

Mensile di progettazione elettronica, attualità scientifica, novità tecnologiche.

€ 6,00

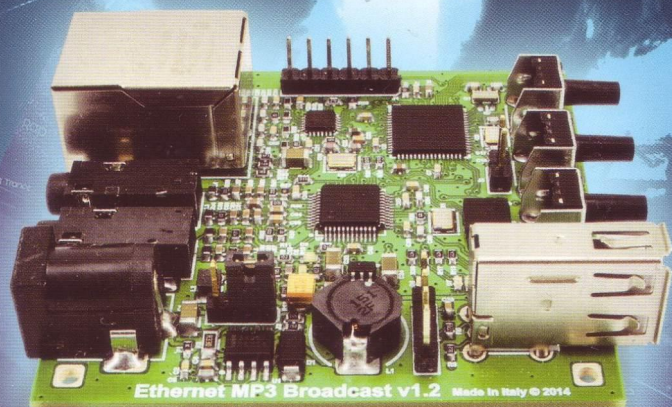
Elettronica In

www.elettronica.in.it

oltre l'elettronica

Anno XXI - n. 195
Maggio 2015

**Raffreddatore
per 3DRAG Choco**



**mp3
Broadcasting
via Ethernet**

- GestIC: prima applicazione
- Arduino Jpeg Camera Application
- Open Motor Control: il controller per tutti

- Radiocomando bidirezionale
- Tre soluzioni per l'alimentazione



Direttore responsabile:

Arsenio Spadoni
(Arsenio.Spadoni@elettronica.in.it)

Redazione:

Gabriele Daghetta, Paolo Gaspari, Boris Landoni,
Marco Magagnin, Davide Scullino,
Alessandro Sottocornola
(redazione@elettronica.in.it)

Grafica:

Alessia Sfulcini, Filippo Gori
(Alessia.Sfulcini@elettronica.in.it)

Ufficio Pubblicità:

Monica Premoli (0331-752668)
(Monica.Premoli@elettronica.in.it)

Ufficio Abbonamenti:

Elisa Guarniero (0331-752668)
(Elisa.Guarniero@elettronica.in.it)

DIREZIONE, REDAZIONE, PUBBLICITÀ:

FUTURA GROUP srl - Divisione Editoriale
via Adige 11 - 21013 Gallarate (VA)
Telefono 0331-752668 Fax 0331-792287

Abbonamenti:

Annuo 10 numeri Euro 45,00
Esteri 10 numeri Euro 45,00 (digitale)
Le richieste di abbonamento vanno inviate a:

FUTURA GROUP srl
via Adige 11, 21013 Gallarate (VA)
tel. 0331-752668.

Distribuzione per l'Italia:

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

Stampa:

ROT03 Spa - Via Turbigo, 11/b
20022 CASTANO PRIMO (MI)

Elettronica In:

Rivista mensile registrata presso il Tribunale di Milano con il n. 245 il 3-05-1995.

Prezzo di copertina Euro 6,00. Gli arretrati nei formati cartaceo e digitale (pdf) sono acquistabili sul sito della rivista al prezzo di Euro 6,00.

Poste Italiane Spa - Spedizione in abbonamento Postale - D.L. 353/2003 (conv. in L. 27/02/2004) art.1 comma 1 - DCB Milano.

FUTURA GROUP srl è iscritta al Registro Operatori della Comunicazione n. 23650 del 2/7/2013.

Impaginazione ed immagini sono realizzati in Desktop Publishing con programmi Adobe InDesign e Adobe Photoshop per Windows.

Tutti i contenuti della Rivista sono protetti da Copyright. Ne è vietata la riproduzione, anche parziale, la traduzione e più in generale la diffusione con qualsiasi mezzo senza l'autorizzazione scritta da parte dell'Editore. I circuiti, il firmware ed il software descritti sulla Rivista possono essere realizzati solo per uso personale, ne è proibito lo sfruttamento a carattere commerciale e industriale. Tutti possono collaborare con Elettronica In. L'invio di articoli, materiale redazionale, programmi, traduzioni, ecc. implica da parte del Collaboratore l'accettazione dei compensi e delle condizioni stabilite dall'Editore (www.elettronica.in.it/ase.pdf). Manoscritti, disegni e foto non richiesti non verranno in alcun caso restituiti. L'utilizzo dei progetti e dei programmi pubblicati non comporta alcuna responsabilità da parte della Società Editrice.

Futura Group
Edizioni

I dieci anni di Arduino e la prima vittoria nella Robocup Jr

Ci sono voluti dieci anni ma è successo.

È successo che a dieci anni dalla sua nascita, una scheda Arduino ha portato alla vittoria un robot nella finale della Robocup Jr Italia, battendo i robot avversari realizzati con piattaforme Lego.

È accaduto poche settimane fa nella categoria Rescue Line, una specialità nella quale i robot debbono svolgere in maniera completamente autonoma una missione particolarmente impegnativa, che comporta la disponibilità di notevoli risorse hardware e software oltre che, per quanto riguarda i team, di competenze in ambiti differenti, dalla meccanica alla matematica, dall'elettronica alla programmazione.

Sicuramente competenze che i ragazzi del Liceo Scientifico di Trento che hanno vinto la gara possiedono, ma che la semplificazione tecnologica introdotta dal progetto Arduino ha consentito di amplificare al punto di battere i Lego.

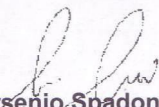
Due progetti - Lego e Arduino - che hanno avuto il merito di avvicinare al mondo della tecnologia schiere di giovani e meno giovani, offrendo gli strumenti per stimolare la creatività di ciascuno di noi.

Tuttavia, pur essendo i prodotti Lego basati su un concetto di grande valore quale quello della modularità, la vittoria di Arduino anche nella robotica evidenzia come questa piccola scheda racchiuda valori ancora più profondi che derivano dal fatto di essere completamente open source, sia nel software che

nell'hardware. Un nuovo modo di condividere la conoscenza privo di barriere, che ha messo in moto, grazie anche alla Rete, un rapporto collaborativo tra utenti e che ha dato vita a comunità che si ritrovano a lavorare insieme non solo attorno a progetti tecnologici, ma anche ad iniziative di tipo scientifico, sociale e didattico.

Il progetto Arduino ha anche avuto il merito di stimolare la nascita di altre iniziative di semplificazione tecnologica, prima fra tutte quella portata avanti dalla Fondazione Raspberry Pi che ha cercato di rendere accessibile ad un pubblico molto più vasto il sistema operativo Linux, con una scheda embedded che realizza un vero e proprio PC a bassissimo costo. La possibilità di avvicinare ai rudimenti della programmazione anche gli studenti più giovani, addirittura gli scolari delle elementari, fa di Arduino un potente strumento anche nel campo della didattica. La vittoria nella Robocup Jr di un robot basato su Arduino sembra confermare che anche la scuola italiana si stia accorgendo di questa opportunità.

Che sia questo, finalmente, il segno di un cambiamento in atto?


Arsenio Spadoni
(Arsenio.Spadoni@elettronica.in.it)



Sommario

Maggio 2015
n. 195

ARTICOLI

39 WIRELESS RADIOCOMANDO BIDIREZIONALE

Consente di ricevere sull'unità di controllo una conferma dell'esecuzione del comando impartito via radio ed è quindi adatto per situazioni nelle quali non c'è portata visiva.



51 HI-TECH GESTIC: PRIMA APPLICAZIONE

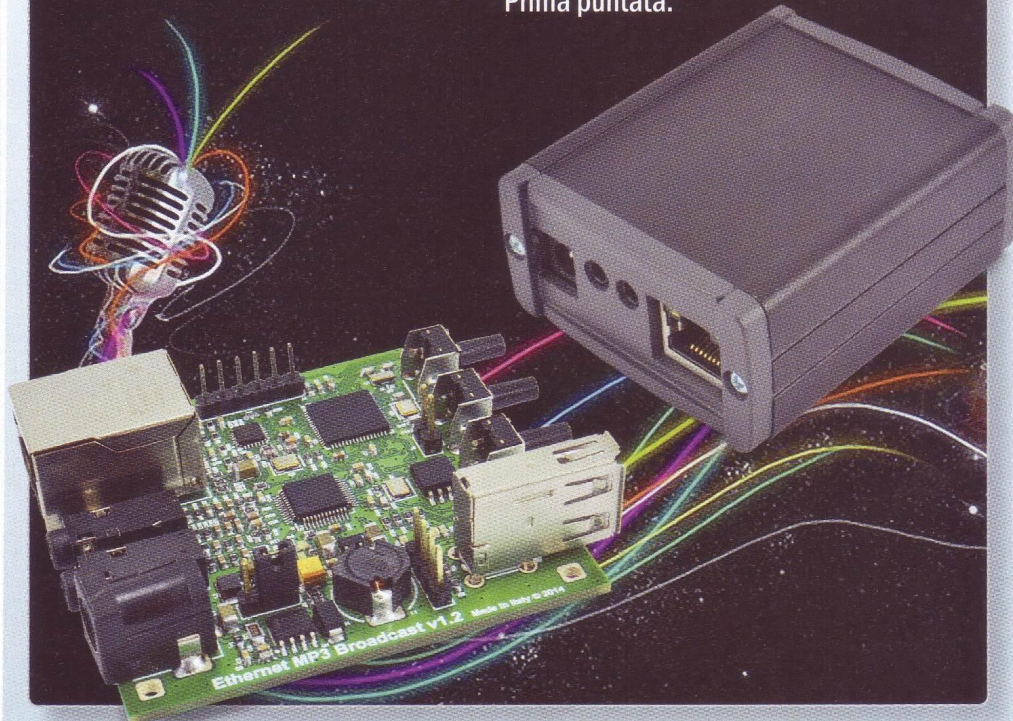
Sperimentiamo con la tecnologia di Microchip che permette di riconoscere i gesti contactless, mediante una demoboard da interfacciare al PC e vediamo come realizzare l'interfacciamento con un microcontrollore PIC. Seconda puntata.



ETHERNET MP3 BROADCASTER

Il progetto di un ricevitore e un trasmettitore audio su rete Ethernet. Prima puntata.

25

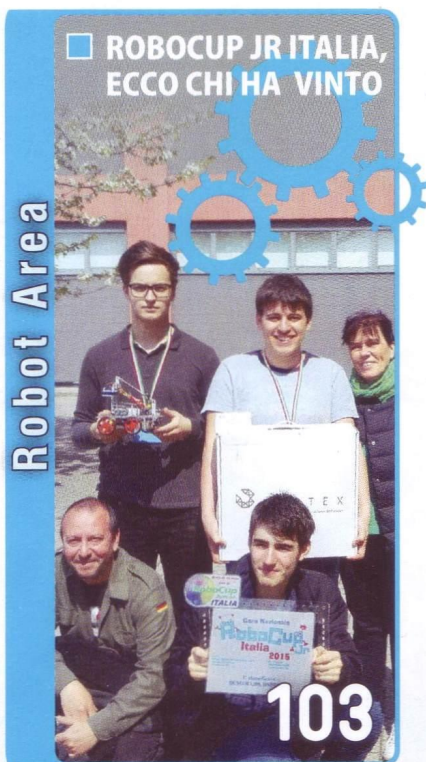


IN COPERTINA:

La rete Ethernet può anche essere utilizzata per diffondere la musica. Ecco come.

ELENCO INSERZIONISTI

- Futura Elettronica
- Microchip
- National Instruments
- RM Elettronica
- Robot & Makers
- www.mdsrl.it



La tiratura di questo numero è stata di 18.500 copie.



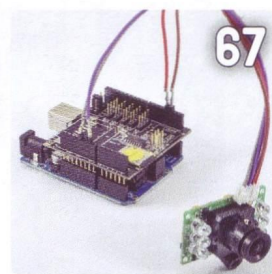
Mensile associato all'USPI, Unione Stampa Periodica Italiana

RUBRICHE

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

67 APPLICAZIONI ARDUINO JPEG CAMERA APPLICATION

Dotiamo Arduino di una JPEG camera seriale con illuminatore IR e realizziamo due applicazioni: nella prima salviamo le riprese a intervalli regolari su SD-Card e nella seconda usiamo un sensore PIR per rilevare le intrusioni e fotografiamo ciò che avviene.



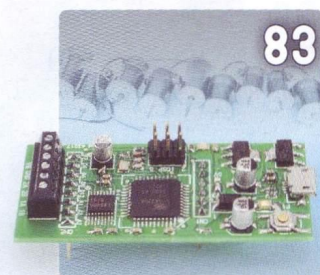
75 HI-TECH RAFFREDDATORE PER 3DRAG CHOCO

Il miglior risultato nella stampa di oggetti in cioccolato si ottiene raffreddando rapidamente gli strati depositi mediante un getto di aria fredda; ecco il dispositivo che permette di farlo.



83 ROBOTICA OPENMOTORCONTROL IL CONTROLLER PER TUTTI

Open source e basato su un microcontrollore ATmega32U4, è dotato di driver per due motori CC a spazzole o uno stepper motor. Può ricevere comandi da USB, seriale o I²C bus.



95 ANTEPRIMA TRE SOLUZIONI PER L'ALIMENTAZIONE

Proponiamo tre circuiti dedicati al tema "power", realizzati con altrettanti integrati della Linear Technology: un selettore automatico della fonte di alimentazione e due DC/DC converter.



113 DIDATTICA CORSO MPLAB X

Continuiamo il nostro viaggio alla scoperta di MPLab X, il nuovo ambiente di sviluppo integrato prodotto e distribuito da Microchip Technology, che soppianta il vecchio MPLab IDE. In questa puntata ci occupiamo di come realizzare applicazioni embedded multitasking con i PIC32 che sfruttano l'USB Device come protocollo di comunicazione.



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

Un display da PC per Raspberry Pi

Ho sentito parlare della possibilità di utilizzare la Raspberry Pi, che di fatto è un PC in miniatura, per mettere insieme un laptop o un tablet semplicemente collegandogli uno schermo LCD alla porta GPIO e una tastiera. Ho visto in giro alcuni ricambi di schermi LCD per portatili e quelli di ultima generazione hanno un'interfaccia a 40 o 30 contatti, che però non so come possa essere fisicamente collegata alla Raspberry Pi. È poi vero che si può fare il collegamento?

Santino Esposito - Bolzano

I display per notebook utilizzano da tempo un'interfaccia standard per la comunicazione dati con il chip grafico; si tratta della LVDS (Low Voltage Differential Signaling), così chiamata perché sfrutta linee differenziali (bilanciate). Nei monitor con retroilluminazione a LED, ed anche nei recenti a CCFL, il connettore, che tipicamente ha 40 contatti, trasporta, oltre ai dati e all'alimentazione per la logica di controllo della matrice LCD, l'alimentazione per la retroilluminazione. Nei monitor LCD più recenti viene utilizzata una connessione semplificata ma più prestante, chiamata eDP (acronimo di Embedded DisplayPort) e sviluppata dalla VESA (Video Electronics Standards Association) a partire dal più datato protocollo DisplayPort. Il relativo connettore è più piccolo: appena 30 contatti. Sul piano elettrico, l'interfaccia è simile per quanto riguarda i livelli di tensione e compatibile con la Video Port delle GPU; va però detto che non è intercambiabile, nel senso che cambia la disposizione dei contatti e qualcos'altro. Per quanto la eDP condivida il protocollo digitale di base, presenta alcune differenze, per capire le quali partiamo dalla considerazione che, sul piano elettrico, l'eDP è sempre un LVDS, perché i segnali dati vengono trasportati da canali differenziali; infatti è questa l'unica soluzione per far viaggiare dati a bassa tensione all'alta velocità richiesta dalla composizione di immagini animate alle risoluzioni anche Full HD 8k dei più raf-



Connettori dei pannelli LCD per notebook e tablet: in alto il tipico LVDS a 40 contatti e in basso il più recente eDP a 30 contatti.

finati standard video multimediali. Nell'LVDS i dati viaggiano su una linea a segnali differenziali a bassa tensione; per capire la necessità di linee bilanciate va detto che per trasportare segnali elettrici di solito bastano due fili e il segnale viaggia riferito al filo di massa (negativo) che ne è il riferimento. In presenza di segnali di disturbo come le interferenze elettriche nell'ambiente, bisogna lavorare ad alta tensione, ma nei circuiti che operano con dispositivi logici, bisogna restare sui 5 volt o addirittura sui 3 volt. Per "immunizzare" le linee dati nei confronti dei disturbi (che sovrapposti al segnale lo renderebbero indecifrabile causando vari problemi, che nel video si traducono in immagini alterate, contenenti dei vuoti o con colori sfalsati) si può ricorrere allora a soluzioni come le linee dati differenziali; in queste, che sono composte da tre fili, il segnale viaggia bilanciato, nel senso che su un filo c'è un segnale e sull'altro lo stesso segnale ma opposto di polarità. I due segnali di polarità opposta (specchiati) sono riferiti alla massa comune, che è il terzo filo, e nell'unità che deve ricevere i dati vengono fatti passare da un comparatore di tensione (se sono dati, altrimenti entrano in un amplificatore differenziale), del quale raggiungono uno l'ingresso invertente e l'altro il non-invertente; con questo tipo di collegamento, siccome il convertitore commuta la propria uscita a livello logico alto quando la tensione sul non-invertente supera quella dell'invertente e viceversa, i segnali si sommano. Invece gli eventuali disturbi, che vanno a sovrapporsi ai segnali dei due fili in egual misura, rappresentando un segnale uguale vengono sottratti

dal comparatore e non danno origine ad alcun segnale, perché si annullano. Ne risulta che rimangono i dati puliti. Questa tecnica consente di trasferire dati a velocità molto elevate anche usando tensioni molto basse come quelle della logica dei chip video (GPU) dei computer. Nella comunicazione tra GPU e pannello LCD, oltre alle linee dati c'è il clock, anche questo differenziale e con una propria linea. Nei monitor a interfaccia LVDS, per accelerare al massimo il trasferimento dei dati si spezzettano le informazioni dirette dalla GPU al pannello LCD in pacchetti che viaggiano su più linee (fino a 4) LVDS in parallelo, dette Lane. Nell'interfaccia degli LCD LVDS ci sono quindi tre fili per canale dati, ovvero fino a 12, nel caso di quattro linee differenziali. Oltre a questi, vanno aggiunti i fili di alimentazione e massa della logica di controllo della matrice e quelli dell'eventuale retroilluminatore a LED, più i segnali di controllo dell'accensione della retroilluminazione a LED come il PWM (BKLT_PWM_DIM), l'Enable (BKLT_ENABLE) ecc. Quanto alla eDP, è un'interfaccia concettualmente simile perché basata su link LVDS, solo che differisce per protocollo e grazie all'adozione di speciali accorgimenti, consente di raggiungere su un singolo canale velocità di comunicazione più elevate di quelle delle Lane LVDS. L'eDP trasporta tipicamente questi segnali nel canale principale (Lane 0): contenuto dei pixel componenti l'immagine, informazioni di temporizzazione del video, Pixel Clock, Hsync, Vsync (sincronismi), informazioni sul rapporto d'immagine (16:9, 24:9 ecc.), profondità di colore, video data error correction e audio digitalizzato (opzionale). Al canale

Terminal No.	Symbol	Function
1	HPD	HPD signal pin
2	NC	Non-Connection
3	Lane1-	Comp Signal Link Lane1
4	Lane1+	True Signal Link Lane1
5	Lane0-	Comp Signal Link Lane0
6	Lane0+	True Signal Link Lane0
7	AUX-	True Signal Auxiliary Ch.
8	AUX+	Comp Signal Auxiliary Ch.
9	NC	Non-Connection
10	NC	Non-Connection
11	NC	Non-Connection
12	NC	Non-Connection
13	VCD1	LED Cathode
14	VCD2	LED Cathode
15	VCD3	LED Cathode
16	VCD4	LED Cathode
17	VCD5	LED Cathode
18	VCD6	LED Cathode
19	NC	Non-Connection
20	NC	Non-Connection
21	NC	Non-Connection
22	V _{SS}	GND
23	V _{SS}	GND
24	V _{SS}	GND
25	V _{DD}	Power Supply, 2.5V
26	V _{DD}	Power Supply, 2.5V
27	V _{DD}	Power Supply, 2.5V
28	NC	Non-Connection
29	VAD	LED Anode
30	VAD	LED Anode

Tipica piedinatura del connettore a 30 poli di un pannello LCD a interfaccia eDP.

principali, nell'eDP si affianca un eventuale canale AUX (Lane 1) che contiene queste informazioni: dati EDID dal display (informazioni sul formato del display), Link Training protocol, Display Control (eDP versione 1.2), frequenza e controllo luminosità della retroilluminazione, controllo potenziato del backlight e del colore, dithering e FRC (Frame Rate Control), power management, Error checking del canale dati principale (protocollo CRC). In virtù del minor numero di canali dati presente, la porta eDP richiede 6 fili in meno (tre per ciascun canale in meno) e altri collegamenti per il clock. Inoltre i segnali dati sono più adatti a quelli generati dai recenti chip video (l'eDP può

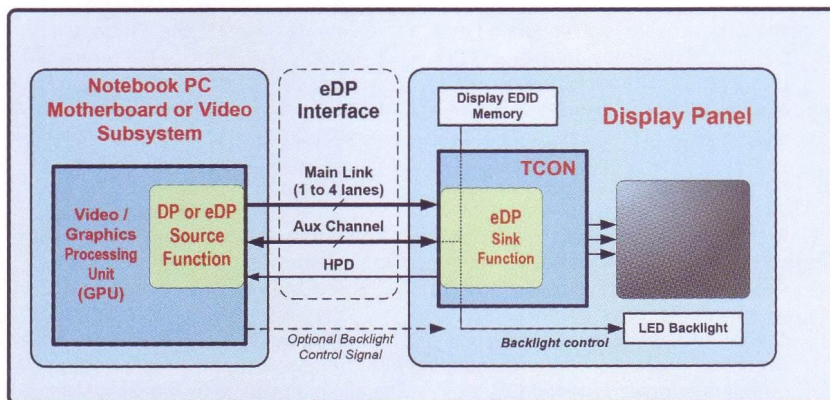
utilizzare l'interfaccia DisplayPort della GPU). Essendoci meno canali dati, si riduce la dimensione della scheda madre del notebook, ma anche la potenza consumata, a tutto vantaggio dell'autonomia del computer quando funziona a batteria; meno canali dati significano anche meno problemi di schermatura dai disturbi prodotti, per esempio dal clock ad alta frequenza. D'altra parte, i connettori hanno misure differenti perché le linee utilizzate per i segnali (comprendendo clock e dati) sono di meno: tipicamente 20 in una LVDS a 40 poli e solo 4 in una eDP semplice a due linee. Non si contano, in questo caso, i comandi della retroilluminazione. L'interfaccia eDP è indubbiamente miglio-

re della LVDS a 40 contatti ed è proprio per questo che i costruttori di pannelli LCD puntano su di essa e sulle sue nuove release. La VESA migliora periodicamente l'interfaccia aggiungendole nuove funzioni; alla versione iniziale eDP v1.0 è presto seguita la v1.1, sostanzialmente identica, e poi la v1.2, nella quale è stata aggiunta la Lane AUX. La versione attualmente utilizzata è la eDP v1.3, cui è stata aggiunta la modalità Panel Self-Refresh (PSR) e implementata in un computer per la prima volta nel 2012. La PSR consente di aggiornare nei cambi di scena soltanto i pixel che subiscono cambiamenti tra un frame e l'altro lasciando inalterati i rimanenti; ciò si traduce in una notevole riduzione del carico di lavoro svolto dalla GPU e, quindi, minori consumi energetici e possibilità di refresh veloci anche con GPU a prestazioni ridotte. Alla v1.3 è succeduta, un paio d'anni fa, la v1.4. Oggi, per venire incontro alle esigenze sempre crescenti di prestazioni determinate dagli schermi Full HD da 8k la Video Electronics Standards Association ha pubblicato (la notizia è di febbraio 2015) le specifiche del nuovo standard Embedded DisplayPort (eDP) versione 1.4a.

Il nuovo standard va a sostituire la versione 1.4 eDP e consente una maggiore velocità di trasferimento di dati video allo scopo di aumentare la risoluzione dei pannelli, avere una maggiore profondità di colore e frequenze di refresh più elevate.

La velocità del collegamento HBR3 è stata aumentata a 8,1 Gbps, il che significa che l'interfaccia, utilizzando quattro linee, può raggiungere un data-rate totale di 32,4 Gbps. Oltre ad un miglioramento della qualità di immagine, l'eDP 1.4a implementa la versione 1.1 del protocollo Display Stream Compression, che consente di ridurre il data-rate richiesto all'interfaccia video o il numero di collegamenti, oltre che contenere la dimensione del frame buffer in modo da abbassare i consumi del sistema e aumentarne, conseguentemente, l'autonomia. L'utilizzo congiunto di HBR3 e DSC v1.1 consentirà al nuovo standard eDP 1.4a di supportare pannelli integrati con risoluzioni fino a 7680×4320 pixel (8K) a 60Hz e 4K a 120Hz con profondità di colore a 10 bit. Altra novità portata dallo standard è la tecnologia "Multi-SST Operation", o MSO, che supporta un nuovo tipo di architettura di visualizzazione a segmenti che permette di incrementare l'efficienza energetica (anche a risoluzioni più basse) e di produrre display più sottili. L'Embedded DisplayPort 1.4a migliora ulteriormente la tecnologia Panel Self Refresh.

E veniamo all'utilizzo di un display da notebook con Raspberry Pi: si può fare a patto di utilizzare un traslatore in grado di convertire i livelli 0/3,3V della scheda negli LVDS ±5V simmetrici; in Internet trovi molti esempi di convertitori del genere.



La valvola rinasce nel transistor a vuoto

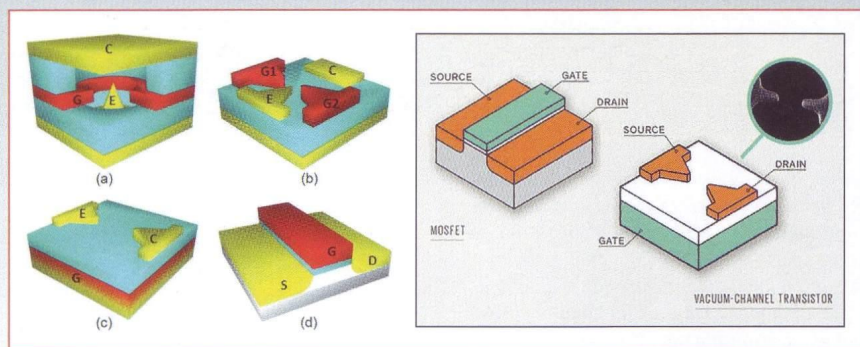
Gli attuali componenti a semiconduttore sembrano arrivati ai loro limiti in fatto di velocità di commutazione (frequenza di lavoro) e per questo i ricercatori sono costantemente a caccia di nuovi materiali e architetture innovative; in quest'ottica è inquadrato il grafene e con esso tutti i materiali detti "bidimensionali" perché costituiti da un solo strato di molecole. Recentemente ho però sentito che un'altra via percorsa è quella della creazione di componenti a vuoto che funzionano concettualmente come le valvole ma che di fatto sono a semiconduttore. Ne avete sentito parlare?

Lorenzo Tovoli - Pordenone

Probabilmente ti riferisci a quella notizia secondo cui ricercatori della NASA hanno progettato e realizzato in laboratorio una sorta di tubo a vuoto che concettualmente somiglia a una valvola termoionica. Più precisamente, si parla di "transistor al vuoto" perché i tecnici del centro di ricerca Ames della NASA hanno immaginato un FET nel quale il gate è fatto di elio. Idealmente dovrebbe essere vuoto, come appunto nelle valvole, ma si è scelto l'elio perché più semplice da gestire con risultati pressoché identici. In altre parole stiamo parlando di una valvola realizzata con i metodi produttivi dei moderni transistor FET. Per spiegare com'è fatto e in che modo funziona questo componente, bisogna dire che nelle tradizionali valvole termoioniche amplificatrici, la più semplice delle quali è il triodo (assimilabile, per comportamento, a un FET a giunzione) c'è un terminale -chiamato griglia- che regola il flusso della corrente che scorre tra gli altri due (anodo e catodo); l'anodo si mantiene a polarità positiva rispetto al catodo e in condizioni normali, oltre una certa tensione gli elettroni rilasciati dal catodo (opportunamente riscaldato da una

resistenza o esso stesso è la resistenza) per effetto termoelettronico, vengono "aspirati" dalla polarità positiva dell'anodo. Polarizzando la griglia negativamente rispetto al catodo, parte di questi elettroni viene respinta e si riduce così la corrente anodica, fino ad arrivare alla totale interruzione della corrente, ottenuta quando la tensione negativa raggiunge il valore di pinch-off. Il jFET a canale N funziona similmente, solo che le cariche elettriche (elettroni) scorrono tra source e drain e il flusso viene controllato dalla tensione tra gate e source. Nel transistor a vuoto ideato dalla NASA e fatto di semiconduttore, c'è un terminale emettitore di elettroni (source) polarizzato negativamente ed uno, che gli sta di fronte, mantenuto a polarità positiva rispetto ad esso (drain); per favorire l'effluvio elettrostatico, il terminale emettitore è a punta, cosicché basta un leggero campo elettrico per innescare il flusso di elettroni. Il gate è un elettrodo anulare che circonda il canale tra il terminale emettitore e quello di drain, la cui tensione, rispetto all'emettitore, permette di strozzare il canale stesso e di modulare la corrente. Ma perché questa architettura? Semplice: perché lavorare nel vuoto (o in un canale pieno di elio, come ipotizzato) permette di far muovere gli elettroni

più rapidamente; inoltre, a differenza di quanto avviene in transistor bipolari e FET, la commutazione da acceso a spento è più rapida. Infatti nei transistor BJT, ciò che limita la velocità di commutazione è la ricombinazione delle cariche nella base, che, nel funzionamento da interruttore (quello richiesto nei dispositivi logici) ritarda il passaggio dalla saturazione all'interdizione, anche malgrado l'adozione di strutture dotate di diodi Schottky per evitare l'eccessivo abbassamento della tensione collettore-emettitore in saturazione. Nei FET a giunzione il problema sono le capacità parassite, che nei MOSFET diventano anche maggiori e tali da lasciare in conduzione il transistor per qualche istante anche dopo la rimozione della tensione di polarizzazione del gate. Secondo i ricercatori della NASA, questi nuovi transistor permetterebbero di arrivare a una frequenza di transizione (la frequenza alla quale, utilizzando il transistor come amplificatore lineare, il guadagno hfe si riduce a 1) a 460 GHz: circa cento volte la massima frequenza dei processori più veloci utilizzati nei nostri computer. Per il momento il nuovo transistor a vuoto è un prototipo fermo in laboratorio, perché prima di poterlo realizzare su scala industriale occorre superare le problematiche dei relativi processi produttivi.



Confronto tra le strutture ipotizzate per i transistor a vuoto (a, b, c) confrontate con quella del MOSFET tradizionale (d).

RM ELETTRONICA

Forniture per hobbisti, scuole e industria

Distributore autorizzato:

**FUTURA
ELETTRONICA**

**EL SE
Kit**

ELETTRONICA

Il tuo punto
di riferimento
per l'elettronica
a ROMA

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

e perciò il data-rate della comunicazione; pertanto, quando nelle specifiche di una scheda video o di una scheda madre si vede scritto che supporta la modalità PCI express 4x, significa che il bus è formato da 4 lane dei dati, ovvero quattro coppie di linee di trasmissione e ricezione unidirezionali.

L'interfaccia in tempo si è rapidamente evoluta sino a raggiungere 500 MB/s per ogni Lane, caratteristici della tecnologia PCI express Rev. 2. Con il rilascio della recente PCI express Rev. 3 (terza generazione) la larghezza di banda è stata raddoppiata, raggiungendo 1 GB/s per Lane, quindi un PCI express Rev. 3 con 16 lane (x16) permette di raggiungere un totale di 16 GB/s di flusso dati per direzione.

Il Port Forward dei router

Vorrei rendere accessibile da remoto tramite Internet una telecamera IP che ho acquistato e installato a casa mia in modo da vedere anche quando sono via cosa accade. Ho una linea ADSL con un router configurato solo per uscire e non per entrare da remoto, a meno di non impostare il Port Forward. Ma cos'è?

Filippo Tommasetti - Pavia

Il router è un instradatore delle comunicazioni su reti dati che smista le comunicazioni presenti su una LAN verso un unico indirizzo IP (corrispondente a quello del provider Internet) e da esso verso i siti e gli indirizzi puntati dalle applicazioni che girano nei computer della rete locale. Il router effettua quindi l'instradamento di indirizzi IP privati verso un unico IP pubblico assegnato alla connessione Internet, operazione, questa, chiamata NAT (Network Address Translation). Il router è progettato per far viaggiare una comunicazione dati dalla LAN a Internet e non viceversa; questa prerogativa conferisce sicurezza alla rete e ai computer che vi si trovano, perché impedisce l'accesso dall'esterno. In deroga a ciò è possibile aprire una "porta" di comunicazione che consente a connessioni avviate da un elemento in Internet di presentarsi sulla LAN; questo si realizza con la funzione Port Forward, che permette di lasciar passare verso un indirizzo IP della LAN le richieste di connessione che arrivano da Internet sulla porta 80 (la porta predefinita del web). Per configurare il Port Forward del router occorre inserire nella sua configurazione una regola basata su una porta pubblica (porta sull'IP pubblico che resterà in ascolto), indirizzo privato (IP del computer della rete cui verranno



Estrusore per realizzare filamenti in plastica partendo da frammenti di materiale plastico, anche riciclato.

dirette le richieste in arrivo dalla porta 80) e protocollo (si può scegliere tra TCP, UDP o entrambi). La configurazione del Port Forward si esegue diversamente in base al tipo di router in dotazione.

Stampa 3D: filamenti riciclati anche dal cibo

Dalla massiccia diffusione delle stampanti 3D per il grande pubblico, quelle a filamento fuso, periodicamente qualche costruttore se ne viene fuori con nuovi materiali utilizzabili in forma di filamento e disponibili in bobina. Dopo il mix di pasta e legno che mi sembra si chiami laywood e che permette di stampare oggetti che al tatto sembrano realmente in legno, e il filamento in TPU con cui stampare particolari di gomma, ho sentito di prodotti a base organica derivati dagli alimentari. Ma che utilizzo e che vantaggi avrebbero?

Ermenegildo Biancori - Catania

Filamenti per stampanti 3D basati su composti organici in realtà ne esistono già: non è un mistero che il PLA, uno dei principali materiali plastici utilizzati nella stampa 3D con la tecnica FDM/FFM, derivi dagli amidi vegetali. Non è quindi fantascientifica, per quanto sia suggestiva, la notizia dell'ideazione di un filamento per stampa 3D ricavato da scarti alimentari e chiamato AgriDust. Non si tratta di una nuova soluzione per stampare prodotti edibili, come fa la nostra stampante 3Drag Choco, in quanto il filamento è stato sviluppato per stampe tradizionali. Il filamento AgriDust si ottiene a partire dall'essiccazione degli scarti vegetali, che vengono poi polverizzati e mischiati con del materiale legante, così da assumere consistenza e da poter essere amalgamati; estrudendo la miscela si ottiene il filamento. Per il momento questo prodotto è alla fase prototipale, perché l'industrializzazione richiede il perfezionamento del

filamento finalizzato sia ad ottenere la plasticità alle temperature di fusione caratteristiche di materiali come il PLA (sotto i 200 gradi centigradi) sia a garantire robustezza e resistenza al degrado nel tempo.

Ma l'idea del riciclo rimane comunque valida ed è già adottata con le plastiche: è il caso della Michigan Technological University, che in uno studio ha dimostrato come riciclare plastica da contenitori possa essere più economico rispetto all'acquisto di filamenti di "prima mano". E non è un caso che sul web proliferino dispositivi e progetti per farsi in casa filamenti ricavati fondendo le bottiglie di plastica in PET e gli scarti di stampa in PLA.

Anche il filamento a base di legno è un esempio che va nella direzione del riciclo: infatti si ottiene dagli scarti della lavorazione del legno o da mobili macinati che altrimenti costituirebbero rifiuti da smaltire. Un buon filamento tipo laywood è commercializzato da Futura Elettronica, www.futurashop.it con il nome PLA Wood e fonde a 205 ÷ 215 °C. È formato da polvere di legno amalgamata a un materiale plastico ed è disponibile nei diametri 1,75 e 3 mm.

Web Forum e supporto tecnico

Hai un problema con uno dei circuiti pubblicati? Vorresti effettuare una modifica?

Accedi al nostro Web Forum dove i nostri tecnici (ma anche gli altri lettori) ti aiuteranno a chiarire qualsiasi dubbio di natura tecnica.

Collegati a:

www.elettronica.in/webforum

■ 22-25 GIUGNO 2015 LASER WORLD OF PHOTONICS Monaco di Baviera • Germania

Prosegue la rivoluzione del laser nelle fabbriche: dalla produzione additiva ai compositi leggeri in carbonio, dalla lavorazione di acciaio ad alta resistenza al vetro temprato, il laser continua ad aprire la strada a nuovi materiali e nuovi processi. Dal 22 al 25 giugno 2015 il

settore si dà appuntamento a Monaco di Baviera per LASER World of PHOTONICS. Aziende espositrici ed esperti del mondo del laser mostreranno nei padiglioni espositivi, nei convegni applicativi e nel World of Photonics Congress come la rivoluzione del laser prosegue senza sosta.

Quest'anno uno dei temi al centro dell'attenzione sarà l'Additive Manufacturing, tecnologia sulla quale punta un numero crescente di aziende in tutto il mondo. Nell'industria ae-

rospaziale e automobilistica, nella costruzione di macchine e impianti, nell'automazione, nella costruzione di turbine, nella produzione di impianti medicali, gioielli e attrezzature sportive, questa tecnologia avanza a passo sostenuto. Nei padiglioni A2 e A3, il salone LASER World of PHOTONICS ne mostrerà potenzialità e applicazioni concrete.

www.world-of-photonics.com



■ 16-18 GIUGNO 2015 IFSEC International 2015 Londra (UK)



Riconosciuto come uno dei pochi appuntamenti internazionali in grado di rivolgersi all'intera catena del valore della sicurezza, IFSEC vedrà rafforzata la propria vocazione grazie al trasferimento all'ExCeL di Londra battezzato con successo lo scorso anno. Dal 17 al 19 giugno 2015, l'evento si prepara infatti ad accogliere oltre 50 mila visitatori e 1000 espositori. La capitale inglese ha già dimostrato del resto di avere grande attrattiva per utilizzatori finali e pubblico professionale, e se l'edizione 2014 è stata concordemente ritenuta la migliore mai realizzata, le attese per un 2015 di "ripresina" sono davvero fortissime. IFSEC International godrà come sempre del pieno supporto della *British Security Industry*

Association (BSIA), che riconosce nella fiera un imprescindibile punto di riferimento per il settore.

IFSEC International attira ogni anno l'intera catena di acquisti nel settore della sicurezza: gli utenti finali cercano di comprendere le ultime tecnologie e le best practice per proteggere i propri affari, mentre installatori e integratori sono desiderosi di scoprire i migliori prodotti per i loro clienti, consulenti e progettisti.

Per tutti, ad IFSEC sarà possibile instaurare nuove relazioni commerciali e trovare occasioni di networking dove i visitatori entreranno in contatto con le persone chiave per il loro business, in incontri one-to-one mirati e quant'altro per mettere in con-

tatto domanda e offerta. Accanto all'esposizione, la manifestazione propone (in apposite aree) convegni e momenti formativi organizzati con la collaborazione di importanti nomi dell'industria della sicurezza.

Molto folta la presenza italiana con tutti i big del settore ed una collettiva organizzata dall'ICE.

www.ifsec.co.uk/



■ 10-12 GIUGNO 2015

Intersolar Europe

Monaco di Baviera • Germania

È la più importante manifestazione al mondo dedicata all'industria dell'energia solare, una piattaforma globale per produttori, distributori, fornitori di servizi e consumatori che vedrà con-

vergere a Monaco di Baviera dal 10 al 12 giugno oltre 40.000 visitatori che avranno l'opportunità di conoscere e toccare con mano prodotti e servizi offerti da oltre 1.000 espositori.

I visitatori avranno anche la possibilità di seguire una serie di eventi collaterali per meglio comprendere le nuove tendenze di mercato mentre l'Innovation Exchange offrirà l'opportunità di conoscere i prodotti e i servizi più innovativi, i migliori dei quali verranno premiati con l'Intersolar AWARD. Numerosi anche i workshop e le presentazioni che offriranno l'opportunità di approfondire la conoscenza di tecnologie e prodotti. Infine, l'Intersolar Europe Conference offrirà ai partecipanti importanti spunti sull'evo-



■ 9-10 GIUGNO 2015

POWER-GEN EUROPE

Amsterdam

POWER-GEN Europe 2015

RAI, Amsterdam 09 - 11 Jun, 2015

Fin dalla sua prima edizione del 1993, POWER-GEN Europe si è affermato come l'evento leader per quanto riguarda la generazione di potenza in Europa, con l'edizione dello scorso anno che ha visto la presenza di più di 11.000 professionisti del settore, 435 espositori e più di 1.000 presenze alle conferenze. Tutti esperti del settore, provenienti da più di 100 nazioni, che per il 40 % dei casi ricoprono posizioni di rilievo nei rispettivi Management.

L'abbinamento con Renewable Energy World Europe consente di affrontare tutti gli aspetti del settore della produzione di energia, con presentazioni di natura tecnica e strategica che affiancano la più grande esposizione di impianti e servizi per la produzione di energia.

Sempre attento all'evoluzione dei mercati, il prossimo giugno

l'evento metterà in luce i molteplici aspetti dei cambiamenti in atto in Europa, alla scoperta delle opportunità che ne deriveranno per le imprese innovative e flessibili che vorranno coglierle.

La transizione del settore energetico in Europa è radicale e permanente: il tradizionale metodo di fornitura dell'energia elettrica che prevedeva una centrale e una linea di connessione unidirezionale verso i clienti sta affrontando la sfida dei sistemi di produzione distribuiti sul territorio che richiede nuovi approcci, nuovi prodotti e nuove competenze. Per i professionisti del settore energia tutto ciò rappresenta un'opportunità di innovazione e nuovi modi di funzionamento, tutti temi che saranno al centro dell'interesse sia di POWER-GEN Europe che di Renewable Energy World Europe.

I partecipanti alle due manifestazioni potranno conoscere importanti op-

portunità nel settore energetico con particolare riferimento a:

- Generazione flessibile
- Sistemi energetici intelligenti
- Generazione distribuita
- Tecnologia del carbone pulito
- Funzionamento dei moderni impianti e modernizzazione
- Stoccaggio di energia
- Sviluppo delle energie rinnovabili
- Il ruolo delle utilities
- Integrazione energetica urbana

POWER-GEN Europe e Renewable Energy World Europe offrono un panorama espositivo completo sui principali fornitori, subfornitori e aziende di servizi in tutta la catena di valore della generazione di energia che, unitamente ai contenuti delle conferenze, consentiranno ai professionisti del settore di farsi un quadro completo delle tendenze tecnologiche e di mercato del settore energia.

www.powergeneurope.com/

luzione dei mercati internazionali anche in relazione alle condizioni geopolitiche.

