere una frequenza che permette di estrarre il segnale in ricezione senza apprezzabile distorsione. Perché ciò avvenga, la frequenza della portante deve essere $8 \div 10$ volte

quella della modulante. Nella TV tradizionale la distanza tra i canali è 8 MHz; siccome alla televisione sono dedicate le bande VHF da 52,5 a 230 MHz ed UHF da 470 a 860 MHz, il numero di canali RF è limitato a 58. Nel digitale, invece, mantenendo le stesse bande occupate, a modulare le portanti non sono i segnali videocompositi corrispondenti alle immagini da trasmettere, ma dati numerici ottenuti convertendo (mediante veloci A/D converter) il videocomposito in formato digitale. In altre parole, le componenti audio e video di ciascun canale televisivo vengono digitalizzate e multiplexate, insieme a quelle di altri tre, quattro o cinque canali; il risultato diventa il segnale digitale modulante che va a modulare la portante RF corrispondente a quello che nella TV analogica era il canale. Da qui esce la prima differenza tra TV analogica e digitale: nella prima il canale televisivo corrisponde al canale radio; nella seconda un canale radio può trasportare da uno a sei canali TV (o programmi, come si usa dire). Dal momento che l'audio e il video digitalizzati richiederebbero una banda consistente, che

obbligherebbe ad usare frequenze di campionamento e quindi data-rate eccessivi per le portanti RF delle bande TV VHF, le componenti digitalizzate vengono compresse secondo uno standard più che collaudato: l'MPEG2. La compressione è indispensabile anche perché l'avvento della TV ad alta definizione comporta il "consumo" di una maggiore larghezza di banda: infatti, mentre la normale televisione analogica ha 625 linee ed un refresh (verticale) di 50 semiquadri al secondo, l'HDTV (High Definition TV) ha una risoluzione di 1.080 linee ed un refresh doppio; per questa ragione le frequenze dei segnali video sono più elevate e digitalizzare il videocomposito in HD comporta la produzione di masse di dati più voluminose. Insieme ai dati audiovisivi, nella modulante di ciascun canale vengono inserite informazioni digitali che permettono al ricevitore di identificare i singoli canali televisivi contenuti ed altri come il televideo. Una caratteristica della TV digitale è che una certa emittente non può trasmettere indipendentemente il proprio segnale radio; quindi ogni canale RF deve comprendere un pacchetto di dati che sono quelli di

più canali televisivi che devono giungergli insieme. Il dispositivo che accorpa più canali TV e li trasmette multiplexati su un'unica portante radio viene chiamato *multiplexer* (o *Mux*) sebbene talvolta venga detto impropriamente *transponder*, perché così si chiama, nella TV da satellite, il sistema che nei satelliti riceve il segnale digitale dalla Terra e lo ritrasmette verso le parabole. Tale dispositivo è un trasmettitore operante su un unico canale radio, che trasporta più canali, modulati da un'apposita sezione elettronica di multiplex, la quale trasforma i dati dei singoli canali televisivi in uno stream. Il transponder è necessario perché se ogni stazione trasmettesse per suo conto sullo stesso canale di altre che appartengono allo stesso gruppo, ne disturberebbe la ricezione. Il termine transponder deriva anche dal fatto che il sistema riceve i dati digitali dei canali da trasmettere (ad esempio via rete in fibra ottica dalle singole stazioni TV comprese in un canale RF) e trasmette in radiofrequenza lo stream. Un'altra differenza sostanziale fra la televisione analogica e quella digitale è che nella prima la ricezione permette di vedere il segnale anche se molto debole: al massimo l'immagine appare sgranata, puntinata (effetto neve) tanto più quanto il segnale si indebolisce; ciò perché nella visualizzazione analogica il video appare tanto più nitido e intenso quanto più forte è il segnale, dal momento che la componente videocomposita modula la portante radio in ampiezza. Invece nella TV digitale l'ampiezza del segnale demodulato è poco rilevante, perché se supera la soglia corrispondente al livello logico alto, i dati vengono decodificati e l'immagine appare in pieno e

perfetta, mentre in caso contrario non si vede del tutto (schermo nero). L'immagine della TV digitale può anche subire alterazioni o essere aggiornata a scatti: ciò dipende dal refresh, da errori di temporizzazione o da momentanea perdita di segnale, che fa permanere per qualche istante il fotogramma precedente e poi torna all'immagine in movimento. Con il digitale terrestre la potenza di trasmissione è inferiore di quella della TV analogica. Il passaggio definitivo al digitale terrestre in Italia avverrà entro il 2012; in una prima fase ("switch over") alcune emittenti trasmetteranno anche in digitale, mentre in quella successiva ("switch-

off") verranno interrotti i segnali della TV analogica. Al momento mancano all'appello solo pochissime regioni.

LA COMPRESSIONE VIDEO

Il cuore della TV digitale, quindi anche del digitale terrestre, è la compressione MPEG-2. Nato ufficialmente nel 1994, questo standard può essere applicato ad una gran varietà di applicazioni e condizioni operative; allo scopo non stabilisce quali metodi di codifica debbano essere utilizzati o i dettagli del codificatore e decodificatore, ma specifica solo i formati in cui devono essere rappresentati i dati all'entrata del decodificatore e un insieme di re-

Dove nasce la TV digitale

I primi esperimenti di TV digitale risalgono agli anni '80 del secolo scorso, quando si compresero i vantaggi della digitalizzazione del segnale video: poter trasmettere senza curarsi dello standard di correzione dell'errore di fase del colore (PAL, SECAM) che affliggeva l'NTSC, ma anche poter fare più copie di un filmato senza degradarne la qualità. Il principale problema della trasmissione digitale era la mole di dati richiesta dalla conversione analogico/digitale, che imponeva data-rate compresi tra 108 e 270 Mbps, che a loro volta esigevano portanti a frequenze più alte di quelle usate oggi dalla TV. Per mantenere le stesse portanti e lasciare inalterati gli stadi di sintonia e le antenne dei televisori, ma anche i trasmettitori TV, la soluzione è stata trovata nella compressione dei dati: quella adottata è la MPEG (*Motion Picture Experts Group*). Si è partiti una ventina d'anni fa con lo standard MPEG-1, il cui obiettivo era immagazzinare audio e video per la prima generazione di CD-ROM, con un data-rate di 1,4 Mbit/s. Parallelamente si è lavorato alla

creazione di un altro sistema di codifica di audio/video adatto alla trasmissione attraverso differenti reti di comunicazione e per l'immagazzinamento in vari formati digitali: così è nato l'MPEG-2 (ISO 13818). Però, malgrado tale standard permettesse la compressione dei segnali audio e video e stabilisse meccanismi di sincronizzazione necessari per poter ricostruire e visualizzare nel ricevitore i differenti segnali originali, non era ancora sufficiente per rendere possibile l'introduzione della TV Digitale. Nel 1993 si è formato il gruppo di lavoro chiamato Digital Video Broadcasting (DVB), obiettivo del quale era la definizione di un sistema di televisione digitale per la diffusione via satellite, cavo e terrestre. Questo gruppo ha creato lo standard chiamato DVB, che oltre allo standard di compressione (fu scelto l'MPEG-2) definisce l'adattamento dello stesso a differenti canali di trasmissione (DVB-Satellitare, DVB-Terrestre, DVB-Cavo). Lo schema di funzionamento della TV digitale, indipendentemente da qual è il mezzo di trasporto del segnale, è quello mostrato nella **Fig. 1**.

gole e processi di decodifica. Ciò permette di sfruttare l'MPEG-2 per la diffusione della TV terrestre, da satellite e via cavo, per l'immagazzinamento di video su supporti ottici, per lo streaming di filmati e videocomunicazione, editing non lineare, post-produzione e in generale in tutte le applicazioni in cui serva trasportare video con buona qualità e occupando bande ristrette. Cerchiamo di spiegare brevemente come avviene la compressione, partendo da quella che riguarda il video e premettendo che il flusso di dati necessario per trasmettere un'immagine non compressa varia tra 162 e 270 Mbps; la tecnica di compressione si basa sulla riduzione, nel segnale video, delle ridondanze *spaziale, temporale e statistica*. Ciò è reso possibile dal fatto che il nostro occhio non è capace di percepire le variazioni ad alta frequenza, nel segnale a colori. La compressione si esegue in tre fasi: nella prima (riduzione della ridondanza spaziale) si riduce il numero di punti (pixel) componenti ogni quadro (il che abbassa la quantità di dati al secondo che deve transitare nel canale televisivo) giocando sul fatto che è possibile ricostruire un pixel compreso tra quelli adiacenti sulla base delle differenze tra questi ultimi. Ad esempio, se in una fila di tre pixel si verifica una variazione cromatica dal rosso al giallo, si può presumere che il pixel accanto al giallo sia verde o tendente ad esso. Questa tecnica si chiama "predizione" ed è simile all'interpolazione usata dagli scanner per aumentare la risoluzione o al ricampionamento adottato da programmi come Adobe Photoshop per incrementare le dimensioni dell'immagine e che si basa sul fatto di introdurre, in una trama "sgranata" di punti,

altri punti che hanno tonalità simili. Da questa operazione vengono ricavate le cosiddette immagini *Intra_frames* (I-frames). La seconda fase della compressione prevede la riduzione della ridondanza temporale, ottenuta con tecniche di compensazione di movimento che riescono a determinare il valore di un insieme di pixel di un frame a partire dall'informazione legata ai frame adiacenti.

La minimizzazione della ridondanza spaziale e di quella temporale fornisce le immagini dette *Predicted* ed i *frame Bidirezionali* (P o B-frames).

Quanto alla ridondanza statistica, si ottiene applicando regole matematiche di quantificazione basate sul comportamento dell'occhio umano nei riguardi delle variazioni di colore; ciò permette una riduzione significativa del data-rate, anche se l'utilizzo della Matrice di quantificazione provoca una perdita di risoluzione dell'immagine. I valori della matrice di quantificazione si possono variare per ottenere una maggiore o minore riduzione del data-rate, ottenendo rispettivamente una minore o maggiore qualità dell'immagine finale. Siccome la compensazione del movimento non è perfetta, non si possono creare lunghe sequenze di immagini nelle quali ognuna dipende dalle immagini precedenti o posteriori, poiché ogni errore si propagherebbe indefinitamente e si amplificherebbe; perciò per ottenere una compressione che restituisca immagini veritiere è necessario disporre delle tre immagini I, P e B. A seconda del tipo di applicazione e della complessità del codificatore, si potranno usare solo I-frame, I e P-frame, o I, P e B-frame. A seconda della concentrazione relativa di ognuna di tali

categorie si otterranno fattori di compressione più o meno elevati. Fin qui tutto ciò che abbiamo visto si riferisce alla compressione; va comunque detto che un decodificatore necessita di informazioni aggiuntive per poter ricostruire adeguatamente i frame. Per fornire queste informazioni e strutturare i dati secondo un'unica sintassi, lo standard MPEG-2 video definisce l'Elementary Stream (ES) che imposta una gerarchia di livelli all'interno di una sequenza, dove quello superiore determina il contesto in cui si definisce la sequenza (parametri video di base come le dimensioni dell'immagine, il rapporto base/altezza dell'immagine), il formato del campionamento, la frequenza d'immagine, il data-rate e le matrici di quantificazione. L'ES, non contiene informazioni di temporizzazione né di sincronizzazione, che permettano la ricostruzione della sequenza correttamente temporizzata. È qui che interviene l'MPEG-2 Systems, che fra le altre cose risolve il problema della sincronizzazione e definisce, a partire dall'Elementary Stream, nuove strutture più adatte alla trasmissione e all'immagazzinamento.

COMPRESSIONE AUDIO

L'audio della TV, soprattutto di quella in alta definizione che implica anche una qualità audio stereo a livello del Compact Disc, "consuma" un data-rate di 1,4 Mbps; con la compressione si può arrivare a soli 200 kbps (ossia sette volte meno) senza pregiudicare significativamente la qualità. La compressione del segnale audio sfrutta le caratteristiche psicoacustiche dell'orecchio umano, nel quale i toni ad alto livello sonoro tendono a sovrastare quelli a basso livello suonati insieme. Il criterio che si

Il trasmettitore DVB

applica è che non si codifica ciò che non si può sentire; la tecnica del caso è di tipo "lossy", cioè a perdita di parte dell'informazione, che poi è una perdita di ridondanza. Il processo di compressione parte dal segnale audio di frequenza compresa fra 0 Hz e 22 kHz, diviso in 32 sottobande di frequenza, ognuna filtrata, quantificata e codificata secondo le caratteristiche dell'orecchio umano.

Lo standard MPEG-2 Audio permette l'invio di 5 canali audio per realizzare l'effetto surround, e fino a 7 canali monofonici per consentire di trasportare il doppiaggio in più lingue selezionabili dall'utente (come avviene nei DVD). Esistono due tipi di codifica: una compatibile con l'MPEG-1 Audio e una non compatibile (AC3). Il segnale risultante è impacchettato insieme ai dati video per formare l'Elementary Stream audio.

IL PROGRAM SPECIFIC INFORMATION (PSI)

L'MPEG-2 System è la parte dell'MPEG-2 che stabilisce come comporre, a partire dagli Elementary Stream audio e video un pacchetto di dati trasportabile, detto TS packet (Transport Stream). Lo standard definisce un metodo per la descrizione dei contenuti dei TS packet finalizzato a semplificare e guidare i processi di demultiplex e presentazione dei programmi (un programma è un contenuto compresso in MPEG-2). Questo metodo si concretizza nel Program Specific Information (PSI) che è un insieme di quattro tavole che raccolgono le informazioni caratteristiche di un programma:

- *Program Association Table (PAT)*;
- *Transport Stream Program Map Table (PMT)*;
- *Network Information Table (NIT)*;

A breve pubblicheremo i progetti di due trasmettitori per digitale terrestre: uno a singolo canale ed uno a quattro canali. Entrambi forniscono il segnale TV pronto da inviare al trasmettitore di potenza o da applicare a un cavo d'antenna per giungere a TV e decoder. Lo stesso sistema potrà essere utilizzato per creare una TV via cavo ad esempio negli alberghi, dove permette di diffondere audiovisivi e contenuti proprietari o informazioni utili alla clientela, utilizzando un miscelatore per sovrapporli ai segnali d'antenna. La particolarità dei dispositivi è che

sono interamente programmabili, quindi potremo definire, ad esempio, il canale radio e i codici LCN dei singoli programmi ecc.



- *Conditional Access Table (CAT)*.

La PAT stabilisce la corrispondenza fra un programma ed il PID dove viaggiano i TS, ognuno dei quali deve contenere una PAT valida e completa e mai criptata. La PMT stabilisce una mappa di relazioni fra il programma e gli elementi che lo compongono. La CAT stabilisce l'associazione fra uno o più sistemi d'accesso condizionato, i relativi EMM stream e qualsiasi altro parametro speciale associato ad essi. I contenuti della NIT sono privati e non vengono specificati dall'MPEG-2 System; in generale riguardano le mappe dei servizi messe in relazione agli identificatori di TS, frequenze di canale, numeri di transponder, caratteristiche di modulazione, ecc. Insieme alle tavole PSI è possibile trasportare anche dati privati, strutturati nello stesso modo in cui lo sono le tavole PSI (Fig. 2). Se le informazioni trasmesse da queste ultime non sono sufficienti o si desidera inviare delle informazioni aggiuntive, si ricorre all'utilizzo di *descrittori (descriptor)*, i quali sono un insieme di strutture sintattiche che permettono il trasporto di informazioni aggiuntive in modo standardizzato. La

NIT si utilizza durante i processi di inizializzazione del decoder in quanto contiene informazioni sulla sintonizzazione.

Una trasmissione in DVB contiene, multiplexati, più TS, ognuno dei quali viene identificato tramite un *transport_stream_id* e ogni programma con un *program_number*; per decodificare un programma, il decoder dovrà conoscere entrambi. Farli conoscere è il compito del DVB-SI.

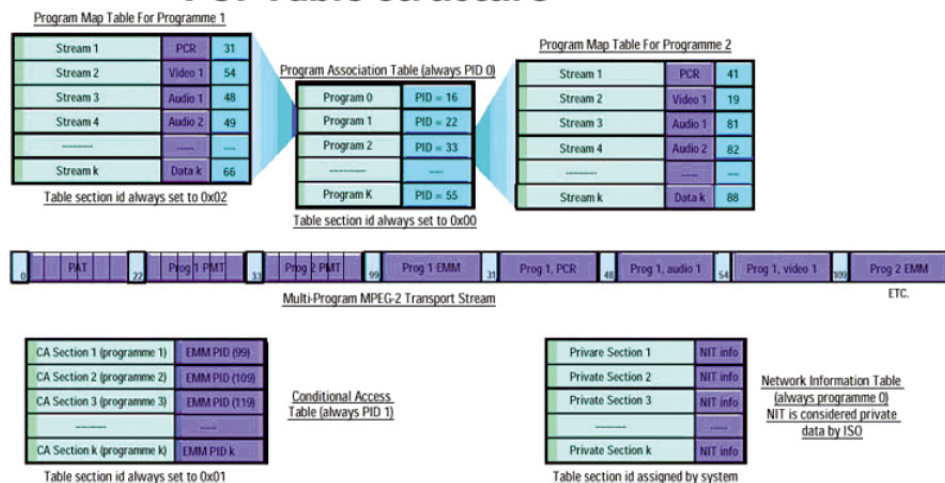
Attraverso la lettura della PMT otterrà i PID corrispondenti al video (PID=54), audio (PID=48,49..) i dati (PID=66) che compongono il programma ed il PID dei pacchetti che contengono il Program Clock Reference del programma (PID=31). Con queste informazioni il decodificatore sarà già in grado di selezionare i TS necessari a ricostruire gli ES che compongono il programma.

IL DIGITAL VIDEO BROADCASTING

Oltre a trovare uno standard di compressione audio/video, uno dei principali scopi del progetto DVB (Digital Video Broadca-

PSI Table structure

Fig. 2



sting) è stato definire tecniche di modulazione e metodi di codifica per la correzione degli errori, ma anche un algoritmo per i sistemi d'accesso condizionato, le informazioni che permettono allo spettatore un accesso facile e rapido al programma desiderato, l'inserimento di dati per applicazioni come i sottotitoli o il teletext. Tutto ciò si riassume nella DVB Service Information (DVB-SI) la quale consente al decoder (anche detto IRD, ossia *Integrated Receiver Decoder*) di riconoscere i servizi contenuti nei dati trasmessi, grazie alle tavole MPEG-PSI. La PSI fornisce informazioni che permettono la configurazione automatica del ricevitore ed il demultiplex e la decodifica di qualsiasi programma contenuto nel TS. Invece le tavole DVB-SI possono fornire anche informazioni sui servizi e sugli eventi trasportati da altri TS, ma anche su TS trasmessi da altri network. Ciò permette al decoder di commutare fra differenti TS in modo impercettibile dall'utente. La DVB-SI permette la sintonizzazione automatica del decoder a seconda del servizio selezionato, la localizzazione dei programmi, l'*Application Programming Interface* (API) ossia il sistema che fornisce la connessione fra le ap-

plicazioni software e l'hardware. Inoltre serve a creare la *Guida dei Programmi Elettronica* (EPG) cioè un'applicazione software creata dai provider di servizi che serve a presentare in modo gradevole e intuitivo tutti i servizi disponibili e agevolare la scelta del telespettatore. Quest'applicazione normalmente si presenta sotto forma di un menu formato da testo, immagini e persino videoclip. Un'altra funzionalità importante è il CA (*Accesso Condizionato*) che serve a definire quali sono i programmi di libero accesso e quelli cui si può accedere dietro identificazione dell'utente (servizi a pagamento). Prima di vedere nel dettaglio il discorso sulla DVB-SI, diciamo che nel DVB il canale televisivo viene chiamato *servizio*; un servizio può contenere una serie di programmi trasmessi sequenzialmente in TS con lo stesso program number (assegnato nella PAT). Ad esempio Canale 5 è un servizio del digitale terrestre. Viene chiamato *evento*, un programma TV all'interno di un servizio: per esempio un film o uno spettacolo o una partita. Vediamo il contenuto delle trasmissioni del digitale terrestre; ogni stringa di dati inviata su un canale RF (ossia su una

frequenza corrispondente ad uno dei canali un tempo della TV analogica) contiene i TS (audio e video compressi in MPEG-2) multiplexati al fine di contenere fino a 6 canali televisivi. Oltre a ciò, c'è tutto un set di dati necessari al decoder per classificare i programmi contenuti nella trasmissione, a distinguerli l'uno dall'altro (cioè ad identificare i canali televisivi multiplexati nel canale radio) a riportare i contenuti ecc.

Per ciascun canale TV (programma) viene trasmesso il segnale dell'LCN (Logical Channel Number) che è un codice con il quale il decoder posiziona un certo canale TV nella rubrica; in pratica l'LCN definisce per ogni canale programma a quale posizione di memoria corrisponde e quindi se si richiama dalla posizione 1, 2, 3 ecc. Per quanto riguarda il nostro Paese, a parte i principali canali, per gli altri non è stata ancora definita una numerazione completa, quindi può capitare che due canali abbiano lo stesso numero LCN o non l'abbiano; in tal caso viene chiesto all'utente di decidere quale canale manterrà la numerazione automatica. I numeri LCN certi sono quelli di RAI 1 (1) RAI 2 (2) RAI 3 (3) Rete 4 (4) Canale 5 (5) Italia 1 (6) e La7 (7): per questi, attivando la funzione LCN nel decoder, la numerazione sarà imposta automaticamente. I canali mancanti dell'identificativo LCN, vengono disposti dal decoder in ordine cronologico in base a quando vengono sintonizzati, quindi, dato che facendo la sintonizzazione automatica il decoder cerca a partire dalla più bassa banda VHF, ordina a partire dal basso; ciò implica che un canale che si trova sulla frequenza 526÷534 MHz (ex canale 28 UHF) viene posizionato in memoria prima di uno contenuto

nella banda tra 470÷478 MHz (ex canale 21 UHF).

Per ciascun programma, la stringa di dati contiene le informazioni sul nome dell'emittente, sulla programmazione (almeno quella in corso e la successiva) e sull'orario; il nome, il codice LCN e la programmazione sono contenuti che possono essere introdotti a piacimento da chi programma la stazione televisiva. A completare il set di dati contenuti nei singoli programmi, ci sono il Parental Control e l'accesso selettivo (CA) che serve a lasciar decifrare al decoder i soli canali liberi ed a bloccare la visione di quelli riservati alla pay TV. La DVB-SI è strutturata in nove tavole trasportate attraverso la struttura Private Section definita dall'MPEG-PSI. Esse sono:

- *Bouquet Association Table* (BAT); oltre a indicare il nome del bouquet, fornisce la lista dei servizi che lo compongono;
- *Service Description Table* (SDT); contiene dati che descrivono i servizi nel sistema, per esempio nome dei servizi, il provider di servizi, ecc.;
- *Event Information Table* (EIT); contiene il nome dell'evento, l'ora di inizio, la durata, ecc.;
- *Running Status Table* (RST); fornisce il flag di un evento, se si stia trasmettendo o no (running/not running) ed aggiorna questa informazione, oltre a permettere la commutazione automatica dell'IRD se questo era stato preventivamente programmato per mostrare un determinato evento;
- *Time and Date Table* (TDT);
- *Time Offset Table* (TOT);
- *Stuffing Table* (ST); si utilizza per validare/invalidare sezioni esistenti;
- *Selection Information Table* (SIT); si usa soltanto nei bitstream finiti (parziali) per esempio,

immagazzinati e trasporta un sommario dell'informazione SI richiesta per descrivere gli stream nei bitstream;

- *Discontinuity Information Table* (DIT); si usa solo nei bitstream finiti; si inserisce dove l'informazione SI del bitstream può essere discontinua.

Il bouquet del DVB è un insieme di servizi che hanno una tematica comune (sport, notizie, film ecc.) e viene definito dalla BAT; i criteri di raggruppamento li stabilisce l'emittente. Come già detto, un programma si identifica attraverso un program number che ha significato soltanto all'interno di un TS; il program number è un numero intero senza segno di 16 bit, pertanto possono esistere fino a 65.535 programmi unici all'interno di un TS (il program number 0 è riservato per la NIT). Quando sono disponibili più TS in un decodificatore (per esempio in un network via cavo), per demultiplexare un programma il decodificatore deve sapere sia il transport stream id che il program number all'interno del servizio desiderato.

Come nel PSI, ogni tavola DVB-SI è formata da varie sotto-tavole, ognuna delle quali è un insieme di sezioni che informano su uno stesso elemento (network, transport stream, servizio, ecc...) a seconda del tipo di sotto-tavola. Le sezioni sono mappate nei TS e possono avere una lunghezza massima di 1.024 byte, salvo le sezioni della tavola EIT, che possono arrivare fino a 4.096 byte. Le informazioni su servizi, eventi, network, ecc. risiedono nel payload della sezione. La struttura di questo varia in base alla tavola cui appartiene l'informazione, però quasi tutte le sezioni si basano sull'utilizzo di descriptor.

Un bouquet può raggruppare ser-

vizi di più di un TS, i quali possono essere anche trasportati in un network differente; in questo caso la NIT contiene informazioni utili al decoder per raggiungere questo network. Un descrittore del bouquet è il *Country Availability*, che indica al decoder per quali paesi è disponibile un certo bouquet; serve quando il segnale TV trasmesso può raggiungere più nazioni ed è utile per evitare la ricezione di trasmissioni, ad esempio, in lingua diversa da quella delle nazioni estranee alla trasmissione. Il descrittore può ricorrere due volte nella sotto-tavola della BAT: una per indicare la lista dei Paesi dove il bouquet è disponibile ed un'altra per indicare dove esso non è disponibile. Se ricorre soltanto una volta, ossia per indicare le nazioni dove il servizio è disponibile o non disponibile, il decoder intende che per qualsiasi altro paese rispettivamente non è disponibile o è disponibile.

Due descriptori interessanti sono il *Multilingual Component* (trasmette una descrizione testuale di uno stream dell'evento in una o più lingue) ed il *Parental Rating* (si usa per stabilire una classificazione dei programmi basata sull'età o su qualsiasi altro criterio, per impedire ai bambini la visione di programmi a loro non adatti). Assieme a tutti gli altri dati, nella trasmissione in digitale è previsto il descrittore del segnale orario TDT (*Time and Date Table*) usato per sincronizzare il clock interno dell'IRD, trasmesso almeno ogni 30 secondi; l'orario è UTC, ossia quello universale (riferito all'ora di Greenwich). Quanto alla Time Offset Table, si basa su un descrittore che comunica al decoder la differenza di orario fra UTC e ora del luogo dove viene ricevuto il TS e permette il passaggio dall'ora legale a quella solare e viceversa. La tabella (RST) viene utilizzata

La compressione MPEG-2

Lo standard MPEG-2 consta di più parti, le più rilevanti delle quali sono: la ISO 13818-1 (specifica come combinare o multiplexare differenti tipi d'informazione multimediale in un solo Transport Stream affinché questo possa essere trasmesso o memorizzato) mentre la 13818-2 definisce la codifica del segnale video e la 13818-3 riguarda la codifica audio. La sezione 13818-5 definisce quali sono i software di simulazione corrispondenti alle prime tre parti, mentre la 13818-6 (Digital Storage Media Command and Control, DSM-CC) specifica protocolli per gestire l'interazione degli utenti con bitstream MPEG-1 o MPEG-2 immagazzinati in DSM. Importante, ai fini della TV digitale, è la specifica 13818-9 Real Time Interface (RTI) che definisce un'interfaccia real-time fra l'adattatore del canale di

trasmissione e il decodificatore del Transport Stream dei sistemi MPEG-2. Vediamo adesso le specifiche MPEG-2 video: la compressione si basa sul fatto che alcune informazioni del segnale video possono essere omesse perché ridondanti, in quanto il nostro occhio non riesce a percepire le variazioni ad alta frequenza nel segnale a colori. La tecnica di compressione si basa sulla riduzione, nel segnale video, delle ridondanze spaziale, temporale e statistica. La ridondanza spaziale e statistica sono minimizzate attraverso l'utilizzo congiunto della Trasformata discreta del coseno (DCT), della Matrice di quantificazione, della lettura a zig-zag e delle codifiche RLC e di Huffman (VLC). Il risultato di questo processo sono le immagini *Intra_frames* (I-Frames) in cui la compressione si ottie-

ne grazie alla predizione dei valori dei pixel a partire dai valori dei pixel adiacenti. La ridondanza temporale viene minimizzata con l'utilizzo di metodi di compressione e di compensazione di movimento, che riescono a determinare il valore di un insieme di pixel di un frame a partire dall'informazione legata ai frame adiacenti. La riduzione combinata delle tre ridondanze dà le immagini dette Predicted ed i frame Bidirezionali (P,B-frames). Cerchiamo di spiegare come avviene la riduzione della ridondanza spaziale: dopo aver digitalizzato i segnali di luminanza e cromaticanza, si divide ogni quadro in blocchi di 8x8 pixel per ridurre il tempo di elaborazione; applicando la DCT ad ogni blocco si ottengono matrici di 64 coefficienti, i quali rappresentano le componenti delle frequenze spaziali che vengono organizzate nella

matrice in modo che nell'origine (angolo superiore-sinistro) ci sia la componente DC, ossia la frequenza zero, corrispondente all'inizio del tracciamento di un frame. L'asse X indica l'aumento della componente della frequenza orizzontale e l'asse Y l'aumento della componente della frequenza verticale. Questa organizzazione (**Fig. A**) implica che i valori dei coefficienti diminuiscano rapidamente man mano che ci avviciniamo all'angolo inferiore destro (massima frequenza). Nell'esempio, soltanto con l'applicazione della DCT e l'organizzazione dei coefficienti, si passa da un blocco B/N di 8x8 valori (pixel), a 4 valori (coefficienti DCT) diversi da zero. Quanto alla ridondanza statistica, diciamo che la matrice di coefficienti DCT viene divisa in una Matrice di quantificazione basata sul comportamento dell'occhio umano (zero per il coefficiente DC, valori elevati per i coefficienti corrispondenti ad alte frequenze). Il risultato è un gran numero di zeri per i coefficienti delle alte frequenze. L'accumulo e la distribuzione degli zeri vengono sfruttati per realizzare una scansione a Zig-Zag in modo da ottenere il maggior numero di zeri allineati consecutivamente, all'entrata del codificatore RLC. La codifica RLC, sostituisce tutti gli zeri consecutivi, con il numero che indica la quantità di zeri consecutivi: 00000000->8.