Nuovi modelli di finanziamento e di business per il fotovoltaico saranno uno dei temi più importanti di questa edizione di Intersolar Europe, anche in considerazione del fatto che gli impianti fotovoltaici stanno giocando un ruolo sempre più importante nella produzione di energia in tutto il mondo. Secondo una ricerca di mercato di IHS, nel 2015 il mercato del fotovoltaico vedrà una crescita della domanda del 25% con ulteriori 53-57 gigawatt (GW) installati.

Mentre in Europa la crescita si mantiene stabile, Cina, Giappone e Stati Uniti stanno spingendo all'insù i mercati mondiali. Importanti cali dei prezzi di produzione accompagnati da altrettante riduzioni dei sussidi governativi stanno creando

nuove sfide e nuove opportunità. Intersolar Europe si svolge in contemporanea con EES Europe, la più importante fiera continentale dedicata ai sistemi di immagazzinamento dell'energia, alle batterie ed alle tecnologie correlate, un settore in grande espansione e con una forte "carica" innovativa. Quest'anno è anche in programma la prima conferenza su tali temi, la EES Europe Conference.

L'area dedicata alla e-mobility (hall B1, booth B1.470) fornirà ai visitatori una visione complessiva sull'evoluzione del settore e sulle nuove tecnologie di accumulo e ricarica.

www.intersolar.de

inter solar
connecting solar business | EUROPE

■ 11 GIUGNO 2015

NI Aerospace & Defense Forum 2015

Roma

"A system level approach: from rapid prototyping to validation, test and deployment": sarà questo il filo conduttore della settima edizione di NI Aerospace & Defense Forum, l'evento organizzato da National Instruments che si terrà l'11 giugno a Roma presso l'Hotel Ergife. L'obiettivo è dimostrare come anche in questo particolare settore l'approccio platform-based di NI offra grandi benefici, riducendo i tempi necessari per passare dalla fase di prototipazione al prodotto finito.

Tale approccio richiede l'adozione di filosofie di integrazione, in cui i dati provenienti da tutte le fasi di sviluppo, dalla ricerca al test di produzione – attraverso validazione e verifica – devono essere memorizzati, verificati e riutilizzati direttamente

sul campo al fine di ottimizzare la progettazione di sistemi. Aerospace & Defense Forum è l'occasione per mettere in pratica nuove idee e realizzare progetti concreti: un'opportunità per generare innovazione attraverso il networking e la condivisione di successi.

Lo stato dell'arte nel settore aerospaziale e le nuove tecnologie saranno al centro della keynote del mattino e durante il dibattito moderato da Maria Clara Mussa di Cybernaua.it; nel pomeriggio sono previsti due track paralleli con esperti del settore che cercheranno di identificare i trend più significativi in questo campo. Al termine della manifestazione verrà premiato col "Best Application in Aerospace & Defence Forum" il progetto più innovativo realizzato con la piattaforma hardware e software di National Instruments.

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

MOSTRE MERCATO

ROSETO DEGLI ABRUZZI (TE)
MOSTRA MERCATO DEL RADIOAMATORE E DELLE TELECOMUNICAZIONI

Palasport di Roseto degli Abruzzi
Organizzazione: ARI ROSETO
Tel: 0858931033
Fax: 0858931033
info@ariroseto.it
www.ariroseto.it
06 e 07 Giugno 2015

NOVEGRO (MI)
RADIANT & SILICON

• Robot & Makers Milano Show
Parco Esposizioni di Novogro (MI)
Organizzazione: Comis Srl
Tel: 027562711 - Fax: 027020835
radiant@parcoesposizioninovegro.it
www.parcosposizioninovegro.it
06 e 07 Giugno 2015

GODEGA DI SANT'URBANO (TV)
FIERA DELL'ELETTRONICA

Centro fieristico di Godega
Organizzazione: Eventi 2000
Tel: 3498632614
silvia@eccofatto.info
<http://eccofatto.eu/>
13 e 14 Giugno 2015

AVEZZANO (AQ)
MILLENNIUM ELETTRONICA & AUTO E MOTO D'EPOCA

Centro Fiere Avezzano
Organizzazione: Millennium Eventi
Tel: 0696840758
info@millenniumeventi.it
www.millenniumeventi.it
20 e 21 Giugno 2015

FRIEDRICHSHAFEN (Germania)
HAM RADIO - Messe Friedrichshafen

Quartiere Fieristico Messe Friedrichshafen
Organizzazione: Messe Friedrichshafen GmbH
Tel: +49 7541708404
Fax: +49 75417082405
hamradio@messe-fn.de
www.hamradio-friedrichshafen.de
26, 27 e 28 Giugno 2015

L'elenco aggiornato di tutte le Mostre Mercato del 2015 è disponibile sul sito www.elettronica.in sul quale è anche possibile scrivere un commento sulle fiere visitate.

L'elenco completo dei più importanti eventi nazionali e internazionali di elettronica, sicurezza e fonti rinnovabili è disponibile sul nostro sito: www.elettronica.in/myMain/Fiere/p_fiere.asp



Atmel hands-on su Cortex M7

Atmel fa un altro passo in avanti nel mercato sempre più competitivo dell'Internet degli oggetti (IoT) presentando un processore Cortex-M che raddoppia le capacità di elaborazione del modello precedente. Il design del nuovo Cortex-M7 Pelican di Atmel nasce per sistemi di tipo embedded che vanno dai router di edge della rete e i sistemi industriali fino ai sistemi automobilistici. Il dispositivo asseconda la crescente domanda di intelligenza, connettività e automazione per tutto quanto va dai sistemi di storsage fino a dispositivi per la sanità. Particolarmente curata la parte di elaborazione del segnale e le opzioni di connettività, mentre migliora il controllo della voce e la precisione della localizzazione via GPS. Per non dimenticare



una durata più lunga della vita delle batterie.

Scopo di questo hands-on technical training è:

- approfondire le specifiche tecniche della famiglia SAM S7/E7 basata su Cortex M7 Atmel
- imparare l'utilizzo dei driver del software package Atmel preparato per i compilatori IAR, Keil, GNU

- apprendere attraverso un esercizio pratico, la configurazione del micro-controllore Cortex M7

Per quanto riguarda l'hardware, Atmel fornirà gratuitamente ai partecipanti il tool ATSAMV71-XULT.

Questi gli argomenti principali dell'hands-on:

- Presentazione Tecnica dei Cortex-M7
- Come si utilizzano i diversi Driver Software Package (IAR, KEIL, GNU)
- Ottimizzazione Software per i bassi consumi per la famiglia SAM S7/E7
- Configurazione del SAM S7/E7 (core/TMC/SRAM/FLASH/etc.) utilizzando la scheda SAMV71-XULT.

A Milano il 19 maggio 2015, Padova il 20 maggio e Bologna il 21 maggio.

www.arroweurope.com

Power Efficiency Technical Forum

Arrow presenta la seconda edizione del Power Efficiency Technical Forum, evento dedicato alle soluzioni innovative e tipologie dei sistemi elettronici finalizzate alla conversione energetica ad elevata efficienza. Sono 11 le aziende che presenzieranno con uno speech, e altrettanti i fornitori che saranno presenti nell'area espositiva. Queste le tematiche più importanti:

- Conversione e la gestione di potenza
- Isolamento
- Ricarica
- Recupero dell'energia
- Alimentatori industriali (compresi quelli specifici per lighting)

Nel corso dell'evento è previsto anche uno spazio espositivo dove sarà possibile testare direttamente le soluzioni, demo board e prodotti, avendo un contatto diretto con ciascun fornitore presente.

La giornata prevederà inoltre l'intervento in plenaria (Vibration Energy Harvesting - it's possibile!) da parte del prof. Saggini dell'Università di Padova.

Queste le aziende presenti e gli argomenti trattati:
Texas Instruments: "Efficiency, Integration and Ease of Design: the New Generation of TI DC/DC Solutions for Industrial and Automotive Applications" Relatore: Nicola Cellurale

On Semi: "GaN HEMTs for High Switching Operation SMPs"

Relatori: Vittorio Crisafulli & Mario Privitera

Linear Technology: "Battery Management Solution: How to Charge and Manage Your Battery System to Optimize Life, Cost and Efficiency" Relatore: Maurizio Pogliani



ARROW ELECTRONICS



Avago: "Optocouplers: The Power of Robust Insulation" Relatore: Ambrogio Rebosio

Kemet: "Power Capacitors for Industrial Markets: Focus on Heavy Duty components" Relatore: Gianpietro Malagoli

Vicor: "Modular Component Power Design Methodology: A Fast, Cost Effective Approach to Power System Design from AC to Point-of-Load" Relatori: Marco La Ciamera & Massimo Gavioli

Infineon: "ICE5101 and New CoolMOS™ CE and P6: More Efficiency in Power Supplies and Lighting, at Less System Cost" Relatore: Irene Babudri

ST Microelectronics: "Smart Galvanic Isolated Drivers & SiC MOSFET Update" Relatore: Riccardo Tosoni

Vishay: "Vishay Sfernice, Power Resistance" Relatore: Franco Chierici

Fairchild: "Fairchild Leadership on Discrete and Power Technology" Relatori: Davide Lissoni & Maurizio Zanardo

Littelfuse: "Littelfuse Protection: When Designing around Power Applications; Circuit Protection is Vital to Safeguard Equipment and Improve User Safety" Relatore: Tim Elliott

A Bologna il 9 giugno 2015 (08:30 - 17:00) presso il Royal Hotel Carlton.

www.arroweurope.com

IoE Talks

Il successo dell'Internet of Everything Italian Forum, confermato da due edizioni di forte impatto che hanno avuto come centro propulsore la città di Milano, è legato anche alla capacità di fare rete da parte di aziende, partner, enti e acceleratori di impresa sparsi nel nostro Paese che ci hanno sostenuto, sponsorizzato e patrocinato.

Tutto questo è una riconferma che la rete funziona e che insieme, collegandosi e connettendosi "accadono cose straordinarie".

Da queste considerazioni nasce il progetto dell'IoE talks lo spin off dell'IoE Italian Forum. Portare il messaggio dell'IoE con il suo potenziale economico e di reale impatto produttivo ai territori, legando le tematiche a quelle che sono le esigenze proprie delle realtà locali.

La prima tappa è quella di Roncade (Trevi- so) il 17 giugno 2015. La location selezionata non è casuale. L'evento si terrà presso la sede di H-Farm: il venture incubator fondato nel 2005 che sostiene la nascita di progetti che semplifichino l'utilizzo degli strumenti e dei servizi digitali a persone e ad aziende, supportando queste ultime nella trasformazione dei loro processi in ottica digitale.

Azienda con una presenza internazionale che però ha mantenuto un indissolubile legame con il territorio d'origine ove ha luogo la sede principale, cuore pulsante di tutto l'ecosistema.

In questa prima edizione dell'IoE talks tema dominante è l'Italia vista come patria della bellezza. Una bellezza che intendiamo legata a concetti di eccellenza, talento, made in Italy, arte, paesaggio, storia...

In poche parole *"la naturale predisposizione tutta italiana all'appassionante creazione del Bello"*.

Gli ospiti della giornata interagiranno con le varie declinazioni di questa "bellezza" e i customer testimonial, legati al territorio veneto, porteranno invece la loro personale storia di successo imprenditoriale legata alle tecnologie di Internet.

A fine evento è prevista la visita guidata alle differenti start up ospitate nella sede di H-Farm.

L'evento sarà un tributo a Marco Zamperini e a supporto del Funky Prize (#funkyprize). <http://ioetalks.it/>

IoE talks

CRESCERE DIGITALI • H-Farm, Roncade (TV) 17 giugno 2015

Altera MAX10 & IDT Components Seminar

Il seminario, la cui partecipazione è gratuita previa registrazione, sarà di carattere prettamente tecnico, con alcune demo pratiche, tenuto da personale specializzato di EBV Elektronik e IDT.

L'agenda prevede:

09:15 – 09:30 Welcome and Registration.

09:30 – 10:00 Max10 Family general introduction and overview.

10:00 – 10:30 Architecture, features and technical details.

10:30 – 11:00 Solving board design challenges with IDT products (part1).

11:00 – 11:15 Coffee Break.

11:15 – 13:00 Practical example with Nios2 system using standard and custom peripherals.

13:00 – 14:00 Lunch.

14:00 – 14:30 Solving board design challenges with IDT products (part2).

14:30 – 16:00 Practical example: Project creation with QSys, system generation and compilation with Quartus2, software download and project debug on the evaluation kit.

A tutti i partecipanti verrà applicato uno sconto del 50% sull'acquisto del kit di sviluppo Altera EK-10M08E144ES/P + Cavo di programmazione USB-BLASTER-CABLE TER. Inoltre i partecipanti riceveranno in omaggio uno USB stick contenente l'ultima versione del tool IDT Timing Commander. Il 20 maggio 2015 a Sesto Fiorentino (FI)

www.ebv.com



TECO '15

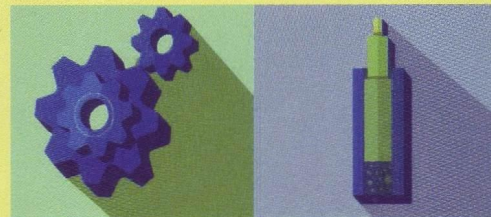
TECO è l'evento pensato per promuovere l'Industria Italiana della Meccatronica, Automazione e Trasmissione di Potenza nel suo complesso, un settore di assoluto rilievo nel panorama economico nazionale ed internazionale.

TECO nasce per riunire i rappresentanti del settore, per coinvolgere i principali protagonisti e offrire un'occasione di confronto sullo stato dell'arte con i più importanti leader di settore, i media e le istituzioni.

Durante l'evento saranno trattati temi particolarmente sensibili come il miglioramento dei processi, la flessibilità, l'efficienza, la sicurezza che oggi più che mai sono concetti base nella ricerca della qualità e delle migliori performance aziendali. Il settore a cui si rivolge TECO è ampio e sarà anche interessante analizzare le sinergie tra i differenti comparti e le importanti leve di competitività rappresentate dalle potenzialità di integrazione e di interdisciplinarietà proprie di questo settore.

Sessione "Tecnologica"

È rivolta a tutti i potenziali relatori e raccoglierà testimonianze su temi come,



ad esempio: materiali, prodotti e soluzioni, progettazione, normativa tecnica.

Sessione "Economica"

Dedicata in particolare ai fornitori di servizi ma aperta anche a best practice di aziende presenti in qualunque punto della filiera. Alcuni spunti: organizzazione aziendale, marketing, comunicazione, vendite, amministrazione, finanza e controllo, assistenza post vendita.

Sessione "Accademica"

Pensata per università e centri di ricerca ma anche per imprese che desiderano raccontare la loro esperienza su temi come: ricerca e sviluppo, collaborazione università-industria, education.

Teco raccoglie in un evento il Gotha del mondo della meccatronica, dell'automazione e delle trasmissioni di potenza e si pone i seguenti obiettivi:

- Passare da un evento focalizzato all'in-

Industrial Technology Efficiency Day

INDUSTRIAL TECHNOLOGY EFFICIENCY DAY

Il 18 Giugno 2015 torna in scena all'IBM Center di Segrate (MI) ITE day, convegno-mostra organizzato da Fiera Milano Media sull'efficienza energetica per l'industria. Giunta alla sua terza edizione, la manifestazione offre al visitatore un quadro completo dell'offerta attualmente disponibile per la realizzazione di soluzioni a elevata efficienza energetica in ambito di impiantistica e automazione industriale. Per fare dell'efficienza una vera arma di innovazione tecnica ed economica non si può fare a meno di passare per tutta quella serie di competenze e tecnologie in ambito automazione, controllo e supervisione, sia che si tratti di un grande impianto siderurgico, di una linea di confezionamento oppure di una singola macchina operatrice.

Industrial Technology Efficiency Day 2015

si rivolge ai protagonisti della filiera tecnologica che si occupano di progettare, realizzare, condurre, mantenere impianti produttivi in ambito manifatturiero e di processo: top manager, direttori tecnici, progettisti, tecnici e responsabili di produzione, direttori di stabilimento, energy manager, tecnici della manutenzione, buyer, ricercatori, tecnici e responsabili R&D, OEM, system integrator, utilizzatori finali.

Particolare evidenza sarà data a prodotti e soluzioni tecnologiche in ambito di automazione, controllo e supervisione grazie ai quali sono possibili vantaggi in termini di efficienza a livello di sistema produttivo: motori elettrici, inverter, azionamenti, avviatori, riduttori, motoriduttori, sistemi di trasmissione della potenza, sistemi di controllo, sistemi di supervisione, software di analisi e dimensionamento, software per la gestione dei carichi, diagnostica, sistemi di alimentazione, trasmissioni meccaniche, elementi di accoppiamento meccanici, sistemi di attuazione oleoidraulica e pneumatica, strumentazione di misura e controllo. Considerata la grande attualità del di-

battito sul tema oggetto della manifestazione, grande risalto avrà la parte convegnistica curata dalle riviste di Fiera Milano Media (Automazione Oggi, Automazione e Strumentazione, Progettare, Rivista di Meccanica Oggi, Energie e Ambiente) in collaborazione con le più rilevanti associazioni del settore e Business International, organizzatore di convegni e congressi del gruppo Fiera Milano Media.

Quest'anno, durante ITE 2015, l'automazione è ancora protagonista nel fare efficienza cui si aggiungono i vantaggi della cogenerazione ad alto rendimento (CAR) per la quale è previsto l'accesso al sistema dei titoli di efficienza energetica (TEE) o certificati bianchi. A fronte anche delle ultime novità normative esamineremo i diversi aspetti legati a soluzioni per l'efficientamento di processi produttivi, illustrando casi pratici.

Quest'anno poi nasce "face2face" corner dove sarà possibile parlare con esperti di tematiche 'calde'.

<http://ite.mostreconvegno.it/>



contro tra associati, i loro fornitori e interlocutori, a un'occasione di incontro/confronto con il mercato.

- Dare visibilità all'offerta tecnologica delle aziende del settore e, in particolare, a quelle associate.
- Consolidare il rapporto tra industria e mondo accademico. Individuare e dare risalto alle competenze distinte dell'industria italiana del settore facendone un'eccellenza a livello internazionale.

A Milano il 16 e 17 giugno 2015.
www.tecomeeting.it/

TECO'15

tecomeeting.it

Power Booster - SILICA Power Workshop

Con la recente acquisizione di International Rectifier, Infineon, già ai vertici del mercato globale dei semiconduttori di potenza, è in grado di offrire un portafoglio prodotti senza pari e con estrema attenzione alle esigenze dei clienti promuove soluzioni tecniche competitive per il mercato del controllo motore e del power in genere.

Obiettivo del workshop è quello di fornire ai partecipanti le basi concettuali e i metodi pratici per affrontare in maniera efficiente il processo di progettazione.

La sessione comune del mattino illustrerà le tecniche e gli algoritmi di controllo: sfruttando un vero inverter sarà possibile entrare nel dettaglio di ogni singolo blocco progettuale e, con l'ausilio di strumenti di laboratorio, analizzare e visualizzare le forme d'onda generate.

Nel pomeriggio i partecipanti potranno dividersi in 2 sessioni parallele in base alle proprie preferenze.

Power Session – Sessione Pratica

Partendo dalla piattaforma presentata nella live session della mattina verranno approfonditi gli aspetti tecnico/pratico e progettuali inerenti la parte di potenza.

SILICA
An Avnet Company

Saranno inoltre presentate le novità tecnologiche riguardanti il pilotaggio e lo stadio di potenza.

Digital session – Sessione Pratica

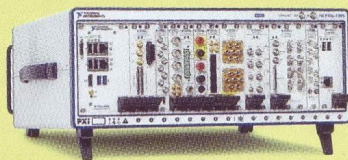
I progettisti digitali, utilizzando come piattaforma un Cortex-M della famiglia XMC, potranno conoscere e sperimentare, attraverso un hands-on training, le caratteristiche e le potenzialità dell'ambiente di sviluppo DAVE e delle specifiche librerie gratuite (Apps) che accelerano drammaticamente il time-to-market per applicazioni tipo controllo motori, alimentatori, inverter solare, elettrodomestici, motori ibridi, LED lighting ed altri controlli di processo.

Il 26 maggio 2015 a Padova e il 27 maggio a Bologna. Maggiori informazioni e iscrizioni sul sito di Silica.

www.silica.com

PXIe-8880, il primo controller embedded PXI di NI basato su Intel Xeon

National Instruments ha annunciato il controller NI PXIe-8880 basato sul processore Intel Xeon, e lo chassis NI PXIe-1085, il primo chassis che utilizza la tecnologia PCI Express Gen 3. La combinazione di otto core, il processore server-class Intel Xeon E5-2618L v3 e il trasferimento dati a 24 GB/s permettono prestazioni eccezionali per applicazioni complesse e multithread, come il test wireless, test per semiconduttore e prototipazione 5G. I clienti possono trarre vantaggio dalla flessibilità della piattaforma PXI sostituendo i controller di vecchia genera-

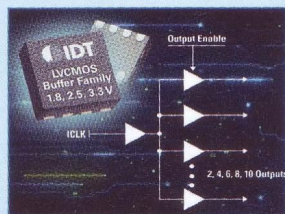


zione con NI PXIe-8880 e raddoppiare le prestazioni nelle applicazioni di test e misura. NI ha collaborato con Intel per fornire la potenza server-class dei processori Intel Xeon per il mercato del test e della misura. Il controller embedded presenta 8 core, una memoria DDR4 superiore ai 24 GB e un backplane con 24 canali di comunicazione PCI Express Gen 3. Questo permette a in-

gegneri e tecnici di raddoppiare la potenza di elaborazione e il trasferimento dati, rispetto ai controller di vecchia generazione. Inoltre, il nuovo chassis a 18 slot incorpora la tecnologia PCI Express Gen 3 con otto canali per slot per un totale di 24 GB/s di throughput, così da garantire ottime prestazioni per i futuri sistemi di test e misura. Unito al software LabVIEW per la progettazione grafica di sistemi, il controller PXI di nuova generazione e lo chassis permettono agli ingegneri e ai tecnici di elaborare dati relativi al test e alla misura con una velocità di trasferimento di 24 GB/s.

<http://Italy.ni.com>

Da IDT una nuova famiglia di buffer a bassissimo jitter



Integrated Device Technology (IDT) ha presentato una nuova famiglia di buffer per segnali di clock che garantisce le migliori prestazioni di jitter sul mercato. Disponibile in un contenitore molto compatto, la famiglia di buffer per clock fanout LVCMOS 5PB11xx garantisce un valore di jitter di fase additivo inferiore a soli 50 fs RMS (da 12 kHz a 20 MHz), una caratteristica che offre ai progettisti

maggiori margini rispetto ai dispositivi della concorrenza per rispettare i requisiti progettuali legati alle reti di distribuzione dei segnali di clock. Le piccole dimensioni del die hanno permesso di inserire il chip all'interno di un contenitore DFN a 8 piedini che occupa uno spazio di appena 2 millimetri per 2. Questi buffer sono la soluzione ideale per realizzare prodotti elettronici di largo consumo ad alte prestazioni, apparecchiature industriali, sistemi per applicazioni informatiche e di comunicazione nei quali vi siano da rispettare vincoli molto stringenti sulle temporizzazioni e sullo spazio occupato sulla scheda.

www.idt.com

Cree presenta i nuovi moduli a LED con le più alte prestazioni del settore

Cree rafforza la famiglia dei moduli a LED Cree LMH2 con l'introduzione dell'innovativo modulo LMH2+, il prodotto più performante del settore dei moduli a LED che offre prestazioni del 30% superiori rispetto alla precedente generazione. LMH2+ garantisce 125 lumen-per-watt con CRI superiore a 90 ed è l'unico modulo a LED che consente ai produttori di illuminazione di ottenere un'efficienza di sistema di 100 lumen-per-watt ed elevata qualità della luce. Caratterizzato da elementi innovativi come la nuova SC5 Technology Platform di Cree, il nuovo modulo a

LED LMH2+ garantisce molteplici nuove applicazioni, consente di realizzare apparecchi più piccoli e riduce i costi del sistema. Inoltre, LMH2+



offre ai produttori di illuminazione la possibilità di affrontare installazioni più impegnative come quelle legate ai vincoli di spazio o elevate temperature ambientali.

LMH2+ è una soluzione innovativa basata sul design di LMH2 e offre significativi miglioramenti in termini di prestazioni rispetto alla generazione precedente.

www.cree.com

Primo progetto di riferimento di un master IO-Link quadruplo

Maxim Integrated mette a disposizione di tutti i progettisti una singola piattaforma per sviluppare il master IO-Link e una varietà di sensori industriali. Il MAXREFDES79# di Maxim Integrated permette ai produttori di abbreviare il time to market dei progetti di automazione di fabbrica basati su IO-Link.

Il MAXREFDES79#, in assoluto il primo progetto di riferimento di un master IO-Link, permette di ridurre il costo ed

il tempo necessario per sviluppare e valutare sia il master stesso, sia i sistemi di sensori IO-Link. A differenza delle precedenti unità di sviluppo per master IO-Link, che erano dotate di un'unica porta, il master IO-Link MAXREFDES79# di Maxim consente di provare fino a quattro sensori contemporaneamente. Essendo un sistema conforme allo standard IO-Link, MAXREFDES79# funziona con qualunque sensore IO-Link oggi disponibile sul

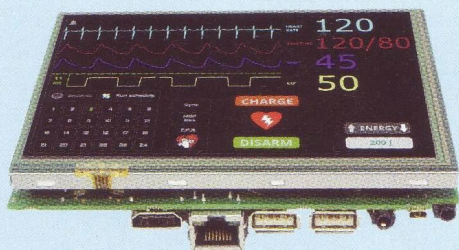
mercato. MAXREFDES79# si affianca ad altri quattro progetti di riferimento di Maxim riguardanti i sensori (il sensore di luce MAXREFDES23#, il sensore di prossimità MAXREFDES27#, l'ingresso digitale a 16 canali MAXREFDES36# ed il sensore di temperatura RTD MAXREFDES42#) per consentire ai clienti di velocizzare e semplificare il processo di simulazione del funzionamento. MAXREFDES79# è una piattaforma di sviluppo ad alta



integrazione che comprende anche lo stack software; il sistema è dotato di quattro porte, una per ciascuno dei master IO-Link (MAX14824), e integra i dispositivi di alimentazione.

www.maximintegrated.com

Da AndersDX Interfacce Utente Embedded a meno di 100 sterline



AndersDX ha presentato un nuovo modello entry level della sua popolare piattaforma DX Embedded Display. Si tratta del nuovo modulo DX1, basato sul microprocessore in tecnologia ARM Cortex A8 a 600 MHz di Texas Instruments. Il DX1 abbina un TFT

touch screen pre-integrato e un sistema operativo Linux, Android o Windows CE pre-caricato. L'aggiunta di un software applicativo lo trasforma in un sistema di prototipazione completo. Il modello DX1 è una piattaforma single touch con display integrato che

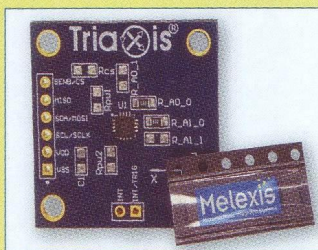
combina un TFT touch screen e un computer single-board in tecnologia ARM, allo scopo di disporre di una piattaforma di prototipazione pronta all'uso, che all'occorrenza può trasformarsi in un vero e proprio sistema produttivo.

www.andersdx.com

Scheda di valutazione EVB MLX90393 dedicata ai sensori definiti da software

Melexis ha presentato una scheda di valutazione (EVB) che facilita lo sviluppo di soluzioni innovative basate sul suo circuito integrato programmabile per la gestione di sensori magnetici MLX90393. La scheda di valutazione EVB MLX90393 ha dimensioni di 30 mm x 30 mm e permette ai progettisti di sperimentare la realizzazione pratica di una vasta gamma di interfacce uomo-macchina (HMI) anche molto sofisticate. Si possono valutare le caratteristiche di molteplici soluzioni, che spaziano dai relativamente più semplici cursori, tasti e slitte lineari fino ai più complessi sensori 3D, come joystick o manopole con tasti incorporati.

Grazie alla tecnologia proprietaria Triaxis sviluppata da Melexis, il circuito integrato per sensori MLX90393 a 16 bit di risoluzione presente sulla scheda genera un'uscita digitale proporzionale alla densità di flusso magnetico rilevata dal sensore su tre assi



perpendicolari (X, Y e Z). Inoltre, è possibile rilevare contestualmente anche le misure di temperatura. Il circuito integrato è un potente blocco funzionale dedicato al rilevamento della posizione e con il quale è possibile realizzare una vasta gamma di applicazioni. Permette ai progettisti di configurare a piacere numerosi parametri funzionali, in modo da ottimizzare il loro sistema di rilevamento in base alla velocità di acquisizione, integrità del segnale e consumo energetico.

www.melexis.com

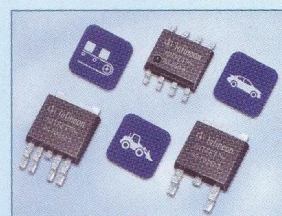
Dai relè elettromeccanici a quelli allo stato solido: arriva la famiglia HITFET+

La strada che porta alla completa sostituzione dei relè elettromeccanici con dispositivi a semiconduttore in grado di offrire la medesima affidabilità ha fatto un ulteriore passo in avanti grazie alla nuova famiglia HITFET+ di Infineon Technologies. I componenti della famiglia HITFET+ (High Integrated Temperature protected MOSFET) offrono prestazioni uniche quali la diagnosi funzionale, feedback digitale di stato, massima robustezza nei confronti dei corto circuiti, e – disponibile per la prima volta – regolazione dello slew rate per compensare e migliorare le perdite di commutazione e ridurre al minimo i disturbi di natura EMC. La famiglia HITFET+ comprende 16 modelli con R_{DS(on)} compreso tra 10 e 800 milliohm, differenti set di funzionalità (ad esempio con o senza feedback di stato) e differenti dimensioni

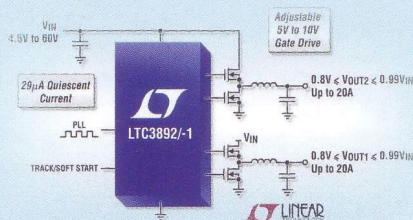
del package (D-PAK con 5 o 3 pin, DSO con 8 pin). I dispositivi HITFET+ con le medesime dimensioni di package sono completamente scalabili, senza che i progettisti debbano modificare il software o il layout del PCB. Il primo modello della famiglia, il BTF3050TE, è già in produzione in volumi.

Nelle applicazioni automobilistiche, questi prodotti possono pilotare le bobine delle valvole con impulsi PWM (Pulse Width Modulation) fino ad una frequenza di 20kHz. Inoltre possono essere impiegati nei circuiti di controllo dell'intensità delle luci.

www.infineon.com



LTC3892, controller step-down da 5V a 10V e 29µA di consumo in standby



Linear Technology presenta l'LTC3892, un controller DC/DC step-down sincrono a due uscite che consuma solo 29 µA quando è attiva un'uscita oppure 34 µA quando sono attive entrambe le uscite.

L'alimentazione in ingresso da 4,5V a 60V protegge contro i tran-

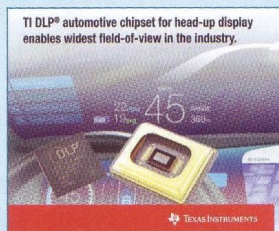
sienti ad alta tensione, supporta il funzionamento continuo durante l'avviamento a freddo nelle applicazioni del settore automotive ed è compatibile con vari tipi di fonti in ingresso e diverse strutture chimiche delle batterie. Ciascuna uscita può essere impostata tra 0,8 V e il 99

% di VIN su correnti di uscita oltre i 20 A con efficienza fino al 96%; questo rende il dispositivo ideale per applicazioni automotive, macchinari pesanti, sistemi di controllo industriale, sistemi robotizzati e applicazioni di telecomunicazione a 12 o 24 V.

www.linear.com

Chipset DLP per proiezione su parabrezza da TI

Texas Instruments ha presentato il primo chipset DLP progettato e certificato per visori a sovrapposizione (head-up display, HUD) in ambito automobilistico. Unendo le qualità di visualizzazione di immagini della tecnologia DLP e l'affidabilità delle applicazioni automotive, il nuovo chipset consente di realizzare HUD con il campo di visione più ampio attualmente disponibile, fino a 12 gradi. Offre inoltre livelli elevati di luminosità, colore e contrasto, su una piattaforma scalabile per le case costruttrici. Il chipset DLP3000-Q1, composto da un dispositivo a microspecchi digitali (DMD) con scheda grafica WVGA (wide video graphics array) da 0,3 pollici e da un controller DLPC120, consente di realizzare HUD di nuova generazione con il campo di visione più ampio in commercio, livelli eccezionali di luminosità, gamma dinamica e precisione cromatica, grande scalabilità. Inoltre, adottando il DLP3000-



Q1 come componente centrale dell'unità di generazione delle immagini (PGU) dell'HUD, l'intero sistema può essere integrato in una varietà di elementi, ad esempio un parabrezza o uno schermo trasparente. Questa applicazione, estendibile a tutti i modelli di veicoli, offre interessanti potenzialità di ritorno sull'investimento per le case costruttrici. Per accelerare lo sviluppo del prodotto finale, TI mette a disposizione un ampio ecosistema di studi di progettazione tra cui DLP Design Network, una rete che fornisce supporto per integrazione di hardware e software e progettazione di sistemi ottici.

www.ti.com

MCP2561/2FD, nuova famiglia di transceiver CAN a basso costo

Microchip annuncia la nuova famiglia di transceiver CAN FD (Controller Area Network Flexible Data-Rate), MCP2561/2FD. Posizionati tra un controller di protocollo CAN ed il bus fisico bifilare CAN, questi transceiver possono utilizzare sia il protocollo CAN che quello CAN FD. Questa famiglia di prodotti non solo aiuta i produttori automotive e industriali a soddisfare le attuali esigenze di comunicazione CAN, ma fornisce anche un valido aiuto per la migrazione verso il più recente network CAN FD di cui è sempre più sentita l'esigenza.

La crescita di reti di comunicazione a bordo veicolo continua ad essere guidata dalla necessità di controllare e monitorare la sempre più

complessa elettronica presente sulle vetture, con la conseguenza che, più Electronic Control Unit (ECU) vengono impiegate e connesse al bus CAN preesistente, più si limita la larghezza di banda a disposizione degli OEM automotive. Oltre a ciò, il tempo di programmazione end-of-line per ECU è in aumento a causa della crescente complessità delle applicazioni e della calibrazione, con i costi per linea di produzione che lievitano.

www.microchip.com



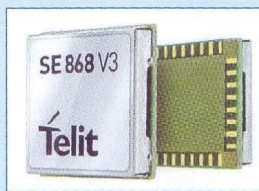
SE868-V3, il modulo GNSS per soluzioni di elevate performance

Telit Wireless Solutions ha annunciato il rilascio di un nuovo modulo GNSS, il SE868-V3. Questo modulo di posizionamento combina GPS, Glonass, Beidou, Galileo, e SBAS e abilita la creazione di soluzioni di navigazione e di posizionamento ad elevate performance.

Il modulo SE868-V3 può navigare fino a -162 dBm e traccia fino a -166 dBm, in modo tale da offrire performance migliorate in

ambienti difficili. È compatibile pin-to-pin con il precedente modulo SE868-V2 e con JF2. Questo modulo avanzato GNSS può tracciare simultaneamente costellazioni GPS e Glonass o GPS e Beidou, ed è Galileo-ready. Il pacchetto 11 x 11mm QFN contiene un potente processore baseband, una memoria Flash SQI e un chip GNSS con integrato un Low Noise Amplifier (LNA). La tecnologia ultra-sensibile RF front-end abilita una navigazione multi-GNSS indoor di elevata qualità anche in aree urbane ad alta densità. Inoltre il modulo SE868-V3 supporta A-GPS nonché SBAS.

www.telit.com



Nuovi processori applicativo ApP Lite TZ5000 per IoT basati su ARM Cortex-A9

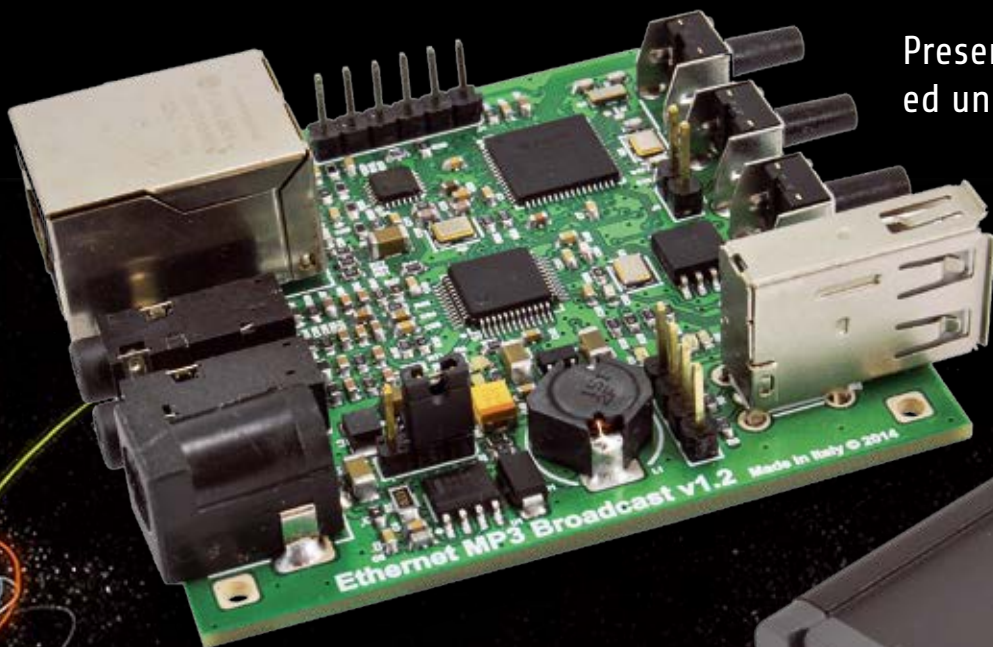
Toshiba Electronics Europe ha esteso la sua gamma di processori applicativi ApP Lite con i nuovi chip TZ5010XBG, TZ5011XBG, TZ5021XBG e TZ5023XBG. Progettati per facilitare lo sviluppo di applicazioni in ambito IoT (Internet of Things), questi nuovi processori sono basati sulla tecnologia ARM Cortex-A9 e permettono di ridurre il consumo di energia. Il sistema hardware proprietario per la gestione della potenza e la tecnologia di integrazione sviluppata da Toshiba permettono di ottenere un'elevata efficienza di memorizzazione, una capacità di elaborazione a bassa latenza e caratteristiche di sicurezza affidabili. Le caratteristiche innovative di questi processori permettono inoltre di passare rapidamente nella modalità di funzionamento a basso consumo in pochi microsecondi e facilitare la transizione a questa modalità rispetto agli approcci basati su software, che possono essere significativamente più



lenti. Si tratta di caratteristiche che aiutano molto i progettisti a ridurre il consumo di energia e contribuiscono inoltre a limitare la produzione di calore nella maggior parte delle applicazioni. Tutti e quattro i nuovi dispositivi della serie TZ5000 comprendono un'interfaccia MIPI DSI (Display Serial Interface) che supporta i formati video ad alta definizione full 1080p/60, mentre i modelli TZ5010 e il TZ5011 integrano un'interfaccia MIPI CSI2 (Camera Serial Interface) che supporta il formato video ad alta definizione full 1080p/30.

www.toshiba.semicon-storage.com

Presentiamo un ricevitore
ed un trasmettitore audio
su rete Ethernet.
Prima puntata.



ETHERNET MP3 OGG VORBIS BROADCASTER

dell'Ing. LUCA PASCARELLA

Quando siamo in un centro commerciale, in un supermercato o un grande negozio, ma anche in altri luoghi aperti al pubblico, usualmente sentiamo della musica proposta da impianti di diffusione sonora che permettono anche, all'occorrenza, di diffondere annunci da parte del personale. Un tempo questi sistemi si avvalevano di un impianto basato su un riproduttore di nastri magnetici o su un ricevitore radio collegato ad amplificatori e casse acustiche, ma oggi le più recenti tecnologie di riproduzione digitale impiegano come fonte BF dei lettori MP3 o dispositivi di broadcasting via Internet. Indipendentemente dal mezzo scelto, un impianto di diffusione sonora per grandi ambienti richiede amplificatori e diffusori acustici da colle-

gare mediante cavi, con ciò che ne deriva in termini di costi e superamento di problematiche relative alla stesura dei cavi. Da diffusori, amplificatori e riproduttori musicali non si può prescindere, ma è possibile risparmiare sui cablaggi affidandosi a dispositivi come quello proposto in questo articolo, mirato a realizzare un impianto di filodiffusione in un ambiente domestico o industriale, in un parco naturalistico, in un luogo di culto, in un centro commerciale e in generale dove serva diffondere musica o messaggi (anche pubblicitari) e sia disponibile una LAN. Il modulo che proponiamo è un Ethernet Broadcaster e consente di campionare un segnale audio analogico e trasmettere il flusso audio corrispondente in qualsiasi stanza di una casa

o di un ufficio e in generale di qualsiasi luogo, sfruttando come infrastruttura di comunicazione la rete ethernet, che può essere facilmente estesa con dispositivi wireless.

Data l'enorme vastità di esigenze applicative che il dispositivo può soddisfare, abbiamo sviluppato un software che ne permette la configurazione tramite un'interfaccia web. Infatti, il modulo può essere usato non solo per realizzare comunicazioni *point-to-point* tra due dispositivi omologhi, ma è possibile realizzare comunicazioni *broadcast* da uno ad N dispositivi. Inoltre l'adozione di protocolli standard nel campo dello streaming audio come *SHOUTcast* e *IceCast* rende il dispositivo compatibile con le web radio e i lettori multimediali per PC come *VLC media player* della *VideoLAN*.

Combinando più moduli con un computer dotato di lettore multimediale è possibile coprire una grande varietà di applicazioni, consentendo di trasmettere all'interno di un edificio o all'aria aperta la propria radio tematizzata o notificare in tempo reale messaggi di servizio.

I formati audio supportati dal modulo sono l'ormai diffuso e comune *MP3* ed il suo diretto concorrente (royalty free) *Ogg Vorbis*, il che lo rende flessibile e compatibile con le esigenze di utenti privati o aziendali. Il tutto è stato pensato per poter essere alloggiato in una comoda scatola metallica della *TEKO* (modello *Tekal 11.29* oppure *Tekal 11.30*). Il circuito è dotato dei connettori che in **Fig. 1** vedete descritti insieme ai pulsanti di comando e ai LED di segnalazione.

Innanzitutto troviamo il connettore d'alimentazione plug coassiale da 2,1 mm con positivo centrale e negativo esterno. Tramite questo connettore si applica l'alimenta-

zione, che deve essere a tensione continua di valore compreso tra 5 e 15 V, con una corrente nominale di almeno 200 mA.

Tramite il jumper J3 (**Fig. 2**) si può cambiare il valore di tensione richiesto: se viene chiuso a destra, è lo stadio di potenza interno a generare i 5 V necessari al funzionamento della board e della periferica USB eventualmente connessa, pertanto la scheda può essere alimentata con una tensione compresa tra i 7 e i 15 V; invece quando il jumper è chiuso sul lato sinistro la scheda va alimentata con 5 volt stabilizzati e il regolatore on-board è escluso.

Per l'audio, sono previste due prese jack da 3,5 mm stereo, una per l'ingresso e una per l'uscita: la prima accetta il segnale da campionare e trasformare in stream audio su ethernet e la seconda fornisce l'audio estratto da uno stream catturato dalla LAN. L'audio del circuito è compatibile con i segnali *consumer audio* a -10 dBV (livello nominale di 316 mV rms) etichettati spesso come *Line-In* e *Line-Out* (ad esempio quelli presenti sui comuni impianti Hi-Fi). Sul lato ethernet, accessibile dal connettore RJ45, la scheda può funzionare a 10 Mbps o 100 Mbps, ma è compatibile con le reti Giga-

bit (1.000 Mbps) pur funzionando al massimo a 100 Mbps.

I due LED presenti ai lati del connettore indicano la presenza di un link instaurato e correttamente funzionante (LED verde) ed attività di traffico regolare (LED arancione).

Passando al lato opposto della scheda troviamo il connettore USB Host, capace di ospitare una comune chiavetta USB (*device Mass Storage*). La tipologia di device supportati è molto ampia: la maggior parte delle comuni chiavette o degli adattatori *micro-SD USB* possono essere utilizzati senza problemi. La compatibilità dipende dal tipo di formattazione che questi dispositivi di massa devono avere; il software da noi realizzato è in grado di riconoscere solo formattazioni di tipo FAT16 o FAT32 ed è compatibile con i *long file name*. Non ci sono significative limitazioni sulla capacità dei dispositivi di memorizzazione di massa applicabili all'USB, a patto che possano essere formattati con uno dei file system precedentemente menzionati. Rimanendo su questo lato è possibile trovare 3 pulsanti e 4 LED. Per comodità di trattazione nomineremo i tre pulsanti con: *left*, *center* e *right* (**Fig. 1**). Similmente,

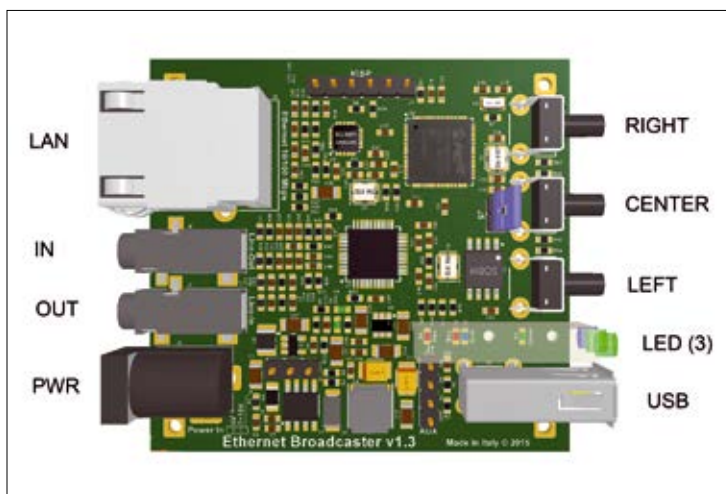


Fig. 1
Comandi e
connessioni
della scheda.



Fig. 2 - Nella figura di sinistra la posizione del jumper consente di alimentare la scheda direttamente con 5 V, invece nell'immagine di destra la scheda può essere alimentata con una tensione compresa tra 7 V e 15 V.

per i LED indicheremo: con *rosso* quello più in alto, *blu* oppure *rosso centrale* quello collocato al centro del light-pipes e *verde* quello collocato più in basso. Durante il normale funzionamento il LED blu lampeggia con una frequenza di circa 1 Hz, dato che questo lampeggio è ottenuto in software senza far ricorso ad interrupt o componenti hardware dedicate; spegnimento o lampeggio irregolare evidenziano un'anomalia nel funzionamento della scheda. Gli altri LED hanno significati differenti in relazione al contesto e saranno descritti nei prossimi paragrafi.

Esplorando l'interno della scheda troviamo il connettore J7, a 4 contatti, le cui funzionalità sono riportate nella **Tabella 1**. Questo connettore può essere utilizzato per estendere le funzionalità del software comunicando tramite protocollo I²C o UART, oppure in bit banging con un dispositivo esterno. Troviamo anche il connettore (J1) di programmazione del microcontrollore principale, utilizzato in fabbrica per installare il bootloader all'interno di un'area della memoria Flash riservata a tale scopo. Gli aggiornamenti del firmware potranno avvenire tramite apposite pagine web, come spiegato successivamente.

SCHEMA ELETTRICO

Il trasmettitore/ricevitore audio su ethernet è dotato di

un microcontrollore molto potente siglato U2: si tratta del PIC32MX695F512H della Microchip, che rappresenta il top di gamma della famiglia 32MX in quanto è dotato di 512 kByte di memoria programma e 128 kByte di memoria RAM, inoltre è dotato di tutto l'hardware necessario per interfacciarsi con un dispositivo USB 2.0 ed un *PHY Ethernet* per realizzare una connessione 10/100 Mbps. La CPU di tipo MIPS M4K® Core a 32-bit, con l'impiego della cache

U7, l'Inter Integrated Circuit (I²C) per dialogare con l'esterno tramite il connettore J7 e l'*Universal Asynchronous Receiver-Transmitter (UART)*, utilizzata anche essa per comunicare opzionalmente con U7 o l'ambiente esterno tramite J7.

Continuando la descrizione del circuito stampato, il secondo componente di maggiore importanza è U7 (VS1063), ossia un *MP3 slave processor* realizzato dalla *VLSI Solution*; lo abbiamo già usato in altri progetti e si

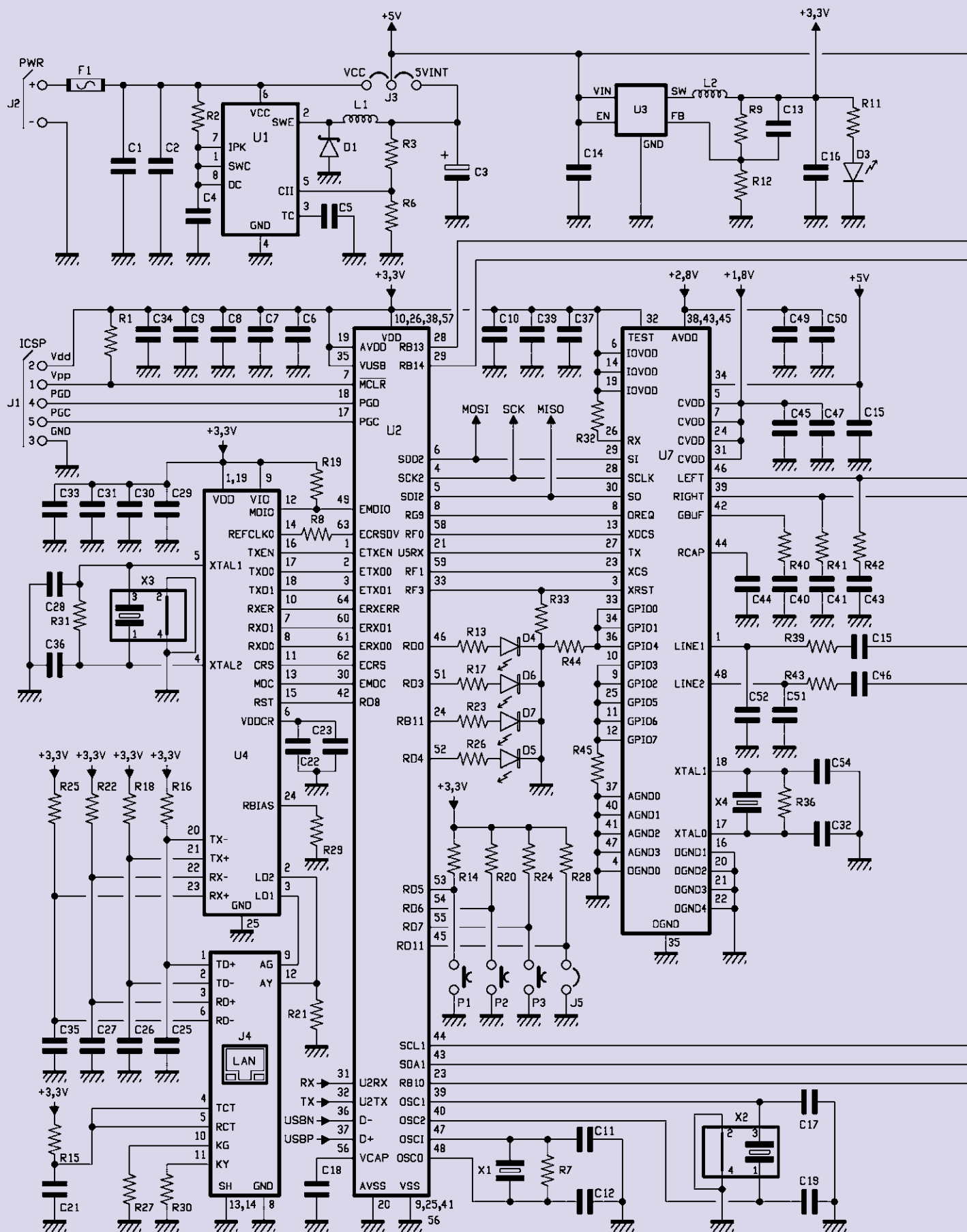
Tabella 1 - Descrizione del connettore J7.

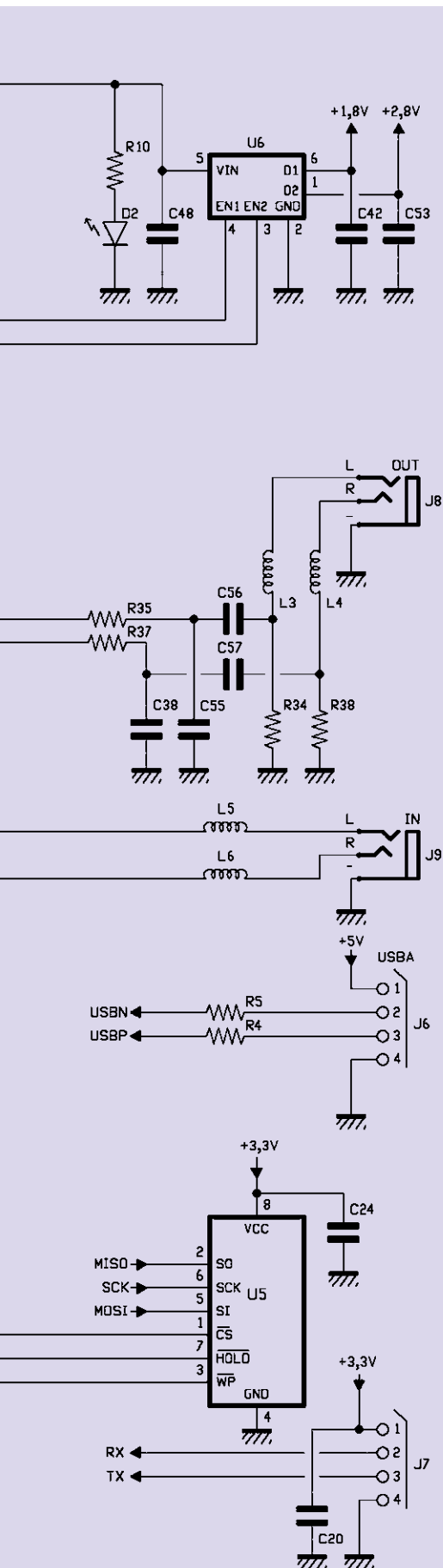
#	Nome	Funzione primaria	Funzione alternativa 1	Funzione alternativa 2
1	Vdd	Alimentazione 3,3 V		
2	RX/SDA/GPIO1	UART RX	I ² C dato	I/O General Purpose 1
3	TX/SCL/GPIO2	UART TX	I ² C clock	I/O General Purpose 2
4	GND	Riferimento di massa		

e della predizione dei salti, può far raggiungere velocità di calcolo di 105 DMIPS con soli 80 MHz di clock; per questo è adatto a gestire diversi flussi audio su rete ethernet come richiesto dalla nostra applicazione. La temperatura operativa può essere compresa tra i -40 e 105 °C, il che rende adatta la nostra scheda a lavorare anche in ambienti difficili e all'esterno.

Il microcontrollore è dotato di periferiche standard come la *Serial Peripheral Interface (SPI)* usata per comunicare con il chip

fa apprezzare per la possibilità di riprodurre o registrare flussi audio codificati in diversi formati, tra cui il *Moving Picture Expert Group-1/2 Audio Layer 3 (MP3)* e l'Ogg Vorbis. I due componenti principali, U2 ed U7 dialogano per mezzo di un canale SPI condiviso con U5; in particolare, del microcontrollore usiamo la periferica SPI numero 2 ed ulteriori quattro linee digitali: xCS, XDACS, RESET e DREQ. Le prime tre sono segnali d'ingresso per U7 e corrispondono rispettivamente al chip select (SPI mode) per la se-



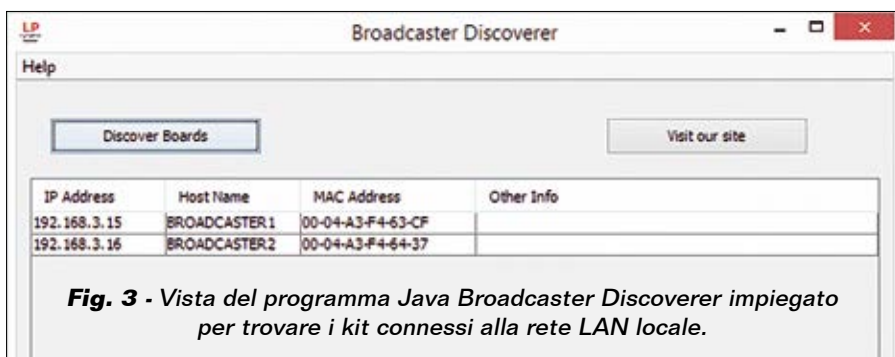


zione controllo, al *chip select* per la sezione dati e ad un impulso di reset per sincronizzare il decoder e il microcontrollore allo startup. L'ultimo segnale, etichettato DREQ, corrisponde a un output del VS1063 ed ha l'obiettivo di segnalare al microcontrollore la disponibilità di spazio nella propria memoria FIFO, per ricevere sia nuovi comandi che nuovi dati. Sfruttando la modalità SPI mode naturale, per il VS1063 diventa molto difficile impiegare lato microcontrollore la *Direct Memory Access (DMA)*. Per poter ottimizzare l'impiego delle risorse disponibili nel PIC, il VS1063 è stato collegato ad U2 anche

basa la dinamicità delle pagine web. Il chip U5 è controllato anche da ulteriori due segnali elettrici generati da U2, WP e HOLD, che consentono di bloccare od inibire momentaneamente la scrittura di nuovi contenuti al suo interno.

Per quanto riguarda la sezione d'alimentazione, il circuito è dotato di ben due regolatori *Low-dropout regulator (LDO)*, ospitati dallo stesso package (SOT236) che nello schema elettrico è chiamato U6.

Quest'ultimo, in sigla AP7312-1828, è un regolatore estremamente preciso ed economico che fornisce in uscita due tensioni



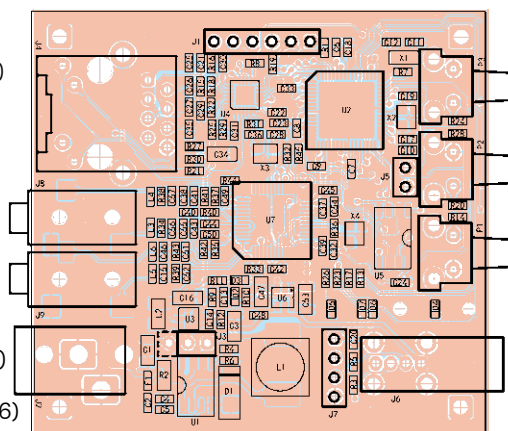
tramite una periferica UART dedicata, che all'occorrenza rende l'impiego della DMA (soprattutto in fase di registrazione) molto più agevole. Infatti è possibile impostare il riempimento di un apposito buffer ed alla saturazione di quest'ultimo generare un interrupt che avvisa il *main*; ciò permette di risparmiare il polling per ogni byte generato dal decoder.

Sullo stesso bus SPI è collegata la memoria Flash SST25VF016B siglata U5, che ha un'architettura CMOS Flash ed una capacità di 16 Mbit (vale a dire 2 MByte), quindi è in grado di ospitare tutte le pagine web dell'applicazione e le librerie *jQuery* su cui si

stabilizzate, rispettivamente di 1,8 V e 2,8 V, con una corrente massima di 150 mA per uscita. Le due tensioni corrispondono ai valori nominali necessari per alimentare il VS1063, rispettivamente per la sezione *DSP Core* e per la sezione analogica. Invece, per alimentare il microcontrollore principale ed il resto della componentistica a 3,3 V (*PHY Ethernet* ed *SPI Flash*) si è scelto di usare un *High Efficiency, Low Ripple, Step-down Converter* o più comunemente un *DC-DC converter* ad alta efficienza (U3) capace di erogare fino ad 1 A di corrente da una tensione d'alimentazione massima di 5,5 V. L'impiego di uno switching al posto di un

Elenco Componenti:

R1: 10 kohm (0603)
 R2: 0,33 ohm (1206)
 R3: 4,7 kohm (0603)
 R4, R5: 27 ohm (0603)
 R6: 1,5 kohm (0603)
 R7: -
 R8: 27 ohm (0603)
 R9: 1 Mohm (0603)
 R10: 2,2 kohm (0603)
 R11: 1 kohm (0603)
 R12: 220 kohm (0603)
 R13: 330 ohm (0603)
 R14: 10 kohm (0603)
 R15: 0 ohm (0603)
 R17: 470 ohm (0603)
 R16, R17, R25: 49,9 ohm (0603) 1%
 R19: 1,5 kohm (0603)
 R20, R21, R24, R28: 10 kohm (0603)
 R22: 49,9 ohm (0603) 1%
 R23, R26, R27: 470 ohm (0603)
 R29: 12,1 kohm (0603) 1%
 R30, R39, R43: 470 ohm (0603)
 R31, R36: 1 Mohm (0603)
 R32, R33: 100 kohm (0603)
 R34, R38: 10 kohm (0603)
 R35, R37, R40: 100 ohm (0603)
 R41, R42: 20 ohm (0603)
 R44, R45: 100 kohm (0603)
 C1: 10 μ F 16 VL ceramico (1206)
 C2, C4: 100 nF ceramico (0603)
 C3: 100 μ F 16 VL ceramico (1206)
 C5: 470 pF ceramico (0603)
 C6 $\div$ C10: 100 nF ceramico (0603)
 C11, C12: 33 pF ceramico (0603)
 C13: 18 pF ceramico (0603)
 C14, C20, C22: 100 nF ceramico (0603)
 C15: 1 μ F ceramico (0603)
 C16: 100 μ F 16 VL ceramico (1206)
 C17: 33 pF ceramico (0603)
 C18: 10 μ F ceramico (0603)
 C19, C32, C54: 33 pF ceramico (0603)
 C22: 1 μ F ceramico (0603)
 C23: 470 pF ceramico (0603)
 C24, C33: 100 nF ceramico (0603)
 C25 $\div$ C28: 10 pF ceramico (0603)
 C29 $\div$ C31: 100 nF ceramico (0603)
 C34: 100 μ F 16 VL ceramico (1206)
 C35, C36: 10 pF ceramico (0603)
 C37, C39, C42: 100 nF ceramico (0603)
 C38, C55: 3,3 nF ceramico (0603)
 C40: 47 nF ceramico (0603)
 C41, C43: 10 nF ceramico (0603)
 C44, C46: 1 μ F ceramico (0603)
 C45, C48 $\div$ C50: 100 nF ceramico (0603)
 C47: 100 μ F 16 VL ceramico (1206)
 C51, C52: 10 nF ceramico (0603)
 C53: 100 μ F 16 VL ceramico (1206)
 C56, C57: 10 μ F ceramico (0603)
 U1: MC34063ADE4
 U2: PIC32MX695F512H-80I/MR (MF1185)
 U3: NCP1529
 U4: LAN8720ACP



U5: SST25VF016B-75-4I-S2AF
 U6: AP7312-1525W6-7
 U7: VS1063
 X1: Quarzo 32.768 kHz
 X2: Quarzo 20 MHz
 X3: Quarzo 25 MHz
 X4: Quarzo 12.288 MHz
 L1: Induttanza 220 μ H 500mA
 L2: Induttanza 4,7 μ H (1210)
 L3 $\div$ L6: Ferrite 600 Ohm @100Hz (0603)
 P1 $\div$ P3: Microswitch 90°
 F1: Fusibile 500mA (0603)
 D1: CDDBA120-G
 D2 $\div$ D4: LED verde (0603)
 D5, D7: LED rosso (0603)
 D6: LED blu (0603)

Varie:

- Plug alimentazione
- Presa Jack stereo da CS (2 pz.)
- Connettore USB-A Femmina da CS
- Connettore Ethernet
- Strip maschio 2 vie
- Strip maschio 3 vie
- Strip maschio 4 vie
- Strip maschio 6 vie
- Jumper
- Circuito stampato

LDO in questo caso si è reso necessario per via del maggiore consumo energetico e quindi in un'ottica di ottimizzazione delle risorse disponibili un DC-DC converter è più efficiente di un LDO anche se il pegno da pagare è un maggiore residuo di ripple sull'uscita ed un maggior numero di componenti esterni (C19, L2, R9, R12, C20 e C21), ma fortunatamente abbiamo potuto ugualmente compiere questa scelta in quanto non sono presenti componenti analogiche che funzionano direttamente a 3,3 V per le quali abbiamo invece usato un LDO come descritto in precedenza. Per completare la descrizione della sezione di potenza, l'ultimo componente impiegato è l'MC34063 (U1): è anche esso un DC-DC converter, il cui compito è accettare un'alimentazione in ingresso compresa tra i 7 V e i 15 V e ridurla a 5 V (erogando al massimo 1,5 A). Partendo dai 5 volt, U3 e U6 ricavano le rispettive tensioni. La tensione di 5 V generata da U1 è utilizzata anche per fornire alimentazione alla porta USB.

Ultimo, ma non per importanza, è il LAN8720 siglato U4, che realizza la parte funzionale della connessione Ethernet. Si tratta di un *Single Chip Ethernet Physical Transceiver (PHY)*. Dei componenti che lo contornano, R19 serve come pull-up della linea MDIO, un po' come avviene nel caso dell'IPC; R21 serve a operare il pull-down in fase di accensione per inizializzare il LAN8720 in modo che sia il chip a generare i 50 MHz richiesti dall'RMII, mentre R29, in combinazione con R16, R18, R22 e R25, serve a stabilire la corrente nei trasformatori del connettore RJ45. Il LAN8720 si interfaccia al MAC interno al microcontrollore U2 tramite un'interfaccia standard per l'intercon-

nessione dei due dispositivi detta *Reduced Media Independent Interface* (RMII). Per funzionare da PHY RMII, U4 necessita del quarzo X3 da 25 MHz, la cui frequenza di oscillazione è moltiplicata per 50 MHz dal PLL interno all'integrato; il clock così ottenuto è usato per pilotare il MAC di U2 tramite la linea nominata REFCLKO. Trattandosi di un segnale con una frequenza molto elevata e con una forma d'onda teoricamente quadrata, il clock produce una notevole quantità di armoniche, soprattutto se l'impedenza d'uscita e l'impedenza d'ingresso dei dispositivi che si affacciano su tale segnale non sono adatte; per questo motivo è stata inserita R8, che ha il compito di ridurre i disadattamenti di impedenza tra la sorgente e la destinazione limitando gli effetti nocivi delle onde riflesse.

Per ottenere gli 80 MHz necessari a U2 si è scelto di usare un quarzo X2 da 20 MHz, che tramite opportune sequenze di moltiplicazione e divisione selezionabili da software produce il segnale ad 80 MHz per la CPU ed il segnale a 48 MHz necessario al funzionamento dell'USB. Nello schema è presente anche il quarzo X1 da 32.768 kHz, che viene impiegato per controllare lo scorrere del tempo nell'RTC interno al PIC al fine di poter avere delle funzionalità software di tempo e calendario.

L'ultimo quarzo presente nella scheda è X4, da 12.288 MHz che è impiegato per far funzionare U7; anche quest'ultimo chip ha al proprio interno un moltiplicatore 2x con cui ricava il clock di riferimento a 24.576 MHz. Tale frequenza è quella consigliata per eseguire correttamente la codifica o la decodifica di flussi audio a 48 kHz (24.576 MHz diviso 512). Il circuito è completato da oppor-

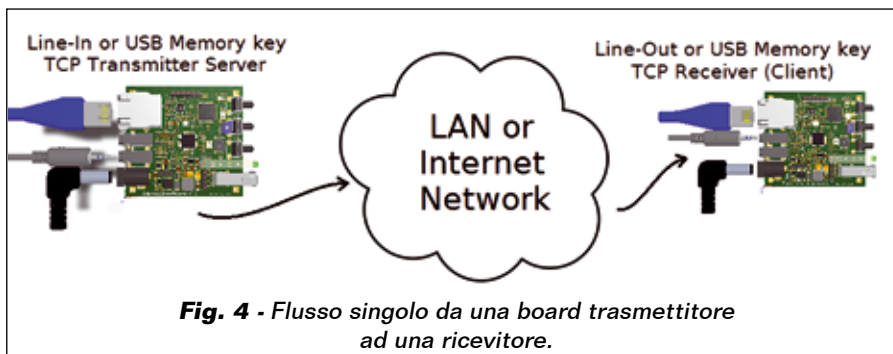


Fig. 4 - Flusso singolo da una board trasmettitore ad una ricevitore.

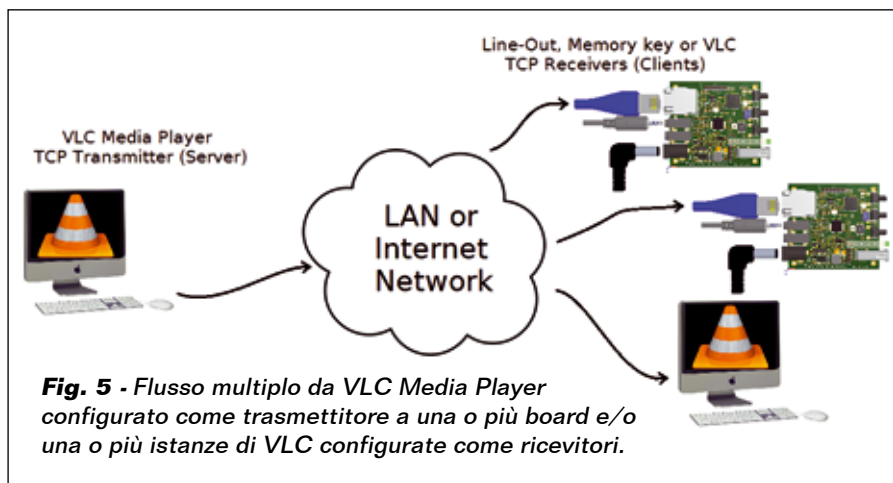


Fig. 5 - Flusso multiplo da VLC Media Player configurato come trasmettitore a una o più board e/o una o più istanze di VLC configurate come ricevitori.

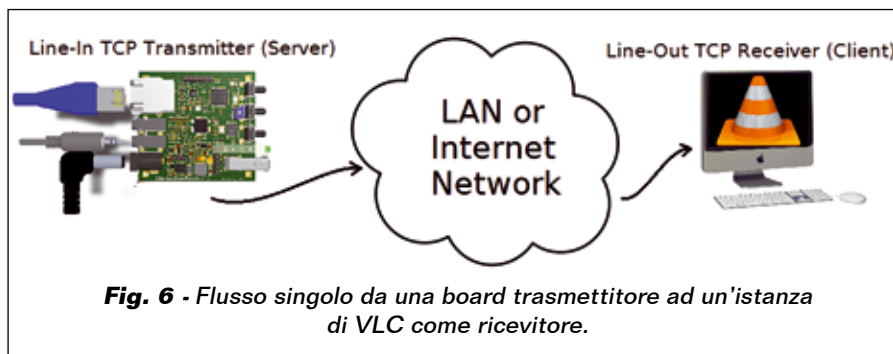


Fig. 6 - Flusso singolo da una board trasmettitore ad un'istanza di VLC come ricevitore.

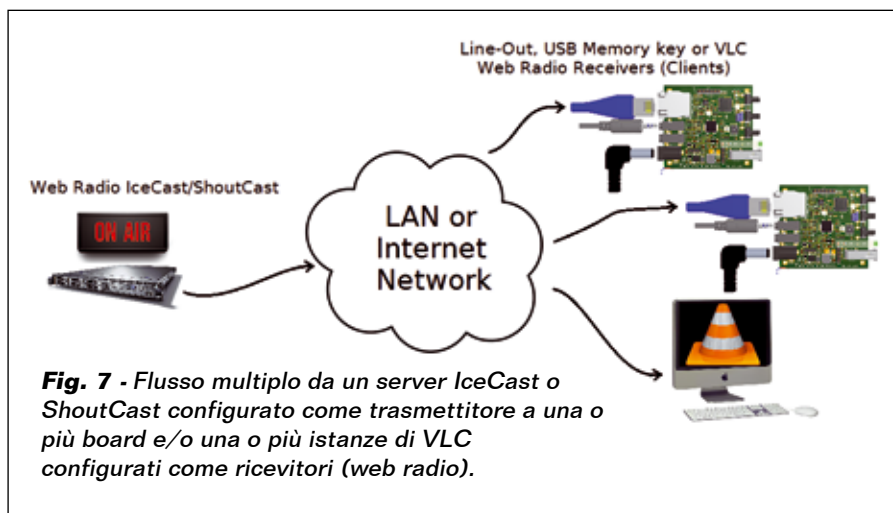


Fig. 7 - Flusso multiplo da un server IceCast o ShoutCast configurato come trasmettitore a una o più board e/o una o più istanze di VLC configurati come ricevitori (web radio).

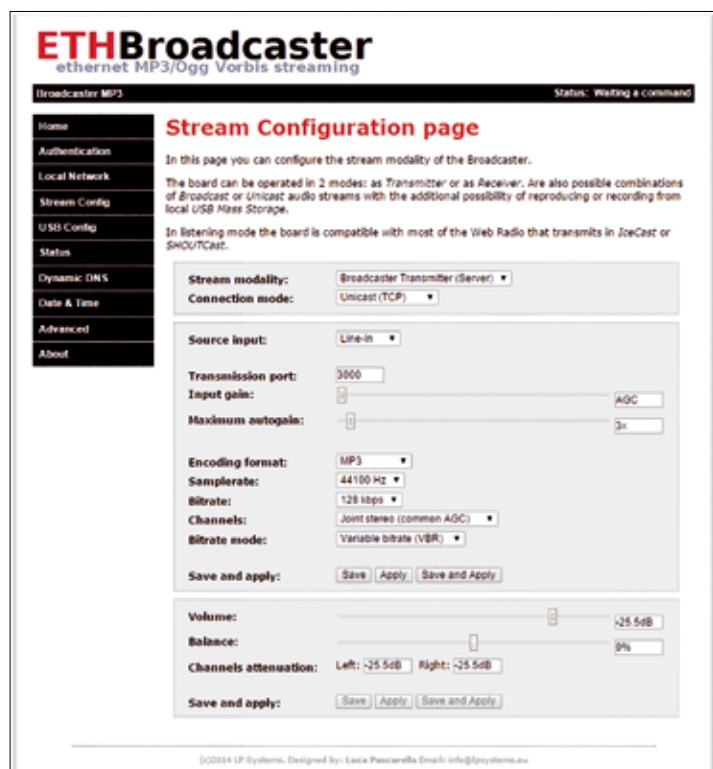
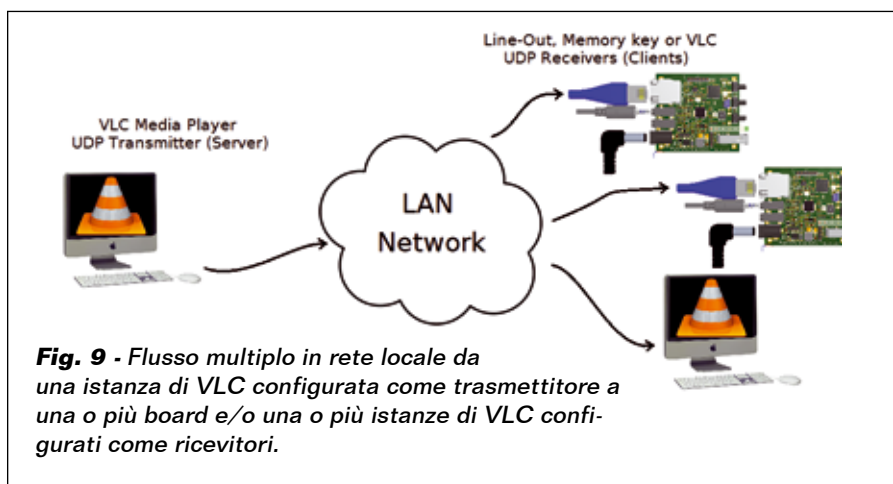
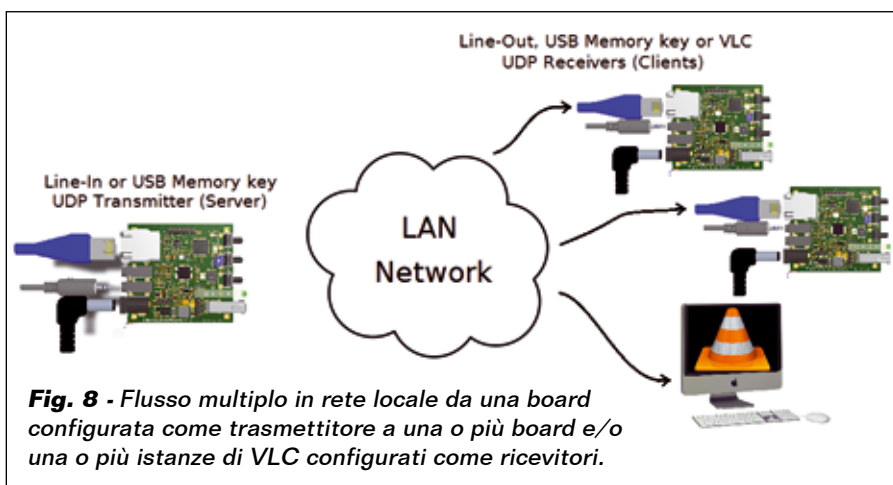


Fig. 10
Vista della
pagina di
configurazione
Stream
Configuration
nella modalità
trasmettitore.

tuni filtri in serie agli ingressi e alle uscite analogiche ed in particolare dall'impiego su ognuno dei segnali audio di ferriti da 600 Ω a 100 MHz che hanno l'obiettivo di attenuare i disturbi elettrici provenienti dalle componenti ad alta frequenza del circuito, tra cui i segnali generati dal PHY Ethernet.

SCENARI D'USO

Per iniziare ad ascoltare la propria musica preferita è sufficiente disporre di un kit, collegarlo alla propria rete ethernet ed accenderlo. Collegandosi all'apposita pagina web (per trovare i dispositivi connessi basta usare l'applicazione per PC scritta in Java *Broadcaster Discoverer* come da Fig. 3) sarà possibile controllare lo stato del dispositivo ed avviarlo. Prima di descrivere qualche scenario d'uso vediamo le possibili modalità di configurazione. Il dispositivo può funzionare da trasmettitore (Server) o da ricevitore (Client) ognuna delle due modalità può essere usata per comunicazioni *Unicast* (basata su protocollo TCP) o *Broadcast* (basata su protocollo UDP); ne consegue che esistono quattro possibili configurazioni. In realtà il dispositivo, essendo compatibile con *IceCast* e *ShoutCast*, può essere programmato per funzionare anche come ricevitore di *web radio*; inoltre, essendo compatibile con *VLC Media Player*, il software standard garantisce una vasta gamma di scenari d'uso. In modalità *Unicast* ovvero *punto-punto*, quindi impiegando il protocollo TCP è possibile:

1. trasmettere da una board a un'altra (Fig. 4);
2. trasmettere da VLC (Fig. 5) e ricevere da una o più board, ovvero ricevere da un computer con installato VLC;
3. trasmettere da una board e

- ricevere da un computer con installato VLC (Fig. 6);
- trasmettere da un server IceCast o ShoutCast e ricevere da una o più board o da un computer con VLC (Fig. 7).

In modalità *Broadcast* ovvero *punto-multipunto*, quindi impiegando il protocollo UDP è possibile:

- trasmettere da una board (Fig. 8) e ricevere da una o più board, oppure ricevere da uno o più computer con VLC.
- trasmettere da un computer con installato VLC (Fig. 9) e ricevere da una o più board, ovvero ricevere da uno o più computer con installato VLC.

CONFIGURAZIONE

Per poter selezionare la modalità di funzionamento desiderata bisogna accendere una board, collegarla alla propria LAN, scoprire l'IP assegnato dinamicamente tramite DHCP (può tornare utile il programma di Fig. 3) e collegarsi, tramite browser, all'apposita pagina web fornita dalla board. La prima pagina web è pubblica e quindi accessibile senza effettuare il login; invece le pagine di configurazione sono protette da username e password. A riguardo notate che sono previsti due ruoli: utente e amministratore; il primo può visualizzare tutte le impostazioni ed avviare o interrompere una comunicazione. Il secondo, godendo di privilegi superiori, può modificare ogni parametro della scheda compresi quelli di rete. Per impostare una modalità di funzionamento bisogna accedere alla voce di menu *Stream config*; ad ogni variazione degli elementi di *Stream modality* e *Connection mode* la pagina modificherà dinamicamente l'elenco dei parametri che è necessario inserire per le rispettive modalità; per questo è necessario usare

Fig. 11
Vista della
pagina di
configurazione
Stream
Configuration
nella modalità
ricevitore.

In corrispondenza
della voce
Remote address
si può inserire
l'indirizzo
remoto del kit
trasmettitore
o l'URL di una
web radio; in
quest'ultimo
caso bisogna
configurare
anche la voce
Remote resource,
che corrisponde
al path della
risorsa richiesta
al server remoto.

un browser con supporto per HTML5 e JavaScript abilitato. Il primo campo, *Stream modality*, consente di selezionare la modalità di ricevitore o trasmettitore indicata anche come server o client per ricordare il ruolo che il software assume rispetto alla rete. Il secondo campo, *Connection mode*, consente di selezionare il protocollo di rete sul quale verranno instradati i messaggi audio; come anticipato in precedenza, il protocollo TCP consente di effettuare collegamenti *punto-punto*. Viceversa, UDP consente di trasmettere in *broadcast* all'interno di una rete locale. Quando la scheda è configurata da trasmettitore, quindi come in Fig. 10, la pagina web consentirà di selezionare la sorgente da trasmettere, che può alternativemente essere l'ingresso *Line-In* oppure i file .mp3 presenti sulla chiavetta USB eventualmente connessa. Quando è selezionato l'ingresso analogico *Line-In* è inoltre possibile selezionare la modalità di codifica dell'audio

(MP3 o Ogg Vorbis) nonché la qualità, il bit-rate ed il guadagno dato all'amplificatore di ingresso, in modo da poter catturare sia segnali deboli che forti. Un ultimo parametro è *Transmission Port*, che rappresenta, in combinazione con l'indirizzo IP assegnato alla scheda, il valore da impiegare nei ricevitori per creare un collegamento audio fra trasmettitore (server) e ricevitore (client). Per applicare i cambiamenti sono stati previsti tre pulsanti che hanno le seguenti funzionalità:

- Save*, consente di salvare in memoria i nuovi parametri, lasciando inalterati quelli in corso, quindi se vengono effettuate delle modifiche durante una connessione già stabilita, questa resterà in vigore e i nuovi parametri saranno effettivi dal successivo riavvio;
- Apply*, consente di testare le impostazioni senza perdere quelle presenti in memoria, quindi riavviando la scheda i valori temporanei verranno

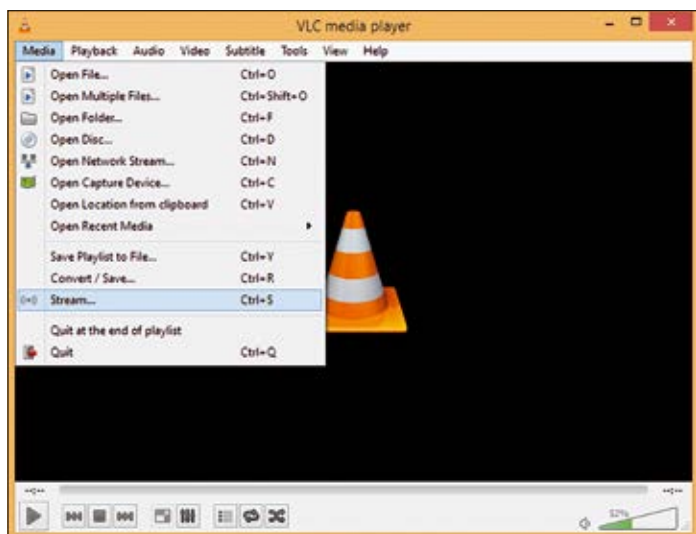


Fig. 12

persi in favore di quelli presenti in memoria;
3. *Save and Apply*, consente di effettuare in un unico passaggio l'applicazione ed il salvataggio dei nuovi parametri, quindi se al momento del clic è presente una connessione, questa verrà interrotta in favore dei nuovi parametri.