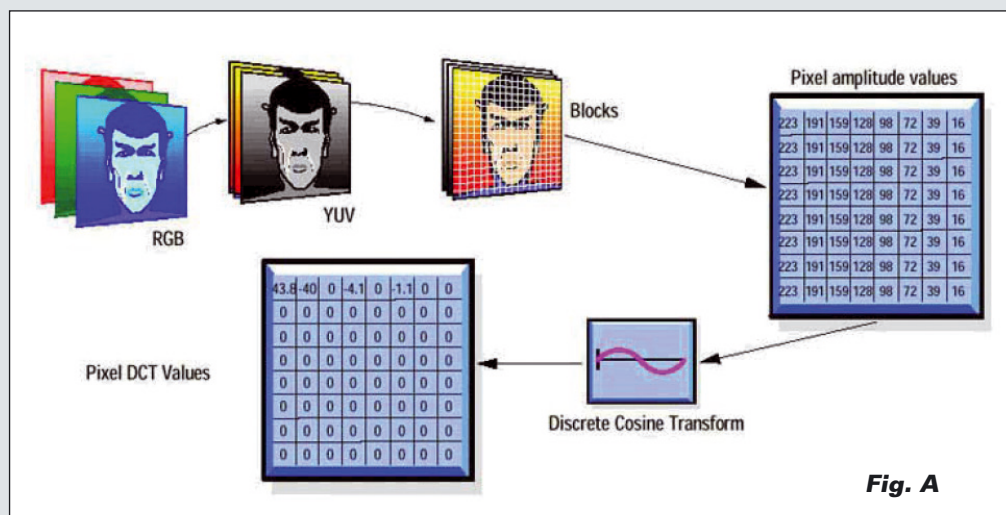


Fig. A

per aggiornare l'orario previsto indicato per un certo programma; a differenza di altre tavole SI che normalmente vengono trasmesse ripetitivamente, le sezioni RST si inviano soltanto una volta, nell'istante in cui l'orario dell'evento cambia. La funzione di questo meccanismo è permet-

tere ai decoder o registratori di avviarsi all'inizio dell'evento programmato.

ELECTRONIC PROGRAM GUIDE

Vediamo adesso in che modo i dati della DVB-SI vengono utilizzati dalla *Electronic Program Guide* e per capirlo facciamo un

esempio: ipotizziamo di voler vedere il canale a pagamento *Eurosport*; innanzitutto scegliamo il bouquet dei canali di sport tra quelli disponibili nel menu creato a partire dai *Bouquet_name_descriptors* estratti dalla BAT. Una volta scelto un bouquet, a partire dalla BAT associata e dal servi-

Alla fine, il codificatore di Huffman codifica quei coefficienti che si susseguono più frequentemente dal punto di vista statistico, in un numero di bit minore. Con tutto ciò si consegue una riduzione significativa del data-rate. Però l'utilizzo della Matrice di quantificazione provoca una perdita di risoluzione irreversibile dell'immagine ricostruita. Giunti a questo punto,

dopo aver eliminato la ridondanza spaziale e quella statistica, il risultato ottenuto è un I-frame. Se precedentemente si fosse eliminata la ridondanza temporale, il risultato sarebbe stato un B o un P-frame.

Vediamo adesso la ridondanza temporale, partendo dal concetto di macroblocco, che è l'insieme di 2 Blocchi di cromaticanza e 2 di luminanza, e confrontando due macroblocchi di frame; se sono uguali non si fa alcuna codifica e semplicemente si indica che i due sono uguali. Se i due macroblocchi sono differenti, nel caso di un P-frame, si controlla se il macroblocco esiste in una posizione differente nel I o P-frame precedente, mentre nel caso di un B-frame, si controlla se il Macroblocco esiste nel I e/o nel P-frame precedente e successivo. Se è così, si trasmettono vettori di movimento per indicare il suo spostamento.

Infine, solo se è un ma-

croblocco nuovo, viene codificato come qualsiasi Macroblocco di un I-Frame. È In questo processo che si consegue il massimo fattore di compressione. Siccome la compensazione di movimento non è perfetta, non si possono creare lunghe sequenze di immagini nelle quali ognuna dipende dalle immagini precedenti o posteriori, poiché ogni errore si amplificherebbe. Da ciò nasce la necessità di tre tipi di immagine I, P e B.

Gli Intra frame (I) sono codificati senza alcun riferimento ad altri frame; contengono tutte le informazioni dell'immagine e sono il punto di partenza per la creazione degli altri tipi di immagine, che derivano da essi. Senza la trasmissione periodica di questi frame, il sistema potrebbe degradarsi rapidamente. Come minimo ci deve essere un I-frame ogni 12 immagini, per evitare la propagazione indefinita di qualsiasi errore. I Predic-

ted frames (P) sono frame predetti a partire da un I o P frame precedente; infine, i Bidirectional frames (B) sono quelli che permettono la massima compressione: sono ottenuti dall'interpolazione fra due frame di tipo I o P precedenti e successivi. A seconda del tipo di applicazione e della complessità del codificatore, si potranno usare solo I, I e P, o I, P e B-frames. La concentrazione relativa di ogni tipo di frame determinerà il fattore di compressione.

Fin qui tutto ciò che abbiamo visto si riferisce alla compressione; va comunque detto che un decodificatore MPEG-2 necessita di informazioni aggiuntive per ricostruire correttamente i frame. Per fornire queste informazioni e strutturare i dati secondo un'unica sintassi, lo standard MPEG-2 Video definisce l'Elementary Stream (ES) che impone una gerarchia di livelli all'interno di una sequenza, dove il livello superiore determina il contesto in cui

si definisce la sequenza, il formato del campionamento, la frequenza d'immagine, il data-rate e le matrici di quantificazione).

Nella **Fig. B** si possono vedere le differenti intestazioni (header) dove risiede l'informazione di sistema. Questi strati, organizzati nel loro insieme formano un Elementary Stream video, quindi possiamo considerare l'ES, come uno stream continuo di frame compressi, codificati insieme alle informazioni necessarie per la sua ricostruzione. Però l'ES, non contiene informazioni di temporizzazione né di sincronizzazione che permettano la ricostruzione della sequenza correttamente temporizzata. È qui che interviene l'MPEG-2 Systems, che fra le altre cose risolve il problema della sincronizzazione, e definisce a partire dall'Elementary Stream, nuove strutture più adatte alla trasmissione e all'immagazzinamento, cioè i TS (*Transport Stream*).

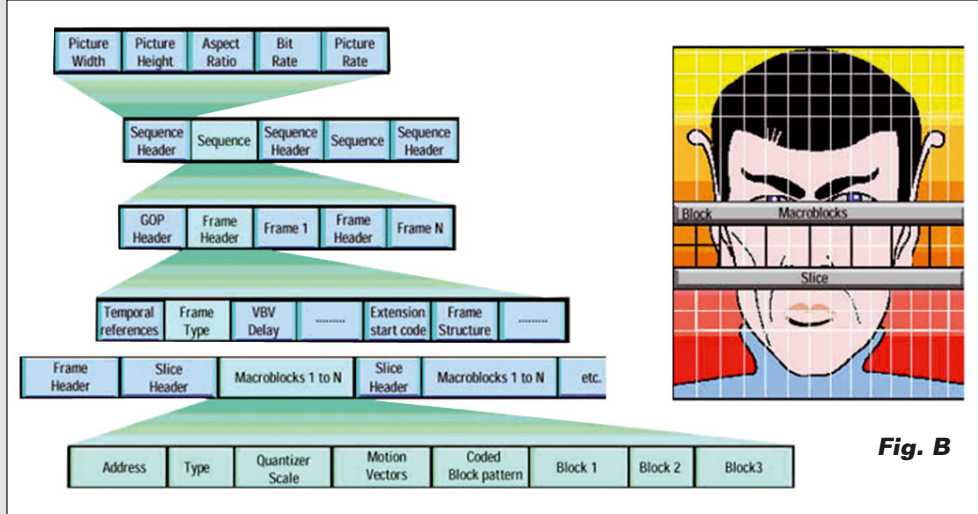


Fig. B

ce_list_descriptor, si visualizza un nuovo menu con i servizi disponibili; da qui selezioniamo il servizio a *Eurosport*. A questo punto la EPG deve accedere al TS che trasporta questo servizio. Per farlo, legge il *transport_stream_id* e quindi accede alla NIT. Nella NIT, cerca i descriptor associati

al *transport_stream_id* precedente e da essi trae le informazioni su frequenza e modulazione, necessarie al decoder per sintonizzare il TS che trasporta *Eurosport*. Un volta che l'IRD ha sintonizzato il TS, la EPG, a partire dal *service_id* (estratto dal *service_list_descriptor*) cerca i pacchetti con

PID=11 e *table_id*=42: tavola SDT del TS attuale. Con le informazioni contenute nella SDT associata al *service_id* precedente, la EPG comunica che il servizio è attivo, e se è presente la EIT Schedule, ci informa anche che la programmazione del canale è disponibile. A partire dal *mosaic_descriptor* la

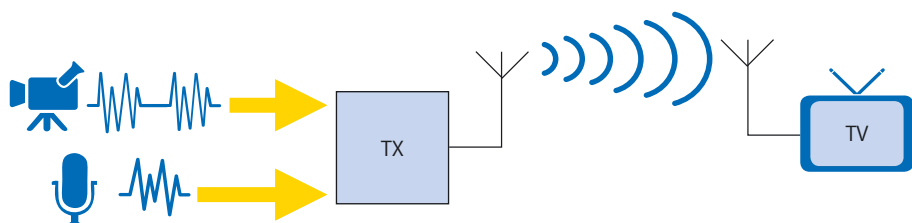
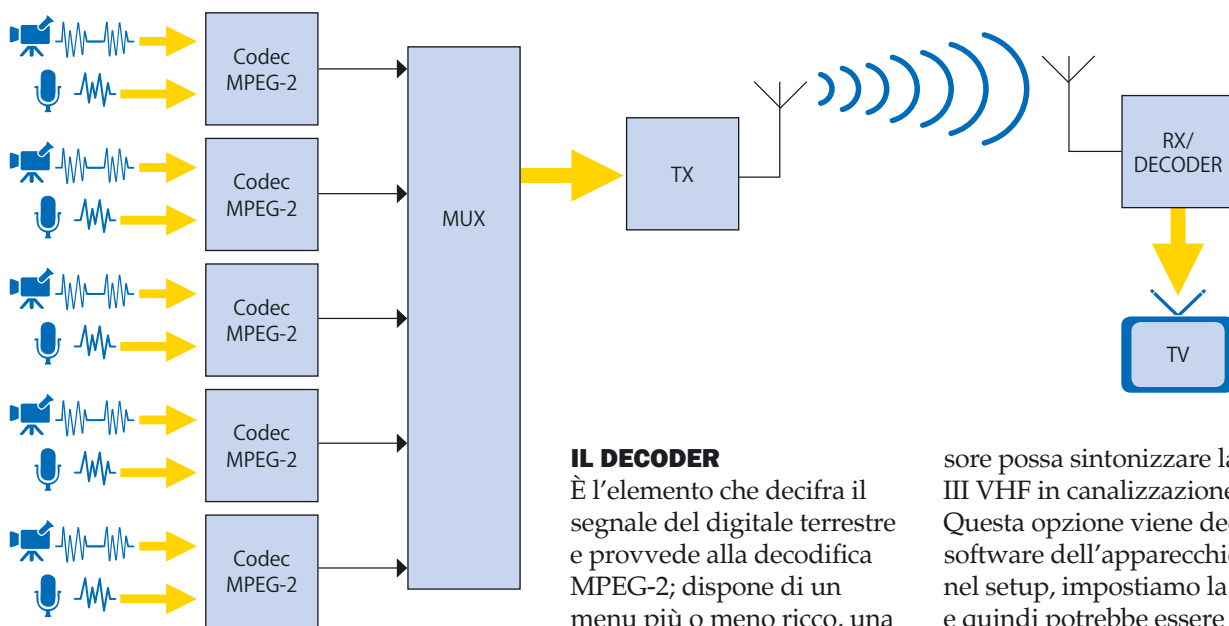


Fig. 3 - Schematizzazione del funzionamento della TV analogica (sopra) confrontato con quello del DVB (sotto).



IL DECODER

È l'elemento che decifra il segnale del digitale terrestre e provvede alla decodifica MPEG-2; dispone di un menu più o meno ricco, una delle cui funzioni è quella che permette di visualizzare,

per ogni servizio il PID, transponder o multiplexer d'appartenenza, canale RF ed altri parametri di quelli finora descritti. Alla schermata corrispondente si accede anche con il tasto INFO, solitamente presente nei telecomandi.

È importante scegliere con cura il decoder DTT o il televisore con decoder integrato. Nel caso in cui si decida di acquistare un *Set-Top-Box* (decoder esterno) bisogna preferire un prodotto che ha il bollino blu Dgtvi (Dgtvi è l'associazione per il digitale terrestre di cui fanno parte RAI, Mediaset, Telecom Italia Media e le associazioni delle emittenti locali) che consente, fra l'altro, l'accesso a servizi interattivi. Nel caso in cui si scelga un televisore con decoder integrato, è opportuno acquistare un apparecchio a bollino bianco Dgtvi. Occorre inoltre verificare che il decoder o televi-

sore possa sintonizzare la Banda III VHF in canalizzazione europea. Questa opzione viene decisa dal software dell'apparecchio quando, nel setup, impostiamo la nazione e quindi potrebbe essere necessario impostare una nazione diversa dall'Italia. Sul mercato esistono attualmente due tipi di decoder:

- il decoder non interattivo (*Zapper*) con cui si ricevono i consueti programmi televisivi;
- il decoder interattivo con cui, oltre a ricevere i programmi, si possono utilizzare i nuovi servizi disponibili con la TV digitale terrestre.

Quest'ultimo tipo va di norma collegato ad Internet, per interagire con i programmi.

I decoder più semplici constano di un tuner TV analogo a quello dei televisori e videoregistratori per la televisione analogica, di un demultiplexer, un microprocessore che lo governa ed un decoder MPEG-2 (Fig. 4). Vi sono però decoder più complessi in grado di offrire funzioni supplementari: a parte quelli interattivi, ci sono apparati dotati di porta USB, che permettono di leggere dalle

EPG ci può mostrare una schermata con una selezione degli eventi attuali e futuri del servizio. La EPG estrae l'*event_id* dal mosaic e cerca i pacchetti con PID=12 e con sezioni di table_id=4E: EIT Present/Following del TS attuale. Quando ha trovato la sezione EIT che corrisponde al *service_id* precedente, legge i descriptor associati all'*event_id* e visualizza orario di inizio, durata e informazione testuale relativa alla gara cercata. Una volta accettato di vedere questo evento, a partire dal *service_id* (è identico al *program_number* utilizzato nella PAT), si trova la PMT corrispondente e si avvia il processo di decodifica. Infine attraverso l'*original_network_id* (estratto dalla BAT) e il *telephone_descriptor* (ottenuto dalla SDT), la EPG può effettuare il pagamento dell'evento attraverso la connessione telefonica.

Pen-Drive file multimediali e riprodurre quindi filmati o anche semplicemente foto; ma non solo, perché molti funzionano anche da videoregistratori, in quanto consentono anche di registrare sulle penne USB i programmi attualmente in visione o altri programmi mentre stiamo guardando un canale TV. Bisogna comunque fare attenzione, perché il fatto che un decoder sia provvisto di USB non implica né che possa riprodurre i film (alcuni leggono solo file immagine JPEG ma non i filmati in AVI o DiVX) né che possa registrare programmi TV: sovente la presa USB è messa lì solo per effettuare gli aggiornamenti del firmware che alcuni costruttori rendono disponibili sul web.

COMPATIBILITÀ CON GLI IMPIANTI TV

Nella fase di switch-over, per non sopprimere di colpo la TV analogica si è cominciato a trasmettere il digitale sui canali RF liberi, convogliando su di essi le principali emittenti televisive raggruppate in transponder. Il passaggio al digitale terrestre avviene teoricamente in modo trasparente nei riguardi degli impianti, tuttavia in qualche caso possono verificarsi problemi: ad esempio negli impianti d'antenna multicanale, in cui i centralini coprono solo determinate bande o sono dedicati a certe frequenze.

In questo caso, dato che per il digitale terrestre si impiegano anche frequenze prima inutilizzate, può succedere che la ricezione dei canali TV convogliati su tali frequenze sia pessima, perché fuori della banda dei moduli amplificatori d'antenna. Occorre perciò cambiare moduli o rimuovere i filtri. Le antenne vanno sempre bene, anche se è abitudine di certi antenisti e fabbriche di antenne proporre antenne dedicate al

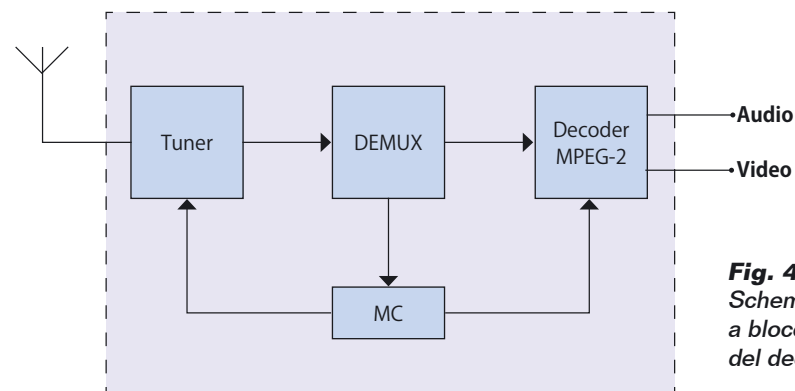


Fig. 4
Schema
a blocchi
del decoder DTT.

digitale: in realtà ciò non trova supporto in alcuna logica tecnica, perché l'antenna non ha riguardo per il tipo di modulazione. Casomai il problema è nell'estensione della banda dei centralini.

Il Mux 1 della RAI (il Mux principale, che include RAIUNO, RAIDUE e RAITRE) sarà quasi ovunque trasmesso in banda VHF, mentre gli altri Multiplexer RAI saranno diffusi in UHF.

Al momento dello switch-off, la banda III^A (VHF) passerà dalla canalizzazione italiana alla canalizzazione europea (Fig. 5).

Il canale mantiene l'ampiezza di 7 MHz, ma cambia nome e può essere spostato di frequenza. Il problema della ricanalizzazione sussiste per i canali E, F e G che non corrispondono più ai canali E6, E7, E8, e E9. Il canale D invece coincide esattamente con il canale E5, così come i canali H, H1 e H2 coincidono con i canali E10, E11 e E12 (quest'ultimo destinato al DAB/DMB).

Pertanto negli impianti centralizzati di tipo "canalizzato" (per esempio quelli condominiali), e comunque dove sia previsto un

filtro di canale, occorre adeguare la banda passante del filtro, altrimenti i multiplexer DTT vengono bloccati e non arrivano al TV o decoder.

Le reti di TV digitale sono tutte realizzate in tecnica SFN (Single Frequency Network) e, dunque, utilizzano una sola frequenza per ciascun multiplexer.

Ciò significa che, per una data frequenza di lavoro, gli "echi" in ingresso al ricevitore rafforzano il segnale principale cui si sovrappongono. Fa eccezione il Mux 1 della RAI, che è funzionante con la tecnica MFN (Multi-Frequency Network): pertanto, in fase di sintonizzazione del decoder o del televisore, può presentarsi per i canali televisivi diffusi sul Mux 1 (RAI Uno, RAI Due e RAI Tre) più di un segnale, proveniente da siti diversi. In questo caso, occorre assicurarsi che la sintonizzazione avvenga sulla frequenza migliore.

Bene, con questo abbiamo concluso; vi diamo appuntamento al prossimo fascicolo, dove presenteremo il nostro primo progetto di trasmettitore per DVB. ■

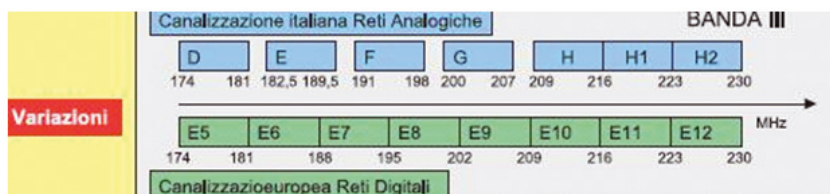


Fig. 5 - Canalizzazione italiana ed europea.

Dalla teoria alla pratica

Da alcuni mesi, lo avrete notato, stiamo proponendo la realizzazione di aeromodelli in grado di volare autonomamente: ormai la tecnologia, grazie anche all'impegno di molte comunità open-source, consente di avere a disposizione le risorse necessarie -hardware e software- a costi decisamente contenuti. Si tratta, in ogni caso, di prodotti molto complessi, così come complessa risulta l'integrazione su uno specifico velivolo. A tutto ciò bisogna aggiungere che quanti si vogliono cimentare in questo campo debbono avere già una buona dimestichezza con gli aeromodelli, sia per quanto riguarda le capacità di pilotaggio, sia per ciò che concerne le competenze di aerodinamica, meteorologia, norme, eccetera. Insomma, se già non è facile fare volare (bene) un aeromodello, è molto ma molto più difficile realizzare un velivolo in grado di volare (anche) in maniera autonoma. Un conto è la teoria, un altro è la pratica. È quanto abbiamo sperimentato anche noi col velivolo descritto in queste pagine che ha richiesto oltre sei mesi di sviluppo da parte di chi, oltretutto, vanta un'esperienza trentennale nel settore. Ma alla fine ci siamo riusciti. E tutto ciò ci riempie di gioia anche perché, a quanto ci risulta, questo è il primo velivolo del genere (di tipo amatoriale, ovviamente) ad aver volato nel nostro Paese. Anche i vari progetti sviluppati da alcune Università non hanno mai preso ... il volo. Speriamo che il nostro velivolo rappresenti uno stimolo anche da questo punto di vista.



IN QUESTO NUMERO

118 ■ EVENTI

Si conclude con la finale mondiale di Istanbul la stagione delle gare di robotica che abbiamo seguito, sostenuto e anche sponsorizzato. Siamo certi che i team italiani in gara dimostreranno tutte le loro capacità anche in campo internazionale.

120 ■ DRONE ZONE REALIZZIAMO UN UAV BIMOTORE

Costruiamo un aeromodello con capacità di volo assistito, basato sulla nuova piattaforma Ardupilot Mega. Sicuramente la prima realizzazione di tipo amatoriale di questo genere che ha volato nel nostro Paese. Prima parte.

134 ■ MONDO 3D COSTRUIAMO E SPERIMENTIAMO UNA STAMPANTE 3D

La realizzazione e l'utilizzo di una stampante 3D alla portata di tutte le tasche in grado di produrre oggetti in robusto ABS, sia come prototipi, sia direttamente utilizzabili in molte applicazioni.



LA SPEDIZIONE ITALIANA AI MONDIALI

RoboCup2011

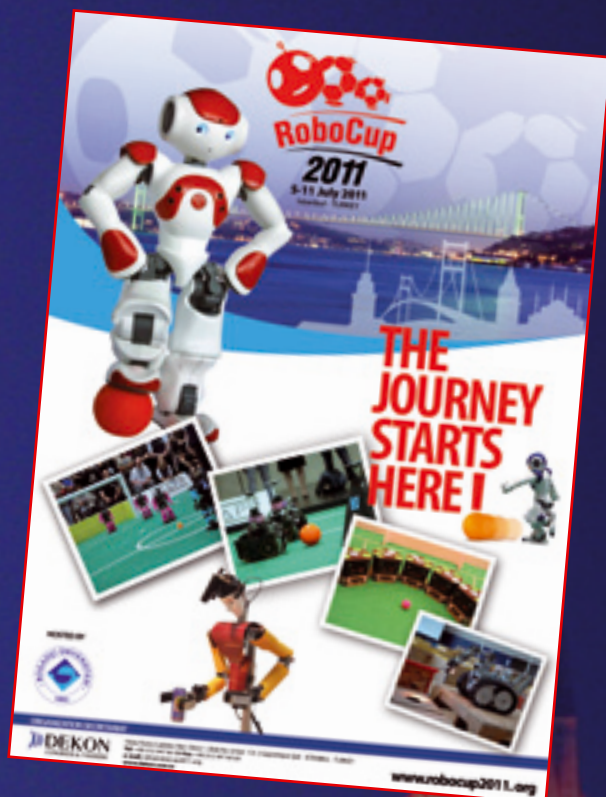
DI ISTANBUL

Dal 5 all'11 luglio Istanbul sarà la capitale mondiale della robotica, almeno di quella didattica, che ha come scopo la diffusione delle discipline legate alla robotica tra gli studenti delle scuole di ogni ordine e grado, dalle elementari all'Università. E cosa c'è di

più stimolante che mettere in competizione tra loro robot di ogni genere le cui capacità sono il frutto dello studio, delle conoscenze e della passione dei team di studenti che li hanno costruiti? L'obiettivo originale della RoboCup, quello di



di Arsenio Spadoni



realizzare entro il 2050 una squadra di robot umanoidi in grado di competere con la squadra vincitrice dei Campionati del Mondo di Calcio è ancora valido, anche se col tempo si sono aggiunti altri traguardi meno impegnativi (sia dal punto di vista tecnico che economico) per dare la possibilità agli studenti più giovani di prendere parte a questo tipo di manifestazioni. È così nata la RoboCup Junior riservata agli studenti fino a 19 anni (Under 19, denominata Secondary), tipicamente gli studenti della nostra scuola media superiore, e la categoria Under14, denominata Primary, solitamente gli studenti della nostra scuola media inferiore. Questi team si confronteranno nelle categorie Dance, Soccer e Rescue. Le scuole italiane che parteciperanno alla finale mondiale della RoboCup2011 di Istanbul si sono qualificate in virtù delle selezioni italiane che si sono svolte in marzo a Roma (RomeCup, organizzata dalla Fondazione Mondo Digitale) e in aprile a Catania (Robocup Junior Italia, organizzata dalla Robocup Jr ITALIA). Sono una dozzina i team italiani che

parteciperanno alle finali mondiali in Turchia: i due gruppi più folti sono rappresentati dalle squadre romane impegnate prevalentemente nelle gare di soccer e da quelle del Triveneto che prenderanno parte alle gare di Rescue. Di seguito l'elenco completo delle squadre italiane.

RESCUE:

- Anacleto, dell'IIS Scalcerle di Padova;
- Skinwires, dell'ITIS Chilesotti di Thiene (Vicenza);
- Ulisse, dell'ITI "Michelangelo Buonarroti" di Trento;
- Robotik, (under 14) della Scuola Media Statale Mario Pluchinotta (CT).