Per poter configurare una board come ricevitore (**Fig. 11**) bisogna impostare il campo *Stream modality* su *Broadcaster Receiver (Client)*; in questo caso la pagina web mostrerà dei nuovi parametri tra cui la possibilità di selezio-

nare il canale d'uscita, che potrà essere:

1. Out;
2. USB;
3. Out più USB.

Inoltre sarà necessario inserire: *Remote address*, che rappresenta l'indirizzo IP della scheda remota o l'URL del server web e *Remote resource*, che nel caso di web radio o VLC come trasmettenti rappresenta la risorsa richiesta (anche detta *path*); infine, in *Remote port* va scritto il numero di porta impostato nel dispositivo trasmettente. Nella pagina saranno presenti anche:

1. *Auto connection*; rappresenta un flag che se abilitato consente alla scheda la connessione automatica all'accensione;
2. *Reconnection times*; ovvero il numero di tentativi di riconnessione in caso di connessione persa, può essere disabilitato, assumere un valore numerico compreso tra 1 e 50 oppure illimitato;
3. *Reconnection delay*; rappresenta il tempo necessario da attendere prima di avviare una connessione;
4. *Connection timeout*: rappresenta il limite di tempo superiore oltre il quale un ritardo sarà

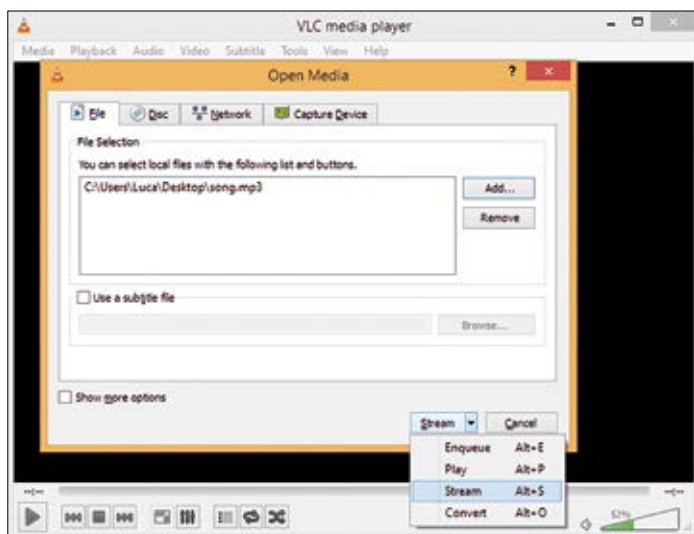


Fig. 13

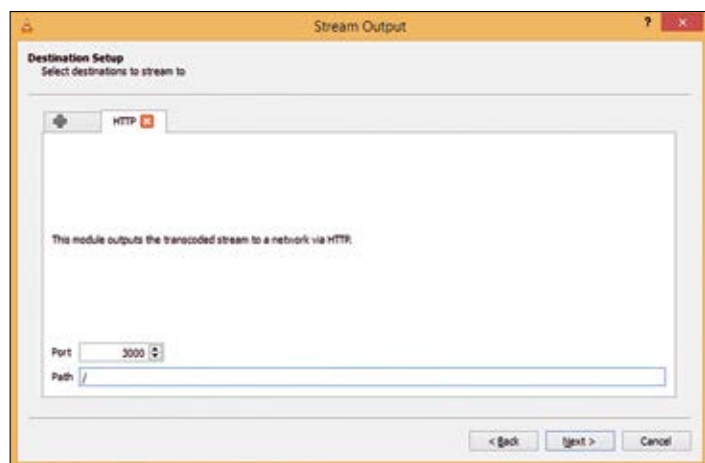


Fig. 14 - La nuova tab HTTP è stata creata; ora bisogna selezionare la porta di trasmissione ed eventualmente assegnare un nome alla risorsa oppure lasciare il carattere / (slash).

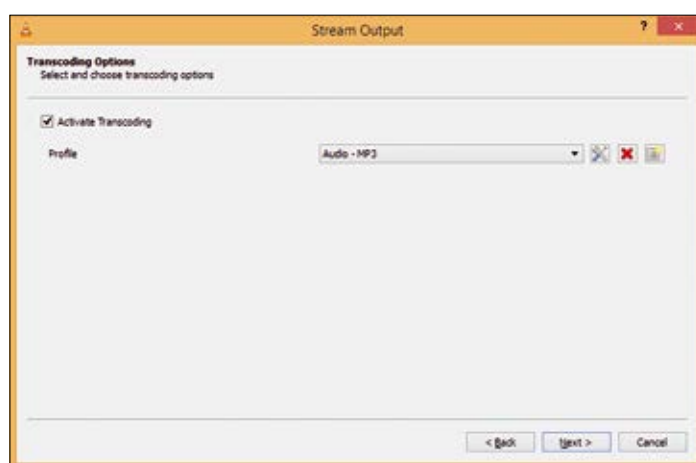


Fig. 15 - Attivazione della transcodifica.

considerato come una caduta di connessione.

Infine gli ultimi parametri relativi a *Bass* e *Treble* consentono di modificare il livello dei toni bassi e dei toni alti. Per ottenere una maggiore flessibilità è possibile adoperare le board in combinazione con *VLC Media Player*; in tal caso bisognerà impostare VLC come trasmettitore o come ricevitore.

Proseguendo per ordine, vediamo il caso in cui VLC funziona da trasmettitore.

Prima di tutto bisogna avviare VLC e selezionare *Stream...* dal menu *Media* (si ottiene lo stesso con la shortcut *Ctrl+S*) come mostrato in **Fig. 12**, quindi selezionare i file audio da trasmettere, cliccare sul pulsante che apre il menu a tendina e poi su *Stream* (**Fig. 13**). A questo punto una nuova finestra di dialogo chiederà di proseguire con la configurazione.

Per impostare la modalità *punto-punto* compatibile con una board è fondamentale selezionare HTTP in corrispondenza di *New destination*. Cliccando poi sul pulsante *Add* nella stessa finestra di dialogo si crea una nuova tab HTTP (**Fig. 14**) dalla quale sarà possibile selezionare la porta di trasmissione ed eventualmente assegnare un nome alla risorsa oppure lasciare il carattere / (slash). L'attivazione della transcodifica di **Fig. 15** consente di modificare il codec dell'audio; al momento sono supportati i due formati *Audio - MP3* ed *Audio - Vorbis (Ogg)*. Infine, cliccando sul pulsante *Stream* si è pronti alla trasmissione. Viceversa, per configurare VLC come ricevitore punto-punto: allo scopo bisogna avviare VLC e cliccare su *Open Network Stream...* dal menu *Media* (Shortcut *Ctrl+N*), quindi,

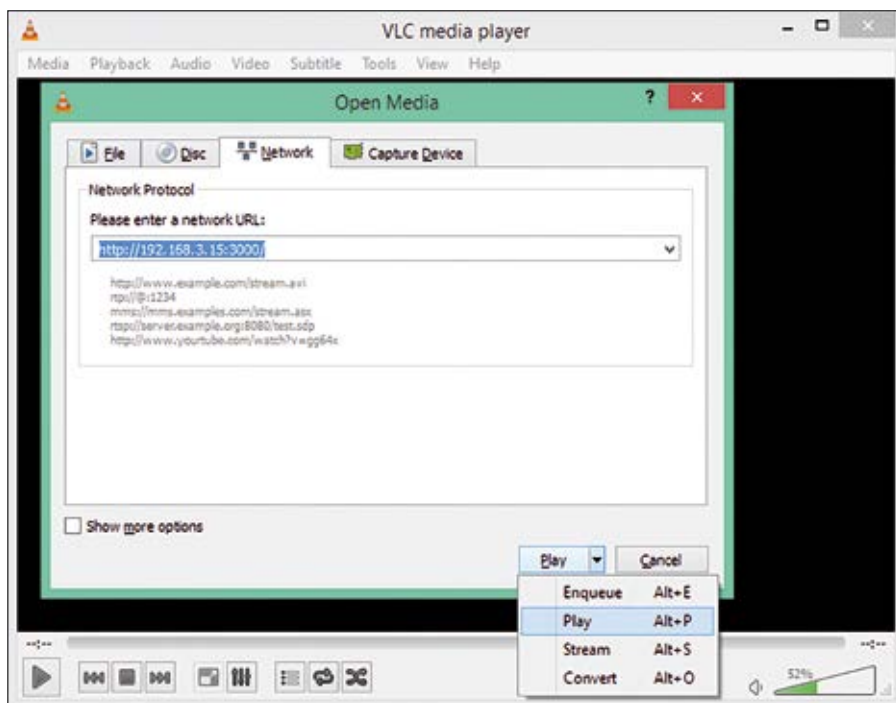


Fig. 16 - Inserimento dell'URL remoto. Ad esempio se la board trasmittente ha IP 192.168.3.15 e trasmette sulla porta 3000 in VLC dovrà essere inserito `http://192.168.3.15:3000/` come URL remoto.

nella finestra di dialogo che si apre, e inserire l'URL corrispondente alla risorsa remota, che nel nostro caso coinciderà con l'IP assegnato ad una board ed al numero di porta; il protocollo dovrà essere HTTP per compatibilità con IceCast e ShoutCast, esattamente come mostrato in **Fig. 16**. Analogamente ai casi precedenti, VLC può essere configurato per trasmettere o ricevere in broadcast su una rete locale sfruttando il protocollo UDP. La procedura da seguire per la configurazione del player come trasmettitore sarà, avvio di VLC, selezione di *Stream...* aggiunta del profilo RTP Audio/Video Profile come in **Fig. 17** e configurazione del profilo RTP/AVP; in questo caso per fare in modo che tutti i dispositivi connessi alla LAN siano capaci di ricevere il flusso è necessario selezionare l'indirizzo di rete broadcast terminante con 255, ad esempio se la board ricevente ha IP 192.168.3.15, in VLC dovrà essere

inserito 192.168.3.255, esattamente come in **Fig. 18**.

Infine la procedura per configurare VLC come ricevitore broadcast è: avvio di VLC selezione della voce *Open Network Stream...* dal menu *Media* (Shortcut *Ctrl+N*); a questo punto si presentano due scenari:



per il MATERIALE

La scheda Modulo per filodiffusione audio su ethernet (Cod. FT1185M) viene fornita montata e collaudata. Può essere acquistata presso Futura Elettronica al prezzo di Euro 78,00. Il Contenitore in alluminio Tekal 11.29 nero, da forare, (Cod. TEKAL1129) è disponibile separatamente al costo di Euro 9.50.

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

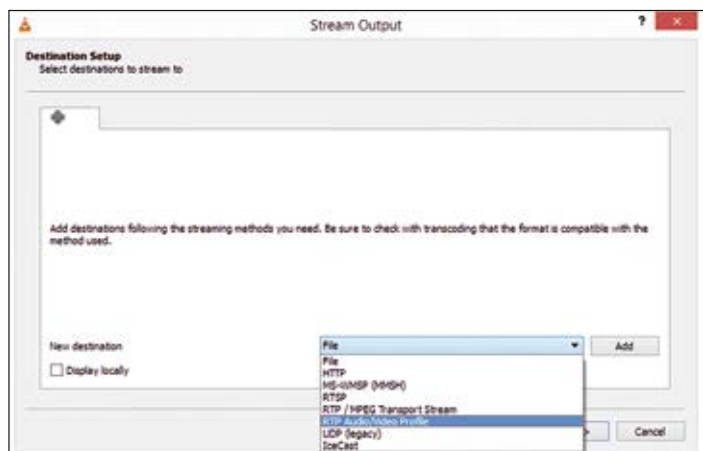


Fig. 17 - Selezione del profilo RTP Audio/Video.

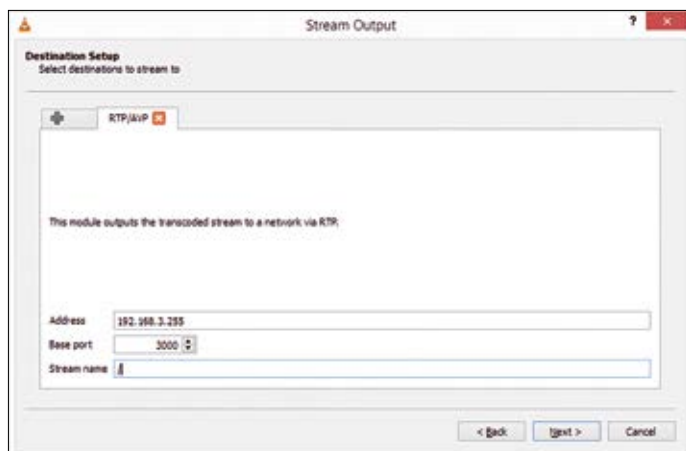


Fig. 18 - Configurazione del profilo RTP/AVP e selezione dell'indirizzo di broadcast, ad esempio 192.168.3.255.

1. per ascoltare un flusso proveniente da una board configurata per trasmettere in broadcast quindi sfruttando il protocollo UDP sarà necessario inserire come URL remoto esattamente la stringa `udp://@:3000`, come mostrato in Fig. 19;
2. per ascoltare un flusso proveniente da un'altra istanza di VLC configurata per trasmettere in broadcast, sempre sfruttando il protocollo UDP, sarà necessario -dalla stessa finestra di dialogo e nella stessa casella mostrata nella Fig. 19- inserire come URL remoto esattamente la stringa `rtsp://@:3000`.

STATO DELLA BOARD

Per conoscere lo stato della board o impartire manualmente dei comandi si può consultare la pagina Stream status che apparirà come in Fig. 20. In corrispondenza di Remote resource verrà mostrato il pulsante Connect o Disconnect nel caso di funzionamento da client, invece Listening o Disconnect nel caso di funzionamento da Server. Page auto-refresh consente di ricaricare all'intervallo di tempo stabilito i risultati della connessione così da poter visualizzare sempre lo stato aggiornato nella pagina senza dover ricaricare l'intero contenuto della pagina manualmente. Le funzionalità di connessione e disconnessione sono offerte anche premendo il pulsante centrale, invece i pulsanti left e right rispettivamente consentono di modificare il valore del volume senza modificare il valore salvato in memoria.

CONCLUSIONI

Concludiamo quest'articolo anticipandovi che nella prossima puntata descriveremo la modalità di aggiornamento del firmware tramite Internet, il resto delle pagine di configurazione e l'ambiente di sviluppo per il software che abbiamo realizzato appositamente per il sistema.

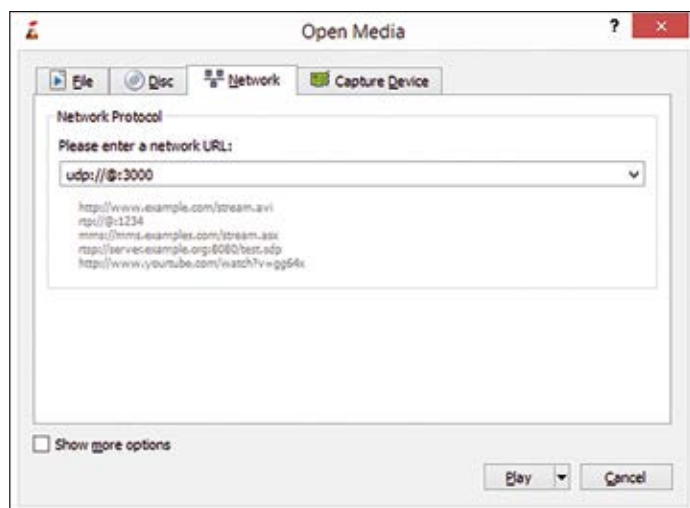


Fig. 19 - Configurazione di VLC per ricevere uno stream broadcast proveniente da una board.

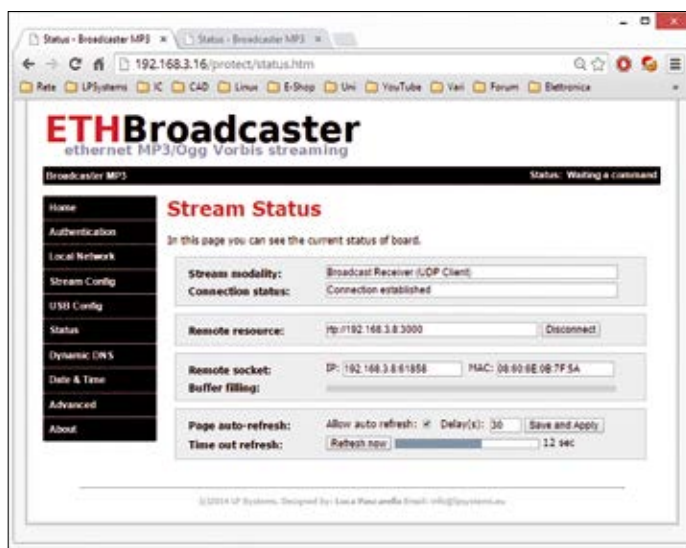
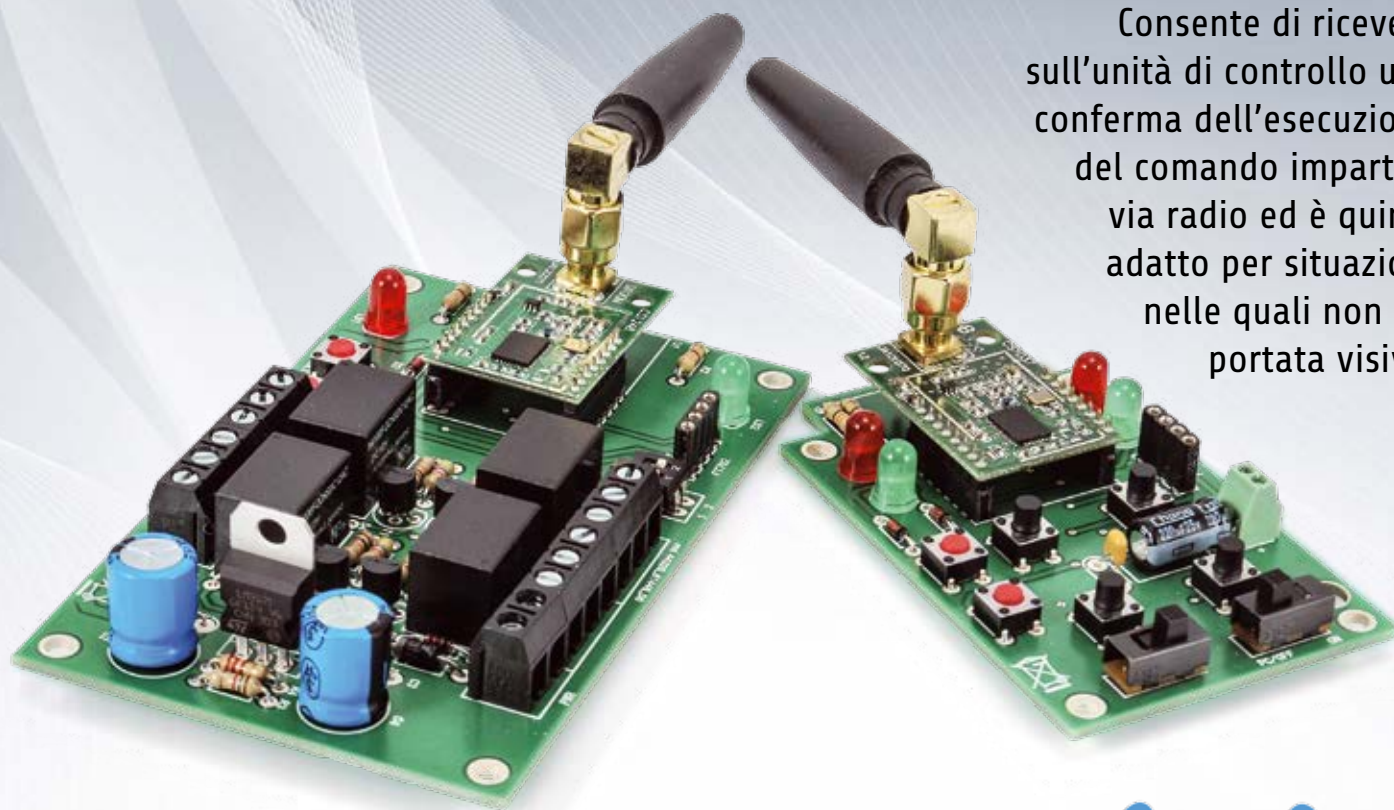


Fig. 25 - Vista della pagina Stream Status in corrispondenza di un ascolto broadcast.



Consente di ricevere sull'unità di controllo una conferma dell'esecuzione del comando impartito via radio ed è quindi adatto per situazioni nelle quali non c'è portata visiva.

RADIOCOMANDO BIDIREZIONALE



di BORIS LANDONI

I progetti di radiocomandi che abbiamo proposto e in generale i dispositivi di radiocomando intesi in senso stretto, sono composti da un'unità trasmittente ed una ricevente: la prima è così chiamata non solo perché invia il comando alla ricevente, ma anche perché trasmette il segnale radio. L'unità ricevente è tale sia perché è soltanto un ricevitore RF, sia perché riceve i comandi, proponendosi di eseguirli. Questo sistema generico ed essenziale ha il limite che chi impartisce un comando trovandosi in una posizione che non permette di verificare a vista che lo stesso venga eseguito, non ha un feedback della propria azione; infatti, se con un TX palmare per apricancello comandiamo l'a-

pertura del cancello motorizzato di casa, di norma vediamo pulsare il lampeggiante e muoversi l'anta (o le ante). Se invece utilizziamo un radiocomando la cui portata supera quella visiva, dobbiamo accontentarci di "credere" che il dispositivo ricevente esegua il comando; ma sappiamo che il comando può non essere trasmesso per un'avaria nel circuito, un'interruzione del collegamento via radio dovuto ad esempio a un disturbo, la mancanza dell'alimentazione del ricevitore ecc. Se vogliamo avere la certezza che il comando venga impartito, ci serve un dispositivo ricevente che sia in realtà un RTX e che in risposta alla ricezione e validazione di un comando, risponda trasmettendo verso la trasmittente

Il modulo RTX

Il modulo su cui si basano le unità del nostro sistema è un RCS1K-868: si tratta di un transceiver radio progettato per applicazioni wireless a basso consumo. L'hardware è basato sul componente Si1000 della Silicon Labs, le cui principali caratteristiche sono una potenza d'uscita modulabile fino a +20dBm (100 mW), un microcontrollore integrato (8051 compatibile) operante con un clock fino a 25 MHz e la modulazione selezionabile tra OOK,FSK,GFSK. Completa la dotazione, un'interfaccia UART a livello TTL-compatibile ma funzionante a 0/3,3V. Il transceiver radio può operare a una frequenza selezionabile tra 240 e 960 MHz, con sensibilità in ricezione di ben 121 dBm e potenza in trasmissione di +20 dBm.

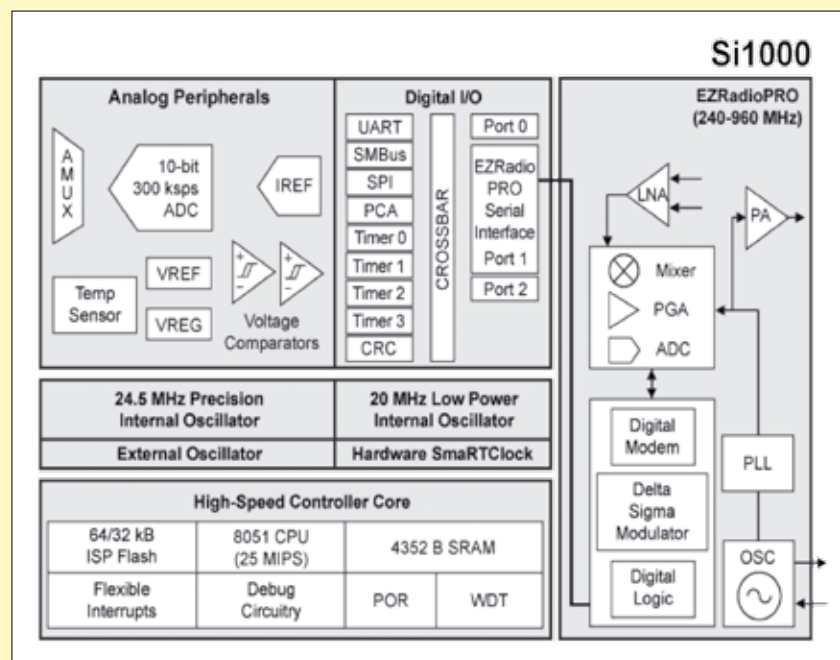
Il modulo, che è basato su di esso, è programmato in modo da implementa-

re funzionalità distinte per la versione trasmittente e quella ricevente. Ogni modulo riporta sui propri contatti, disposti lateralmente sulle due file di 8 contatti a passo 2,54 mm, le linee di interesse.

Nella versione TX (destinata all'unità di comando) abbiamo quattro input per ricevere i comandi da pulsanti e una linea analogica per leggere un potenziometro; nel progetto descritto in queste pagine tale funzione, che nella ricevente gestisce l'uscita PWM, non viene usata.

Ci sono poi gli ingressi di reset e le uscite per i LED di segnalazione. Nella versione RX (unità ricevente) sono presenti gli ingressi per il reset e le uscite per LED e uscite di comando (che sono quattro, ognuna corrispondente a un ingresso o pulsante della trasmittente).

	MIN	TYP	MAX	UNIT
Tensione d'alimentazione	2,2	3	3,6	Vcc
Corrente assorbita (RX Mode/TX Mode)		20/34		mA
Corrente assorbita (TX Mode a +20 dBm)		85		mA
Corrente assorbita (TX Mode a 0 dBm)		10		mA
Corrente assorbita (Sleep mode)		<0,1		µA
Tempo di avvio da Sleep a RX/TX Mode)		<2		µS
Potenza in trasmissione	-3		+20	dBm
Temperatura di funzionamento	-10		+55	°C
Sensibilità in ricezione (Data-Rate=1,2 kb/s)		-112		dBm
Data-Rate		500		kbit/s



la conferma. In un sistema del genere non ha più senso parlare di trasmettitore e ricevitore, almeno stando sul piano delle onde radio, in quanto ciascuna delle unità è ricetrasmittente: il TX trasmette il segnale RF quando viene richiesto l'invio di un comando e riceve la conferma dall'RX, che a sua volta è sia ricevitore che trasmettitore radio.

I termini trasmettitore e ricevitore sono solo riferiti al comando e possono quindi assumere il significato di *unità di comando* (o di controllo) e *unità esecutrice*. In queste pagine proponiamo un sistema di radiocomando con conferma, ovvero bidirezionale, dove l'unità di comando è composta da un modulo RTX radio e quella esecutrice (ricevente) monta lo stesso modulo, solo che esegue i comandi attivando opportunamente i relé (sono quattro) di bordo.

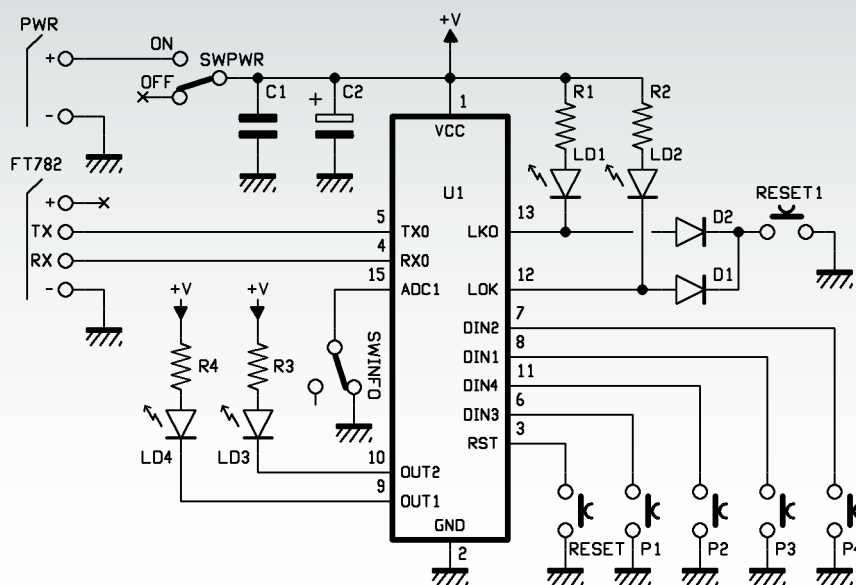
Il modulo non è solamente un ricetrasmittente radio, ma incorpora un microcontrollore 8051, per gestire il protocollo di comunicazione, che può funzionare sia da trasmettitore che da ricevitore. Il protocollo prevede anche una codifica di sicurezza che consente l'esclusività del comando. Con due esemplari -uno impostato per funzionare da trasmettitore e l'altro per fare da ricevente dei comandi- è possibile realizzare un sistema di radiocomando bidirezionale che permette al trasmettitore di conoscere lo stato delle uscite della ricevente sia dopo aver impartito un comando, sia in qualsiasi momento lo si desidera.

Il produttore fornisce coppie di moduli abbinate con lo stesso codice, in modo che uno sia l'unità di controllo e l'altro quella ricevente i comandi. Con una coppia abbiamo realizzato il progetto che trovate in queste pagine

tronica) nel quale giri l'apposito software reso disponibile con i moduli. Di questo parleremo più avanti. Per ora vi basti sapere che nei moduli il produttore carica un



[schema elettrico **UNITÀ DI COMANDO**]



firmware predefinito che imposta la modulazione, la frequenza operativa (nel nostro caso è 868 MHz), la codifica implementata, la modalità operativa (trasmittente o ricevente dei comandi) oltre a un codice che permetta di abbinare le coppie trasmittente/ricevente. Il modulo può essere alimentato con tensioni bassissime (2,2÷3,3 volt) ed è perciò adatto a dispositivi funzionanti a pile o batterie; il suo ingresso per la trasmissione e l'uscita dati sono quindi compatibili con livelli logici 0/3,3V.

L'UNITÀ DI COMANDO

Analizziamo dunque il circuito che trasmette i comandi e riceve dall'unità esecutrice lo stato dell'uscita corrispondente; esso consta del modulo RCS1K-868 contornato da sei pulsanti (quattro dei quali servono a comandare i relé dell'unità ricevente), un deviatore e quattro LED, e alimentato tramite un secondo deviatore che permette di prendere l'alimentazione dai contatti + e - PWR o spegnere il circuito. Ricordiamo che l'alimentazione può essere fornita da due pile stilo (AA) o ministilo (AAA), ovvero da un alimentatore stabi-

lizzato che fornisca da 2,5 a 3,5 Vcc e una corrente continua di almeno 130 milliampere.

Il modulo è, nella coppia, quello che funziona da TX e il suo UART, facente capo ai piedini TX0 ed RX0, si collega al connettore di interfaccia seriale siglato FT782, che permette di collegare l'unità a un computer dotato di porta USB tramite un convertitore TTL/USB.

La funzione dei pulsanti P1÷P4 varia in base alla modalità operativa selezionata con il deviatore SWINFO; quest'ultimo permette di chiudere a massa o lasciare aperto (il resistore di pull-up interno al modulo in questo caso mantiene la linea corrispondente a 1 logico) il piedino ADC1, che decide la modalità, ovvero se il circuito deve inviare un comando che interviene sui relé della scheda esecutrice o se deve solo richiedere a questa lo stato dei relé. Chiudendo SWINFO verso GND si abilita la modalità Attivazione Info di stato delle uscite dell'RX e cambio di stato: ciò vuol dire che quando si preme uno dei pulsanti P1÷P4 (P1 vale per il canale 1, P2 per il 2, P3 per

il 3 e P4 per il canale 4), l'uscita corrispondente nella ricevente viene attivata a tempo nella modalità monostabile, ovvero cambia di stato se nella ricevente è impostato il funzionamento bistabile. Ogni volta che riceve un comando, l'unità esecutrice lo esegue e quando lo ha fatto trasmette verso l'unità di comando un messaggio di esecuzione, che viene segnalato localmente dai quattro LED, i quali hanno le seguenti funzioni:

- LD1 (rosso): si accende quando la trasmittente non riceve risposta dall'unità ricevente;
- LD2 (verde): si accende quando la trasmittente riceve risposta dall'unità ricevente;
- LD3 (verde): uscita relativa al pulsante premuto attiva;
- LD4 (rosso): uscita relativa al pulsante premuto non attiva.

Se il pulsante della trasmittente viene tenuto premuto per meno di 2 secondi si ottiene dalla ricevente solo la segnalazione di stato, mentre se viene premuto per un tempo superiore, oltre a impostare come richiesto lo stato della relativa uscita, la ricevente invia la segnalazione di stato corrispondente.

Lasciando aperto SWINFO (linea ADC1 open) si entra nella moda-

lità Attivazione Info di stato delle uscite della ricevente: questo significa che ogniqualvolta sull'unità trasmittente si preme un pulsante tra P1 e P4, si invia alla ricevente solo la richiesta di stato del canale corrispondente e non viene modificata in alcun modo la situazione dei relé. Le risposte fornite dai LED sono:

- LD1 (rosso): l'unità trasmittente non riceve risposta dalla ricevente;
- LD2 (verde): la trasmittente riceve la prevista risposta dall'unità ricevente.

LD3 e LD4 in questa modalità non vengono usati; inoltre, se il pulsante della trasmittente viene tenuto premuto a lungo, si ha sempre e solo la segnalazione di stato e non l'attivazione del relé corrispondente.

Dallo schema elettrico vedete che i pulsanti di comando dei canali dell'unità esecutrice, ovvero di richiesta dello stato di questi ultimi, sono collegati agli ingressi digitali DIN1, 2, 3, 4 del modulo U1, ciascuno dei quali è provvisto internamente di un resistore di pull-up collegato alla linea di alimentazione. I pulsanti RESET e RESET1 servono rispettivamente a resettare il modulo RCS1K-868 e a caricare il bootloader (questa

funzione serve solo per l'upgrade del firmware, che a noi non interessa): il primo è collegato alla linea RST del modulo, mentre il secondo porta a massa, insieme, i contatti LKO e LOK, normalmente destinati al comando dei LED LD1 e LD2. Per consentire di collegare un solo pulsante a due linee senza cortocircuitarle nel normale utilizzo, abbiamo inserito i diodi D1 e D2, che in condizioni normali sono interdetti con qualsiasi combinazione dei contatti 12 e 13 (con il primo a livello alto D1 conduce e D2 no, viceversa conduce D2 e D1 no, con entrambe le linee a zero i diodi sono tutti e due interdetti e con tutte e due a 1 conducono ma non c'è interferenza) mentre quando il pulsante chiude a massa conducono e portano i piedini 12 e 13 a circa 0,6 volt, tensione che vale come 0 logico. LD1 e LD2 si accendono in contemporanea quando si preme il tasto RESET1.

Prima di concludere la descrizione del trasmettitore va detto che nel modulo RTX è implementata una funzione di aumento progressivo della potenza in uscita voluta per aumentare l'autonomia se il circuito è alimentato a batteria: quando si impartisce un comando, ovvero si preme uno dei pulsanti dell'unità di comando, per impostazione predefinita il modulo trasmette alla minima potenza, che è pari a 0 dBm; se l'unità non riceve alcuna risposta ("ricevuta di ritorno") dalla ricevente, aumenta progressivamente la potenza in uscita in tre passi (arrivando fino a 100 mW) fino a quando non riesce a ricevere il segnale dall'unità RX.

UNITÀ ESECUTRICE

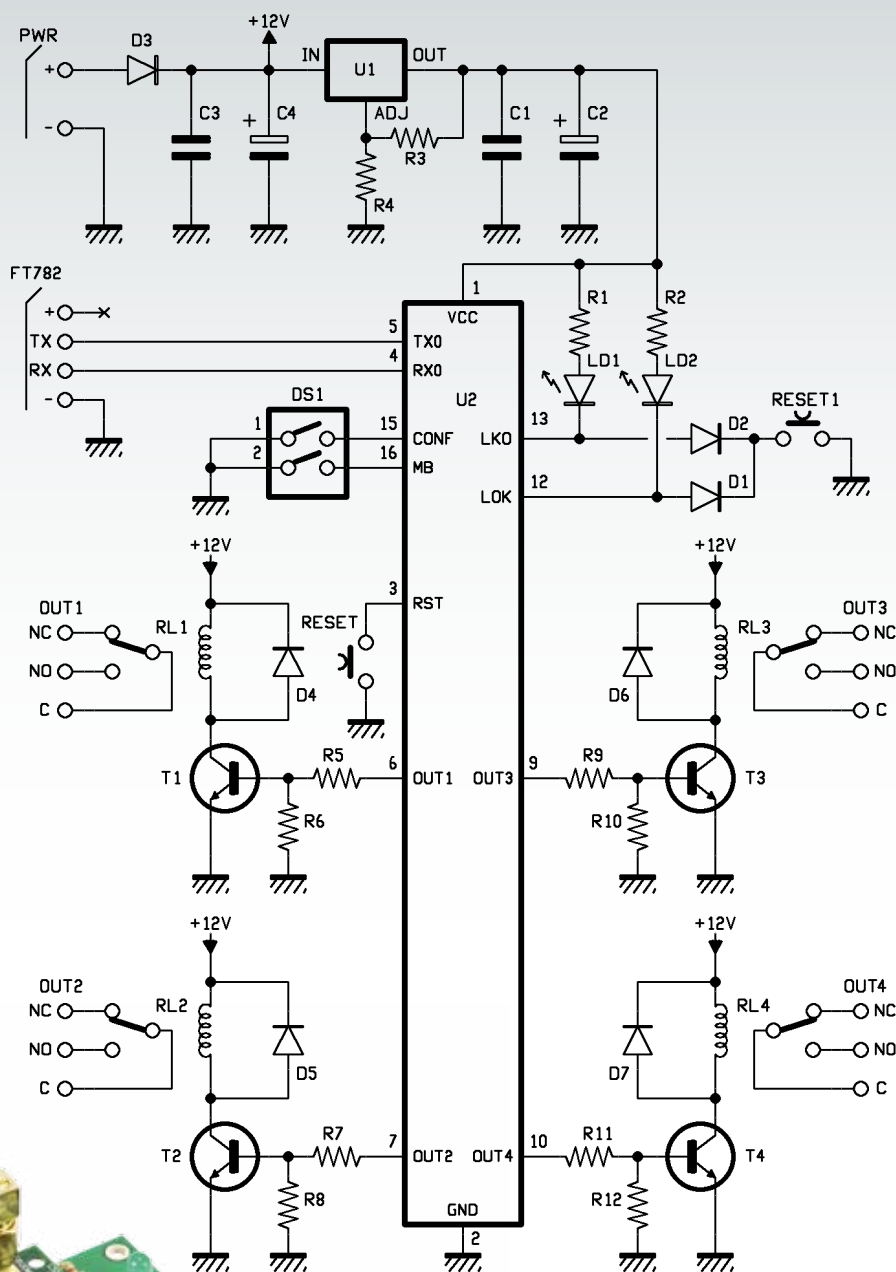
È la ricevente i comandi e si basa sul solito modulo, questa volta contornato da quattro relé mo-

CARATTERISTICHE TECNICHE

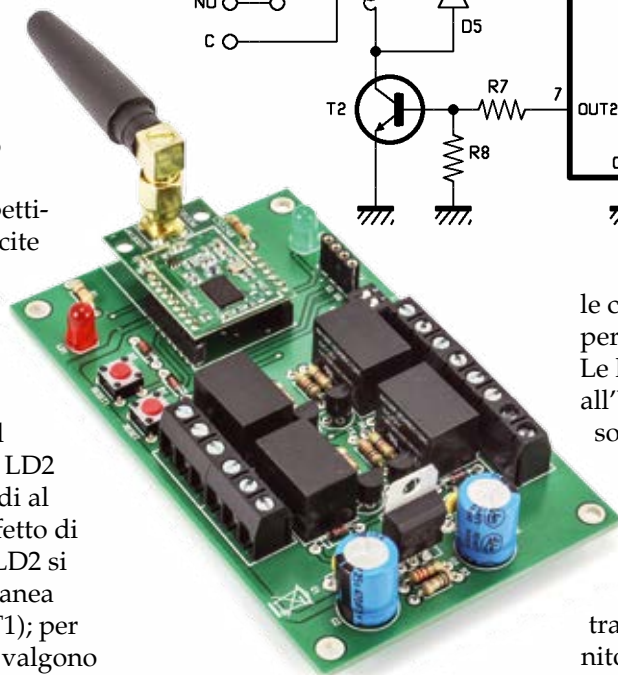
- Conferma comando inviato (ricevuta di ritorno) e possibilità di interrogare la ricevente solo per verificare lo stato dei 4 canali.
- Modulazione GFSK a 19,2 kbps
- Frequenza di lavoro di 869,5 MHz
- Aumento progressivo della potenza in uscita fino alla ricezione del segnale o fino alla massima potenza pari a 20dBm (100mW).
- Numero di trasmissioni da abbinare alla ricevente: 65.000
- Consumo in stand by < 0,1µA
- Consumo in TX mode: 10mA @0 dBm e 85mA @ 20dBm
- Portata fino a 500 metri in campo libero con trasmissione a 20 dBm.
- Sistema configurabile tramite interfaccia seriale della centralina RX

[schema elettrico **UNITÀ ESECUTRICE**]

Nello schema ritroviamo i soliti pulsanti RESET e RESET1 che servono rispettivamente a resettare le uscite dell'unità ricevente e a caricare il bootloader. Rivediamo la stessa configurazione dei contatti LKO e LOK, normalmente destinati al comando dei LED LD1 e LD2 ma collegati con due diodi al pulsante RESET1 (per effetto di tale connessione, LD1 e LD2 si accendono in contemporanea premendo il tasto RESET1); per questa sezione circuitale valgono



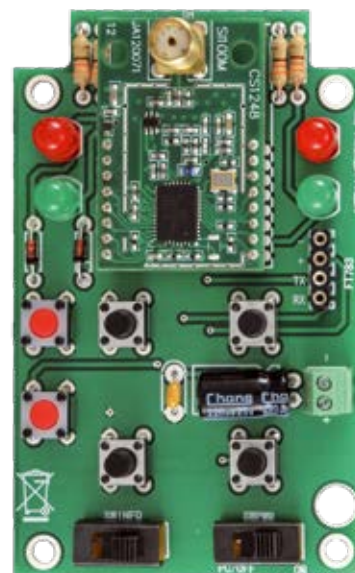
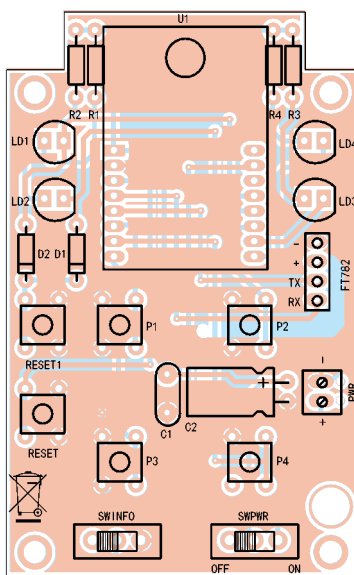
le considerazioni fatte poco fa per il circuito di comando. Le linee TX0 ed RX0, relative all'UART, anche in questa unità sono portate al connettore che permette di applicare un convertitore TTL/USB con cui il circuito può essere collegato e interfacciato al computer per la configurazione tramite l'apposito software fornito dal produttore. I dip-switch



[piano di montaggio **UNITÀ DI COMANDO**]

Elenco Componenti:

R1: 330 ohm	P4: Microswitch
R2: 330 ohm	RESET: Microswitch
R3: 330 ohm	RESET1: Microswitch
R4: 330 ohm	SWINFO: Deviatore slitta
C1: 100 nF ceramico	SWIPWR: Deviatore slitta
C2: 220 μ F 16 VL elettrolitico	
D1: 1N4148	Varie:
D2: 1N4148	- Morsettiera 2 poli passo 2,54mm
LD1: LED 5mm rosso	- Strip femmina 8 vie passo 2mm (2 pz.)
LD2: LED 5mm verde	- Strip femmina 4 vie passo 2,54mm
LD3: LED 5mm rosso	- Circuito stampato S1194
LD4: LED 5mm verde	
U1: RCS1K-868-TX	
P1: Microswitch	
P2: Microswitch	
P3: Microswitch	



connessi alle linee CONF (15) e MB (16) permettono di impostare le modalità di funzionamento del modulo programmato come ricevente e quelle delle uscite. Più esattamente, per DS1:

- ON attiva la configurazione dei parametri;
- OFF attiva il funzionamento normale.

La configurazione dei parametri (DS1 in ON) si effettua mediante la connessione seriale, ovvero da computer.

Quanto a DS2, definisce la modalità di funzionamento delle uscite a relé e vale per tutte e quattro:

- ON imposta la monostabile;
- OFF imposta la bistabile.

Ciascuna delle uscite può gestire carichi elettrici funzionanti a un massimo di 125Vca che assorbano 1 A, ovvero, in continua, fino a 60 V. Lo scambio è disponibile in toto, il che permette di controllare circuiti che richiedano l'apertura o la chiusura del contatto, ovvero la commutazione della corrente.

L'unità ricevente va alimentata (tra i morsetti + e - PWR) con

una tensione continua di valore compreso tra 11 e 14 V, meglio se stabilizzata; la fonte di alimentazione utilizzata deve poter fornire una corrente di almeno 200 mA.

La tensione applicata a PWR passa attraverso il diodo D3, inserito per proteggere il circuito dall'inversione di polarità, e -opportunamente filtrata dai condensatori C3 e C4- alimenta i circuiti delle bobine dei relé; raggiunge, inoltre, l'ingresso del regolatore lineare LM317T, un componente in TO-220 che permette di ricavare in uscita una tensione stabilizzata dipendente dal rapporto tra le resistenze R2/R3, che formano la rete di retroazione tra uscita (OUT) e ADJ. Nel nostro caso l'U1 fornisce 3,3 Vcc (ben filtrati dai condensatori C1 e C2), con i quali viene fatto funzionare il modulo RTX (LED compresi).

REALIZZAZIONE PRATICA

Descritti i circuiti si può vedere come costruirli: partiamo dai circuiti stampati, che sono uno per ciascuna unità e si ottengono per fotoincisione partendo dalle tracce lato saldature e lato com-

ponenti (le basette sono entrambe a doppia faccia) scaricabili insieme agli altri file del progetto dal nostro sito www.elettronica.in.it. Incise e forate le piastre, bisogna innanzitutto realizzare le vie di connessione tra le piazzole libere delle due facce (i fori che ospitano i reofori dei componenti fanno già da interconnessioni) usando degli spezzoni di filo in rame sottile (0,5 mm), ovvero avanzi di terminali di resistenze da 1/4 di watt. Tale operazione non è richiesta se si acquistano i c.s. già pronti, giacché questi sono a fori metallizzati.

Per l'unità trasmittente occorre montare per prime le resistenze e gli strip femmina a passo 2,54 mm da 8 poli ciascuno (in essi si inserirà il modulo RCS1K-868) e da 4 poli per la connessione al convertitore TTL/USB; poi è la volta dei pulsanti (quelli di reset sono più bassi, in modo che inserendo la scheda in un contenitore non debbano essere premuti inavvertitamente) e dei deviatori, dei LED e della morsettiera per l'alimentazione. Il condensatore elettrolitico va montato in orizzontale. Una volta posizionato

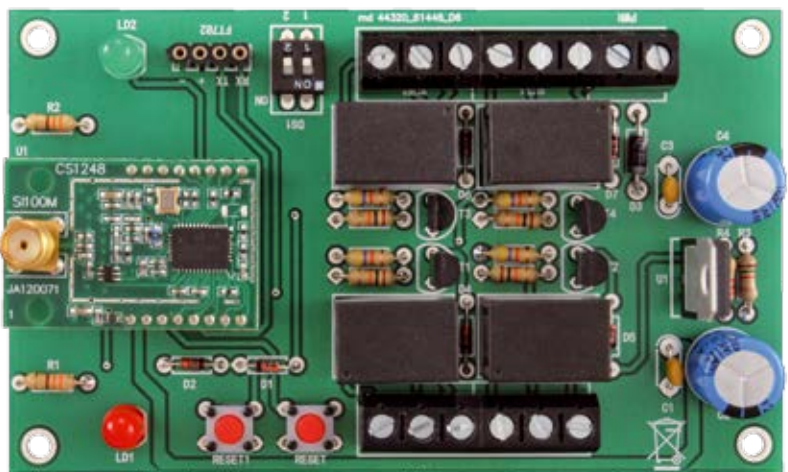
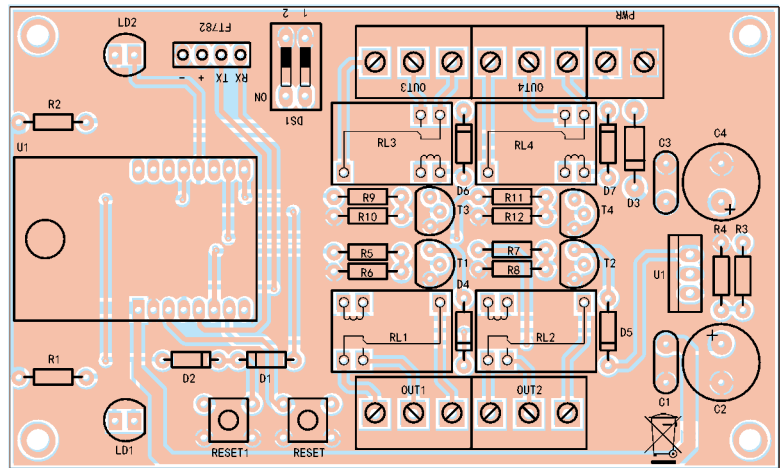
[piano di montaggio UNITÀ ESECUTRICE]

il modulo RTX, sul suo connettore SMA va applicata l'antenna accordata a 868 MHz.

Quanto all'unità ricevente, il montaggio inizia con le resistenze e i diodi al silicio, proseguendo poi con gli strip femmina per il modulo radio e per la connessione al convertitore TTL/USB (vale quanto detto per quelli dell'unità trasmittente), con il dip-switch e i due pulsanti; è poi la volta dei condensatori non polarizzati, dei transistor, dei LED e delle morsettiere. Il montaggio si conclude con gli elettrolitici (da montare, stavolta, in piedi), dei quattro relé e del regolatore integrato LM317T. Completate le saldature, si innesta il modulo RTX e lo si completa dell'antenna. Per tutte le fasi del montaggio seguite i disegni di disposizione dei componenti pubblicati in queste pagine, che mostrano l'orientamento degli elementi.

CONFIGURAZIONE DEL SISTEMA

L'impostazione predefinita prevede il funzionamento anzidetto, ovvero, nel modo normale, ogni pulsante della trasmittente determina una variazione nello stato del relé corrispondente sulla ricevente, variazione che dura il tempo impostato nel caso della modalità monostabile e che invece è permanente nel modo bistabile. È possibile modificare le impostazioni e anche rifare l'abbinamento fra trasmittente e ricevente entrando in programmazione, ovvero usando l'apposito software. La programmazione si effettua sull'unità ricevente, impostato il DS1 su ON, quindi, dopo aver collegato la scheda a un PC tramite l'adattatore TTL/USB e installato il software, avviando quest'ultimo. A proposito dell'adattatore, ricordiamo che l'RTX deve comunicare con livelli 0/3,3V, quindi prima di alimen-



Elenco Componenti:

R1: 330 ohm	elettrolitico	U2: RCS1K-868-RX
R2: 330 ohm	D1: 1N4148	RESET: Microswitch
R3: 1,5 kohm	D2: 1N4148	RESET1: Microswitch
R4: 2,2 kohm	D3: 1N4007	RL1: Relé 12V 1 scambio
R5: 4,7 kohm	D4: 1N4148	RL2: Relé 12V 1 scambio
R6: 10 kohm	D5: 1N4148	RL3: Relé 12V 1 scambio
R7: 4,7 kohm	D6: 1N4148	RL4: Relé 12V 1 scambio
R8: 10 kohm	D7: 1N4148	
R9: 4,7 kohm	D8: 1N4148	Varie:
R10: 10 kohm	DS1: Dip-Switch 2 vie	- Morsettieria 2 poli
R11: 4,7 kohm	T1: BC547	- Morsettieria 3 poli
R12: 10 kohm	T2: BC547	(4 pz.)
C1: 100 nF ceramico	T3: BC547	- Strip femmina 8 vie
C2: 470 µF 25 VL	T4: BC547	passo 2mm (2 pz.)
elettrolitico	LD1: LED 5mm rosso	- Strip femmina 4 vie
C3: 100 nF ceramico	LD2: LED 5mm verde	passo 2,54mm
C4: 470 µF 25 VL	U1: LM317	- Circuito stampato S1195

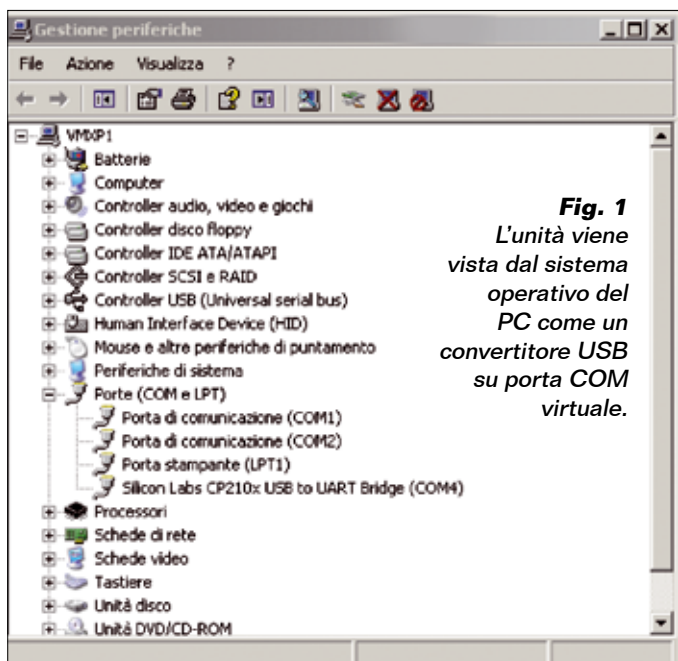


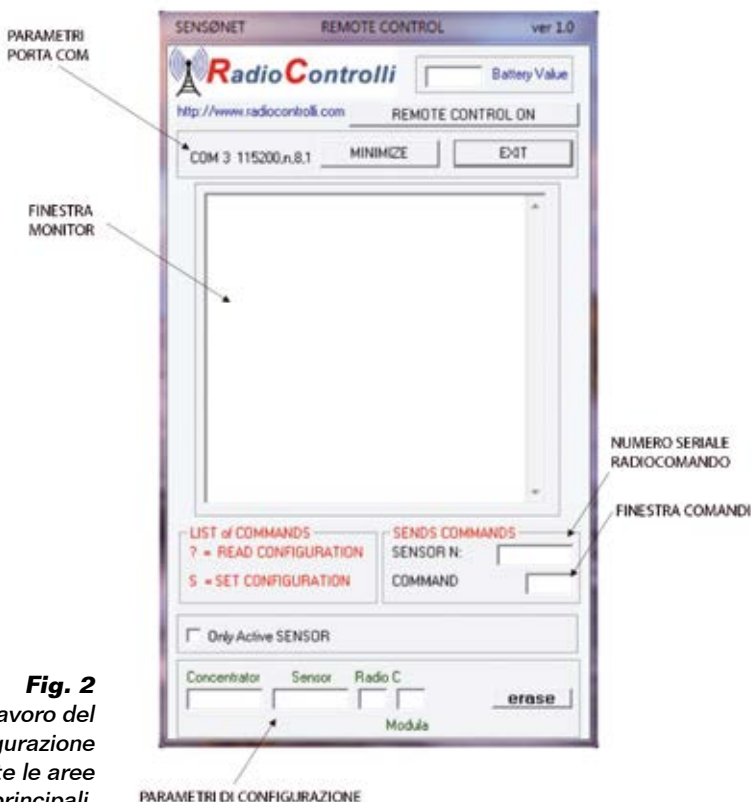
Fig. 1
L'unità viene vista dal sistema operativo del PC come un convertitore USB su porta COM virtuale.

tarlo occorre impostare il relativo jumper su 3,3V.

Partiamo dall'installazione del software (SENSONET RADIO-COMANDO), che viene fornito come archivio compresso (si scarica dal nostro sito www.elettronica-in.it). Lanciando l'applicazione *setup.exe*, appare la finestra di dialogo che ricorda di chiudere tutte le applicazioni aperte nel sistema operativo (altrimenti non sarà possibile aggiornare eventuali file condivisi) e quindi fare clic sul pulsante OK. Si passa così alla finestra di dialogo dalla quale bisogna scegliere la directory di installazione e fare clic in alto a sinistra. Completata l'installazione si può collegare l'unità ricevente al computer tramite una porta USB libera e l'apposito adattatore; notate che la scheda deve essere alimentata dal proprio alimentatore, giacché l'adattatore TTL/USB collega solo i canali dati e la massa, che deve essere comune.

Inserito il connettore del cavo USB nel computer, deve avviarsi il riconoscimento (diamo per scontato che stiate utilizzando Windows, almeno dalla versione XP) e deve essere assegnata

Fig. 2
Finestra di lavoro del software di configurazione con descritte le aree principali.



alla periferica una porta seriale virtuale (**Fig. 1**). Nel caso l'installazione fallisse e fosse necessario individuare manualmente il driver della scheda adattatrice (FTDI), è possibile scaricarlo da www.ftdichip.com, ovvero su www.futurashop.it.

Completata l'installazione, si avvia il programma facendo clic sull'icona che appare nel nuovo gruppo SENSONET creato nel menu Programmi: poco dopo comparirà la finestra di dialogo mostrata in **Fig. 2**, nella quale bisognerà fare clic sul pulsante COM?? e apparirà una finestra nella quale bisognerà scegliere, facendo clic sul pulsante corrispondente, la porta COM cui il dispositivo è collegato; per confermare la scelta occorrerà fare clic sul pulsantino X di chiusura della finestra. La schermata di **Fig. 2** mostrerà ora in centro una finestra di dialogo come quella visibile in **Fig. 3**, nella quale bisogna fare clic sul pulsante OK.

Fatto ciò, si può prendere l'unità trasmittente, equipaggiata della propria antenna, alimentarla e premere i quattro pulsanti uno alla volta; nella finestra di monitor della schermata vedrete apparire i dati ricevuti, ognuno dei quali occupa una riga (**Fig. 4**). Nel caso d'esempio, notate che 7E7E7E7E è il numero di serie assegnato in fabbrica ad ogni radiocomando, mentre 8,1,2,4 identifica il numero di tasto premuto. Nel caso proposto dalla **Fig. 4**, 03FF è il valore esadecimale relativo alla posizione del potenziometro nella modalità PWM dell'RCS1K-868, che in questa applicazione non è utilizzata. È possibile modificare il numero di serie procedendo dalla sezione SENDS COMMANDS, come verrà spiegato tra breve. Il modulo RTX riferito all'unità ricevente, ha un address (ID) specifico di 8 byte definito in fabbrica, che può essere ad esempio 1E010001 (per impostazione pre-

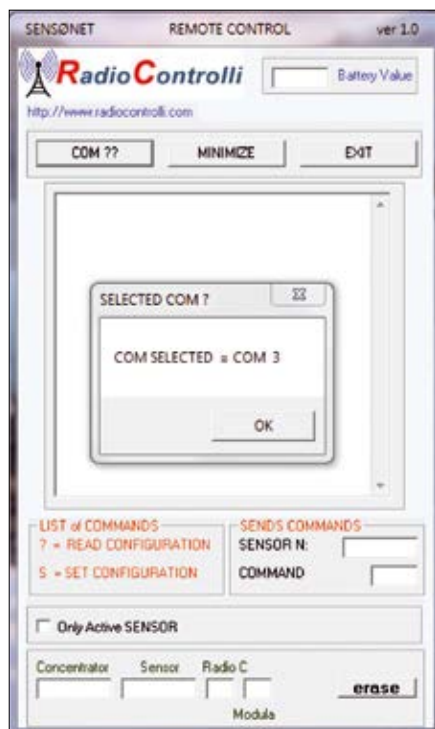


Fig. 3

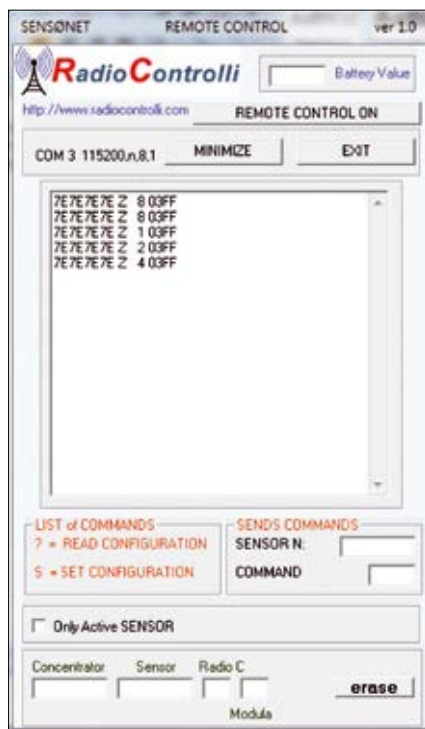


Fig. 4 - Finestra di monitor con in evidenza i dati ricevuti dall'unità di comando.

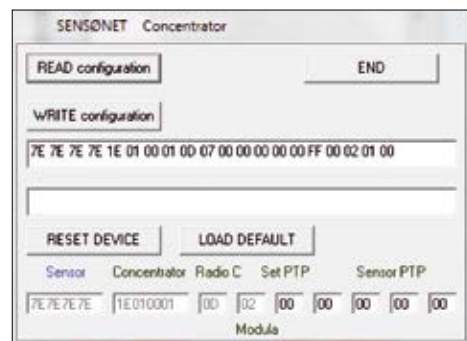


Fig. 5



Fig. 6

definita il parametro di broadcast risulta settato a 00). Il modulo RTX dell'unità trasmettente viene invece configurato in fabbrica con questi parametri:

Serial Number ID = 7E7E7E7E
Gateway Reference = 7E7E7E7E

In queste condizioni la ricevente con ID=1E010001 riconosce l'unità di comando con Factory ID=7E7E7E7E. Se non si conoscono i parametri di configurazione del modulo è possibile (sempre con il DS1 della ricevente chiuso) vederli nella finestra di dialogo illustrata in **Fig. 5**, dove appariranno facendo clic sul pulsante READ configuration che si trova in alto a sinistra. Nel caso mostrato, il numero di serie del modulo RTX della ricevente è "1E010001"; questo numero è definito dalla fabbrica e non è modificabile, come non sono modificabili i parametri Radio (Frequenza e Modulazione). È

invece modificabile il parametro "Set PTP" che può assumere i seguenti valori:

- 00 = funzionalità broadcasting; la centralina RX è abilitata a ricevere i dati di tutte le trasmettenti ad essa abbinate ed anche i dati delle trasmettenti con numero di serie 7E7E7E7E (predefinito);
- 01 = broadcasting la centralina RX colloquia solo con le trasmettenti ad essa abbinate;
- 02 = viene attivata la funzione Peer To Peer verso il numero di trasmettente definito nelle caselle a destra (Sensor PTP).

Per abbandonare la configurazione si deve fare clic sul pulsante END e aprire il dip-switch DS1.

MODIFICA ID TRASMETTENTE E ABBINAMENTO ALLA RICEVENTE

Tutti i moduli transceiver RCS1K-868 programmati per funzionare nell'unità di comando vengono configurati con:

Serial Number ID=7E7E7E7E
Gateway Reference=7E7E7E7E
Per evitare che, trovandosi due sistemi di radiocomando ad operare nello stesso raggio d'azione l'uno interferisca con l'altro, è necessario modificare il serial number con un numero desiderato ed è necessario modificare il Gateway Reference indicando il numero di serie dell'RTX della ricevente. Per spiegarvi come procedere, supponiamo di modificare i parametri di fabbrica in: Serial Number ID = 00000001



Fig. 7

Gateway Reference = 1E010001
Allo scopo, bisogna alimentare la trasmittente interessata e poi, con la ricevente impostata in configurazione (DS1 deve essere chiuso) si deve, nella casella SENSOR N: della finestra di lavoro del programma, inserire il codice 7E7E7E7E e poi scrivere nella casella COMMAND il comando "?". La situazione è quella illustrata in Fig. 6.

Fatto questo, sull'unità trasmittente bisogna portare il piedino RX del modulo U1 a massa; ciò si ottiene facilmente ponticellando con uno spezzone di filo in rame il contatto 3 del connettore siglato FT782 con quello di massa. Nella finestra di dialogo di Fig. 6 nel giro di qualche istante apparirà la situazione di Fig. 7.

Adesso bisogna assegnare un numero di serie alla trasmittente, per esempio 00000001, ed inserire il numero di serie della ricevente precedentemente annotato (1E010001) quindi inviare il comando "S" dalla finestra di dialogo, scrivendo la lettera S

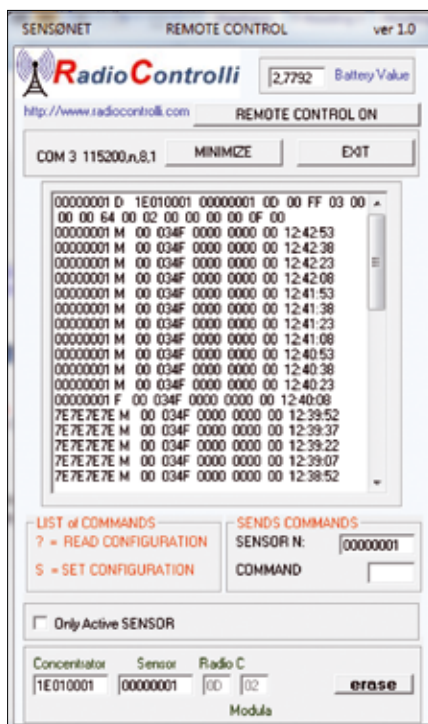


Fig. 8

maiuscola e premendo il tasto INVIO. Per accertare l'avvenuta variazione bisogna utilizzare il comando erase (azzeramento visione parametri di configurazione) inserire in SENSOR N. 00000001 ed in COMMAND "?" e verificare che i parametri siano stati modificati; ne risulterà una finestra di dialogo tipo quella di Fig. 8. Per uscire dalla modalità di abbinamento bisogna aprire il ponticello, sulla trasmittente, tra TX e GND. Ricordiamo che a ciascuna ricevente possono essere abbinate fino a 65.000 trasmissioni.

VERIFICA DEI TX ABBINATI

In ogni momento è possibile sapere quali trasmissioni sono abbinate a una ricevente: dopo aver chiuso il DS1 su quest'ultima ed aver visualizzato la stessa finestra di dialogo di Fig. 5 (ma vuota), bisogna fare clic in questa sul pulsante READ configuration. In risposta, si otterrà la finestra compilata come mostrato, ad esempio, in Fig. 9. Per fare in

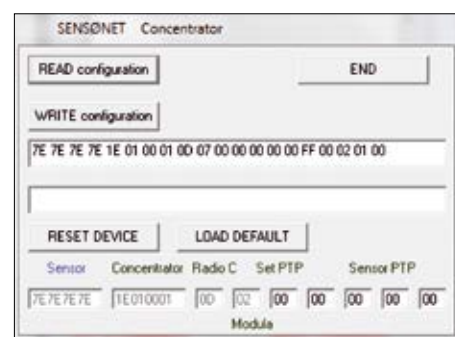
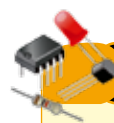


Fig. 9

modo che la ricevente risponda ai soli comandi provenienti dalle unità trasmissioni abbinate, occorre inserire nel campo denominato Set PTP il codice "01". A questo punto occorre fare clic sul pulsante WRITE configuration e salvare la configurazione; da questo momento la ricevente accetterà soltanto i comandi provenienti dalle trasmissioni abbinate.

Nel caso occorresse aggiungere altre trasmissioni, è necessario settare il parametro PTP a "00"; verrà riconosciuta la trasmittente 7E7E7E7E, che poi dovrà essere indirizzata alla centralina RX (ID univoco). Sarà poi necessario reimpostare PTP a "01".



per il MATERIALE

Le schede vengono fornite in scatola di montaggio e comprendono il modulo radio e l'antenna 868 MHz. Il telecomando (cod. FT1194K) è disponibile ad Euro 42,00 e la scheda esecutrice (cod. FT1195K) ad Euro 48,00. I singoli moduli sono disponibili anche separatamente; il modulo configurato come ricevitore (cod. RCS1K868-RX) viene venduto ad Euro 28,50, mentre il modulo configurato come trasmettitore (cod. RCS1K868-TX) ad Euro 28,50.

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

GESTIC: PRIMA APPLICAZIONE

di MATTEO DESTRO

Sperimentiamo con la tecnologia di Microchip che permette di riconoscere i gesti contactless, mediante una demoboard da interfacciare al PC e vediamo come realizzare l'interfacciamento con un microcontrollore PIC. Seconda puntata.

Nella precedente puntata abbiamo analizzato in modo approfondito la tecnologia Microchip che permette il riconoscimento delle Gesture attraverso la perturbazione di un campo elettrico appositamente creato ed abbiamo introdotto e descritto il relativo software di parametrizzazione da PC. In queste pagine presenteremo la scheda di valutazione che permette di lavorare con l'integrato MGC3130, nonché la scheda bridge per l'interfacciamento dell'integrato MGC3130 verso un PC allo scopo di compiere la parametrizzazione. Spiegheremo in dettaglio come interfacciarsi all'MGC3130 sfruttando un microcontrollore Microchip, dando così degli spunti per sviluppare le vostre applicazioni. L'integrato MGC3130 si presenta in case QFN da 28 pad, cinque dei quali sono riservati al collegamen-

to degli elettrodi di ricezione (RX) ed uno è per collegare l'elettrodo di trasmissione (TX). I contatti di ricezione sono configurabili via software, ovvero è possibile decidere a quale elettrodo assegnare il polo cardinale Nord piuttosto che il Sud ecc. Questo serve per agevolare il routing delle piste di collegamento degli elettrodi. Le interfacce di comunicazione messe a disposizione dall'integrato sono di tipo seriale, in particolare si ha la possibilità di scegliere tra una comunicazione I²C piuttosto che la SPI. Le interfacce sono riferibili ai pin SI₀, SI₁, SI₂ e SI₃. L'interfaccia I²C è mappata sui pin SI₀ e SI₁; in dettaglio, la linea SDA è connessa al pin SI₀ mentre la linea SCL al pin SI₁. Diversamente è per l'interfaccia SPI: il pin SI₀ identifica la linea MISO (Master Input Slave Output), il pin SI₁ la linea MOSI (Master Output Slave

MGC3130: breve panoramica

L'integrato MGC3130 è un sistema di riconoscimento tridimensionale delle gesture basato sulla tecnologia Gestic. Alcune delle possibili applicazioni sono:

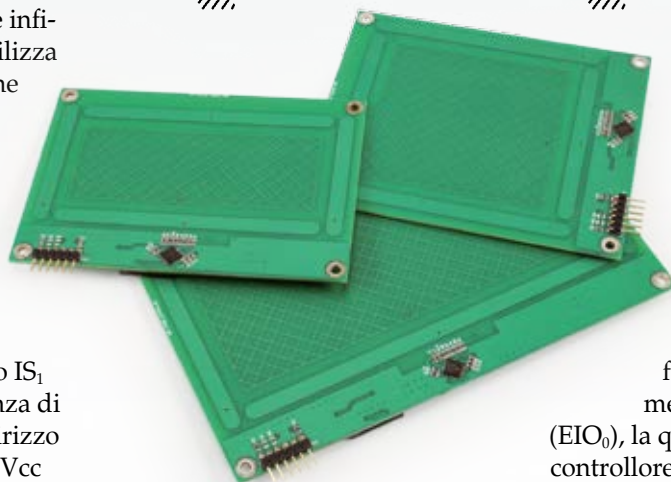
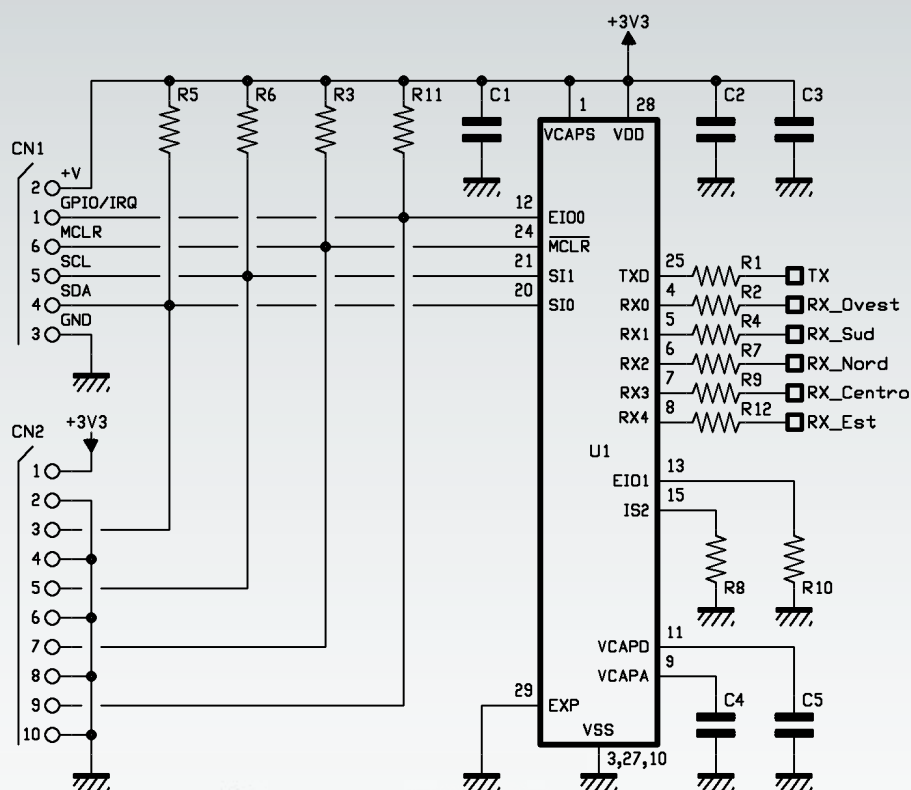
- smartphone;
- tastiere;
- notebook/tablet;
- sistemi di controllo remoto;
- sistemi di controllo per videogiochi.

Le caratteristiche chiave dell'integrato possono essere così riassunte:

- riconoscimento dei movimenti della mano/dita in modo tridimensionale;
- rilevamento della posizione rispetto agli assi X, Y e Z;
- rilevamento di prossimità e touch;
- distanza di riconoscimento delle gesture da 0 a 15 cm;
- capacità di rilevare variazioni di capacità degli elettrodi anche inferiori al femtofarad;
- risoluzione spaziale superiore ai 150 dpi;
- auto-calibrazione;
- libreria Colibri e firmware integrato aggiornabile.

Input), il pin SI₂ la linea CS e infine SI₃ la linea SCLK. Se si utilizza l'interfaccia di comunicazione I²C, si devono configurare i pin EIO₁(IS₁) e IS₂ per assegnare l'indirizzo fisico alla periferica. Si possono assegnare solo due indirizzi fisici (0x42 o 0x43) collegando verso massa o verso Vcc i due pin di cui sopra. In dettaglio, mettendo IS₁ e IS₂ a massa tramite resistenza di pull-down si seleziona l'indirizzo 0x42; se invece si porta IS₂ a Vcc

[schema elettrico - SEZIONE ELETTRODI]



tramite resistenza di pull-up, lasciando IS₁ collegato verso massa, si seleziona l'indirizzo 0x43.

Per concludere la descrizione dell'interfaccia di comunicazione menzioniamo la linea TS (EIO₀), la quale indica al microcontrollore quando sono presenti dei dati validi da leggere dopo il riconoscimento di una gestura. La linea TS viene comandata sia dall'integrato MGC3130 che dal microcontrollore, secondo un preciso schema di funzionamento descritto nella **Tabella 1**.

Tabella 1

MGC3130 TS	PIC TS	Stato
HIGH	HIGH	Non ci sono dati da leggere. MGC3130, se necessario, può aggiornare il buffer dei dati in uscita; il microcontrollore non sta impegnando la linea.
LOW	HIGH	Sono disponibili dei dati da leggere nel buffer di uscita di MGC3130. Questa condizione indica al microcontrollore che può iniziare una sessione di lettura.
LOW	LOW	Il microcontrollore inizia la lettura del buffer di uscita dell'integrato MGC3130. In questa condizione l'MGC3130 non può aggiornare il suo buffer di uscita fintantoché il microcontrollore non rilascerà la linea TS.
HIGH	LOW	Il microcontrollore non ha ancora finito di leggere i dati caricati precedentemente nel buffer, quindi l'MGC3130 non potrà trasferire i dati nel buffer di uscita fintantoché il microcontrollore non porterà a livello alto la linea indicando di avere concluso la precedente lettura.

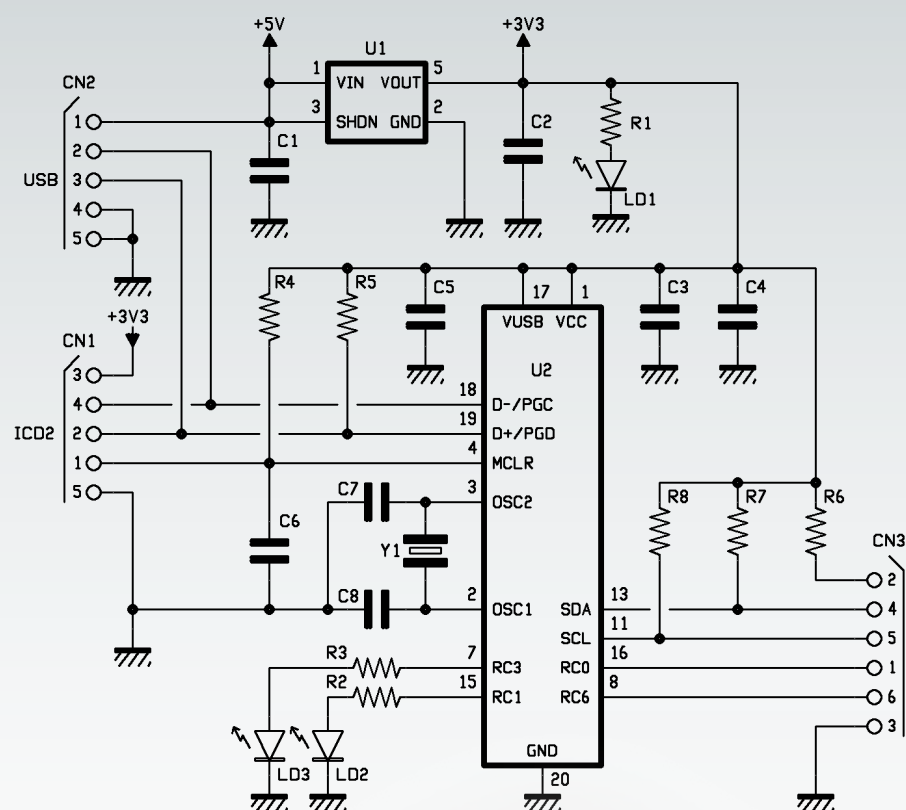
SCHEMA ELETTRICO ELETTRODO

Cominciamo con la descrizione degli schemi elettrici partendo dalle schede elettrodo, le quali si distinguono le une dalle altre per dimensioni dell'area elettrodi:

[schema elettrico - **SCHEDA BRIDGE**]

sono una da 80x80 mm, una da 100x50 mm e una da 140x90 mm. Dato che i componenti montati su tutte e tre le schede sono identici, descriveremo solo una sola delle tre e in particolare quella che ha l'area elettrodi di 150x50 mm. Partiamo col dire che ciascuna scheda è basata su un integrato MGC3130; la scelta di montarlo a bordo nasce dal fatto che il chip deve trovarsi il più vicino possibile agli elettrodi per minimizzare l'effetto capacitivo delle piste che ne collegano le linee ad essi. Per ognuno degli elettrodi di ricezione (R_{X0} , R_{X1} , R_{X2} , R_{X3} e R_{X4}) è stata prevista una resistenza da 10 kohm, la quale, associata alla capacità presente su ogni ingresso, realizza un filtro passa basso necessario ad eliminare i disturbi ad alta frequenza. Il filtro così realizzato ha una frequenza di taglio di 3 MHz. L'elettrodo di trasmissione TX non vuole alcuna resistenza e quella che vedete è in realtà un ponticello (0 ohm).

I pin SI_0 e SI_1 identificano rispettivamente le linee SDA e SCL del bus I²C, per ognuna delle quali è stata predisposta una resistenza di pull-up da 1K8; queste resistenze possono non essere montate se sono già state previste sulla scheda di interfaccia da collegare alla scheda elettrodo. Nel nostro caso



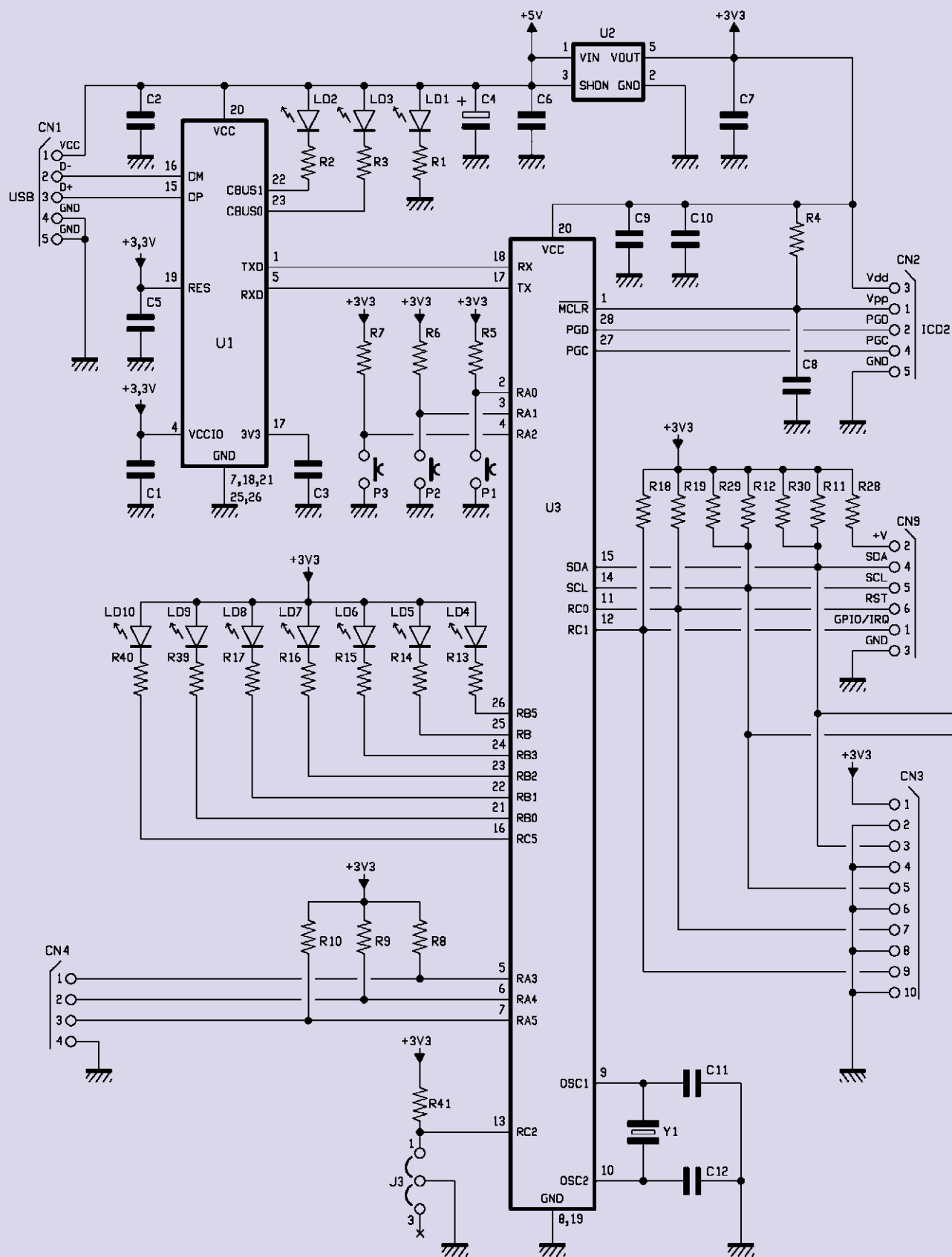
ne facciamo a meno in quanto sia la nostra scheda di interfaccia sia la scheda bridge I²C to USB prevedono tali resistenze di pull-up. L'indirizzo assegnato alla periferica I²C è 0x42, quindi i pin IS_2 e EIO_1 (IS_1) devono essere connessi a massa tramite una resistenza di pull-down. La linea MCLR, utile

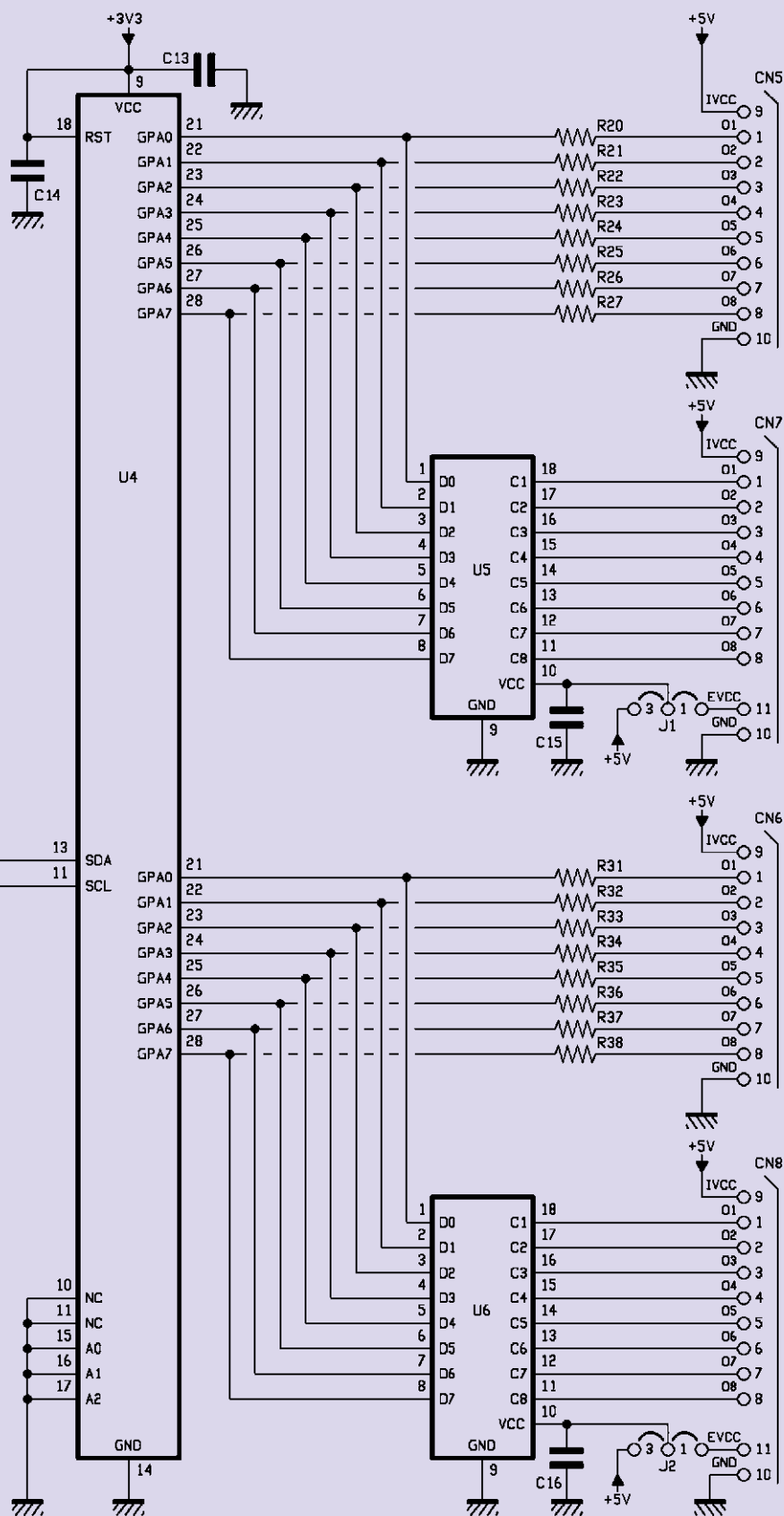


Tabella 2

P1	P2	P3	Tempo selezionato
ON	OFF	OFF	Tempo monostabile 1 secondo. Premere pulsante P1 per più di due secondi. LED associato LED 4.
OFF	ON	OFF	Tempo monostabile 5 secondi. Premere pulsante P2 per più di due secondi. LED associato LED 5.
OFF	OFF	ON	Tempo monostabile 10 secondi. Premere pulsante P3 per più di due secondi. LED associato LED 6.
ON	ON	OFF	Tempo monostabile 15 secondi. Premere pulsanti P1 & P2 per più di due secondi. LED associato LED 7.
OFF	ON	ON	Tempo monostabile 20 secondi. Premere pulsanti P2 & P3 per più di due secondi. LED associato LED 8.
ON	OFF	ON	Tempo monostabile 25 secondi. Premere pulsanti P1 & P3 per più di due secondi. LED associato LED 9.
ON	ON	ON	Tempo monostabile 30 secondi. Premere pulsanti P1 & P2 & P3 per più di due secondi. LED associato LED 10.

per resettare l'integrato MGC3130, necessita di una resistenza di pull-up; idem la linea GPIO/IRQ (EIO_0), anche detta linea TS. Tutte le resistenze di pull-up e pull-down sono da 10 kohm, tranne quelle previste per le linee SDA e SCL. La linea TS, insieme alla MCLR, completa la comunicazione verso il microcontrollore; la linea TS è assolutamente necessaria per la gestione della comunicazione tra il microcontrollore e l'integrato, mentre la MCLR potrebbe





essere trascurata forzandola sulla scheda elettrodo a +Vcc. Tuttavia consigliamo vivamente di collegare anch'essa al microcontrollore, in modo da averne il pieno controllo. L'alimentazione dell'integrato è a +3,3 Vcc e prevede i soliti condensatori di filtro, in aggiunta a due condensatori da 4,7 µF sui pin Vcapa e Vcapd (Pin 9 e 11).