DANCE:

- T4AT, dell'ITIS "A.Rossi" di Vicenza;
- Marty, dell'ITI "Archimede" di Catania;

SOCCER:

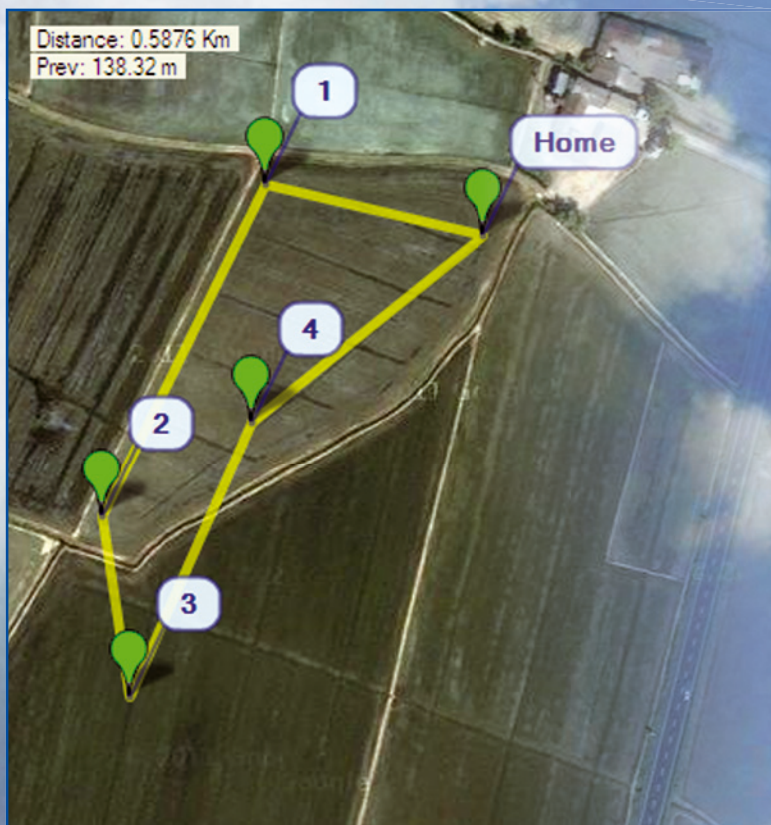
- Cardano Robotic Team, dell'IS Piazza della Resistenza di Monterotondo (Roma);
 - Leonardo, dell'Itis Da Vinci di Arcidosso (GR);
 - Manetti, dell'Itis Da Vinci di Arcidosso (GR);
 - SPQR Levis, degli Itis Von Neumann e Galilei di Roma;
 - SPQR Gravis; degli Itis Von Neumann e Galilei di Roma;
 - SPQR Magis degli Itis Von Neumann e Galilei di Roma;
- Le squadre romane sono delle veterane, avendo già partecipato ad alcune finali mondiali; non così le squadre del Triveneto che però hanno dalla loro una grinta formidabile. Per la prima volta partecipa alla manifestazione una squadra Under 14 (Robotik) mentre, come al solito, brillano per la loro assenza le Università italiane, ad eccezione di un team italo-cileno. □



REALIZZIAMO UN UAV BIMOTORE

di MARCO MAGAGNIN

**Costruiamo un aeromodello
con capacità di volo assistito,
basato sulla nuova
piattaforma Ardupilot Mega.
Prima parte.**



Se ci seguite assiduamente avrete certo notato che ultimamente ci siamo cimentati nella modifica di aeromodelli RC al fine di dotarli di un sistema di volo assistito integrato da una stazione di controllo a terra implementata da un PC ed un apposito software; i progetti fin qui proposti permettono la stabilizzazione del volo di un Easy Star, aeromodello molto semplice ma dalle ottime caratteristiche di volo. Il sistema di controllo del velivolo adottato è Ardupilot, il cui studio ci ha permesso di acquisire le conoscenze e competenze atte a realizzare, configurare, utilizzare ed ottimizzare un modello di aereo dotato di un sistema di controllo della stabilizzazione e della navigazione. In queste pagine vi proponiamo un progetto più sofisticato, che rappresenta un notevole salto di qualità rispetto al precedente, in quanto adottiamo come sistema di controllo il più prestante ArdupilotMega, con il suo corredo di software e di moduli complementari, sempre sviluppato dal gruppo di lavoro DiyDrones. L'integrazione della famiglia di prodotti ArdupilotMega con i moduli di telemetria e



con le applicazioni software di assistenza al volo (Ground Station) e di configurazione (Mission Planner), insieme ad altre che vedremo in seguito (come l'Antenna Tracker e l'On Screen Display) ci permetteranno di costruire ed imparare a gestire un "insieme di sistemi" di elevata qualità, dalle caratteristiche molto simili a quelle di sistemi professionali dal costo inavvicinabile. La piattaforma ArdupilotMega (che da qui in poi, per brevità, chiameremo APM) sviluppata in tecnologia ArduinoMega, è disponibile sia con processore ATmega1280 che ATmega 2560. Per ora il software disponibile non è stato adattato per funzionare con Atmel2560, quindi nel caso volesse intraprendere la realizzazione direttamente con tale piattaforma, verifica prima sul sito Internet <http://diydrones.com> lo stato di avanzamento del progetto, perché gli sviluppi procedono piuttosto velocemente.

APM e i suoi moduli associati sono adatti all'impiego su diverse tipologie di modelli di aereo, sia a due che a tre assi e su modelli

ad ala rotante; per ciascuna tipologia è disponibile una specifica versione di software, denominata ArdupilotMega per i modelli di aereo e Arducopter per i modelli ad ala rotante. Le caratteristiche principali della piattaforma APM sono:

- possibilità di gestire diverse configura-

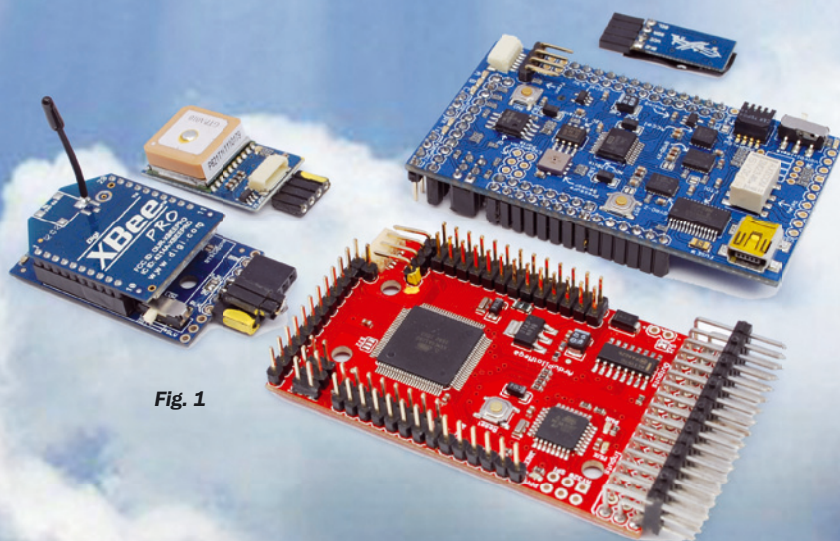


Fig. 1

Fig. 2



Fig. 3

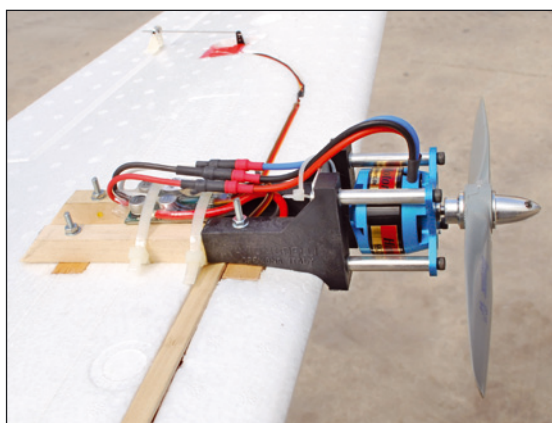


Fig. 4

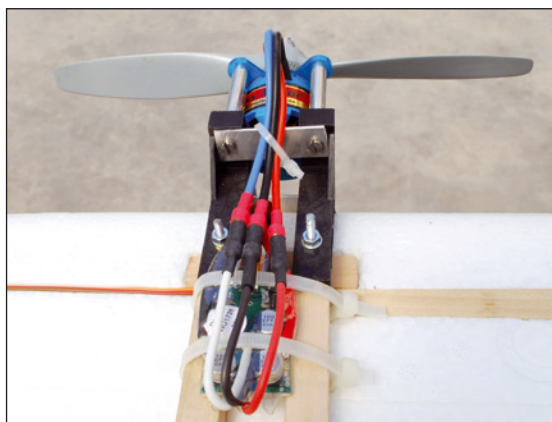
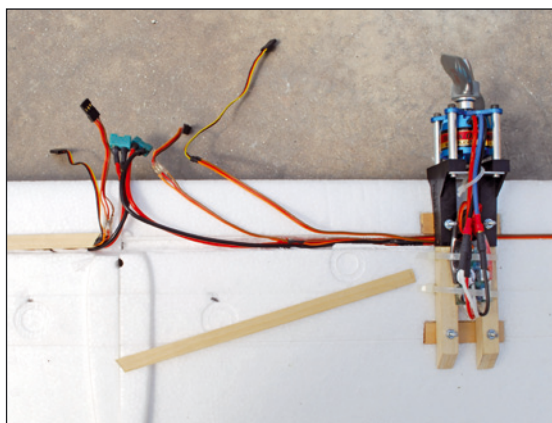


Fig. 5



zioni di modelli a due e a tre assi, controllando timone di profondità, timone di direzione, alettoni e motore;

- capacità del software di determinare automaticamente la configurazione hardware e la tipologia dei componenti collegati; ad esempio è in grado di riconoscere automaticamente quale tipo di GPS è collegato e con quale tipologia di firmware è configurato;
- possibilità di integrare e configurare nuovi sensori come il magnetometro;
- possibilità di sfruttare appieno la telemetria bidirezionale, basata sul protocollo Mavlink, e quindi di impartire comandi e modificare le modalità di funzionamento del software da remoto;
- nuove modalità di controllo di volo, come Auto take-off e Auto landing;
- disponibilità di un registratore dei dati di volo per conservare i dati di una missione e poterli analizzare in differita.

Oltre a ciò, la piattaforma offre l'applicazione software di configurazione Ardupilot Mega Planner, la quale permette di:

- caricare il software su APM senza necessità di compilazione e quindi senza la obbligatorietà di utilizzare l'IDE di sviluppo di Arduino;
- possibilità di modificare i parametri di configurazione e di calibrare i controllori PID on line, anche in volo;
- possibilità di impostare le caratteristiche di una missione in modo grafico e di caricarla nel sistema.

Per questo progetto abbiamo scelto il modello di aereo Mentor prodotto dalla Multiplex, caratterizzato da ottime doti di volo e da un volume e una capacità di carico decisamente superiori a quella dell'EasyStar. Si tratta di una semiriproduzione, in EPP (polipropilene espanso) termoformato, di un aereo da turismo ad ala alta con carrello bicipite e ruotino posteriore controllato dallo stesso comando del timone verticale. Le ali hanno un'apertura di 160 cm e sono dotate di alettoni.

Nel nostro progetto abbiamo previsto, oltre all'elettronica di controllo, l'installazione di una telecamera in grado di trasmettere le immagini a terra.

Per evitare la vista dell'elica nelle riprese della telecamera e per disporre del maggior spazio possibile per i componenti all'interno della fusoliera, abbiamo deciso di sostituire il motore posizionato nel muso dell'aereo con una coppia di motori installati sulle ali. Con questa modifica la telecamera può essere montata anteriormente, nello spazio lasciato libero dal motore originario.

Abbiamo scelto il Mentor in scatola di montaggio, nella versione "tutta da montare", in modo da poter eseguire le modifiche che ci necessitano prima di assemblare e incollare le parti. Il materiale occorrente a realizzare il nostro UAV è:

- una trasmittente RC funzionante su una frequenza omologata per modellismo con emissione PPM per il controllo dei servocomandi;
- una ricevente compatibile con la trasmittente, sempre con uscita di segnale PPM verso i servocomandi.
- una batteria LiPO 3 celle da 3.200 mA o 4.000 mA;

Per la motorizzazione abbiamo due scelte:

- se volete rispettare il modello originale che prevede un solo motore montato nel muso dell'aereo, vi occorre il kit di motorizzazione della Multiplex cod. 33 2641 che comprende il motore Brushless Himax C 3528-1000 da 450 watt (cod. 33 3028) il regolatore di giri MULTIcont BL-54 cod. 7 2277, l'elica da 11x5,5 (cod. 73 3179) e il supporto per l'elica con mozzo da 5 mm (cod. 33 2326);
- in alternativa, se preferite la versione bimotore, sono necessari due motori Himax Outrunner HC3510-1540 da 250 watt (cod. 33 3021) due regolatori di giri Multiplex Multicont BL37/II cod. 7 2276 e due eliche Graupner 8x6, una destra e una sinistra.

Per la parte elettronica abbiamo utilizzato (Fig. 1):

- un modulo ArduPilot Mega - Arduino Compatible UAV Controller w/ ATmega1280;
- un modulo ArduPilotMega IMU Shield/OilPan Rev-H;

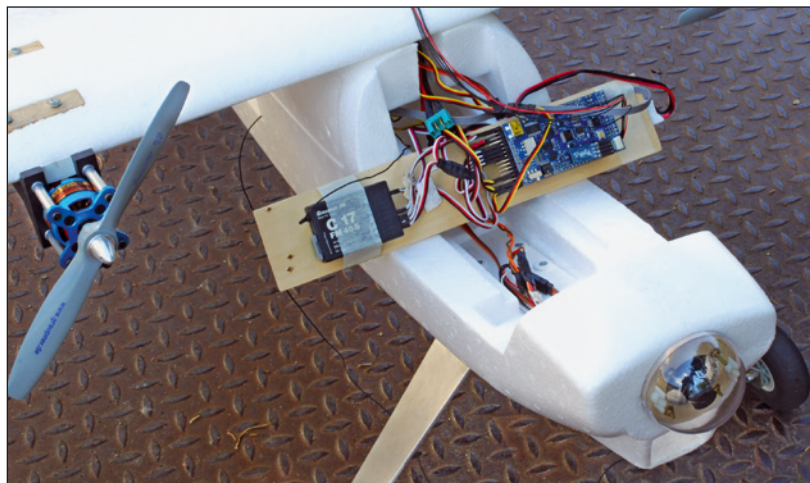
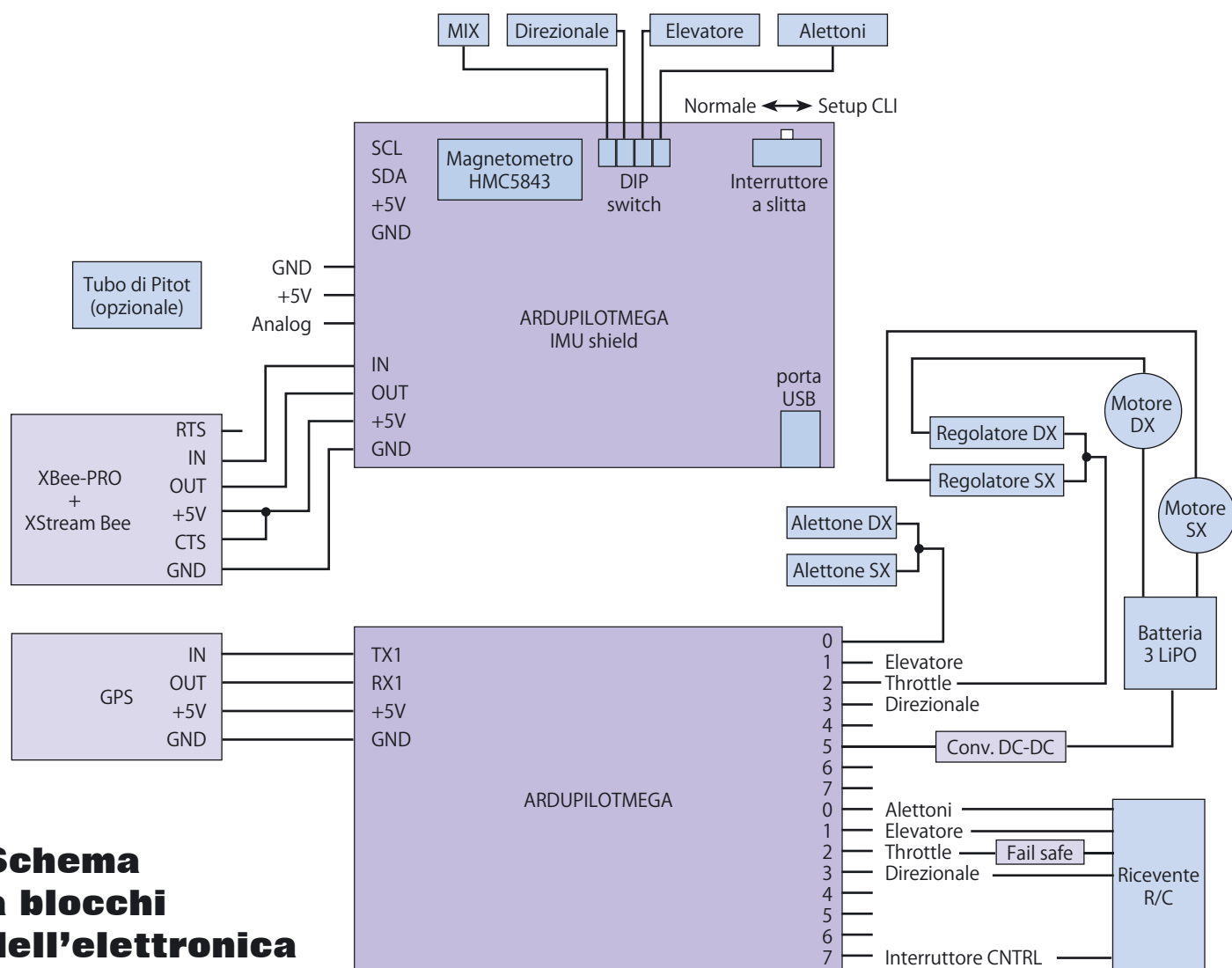


Fig. 6

- un modulo GPS mediatek (MediaTek MT3329 GPS 10Hz+Adapter Basic) con relativo cavo di prolunga da 25 cm (EM-406/uBlox/MTK Adapter Cable 25 cm);
- un magnetometro HMC5843 - Triple Axis Magnetometer Rev 1.2;
- un modulo FTDI USB (FTDI Basic Breakout - 5 V Cod. Sparkfun DEV-09716) per la programmazione del modulo GPS con il PC;
- un modulo convertitore di tensione DC-DC cod. Futura Elettronica 7300-BOB09370) con una resistenza di trim da 1,34 kohm (nel caso non trovaste il valore preciso utilizzatene uno leggermente più basso);
- modulo USB per la telemetria lato PC XBee Explorer USB cod. Sparkfun: WRL-08687);
- modulo FTDI per la telemetria lato aeromodello (XtreamBee Board);
- due moduli XBee PRO lato PC e lato modello per la trasmissione e ricezione della telemetria (XBee Pro - Wire Antenna cod. Sparkfun WRL-08742); sono possibili altre combinazioni di moduli XBee su frequenze diverse e con antenna cablata o esterna (resta inteso che i due moduli devono essere compatibili tra di loro);
- un modulo Intelligent Failsafe (Fusion FS-01) da interporre tra la ricevente e il canale Throttle dell'APM in modo da poter gestire da software la condizione di perdita del segnale radio.

Procediamo ora con la realizzazione, partendo dalle modifiche da apportare al mo-

Schema a blocchi dell'elettronica



dello: per montare i due motori sulle ali abbiamo costruito due gondole a partire da altrettanti supporti per motori a scoppio modificati (Fig. 2). I supporti orizzontali sono stati prolungati con dei listelli di legno a sezione quadrata (Fig. 3 e Fig. 4) fissati con prigionieri filettati da 2,5 cm di lunghezza (ricavati da una barra filettata di diametro 3 mm o da spezzoni di viti). I prigionieri sono avvitati, previa filettatura, per metà nel listello di legno e per metà nel supporto in nylon e poi incollati con colla cianoacrilica per EPP. La parte superiore del supporto è stata sagomata per seguire il profilo dell'intradosso del bordo d'attacco dell'ala.

Due placchette distanziali in alluminio tengono in squadra il supporto del motore, fissato a quest'ultimo con la piastra frontale,

le colonnine e le viti originali (incluse nella confezione del motore). Per fissare i supporti alle ali sono stati annegati nel profilo e incollati, sempre con la colla ciano per EPP, quattro rinforzi ricavati da listelli di legno a cui i supporti stessi sono fissati con viti passanti.

Prima di incollare i semigusci della fusoliera è opportuno creare un passaggio per i cavi della telemetria e della trasmittente video tra il vano della cabina e l'alloggiamento originariamente previsto per il servo destinato al gancio di traino degli alianti, nella parte posteriore della cabina; infine bisogna sagomare l'alloggiamento in modo da potervi alloggiare la trasmittente della videocamera.

Poi occorre ricavare due scanalature nelle semiali, allargando quelle già predisposte

per i cavi dei servocomandi degli alettoni, in modo da poter alloggiare anche i cavi dei regolatori di giri (cavo di segnale più cavo d'alimentazione). Se si realizzano le scanalature con una certa attenzione è poi possibile chiuderle con listelli a filo fissati con nastro adesivo.

Data la quantità di cavi è poi necessario allargare la scanalatura sul davanti della fusoliera, per facilitare il passaggio dei cavi stessi all'interno della cabina (**Fig. 6**). A questo punto è possibile continuare l'assemblaggio del modello seguendo le istruzioni del costruttore.

Nel caso abbiate scelto la configurazione bimotore, non incollate, nel muso dell'aereo, le quattro longerine laterali ed il supporto frontale del motore, in quanto non sono più necessari.

Per alloggiare la telecamera è stato realizzato un supporto circolare in legno compensato da 3 mm.

Ai lati dell'apertura frontale sono stati incollati due spessori in EPP (recupero delle modifiche precedenti) in modo da permettere il fissaggio del supporto della telecamera con due viti incollate ad essi. La telecamera è fissata al supporto circolare in compensato con quattro viti da 2 mm di diametro e la sua inclinazione è regolata mediante spessori infilati sulle viti ricavati da un tubetto bowden.

La semisfera trasparente a protezione della telecamera è ricavata da una pallina per albero di Natale montata a pressione (**Fig. 6**). Nella versione bimotore i regolatori di tensione sono montati con due fascette agli stessi supporti dei motori.

Nella nostra realizzazione abbiamo optato per la configurazione dei motori controrotante, in modo da azzerare la coppia generata dalla trazione, con risultati decisamente soddisfacenti. Questa soluzione richiede l'impiego di un'elica destrorsa e di un'elica sinistrorsa (**Fig. 7**).

Il collegamento dei cavi dei servocomandi degli alettoni e dei cavi di controllo dei regolatori di tensione sono realizzati con cablaggi ad "Y", in modo da impegnare un solo canale della ricevente ciascuno. Per il cavo ad "Y" proveniente dai regolatori dei motori, abbiate l'accortezza di eliminare il conduttore del positivo (colore rosso cen-

trale) in quanto la tensione necessaria al funzionamento di tutta l'elettronica viene fornita dal convertitore switching DC-DC. Per "tradizione" aeronautica, guardando l'aereo da dietro, il motore destro gira in senso orario ed il sinistro in senso antiorario.

Fate la prova senza montare le eliche; se necessario, il cambio del senso di rotazione di un motore brushless si ottiene scambiando il collegamento di due qualsiasi dei tre cavi che collegano il regolatore di giri al motore (prima staccate la batteria). Nel caso abbiate difficoltà a reperire le eliche controrotanti, utilizzate pure i motori configurati per girare nello stesso senso, al più sarà necessario un po' di trim per la correzione dell'imbardata dovuta alla coppia dei motori.

Una volta terminata la costruzione del modello, soprattutto nel caso della configurazione bimotore, è opportuno eseguire qualche prova in volo prima di installare l'elettronica e la telecamera, in modo da conoscere il comportamento del modello quando avrà a bordo il suo carico prezioso. Prestate particolare cura nel "centraggio" del modello, soprattutto nella configurazione bimotore, spostando la batteria all'interno della fusoliera fino a trovare la posizione ottimale.

Verificate che i comandi funzionino nel verso giusto; in caso contrario utilizzate la funzione "reverse" sulla trasmittente. Accertatevi che i motori ruotino nel senso corretto e montate le eliche utilizzando i mozzi di alluminio inclusi nelle confezioni di ciascun motore; ricordate di montare il mozzo sull'albero motore, infilare l'elica sul

Fig. 7



mozzo nel verso giusto, ingrassare leggermente la parte sporgente della filettatura del mozzo ed infine avvitarlo l'ogiva serrando senza esagerare (ma abbastanza da evitare lo svitamento in volo).

La funzione del grasso è evitare il "grip-paggio" dell'ogiva sulla filettatura, caratteristica antipatica degli accoppiamenti in alluminio, che impedisce di smontare l'insieme senza romperlo.

Infine, un'ultima avvertenza sulla configurazione bimotore: in decollo, a causa del momento torcente dovuto al braccio di leva esistente tra il vettore della trazione, a livello della corda alare, ed il vettore re-

sistenza, a livello del suolo, se date tutta potenza il modello tende a "capottare".

Con questa configurazione, come in aeronautica vera, il decollo deve essere eseguito cabrando a fondo il modello ancora fermo e poi iniziando la corsa accelerando progressivamente, quindi rilasciando il comando e cabrando man-mano che il modello prende velocità. Con un po' di pratica e sensibilità si otterranno decolli in pochi metri molto realistici ed armoniosi, utilizzando solo una parte della potenza disponibile.

In volo, invece, dato che la trazione è pressoché in asse con la direzione di volo, il modello non presenta alcuna tendenza a variare l'incidenza al variare della velocità, facilitando molto anche il compito del software di stabilizzazione.

Una volta terminata la cellula, iniziamo ad assemblare l'avionica, che installeremo poi sul nostro modello; tutti i circuiti elettronici saranno interconnessi come mostrato nello schema a blocchi.

Vale la pena di porre l'accento su due componenti particolarmente importanti presenti sul modulo ArdupilotMega IMU Shield: l'interruttore a levetta posto in prossimità del connettore a pettine di collegamento alla ricevente ed ai servocomandi (Fig. 8); la matrice di interruttori dip-switch presenti subito dietro il primo interruttore. L'interruttore a slitta serve per impostare il modo di funzionamento del software di controllo dell'APM in modo Setup (con la levetta rivolta verso il connettore a pettine) dove viene attivata la CLI (interfaccia a linea di comando) necessaria alla gestione "materiale" del modulo, oppure in modalità normale (levetta spostata nell'altro senso) che è quella in cui l'APM gestisce il velivolo.

Quanto ai dip-switch, servono per gestire il modo normale o misto (MIX) dei comandi di elevatore ed alettoni come in una configurazione tutt'ala, e per il reverse dei diversi comandi singoli di timone elevatore, direzionale ed alettoni.

Preparazione dei componenti hardware

Armati di saldatore e di molta pazienza, si comincia ad assemblare il complesso centrale del sistema composto dal modu-

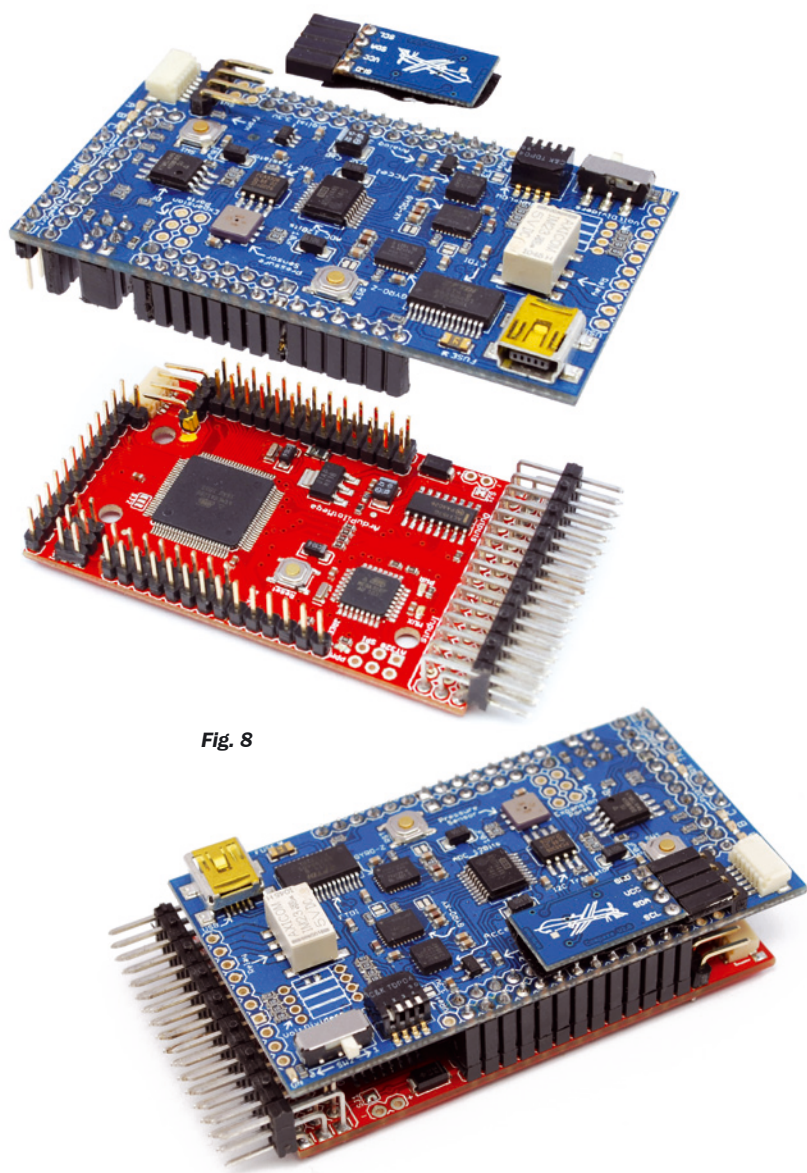


Fig. 8

lo APM e dall'ArduPilot Mega IMU Shield; conviene inizialmente tagliare a misura e posizionare tutti gli strip di pin assemblando i moduli a secco, verificare e riverificare, tenendo i moduli in posizione con dei pezzi di carta da mascheratura, quindi puntare con il saldatore i pin sugli angoli per rendere solido l'assemblaggio e poi saldare tutti gli altri. Saldare i pin a 90 gradi per collegare il GPS, il magnetometro, la telemetria e l'eventuale sensore di velocità, quindi modificate il modulo GPS per poter aggiornare il firmware e collegarlo al modulo APM. Il modulo APM richiede, per funzionare correttamente, che sul GPS sia installata la versione di firmware 1.4 oppure 1.6.