Per l'interconnessione con la scheda demo abbiamo previsto il connettore CN2: questo è di tipo FPC da dieci pin con passo 1 mm. Il cavo di collegamento da abbinare ha un connettore WR-FFC, sempre da dieci poli con passo 1mm, lungo 50 mm. Volendo, esistono cavi più lunghi: da 100 mm e oltre. Il secondo connettore presente sulla scheda è un normalissimo BERG pensato per la connessione della scheda di interfaccia I²C to USB di cui parleremo a breve.

SCHEMA ELETTRICO BRIDGE

Analizziamo ora la scheda bridge I²C to USB che provvede all'interfacciamento delle schede elettrodo con il PC. Sulla scheda bridge sono presenti i componenti essenziali, tra cui il microcontrollore PIC18F14K50 programmato con il firmware fornito dalla Microchip. Il microcontrollore ha a bordo un'interfaccia USB nativa e quindi tramite i pin D+ e D- (Pin 19 e 18) può essere direttamente collegato alla USB del PC. Per la linea D+ è necessaria una resistenza di pull-up da 150 kohm. Il firmware memorizzato nel PIC ha una sezione dedicata alla gestione dell'interfaccia USB, che avviene sfruttando le librerie Microchip. Dal connettore USB CN2, oltre ai segnali D+ e D- si preleva la tensione di alimentazione +5Vdc la quale attraverso l'integrato U1 viene ridotta a +3,3 Vcc. U1 è un regolatore di tensione LDO Microchip MCP1801T-3302I/OT, in grado di fornire fino a un massimo di 150 mA di corrente. È

Tabella 3

Uscita	Gesture
01	Tap su elettrodo Sud
02	Tap su elettrodo Ovest
03	Tap su elettrodo Nord
04	Tap su elettrodo Est
05	Tap elettrodo Centrale
06	Doppio Tap su elettrodo Sud
07	Doppio Tap su elettrodo Ovest
08	Doppio Tap su elettrodo Nord
09	Doppio Tap su elettrodo Est
010	Doppio Tap elettrodo Centrale
011	Gesture movimento mano da Ovest verso Est
012	Gesture movimento mano da Est verso Ovest
013	Gesture movimento mano da Sud verso Nord
014	Gesture movimento mano da Nord verso Sud
015	Gesture movimento circolare orario
016	Gesture movimento circolare antiorario

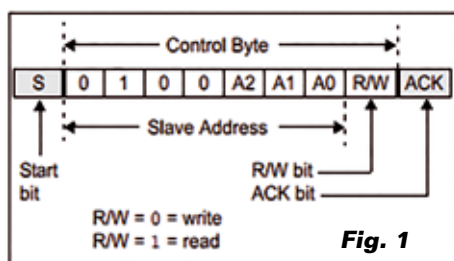


Fig. 1

realizzato in case SOT-23-5 molto compatto e adatto alla nostra applicazione. Come si evince dallo schema elettrico, oltre ai pin di ingresso e uscita (Vin e Vout) si deve collegare a +5 Vcc anche il pin SHDN per mantenere sempre attiva l'uscita dell'integrato. Il pin SHDN, se è posto fisso a livello logico alto, mantiene attiva la sezione di uscita e quindi l'elettronica a valle viene alimentata; al contrario, se è mantenuto a livello logico basso la sezione di uscita è disabilitata e l'integrato entra nello stato di "low quiescent current shutdown" assorbendo una corrente tipica di soli 0,01 μ A. Tutta l'elettronica della scheda bridge e quindi tutte le elettroniche degli elettrodi ad essa collegati saranno alimentati a +3,3 V come da specifica. I pin D+ (PGD) e D- (PGC) del microcontrollore sono collegati anche al connettore per la pro-

grammazione in-circuit CN1. Oltre a tali pin, si deve portare anche il segnale MCLR (pin 4 del microcontrollore) che è fondamentale per la programmazione in-circuit. Al connettore CN3 ritroviamo i segnali da portare alla scheda elettrodo per l'interfacciamento all'integrato MGC3130: in particolare, notiamo i segnali SDA_MGC3130 e SCL_MGC3130, per il bus I²C, nonché il segnale MCLR_MGC3130 per resettare l'integrato in caso di necessità. Completa la serie di connessioni la linea EIO0 (Linea TS). Per le linee SDA e SCL sono state previste le resistenze di pull-up, che su questa scheda sono montate diversamente dalle schede elettrodo come detto prima. L'alimentazione viene portata al pin2 del connettore tramite la resistenza R6, che in realtà è un ponticello. Le linee SDA e SCL sono collegate direttamente alla periferica I²C del PIC18F14K50. Non rimane che collegare un paio di LED di diagnostica per evidenziare la presenza della connessione USB tra il PC e la scheda (LD2), nonché l'attività presente sul bus I²C (LD3). Il diodo LD1 (rosso) segnala quando la scheda è alimentata. Sulla linea di alimentazione del microcontrollore abbiamo previsto i soliti condensatori di filtro.

Il quarzo scelto per il PIC è da 12 MHz; questa scelta nasce dal fatto che l'interfaccia USB integrata nel PIC deve lavorare a 48 MHz e che il suo clock è pari a quello principale moltiplicato per 4. I condensatori C7 e C8 completano la configurazione del generatore di clock a quarzo.

SCHEMA ELETTRICO DELLA DEMOBOARD

Passiamo ora a descrivere la nostra scheda demo, la quale mette a disposizione una serie di funzionalità per provare e testare le po-

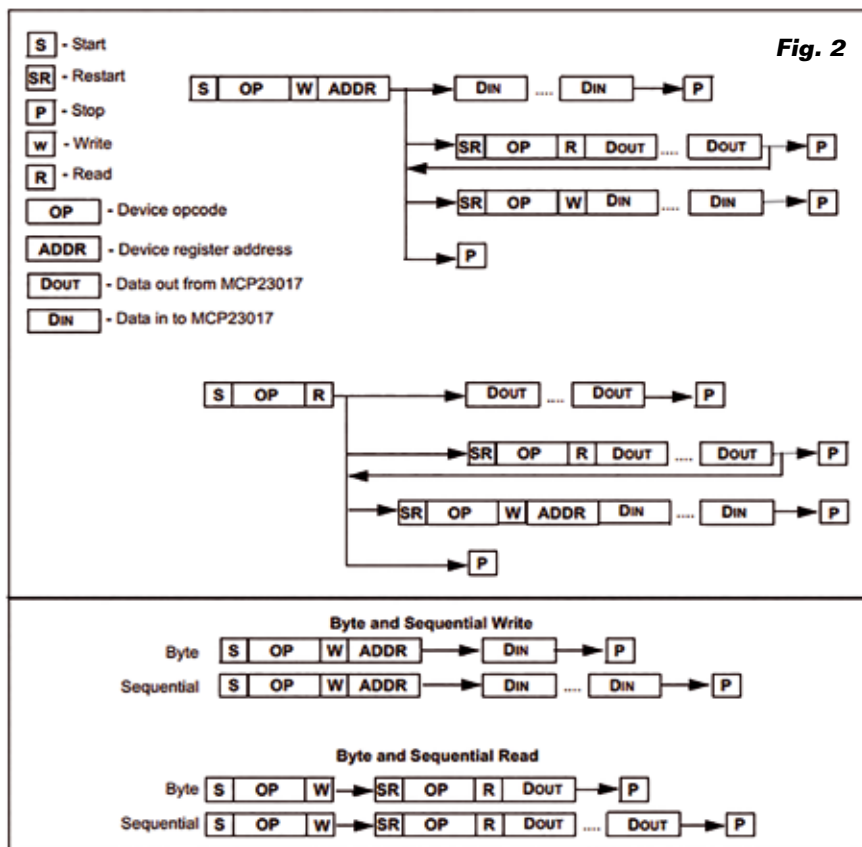


Fig. 2

tenzialità dell'integrato MGC3130. Oltre a interagire con l'integrato MGC3130 la scheda permette di trasferire le gesture riconosciute sulle sue sedici uscite digitali per la gestione di attuatori esterni alla scheda; inoltre mette a disposizione un'interfaccia USB verso il PC per l'interazione con un nostro software di supporto allo sviluppo del progetto Gestic.

Il microcontrollore utilizzato è un PIC18F2620 il quale mette a disposizione uno spazio programma in Flash di 32.768 word (in altre parole 32.768 istruzioni), 3.986 byte di SRAM e ben 1.024 byte di EEPROM.

La frequenza massima operativa interna con PLL attivo è di 40 MHz sfruttando, come nel nostro caso, il quarzo esterno da 10 MHz. Ricordiamo che il PLL interno moltiplica per 4 la frequenza di oscillazione del quarzo.

La porta A del PIC viene utilizzata per il collegamento di tre pulsanti generici più tre uscite di diagnostica utilizzate per il debug del FW. Solitamente sono linee di trigger da monitorare con l'ausilio di un oscilloscopio; tale linee sono collegate al connettore CN4. La porta B viene interamente utilizzata, tranne le linee PGD e PGC, per la gestione dei LED che andremo a descrivere in seguito. Il settimo LED è collegato alla porta C (RC5). Per avere un collegamento verso il PC, si sfrutta la comunicazione seriale messa a disposizione dal PIC18F2620 e mappata sui pin

RC6 (TX) e RC7 (RX). La periferica viene fatta funzionare alla velocità di 115.200 baud, con 8 bit di dati e 1 bit di stop (configurazione classica). I segnali TX e RX vengono poi processati dal classico FT232RL per l'interfacciamento con la periferica USB del PC. Due LED di stato, LD3 e LD2, evidenziano lo scambio dati tra la nostra scheda di interfaccia e il nostro software installato sul PC. I pin RC3 (SCL) e RC4 (SDA) vengono utilizzati per interfacciarsi alla scheda elettrodo e quindi all'integrato MGC3130, nonché per pilotare l'I/O expander della Microchip per la gestione delle uscite digitali associate alle gesture. I segnali reset (RC0) e GPIO/IRQ (RC1 linea TS) servono per completare la sezione di comunicazione verso l'integrato MGC3130.

Il jumper J3, collegato al pin RC2, serve per decidere se le uscite digitali si devono comportare come monostabili o bistabili. Se il jumper è inserito e quindi l'ingresso è a livello logico basso le uscite, si devono comportare come monostabili; viceversa, funzionano come bistabili. Nella condizione di lavoro monostabile è possibile decidere il tempo di eccitazione dell'uscita sfruttando i tre pulsanti P1, P2 e P3: tenendo premuto il pulsante P1 per più di due secondi, verrà selezionato come tempo per il monostabile 1 secondo, mentre facendo altrettanto con P2 si impostano 5 secondi ecc.

Per evidenziare le possibili selezioni disponibili si sfruttano i LED collegati alla porta B del microcontrollore. Con i tre pulsanti a disposizione, si possono realizzare sette combinazioni utili al

fine di selezionare sette diversi valori di tempo per la funzione monostabile.

La **Tabella 2** mostra le possibili combinazioni con

i pulsanti P1, P2 e P3 e i rispettivi tempi selezionabili (ON significa premuto; OFF significa non premuto).

Facciamo notare che il tempo selezionato è globale e quindi viene associato a tutte le uscite digitali. In altre parole, una volta selezionato, ad esempio, un tempo di 5 secondi, questo sarà associato a tutte le uscite monostabili. Non è possibile associare tempi diversi per ogni uscita.

Notate che la condizione bistabile significa che l'uscita inverte la propria condizione ad ogni riconoscimento della gesture associata a tale uscita, quindi quando viene rilevato un gesto l'uscita si attiva, alla successiva l'uscita si disattiva e via di seguito.

La condizione monostabile invece eccita l'uscita associata alla gesture e la mantiene per il tempo selezionato (vedere tabella); inoltre il timer associato è retriggevole, nel senso che se viene rilevato un gesto prima dello scadere del tempo impostato, il tempo riparte da capo, prolungando lo stato di attivazione dell'uscita.

Analizziamo ora la sezione delle uscite digitali: grazie all'integrato MCP23017 possiamo rendere disponibili sedici uscite digitali associate alle gesture riconoscibili dall'integrato MGC3130.

L'integrato MCP23017 è un I/O expander configurabile ed è quindi possibile decidere quale dei suoi pin si debba comportare come ingresso e quale come uscita. Nella nostra applicazione abbiamo bisogno di sole uscite, quindi le Port A (GPA0÷7) e Port B (GPB0÷7) di tale integrato vengono configurate come uscite. Per eseguire questa operazione il microcontrollore si deve preoccupare, allo start-up, di inviare una serie di byte all'integrato per la programmazione di una serie di registri interni.



Fig. 3

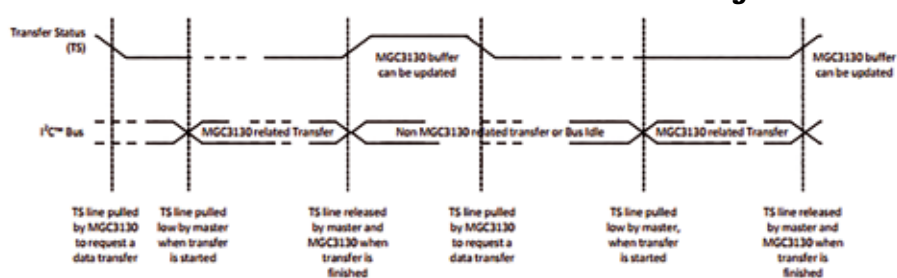
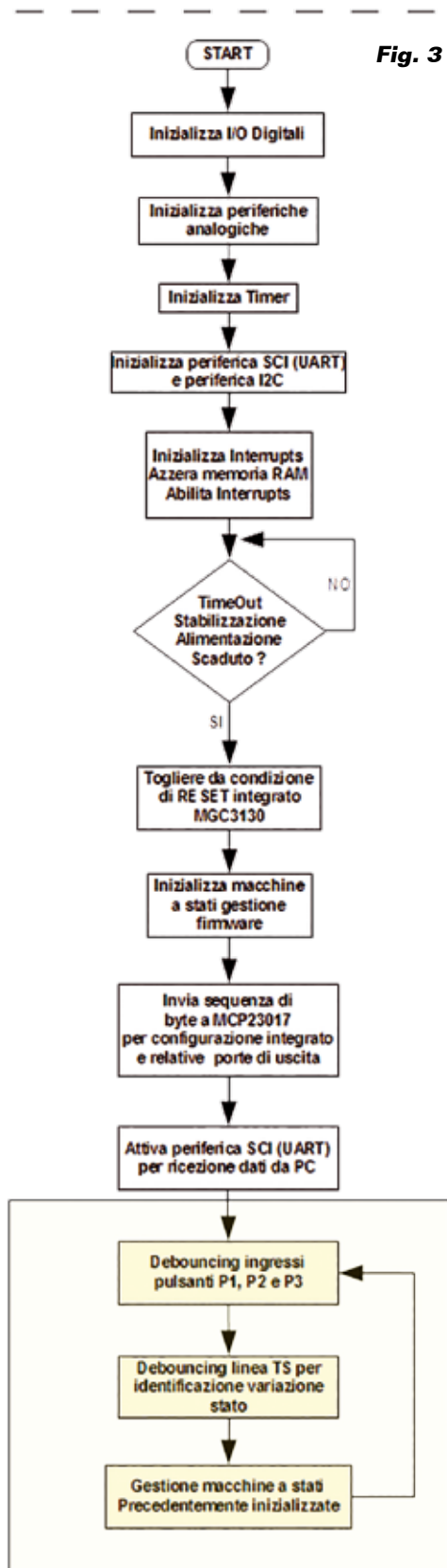


Fig. 4

L'interfaccia di comunicazione, come già ampiamente detto, è una I²C-Bus e per configurare l'indirizzo dell'integrato si devono impostare opportunamente i pin A0, A1 e A2; in questo caso sono tutti e tre collegati a GND, quindi l'address è zero. Disponendo di tre bit di indirizzo è possibile connettere sullo stesso bus fino a otto MCP23017.

L'indirizzo fisico del nostro integrato sarà 0x40, anzi, per l'esattezza 0x41 in lettura e 0x40 in scrittura; questo perché il bit meno significativo identifica il bit di R/W, come si vede in Fig. 1.

La Fig. 2 evidenzia il protocollo di comunicazione da utilizzare per la lettura/scrittura dei registri di MCP23017.

Le linee delle porte A e B vengono portate direttamente sui connettori di uscita CN5 e CN6 tramite una resistenza di protezione da 1 kohm (R20÷R27 e R31÷R38) cosicché si ha la possibilità di pilotare elettroniche esterne. I connettori riportano in uscita anche l'alimentazione +5 Vcc e la massa. Un esempio di utilizzo di queste uscite è tramite la scheda di Futura Elettronica (www.futura-shop.it) siglata 2846-RELAY8CH,

la quale mette a disposizione otto relé con bobina alimentata a 5 V. L'eccitazione/diseccitazione del relé avviene tramite interfaccia optoisolata, quindi quando l'uscita digitale della nostra scheda è a livello logico alto il relé di tale scheda sarà eccitato, viceversa sarà diseccitato. Facciamo notare che l'alimentazione +5 V viene prelevata direttamente dal connettore USB connesso al PC, quindi la corrente che si può ottenere è limitata (al massimo 500 mA). Oltre alla scheda a 8 canali, se vi servono meno uscite potete utilizzare, sempre della Futura Elettronica, altre due unità, rispettivamente a due e quattro relé; i rispettivi codici prodotto sono: 2846-RELAY2CH (scheda due relé), 2846-RELAY4CH (scheda quattro relé).

Le uscite digitali, oltre ad essere portate direttamente sui connettori di uscita CN5 e CN6, vengono ripetute sugli integrati ULN2803ADW (U5 e U6) i quali sono degli array di transistor configurati a Darlington. Le loro uscite (C1÷C8) vengono riportate sui connettori di uscita CN7 e CN8. Su questi connettori è presente sia l'u-

Fig. 5

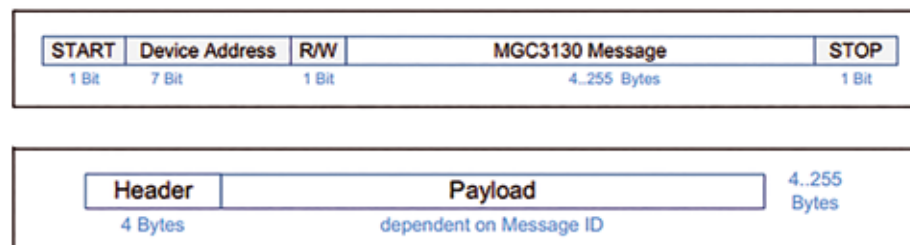


Fig. 6

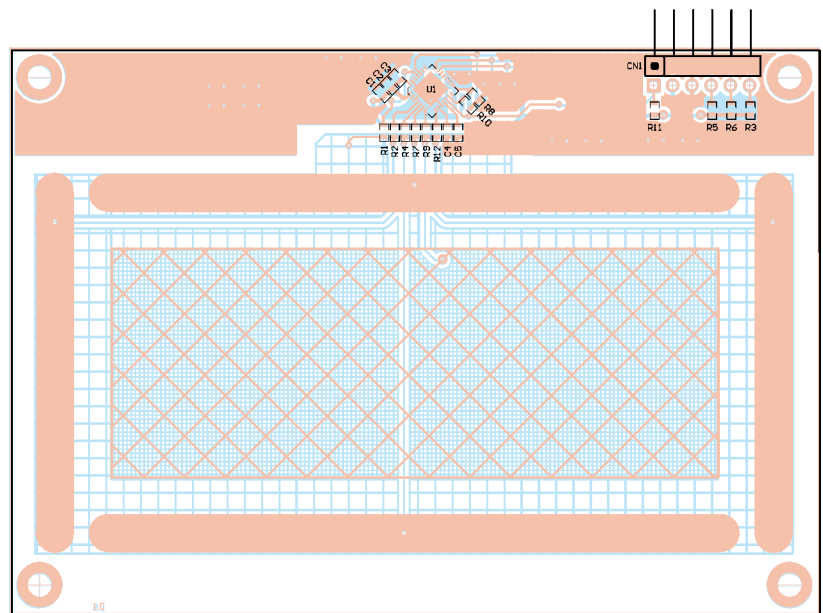
scita a 5 V della nostra scheda, sia un ingresso (pin 11) cui può essere applicata una tensione esterna che, tramite i jumper J1 e J2, può essere riportata sul pin Vcc degli integrati U5 e U6; ciò permette di pilotare dei relé con tensione della bobina diversa da quella prevista a bordo scheda, ovvero di 12 o 24 Vcc. Infatti la tensione dei relé dipende da quella applicata alla linea positiva comune dei diodi di protezione dell'ULN2803ADW, ovvero il piedino 10.

Le gesture che è possibile riportare sulle uscite digitali sono riepilogate nella **Tabella 3**.

L'alimentazione +3,3 V necessaria al corretto funzionamento dell'elettronica della scheda demo, nonché l'alimentazione tramite i connettori CN3 o CN9 portata alla scheda elettrodo, viene generata dal regolatore di tensione Low Drop Voltage MCP1801T-3302I/OT già utilizzato nella scheda Bridge. Il connettore CN9 è identico al connettore presente sulla scheda bridge, quindi un semplice BERG; invece, per chi lo preferisce, abbiamo predisposto anche il connettore CN3, che è un FPC da dieci pin con passo di 1 mm. Il cavo corrispondente è un WR-FFC, sempre da dieci poli con passo 1 mm, lungo 50 mm anch'esso.

REALIZZAZIONE PRATICA

Descritte le tre schede, vediamo come realizzarle in pratica partendo dai rispettivi circuiti stampati, ottenibili per fotoincisione a partire dalle tracce lato rame che potete scaricare dal nostro sito Internet; notate che in queste pagine, per brevità, abbiamo riportato solo il piano di montaggio della scheda con elettrodo da 100x50 mm, tuttavia le indicazioni fornite qui di seguito sono valide per tutte e tre, dato che tutte hanno gli stessi componenti.



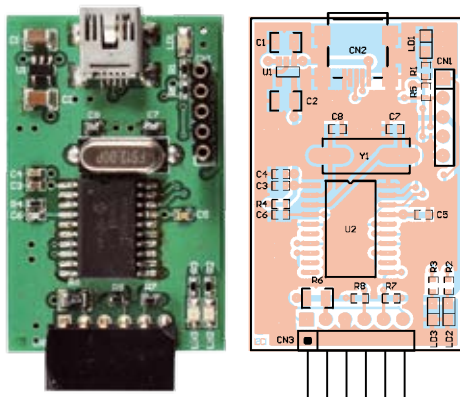
Elenco Componenti:

R1: 0 ohm (0603)	R12: 10 kohm (0603)	C5: 4,7 μ F 6,3 VL ceramico (0603)
R2÷R4: 10 kohm (0603)	C1: 100 nF 50 VL ceramico (0603)	U1: MGC3130
R5: non montare	C2: 100 nF 50 VL ceramico (0603)	
R6: non montare	C3: 10 μ F 6,3 VL ceramico (0603)	Varie:
R7÷R12: 10 kohm (0603)	C4: 4,7 μ F 6,3 VL ceramico (0603)	- Strip maschio 6 vie 90°
R8: 10 kohm (0603)		- Connettore WR-FPC
R9: 10 kohm (0603)		10 pin passo 1.0
R10: 10 kohm (0603)		- Circuito stampato
R11: 10 kohm (0603)		

[piano di montaggio **SCHEDA BRIDGE**]

Elenco Componenti:

R1 ÷ R3: 680 ohm (0603)
 R4: 22 kohm (0603)
 R5: 150 kohm (0603)
 R6: 0 ohm (0603)
 R7, R8: 1,8 kohm (0603)
 C1: 10 μ F 16 VL ceramico (1206)
 C2: 10 μ F 16 VL ceramico (1206)
 C3: 100 nF ceramico (0603)
 C4: 10 μ F 6,3 VL ceramico (0603)
 C5: 10 μ F 6,3 VL ceramico (0603)
 C6: 220 pF ceramico (0603)
 C7, C8: 22 pF ceramico (0603)
 U1: MCP1801T-3302I/OT
 U2: PIC18F14K50-I/SO
 Y1: Quarzo 12 MHz (FOXSDLF/120-20)
 LD1: LED rosso (0805)
 LD2, LD3: LED verde (0805)



Varie:

- Strip femmina 5 vie
- Strip femmina 6 vie 90°
- Connettore mini USB
- Circuito stampato

Per ognuno dei tre PCB, i componenti vanno montati sul top layer, mentre il solo connettore CN2 è posizionato sul bottom layer. Ricordiamo che il case dell'integrato MGC3130 è di tipo QFN e che quindi i contatti sono sotto il corpo; quindi per saldare questo componente bisogna utilizzare un saldatore ad aria calda, dopo aver deposto un leggero velo di lega saldante sui pad corrispondenti dello stampato e aver interposto uno strato di fluxante in pasta poco denso. Il fluxante anticipa la fusione della lega saldante e aiuta a posizionare quest'ultima una volta sciolta. Sconsigliamo la saldatura con un saldatore classico perché, anche se ha la punta finissima, diventa difficoltoso evitare di fare cortocircuiti tra i contatti. Una volta saldato l'integrato, controllate se tutte le saldature sono ben fatte e che non ci siano baffi di stagno che accidentalmente hanno unito contatti adiacenti; aiutatevi con una buona lente di ingrandimento. Fatto ciò, provvedete alla saldatura di tutte le resistenze e dei condensatori, nonché del connettore CN1 e, se volete, del connettore CN2. Quest'ultimo è

facoltativo, in quanto la nostra scheda di interfaccia prevede la possibilità di connettersi sia a CN1 che a CN2. Passate adesso al montaggio della scheda di interfaccia USB, vi consigliamo di cominciare dal microcontrollore in quanto è un componente SMD seguita dall'integrato MCP1801T. Montare tutte le resistenze e i condensatori seguiti dai LED, il quarzo e infine i connettori.

Il firmware da caricare nel microcontrollore ve lo forniamo noi, non dovrete fare altro che utilizzare un PicKit2 o 3 per eseguire la programmazione (previa costruzione dell'adattatore verso il nostro connettore CN1). Questa scheda non prevede alcuna taratura. Non rimane che collegarla alla presa USB del PC e vedere se il LED rosso LD1 si accende, controllare quindi tutte le alimentazioni del circuito. Vi ricordiamo che il connettore USB utilizzato è un mini-USB classico.

Per collegare la scheda bridge a uno dei nostri elettrodi assicuratevi prima di avere staccato il cavo USB dal PC, collegate la scheda bridge alla scheda elettrodo rispettando la polarità evidenziata

dal pallino (devono coincidere su entrambi i lati) e infine collegate il cavo USB.

Avviate il programma Aurea che avrete installato nel vostro PC e dovrete vedere iniziare a lampeggiare i LED LD2 e LD3 a indicare che c'è comunicazione tra il software Aurea sul PC e l'integrato MGC3130 montato sulla scheda elettrodo. A questo punto avete due possibilità: parametrizzate l'integrato MGC3130 secondo le vostre necessità seguendo i passi descritti nella precedente puntata oppure caricate la parametrizzazione da noi fornitavi direttamente in MGC3130 con la sola accortezza di utilizzare il giusto file a seconda della dimensione degli elettrodi.

Se invece avete realizzato un vostro personale elettrodo con delle dimensioni diverse dalle nostre e quindi con un diverso layout delle piste, dovete provvedere a parametrizzare l'integrato MGC3130 come già ampiamente spiegato. E passiamo all'ultima scheda: la demoboard. Per il montaggio si deve partire con il saldare tutti gli

Tabella 4

Messaggi per la gestione sistema di controllo	
ID	Nome
0x15	Codice pacchetto stato del sistema
0x06	Codice pacchetto per richiesta messaggio
0x83	Codice pacchetto versione firmware & Info
0xA2	Codice pacchetto per impostazione parametri operativi
Messaggi contenente i dati in uscita dopo riconoscimento gesture	
ID	Nome
0x91	Codice pacchetto contenente i dati dopo riconoscimento gesture
Messaggi per aggiornamento libreria GestIC	
ID	Nome
0x80	Codice pacchetto inizio aggiornamento firmware
0x81	Codice pacchetto aggiornamento blocco Flash
0x82	Codice pacchetto aggiornamento FW completato

integrati SMD partendo dall'integrato FT232RL, che è il più critico, passando poi al microcontrollore, all'I/O expander ecc. Completate con le resistenze e i condensatori, i LED e, infine, i connettori previsti.

DESCRIZIONE FIRMWARE

A questo punto diamo una breve descrizione del firmware che gestisce la nostra scheda demo e spieghiamo come ci si deve interfacciare verso l'integrato MGC3130. Iniziamo subito con il dire che il firmware è stato scritto in linguaggio C, utilizzando il compilatore C18 Ver. 3.37 della Microchip. Come al solito, abbiamo suddiviso il firmware in file distinti in modo da rendere più agevole identificare le sezioni di codice e il mantenimento dello stesso. Per ogni file *“.C”* (cartella Source Files) esiste l'equivalente *“.H”* (cartella Header Files): nel primo ritroviamo il codice C e nel secondo le definizioni delle costanti e delle funzioni. La **Fig. 3** mostra il diagramma di flusso della sezione principale del firmware; come si può vedere, al power-up il microcontrollore provvede ad eseguire una serie di inizializzazioni delle sue periferiche interne partendo dalle porte di ingresso, sia parte digitale che analogica, inizializza i timer, le periferiche di comunicazione I²C e SCI, inizializza tutta la SRAM (solo al power-up, se si è in presenza di un reset spurio del micro la SRAM non viene re-inizializzata), vengono abilitati gli interrupt, sia di periferica che di sistema. Fatto ciò, il microcontrollore entra in un loop di attesa di 250 ms dove mantiene l'integrato MGC3130 in stato di reset fin quando l'alimentazione si stabilizza. Dopodiché viene tolta la condizione di reset, vengono inizializzate tutte le macchine a stati per la gestione dei task e viene inizializzato l'integrato MCP23017, affinché tutti i

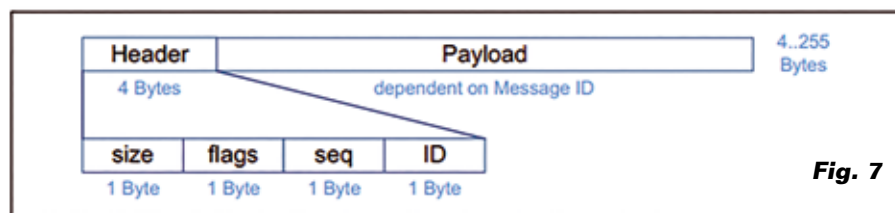


Fig. 7

84	00	00	83	AA	63	80	E6	0C	64	15	20	31	2E	30	2E	30	3B	70	3A	48	69	6C	6C	73	74	61	72
56	30	31	3B	44	53	50	3A	49	44	39	30	30	30	72	31	38	34	39	3B	69	3A	42	3B	66	3A	32	32
35	30	30	3B	6E	4D	73	67	3B	73	3A	42	65	74	61	32	72	31	30	34	30	3A	31	30	34	39	3A	4D
4F	3B	63	3A	4D	4B	49	3B	74	3A	32	30	31	33	2F	31	31	2F	30	38	20	31	33	3A	30	33	3A	30
00	10	00	00	55	AA	90	65	20	20	80	0F	FF	00	FF	00	E1	EA	00	00								

Fig. 8

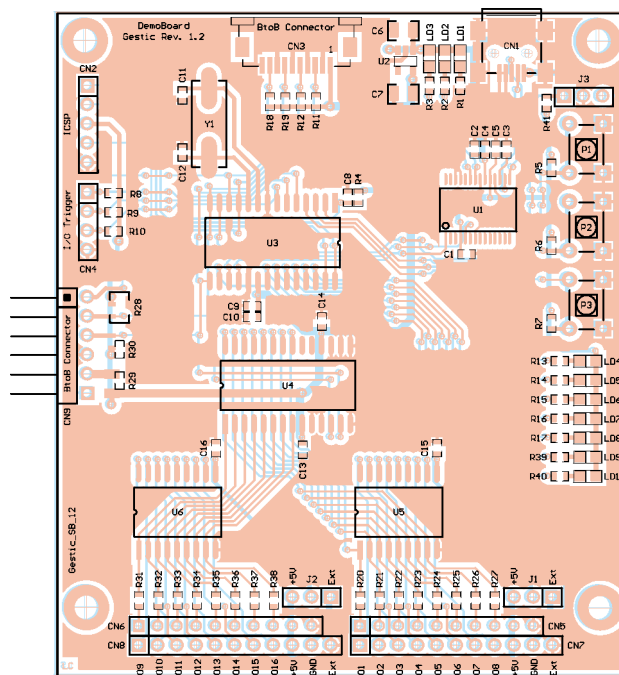
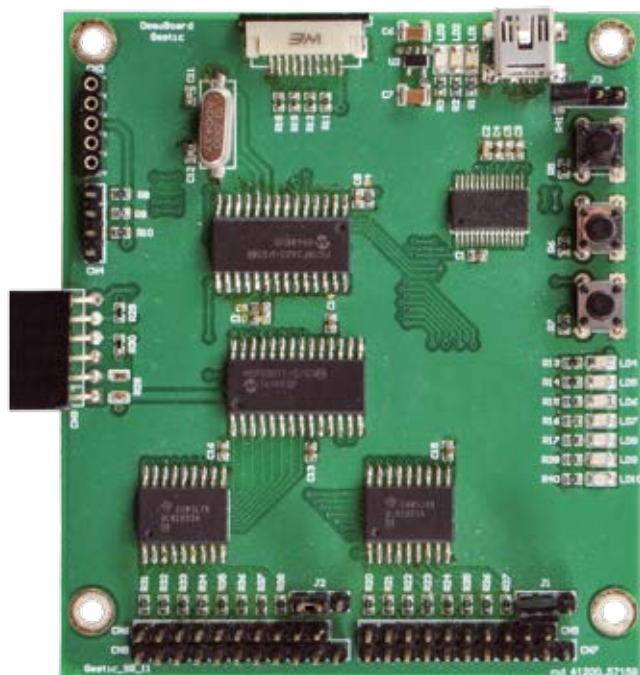
suoi punti di I/O siano configurati come uscite. In ultimo viene attivata in ricezione la periferica SCI (UART) che rimane in attesa di comandi da parte della nostra applicazione *“Gestic Tester”*.

Dopo tutte queste inizializzazioni, il microcontrollore inizia un loop infinito detto *“main”* dove processa tutte le macchine a stati dei vari Task gestiti, esegue il debouncing degli ingressi digitali (pulsanti P1, P2 e P3), monitora la linea TS di MGC3130 per la sua corretta gestione durante le fasi di lettura dati.

Analizziamo ora come interfacciarsi all'integrato MGC3130: per prima cosa diamo uno sguardo alla **Fig. 4**, dove viene evidenziato l'andamento della tensione sulla linea TS rispetto alle fasi della comunicazione I²C. Nel momento in cui viene rilevata una gesture da parte dell'MGC3130 di una scheda elettrodo affacciata sul bus, e quindi il buffer di uscita di quest'ultimo è stato caricato con i dati aggiornati, la linea TS, se libera, viene portata a livello basso da tale integrato. A questo punto il microcontrollore, tramite una gestione via interrupt sulla periferica capture (RC1), si accorge che la linea è andata a zero logico ed esso stesso, prima di iniziare la lettura, blocca la linea commutando il pin RC1 come uscita e portandolo a livello logico basso con capture disabilitato. Fatto ciò può iniziare una sessione

di lettura su bus I²C dei dati resi disponibili dall'MGC3130. Conclusa la lettura, il microcontrollore rilascia la linea TS portandola a livello logico alto e riconfigurandola nuovamente come ingresso (capture nuovamente attivo). A questo punto anche l'MGC3130 rilascia la linea TS e rimane in attesa di una nuova gesture, per poi ricominciare il ciclo.

I pacchetti dati che viaggiano sul bus I²C durante la comunicazione tra MGC3130 e microcontrollore devono rispettare una determinata formattazione: la **Fig. 5** generalizza il pacchetto dati evidenziando lo start bit, il byte di indirizzamento del device e subito dopo il pacchetto dati ricevuto/inviato da/all'MGC3130; segue bit di stop. Analizzando in dettaglio il pacchetto dati (**Fig. 6**) notiamo che si deve impostare una sezione di intestazione, detta *“Header”*, seguita dal *“Payload”*. Entrando ancora più nel dettaglio (**Fig. 7**) si nota che l'intestazione prevede quattro byte: il primo identifica la lunghezza del pacchetto dati compresa l'intestazione, il secondo byte serve per i *“Flags”* attualmente non utilizzato, il terzo byte indica la sequenza di messaggi inviata dall'MGC3130; tale byte è un numero che va da 0 a 255. Questa informazione può essere utilizzata dal microcontrollore per verificare se ha perso dei messaggi inviati dall'integrato; raggiunto il valore



Elenco Componenti:

R1 ÷ R3: 470 ohm (0603)
 R4: 22 kohm (0603)
 R5 ÷ R10: 4,7 kohm (0603)
 R11, R12: 1,8 kohm (0603)
 R13 ÷ R17: 680 ohm (0603)
 R18, R19: non montare
 R20 ÷ R27: 1 kohm (0603)
 R28: 0 ohm (1206)
 R29, R30: non montare
 R31 ÷ R38: 1 kohm (0603)
 R39, R40: 680 ohm (0603)
 R41: 4,7 kohm (0603)
 C1, C3: 100 nF 50 VL
 ceramico (0603)

C4, C5: 100 nF 50 VL
 ceramico (0603)
 C2: 10 μ F 6,3 VL
 ceramico (0603)
 C6, C7: 10 μ F 16 VL
 ceramico (1206)
 C8: 220 pF 50 VL
 ceramico (0603)
 C9: 100 nF 50 VL
 ceramico (0603)
 C10: 10 μ F 6,3 VL
 ceramico (0603)
 C11; C12: 22 pF 50 VL
 ceramico (0603)

C13 ÷ C16: 100 nF 50 VL
 ceramico (0603)
 U1: FT232RL
 U2: MCP1801T-3302I/OT
 U3: PIC18F2620-I/SO
 (MF1188)
 U4: MCP23017
 U5, U6: ULN2803ADW
 Y1: Quarzo 10 MHz
 (HC49/4HSMX)
 LD1, LD2: LED rosso (0805)
 LD3: LED giallo (0805)
 LD4 ÷ LD10: LED verde (0805)
 P1 ÷ P3: Microswitch

Varie:
 - Strip femmina 6 vie 90°
 - Connettore WR-FPC
 10pin passo 1.0
 - Strip femmina 5 vie
 - Strip maschio 4 vie
 - Strip maschio 10 vie (2 pz.)
 - Strip maschio 11 vie (2 pz.)
 - Strip maschio 3 vie (3 pz.)
 - Connettore mini USB
 - Circuito stampato

255 il contatore inizia nuovamente da zero. Il quarto byte dell'Header è "ID", che identifica il codice del messaggio.

Il "Payload" che segue dipende dal codice "ID" presente nell'intestazione e di conseguenza ha una lunghezza variabile a seconda dei casi. I codici "ID" li possiamo riassumere nella **Tabella 4**.

Descriviamo ora alcuni pacchetti inviati dall'MGC3130: dopo il Power-Up, l'integrato invia sempre un pacchetto dati contenente la revisione firmware caricata nell'in-

tegrato; la **Fig. 8** mostra questo pacchetto che andremo ad analizzare. I primi quattro byte identificano la sezione "Header" (sezione cerchiata in rosso) dal quale ricaviamo che la lunghezza totale del pacchetto dati è di 132 Byte (0x84) e l'"ID" del pacchetto è 0x83 (il codice 0x83 identifica il pacchetto dati con la versione firmware/libreria caricata nell'MGC3130). Subito dopo la sezione di "Header" segue il "Payload": il primo byte del "Payload" (sezione cerchiata in verde) dice se nell'integrato è

caricata una libreria Gestic valida oppure no. Se il valore caricato è 0x00 vuol dire che non c'è alcuna libreria caricata, mentre se è 0x0A significa che è stata caricata una libreria non valida. Se, invece, il codice è 0xAA, la libreria caricata è valida. Seguono due byte (sezione evidenziata in azzurro) che identificano la revisione hardware dell'integrato: la versione caricata è la 99.128 (letto come 0x63 e 0x80). Infine analizziamo la sezione evidenziata di arancione, la quale identifica la revisione firmware

caricata ed è in codice ASCII. Ogni parametro della revisione FW è separato dal carattere “;” come indicato nell’esempio seguente:

1.0.0;p:HillstarV01;DSP:ID9000r1849;i:B;f:22500;nMsg;s:Beta2r1040:1049:MO;c:MKI;t:2013/11/0813:03:0;...