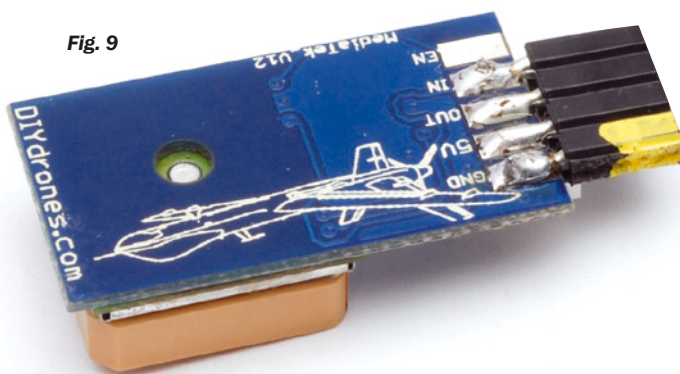
Per modificare il firmware del GPS è necessario saldare un connettore femmina a quattro poli sulle piazzole, al di sotto del modulo GPS, che portano all'esterno alcuni dei segnali della seriale del modulo stesso; abbiamo approfittato di questa modifica anche per collegare, tramite questo connettore, il GPS al modulo APM, con un cavo un poco più robusto di quello in dotazione (**Fig. 9**).

Saldare le strip di pin al modulo XstreamBee (per il collegamento al modulo FTDI utilizzate il connettore femmina incluso nella confezione) e poi aggiungete un cavallotto tra i piedini CTS e +5V del connettore FTDI, per consentire il funzionamento in streaming del modulo XBee; nel nostro caso abbiamo saldato un connettore femmina a due pin e realizzato un cavallotto rimovibile per non rendere "definitiva" la modifica.

Saldare ora un connettore femmina a quattro pin al modulo magnetometro, avendo cura di tenere lo stesso a filo del circuito stampato montato con il lato componenti verso il basso; tagliate delicatamente con un tronchesino la parte di pin in eccesso ed eliminate delicatamente, aiutandovi con un paio di pinze, la parte di circuito stampato con i due fori di fissaggio (la basetta è già incisa per facilitare l'operazione).

Realizzate il cavo di collegamento tra modulo telemetria e modulo APM; il cavo è lungo circa 50 cm, ad un capo vi è saldato un connettore femmina a quattro poli angolato a 90 gradi e dall'altro ha un connettore maschio a 5 poli collegato come nello

Fig. 9

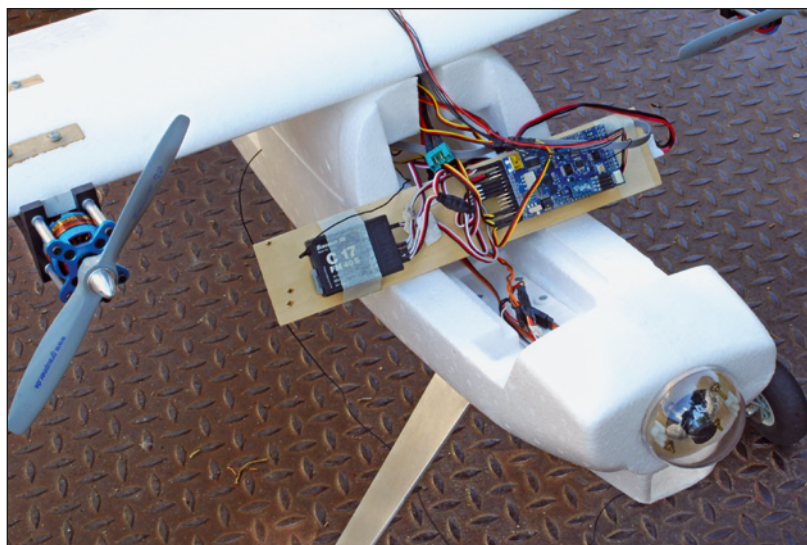


schema elettrico. Fatto ciò, realizzate il cavo di collegamento tra modulo GPS e modulo APM, lungo circa 25 cm, avente ad un capo un connettore femmina a quattro poli ed all'altro un connettore maschio sempre a 4 poli.

Modificare il firmware del modulo GPS

Il modulo GPS è preconfigurato in fabbrica. A seconda del periodo di fabbricazione può essere precaricato con diverse versioni di firmware. I più vecchi hanno a bordo la versione 1.2 mentre il modulo ArduPilotMega IMU, per colloquiare con il GPS richiede che su quest'ultimo sia installata almeno la versione 1.4 (sui modelli più recenti in fabbrica viene installata la versione 1.6). Il nostro sistema accetta e riconosce automaticamente sia la versione 1.4 che la 1.6 del firmware GPS. Se il GPS in vostro possesso ha già installata una di queste versioni, non dovete fare nulla, ma se ha la versione 1.2 oppure se non sapete quale release c'è a bordo, installate una delle

Fig. 10



versioni indicate. Noi, per pigrizia, abbiamo utilizzato il GPS usato per l'Easystar descritto negli articoli precedenti, dove avevamo già installato la versione 1.4. Trovate le istruzioni per modificare il firmware del GPS nel numero 155 di Elettronica IN. Per aggiornare il firmware del GPS scaricate applicazione e firmware da <http://code.google.com/p/ardupilot/downloads/list> (il file è *MTK_DIYdrones.zip*).

Nel file scaricato si trovano la versione 1.4, l'applicazione necessaria a caricarlo sul GPS attraverso la porta seriale e le istruzioni in inglese. La versione di firmware 1.6 è scaricabile da:

http://api.ning.com/files/taDZ5KJK**s3X*OdmnVy2mgVbaAZX0TpLaZYkQVvQi2GMhS-glD*N6S6rC5sv-YewSZfjdABxz3GnReh4r4OQmA__/_AXN1.30_2389_3329_384.1151100.1_v16.bin.

Quando arrivate al punto di eseguire il comando Download dal sottomenu Connection, scegliete il file *AXN1.30_2389_3329_384.1151100.1_v16*.

Configurare i moduli di telemetria

Per rendere funzionante la telemetria è necessario eseguire una serie di passi: colle-

gate al PC il modulo XBee Explorer USB con il cavetto USB; se il modulo non viene riconosciuto automaticamente dovete reperire il driver dal sito www.ftdichip.com/Drivers/VCP.htm.

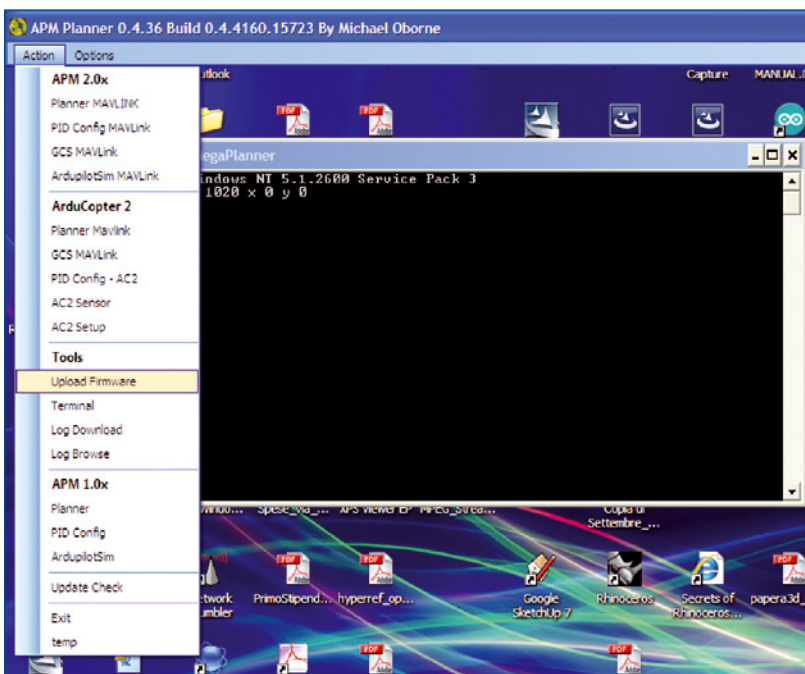
Per configurare i moduli XBee recuperate ed installate il software X-CTU (reperibile su www.digi.com/support/productdetail.jsp?pid=3352&osvid=57&s=316&tp=5&tp2=0) e, come descritto nel numero 152 di Elettronica IN (dicembre 2010, gennaio 2011) pagina 141 e seguenti, configurare i due moduli XBee in modo che possano comunicare tra loro. In particolare, per ciascuno dei due moduli XBee:

- selezionare il tipo di modulo XBee da inizializzare (il tipo di modulo si può leggere sul rovescio del modulo stesso, quello che non si vede quando montato sulla piedinatura);
- leggere i dati dal modulo (pulsante READ);
- impostare un PAN-ID a piacere, ma rigorosamente identico per i due moduli;
- impostare la velocità della porta seriale (il modulo APM prevede una velocità preconfigurata di 57.600 bps);
- inizializzare il modulo XBee (pulsante WRITE).

Lasciate uno dei due moduli XBee sul modulo XBee Explorer USB. Installate l'altro modulo XBee sul modulo XstreamBee Board che verrà collegato ad APM.

Per l'alimentazione di tutto il sistema, abbiamo deciso di optare per un convertitore switching DC-DC separato da 6 A per non rischiare di sovraccaricare gli alimentatori dei regolatori di giri. Il modulo utilizzato è reperibile presso la Futura Elettronica (con il codice 7300-BOB09370) e descritto nel numero 154 di marzo 2011 di Elettronica IN. La tensione di 5 volt è ottenuta utilizzando una resistenza di regolazione da 1,34 kohm. Ora è possibile cablare tutto il sistema all'interno del modello rispettando lo schema a blocchi e la descrizione dei contatti per il collegamento ai canali della ricevente e ai cavi dei servocomandi. Nel nostro modello, per compattare il tutto e mantenere ordinati i cavi, abbiamo scelto di fissare la ricevente ed il modulo APM su una basetta di compensato in grado di scorrere all'in-

Fig. 11



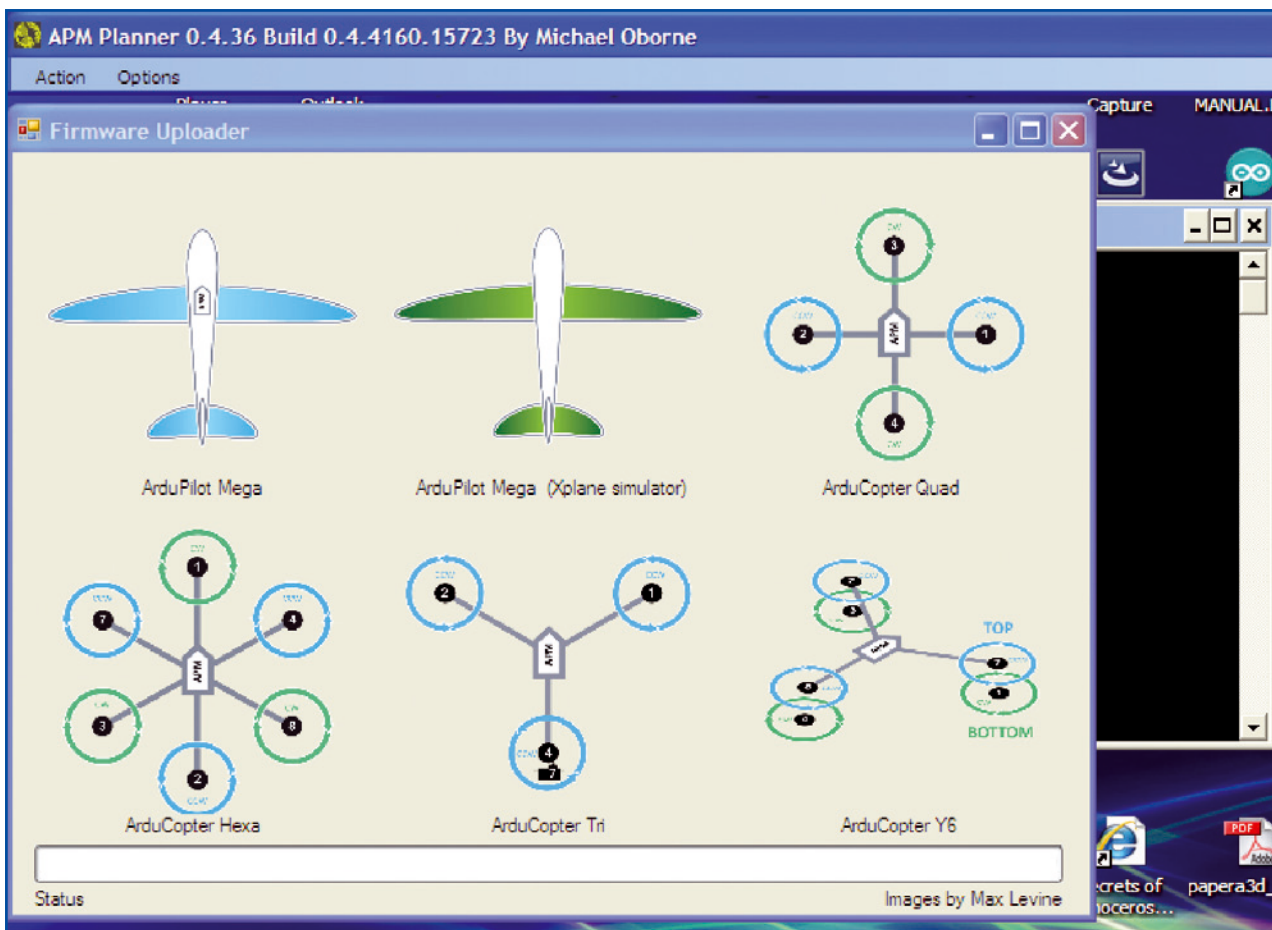


Fig. 12

terno della fusoliera (Fig. 10). Fate molta attenzione ad orientare il modulo APM con le prese GPS e telemetria verso il muso dell'aereo e a centrarlo perfettamente sulla basetta, in modo da creare le condizioni per il funzionamento corretto del modulo IMU. A questo punto per inizializzare e configurare il software di APM esistono due possibilità: quella più semplice consiste nello scaricare unicamente il software APM Mission Planner ed utilizzarlo anche per caricare il firmware sul modulo APM; l'alternativa utilizza l'IDE di Arduino per compilare ed eseguire l'upload del software. Vi diamo le indicazioni per entrambe le possibilità, nel nostro caso abbiamo preferito la via "classica" anche per dare un'occhiata al software ed eseguire una verifica di compilazione.

Cominciamo con lo scaricare tutti i software necessari: l'IDE di sviluppo di Arduino, è disponibile gratuitamente dal sito <http://arduino.cc/en/Main/Software>; la versione da installare è la 022 o successiva. Dal sito <http://code.google.com/p/ardu->

[pilot-mega/downloads/list](http://code.google.com/p/ardu-pilot-mega/downloads/list) scarichiamo il sorgente di APM *APM2.xx.zip* ed il software dell'applicazione APM Mission Planner *APM Mission Planner04.xx.zip*.

Scompattare i due file nel disco C rispettando le directory originali. Infine, dal sito <http://code.google.com/p/happykillmore-gcs/downloads/list>, scaricare l'ultima versione del software della Ground Control Station *Setup_xxxxxxx.exe* ed installarla. Colleghiamo ora la presa USB del modulo APM al PC con un cavo USB.

Apriamo l'IDE di Arduino e in Tools->Board selezioniamo Arduino Mega (ATmega1280) e in Tools->Serial Port definiamo la porta COM alla quale è collegato APM ed infine in File->Preferences nella Sketchbook location inseriamo la directory che contiene il software di APM.

Per rendere attive le impostazioni è necessario chiudere e riaprire l'IDE.

Aprire il file *ArduPilotMega.pde* compilate ed eseguite l'upload sul modulo APM.

Nel caso preferiate scegliere la via dell'installazione diretta del firmware da APM

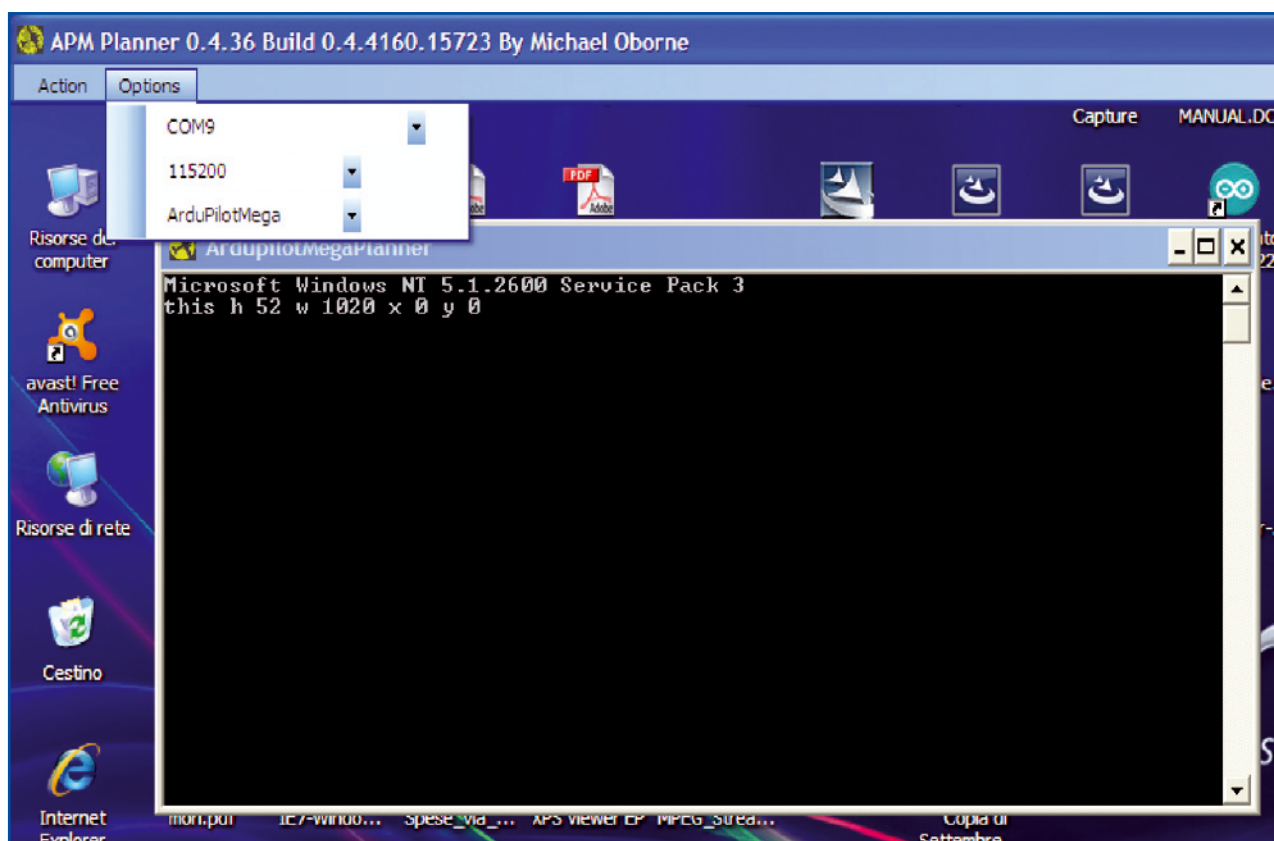


Fig. 13

Mission Planner, aprite quest'ultima applicazione e in Options impostate la porta COM e la velocità. Utilizzate la funzione Action->Tools->Upload Firmware e selezionate la tipologia di modello (Fig. 11 e Fig. 12).

A questo punto è possibile eseguire la configurazione iniziale del modulo APM (Fig. 13). Le grosse novità associate alla disponibilità del modulo APM e del suo software di gestione sono:

- la capacità del software di autoapprendere la tipologia e le caratteristiche di installazione dei sensori collegati ad APM; infatti non abbiamo configurato alcun parametro per dire che tipo di GPS abbiamo collegato e con quale release di firmware, né se abbiamo collegato un magnetometro e con quale orientamento;
- la possibilità di inizializzare, configurare e calibrare i parametri ed i valori di funzionamento dell'APM tramite applicazioni esterne sia a terra che in fase di volo; le configurazioni e le verifiche iniziali si eseguono utilizzando la modalità CLI (Command Line Interpreter) implementata nel software ArdupilotMega,

mentre in fase "operativa" vengono utilizzate le funzioni di controllo "online" disponibili nelle applicazioni Ground Control Station e APM Mission Planner.

Dunque, per la prima inizializzazione dobbiamo utilizzare la CLI, dopo avere acceso la trasmettente del radiocomando con APM collegato al PC con il cavo USB.

Per attivarla è necessario portare l'interruttore a slitta sul modulo APM in posizione Setup, ovvero verso il connettore dei cavi dei servocomandi, eseguire il Reset del modulo con l'apposito pulsante, lanciare dall'IDE il Serial monitor e configurare la porta COM e la velocità di comunicazione a 115.200 bps, oppure utilizzare la funzione Terminal dal Mission Planner, sempre dopo avere configurato la porta COM e la velocità. L'interfaccia a linea di comando si presenta come un'applicazione Telnet, ed è caratterizzata da una sintassi di comandi organizzati in una gerarchia di menu. Digitando "Help" nella linea di comando (o al prompt della finestra CMD nel caso usiamo la APM Mission Planner) otteniamo un menu di funzioni disponibili. Digitando uno dei comandi si entra nel sottomenu corri-

spondente; la lista delle funzioni disponibili è sempre visibile digitando "Help", digitando "Exit" si torna indietro di un livello (Fig. 14).

Il primo livello offre le seguenti funzioni:

- "logs" gestione del log on line;
- "setup" configurazione iniziale;
- "test" lista e verifica impostazioni.

Digitiamo "Setup" e poi "help" per ottenere le funzioni disponibili in Setup; ci appare la lista delle funzioni:

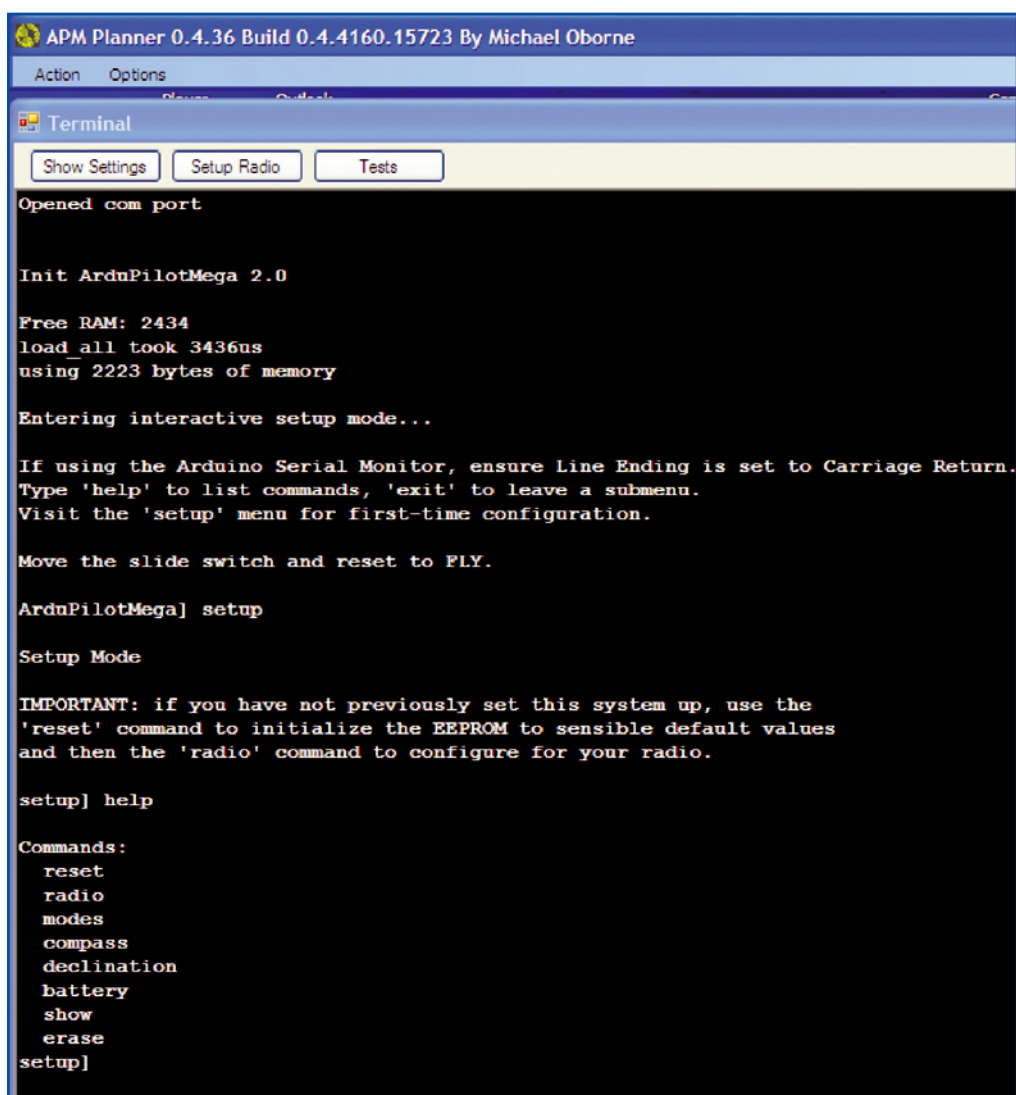
- "reset" inizializza APM caricando in EEPROM i valori di default;
- "radio" calibra i segnali ricevuti dall'equipaggiamento RC;
- "modes" imposta le modalità di volo associate alle posizioni dell'interrutto-

re di controllo;

- "show" elenca i valori impostati in EEPROM.

La prima inizializzazione richiede la seguente sequenza di comandi.

- "Reset", per caricare in EEPROM i valori iniziali di tutti i parametri; risponde "Y" alla richiesta di conferma.
- "Radio", per calibrare i valori relativi alla corsa dei comandi del radiocomando, stick e interruttore; muoviamo gli stick della radio a fondo corsa in tutte le direzioni, spostiamo l'interruttore di controllo in tutte le posizioni, poi premiamo il tasto "Enter" per confermare i valori impostati.
- "Modes" questo comando ci permette



The screenshot shows the APM Planner 0.4.36 Build 0.4.4160.15723 By Michael Osborne window. The 'Terminal' tab is active, displaying the following text:

```
Opened com port

Init ArdnPilotMega 2.0

Free RAM: 2434
load_all took 3436us
using 2223 bytes of memory

Entering interactive setup mode...

If using the Arduino Serial Monitor, ensure Line Ending is set to Carriage Return.
Type 'help' to list commands, 'exit' to leave a submenu.
Visit the 'setup' menu for first-time configuration.

Move the slide switch and reset to FLY.

ArdnPilotMega] setup

Setup Mode

IMPORTANT: if you have not previously set this system up, use the
'reset' command to initialize the EEPROM to sensible default values
and then the 'radio' command to configure for your radio.

setup] help

Commands:
  reset
  radio
  modes
  compass
  declination
  battery
  show
  erase
setup]
```

Fig. 14



di configurare le modalità di volo associate all'interruttore a tre posizioni sulla trasmettente; posizioniamo l'interruttore sulla prima posizione verifichiamo che sia impostato a "Manual". Se non è così l'interruttore potrebbe essere invertito e il "Manual" trovarsi nella posizione opposta dell'interruttore. Verificate e scegliete come usare l'interruttore, eventualmente utilizzando il "reverse" sulla trasmettente. Posizionate l'interruttore sulla seconda posizione: utilizzando lo stick degli alettoni ogni spostamento a fondo corsa verso destra ci permette di cambiare la modalità di volo associata; una volta impostata quella prescelta, passiamo alla terza posizione e ripetiamo l'operazione. Al termine confermiamo il tutto con il tasto "Enter".

- Se abbiamo installato il magnetometro, dobbiamo abilitarlo digitando "Compass on"; verifichiamo digitando "Compass" lo stato del magnetometro. Dal menu "Test" è possibile accedere ad un elenco di funzioni che ci permettono di verificare il funzionamento dei vari componenti installati; ciascuna funzione mostra, una volta richia-

mata, i valori che vengono letti dai corrispondenti ingressi; per uscire dalla funzione premere il tasto "Enter".

- "radio": mostra i valori letti su ciascun canale della radio.
- "gps": mostra i dati letti dal modulo GPS, deve essere eseguita con il modello in un luogo dove è ricevibile il segnale GPS. In caso contrario viene mostrato il valore 0.
- "imu": mostra l'orientamento calcolato sugli assi X, Y e Z.
- "gyro": mostra i valori letti dagli accelerometri e dai giroscopi.
- "battery": mostra il voltaggio della batteria se è stato attivato il monitoraggio del voltaggio della batteria.
- "relay": verifica il funzionamento del relé installato sul modulo ArdupilotMega IMU.
- "waypoints": elenca le "boe" GPS memorizzate in EEPROM.
- "airspeed": mostra le letture provenienti dal tubo di Pitot, se installato.
- "airpressure": mostra le letture provenienti dal sensore di pressione installato sul modulo ArdupilotMega IMU.

- "compass": mostra le letture del magnetometro, se installato e attivato.
- "xbee": mostra le stringhe di dati provenienti dal modulo Xbee, se installato.
- "eedump": ricarica le impostazioni correnti sulla EEPROM.

Per uscire dalla modalità CLI è necessario riportare la levetta dell'interruttore a slitta in posizione di volo, verso i DIP switch e resettare APM con l'apposito pulsante.

Bene, con questo per il momento abbiamo concluso; lavorate su queste impostazioni e preparate il tutto. Il mese prossimo vi spiegheremo come lavorare con la piattaforma software che realizza la Ground Control Station Happy Killmore, della quale con APM utilizzeremo appieno anche le funzionalità di controllo bidirezionale basata sul protocollo Mavlink. ☐

Guarda l'UAV in azione su YouTube!



Vuoi vedere le prove di volo del nostro UAV? Vuoi vederlo volare in maniera completamente autonoma seguendo un percorso prefissato? Nulla di più semplice: collegati alla pagina <http://www.youtube.com/watch?v=2Y0N-6ka5hU> dove abbiamo postato un filmato con i preparativi, la messa a punto e le prove di volo.

COSTRUIAMO E SPERIMENTIAMO UNA STAMPANTE 3D

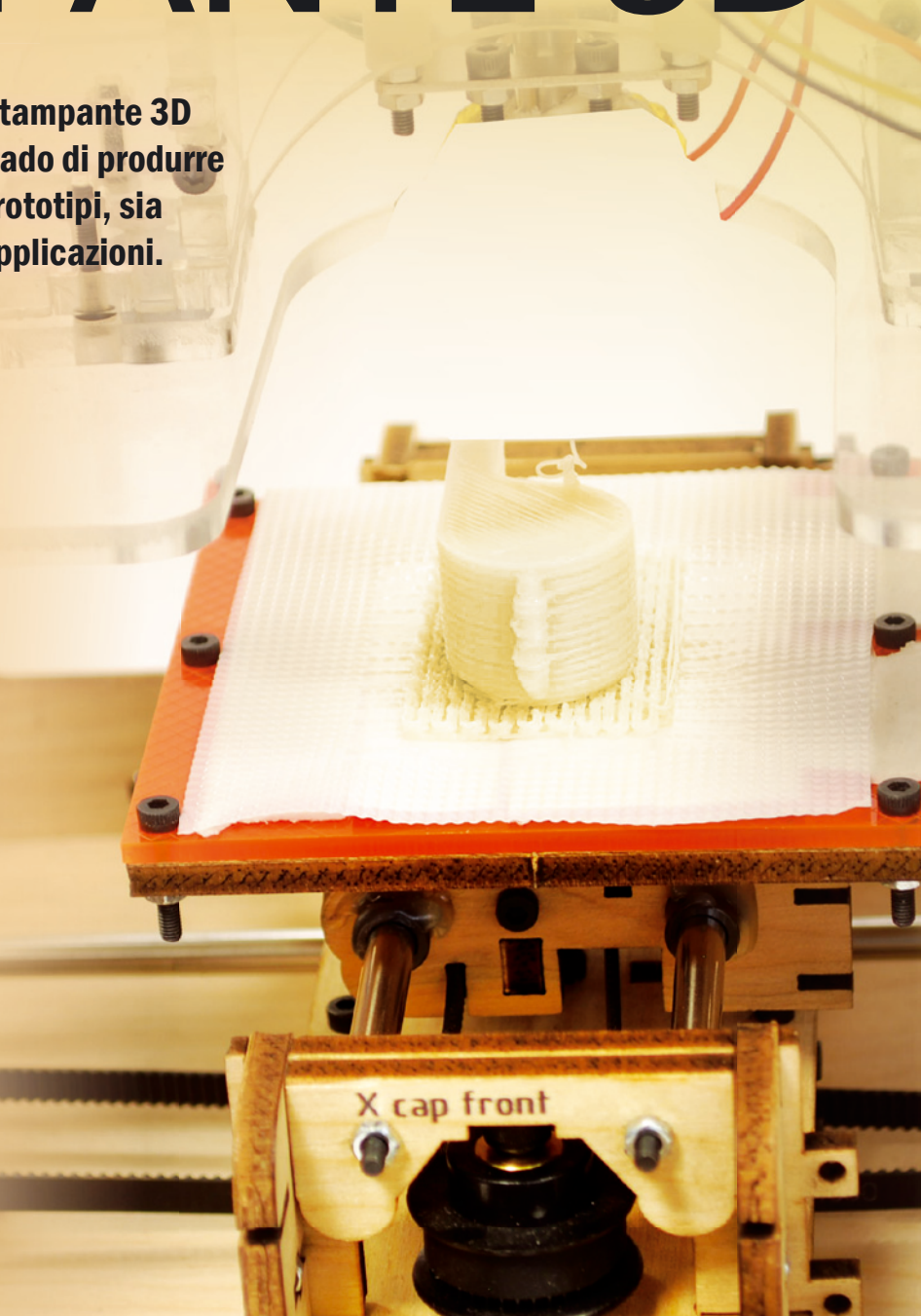
di MARCO MAGAGNIN

La realizzazione e l'utilizzo di una stampante 3D alla portata di tutte le tasche in grado di produrre oggetti in robusto ABS, sia come prototipi, sia direttamente utilizzabili in molte applicazioni.

Nel numero di Maggio 2011 abbiamo presentato una panoramica sullo stato dell'arte dei sistemi di stampa 3D applicati ai settori tecnologici e produttivi più avanzati. Nello stesso articolo si accennava alla ricaduta di queste tecnologie verso applicazioni alla portata dell'utenza amatoriale.

Per non smentire lo spirito pratico, ancorché innovativo, della rivista, in questo numero vi proponiamo la realizzazione e l'utilizzo di una stampante 3D alla portata di tutte le tasche, in grado di produrre oggetti in robusto ABS, sia come prototipi, sia direttamente utilizzabili in molte applicazioni.

L'esigenza di possedere una macchina di questo tipo è sorta a fronte della necessità di realizzare prototipi e/o piccole serie di componenti da impiegare nei nostri progetti e non reperibili in commercio. Le esigenze spaziano da componenti per la robotica (azionamenti, cingoli, supporti per sensori, cinematismi) ai componenti specializzati per i nostri modelli di UAV (supporti motori, supporti e pan and tilt per telecamere, sensori e sistemi di comunicazione) e per altri progetti in sviluppo. Da un punto di vista meccanico e struttu-



rale una macchina di questo tipo è molto simile ad una macchina CNC 3D controllata da computer.

La differenza sostanziale sta nel fatto che una CNC tradizionale, ad esempio una fresa, produce il pezzo desiderato per asportazione di materiale, in passate successive, a partire da un blocco di materiale compatto di dimensioni opportune. Il movimento degli assi X e Y portano la fresa nei punti dove deve essere asportato il materiale mentre l'abbassamento dell'asse Z, solidale con la fresa, permette l'asportazione di strati di materiale sempre più profondi.

Il vantaggio di questa lavorazione consiste nella possibilità di lavorare le parti utilizzando i materiali definitivi, anche metalli molto resistenti, mentre la limitazione principale sta nella impossibilità di lavorare il materiale nei cosiddetti "sottosquadri" cioè le zone del pezzo da realizzare in ombra rispetto alla traiettoria dell'utensile.

Contrariamente ad una CNC tradizionale, la stampante 3D produce il pezzo finito per aggiunta incrementale di materiale, in strati successivi, a partire dalla base e via via proseguendo verso l'alto.

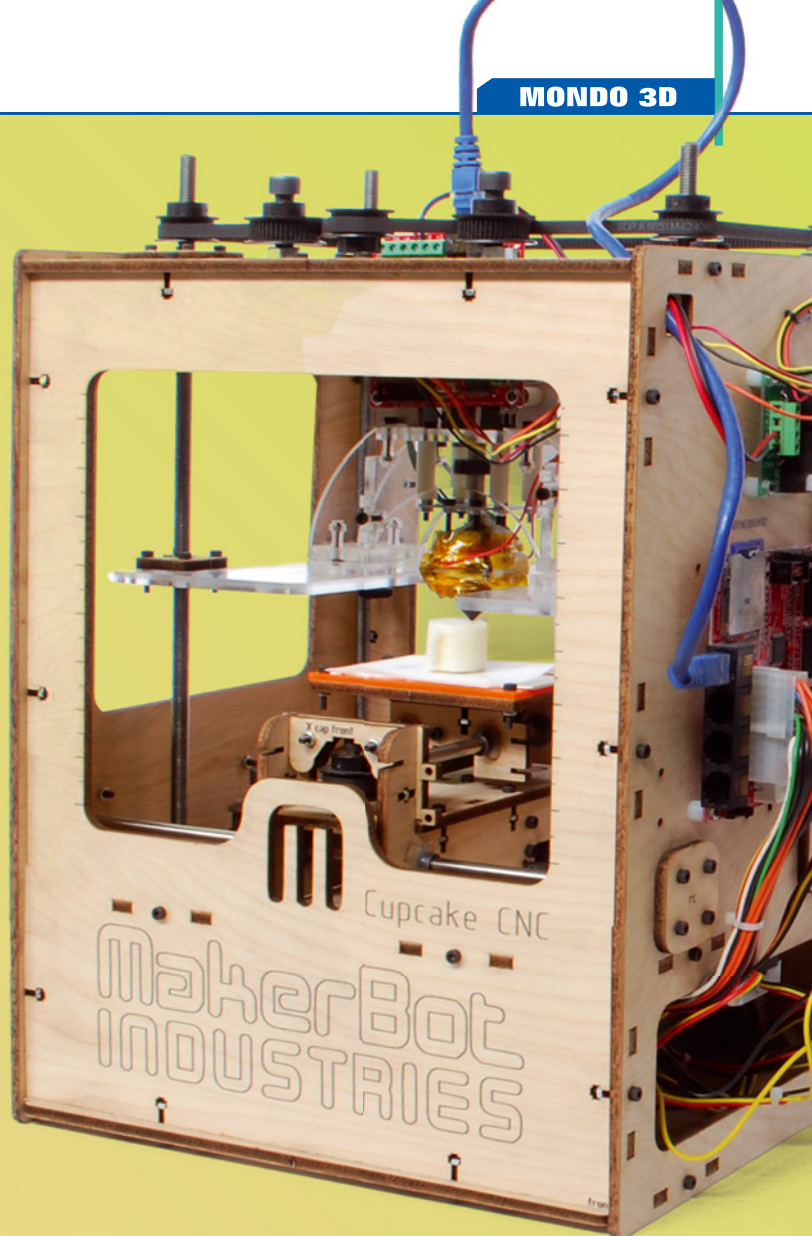
La meccanica degli assi X e Y è simile a quella di una CNC mentre l'asse Z lavora a rovescio, partendo dal basso e salendo man mano che il pezzo viene formato.

L'utensile montato sull'asse Z è costituito da un estrusore, ovvero da un dispositivo, alimentato da un tondino di ABS, in grado di produrre un filamento continuo di ABS fuso dello spessore di 0,5 mm. L'oggetto finale viene costruito per deposito successivo di strati di ABS fuso.

Al termine di ogni strato l'asse Z si alza dello spessore di uno strato ed il movimento degli assi X ed Y permette la realizzazione dello strato successivo nella forma corretta.

Questo tipo di produzione permette di realizzare oggetti anche con diversi sottosquadri, strutture con nervature chiuse ed anche oggetti all'interno di altri oggetti, come la pallina all'interno di un fischiello.

Per le nostre realizzazioni volevamo una macchina dalle buone prestazioni, che ci permettesse di



fare esperienza sulla tecnologia sottostante e che ... costasse una cifra alla portata di un hobbista.

Scartate le diverse soluzioni commerciali professionali dal costo proibitivo, ci siamo orientati verso un progetto open source, dove tutta la documentazione, i disegni della meccanica, gli schemi della elettronica ed il software di funzionamento fossero disponibili ed utilizzabili in licenza Open.

Il progetto originale è quello della RepRap; nel nostro caso, per evitare una caccia al tesoro di componenti, parti meccaniche, realizzazione di circuiti stampati, saldature SMD, ecc. abbiamo utilizzato un kit "tutto da montare" reso disponibile dalla Makerbot Industries (sito <http://www.makerbot.com/>).

Per la nostra realizzazione abbiamo scelto i seguenti componenti:

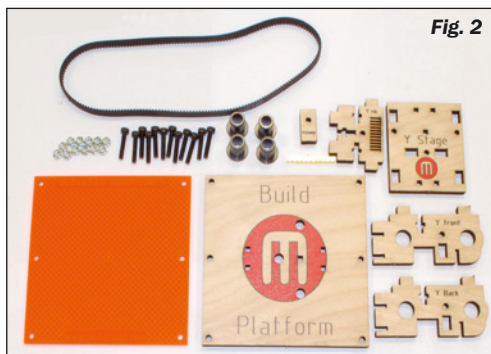


Fig. 2

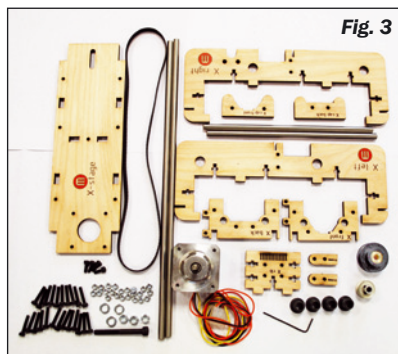


Fig. 3



Fig. 4

La nostra macchina 3D è disponibile in versione da montare con tutti i pezzi già sagomati e forati. Le schede elettroniche vengono fornite già montate e collaudate.

- Meccanica Cupcake CNC: meccanica CNC tre assi, con tre motori passo passo per il controllo del movimento; permette di stampare pezzi fino a 10 cm x 10 cm di lato e 13 cm di altezza,
- Estrusore Plastruder MK5,
- Scheda madre Makerbot Rep Rap versione 1.2, basata sulla architettura Arduino,
- Scheda controllo estrusore versione 2.2, anch'essa basata su architettura Arduino,
- 3 schede per il controllo dei motori passo passo,
- Scheda controllo relè per il riscaldamento dell'estrusore,

- Un alimentatore da PC (assicurarsi che il selettore della tensione sia configurato per l'alimentazione di rete a 220 volt).

L'inizio della nostra realizzazione risale a qualche mese fa e di conseguenza la scelta dei moduli è stata dettata dallo stato di sviluppo del progetto a quella data.

Oggi sono disponibili dei moduli di elettronica più evoluti, sempre basati su Arduino e delle varianti della meccanica che ne ottimizzano le funzionalità.

Nel caso vogliate intraprendere la realizzazione vi consigliamo di utilizzare le ultime versioni dei moduli di controllo e di adottare la piastra di stampaggio riscaldata.

Costruzione della meccanica CNC

Nelle foto si possono vedere le varie parti componenti il telaio e la meccanica per la movimentazione degli assi X, Y e Z (Figure 2, 3 e 4).

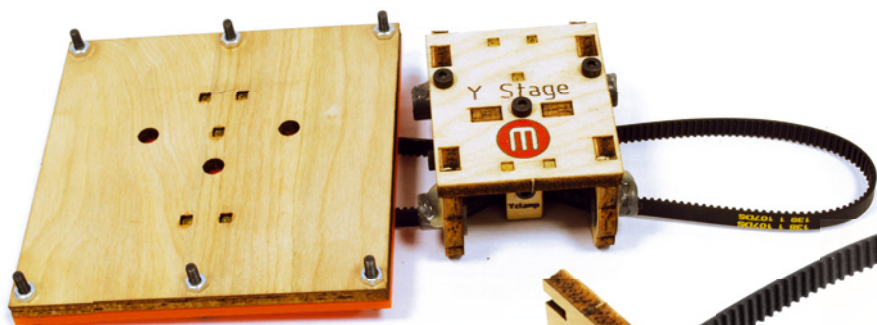


Fig. 5

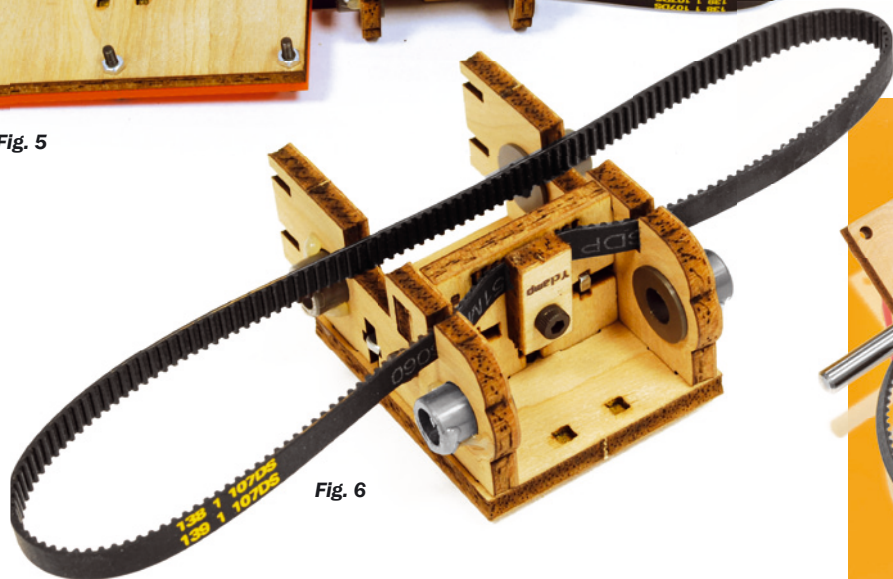


Fig. 6

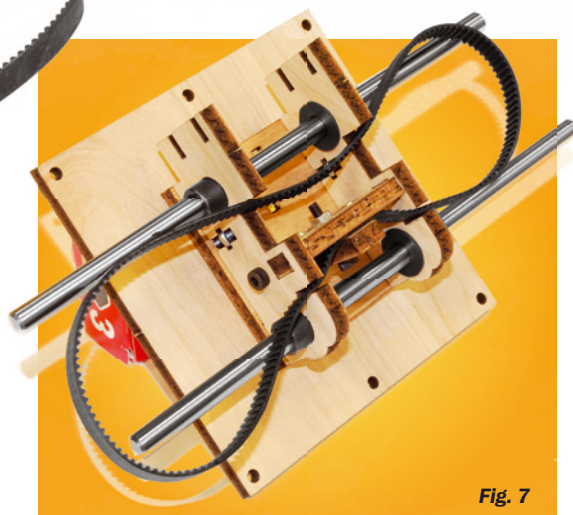


Fig. 7

La movimentazione degli assi avviene per mezzo di cinghie dentate trainate da ingranaggi collegati a motori passo passo, a loro volta comandati dalla elettronica di controllo. Il criterio di riconoscimento delle diverse cinghie dentate è dato dal numero di denti; il numero di denti è stampigliato sul dorso di ciascuna cinghia come visibile nella **Figura 6**.

Come prima operazione assemblate la base di stampa che è anche solidale con l'asse Y (**Fig. 5**).

Inserite i magneti nelle sedi appropriate e fissateli con alcune gocce di colla. Assemblate il carrello porta piastra e posizionate in sede la cinghia da 196 denti, bloccandola lateralmente con il fermo in legno e la relativa vite con dado.

Posizionate le quattro boccole con flangia ed inserite le due barre di scorrimento (le due più corte tra quelle fornite); fissate in sede le boccole con qualche goccia di colla a caldo (**Fig. 7**).

Sfilate le barre di scorrimento.

Ora assemblate il carrello dell'asse X, a forma di culla. I quattro pezzi vanno uniti con viti e dadi rispettando le scritte di descrizione che indicano l'orientamento del pezzo.

Montate la puleggia di trascinamento sul motore passo passo, e montate il motore sul carrello, dalla parte del foro più grande. Fissatelo con le viti.

Nella parte opposta del carrello posizionate la puleggia di tensione della cinghia dentata.

Collocate il carrello dell'asse Y al centro della culla e posizionate la cinghia sulle pulegge.

Fissate da una parte il fermo delle barre di scorrimento, inserite le barre dalla parte opposta e fissate il secondo fermo (**Fig. 8**).

Bloccate la cinghia da 264 denti sul bloccetto ferma cinghia e fissatelo nella sua sede al di sotto della culla, facendo passare la cinghia negli intagli predisposti.

Sempre al di sotto della culla posizionate le quattro boccole per lo scorrimento delle barre ed inserite le barre. Se queste scorrono senza sforzo fissate in posizione le boccole con qualche goccia di colla a caldo (**Fig. 9**).

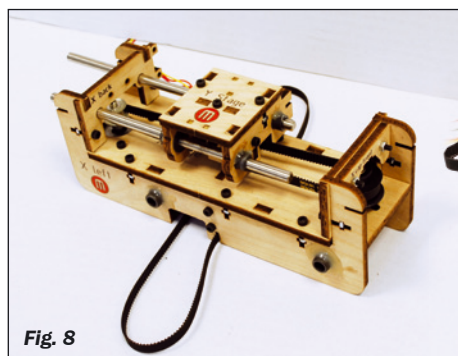


Fig. 8

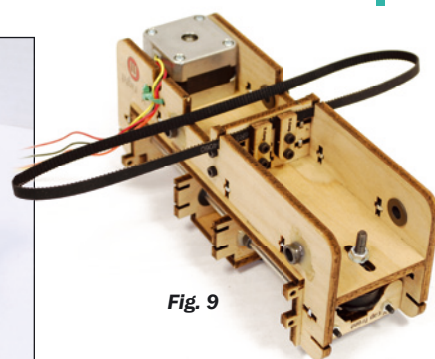


Fig. 9

Sfilate momentaneamente le barre e portate la cinghia dell'asse Y alla giusta tensione posizionando correttamente la puleggia libera. Non tendete troppo la cinghia.

Ora potete iniziare ad assemblare il telaio esterno della Cupcake,

Fissate con le apposite viti e dadi nella parte inferiore del ripiano intermedio i quattro supporti quadrati per i cuscinetti dell'asse Z.

Montate il ripiano intermedio sul pannello frontale della Cupcake, posizionate i dadi nelle apposite fessure e puntate le viti a brugola. Non stringete le viti fino a montaggio terminato (**Fig. 10**).

Allo stesso modo montate i ripiani inferiori.

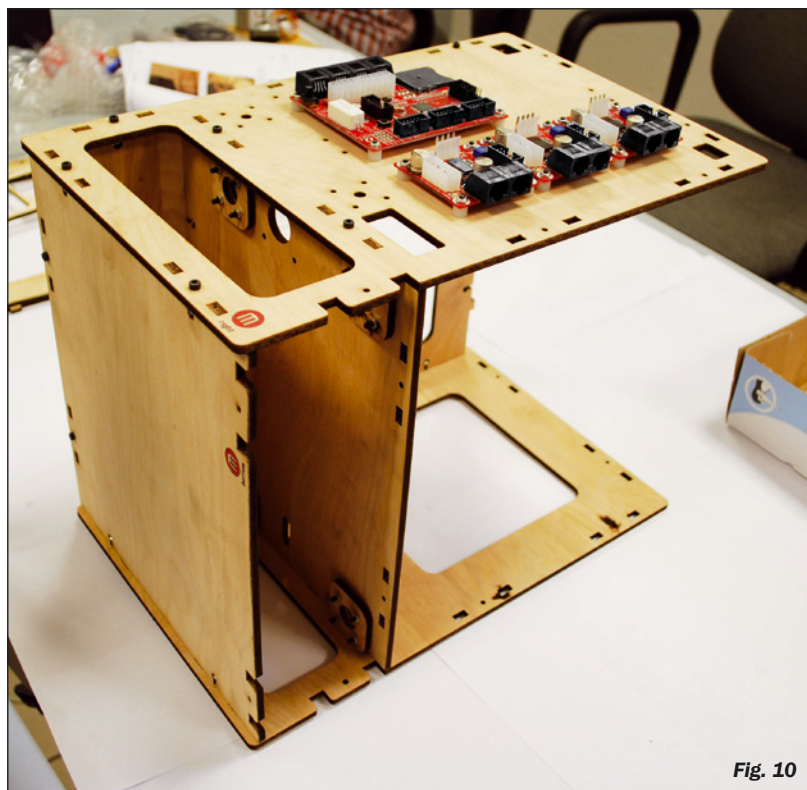
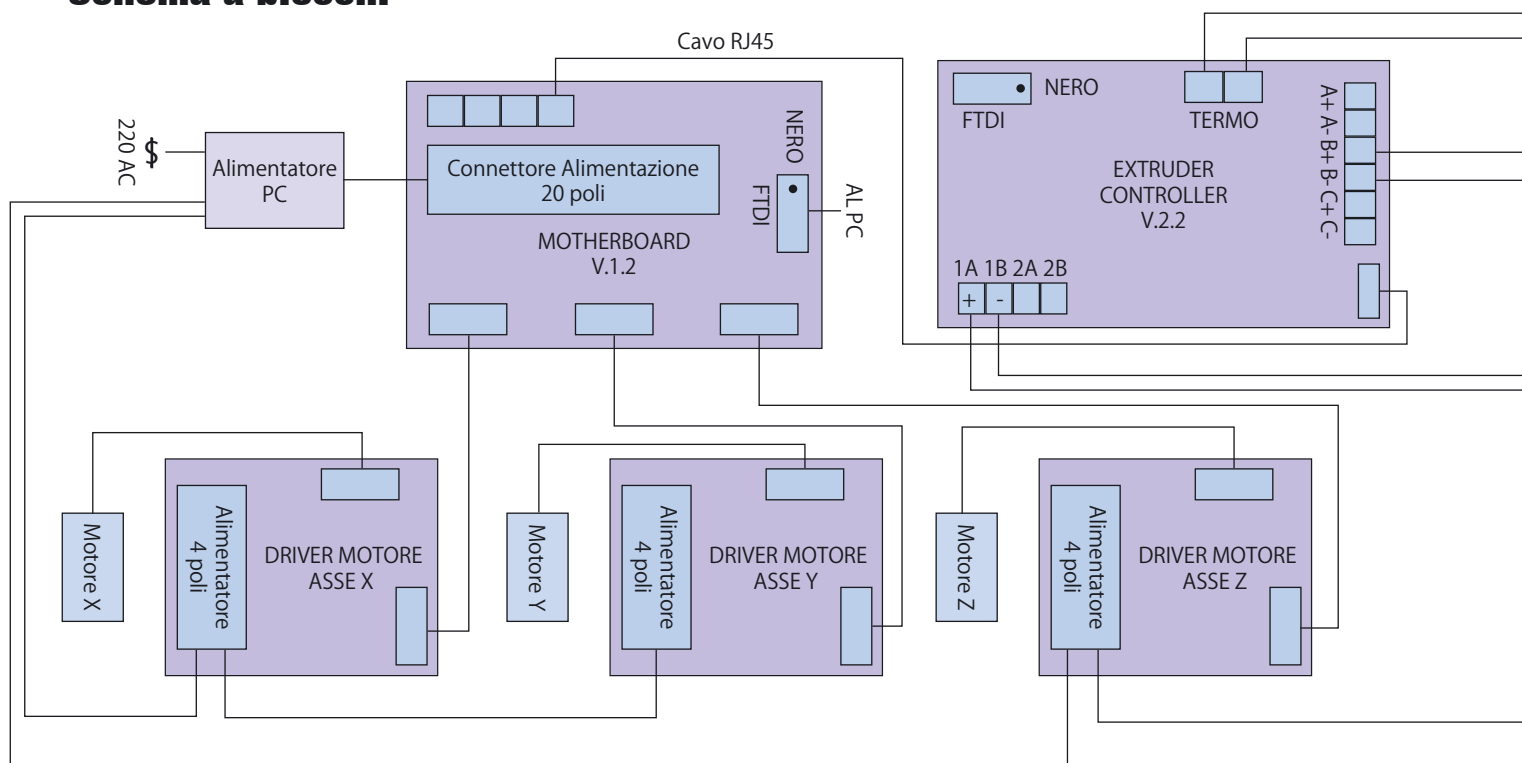


Fig. 10

Schema a blocchi



re e superiore e gli altri tre pannelli laterali.

Fate combaciare bene tutti gli incastri, verificate che tutti i pannelli siano in squadra e stringete le viti a brugola per rendere solida tutta la costruzione. Non esagerate nello stringere le viti.

Montaggio dell'elettronica sulla meccanica CNC

Nello schema a blocchi viene descritto come collegare tra loro i vari moduli che compongono la stampante 3D.

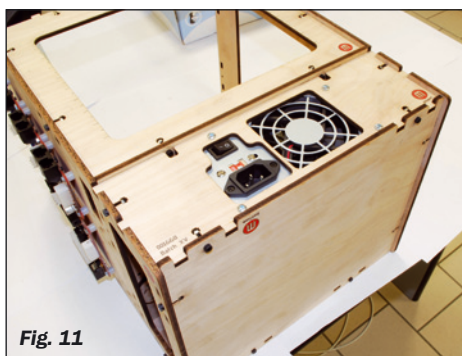


Fig. 11



Fig. 12

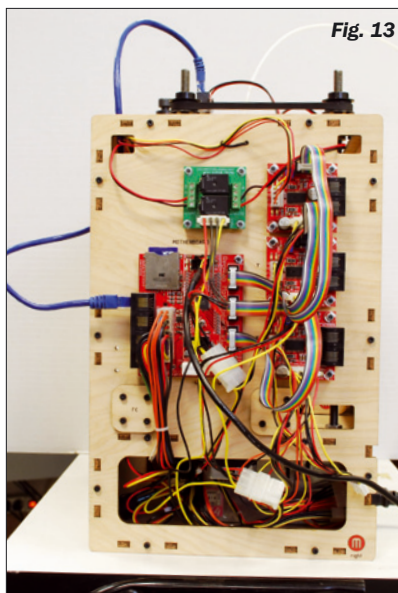


Fig. 13

Per prima cosa fissiamo i moduli sulla fiancata della stampante. Per ciascun modulo usiamo quattro distanziali e quattro viti con relativo dado.

Può essere necessario eseguire dei fori aggiuntivi rispetto a quelli già disponibili sul telaio della stampante per il montaggio della scheda relè.