I punti chiave di questa stringa sono quelli evidenziati in azzurro, giallo e verde; la parte di stringa evidenziata in grigio è riservata. La parte evidenziata in azzurro identifica la piattaforma e inizia sempre con una “p.”; la parte evidenziata in giallo identifica la versione della suite colibri e inizia sempre con “DSP.”. Infine, la parte evidenziata in verde riporta quando è stata compilata la versione di libreria e inizia sempre con “t.”. Quest’ultimo pacchetto dati viene sempre inviato dall’MGC3130 dopo il power-up. Un altro pacchetto dati fondamentale per la nostra applicazione è quello con codice “ID” 0x91 ovvero quello che trasmette le informazioni inerenti alla gesture riconosciuta dall’integrato. Come in tutti i pacchetti, abbiamo la sezione “Header”, dove ritroviamo il codice “ID” 0x91, seguita dal “Payload”. Del “Payload” bisogna prima di tutto considerare i primi quattro byte, di cui i primi due identificano la mascheratura dei dati in uscita, ovvero cosa è compreso nel pacchetto e cosa no. Consideriamoli come una singola word che di seguito analizzeremo bit per bit.

- Bit 0: identifica il campo “DSPStatus”; se è “1” è incluso nel pacchetto.
- Bit 1: identifica il campo “GestureInfo”; se è “1” è incluso nel pacchetto.
- Bit 2: identifica il campo “TouchInfo”; se è “1” è incluso nel pacchetto.

- Bit 3: identifica il campo “AirWheelInfo”; se è “1” è incluso nel pacchetto.
- Bit 4: identifica il campo “xyzPosition”; se è “1” è incluso nel pacchetto.
- Bit 5: identifica il campo “NoisePower”; se è “1” è incluso nel pacchetto.
- Bit 6 e 7: riservati.
- Bit 8 ÷ 10: identifica il campo “Configurazione elettrodi”.
- 000: identifica configurazione a quattro elettrodi senza elettrodo centrale.
- 001: identifica configurazione a cinque elettrodi.
- Bit 11: identifica il campo “CICData”; se è “1” è incluso nel pacchetto.
- Bit 12: identifica il campo “SDDData”; se è “1” è incluso nel pacchetto.
- Bit 13 ÷ 15: riservati.

Quindi a seconda della configurazione di cui sopra la lunghezza del “Payload” varierà. I bit di interesse vengono configurati durante la parametrizzazione oppure sfruttando l’“ID” 0xA2 per la modifica dei parametri operativi durante il funzionamento dell’applicazione. Nel nostro caso i bit da zero a cinque sono tutti settati a “1” e quindi inclusi nel “Payload”. Diversamente bisogna modificare

il codice firmware della nostra demo per tenere conto della lunghezza variabile del “Payload” (non implementato nel nostro firmware).

Il quarto byte dei primi quattro è una maschera che indica se i dati presenti nel “Payload” sono validi oppure no. Analizziamoli anch’essi bit per bit:

- Bit 0: identifica il campo “PositionValid”; se è “1” indica che il campo “xyzPosition” contiene dati validi.
- Bit 1: identifica il campo “AirWheelValid”; se è “1” indica che il campo “AirWheelInfo” contiene dati validi.
- Bit 2: identifica il campo “RawDataValid”; se è “1” indica che i campi “CICData” e “SDDData” contiene dati validi.
- Bit 3: identifica il campo “NoisePowerValid”; se è “1” indica che il campo “NoisePower” contiene dati validi.
- Bit 4: indica il campo “EnvironmentNoise”; se è “1” indica che è stato rilevato del rumore di fondo.
- Bit 5: detto “Clipping”, indica se il convertitore ADC ha tagliato.
- Bit 6: riservato.
- Bit 7: detto “DSPRunning” indica se il sistema sta girando o se è in modalità “sleep”.

Tabella 5

Header	Payload			
	ID	Riservato	Argomento 0	Argomento 1
0x10 0x00 0x00 0xA2	0x82 0x00	0x00 0x00	0x05 0x00 0x00 0x00	0x43 0x21 0x00 0x00

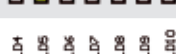
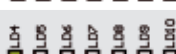
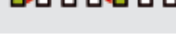
Tabella 6

Header	Payload			
	ID	Riservato	Argomento 0	Argomento 1
0x10 0x00 0x00 0xA2	0x82 0x00	0x00 0x00	0x04 0x00 0x00 0x00	0x31 0x04 0x00 0x00

Tabella 7

Header	Payload			
	ID	Riservato	Argomento 0	Argomento 1
0x10 0x00 0x00 0xA2	0xFF 0x00	0x00 0x00	0x01 0x00 0x00 0x00	0x00 0x00 0x00 0x00

Tabella 8 - Associazione delle gesture alle segnalazioni dei LED della demoboard.

Gesture	LED associato	
Touch elettrodo Sud	LED 5 acceso per 250 ms	
Touch elettrodo Ovest	LED 6 acceso per 250 ms	
Touch elettrodo Nord	LED 4 acceso per 250 ms	
Touch elettrodo Est	LED 7 acceso per 250 ms	
Touch elettrodo Centrale	LED 8 acceso per 250 ms	
Tap elettrodo Sud	LED 5 acceso per 500 ms	
Tap elettrodo Ovest	LED 6 acceso per 500 ms	
Tap elettrodo Nord	LED 4 acceso per 500 ms	
Tap elettrodo Est	LED 7 acceso per 500 ms	
Tap elettrodo Centrale	LED 8 acceso per 500 ms	
Double Tap elettrodo Sud	LED 5 acceso per 1 s	
Double Tap elettrodo Ovest	LED 6 acceso per 1 s	
Double Tap elettrodo Nord	LED 4 acceso per 1 s	
Double Tap elettrodo Est	LED 7 acceso per 1 s	
Double Tap elettrodo Centrale	LED 8 acceso per 1 s	
Gesture mano da Ovest a Est	LED accesi in sequenza dal LED 4 al LED 8 con intervallo di 250 ms	
Gesture mano da Est a Ovest	LED accesi in sequenza dal LED 8 al LED 4 con intervallo di 250 ms	
Gesture mano da Sud a Nord	LED accesi in sequenza a partire dai LED 4 e 8 verso LED 6 con intervallo di 250 ms	
Gesture mano da Nord a Sud	LED accesi in sequenza a partire dal LED 6 verso LED 4 e 8 con intervallo di 250 ms	
Gesture mano da Ovest a Est sul bordo (Edge Flick)	LED accesi in sequenza dal LED 4 al LED 8 con intervallo di 100 ms	
Gesture mano da Est a Ovest sul bordo (Edge Flick)	LED accesi in sequenza dal LED 8 al LED 4 con intervallo di 100 ms	
Gesture mano da Sud a Nord sul bordo (Edge Flick)	LED accesi in sequenza a partire dai LED 4 e 8 verso LED 6 con intervallo di 100 ms	
Gesture mano da Nord a Sud sul bordo (Edge Flick)	LED accesi in sequenza a partire dal LED 6 verso LED 4 e 8 con intervallo di 100 ms	
Movimento circolare orario	LED accesi in sequenza fissa da LED 4 a LED 8	
Movimento circolare antiorario	Esegue movimento contrario del punto precedente	

Analizziamo ora la parte variabile del **"Payload"**: in particolare le voci **"GestureInfo"**, **"TouchInfo"**, **"AirWheelInfo"** e **"xyzPosition"**. Partiamo da **"GestureInfo"**, che occupa quattro byte e contiene diverse informazioni sulle gesture riconosciute. Il primo byte è composto dai Bit 0÷7 che identificano la gesture riconosciuta con numeri in formato decimale, che sono:

- 0 = nessuna gesture;
- 1: garbage;
- 2: gesture movimento mano da Ovest a Est;
- 3: gesture movimento mano da Est a Ovest;
- 4: gesture movimento mano da Sud a Nord;
- 5: gesture movimento mano da Nord a Sud;
- 6: gesture movimento circolare in senso orario;
- 7: gesture movimento circolare in senso anti-orario.

Il secondo byte, composto dai Bit 8÷11, è vuoto (bit non usati). Invece i Bit 12÷15 definiscono la tipologia di gesture riconosciuta (in formato decimale):

- 0 = garbage;
- 1 = gesture movimento mano;
- 2 = gesture circolare.

Il Bit 16 permette di sapere quando è stata rilevata la gesture di passaggio della mano sul bordo dell'elettrodo (Edge Flick).

I Bit 17÷30 sono riservati. Infine, il Bit 31 vale 1 quando è in corso il riconoscimento di una gesture, vale a dire quando c'è una variazione nel campo elettrico e l'integrato Microchip sta cercando di decifrarla.

La voce **"TouchInfo"** occupa anch'essa quattro byte e contiene le informazioni su come gli elettrodi sono stati toccati dalle dita dell'utente; i primi due sono così composti:

- Bit 0 = se vale **"1"** è stato tocca-

- to l'elettrodo Sud;
- Bit 1 = se vale "1" è stato toccato l'elettrodo Ovest;
 - Bit 2 = se vale "1" è stato toccato l'elettrodo Nord;
 - Bit 3 = se vale "1" è stato toccato l'elettrodo Est;
 - Bit 4 = se vale "1" è stato toccato l'elettrodo Centrale;
 - Bit 5 = se vale "1" è stato dato un colpo sull'elettrodo Sud;
 - Bit 6 = se vale "1" è stato dato un colpo sull'elettrodo Ovest;
 - Bit 7 = se vale "1" è stato dato un colpo sull'elettrodo Nord;
 - Bit 8 = se vale "1" è stato dato un colpo sull'elettrodo Est;
 - Bit 9 = se vale "1" è stato dato un colpo sull'elettrodo Centrale;
 - Bit 10 = se vale "1" è stato dato un doppio colpo sull'elettrodo Sud;
 - Bit 11 = se vale "1" è stato dato un doppio colpo sull'elettrodo Ovest;
 - Bit 12 = se vale "1" è stato dato un doppio colpo sull'elettrodo Nord;
 - Bit 13 = se vale "1" è stato dato un doppio colpo sull'elettrodo Est;
 - Bit 14 = se vale "1" è stato dato un doppio colpo sull'elettrodo Centrale;
 - Bit 15 = riservato.

Notare che per "colpetto" si intende toccare l'elettrodo e ritrarre la mano. Per quanto riguarda il terzo byte (Bit 16÷23) contiene il valore del contatore a 8 bit che esprime il tempo tra l'inizio del movimento della mano e la conclusione della gesture.

Ogni unità vale 5 ms.

Infine, i Bit 24÷31 sono riservati. La voce "AirWheelInfo" occupa due byte, il primo dei quali è un contatore e indica fino a dove è arrivata la gesture ovvero quanti giri completi sono stati fatti; ogni 32 digit si ha una rotazione completa.

Se il valore viene incrementato si ha una rotazione in senso orario viceversa in senso antiorario. Il secondo byte è riservato.

La voce "xyzPosition" occupa sei byte, due per ogni coordinata, e indica la posizione rispetto all'origine che si trova sull'angolo formato dagli elettrodi Sud-Ovest in basso a sinistra.

Il resto dei dati presenti nel "Payload" non interessa questa trattazione; chi volesse ragguagli potrà consultare il data-sheet. Diamo ora una brevissima descrizione per quanto riguarda il codice "ID" 0xA2, che serve per settare i parametri operativi da firmware. Nella nostra demo non è usato ma potrebbe tornare utile in diverse occasioni. Il messaggio si presenta con il solito "Header" da quattro byte con lunghezza fissa a 0x10 (Sedici Byte) e ovviamente "ID" 0xA2. Il "Payload" è diviso in quattro parti distinte:

- "RunTimeParameterID" (composta da due byte) identifica l'ID (che è univoco) del parametro da modificare;
- due byte riservati;
- "Argument0" (quattro byte) che dipende dall'ID; se non usato deve essere posto a 0x00000000;
- "Argument1" (quattro byte) che dipende dall'ID; se non usato deve essere posto a 0x00000000.

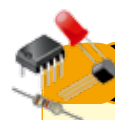
Per esplicitare il concetto facciamo un esempio di configurazione supponendo di voler modificare la quantità e la sequenza di salto delle frequenze di trasmissione per l'elettrodo TX. Il codice "ID" del parametro è 0x82; l'argomento 0 può accettare i valori tra 1 e 5 mentre l'argomento 1 vuole l'ordine di salto delle frequenze. Quindi se il pacchetto dati diventa come in **Tabella 5** significa che si sta ordinando all'MGC3130 di usare tutte e cinque le frequenze (Argomento 0) e la sequenza in

cui si deve testarle (Argomento 1) 4 → 3 → 2 → 1 → 0. I dati inseriti in "Argomento 0" vanno espressi come "nibble". Se invece si vogliono usare quattro frequenze con una sequenza del tipo 3 → 1 → 0 → 4, il pacchetto dev'essere composto come in **Tabella 6**.

Ricordiamo che una volta modificato il parametro questo deve essere "congelato" in Flash e quindi si deve inviare un pacchetto dati che dica all'MGC3130 di salvare i dati ricevuti in modo permanente.

Il codice "ID" per questo pacchetto è 0xFF00; l'argomento 0 può accettare solo i seguenti valori 0, 1 e 2. Con il valore 0 si dice all'MGC3130 di congelare la categoria "AFE", con 1 la categoria "DSP" e con 2 la categoria "System".

Si capisce che i parametri sono suddivisi in tre categorie distinte. L'argomento 1 non è utilizzato. Supponendo di voler salvare in Flash i parametri "DSP" il pacchetto sarà come in **Tabella 7**. Bene, per il momento abbiamo concluso; nella prossima puntata parleremo del software che abbiamo sviluppato per abbinarlo alla nostra scheda demo e spiegheremo come integrare l'MGC3130 nel mondo Arduino.



per il MATERIALE

Il software Aurea di Microchip può essere scaricato dal sito www.microchip.com/gestic. Il sensore di prossimità (cod. MGC3130) è reperibile presso Futura Elettronica a 9,80 Euro.

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

Dotiamo Arduino di una JPEG camera seriale con illuminatore IR e realizziamo due applicazioni: nella prima salviamo le riprese a intervalli regolari su SD-Card e nella seconda usiamo un sensore PIR per rilevare le intrusioni e fotografiamo ciò che avviene.



ARDUINO JPEG CAMERA APPLICATION

..... degli Ing. TOMMASO E ALESSANDRO GIUSTO

Da un po' di tempo sono comparse sul mercato telecamere miniaturizzate munite di interfaccia di comunicazione seriale. Rispetto alle camere tradizionali, offrono il vantaggio di una più semplice integrazione con sistemi quali Arduino o schede basate su microcontrollori. È abbastanza facile comprendere che utilizzare nei propri progetti una telecamera munita di interfaccia videocomposito o USB richiede uno sforzo notevole in termini sia hardware che software; se invece si decide di puntare su una camera ad interfaccia se-

riale, il tutto risulta più semplice, in quanto tutte le schede di prototipazione (Arduino in primis) e tutti i microcontrollori (Microchip, Atmel ecc.) dispongono come dotazione di almeno una porta seriale. Il concetto base di queste camere è che è possibile inviare tutti i comandi di impostazione, esecuzione screenshot e lettura immagini attraverso opportune sequenze di byte via porta seriale. Nel nostro caso utilizzeremo una camera della LinkSprite collegata ad una scheda Arduino Uno. Particolarità di questa telecamera è la possibilità

La JPEG camera

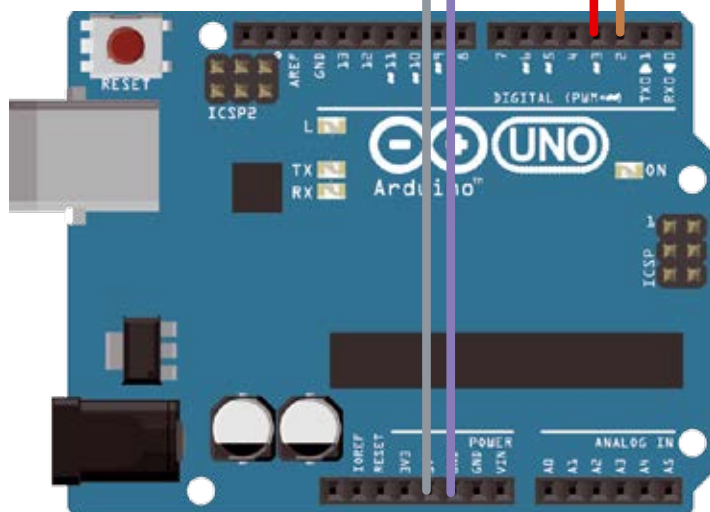
La telecamera utilizzata nelle applicazioni descritte in questo articolo è prodotta dalla LinkSprite ed è del tipo ad interfaccia seriale a livello TTL (il produttore ne costruisce anche una a livello RS232); inoltre è dotata di un illuminatore ad infrarossi per effettuare riprese al buio. L'illuminatore è composto da sei LED IR posti accanto all'obiettivo della telecamera e viene acceso automaticamente dalla logica in base all'illuminazione nell'ambiente rilevata da un fotoreistore, opportunamente chiuso lateralmente (da un tubo di gomma) per evitare di essere influenzato dai LED una volta che si accendono.

Le caratteristiche principali della JPEG camera sono:

- risoluzione VGA/ QVGA/ 160x120 pixel;
- supporto per la cattura di immagini salvate in formato JPEG;
- interfaccia seriale TTL con baud-rate impostabile (quello predefinito è 38.400 bps);
- tensione di alimentazione compresa fra 3,3 e 5 Vcc;
- assorbimento: 80-100mA;
- dimensioni: 46 x 26 mm.

colore filo	significato
rosso	RXD (IN)
marrone	TXD(OUT)
viola	GND
grigio	5V DC

Collegamento tra Arduino Uno e la telecamera a interfaccia seriale; quest'ultima viene fornita completa di cavetto (la tabella qui sopra illustra le connessioni), i cui fili vanno collegati, mediante pin-strip (perché hanno connessioni femmina come Arduino e gli shield) come indicato nell'immagine.



di catturare immagini anche in notturna; infatti dispone di un sensore di luminosità e di alcuni LED a infrarossi attivati in automatico quando la luce rilevata scende al di sotto di una determinata soglia (riferitevi alle immagini presenti in queste pagine per farvi un'idea del funzionamento dei LED infrarossi).

La telecamera è in grado di catturare immagini ad alta risoluzione e di inviarle in formato JPEG attraverso la propria porta seriale. Dispone di un connettore a quattro poli dove sono presenti due pin dedicati all'alimentazione (+5V e GND) ed altrettanti corrispondenti ai canali dati (RX e TX) della porta seriale.

In questo articolo mostreremo come integrare questa camera all'interno di due progetti basati su Arduino: il primo realizza un sistema di salvataggio periodico di screenshot, mentre con il secondo si può realizzare un sistema di salvataggio attivato solo se l'apposito sensore PIR (radar a infrarossi passivi) rileva il movimento di un corpo nella zona inquadrata dalla telecamera; in pratica questa seconda applicazione costituisce un dispositivo di videosorveglianza time-lapse che registra quello che accade in un locale a seguito del rilevamento di persone o veicoli in movimento. Naturalmente bisogna che il sensore PIR sia puntato nella stessa direzione in cui è orientata la telecamera, così che le riprese siano quelle della zona dove effettivamente si verifica l'intrusione.

Le immagini verranno salvate su una SD-Card in formato file JPEG e pertanto a livello hardware faremo uso di un opportuno SD-Card shield (disponibile presso Futura Elettronica).

Infine, a livello software utilizzeremo delle librerie (JPEGCamera)

Listato 1

da noi sviluppate ad hoc per questi esempi d'uso della camera.

HARDWARE DI SISTEMA

Per quanto riguarda la prima applicazione, l'hardware del sistema è composto da una scheda Arduino Uno Rev. 3 più uno shield SD-Card usato per interfacciarsi alla SD dove verranno salvati i file delle immagini; il tutto, abbinato alla camera seriale. Lo shield per SD dispone di quattro ponticelli che ci permettono di scegliere quale pin di Arduino utilizzare come chip select (è possibile scegliere tra D4, D8, D9 e D10); per chip select si intende la linea che abilita lo shield.

Nei nostri esempi software utilizziamo D4, pertanto andrà inserito il ponticello corrispondente. Per le connessioni, la camera dispone di un connettore a quattro poli per alimentazione e segnali della seriale e viene fornita con un apposito cavetto a quattro fili (colori: rosso, marrone, viola e grigio) terminanti con jumper femmina. Facciamo notare che per la comunicazione seriale tra Arduino e camera abbiamo deciso di non utilizzare la porta seriale hardware, ma ne usiamo una software (mappata sui pin D2 e D3 di Arduino); in questo modo la libreria software lascia la porta HW libera per il debug dell'applicativo utente.

Per il collegamento tra Arduino e camera è necessario collegare il filo rosso (RXD) al pin D3 di Arduino; il marrone (TXD) a D2; il grigio al segnale +5V ed infine il viola a massa (GND).

Per quanto riguarda la seconda applicazione, all'hardware appena descritto va aggiunto un sensore PIR; quello che abbiamo utilizzato per il nostro prototipo è una periferica dotata di una certa "intelligenza" on-board, che

```

/*****
 * Inclusione librerie
 *****/
#include <JPEGCamera.h>
#include "SoftwareSerial.h"
#include <SD.h>

// Chunk size per salvataggio immagine
#define CHUNK_SIZE 64

/*****
 * Definizione pin I/O Arduino
 *****/
// Led scheda Arduino
const int pinBoardLed = 13;
// Pin Arduino Chip Select SD
const int pinSDChipSelect = 4;

/*****
 * Variabili programma
 *****/

// Oggetto JPEG Camera
JPEGCamera jpegCamera;

// JPEG file
File jpegFile;
// Contatore immagine
byte i = 0;

/*****
 * Codice programma
 *****/

// Inizializzazione Scheda
void setup() {
  // Inizializzo I/O
  pinMode(pinBoardLed, OUTPUT);
  pinMode(pinSDChipSelect, OUTPUT);
  // Accendo led Arduino
  digitalWrite(pinBoardLed, HIGH);

  // Inizializzo JPEG Camera
  jpegCamera.begin();

  // Init SD
  pinMode(10, OUTPUT);
  if (!SD.begin(pinSDChipSelect)) {
    // Lampeggio led Arduino
    for (;;) {
      digitalWrite(pinBoardLed, HIGH);
      delay(500);
      digitalWrite(pinBoardLed, LOW);
      delay(500);
    }

    // Spengo led Arduino
    digitalWrite(pinBoardLed, LOW);
  } // Chiusura funzione setup

  // Programma Principale
  void loop() {
    boolean isEnd = false;
    uint16_t address = 0x0000;
    uint16_t chunkSize = CHUNK_SIZE;
    byte body[CHUNK_SIZE];

    // Imposto dimensione immagine
    jpegCamera.setImageSize(jpegCamera.ImageSize320x280);
    // Reset JPEG Camera
    jpegCamera.reset();

    // Scatto immagine
    jpegCamera.takePicture();

    // Identifico nome file
    char fileName[12] = "image00.jpg";
    fileName[5] = ((i / 10) + 0x30);
    fileName[6] = ((i % 10) + 0x30);

    // Apro file
    jpegFile = SD.open(fileName, FILE_WRITE);

    // Leggo/salvo i dati immagine
    isEnd = false;
    while (isEnd == false) {
      jpegCamera.readJpegFileContent(address, chunkSize, body, &isEnd);
      address += chunkSize;
      // Salvo i dati sul file
      jpegFile.write(body, chunkSize);
    }

    // Chiudo file
    jpegFile.close();
    // Fermo immagine
    jpegCamera.stopTakingPictures();
    // Attesa
    delay(1000);

    // Prossimo file
    i = ((i + 1) % 100);
  } // Chiusura funzione loop

```

Listato 2

```

/*****
* Inclusione librerie
*****/
#include <JPEGCamera.h>
#include "SoftwareSerial.h"
#include <SD.h>

// Chunk size per salvataggio immagine
#define CHUNK_SIZE 64

/*****
* Definizione pin I/O Arduino
*****/
// Led scheda Arduino
const int pinBoardLed = 13;
// Pin Arduino Chip Select SD
const int pinSDChipSelect = 4;
// Sensore PIR scheda Arduino
const int pinPIRSensor = 5;

/*****
* Variabili programma
*****/

// Oggetto JPEG Camera
JPEGCamera jpegCamera;

// JPEG file
File jpegFile;
// Contatore directory
int dirCount = 0;
// Contatore immagine
byte fileCount = 1;

boolean flagSensorePIR = false;

/*****
* Codice programma
*****/

// Inizializzazione Scheda
void setup() {
    // Inizializzo I/O
    pinMode(pinBoardLed, OUTPUT);
    pinMode(pinSDChipSelect, OUTPUT);
    pinMode(pinPIRSensor, INPUT);
    // Accendo led Arduino
    digitalWrite(pinBoardLed, HIGH);

    // Inizializzo JPEG Camera
    jpegCamera.begin();

    // Init SD
    pinMode(10, OUTPUT);
    if (!SD.begin(pinSDChipSelect)) {
        // Lampeggio led Arduino
        for (;;) {
            digitalWrite(pinBoardLed, HIGH);
            delay(500);
            digitalWrite(pinBoardLed, LOW);
            delay(500);
        }
    }

    // Identifico prima directory libera
    for (;;) {
        char dirName[7] = "DIR000";
        dirName[3] = ((dirCount / 100) + 0x30);
        dirName[4] = (((dirCount % 100) / 10) + 0x30);
        dirName[5] = ((dirCount % 10) + 0x30);
        // Se directory non esiste
        if (SD.exists(dirName) == false)
            break;

        // Prossima directory
        dirCount++;
    }

    // Spengo led Arduino
    digitalWrite(pinBoardLed, LOW);
} // Chiusura funzione setup

// Programma Principale
void loop() {
    boolean isEnd = false;
    uint16_t address = 0x0000;
    uint16_t chunkSize = CHUNK_SIZE;
    byte body[CHUNK_SIZE];

    // Imposto dimensione immagine
    jpegCamera.setImageSize(jpegCamera.ImageSize320x280);
    // Reset JPEG Camera
    jpegCamera.reset();

    // Scatto immagine
    jpegCamera.takePicture();

    // Identifico prima directory libera
    char dirName[7] = "DIR000";
    dirName[3] = ((dirCount / 100) + 0x30);
    dirName[4] = (((dirCount % 100) / 10) + 0x30);
    dirName[5] = ((dirCount % 10) + 0x30);
    // Se directory non esiste

```

implementa due diverse modalità di trigger, prevede un tempo di ritardo (tra il rilevamento del movimento e l'attivazione dell'uscita di trigger) programmabile tra 0,3 a 180 secondi, ed è capace di un angolo di rilevamento di 120° e una portata massima di 7 metri. Il sensore dispone di tre piedini (uscita TTL 0V – 3.3V, positivo e negativo di alimentazione). Per il collegamento con Arduino è necessario crearsi un cavetto a tre poli in cui i pin positivo e negativo di alimentazione andranno posizionati in +5V e GND di Arduino mentre l'uscita andrà portata nel pin D5 di Arduino. L'uscita è quella tramite la quale il dispositivo fornisce il segnale di trigger seguente il rilevamento del sensore PIR.

LIBRERIA ARDUINO

JPEGCamera

Per consentire ad Arduino di gestire la telecamera JPEG abbiamo sviluppato un'apposita libreria (libreria JPEGCamera) che offre tutte le funzioni di inizializzazione, gestione cattura screenshot e salvataggio immagini relative alla camera LinkSprite. La libreria fornisce un oggetto denominato *JPEGCamera* da utilizzare per la gestione completa dell'hardware.

In essa è presente una funzione *Begin* che inizializza tutto il sistema per la gestione della camera, una *Reset* che fisicamente resetta la camera, una *setImageSize* che imposta la dimensione dell'immagine catturata (è possibile scegliere tra 160 per 120 pixel; 320 per 280 e 640 per 480); c'è poi la funzione *takePicture*, che comanda alla camera di catturare l'immagine correntemente inquadrata (sostanzialmente è la funzione che esegue lo screenshot) e la *readJpegFileSize*, che legge la dimensione (in byte) del

file dell'immagine memorizzata. Altre funzioni sono la *readJpegFileContent*, che si occupa di leggere (un pacchetto alla volta) i dati corrispondenti all'immagine ripresa ed infine la *stopTakingPictures*, che ferma la cattura degli screenshot.

La funzione *readJpegFileContent* merita una menzione particolare: la lettura dei dati avviene un pacchetto (di 64 byte, nel nostro caso...) alla volta. Per fare questo è necessario, ogni volta che si chiama la routine, indicare l'indirizzo di inizio lettura (il primo caso è pari a 0, il secondo a 64, il terzo a 128 e così via) e la dimensione del pacchetto da leggere (nel nostro caso, sempre 64; facciamo notare che non è possibile superare il valore di 64). La funzione richiede come input un array di byte dove verranno

Fig. 1



Fig. 2

Listato 2

```
if (SD.exists(dirName) == false)
// Crea directory
SD.mkdir(dirName);

// Identifico nome file
char pathComplete[22] = "DIR000/image00.jpg";
pathComplete[3] = ((dirCount / 100) + 0x30);
pathComplete[4] = ((dirCount % 100) / 10) + 0x30);
pathComplete[5] = ((dirCount % 10) + 0x30);
pathComplete[12] = ((fileCount / 10) + 0x30);
pathComplete[13] = ((fileCount % 10) + 0x30);

// Apro file
jpegFile = SD.open(pathComplete, FILE_WRITE);

// Leggo/salvo i dati immagine
isEnd = false;
while(isEnd == false) {
  jpegCamera.readJpegFileContent(address, chunkSize, body, &isEnd);
  address += chunkSize;
  // Salvo i dati sul file
  jpegFile.write(body, chunkSize);

  // Se sensore PIR non rilevato
  if (flagSensorePIR == false) {
    // Se sensore PIR attivo
    if (digitalRead(pinPIRSensor) == HIGH) {
      // Prossimo file è il 6 (verrà incrementato in seguito)
      fileCount = 5;
      flagSensorePIR = true;
    }
  }

  // Chiudo file
  jpegFile.close();
  // Fermo immagine
  jpegCamera.stopTakingPictures();
  // Attesa
  delay(100);

  // Se sensore PIR non rilevato
  if (flagSensorePIR == false) {
    // Se sensore PIR attivo
    if (digitalRead(pinPIRSensor) == HIGH) {
      // Prossimo file è il 6
      fileCount = 6;
      flagSensorePIR = true;
    }
    // Se sensore PIR non attivo
    else {
      // Prossimo file number
      if (fileCount == 5)
        fileCount = 1;
      else
        fileCount++;
    }
  }
  // Se sensore PIR rilevato
  else {
    // Se salvati 5 file
    if (fileCount == 10) {
      // Resetto file
      fileCount = 1;
      // Prossima directory
      dirCount++;
      // Indico reset PIR
      flagSensorePIR = false;
    }
    else
      fileCount++;
  }
} // Chiusura funzione loop
```

salvati i dati; inoltre ritorna in uscita il parametro *isEnd*, il quale indica se l'operazione di lettura dell'immagine è stata completata.

SKETCH ESEMPIO 1

Partendo dall'inizio analizziamo il sorgente del primo esempio di utilizzo della camera JPEG (**Listato 1**) che prevede il salvataggio continuo delle immagini. Includiamo le librerie necessarie per il funzionamento

(chiaramente la *JPEGCamera.h*, la *SoftwareSerial.h* - in quanto utilizzeremo una seriale software- e la *SD.h*, che ci servirà per gestire il salvataggio delle immagini sulla SD-Card) e definiamo la dimensione (in byte) del singolo pacchetto oggetto del salvataggio (*CHUNK_SIZE*; 64 byte).

In seguito definiamo il pin di Arduino (D4) che useremo come chip select per la SD-Card e definiamo alcune variabili che

Il sensore PIR

Il sensore PIR da utilizzare nella seconda delle applicazioni descritte in questo articolo è un sistema munito di PIR, scheda di controllo e uscita TTL (livello alto 3,3V - basso 0V). L'alimentazione è compresa tra 5V e 12V (il consumo a riposo è di 65 μ A) e la distanza di rilevamento è configurabile tra 3 e 7 metri (impostabile tramite uno dei trimmer a bordo). L'angolo di rilevamento è minore di 120° e il tempo di ritardo di attivazione è configurabile tra 0,3 e 180 secondi (impostabile tramite il secondo trimmer).

Il sensore prevede due modalità di trigger (selezionabili da jumper):

- nella prima, denominata *non ripetibile*, quando il dispositivo rileva il movimento, l'uscita TTL

si porta a livello alto; dopo che è trascorso il tempo del ritardo impostato, l'uscita torna automaticamente al livello logico basso;

- nella seconda, denominata *trigger ripetibile*, quando il sensore rileva oggetti in movimento, l'uscita va a livello alto e rimane in tale stato fin tanto che il sensore rileva il movimento, senza considerare il ritardo impostato, poi torna ad assumere lo zero logico.



useremo all'interno del programma (in particolare *jpegCamera*, usato per la completa gestione della camera). All'interno della funzione *setup* inizializziamo la scheda e richiamiamo la *begin* della camera; in seguito iniziamo la *libreria SD* indicando che vogliamo usare il pin D4 per gestire il chip select.

Nella funzione *loop* gestiamo la cattura e il salvataggio vero e proprio delle immagini; come prima cosa impostiamo che vogliamo catturare immagini da 320 per 280 pixel e tramite *jpegCamera.takePicture* avviamo la cattura del singolo screenshot. In seguito definiamo il nome del file di salvataggio (utilizziamo la convenzione *imageXX.jpg* dove il valore XX varia da 00 a 99) e attraverso *SD.open* creiamo ed apriamo in scrittura il file sulla SD card. In seguito, attraverso un ciclo *while* gestiamo la completa lettura e salvataggio delle immagini; con *jpegCamera.readJpegFileContent* leggiamo blocchetto alla volta i byte del file *jpeg* ed attraverso la *write* dell'oggetto *File* li salviamo.

All'uscita del ciclo *while* tutti i

byte dell'immagine sono stati letti e salvati; richiamiamo quindi *close* dell'oggetto *File* per chiudere il file e tramite *jpegCamera.stopTakingPictures* indichiamo alla camera di fermare la cattura delle immagini.

Prima di terminare la *loop* attendiamo un secondo ed incrementiamo la variabile che ci tiene conto del numero del file.

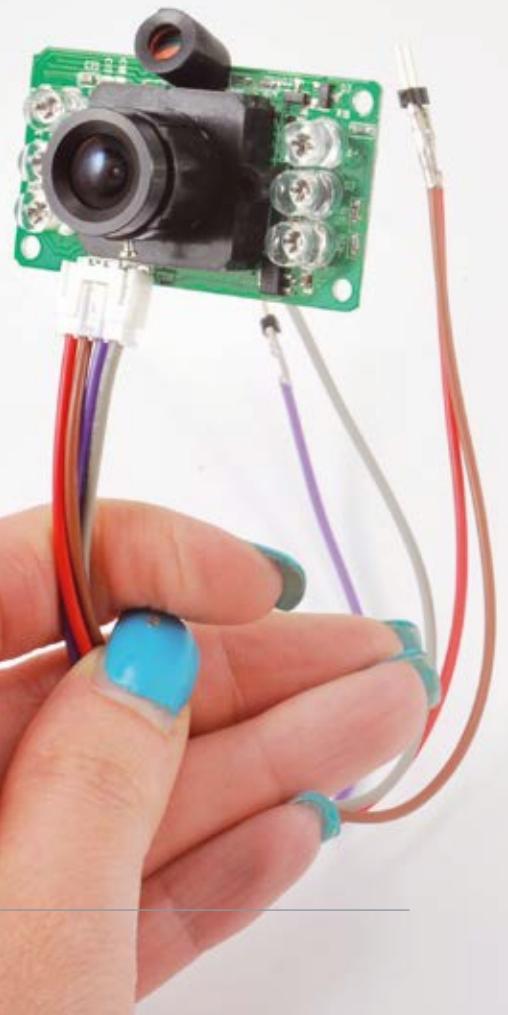
SKETCH ESEMPIO 2

Il secondo esempio, a differenza del primo gestisce anche un sensore PIR e pertanto può essere utilizzato come demo per la realizzazione di un sistema di anti-intrusione.

L'algoritmo di funzionamento è il seguente: ciclicamente Arduino salva delle immagini (max. 5 – *image01.jpg*, ..., *image05.jpg* – con sovrascrittura continua delle più

vecchie) in cartelle (*DIR000*, ..., *DIR999*) all'interno della SD; quando si rileva un allarme il ciclo di salvataggio continua ma cambia il nome delle immagini che vengono salvate come *image06.jpg*, ..., *image10.jpg* ed in seguito si incrementa il nome della cartella e il tutto riprende. Le prime righe di questo software sono uguali al precedente (inclusione di librerie, definizione del blocchetto di salvataggio e pin Arduino e dichiarazione di alcune variabili).

La funzione *setup* è simile a quella del **Listato 1**, ma in più viene identificata sulla SD la prima directory libera (il numero della cartella è indicata dalla variabile *dirCount*). Anche il funzionamento della *loop* è abbastanza simile al precedente esempio; le parti differenti sono la gestione della cartella e del nome del file



di destinazione dell'immagine. In più, alla fine della funzione viene verificato se è presente un segnale di allarme in arrivo dal sensore PIR e di conseguenza viene gestito tutto il "meccanismo" di conteggio di directory e nome del file. Lo sketch dell'applicazione è quello contenuto nel **Listato 2**.

REALIZZAZIONE PRATICA

Il progetto presentato in queste pagine richiede una scheda Arduino Uno, uno shield SD-Card, una camera seriale con illuminatore all'infrarosso e un sensore PIR. Lo shield SD-Card è un prodotto che si trova in commercio in scatola di montaggio (si può ordinare su www.futurashop.it) e per la sua realizzazione si rimanda alle relative istruzioni fornite dal produttore; i componenti da saldare sono quelli in formato tradizionale (ponticelli e connettori maschio-femmina Arduino) in quanto i componenti SMD (in particolare il connettore SD) sono già montati sullo stampato dello shield. È necessario porre particolare attenzione al collegamento tra Arduino e la camera seriale JPEG, nonché con il sensore PIR. La telecamera seriale è munita di un connettore a quattro poli

e viene fornita con un cavetto di collegamento (con fili di colore grigio, viola, marrone e rosso). Per il collegamento ad Arduino è necessario tagliare un lato del cavetto e sui quattro fili saldare dei pin maschio che siano in grado di entrare nei pin femmina di espansione di Arduino; in alternativa, si possono usare dei pin-strip da introdurre nei terminali femmina dei fili del cavetto della JPEG camera, i quali -essendo maschio/maschio- permettono di introdurre nei connettori di espansione di Arduino (o dello shield) i fili del cavetto senza effettuare saldature.

In particolare, il filo marrone andrà inserito in D2, il rosso in D3, il viola in GND e il grigio in +5V (per maggiore chiarezza, riferitevi alle immagini in queste pagine).

Quanto al sensore PIR, viene fornito con un connettore a tre poli maschio formato DIP passo 2,54 mm in cui sono presenti VCC, OUT e GND; per il collegamento ad Arduino è necessario crearsi un opportuno cavetto a tre poli, ovvero saldare tre fili ai rispettivi contatti del connettore. Il segnale OUT del sensore andrà collegato al pin D5 di Arduino mentre invece per VCC (+5V) e GND faccia-



Il materiale utilizzato in questo progetto può essere acquistato presso Futura Elettronica. La board Arduino Uno REV3 (Cod. ARDUINO-UNOREV3) costa 24,50 Euro, la telecamera a colori JPEG con interfaccia seriale (Cod. JPEGIRCAMERA) è disponibile a 69,00 Euro, il sensore PIR (Cod. PIRMOD) a 5,00 Euro e la SD-Card shield per Arduino (Cod. SDCARDSHIELD) è in vendita a 12,00 Euro.

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

mo notare che quelli disponibili sui connettori di espansione sono già occupati dalla camera JPEG; pertanto suggeriamo di creare un unico cavetto per camera e sensore PIR (in cui unire i due segnali VCC e GND) oppure in qualche modo di creare uno sdoppiamento di questi pin di Arduino. Ad esempio, potete sdoppiare il filo del 5V subito dopo l'inserzione in Arduino; per la massa ciò non è necessario, in quanto Arduino dispone, nei connettori di espansione, di due GND. ■



Arduino

la piattaforma open source alla portata di tutti

STARTER KIT V5 CON ARDUINO UNO REV3

Set contenente tutti i componenti necessari per sviluppare applicazioni con ARDUINO UNO REV3. Il kit comprende: scheda ARDUINO UNO Rev 3, cavo USB 2.0, breadboard 400 contatti, motore elettrico miniaturizzato, display LCD 2x16, fotoresistenza, MOSFET, LED colorati e molti altri componenti necessari per cominciare ad utilizzare questa potente piattaforma hardware.



cod:
ARDUKITV5

€ 59,00

IVA inclusa.

FUTURA ELETTRONICA

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

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

Il miglior risultato nella stampa di oggetti in cioccolato si ottiene raffreddando rapidamente gli strati depositi mediante un getto di aria fredda; ecco il dispositivo che permette di farlo.

RAFFREDDATORE PER 3DRAG CHOCO

di GABRIELE DAGHETTA

A parte il mantenimento della temperatura al di sotto di quella prestabilita (necessario a evitare di perdere la tempra) per ottenere in fretta stampe non deformate bisogna affrontare il problema del tempo di raffreddamento del cioccolato, che influisce sulla velocità di stampa: infatti deporre uno strato di cioccolato fuso e quindi caldo, su uno strato sottostante non ancora solidificato, può deformare l'oggetto in stampa; pertanto l'applicazione di uno strato deve essere effettuata quando quello sotto-

stante è sufficientemente freddo da resistere sia al peso del cioccolato soprastante, sia al calore apportato dal cioccolato fuso deposto.

Diciamo che lavorando con una quantità relativamente limitata come quella che può uscire da un ago con diametro interno di 0,9 mm (quale quello adottato nelle nostre prove con la 3Drag Choco) si può andare abbastanza veloce, ma ognuno dovrà, in base alla propria esperienza, scegliere la velocità di stampa appropriata, sfruttando adeguatamente

La 3DRAG CHOCO

Proposto nel fascicolo n° 189, il kit di modifica consiste in una grande siringa per uso medico dotata di un ago modificato, montata su una staffa da applicare alla barra orizzontale della 3Drag al posto della testa di stampa originale; il pistone della siringa è azionato da un meccanismo a vite senza fine messa in rotazione da un motore passo-passo, che la scheda controller gestisce al posto dello stepper-motor di alimentazione della testa originale. Il sistema che usiamo per deporre il cioccolato, sotto forma di filo fuso, consiste nel riscaldare il corpo della siringa ed ancor più l'ago mediante un riscaldatore elettrico a resistenza: l'ago è inserito in un cilindro (camicia) forato per farcelo passare e per ospitare anche l'innesto, capace di portare il tutto alla temperatura richiesta e mantenercelo. In questo modo il cioccolato che si trova nella testa (zona di uscita dell'iniettato) della siringa si fonde e può colare. Affinché ciò avvenga, bisogna spingere lo stantuffo della siringa stessa e farlo con regolarità: qui entra in gioco l'azionamento dell'alimentatore, che consiste in una vite senza fine che spinge verso il basso lo stantuffo stesso.