Una buona disposizione dei moduli e delle schede è quella mostrata nelle (Figure 10 e 13).

Inseriamo e fissiamo l'alimentatore nel suo alloggiamento nella parte inferiore del telaio della stampante e facciamo fuoriuscire i cavi dalla finestratura (Figure 11 e 12). Completate il montaggio del telaio con la meccanica dell'asse Z.

Montate sul pannello di plexiglass i quattro fermi per i dadi di trascinamento del pannello.

Su ciascuna delle quattro barre filettate montate i cuscinetti ad una delle estremità come indicato nella foto in alto a destra.

Avvitare un dado fino a circa metà

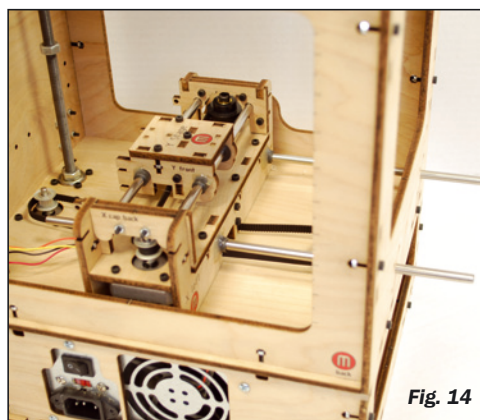
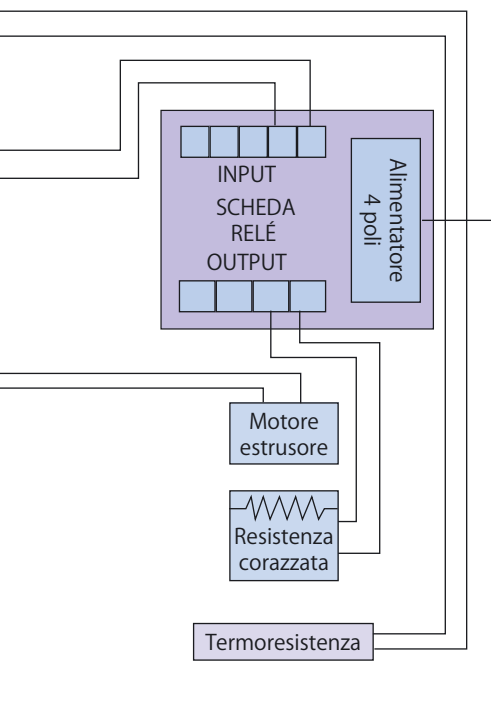


Fig. 14

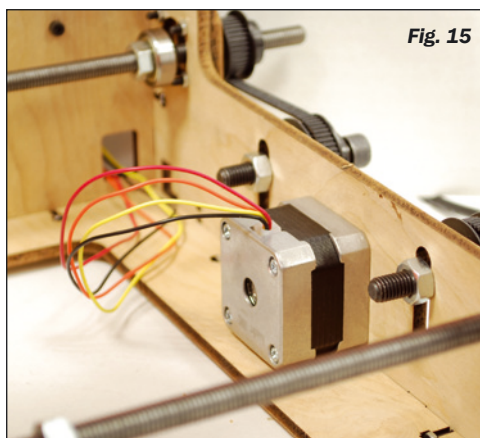


Fig. 15

corsa di ciascuna barra filettata e, sulle estremità rimaste libere ancora un dado, un cuscinetto ed un ultimo dado (Fig. 15).

Inserite l'estremità di ciascuna barra filettata nei fori in alto del ripiano superiore del telaio, tenendo il lato con i cuscinetti già bloccati verso il basso, e poi alloggiare i cuscinetti nella parte bassa nelle apposite sedi (Fig. 14).

Montate i quattro reggi cuscinetto quadrati in legno nella parte superiore del ripiano e fissateli ciascuno con quattro viti e dadi.

Spingete i cuscinetti superiori contro i supporti avvitando il dado inferiore senza forzare. Montate le quattro pulegge e bloccatele con gli appositi grani.

Montate il motore passo passo nel foro centrale del ripiano superiore verso il pannello frontale, con il motore posizionato nella parte inferiore del ripiano.

Fissate la puleggia all'albero motore di trascinamento e montate le due pulegge di tensionamento della cinghia. Sistemate la cinghia come nella foto e tendetela usando le due pulegge di fianco al motore; la cinghia deve essere tesa senza esagerare per non sovraccaricarla durante il funzionamento.

Ora installate il carrello con gli assi X ed Y all'interno del telaio. Posizionate i due fermi per le barre di scorrimento dalla parte dove sarà montata l'elettronica.

Fissate la piccola puleggia in al-

luminio sul motore passo passo, e montate il motore nel suo alloggiamento nella parte inferiore del pannello intermedio, a destra guardando la macchina dal davanti. Montate la puleggia libera sul lato opposto.

Appoggiate il carrello all'interno del telaio, la parte della cinghia libera di scorrere deve guardare la parte frontale della macchina. Posizionate la cinghia sulle pulegge ed inserite le barre di scorrimento facendole passare nelle apposite boccole. Montate i due ultimi fermi per le barre. Tendete la cinghia dell'asse X, ancora una volta senza esagerare.

Appoggiate il ripiano in plexiglas all'interno del telaio, la parte incavata rivolta verso il fronte della macchina e i supporti in legno rivolti verso l'alto.

Avvitare i dadi liberi fino a che si trovino tutti alla stessa altezza ed il piano in plexiglass, una volta appoggiato, risulti in bolla da tutti i lati.

Montaggio dell'estrusore

Il "cuore", e di conseguenza il componente più delicato da realizzare del sistema di stampa, è l'estrusore ed in particolare la punta riscaldante.

La punta riscaldante è responsabile della qualità e continuità del filamento di ABS fuso, il tondino di ABS viene alimentato in continuo per mezzo di un dispositivo di avanzamento a velocità costante,

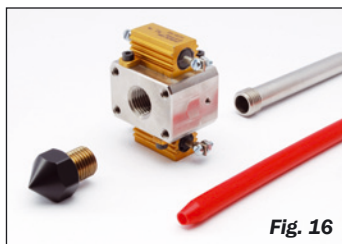


Fig. 16

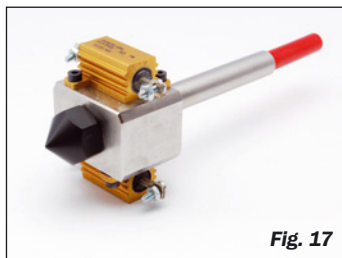


Fig. 17

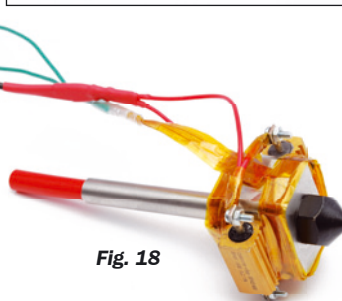


Fig. 18

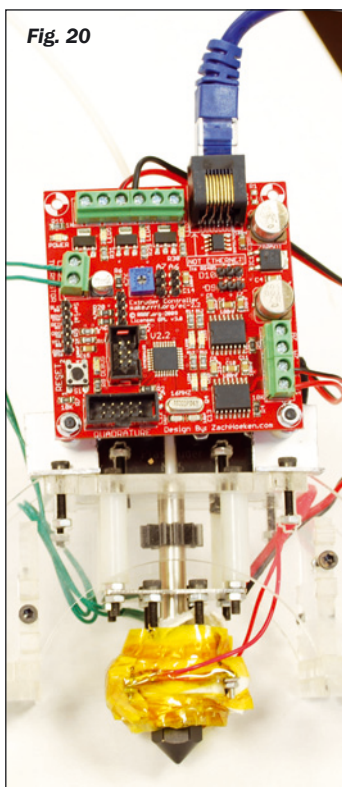


Fig. 20

costituito da un motore che aziona un rullo dentato che, a sua volta, trascina il tondino di ABS; un sistema di regolazione permette di calibrare la pressione del tondino sul rullo dentato.

Estrusore e dispositivo di alimentazione sono montati solidalmente su un ponte di plexiglass, a sua volta fissato alla piattaforma dell'asse Z. L'estrusore è composto da un corpo riscaldante in acciaio inossidabile con all'interno una guida in teflon per evitare il surriscaldamento del tondino di ABS prima che questo raggiunga l'ugello (Fig. 16).

Iniziate a montare l'estrusore a partire dalla testa riscaldante. Per prima cosa sporcate il filetto dell'ugello con frena filetti, prendete il corpo riscaldante a forma di parallelepipedo ed avvitate a fondo l'ugello dalla parte senza i fori filettati per le viti. Stringete con una chiave senza forzare, proteggendo i lati dell'ugello con del cartoncino, per non rovinare la protezione in teflon dello stesso.

Avvitate il tubo in acciaio inossidabile dell'elemento riscaldante dalla parte opposta dell'ugello, stringete senza forzare eccessivamente.

Avvitate le due resistenze corazzate sulle facce più lunghe dell'elemento riscaldante, utilizzando le viti in dotazione (Fig. 17).

Le due resistenze corazzate da 10 ohm vanno collegate in parallelo usando il cavo con la guaina in teflon.

Non saldate il cavo sui terminali delle resistenze usando lo stagno in quanto durante il funzionamento le resistenze raggiungono temperature vicine al punto di fusione dello stagno stesso con conseguenze immaginabili.

Abbiamo preferito utilizzare dei capicorda ad occhiello dove abbiamo crimpato gli spezzoni di cavo che poi abbiamo avvitato a fondo sui terminali delle resistenze.

Ai capi liberi saldiamo due prolunghe in cavo normale isolando i punti saldati con spezzoni di guaina termo restringente.

Ora è il momento di preparare la termo resistenza. Copriamo i terminali della resistenza, tranne mezzo centimetro finale, con due strati di nastro isolante Kapton (un nastro isolante resistente ad altissime temperature) collegate un cavo di prolun-

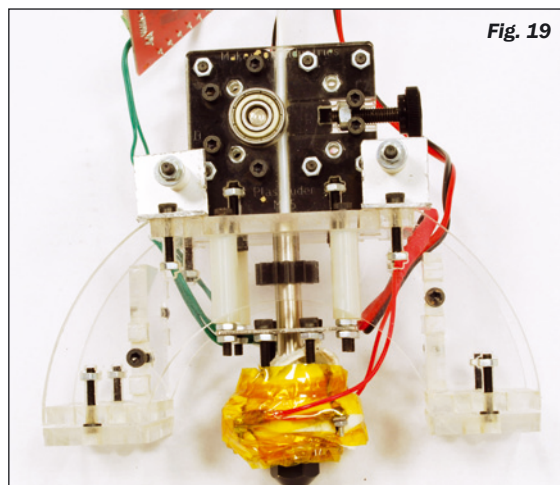


Fig. 19

ga a ciascun terminale fissandolo con un codolo ricavato da un capicorda e crimpato, per sicurezza saldiamo a stagno il collegamento così ottenuto ed isoliamo con nastro Kapton.

Fissiamo la testa in vetro della termo resistenza al corpo riscaldante con il nastro Kapton.

A questo punto prepariamo due spezzoni di nastro ceramico lunghi rispettivamente 7 e 9 cm, forateli al centro in modo da far passare il tubo di acciaio inossidabile.

Coprite con il nastro ceramico la testa riscaldante, con l'accortezza di marcare i punti con i fori filettati. Copriamo l'isolamento in nastro ceramico con diversi strati di nastro Kapton come visibile in Fig. 18. Inseriamo la guida in teflon nell'estrusore con la parte conica verso l'interno; inserite bene fino in fondo la guida in teflon e tagliate a filo la parte eccedente.

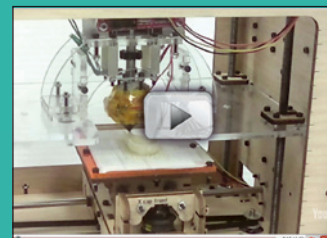
Montate la piastra di fissaggio con le viti ed infilate il radiatore a stella sulla parte alta del tubo in acciaio inossidabile.

Assembliamo il corpo in plexiglass dell'estrusore facendo attenzione a non invertire i singoli pezzi.

Inseriamo il cuneo di pressione del tondino di ABS prima di montare l'ultima piastra in plexiglass.

Montate il trascinatore dentato sull'albero motore, inserite il cuscinetto nella sua sede e completate l'installazione del motore sul corpo centrale, fissandolo con le apposite viti (Fig. 19).

Completiamo il montaggio della testa dell'estrusore sul corpo centrale usando le viti ed i distanziali e fissandolo con i dadi



Vuoi vedere in azione la stampante descritta in questo articolo? Nulla di più facile. Collegati a YouTube (www.youtube.com/watch?v=RliDs_U77w) e potrai vedere tutte le nostre prove e la stampa degli oggetti che, istante dopo istante, prendono forma seguendo fedelmente le informazioni del nostro programma CAD col quale sono stati disegnati.

sulla piastra in acciaio inossidabile. Facciamo attenzione all'allineamento dei condotti che guidano il passaggio del fondino di ABS lungo tutto il corpo dell'estrusore fino all'ugello della punta riscaldante. Fissiamo i cavi come indicato nella foto.

Montiamo il modulo di controllo sull'estrusore; nella nostra realizzazione abbiamo utilizzato due squadrette aggiuntive come è visibile nella foto; le nuove versioni dei moduli possono essere montati utilizzando i fori e i dadi annegati opportunamente predisposti (Fig.20).

Prima di montare l'estrusore, completamente assemblato, sulla meccanica della Cupcake verifichiamo con un multimetro la correttezza del montaggio delle resistenze sulla testa riscaldante.

Ai capi delle resistenze corazzate dovremo misurare circa 5 ohm mentre ai capi della termo resistenza un valore leggermente inferiore a 100 kohm, dipendente dalla temperatura ambiente; circa 90 kohm a 20 gradi.

Dopo questa verifica possiamo montare il complesso dell'estrusore sul piano di plexiglass dell'asse Z e completare il cablaggio della elettronica.

Collegate i cavi provenienti dal motore

ai punti 1A e 1B del modulo di controllo dell'estrusore facendo attenzione a rispettare la polarità.

Collegate i cavi provenienti dalle resistenze corazzate al connettore OUTPUT della scheda relè.

Collegate con un cavo i punti B+ e B- del modulo dell'estrusore al connettore INPUT della scheda relè.

Infine collegate il cavo proveniente dalla termo resistenza al connettore THERMO del modulo dell'estrusore.

Verificate accuratamente tutto il montaggio prima di passare alla installazione e configurazione del software di gestione della stampante 3D.

Logica di funzionamento

Il processo di produzione di un oggetto con la stampante 3D ha inizio con il disegno dello stesso mediante un software cad 3D in grado di produrre la descrizione dell'oggetto in formato STL (stereolithography). Il formato STL descrive le superfici tridimensionali dell'oggetto come una nuvola di triangoli (Fig. 21).

Una volta ottenuto il formato STL dell'oggetto da stampare è necessario "affettare" orizzontalmente l'oggetto in tanti strati

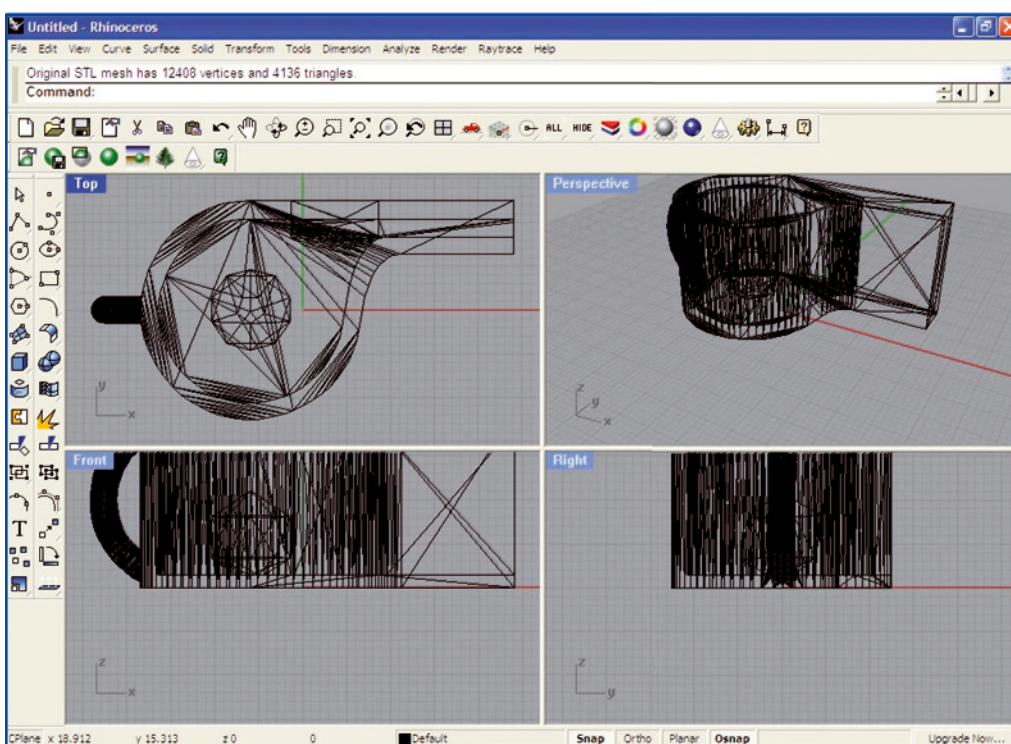


Fig. 21

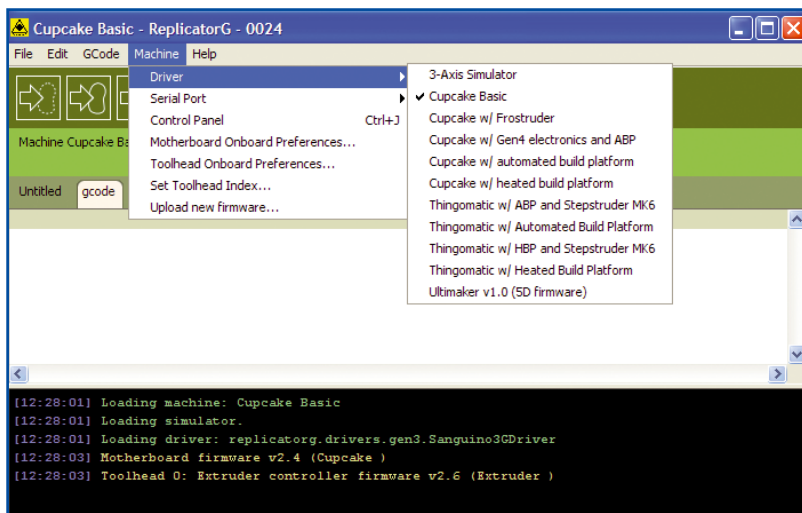


Fig. 22

ognuno dello spessore del filamento di ABS fuso. Ciascuna “fetta” deve poi essere descritta in linguaggio GCODE, che è un linguaggio molto utilizzato per pilotare le macchine CNC.

Il linguaggio GCODE è costituito da un elenco di comandi nel formato Gxxx seguiti da parametri, ciascuno dei quali rappresenta un tipo di comando che deve essere eseguito dalla meccanica della macchina, come quelli specificati nell’esempio seguente:

G1 X5 Y-5 Z6 F3300.0 (Sposta l’utensile alla posizione <x,y,z>=<5,-5,6> alla velocità 3300.0)

G21 (imposta le misure in mm)

G90 (Sposta l’utensile alla posizione zero)

G92 X0 Y0 Z0 (Considera la posizione corrente come posizione zero <x,y,z>=<0,0,0>)

Nel caso della stampante 3D il linguaggio GCODE è stato arricchito con alcuni comandi specifici per gestire la specificità dell’estrusore.

- M104 S220 T0 (imposta la temperatura a 220 gradi)
- M101 (imposta il senso di rotazione del motore in avanti)
- M108 S255 (imposta la velocità del motore a 255)

Entrambe le operazioni vengono eseguite utilizzando l’applicazione Skeinforge, che è già inclusa nel pacchetto ReplicatorG e che utilizzeremo per gestire la stampante.

Una volta ottenuto il file con le istruzioni GCODE non ci resta che farlo eseguire; l’applicativo ReplicatorG invia i comandi GCODE alla Motherboard tramite USB; il firmware della Motherboard interpreta i comandi GCODE trasformandoli in azioni della meccanica.

Il riscaldamento della punta avviene per mezzo delle resistenze corazzate collegate con un relé ad uno dei connettori provenienti dall’alimentatore.

Mediante la lettura della termo resistenza il firmware dell’estrusore mantiene costante la temperatura della punta riscaldante per garantire un filamento di dimensioni e fluidità costante.

Installazione e configurazione del software di gestione della stampante 3D

Come prima operazione è necessario scaricare l’applicativo ReplicatorG, ovvia-

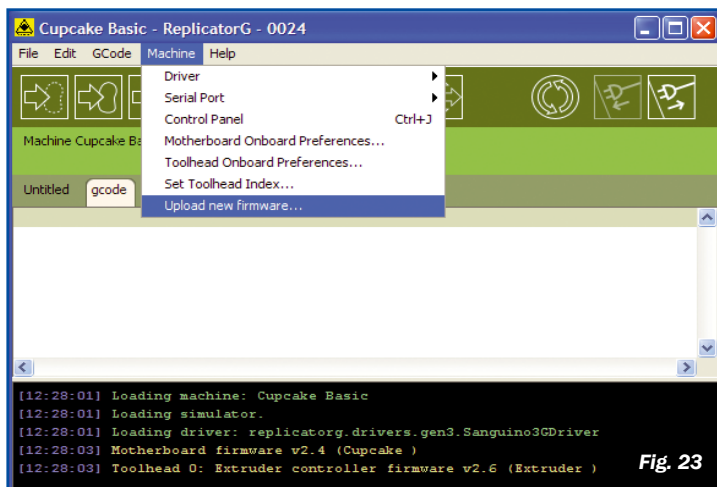


Fig. 23

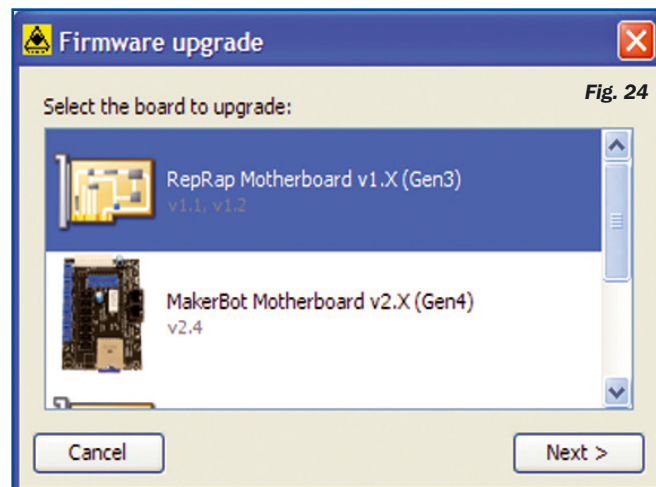


Fig. 24

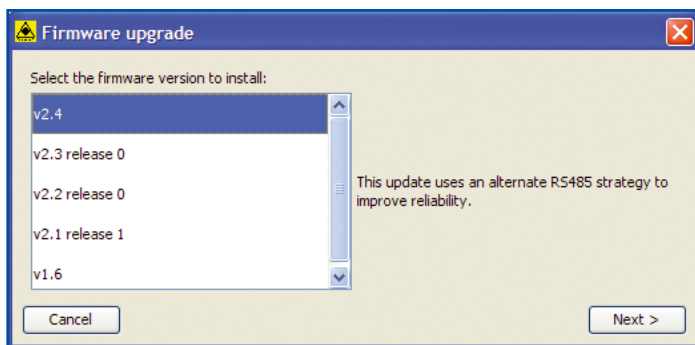


Fig. 25

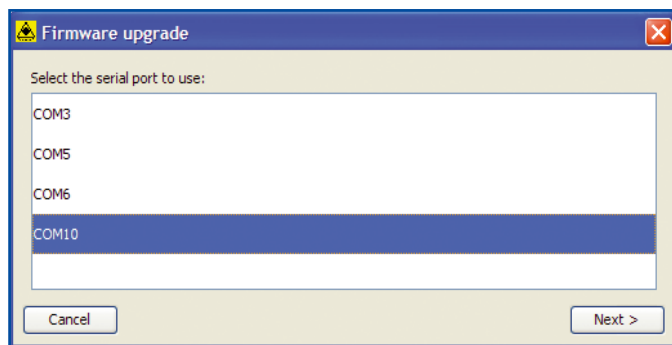


Fig. 26

te open source (<http://replicat.org/>).
Lo trovate all'indirizzo:
<http://replicat.org.googlecode.com/files/replicat.org-0025-windows.zip>
L'applicativo richiede Java 1.5 o successivo e l'ambiente di sviluppo Python 2.6.
Li potete scaricare da <http://java.com/it/> e da <http://replicat.org/installation-windows>.

Non allarmatevi, per l'uso della stampante non è richiesta la conoscenza di alcun linguaggio di programmazione; Python è unicamente un requisito per il funzionamento di alcune delle funzionalità disponibili all'interno dell'applicativo ReplicatorG.
Una volta installato il software la prima operazione da eseguire è il caricamento del firmware sui moduli Motherboard e controllo dell'estrusore.
Collegiamo il cavo USB alla Motherboard ed al nostro PC.
Accendiamo la stampante 3D usando l'interruttore dell'alimentatore più quello in basso sulla Motherboard.
ReplicatorG dovrebbe riconoscere la porta COM e connettersi alla stampante in automatico.

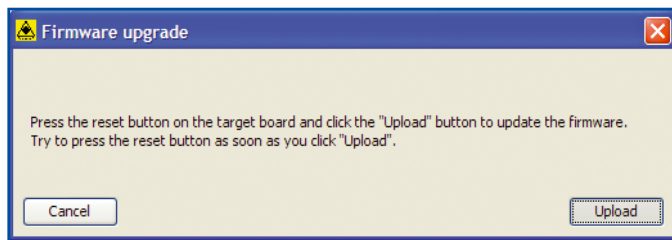


Fig. 27

Dal menu Machine impostate il tipo di meccanica -> Cupcake Basic (Fig. 22).
Ora eseguite l'upload del firmware sulla Motherboard. Dal menu Machine selezionate Upload new firmware (Fig. 23). Nella finestra impostate Rep Rap Motherboard v.1.x (Gen.3) e premete Next (Fig. 24). Selezionate la versione di firmware 2.4, ancora Next (Fig. 25) e selezionate la porta COM (Fig. 26). Un ultimo Next e apparirà la finestra di upload. Per eseguire l'upload, subito dopo avere premuto il bottone Upload bisogna premere il pulsante di Reset sulla Motherboard e rilasciarlo. Se l'upload dà errore ripetete l'operazione (Fig. 27, 28,29); possono essere necessari diversi tentativi per raggiungere il sin-

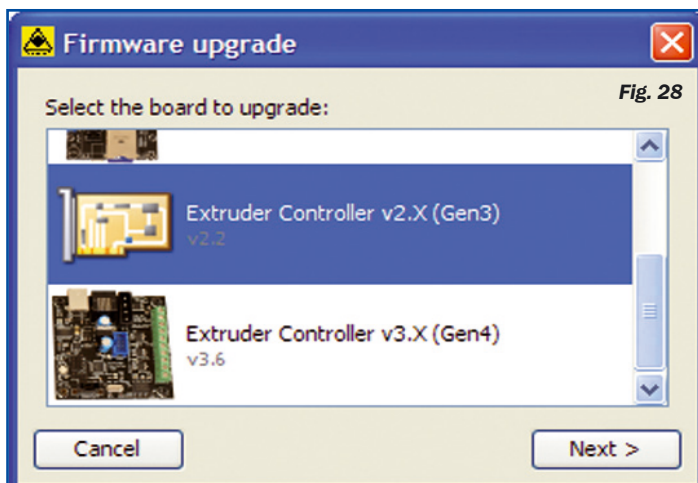


Fig. 28

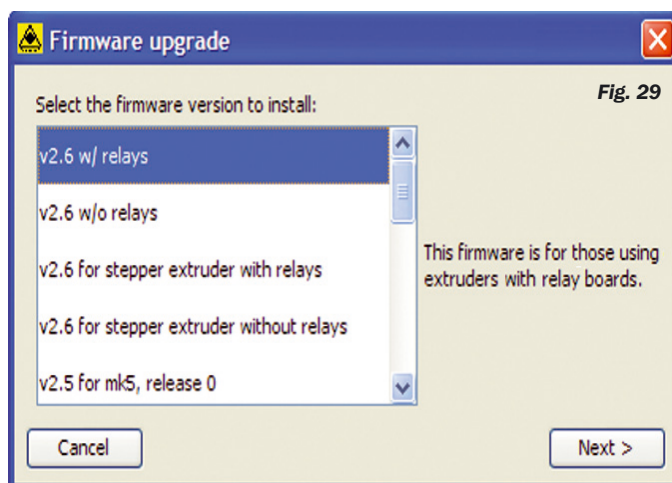


Fig. 29

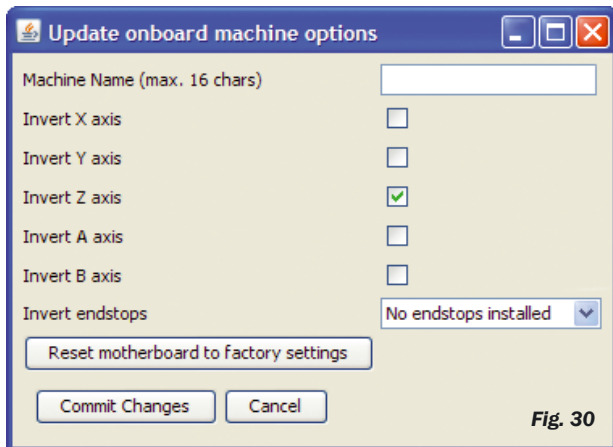


Fig. 30

cronismo giusto: così è la vita.

Spegnete la stampante, spostate il cavo FTDI -> USB dalla Motherboard al connettore FTDI del modulo di controllo dell'estrusore, riaccendete e ricominciate la procedura di Upload del firmware per il modulo dell'estrusore.

Dal menù Machine selezionate Upload new firmware. Nella finestra impostate Extruder controller v. 2.x (Gen.3) e premete Next. Selezionate la versione di firmware v 2.6 with Relays, ancora Next e selezionate la porta COM. Ancora un ultimo Next e apparirà la finestra di upload. Anche qui per eseguire l'upload, subito dopo avere premuto il bottone Upload bisogna premere il pulsante di Reset sulla Motherboard e rilasciarlo. Se l'upload dà errore ripetete l'operazione. Dal menu Machine -> Motherboard Onbo-

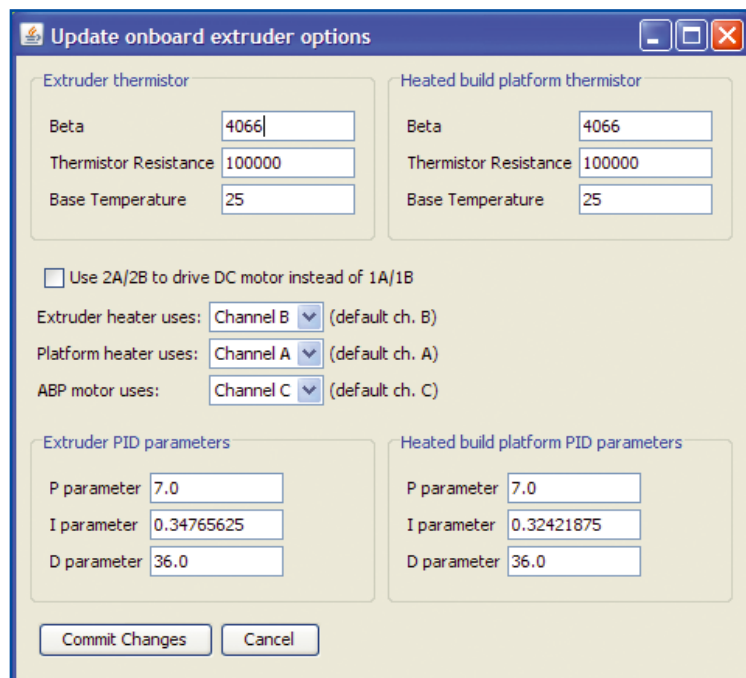


Fig. 31

ard Preferences verificate che sia selezionata l'opzione Invert Z axis (Fig. 30).

Dal menu Machine -> Toolhead Onboard Preferences impostate i parametri come li vedete nella Fig. 31.

Nella finestra dei settaggi vedete una serie di parametri, i più importanti sono il valore della termo resistenza nel gruppo "Extruder thermistor" nel nostro caso 100 kohm e i parametri di calibrazione impostati nel gruppo "Extruder PID parameters" che calibrano il meccanismo di mantenimento

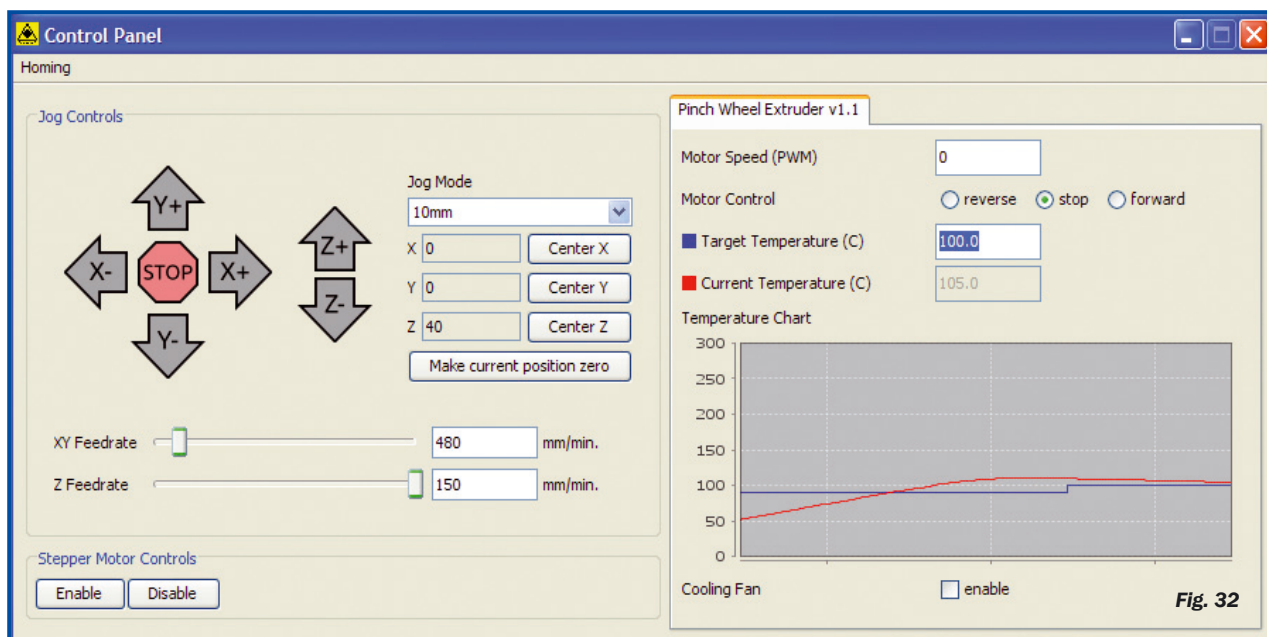


Fig. 32

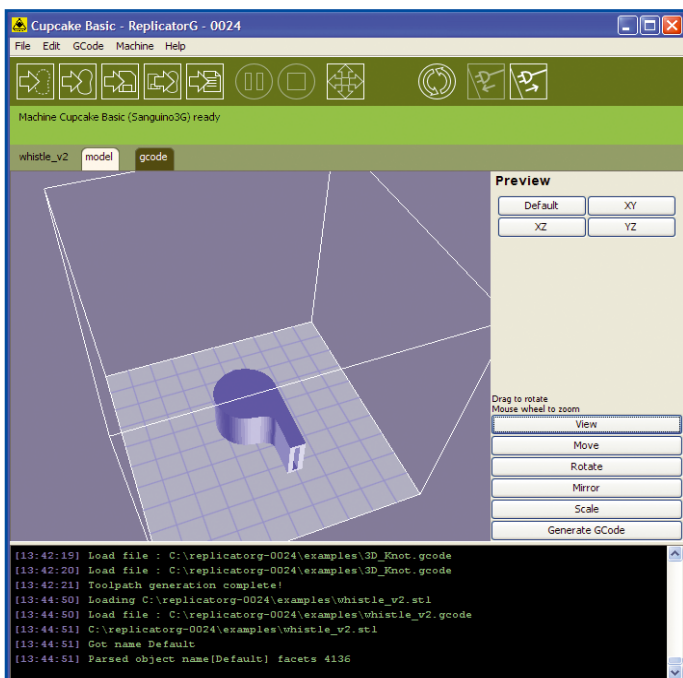


Fig. 33

della temperatura anticipato nel paragrafo precedente.

Spegnete di nuovo, ricollegate il cavo FTDI alla Motherboard e riaccendete.

Adesso è possibile eseguire un test sul funzionamento generale della macchina. Aprite il Control Panel dal menu Machine -> Control Panel o premendo il pulsante a quattro frecce sulla barra in alto. Premendo le frecce dei vari assi il piatto di stampa e l'asse Z si devono muovere di conseguenza; attenti a non arrivare a fondo corsa.

Per calibrare gli assi portare il piatto di stampa al centro e selezionare "Make current position zero" (Fig. 32).

Ora inserite il tondino di ABS nella sua guida sino in fondo, poi stringete un po' le vite di regolazione del cuneo di pressione.

Alzate l'asse Z fino quasi in cima e provate il funzionamento della punta riscaldante.

A destra nella finestra sono presenti due campi, quello segnato con un quadratino blu permette di impostare una temperatura, quello sotto, in rosso, evidenzia la temperatura misurata dalla termo resistenza. Inizialmente impostate una temperatura di sicurezza, per esempio 50 gradi. Selezionate lo stop nel blocco "Motor control"; sentirete scattare il relè. Il grafico sotto la linea blu indica la temperatura impostata mentre quella rossa la temperatura misurata sulla punta riscaldante.

La linea rossa sale man mano sino a rag-

giungere la blu, a questo punto il relè scatta di nuovo interrompendo la tensione alle resistenze corazzate, quando la temperatura scende il relè viene riattivato di nuovo.

A passi successivi, verificando che tutto continui a funzionare, portate la temperatura fino a 220 gradi, che è la temperatura di fusione dell'ABS.

Una volta raggiunta e stabilizzata la temperatura impostate 255 nel campo "Motor Speed (PWM)" e selezionate "forward".

Fig. 34

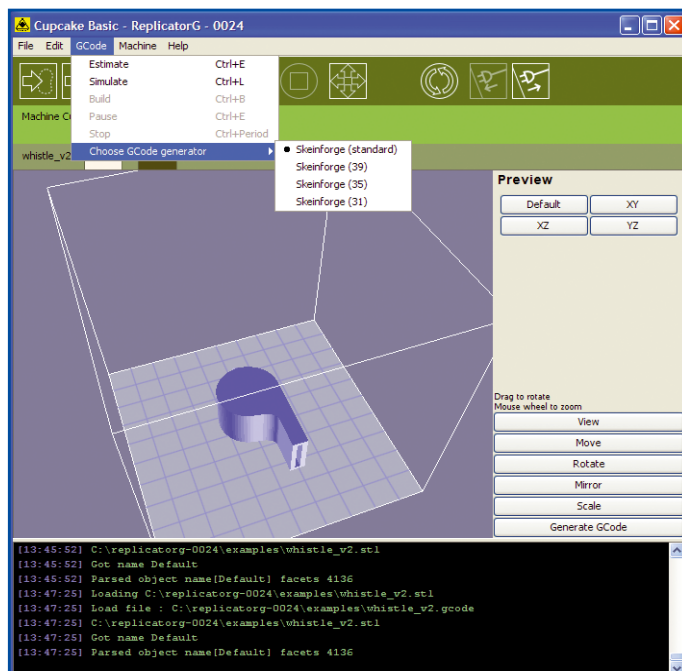
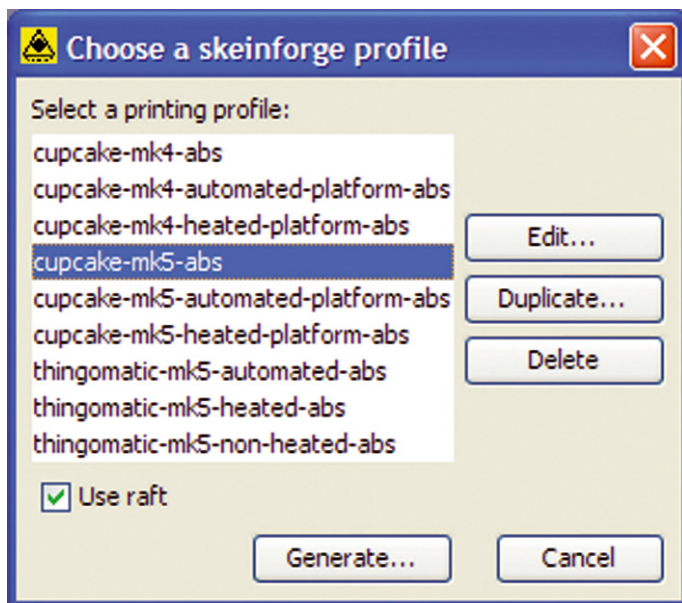


Fig. 35

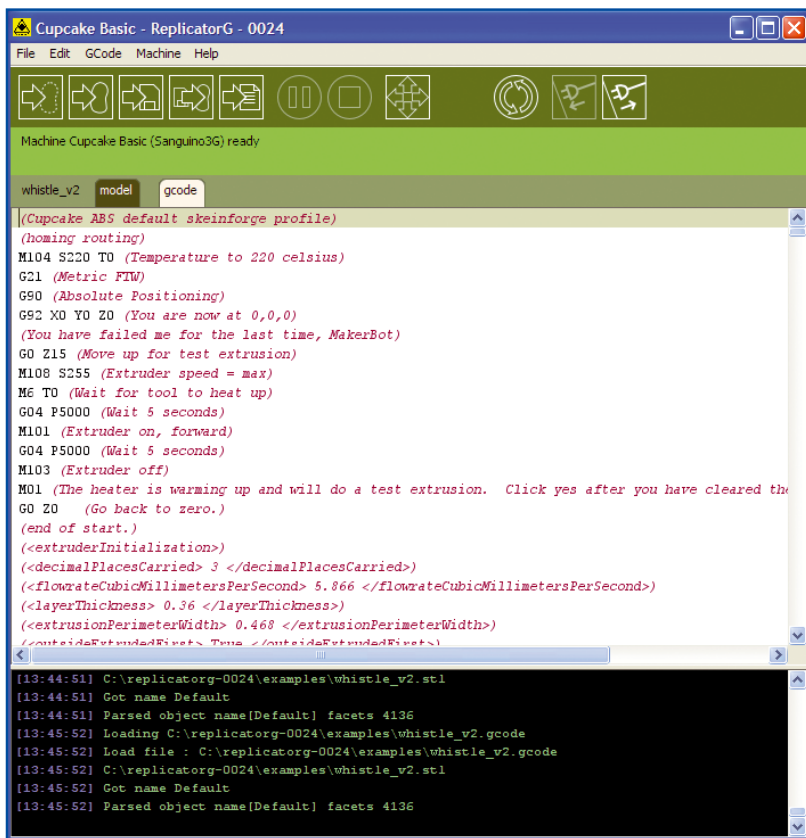


Fig. 36

Un sottile filamento di ABS fuso dovrebbe uscire dalla punta dell'estrusore. Attenzione che scotta. Facciamolo funzionare per un po' e regolate il cuneo di pressione per la migliore condizione, il tondino di ABS deve essere trascinato senza scivolare ma



Fig. 37

senza essere pressato eccessivamente.

Riportate la temperatura a zero e selezionate "stop" al motore.

Siete pronti per produrre il primo oggetto.

Per facilitare l'adesione del pezzo da stampare al piatto di stampa, abbiamo posizionato su quest'ultimo un quadrato ricavato da un comune piatto di plastica fissato con nastro biadesivo su tutta la superficie. La parziale fusione della plastica durante la stampa permette un'ottimale adesione del pezzo; il piatto potrà essere riutilizzato più volte (Fig. 37).

Staccate il filamento prodotto e portate a zero l'asse Z, la punta dell'estrusore deve essere distante circa mezzo millimetro dal piatto di stampa. Azzerate la posizione degli assi. Da File -> Open scegliete un oggetto nella Cartella Examples. Per esempio il fischietto whistle_v2.stl (Fig. 33).

Si apre una anteprima dell'oggetto che verrà stampato. Con i pulsanti a destra è possibile vedere l'oggetto con orientamenti diversi e scalarne le dimensioni.

Da Gcode -> Choose Gcode Generator selezionate Skeinforge (Standard), Fig. 35. Selezioniamo Cupcake-mk5-abs nella finestra successiva (Fig. 34).

A destra in basso selezioniamo Generate Gcode.

L'operazione è abbastanza lunga.

Il risultato è il codice Gcode che verrà interpretato dalla logica di controllo della stampante per costruire l'oggetto (Fig. 36).

Ora selezioniamo Gcode -> Build. oppure premiamo l'apposita icona nella barra in alto.

Succederanno le seguenti cose: la stampante solleva l'asse Z di un paio di centimetri e la punta riscaldante comincia a scaldare. Raggiunta la temperatura di 220 gradi, viene emesso un filamento di test che dovete togliere (scotta); poi l'asse Z torna in posizione zero e l'oggetto comincia a formarsi sotto i vostri occhi. Per prima cosa viene realizzata una base di appoggio poi, strato dopo strato, la punta deposita una sorta di tessuto di ABS fuso fino a completare l'oggetto. Al termine la punta si alza, lasciate raffreddare un po' l'oggetto realizzato per evitare di deformarlo; prelevatelo dalla piastra di stampa, staccatelo dalla base di supporto ... et voilà. Pronto all'uso. □

La filosofia open source prende il volo col primo lancio di un vettore suborbitale

Una comunità di appassionati danesi, la Copenhagen Suborbitals, ha costruito e lanciato con successo un razzo in cima al quale era fissata una piccola capsula con un manichino. È la prima volta che un gruppo amatoriale effettua un esperimento del genere, solitamente riservato ad enti governativi o grandi gruppi industriali del settore aerospaziale. La comunità nata attorno a questo progetto non ha fini di lucro, si autofinanzia e nasce con l'intento di condividere quanto più possibile le informazioni tecniche e le esperienze in campo spaziale in modo da stimolare e coinvolgere il maggior numero di persone in questo tipo di attività. Il team è guidato da Kristian von Bengtson, un progettista che in passato ha lavorato anche per la NASA e da Peter Madsen, specializ-

zato nella progettazione di sottomarini. Attualmente al progetto lavorano direttamente un'altra dozzina di persone con competenze in vari campi, dall'elettronica, all'informatica alla motoristica. La comunità di sostenitori è invece composta da più di 2.000 persone e da alcune società private: complessivamente sono stati raccolti oltre 50.000 euro per la realizzazione del primo prototipo. L'obiettivo finale è quello di realizzare un razzo più grande con una capsula in grado di ospitare una persona e compiere un volo suborbitale completo superando un'altezza di 100 km. Il razzo utilizzato per questo primo esperimento è stato battezzato HEAT-1X, è alto 9 metri e pesa circa 1,6 tonnellate; la capsula posta in cima si chiama Tycho Brahe, è lunga 3,5 metri ed ha un



diametro di 64 centimetri: in questo primo volo conteneva un manichino mentre per il futuro dovrebbe ospitare lo stesso Peter Madsen. La capsula dispone di una calotta trasparente per poter controllare visivamente il volo ed effettuare riprese video. Per il lancio è stata utilizzata una piattaforma galleggiante, una specie di catamarano battezzato Sputnik.

www.copenhagensuborbitals.com

Primo volo internazionale per Solar Impulse

Dopo un volo della durata di 12 ore e 59 minuti, non usando carburante ma unicamente l'energia del sole, Solar Impulse HB-SIA ha ultimato il suo primo volo internazionale atterrando a Bruxelles, la capitale della Comunità Europea, alle 21:39 accolto da una folla di sostenitori e dal team di Solar Impulse guidato da Bertrand Piccard che ha salutato per primo il pilota André Borschberg che era ai comandi del velivolo.

Solar Impulse è decollato questa mattina alle 08:40 dall'aeroporto di Pa-

yerne, in Svizzera, ed ha poi seguito il piano di volo prestabilito. L'aereo ha attraversato l'Alsazia verso Nancy e Metz, prima di sorvolare il Granducato del Lussemburgo per poi arrivare in Belgio ed atterrare sulla pista 02 dell'Aeroporto Internazionale di Bruxelles-Zaventem.

Grazie al sistema di comunicazione satellitare di bordo sviluppato da Swiscom, il direttore di volo Raymond Clerc e il suo team sono stati in grado di supportare il pilota durante tutte le fasi del volo dal "Mission Control Center" di Payerne.

Durante il tragitto il velivolo ha volato ad un'altezza media di 1.800 metri per

evitare il normale traffico aereo commerciale che sfrutta corridoi ad un'altezza di 8.000-10.000 metri; solamente nelle aree aeroportuali Solar Impulse ha rischiato di trovarsi vicino ad aerei commerciali che con le loro turbolenze – dovute alle dimensioni ed alla velocità – avrebbero potuto disturbare il volo di questo enorme ma fragile velivolo che viaggia a circa 50 km/h e che ha una scarsissima manovrabilità.

Da Bruxelles Solar Impulse raggiungerà Parigi dove sarà l'ospite d'onore del prossimo 49° Salone Internazionale dell'Aeronautica.

www.solarimpulse.com



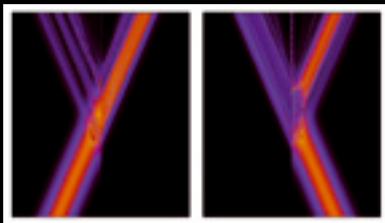
Quando le onde sono a senso unico

Potrebbe preludere ad applicazioni nel campo dei computer e dell'acustica la scoperta, pubblicata su "Physical Review Letters" di due ricercatori dell'Istituto dei sistemi complessi del Cnr di Firenze e dell'Università dell'Insubria (Como) che hanno dimostrato teoricamente che è possibile creare dei percorsi "a senso unico" per la luce e il suono, esplorando le nuove frontiere della fisica non lineare.

Normalmente la luce non ha una direzione di propagazione preferenziale e si può trasmettere con uguale intensità, per esempio, da destra a sinistra o viceversa. Ma esistono materiali detti "non lineari" che, al contrario di quelli tradizionali, ne permettono il passaggio da alcune direzioni e non da altre, producendo un percorso "a senso unico" per le onde luminose.

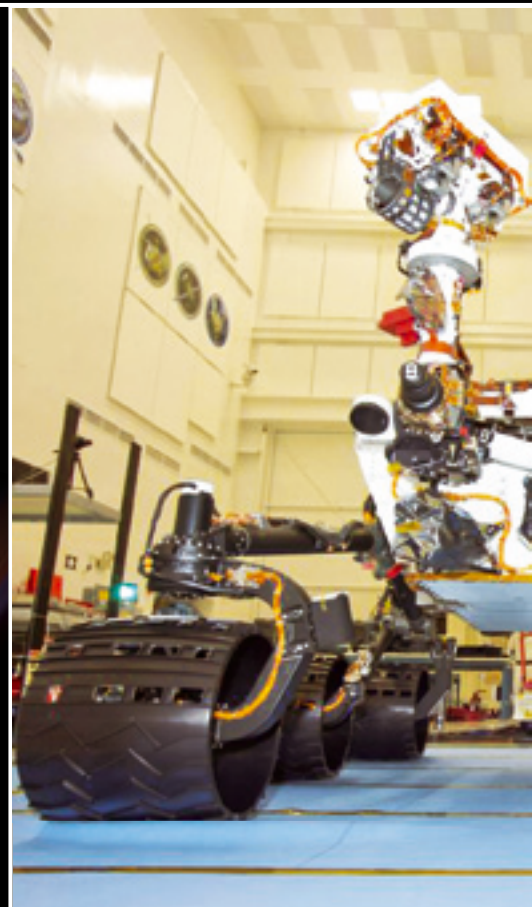
Questa possibilità è stata prevista teoricamente da Stefano Lepri, ricercatore dell'Istituto dei sistemi complessi del Consiglio nazionale delle ricerche di Fi-

renze (Isc-Cnr), e Giulio Casati, professore presso il Centro per i sistemi complessi dell'Università dell'Insubria: da un loro studio, sono scaturite speculazioni teoriche secondo le quali sarebbe possibile ottenere questo risultato con altri tipi di onde, dunque anche con il suono. La ricerca prelude al potenziale utilizzo di questi materiali per applicazioni ottiche e acustiche in diversi campi, dai computer quantistici all'insonorizzazione ambientale.



Il carattere rivoluzionario dello studio sta nel fatto che la possibilità di trasmissione a senso unico delle onde è negata dal teorema della reciprocità, secondo il quale le onde che attraversano un materiale in una direzione non potranno in nessun caso essere 'bloccate' nella direzione opposta. "Per superare questo limite dobbiamo andare oltre le concezioni classiche della fisica lineare", conclude il ricercatore.

www.cnr.it



Le bellissime immagini della ISS riprese da Paolo Nespoli

Sul sito della NASA sono state pubblicate le bellissime immagini della Stazione Spaziale Internazionale (ISS) riprese da Paolo Nespoli il 23 maggio scorso dalla Soyuz, prima del rientro a terra con la missione STS-134, dopo quasi sei mesi a bordo della ISS.

Le immagini dello Shuttle attaccato alla Stazione Spaziale Internazionale sono uniche non solo per la loro bellezza, ma perché è la prima volta che uno Shuttle si trova ancora attaccato alla ISS mentre la Soyuz parte. Ed è da questa prospettiva che l'astronauta dell'ESA Paolo Nespoli, ha scattato le foto, grazie anche alla collaborazione degli altri due membri dell'equipaggio della Soyuz, Cady Coleman e Dmitry Kondratev, dell'equipag-

gio a bordo della ISS e del Centro di Controllo a terra.

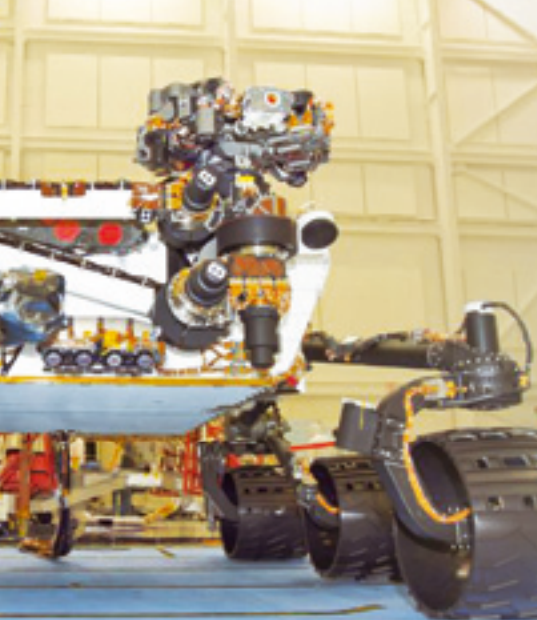
Durante il volo di rientro, quando la capsula si trovava a circa 600 piedi dalla Stazione, il Centro di Controllo di Mosca, situato subito fuori la capitale, ha guidato la Stazione Spaziale Internazionale per farla ruotare di 130 gradi. Questa manovra ha permesso a Nespoli di catturare immagini digitali e video in alta risoluzione.

Nelle immagini si riconoscono lo Shuttle Endeavour, che ha portato sulla ISS un altro italiano, Roberto Vittori, ed il modulo di trasferimento automatico ATV. Endeavour si è staccato il 30 maggio per atterrare il primo giugno.

www.nasa.org



Ultimi test per Curiosity prima della partenza per Marte



Al Jet Propulsion Laboratory di Pasadena si stanno ultimando gli ultimi test di Mars Science Laboratory (MSL), meglio noto come Curiosity, il rover che entro quest'anno verrà lanciato verso Marte con l'obiettivo di individuare la zona ottimale per lo sbarco delle future missioni umane sul pianeta rosso. In questi giorni sono state pubblicate sul sito del JPL le immagini relative ai test di mobilità che si sono svolti lo scorso 3 giugno presso lo Spacecraft Assembly Facility. A breve il rover verrà inviato a Cape Canaveral da dove partirà per il lungo viaggio che lo porterà sul pianeta rosso: il lancio è programmato per la fine di novembre di quest'anno.

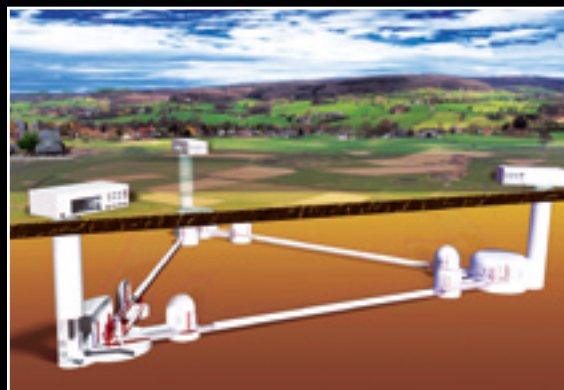
Dopo le missioni dei rover Sojourner del 1997 (rimasto attivo per un mese soltanto) e di Spirit e Opportunity del 2004 (Opportunity è ancora attivo mentre Spirit ha cessato di funzionare da poco), entro i primi mesi del 2012 su Marte dovrebbe arrivare

Curiosity, un rover molto più grande dei precedenti e dotato di strumentazione molto più avanzata. La durata minima della missione dovrebbe essere di almeno due anni ma, visti i precedenti di Spirit e Opportunity, gli scienziati sperano in una vita operativa molto più lunga. Curiosity pesa circa 800 kg, trasporta 65 kg di strumenti scientifici ed è in grado di muoversi ad una velocità di 90 metri all'ora. Tra gli strumenti più interessanti presenti a bordo di Curiosity un laser di potenza in grado di vaporizzare un pezzettino di roccia ad alcuni metri di distanza. Un'apposita telecamera sarà in grado di determinare il tipo di materiale roccioso analizzando lo spettro della luce emessa dalla roccia vaporizzata. Sviluppato dal Los Alamos National Laboratory e dal CESR Laboratory, il laser infrarosso irradia impulsi a 1067 nm della durata di 5 ns con una densità di potenza pari a 1 GW/cm², generando un'energia di 14 mJ.

<http://marsprogram.jpl.nasa.gov>

Einstein Telescope (ET) è il nome scelto per uno dei progetti più ambiziosi dei fisici europei dopo l'acceleratore LHC di Ginevra: un enorme telescopio a forma di triangolo con un perimetro di 30 chilometri realizzato sottoterra per captare le onde gravitazionali, increspature dello spaziotempo che si propagano alla velocità della luce, generate dai "cataclismi cosmici", come l'esplosione di supernovae o lo scontro tra buchi neri. Prodotte da masse stellari in movimento accelerato, le onde gravitazionali hanno un'ampiezza infinitesima. Per questa loro caratteristica rivelarle è davvero difficile, così difficile che non ci siamo ancora riusciti, cosicché oggi rappresentano il tassello mancante alla verifica sperimentale della Teoria della Relatività Generale di Albert Einstein. Per intercettarle stiamo costruendo strumenti sempre più sofisticati di cui ET rappresenterà la prossima frontiera. "Grazie alla sua estrema sensibilità alle basse frequenze, ET ci permetterà di osservare con regolarità le onde gravitazionali inaugurando così l'era di un nuovo tipo di astronomia, l'astronomia gravita-

ET, telescopio gigante per ascoltare i suoni dello spazio tempo



zionale: sarà una vera e propria rivoluzione scientifica", spiega Michele Punturo, dell'Istituto Nazionale di Fisica Nucleare (INFN) di Perugia e coordinatore scientifico dello studio di progetto che la comunità dei ricercatori europei ha presentato recentemente. "ET è sostenuto dalla Commissione Europea e inserito nel VII Programma Quadro: questo testimonia il suo valore nell'ambito della ricerca fondamentale e tecnologica", sottolinea Federico Ferrini, direttore di EGO e coordinatore del progetto di studio per ET. "ET - prosegue Ferrini - si propone come un progetto comune che riunisca tutti gli

scienziati europei impegnati nello studio delle onde gravitazionali e rappresenterà per loro uno strumento di indagine completamente nuovo". "Dopo tre anni di lavoro, la comunità scientifica impegnata nello studio delle onde gravitazionali, che comprende più di 200 scienziati provenienti dall'Europa e dal resto del mondo, presenta lo studio sull'Einstein Telescope, un progetto che apre la strada alla scoperta di un universo ancora sconosciuto", aggiunge Fulvio Ricci della sezione INFN di Roma 1 La Sapienza.

www.infn.it

Un secondo? Basta per inviare i dati contenuti in 700 DVD!

Gli scienziati del Karlsruhe Institute of Technology (KIT) sono riusciti a trasmettere mediante un raggio laser una quantità di dati impressionante: ben 26 terabit al secondo, pari alle informazioni contenute in 700 DVD. Si tratta del volume di dati più grande mai trasportato tramite un raggio laser. La trasmissione è avvenuta ad una distanza di 50 chilometri. Con questo esperimento, il team di ricercatori del KIT guidati dal professor Jürg Leuthold è riuscito a superare il proprio precedente record stabilito nel 2010 quando, per la prima volta, era stata superata la soglia dei 10 terabit al secondo, vale a dire una velocità di trasmissione di 10.000 miliardi di bit al secondo. Questo successo è dovuto ad un nuovo processo di elabora-

zione dei dati che prevede una decodifica opto-elettrica. Il team di Leuthold applica per la codifica dei dati la tecnologia OFDM (orthogonal frequency division multiplexing); per molti anni questo processo – che si basa sull'algoritmo della trasformata di Fourier veloce – è stato usato con successo nelle comunicazioni mobili. Sempre più spesso Internet viene utilizzato per l'invio di video che richiedono una elevata velocità di trasmissione: banda e velocità si fanno sempre più necessarie. Linee da 100 gigabit al secondo (pari a 0,1 terabit al secondo) sono ormai considerate indispensabili per qualsiasi

tipo di utente ed attualmente la ricerca si concentra su linee di trasmissione in grado di operare tra 400 Gigabit/s a 1 Tbit/s. Gli studi degli scienziati di Karlsruhe sono quindi più che mai importanti. Società e centri di ricerca di tutto il mondo stanno lavorando insieme al team del professor Leuthold per un ulteriore miglioramento della tecnologia e per una sua rapida industrializzazione; tra queste: Agilent, Micram Deutschland, Time-Bandwidth Svizzera, Finisar Israele, e l'Università di Southampton.

www.kit.edu



Al via la missione del cacciatore di antimateria

Lo shuttle Endeavour ha lasciato la Terra per la sua ultima missione: a bordo porta un prezioso carico: il rivelatore di particelle AMS, che sarà collocato dagli astronauti sulla Stazione Spaziale Internazionale.

Il cacciatore di antimateria, come è stato soprannominato AMS (Alpha Ma-

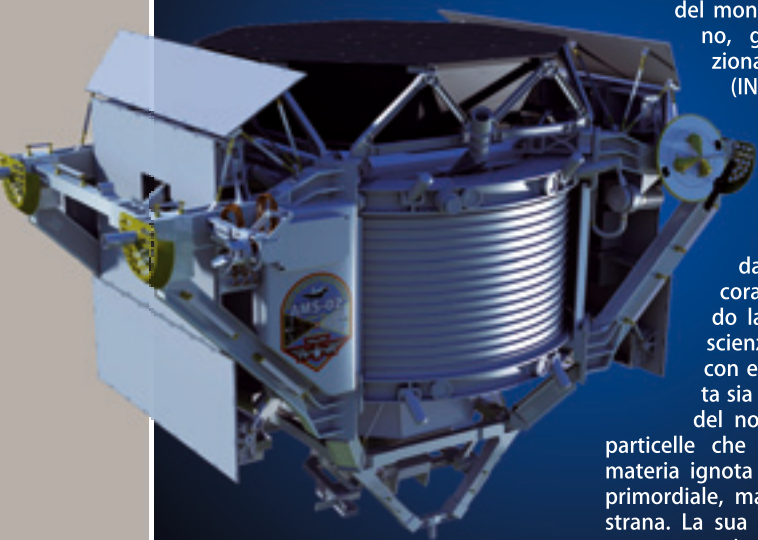
gnetic Spectrometer), è partito con lo shuttle Endeavour dal Kennedy Space Center di Cape Canaveral e inizia così la sua missione scientifica. È il primo esperimento di Big Science a operare sulla Stazione Spaziale Internazionale (ISS), dove sarà collocato dagli astronauti tra pochi giorni, ed è stato realizzato con il fondamentale contributo del mondo della ricerca italiana, grazie all'Istituto Nazionale di Fisica Nucleare (INFN) e all'Agenzia Spaziale Italiana (ASI).

La missione di AMS è scovare le tracce delle particelle più misteriose. A queste particelle, previste dalla teoria ma non ancora rivelate, stanno dando la caccia da tempo gli scienziati di tutto il mondo, con esperimenti sia in orbita sia nei luoghi più estremi del nostro pianeta. Sono le particelle che compongono questa materia ignota chiamata antimateria primordiale, materia oscura, materia strana. La sua scoperta aprirebbe le porte a nuovi scenari della conoscen-

za. Potrebbe portare alla luce l'esistenza di antistelle e antigalassie. Svelarci la natura della materia oscura che dovrebbe costituire circa un quarto di tutto l'universo, una quantità cinque volte superiore a quella della materia ordinaria di cui è fatto ciò che oggi noi conosciamo. Mostrarci un "mondo" di una materia ultradensa composta di quark strange, anziché di quark up e down come quelli che compongono la materia ordinaria. Questa "altra" materia potrebbe costituire un altro universo, diverso da quello oggi noto. Una volta collocato a bordo della ISS, AMS continuerà ad operare per almeno 10 anni, raccogliendo dati e inviando a terra un'enorme quantità di informazioni veicolate da 300 mila canali, pari a tutti quelli della Stazione Spaziale. Questi dati, che complessivamente ammontano a circa 100 Terabyte di memoria, saranno gestiti da grandi centri di calcolo anche in Italia per poi essere studiati dai ricercatori di tutto il mondo.

I primi risultati di fisica si attendono già entro i primi 7 mesi dalla ricezione dei dati.

www.infn.it





dell'ing.
MIRCO
SEGATELLO

Conoscere e usare Arduino

Conosciamo ed impariamo ad usare un nuovo shield: quello che permette di dotare il nostro Arduino della connettività Bluetooth. Undicesima puntata.

gni giorno o quasi, nascono shield dedicati a realizzare con Arduino tutte le funzioni che già conosciamo e vediamo associate a dispositivi di uso comune. In questa puntata impareremo ad utilizzare quello creato e sviluppato per implementare il Bluetooth. Per la precisione, vogliamo interfacciare una scheda Arduino ad un PC tramite un collegamento wireless utilizzando lo standard Bluetooth, già impiegato con successo nella comunicazione locale tra cellulari, negli auricolari e vivavoce wireless per automobile, nella strumentazione, ma anche nelle periferiche del computer e nei lettori di codice a barre senza fili. Come vedremo, dotare Arduino di

un'interfaccia Bluetooth risulterà assai semplice e per niente costoso; sarà inoltre una valida soluzione per comunicazione wireless, alternativa a quella rappresentata dai moduli radio XBee o WiFi. Come sempre, in questo corso, partiamo dalla descrizione dell'hardware, per poi proseguire con lo sviluppo della parte software.

In sintesi, gli elementi che abbiamo utilizzato per questa puntata, sono i seguenti:

- una scheda Arduino duemilanove (va bene anche Arduino UNO);
- una XBee shield, che è la versione prodotta dalla Libelium (www.libelium.com);
- un modulo Bluetooth Bee della Seedstudio

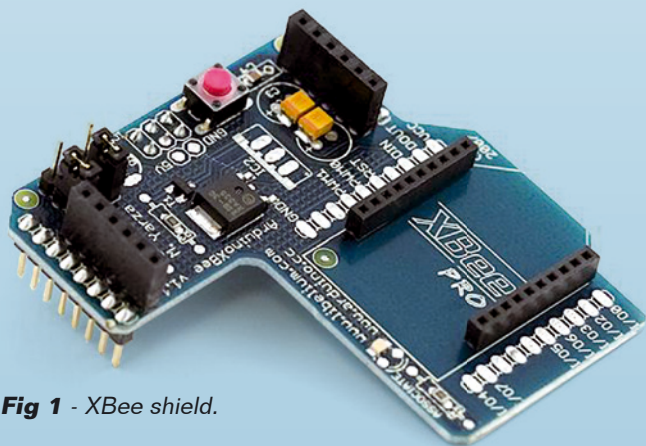


Fig 1 - XBee shield.

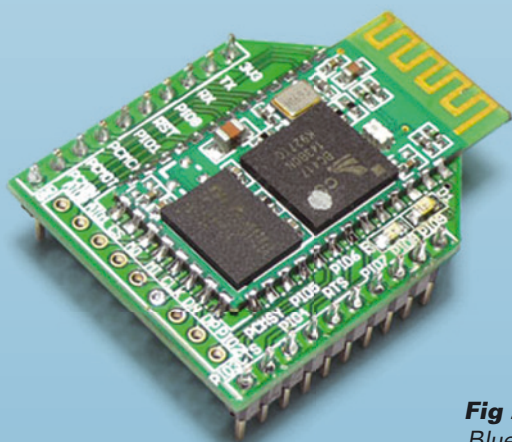


Fig 2 - Modulo Bluetooth Bee.

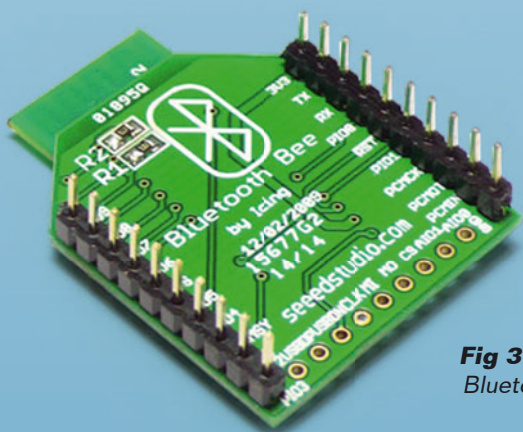


Fig 3 - Modulo Bluetooth Bee.

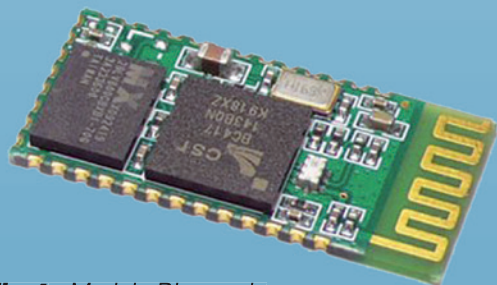


Fig 4 - Modulo Bluetooth.

(www.seeedstudio.com) fornito dalla Futura Elettronica (codice 7300-BLUETBEE).

È possibile acquistare a parte il solo modulo Bluetooth (Futura Elettronica, codice 7300-BLUETOOTHMOD) ma, viste le ridotte dimensioni, il suo utilizzo non è per niente agevole. Pur costando di più, consigliamo l'utilizzo del modulo già cablato sulla piccola basetta XBee, facilmente utilizzabile con la sua XBee shield.

Vediamo ora brevemente le caratteristiche di questo ricetrasmittitore Bluetooth appositamente progettato per realizzare collegamenti trasparenti tra due apparecchiature. Questo modulo contiene un ricetrasmittitore funzionante nella banda ISM (ricordiamo che lo standard Bluetooth riguarda connessioni effettuate alla frequenza di 2,4 GHz) ed è in grado di gestire collegamenti seriali sino a 460.800 bps.

È compatibile con le specifiche Bluetooth v2.0 + EDR, con una potenza in trasmissione di 4 dBm (Classe 2) che gli consente collegamenti sino a circa 10 metri.

Il modulo UART interno ha il baud-rate programmabile e la gestione del transito dati è asservita ad un sistema di crittografia; l'alimentazione è a 3,3 volt e dispone di un'antenna integrata nel PCB.

Il modulo UART interno è configurato, per impostazione predefinita, per una comunicazione a 38.400 baud con 8 bit di dati, 1 bit di stop, nessuna parità, nessun controllo di flusso; sono comunque supportate le seguenti velocità di trasferimento: 9.600, 19.200, 38.400, 57.600, 115.200, 230.400, 460.800.

Sono disponibili le linee CTS e RTS per il controllo del flusso dati e due LED (uno verde e uno rosso) per la segnalazione dello stato di funzionamento.

Quando il modulo si trova nello stato "sconnesso", si vedrà il solo LED verde lampeggiare 2 volte al secondo. In attesa di collegamento ci saranno sia il LED verde che il LED rosso a lampeggiare una volta la seconda, mentre nello stato "connesso" lampeggerà il solo LED verde una volta al secondo.

Per impostazione predefinita, il modulo è predisposto alla connessione automatica con l'ultimo dispositivo non appena viene acceso; il codice di accesso è impostato a "0000". Per

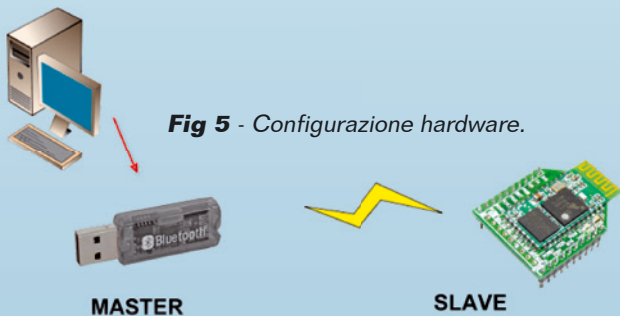


Fig 5 - Configurazione hardware.

“ultimo dispositivo” si intende quello con cui si è collegato prima di essere spento.

Per i dettagli relativi alla funzione dei pin di collegamento, vi rimandiamo alla lettura dei data-sheet del prodotto.

Volendo realizzare un collegamento wireless con un PC, dovremmo necessariamente dotarci di un adattatore Bluetooth (Bluetooth dongle) da inserire in una porta USB libera del PC. Per le nostre verifiche sul campo abbiamo utilizzato il Bluetooth USB dongle prodotto dalla Velleman e distribuito dalla Futura Elettronica con il codice PCUSBBT; esso viene fornito con il software Bluesoleil della IVT Corporation (il più diffuso per far dialogare dispositivi Bluetooth con un com-

puter) al quale faremo riferimento in questa puntata del corso. Il modulo Bluetooth Bee sarà inserito nella XBee shield e a sua volta su Arduino, ma, per poterlo utilizzare, è necessario programmarlo; la programmazione avviene tramite l’invio di semplici comandi seriali. È possibile inserire il modulo Bluetooth Bee in un adattatore USB-XBee, in modo da consentire la programmazione via PC, ma preferiamo sia direttamente Arduino, tramite un apposito sketch, ad inviare le impostazioni. Elenchiamo qui di seguito i comandi a disposizione.

Imposta modalità di funzionamento

`\r\n+STWMOD=0\r\n` Client (slave)
`\r\n+STWMOD=1\r\n` Server (master)

Imposta Baud-rate

`\r\n+STBD=115200\r\n` Imposta baudrate 115200

Baudrate supportati: 9600, 19200, 38400, 57600, 115200, 230400, 460800.

Imposta il nome del dispositivo

`\r\n+STNA=abcdefg` Imposta il nome “abcdefg”

Listato 1

```
/*
BluetoothBee Demo Code
2011 ElettronicaIN

Hardware:
  modulo BluetoothBee
  XBee shield Libelium
  Arduino duemilanove
  USB Bluetooth dongle Velleman

Questo sketch configura il modulo come slave
*/

void setup()
{
  Serial.begin(38400); //Imposta l'UART a 38400 baud
  delay(1000);
  Serial.print("\r\n+STWMOD=0\r\n"); //Imposta il modulo come slave
  Serial.print("\r\n+STNA=Arduino\r\n"); //Assegna il nome "Arduino" al modulo
  delay(2000); // Ritardo necessario per la configurazione.
  Serial.print("\r\n+INQ=1\r\n");
  delay(2000); // Ritardo necessario per abilitare la comunicazione.
}

void loop()
{
  delay(1000);
  int sensorValue = analogRead(A0); //Legge il canale analogico A0
  Serial.println(sensorValue, DEC); //Invia il dato via UART
}
```

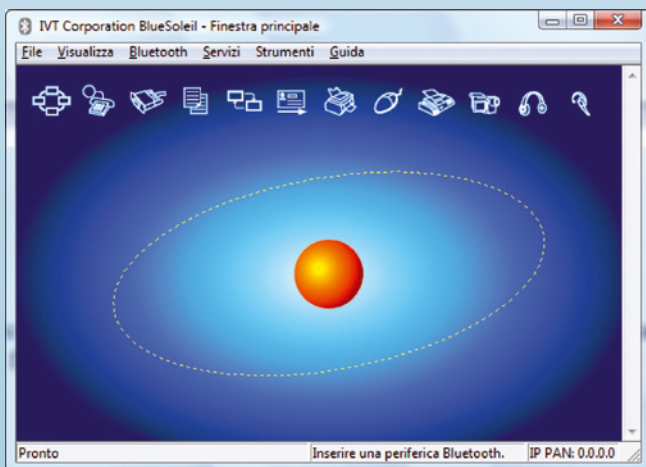


Fig 6 - Schermata di avvio di Bluesoleil.

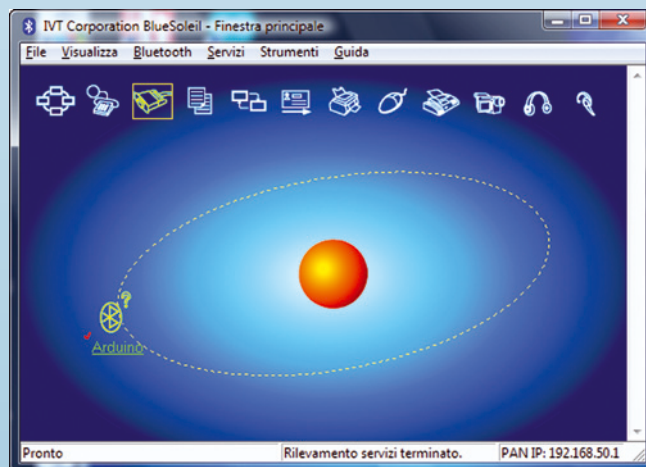


Fig 8 - Ricerca servizi della periferica trovata.

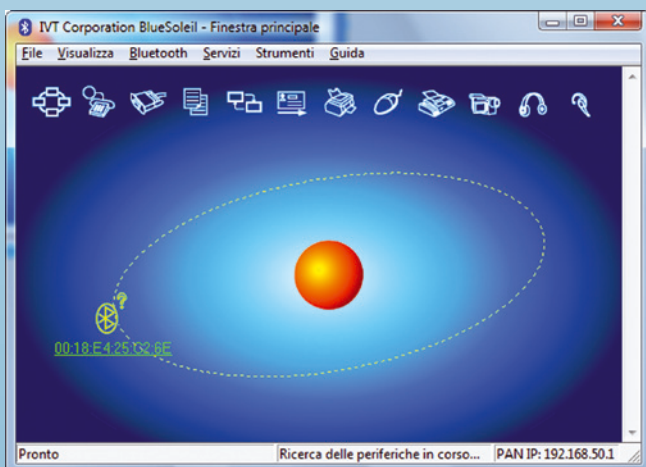


Fig 7 - Ricerca periferiche.

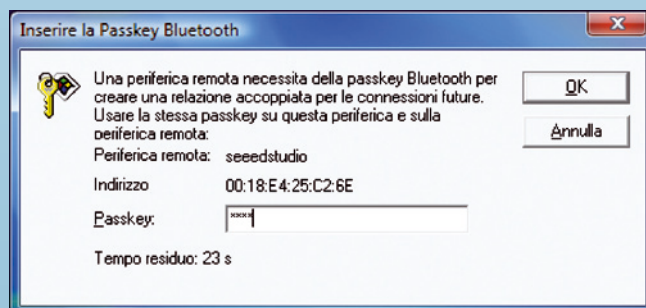


Fig 9 - Inserimento Passkey.

Autoconnessione con l'ultimo dispositivo connesso

`\r\n+STAUTO=0\r\n` Proibito
`\r\n+STAUTO=1\r\n` Permessso

Permette la connessione del dispositivo

`\r\n+STOAUT=0\r\n` Proibito
`\r\n+STOAUT=1\r\n` Permessso

Imposta PIN code (Passkey)

`\r\n+STPIN=2222\r\n` Imposta il PINCODE "2222"

Cancella PIN code (Passkey assegnata dal microcontrollore)

`\r\n+DLPIN\r\n` Cancella PINCODE

Legge l'indirizzo del dispositivo

`\r\n+RTADDR\r\n` Riporta l'indirizzo del dispositivo

Auto riconnessione quando viene perso il segnale con il dispositivo master

`\r\n+LOSSRECONN=0\r\n` Proibito
`\r\n+LOSSRECONN=1\r\n` Permessso

Ricerca moduli

a) Nel caso di modulo configurato come Master:

`\r\n+INQ=0\r\n` Ferma ricerca dispositivi
`\r\n+INQ=1\r\n` Avvia ricerca dispositivi

b) Nel caso di modulo configurato come Slave:

`\r\n+INQ=0\r\n` Ferma visibilità dispositivo
`\r\n+INQ=1\r\n` Avvia visibilità dispositivo

Siamo adesso pronti per scrivere lo sketch occorrente alla gestione della comunicazione in Bluetooth, il quale, una volta caricato su Arduino, configurerà il modulo e, successivamente, andrà a leggere l'ingresso analogico 0 ed invierà al PC i dati acquisiti, ad intervalli di un secondo, via Bluetooth. Il codice corrispondente è quello illustrato nel Listato 1. La configurazione del modulo avviene inviando semplici caratteri seriali nello standard

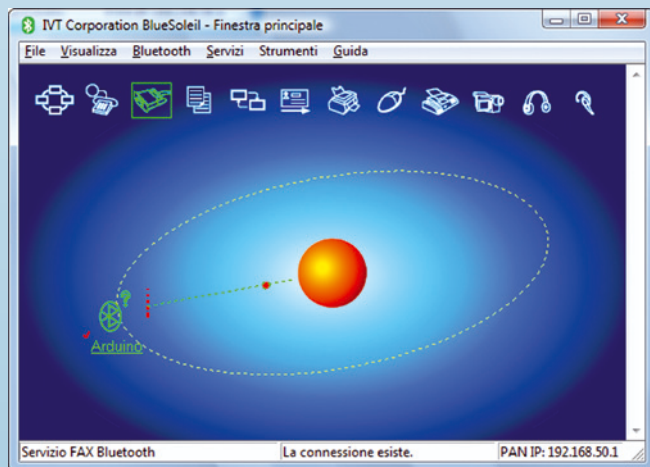


Fig 10 - Connessione stabilita.

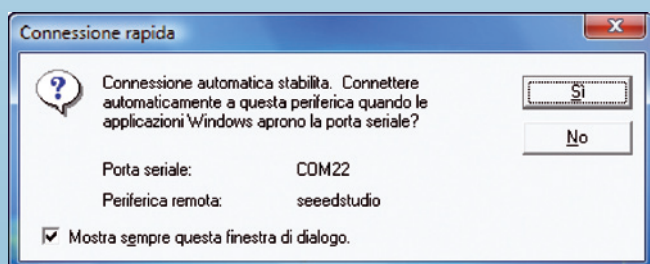


Fig 11 - Porta COM virtuale utilizzata dalla connessione.

specificato nei data-sheet. Il comando "`\r\n+STWMOD=0\r\n`" imposta il modulo come Slave; sarà l'adattatore Bluetooth sul PC a svolgere la funzione di Master.

Il comando "`\r\n+STNA=Arduino\r\n`" assegna al modulo il nome "Arduino"; infine, il comando "`\r\n+INQ=1\r\n`" abilita la visibilità del modulo da parte del Master.

Tutte queste fasi sono chiaramente indicate dallo stato di lampeggio dei LED. Quando il LED verde lampeggia due volte al secondo, significa che il modulo non è connesso; ciò accade, ad esempio, quando alimentate quest'ultimo senza prima averlo configurato.

Quando si abilita la visibilità del modulo, lampeggiano una volta al secondo i LED sia verde che rosso, mentre una volta che viene stabilita una connessione, lampeggia il solo LED verde, sempre con la cadenza di una volta al secondo. Durante la fase di programmazione di Arduino con lo sketch riportato nel **Listato 1** è importante che gli interruttori della XBee shield siano posti in modalità USB. Quindi staccate Arduino, impostate gli interruttori della XBee Shield su XBee ed alimentate la scheda tramite la presa Plug. Questo è neces-

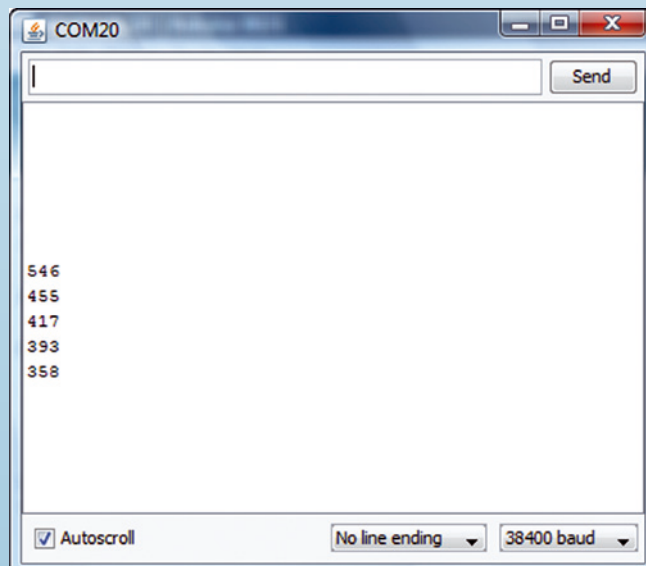


Fig 12 - Dati ricevuti da Serial monitor di Arduino.

sario perché il modulo Bluetooth Bee dialoga con il microcontrollore di Arduino tramite il modulo UART, il quale non deve entrare in conflitto con eventuali segnali giunti via USB. Passati alcuni secondi dall'accensione, il modulo Bluetooth sarà operativo e visibile. Avviate il software Bluesoleil, installato precedentemente, inserite il Bluetooth dongle, attendete venga riconosciuto, quindi selezionate Bluetooth-Rileva periferiche Bluetooth; sul modulo Bluetooth Bee devono lampeggiare entrambi i LED. Cliccate con il pulsante destro sopra l'icona della periferica trovata e, dal menu contestuale cui accedete, attivate la funzione di ricerca servizi. Verrà chiesta la Passkey di accesso alla periferica; il valore predefinito è 0000 (quattro zeri). Sempre con il pulsante destro del mouse sopra la periferica, dal menu contestuale avviate il servizio SPP (servizio porta seriale). A questo punto inizierà a lampeggiare lentamente il solo LED verde. Visualizzate le proprietà della connessione SPP per sapere a quale COM virtuale è stata associata la comunicazione con il modulo. Avviate l'IDE di Arduino, aprite Serial monitor sulla COM, assegnata all'adattatore Bluetooth, con il Baud Rate di 38.400 ed attendete l'arrivo dei dati acquisiti da Arduino. Notate che è come se Arduino fosse collegato via cavo USB, ma in pratica è connesso in modalità wireless. Non essendo cablate le linee RTS e CTS dal modulo Bluetooth Bee verso Arduino, non sarà possibile la programmazione di Arduino tramite questo collegamento. ■