La cosa riesce meglio se tutto il cioccolato nella siringa è "ammollato": per questo l'intero corpo della siringa è introdotto in un cilindro in alluminio il cui duplice scopo è sostenere la siringa stessa e portarle il calore prodotto dal riscaldatore, che è posto alla base ed è lo stesso che riscalda la camicia che circonda l'ago. Malgrado l'alluminio sia buon conduttore di calore, il caldo prodotto dal riscaldatore alla parete esterna della siringa rimane sufficiente a mantenere molle il cioccolato.

La siringa può essere caricata in due modi: colando all'interno della siringa cioccolato già fuso a bassa

temperatura, quindi inserendola nell'apposito cilindro; inserendo nella siringa scaglie di cioccolato che poi fonderanno grazie all'azione del riscaldatore, una volta inserita la siringa nel cilindro. Per consentire alla 3Drag di stampare oggetti in cioccolato, bisogna innanzitutto eliminare dalla stampante il piatto riscaldato, che ovviamente qui non serve, dato che il cioccolato deve solidificare e possibilmente alla svelta. Poi occorre montare un estrusore speciale (da sostituire a quello utilizzato per estrarre materiali plastici) che abbiamo realizzato e appositamente dotato di una comunissima siringa da 60 ml, il cui pistone viene azionato mediante un motore passo-passo di tipo NEMA17, oltre che di un riscaldatore cilindrico per mantenere ad una temperatura di circa $32 \div 33^\circ\text{C}$ il cioccolato contenuto nella siringa.

Il cioccolato, per poter essere estruso attraverso l'ago della siringa (avente un diametro interno di soli 0,9 mm) deve essere sciolto; ciò si ottiene attivando opportunamente il riscaldatore in cui è inserita



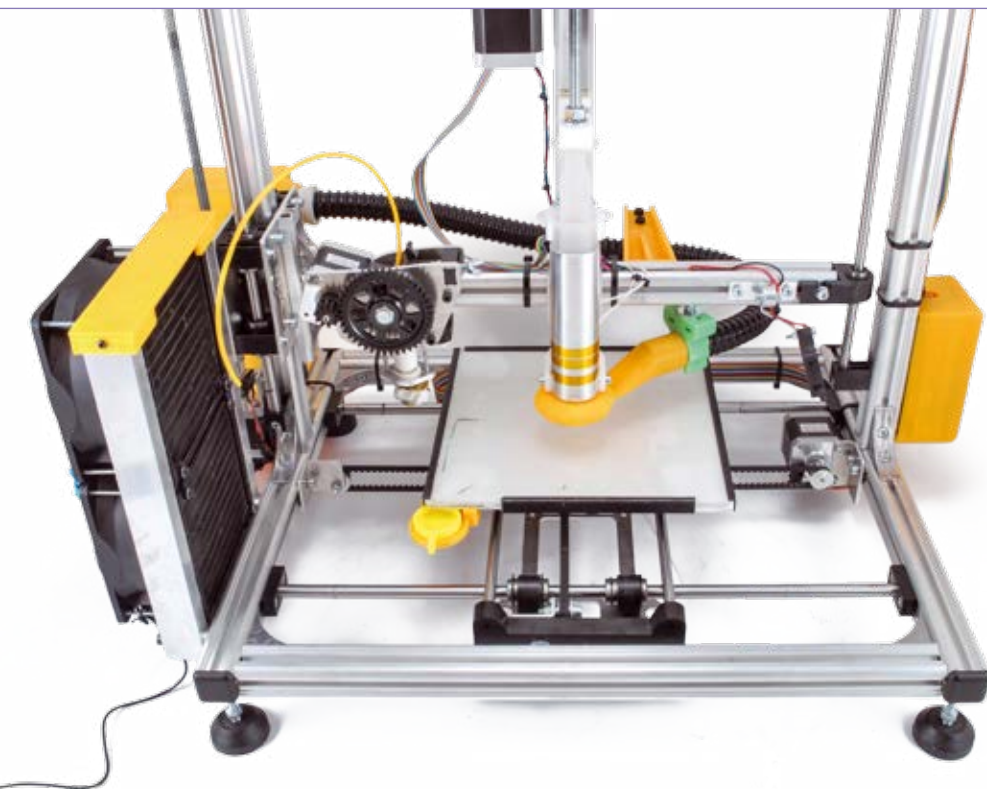
la siringa, ma senza superare la soglia dei 33°C , onde evitare di far perdere la "tempra" al cioccolato.

La perdita di "tempra" del cioccolato impedirebbe di stampare oggetti che si sviluppano in altezza, quindi il cioccolato stesso non riuscirebbe più a solidificare a temperatura ambiente ($24 \div 27^\circ\text{C}$).

le funzioni e le impostazioni del client di stampa (Repetier-Host, Cura ecc.) come riterrà opportuno. Per raffreddare più in fretta il cioccolato (a tutto vantaggio della velocità di stampa) abbiamo tentato varie soluzioni, tutte provate in laboratorio: la prima è stata applicare al posto del piatto riscaldato una piastra in alluminio, di pari dimensioni, raffreddata mediante celle di Peltier, a loro volta ventilate sul lato caldo mediante dissipatori di calore e ventole. Questa soluzione si è dimostrata però adatta solo a realizzazioni poco spesse, come le scritte in cioccolato, perché non consente di raffreddare gli strati alti. Inoltre l'insieme delle celle di Peltier e dei dissipatori pesa e causa una certa inerzia che ostacolerebbe il movimento del piatto, quando si stampa ad alta velocità.

Abbiamo quindi ipotizzato il raffreddamento diretto del cioccolato deposto, inizialmente con un sistema a "tubi raffreddatori" basato su aria compressa, che consente la refrigerazione rapida localizzata ma richiede l'abbinamento a un compressore d'aria, che è ingombrante e rumoroso e che non rappresenta certo il migliore abbinamento a una macchina da cucina e da pasticceria.

A questo punto abbiamo deciso di orientarci nuovamente verso le celle di Peltier, per realizzare un sistema fisso e compatto di generazione del freddo col quale raffreddare il flusso d'aria diretto da una ventola direttamente contro la zona in cui viene deposto il cioccolato fuso; questa soluzione si è dimostrata la migliore, sebbene, malgrado le dimensioni ridotte della bocca che soffia l'aria fredda, in qualche modo il flusso investe l'ugello contenente l'ago tanto da raffreddarlo abbastanza da impedire il riscaldamento del cioccolato alla temperatura



necessaria a fondere. Il problema è stato risolto convogliando il getto d'aria fredda sul cioccolato, grazie a un'attenta progettazione del diffusore che permette di evitare che l'aria fredda non interessi l'ugello dell'estrusore ma vada direttamente in basso sul cioccolato depositato.

In questo articolo descriviamo il sistema di raffreddamento istantaneo, la cui utilità si apprezza quando si vuol stampare molto veloce, ma che comunque serve anche a bassa velocità.

IL RAFFREDDATORE

Questa "macchina" è insieme un impianto frigorifero e un convogliatore d'aria refrigerata; l'impianto frigorifero è l'unità che produce il freddo ed è basata su sei celle di Peltier da 12V - 51 W, montate su radiatori di calore dal lato freddo e raffreddati sul lato caldo grazie a un sistema di refrigerazione a circolazione d'acqua. Sul dissipatore del lato freddo, opportunamente racchiuso in un condotto, viene soffiata -tramite un'elettroventola- dell'aria prelevata dall'ambiente, la quale lam-

bendo le alette si raffredda e poi giunge, tramite un apposito tubo convogliatore, sulla zona in cui viene depositato il materiale (ovvero sotto l'ugello dell'estrusore).

Per la generazione del freddo abbiamo optato per le celle di Peltier in quanto, seppure non abbiano una grande efficienza (un sistema frigorifero a ciclo di Carnot basato su pompa di calore avrebbe la stessa potenza frigorifera con un consumo di elettricità nettamente minore) presentano il vantaggio di poter realizzare impianti allo stato solido e piuttosto compatti, di facile gestione da parte di una scheda elettronica.

Per comprendere il funzionamento del sistema che abbiamo realiz-

zato bisogna spiegare -a chi non lo sa- cosa sono le celle di Peltier: questi dispositivi si basano sull'effetto Seebeck, che si verifica nelle giunzioni tra due diversi metalli che hanno differente struttura atomica; senza scendere nella Fisica, si può dire che la diversa disposizione degli elettroni di valenza di metalli differenti causa, quando i due sono posti a contatto, una differenza di potenziale. Questa tensione, che è molto piccola (dell'ordine dei mV) si misura tra i due metalli e varia in funzione della temperatura cui la giunzione è sottoposta, nel senso che cresce all'aumentare del calore e cala al diminuire della temperatura; questa dipendenza si spiega col fatto che il riscaldamento aumenta l'agitazione termica degli atomi che costituiscono i metalli e quindi il potenziale posseduto dagli elettroni periferici.

Realizzando una giunzione tra due metalli differenti, disposti alternativamente (cioè un metallo tra due pezzi di metallo differente) si ottiene la cosiddetta "pila termoelettrica": scaldando una giunzione e raffreddando l'altra, viene generata una differenza di potenziale direttamente proporzionale alla temperatura e, chiudendo gli estremi su un utilizzatore elettrico, in questo scorre elettricità.

La pila termoelettrica può funzionare anche al contrario: alimentata con una corrente elettrica, una

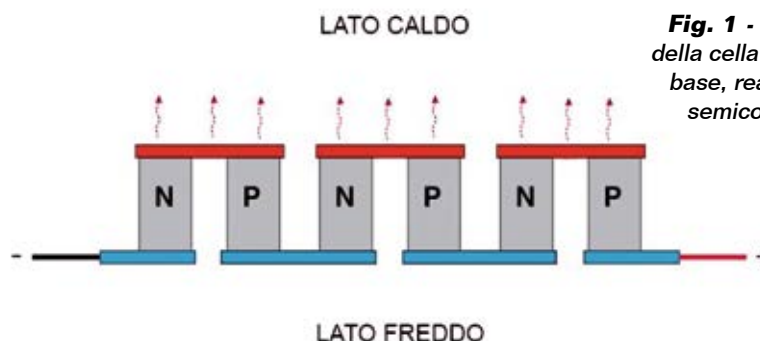
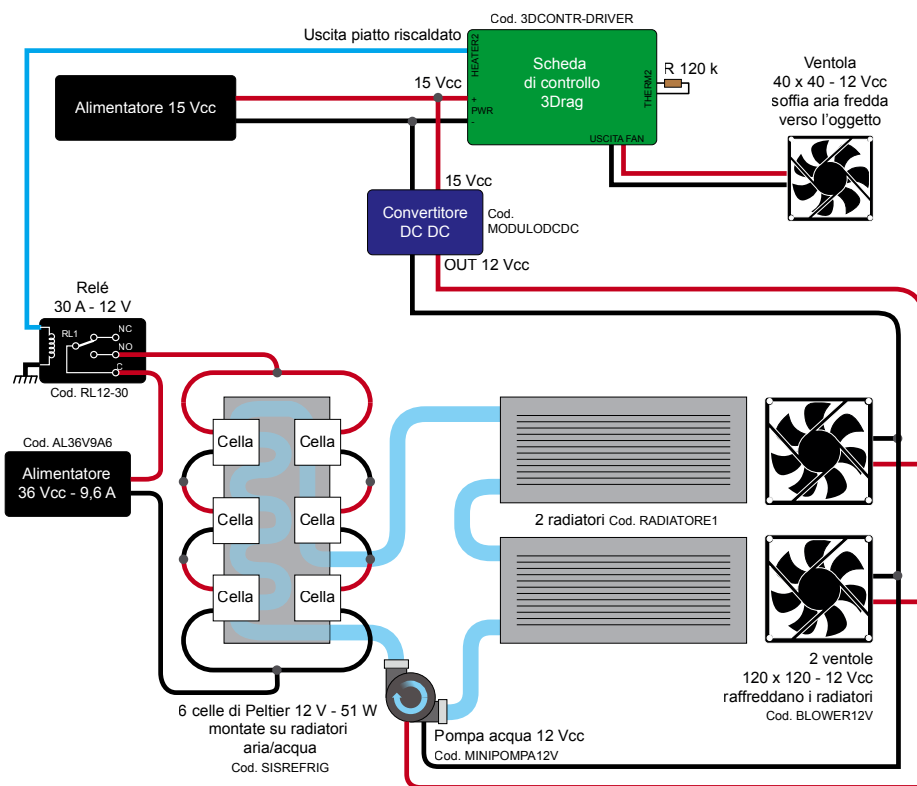


Fig. 1 - Struttura della cella di Peltier base, realizzata in semiconduttore.



Schema elettrico e idraulico del sistema che genera il freddo e spinge l'aria raffreddata verso il cioccolato depresso.

giunzione si riscalda e l'altra si raffredda; questo fenomeno si deve all'effetto Peltier e spiega il funzionamento della cella di Peltier. Le celle si possono costruire con i metalli, tuttavia l'elevata conducibilità termica di tali materiali porterebbe ad avere minime differenze di temperatura tra i lati caldo e freddo; per questa ragione si ricorre ai semiconduttori, che avendo una resistenza termica molto maggiore- consentono di isolare meglio le due giunzioni e incrementare il salto termico. Affinché si possa usare per costruire le celle di Peltier, il silicio deve essere drogato P ed N: per l'esattezza, la cella elementare è una struttura formata da un blocchetto N saldato ad uno P e quest'ultimo è saldato a un nuovo blocco di tipo N; l'insieme è realizzato in modo che i lati in comune delle zone P ed N siano "aperti", vale a dire uniti da una lamina di materiale conduttore. La struttura PNP è quindi composta come in Fig. 1.

La figura mostra anche quale lato diventa caldo e quale freddo in base al verso della corrente, perché la caratteristica della cella di Peltier è che i lati caldo e freddo si invertono cambiando la polarità della tensione di alimentazione e quindi il verso della corrente. Naturalmente il produttore indica qual è il lato caldo e quello freddo se si alimenta la cella con il positivo sul filo + (di solito è il rosso) e il negativo sul - (di solito è il nero). In qualche modello di cella di Peltier il lato caldo e quello freddo vanno rispettati, perché i



Fig. 2 - Il blocco assemblato con le celle di Peltier e le serpentine.

materiali che rivestono i due sono differenti.

Di norma i due lati esposti delle giunzioni di silicio sono ricoperti da lamine di allumina, materiale che costituisce un ottimo isolante elettrico, ma che possiede grande conducibilità termica.

Le celle di Peltier vengono caratterizzate, oltre che per potenza elettrica assorbibile, per salto termico raggiungibile: il salto termico è la differenza di temperatura che si può verificare tra i lati caldo e freddo.

La potenza frigorifera della cella è sempre inferiore alla potenza elettrica assorbita, perché una buona parte viene dissipata sotto forma di calore; per questo le celle di Peltier sono convenienti solo dove il basso rendimento energetico sia trascurabile rispetto a vantaggi come la compattezza e la rapidità di raggiungimento della temperatura richiesta.

Detto ciò, vediamo come è strutturato il raffreddatore.

COMPOSIZIONE DEL SISTEMA

Il gruppo frigorifero è formato da due dissipatori di calore alettati su un lato e piani da quello opposto, disposti affacciati per le alette e racchiusi lateralmente in modo che l'aria che vi passa non esca e non entri (sono fissati tra loro mediante bulloncini e

Fig. 3 - Il blocco dissipatori in alluminio/ celle di Peltier e serpentine è assemblato con bulloncini avvitati sui dissipatori.



colonnine filettate); sulle parti piane sono montate per ciascun dissipatore sei celle di Peltier da 51 W, collegate tra loro in serie come mostra lo schema in queste pagine. I due blocchi da tre celle sono tra loro connessi in parallelo, in modo da alimentare il tutto con un alimentatore capace di erogare 36 Vcc. Abbiamo optato per la connessione serie-parallelo in quanto è una via di mezzo che consente di utilizzare un alimentatore a tensione d'uscita reperibile in commercio con una corrente che non sia troppo elevata, così da limitare le perdite di potenza nei cavi di collegamento; le tre celle in serie di ogni lato, essendo ognuna da 12V nominali richiedono 36V e i due gruppi in parallelo si alimentano con tale tensione. Ogni fila assorbe poco più di 4 ampere, quindi i 9,6 ampere erogati dal nostro alimentatore a 36 V sono più che sufficienti.

Le celle di Peltier sono fissate con il lato freddo alle parti piane dei radiatori di alluminio (mediante collari metallici a vite) mentre sui loro lati caldi si appoggiano due serpentine per liquidi realizzate anch'esse in alluminio (Fig. 2 e Fig. 3): queste servono a far circolare acqua in modo che il calore prodotto dalle celle venga asportato con maggiore facilità e che i lati freddi raggiungano le temperature più basse possibili. Ogni serpentina si presenta esternamente come un parallelepipedo con le facce più estese ben piane e levigate (proprio per aderire meglio alle celle di Peltier) e due raccordi tubolari a punta di freccia conica per introdurre i tubi in gomma.

Per agevolare il trasferimento del calore, tra i lati freddi e i dissipatori e tra quelli caldi e le superfici delle serpentine, bisogna interporre della pasta al silicone per elettronica.

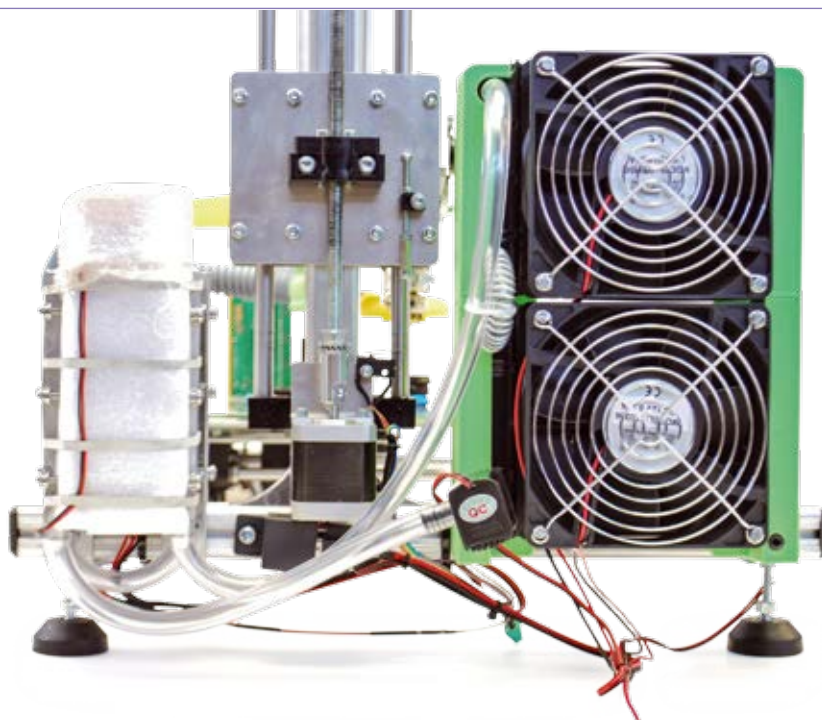


Fig. 4 - Gruppo di raffreddamento ad acqua: gli scambiatori acqua/aria sono in serie, come le serpentine montate sul corpo refrigerante.

Nel nostro progetto, dato che la scelta non crea particolare criticità, abbiamo posto le serpentine idraulicamente e termicamente in serie, nel senso che una riceve il tubo in arrivo dallo scambiatore acqua/aria e con uno spezzone di tubo a "U" l'acqua in uscita entra nell'altra, la cui uscita finisce sul tubo che riporta al radiatore aria/acqua (Fig. 4).

Questo scambiatore è a sua volta formato da due corpi indipendenti collegati in serie a loro volta: un tubo unisce l'estremità superiore di quello in basso e quella inferiore del corpo in alto, mentre i due tubi di gomma alle estremità raggiungono gli estremi liberi delle serpentine. Il circuito in serie, se da un lato ha come difetto il fatto che l'acqua già scaldata dai lati caldi delle celle di Peltier che ricevono l'acqua dagli scambiatori entra nella serpentina di ritorno allo scambiatore stesso, dall'altro mette al riparo dagli inconvenienti causati dalla formazione di eventuali bolle d'aria nel circuito idraulico, che si presenterebbero nel caso di circuito in parallelo. Ogni corpo del radiatore è raffreddato da un'elettroventola a 12 Vcc da 120x120 mm, grande

abbastanza da smaltire il calore anche in estate.

Notate che lo scambiatore acqua/aria può anche essere unico. In tutti i casi, al tubo in basso si deve applicare l'elettropompa a 12V, che è del tipo MINIPOMPA12V commercializzato dalla Futura Elettronica (www.futurashop.it); si tratta di un dispositivo di piccole dimensioni e molto leggero, capace però di una portata di 240 litri/ora, di spingere acqua fino a un'altezza di 3 metri e protetta, sul lato elettrico, fino al grado IP68. L'assorbimento è di 400 mA. Il blocco con i dissipatori in alluminio e le celle di Peltier va rivestito lateralmente (ma anche tutto) con del materiale isolante, come dell'espanso o della spugna, in modo da disperdere meno calore possibile. Una buona soluzione è anche il cosiddetto Depron (lo spessore consigliato è 6 mm) che è un espanso usato come base isolante termoacustica per le tappezzerie e la cui leggerezza e



Fig. 5

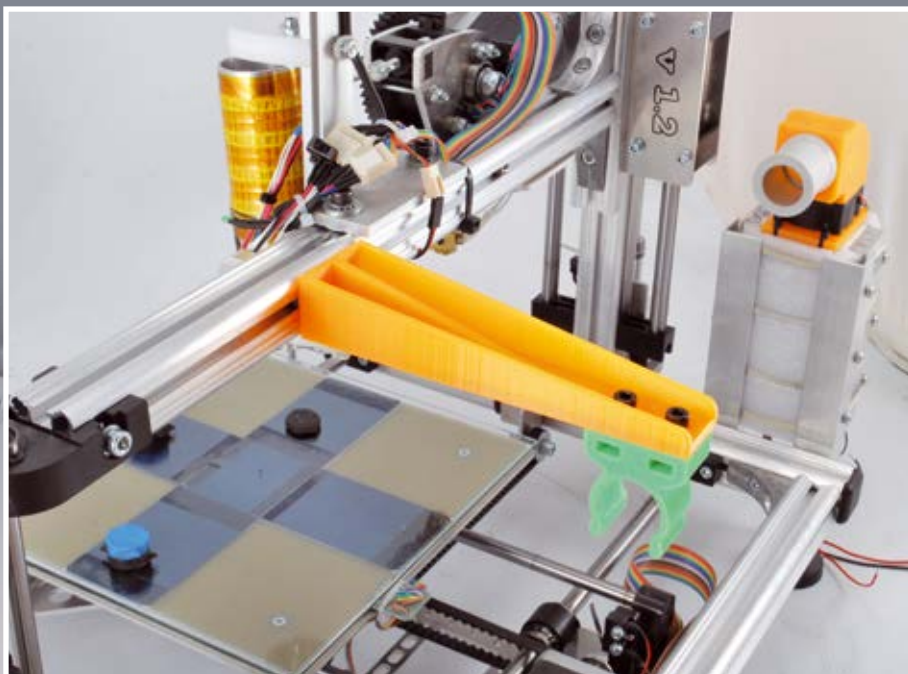


Fig. 7 - Braccio di sostegno del tubo corrugato, fissato alla barra orizzontale porta-testa della 3Drag.



Fig. 6 - Blocco celle Peltier con in cima ventola e convogliatore, più particolari di raccordo.

sagomabilità sono da anni apprezzati nell'aeromodellismo.

Affinché la macchina funzioni come si deve, il predetto corpo va disposto in verticale, con in cima la ventola da 40x40 mm che aspira l'aria raffreddata e la invia al convogliatore in plastica sul quale poi si applica il tubo corrugato in gomma che la porta al "getto" posto intorno alla testa di stampa della 3Drag Chocolate. La disposizione in verticale permette di far colare la condensa che si forma inevitabilmente a causa dell'umidità dell'aria quando la macchina refrigerante va in scongelamento; questa fase viene gestita dall'elettronica che descriveremo tra breve; la precipitazione della condensa è la ragione per cui la ventola è disposta

superiormente e non in basso (si bagnerebbe...). La ventola deve immettere l'aria in un convogliatore in plastica stampabile in 3D con la 3Drag seguendo i file .STL scaricabili dal nostro sito www.elettronica.in.it insieme agli altri file del progetto; il convogliatore va applicato alla sommità della ventola 40x40, accoppiata con l'apposito raccordo adattatore in plastica (anche questo stampabile in PLA con la 3Drag partendo dagli STL che forniamo) e terminato con un raccordo per tubi da impianti elettrici introdotto e avvitato (con l'apposita ghiera) nella parete terminale, che è già forata allo scopo. Il raccordo, che è da 25 mm e a tenuta d'acqua (IP66) permette di applicare il tubo corrugato da 25 mm senza perdite. Il convogliatore completo di ventola e raccordo appare come in Fig. 5. L'insieme del raffreddatore e del convogliatore si presenta, completamente assemblato, come in Fig. 6. Siccome tutte le ventole sono a 12 volt, vengono alimentate in parallelo da un convertitore DC/DC che fornisce tale tensione

in uscita. Il converter è alimentato dall'uscita dello stesso alimentatore a 15 Vcc che fa funzionare l'elettronica di governo del sistema. Il tubo corrugato da utilizzare per portare l'aria fredda convogliata direttamente verso l'estrusore è abbastanza lungo e -considerata la sua rigidità- deve fare un'ansa; va quindi sorretto mediante un braccio montato sul profilato orizzontale (lo stesso che regge la testa di stampa) della 3Drag ed una clip fissata ad esso. Entrambe queste parti si stampano in PLA con la 3Drag, partendo dai file .STL che forniamo. Il braccio si fissa alla barra in alluminio porta-testa mediante dadi quadrati fatti scorrere nelle scanalature del profilato, quindi serrando con i

bulloni previsti. La clip si avvitata all'estremità opposta del braccio con bulloni 5MA e dadi (sono previste apposite cave nella base della clip). L'insieme appare come in Fig. 7. Realizzato l'aggancio, bisogna intestare un'estremità del tubo corrugato con l'apposito diffusore di aria fredda, visibile in Fig. 8, che è un manicotto (stampabile anch'esso in PLA con la 3Drag) terminante con una ciambella aperta nella parte inferiore in modo da far uscire l'aria inferiormente affinché finisca sul materiale deposto; questo accorgimento, unito al fatto che il diffusore va montato in modo che nell'interno della ciambella penetri totalmente l'ugello dell'estrusore (Fig. 9), ci risparmia di dover



Fig. 8 - Corrugato con applicato il diffusore a ciambella.



Fig. 9 - Dettaglio del diffusore visto dal basso: si nota l'apertura inferiore.

coibentare l'esterno di quest'ultimo. Una volta applicata l'altra estremità del tubo corrugato al convogliatore situato sul gruppo raffreddatore, la macchina si deve presentare come mostrato nella Fig. 10.

L'UNITÀ DI CONTROLLO

Passiamo adesso all'elettronica che governa il raffreddatore, la quale non è altro che il controller della stampante 3Drag v1.2, del quale utilizziamo l'uscita per il piatto riscaldato (il piatto qui non serve in quanto non c'è nulla da scaldare, ma casomai il pezzo stampato va raffreddato...) per alimentare, tramite un relé

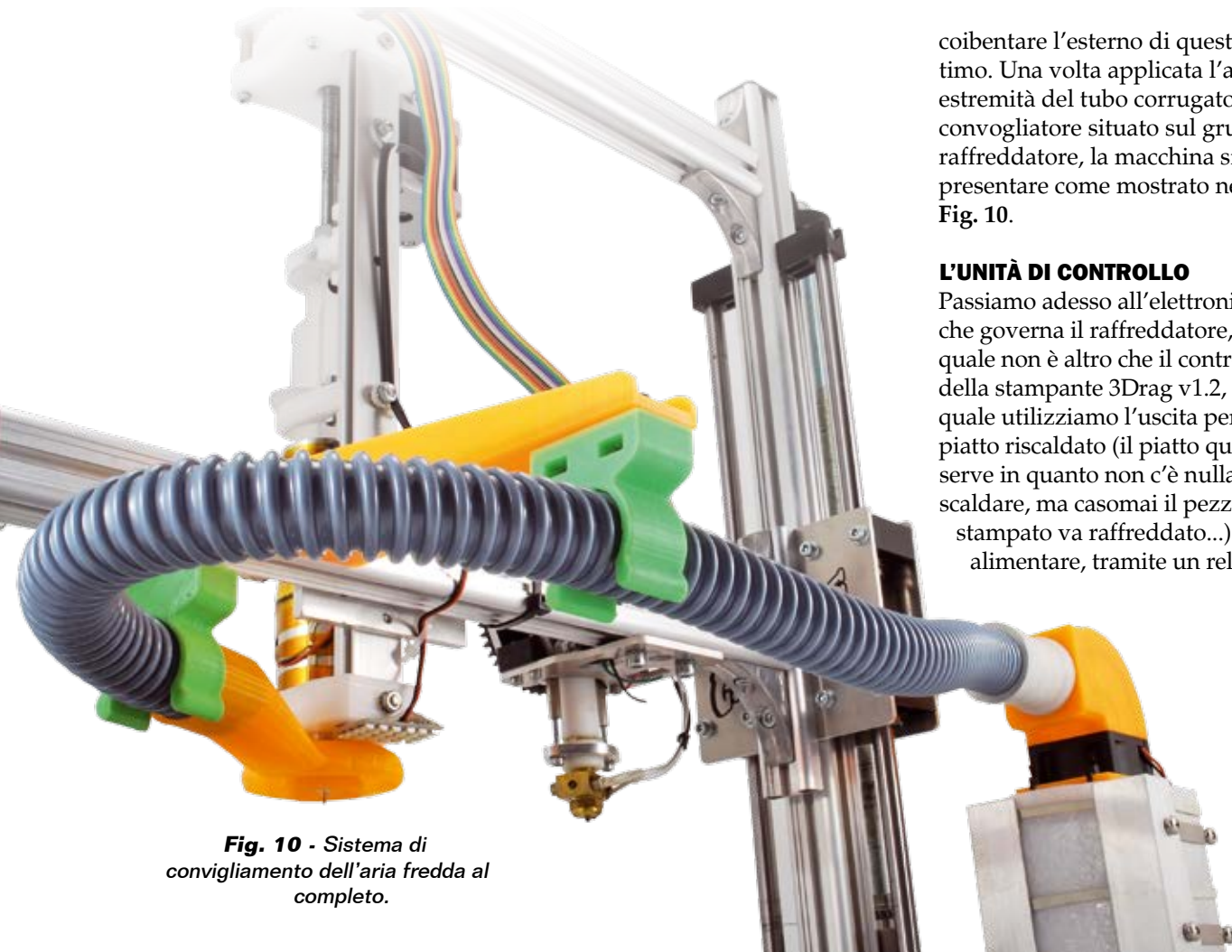


Fig. 10 - Sistema di convogliamento dell'aria fredda al completo.



per il MATERIALE

Il sistema refrigerante a celle di Peltier per la stampa a cioccolato (cod. SISREFRIG) è disponibile al prezzo di Euro 115,00. Il radiatore ad acqua (cod. RADIATORE1) costa Euro 29,00. La mini pompa ad immersione (cod. MINIPOMPA12V) è disponibile a Euro 19,90. La Ventola 120X120/12Vdc (cod. BLOWER12V) è in vendita a Euro 9,50. Il convertitore DC/DC 1,5-35V (cod. MODULODCDC) a Euro 4,30. Il Relè di potenza 12V/30A (cod. 5230-RL12-30) costa Euro 3,50. L'alimentatore switching 36Vdc/9,6A (cod. AL36V9A6) è in vendita a Euro 47,00. Il set di modifica della 3Drag per la stampa a cioccolato (cod. 3DCHOCO) è disponibile al prezzo di Euro 178,00. 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>

monoscambio 12V-30A, il circuito delle celle di Peltier. Usiamo anche l'uscita FAN (da gestire mediante l'apposito comando nel software client di stampa) per comandare la ventola che soffia l'aria fredda. Naturalmente per effettuare il controllo del caso è necessario che nel microcontrollore ATmega2560 della scheda controller sia installato l'apposito firmware modificato che trovate, insieme ai file del progetto, nel nostro sito web www.elettronica.in.it.

Il firmware non gestisce la temperatura dell'aria fredda, ma, per evitare che la condensa dovuta all'umidità dell'aria ghiacciando formi uno strato di brina tale da ostruire il condotto formato tra le alette dei radiatori in alluminio, periodicamente bisogna spegnere

le celle di Peltier manualmente (usando l'apposito comando manuale del piatto riscaldato di Repetier-Host, per esempio, ovvero quello di Cura) per consentire lo scioglimento del ghiaccio (è in questa fase che può verificarsi il gocciolamento dalla parte inferiore del blocco raffreddatore) formatosi nei radiatori. Il periodo di spegnimento potete sceglierlo sulla base della vostra esperienza e in funzione dell'umidità dell'aria (più l'aria è umida, più frequentemente occorre sbrinare e viceversa); chiaramente all'inizio dovete fare qualche prova per prenderci confidenza.

Siccome il firmware non deve gestire la temperatura del piatto riscaldato e quindi leggere l'NTC previsto in origine, lo "inganniamo" collegando un resistore da 120 kohm $\frac{1}{4}$ di watt ai contatti previsti per il termistore. Il cablaggio della macchina è quello della 3Drag e considerato

che la modifica 3Drag Chocolate si applica a una macchina già costruita, non stiamo a ripeterlo; quello che vi interessa è il cablaggio delle varie parti elettriche e dell'uscita del controller, per i quali vale il solito schema visibile in queste pagine. Per la scheda controller e il DC/DC converter abbiamo previsto dei contenitori in PLA (Fig. 11), stampabili con la 3Drag (anche per questi particolari sono disponibili i file .STL); avendo chiuso il controller in una scatola, per consentirne la ventilazione richiesta dal riscaldamento dei moduli driver, sono previste delle feritoie, nonché una ventolina 40x40 mm opzionale da alimentare con i soliti 12V, stavolta prelevati dal controller stesso. I due contenitori vanno fissati a uno dei profilati laterali della stampante con delle viti (la cui testa va fatta scorrere nelle scanalature dei profilati) provviste di distanziali e di dadi.

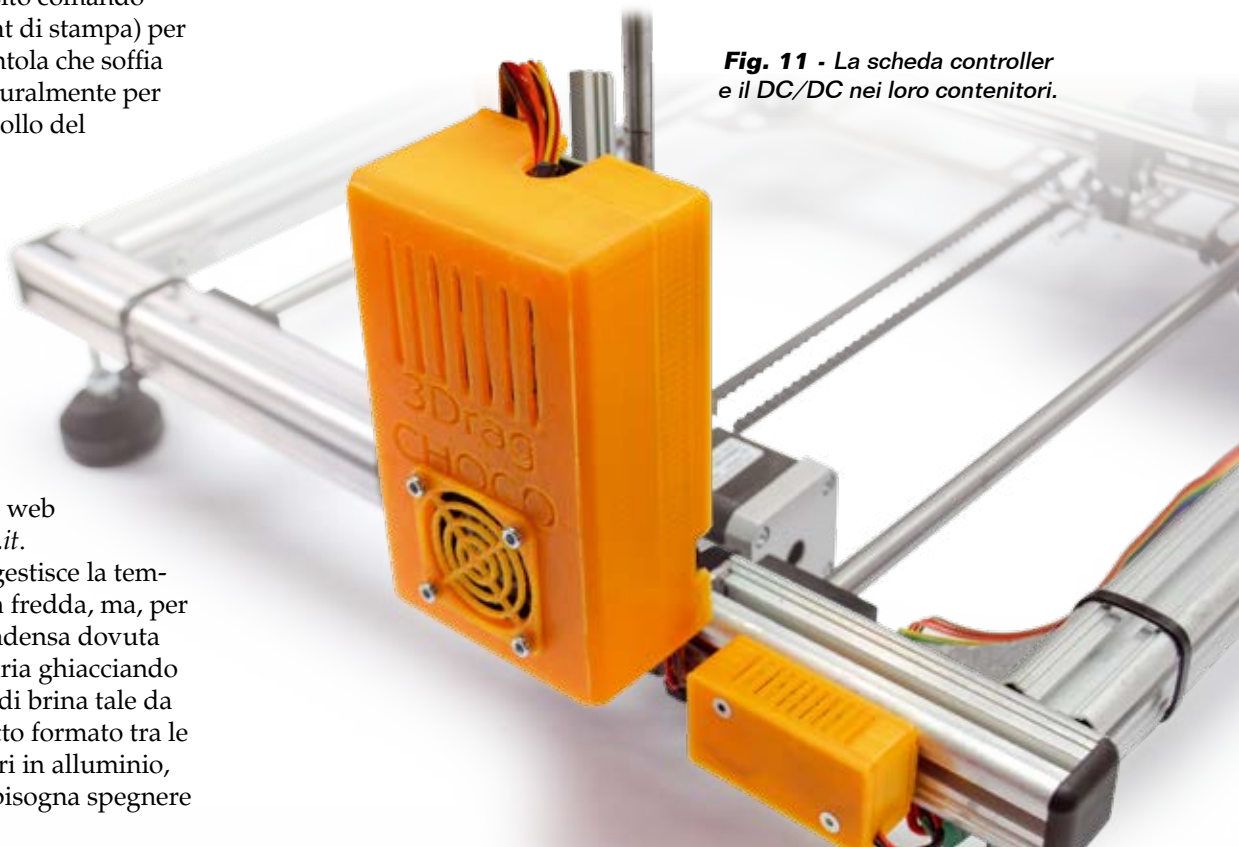


Fig. 11 - La scheda controller e il DC/DC nei loro contenitori.

OPENMOTORCONTROL

IL CONTROLLER PER TUTTI

Open source e basato su un microcontrollore ATmega32U4, è dotato di driver per due motori CC a spazzole o uno stepper motor. Può ricevere comandi da USB, seriale o I²C bus.



..... dell'Ing. MIRCO SEGATELLO

Per chi fa robotica, una delle problematiche da risolvere è la gestione dei motori per la trazione, ovvero come alimentare correttamente i motori per far avanzare il proprio robot. Se lavorate con Arduino, la prima e immediata soluzione è l'utilizzo di uno shield; in commercio se ne trovano diversi, dai più semplici che permettono il controllo separato di due piccoli motori in corrente continua, ai più evoluti che possono anche misurare la cor-

rente assorbita. Indipendentemente dal produttore, gli shield si basano tutti sull'utilizzo di un driver di potenza (di solito un L298) interfacciato direttamente con le uscite PWM di Arduino e contornato da pochi altri componenti. Sicuramente l'uso di uno shield è una soluzione valida, ma è necessario utilizzare almeno quattro uscite di Arduino: di solito due per regolare la velocità e due per la direzione. Se invece utilizzate un microcontrollore generico

oppure un Atmel stand-alone o una board diversa da Arduino, le cose si complicano un pochino, perché in commercio è difficile reperire driver con un'interfaccia più flessibile ed il costo comincia a lievitare. Se poi avete la necessità di comandare più di due motori, la cosa si complica parecchio, anche per chi utilizza una scheda Arduino, perché sorgono problematiche sia sul piano hardware che su quello della programmazione del dispositivo.

Per venire incontro alle esigenze di chi vuole gestire piccoli motori in corrente continua con logiche programmabili, abbiamo pensato di progettare un driver tutto nostro che permettesse di avere un'elevata flessibilità operativa e soprattutto fosse open source, al fine di permettere a chiunque di poterlo adattare alle proprie esigenze.

Il nome che gli abbiamo dato è significativo della filosofia progettuale: OpenMotorControl, che abbiamo abbreviato in OMC21, dove 21 indica il numero di canali (due in questo caso) e la corrente gestita per ciascun canale (1A).

Come vedremo, si tratta di un circuito stand-alone pilotabile da varie logiche, dotato di un'in-

terfaccia di comunicazione che prevede diversi protocolli di comunicazione ed è adatta alle più svariate esigenze.

SCHEMA ELETTRICO

Tipicamente, nel realizzare un controller per motori in corrente continua si ricorre al cosiddetto ponte ad H, ovvero un circuito formato da quattro transistor montati a ponte, in grado di comandare velocità e direzione del motore attraverso l'erogazione di impulsi PWM e l'inversione della polarità all'uscita.

Un driver del genere si trova in forma monolitica in vari circuiti integrati; solo in casi particolari, come è stato nella scheda di controllo dell'Openwheels (progetto di veicolo autobilanciante pubblicato a partire dal fascicolo n° 172) ha senso progettare un simile driver con componenti discreti. Per la nostra applicazione, un semplice ed economico circuito integrato contenente un ponte ad H è più che sufficiente, anche perché, permette di contenere costi ed ingombri. Il circuito integrato adottato è siglato LV8405 ed è un completo ponte ad H a MOSFET ad elevate prestazioni che implementa anche un controllo sulla tensione di alimentazione e dispone di protezioni sia in temperatura che in corrente. Viene venduto nel package SMD SSOP16 e quindi

non è direttamente utilizzabile in breadboard, ma è necessario sia previsto un apposito PCB. Pur essendo piccolissimo, presenta

caratteristiche di tutto rispetto:

- 2 canali con controllo forward/reverse;
- basso consumo;
- bassa resistenza serie dei MOSFET (0,75 ohm);
- protezioni integrate da bassa tensione e surriscaldamento;
- quattro modi di funzionamento (forward/reverse, brake, stop);
- corrente massima di uscita 1,4A;
- corrente di uscita (picco) di 2,5A.

La bassissima resistenza dei MOSFET usati per il controllo del motore (soli 0,75 ohm) permette un rendimento molto elevato e presenta una dissipazione di calore molto contenuta.

Per realizzare un controllo motori facile da interfacciare e programmare è però necessario abbinare questo IC ad un microcontrollore debitamente programmato: la scelta è caduta sull'ATmega32U4 prodotto dalla Atmel, lo stesso utilizzato sulla scheda Arduino Leonardo. Nella scheda Leonardo, a differenza di Arduino Uno, che usa il microcontrollore ATmega328, viene usato l'ATmega32U4 perché implementa al suo interno anche tutto l'hardware per la comunicazione USB, eliminando la necessità di utilizzare esternamente un convertitore USB/seriale. Le caratteristiche sono in generale migliori dell'ATmega328, essendo un componente di ultima generazione; inoltre il micro è perfettamente compatibile con l'ambiente di sviluppo (IDE) di Arduino. Pochi altri componenti ed ecco disponibile un controllo motori piccolo, economico e dalle caratteristiche interessanti:

- configurazione a doppio ponte H per pilotare due motori in cc o uno stepper motor bipolare;

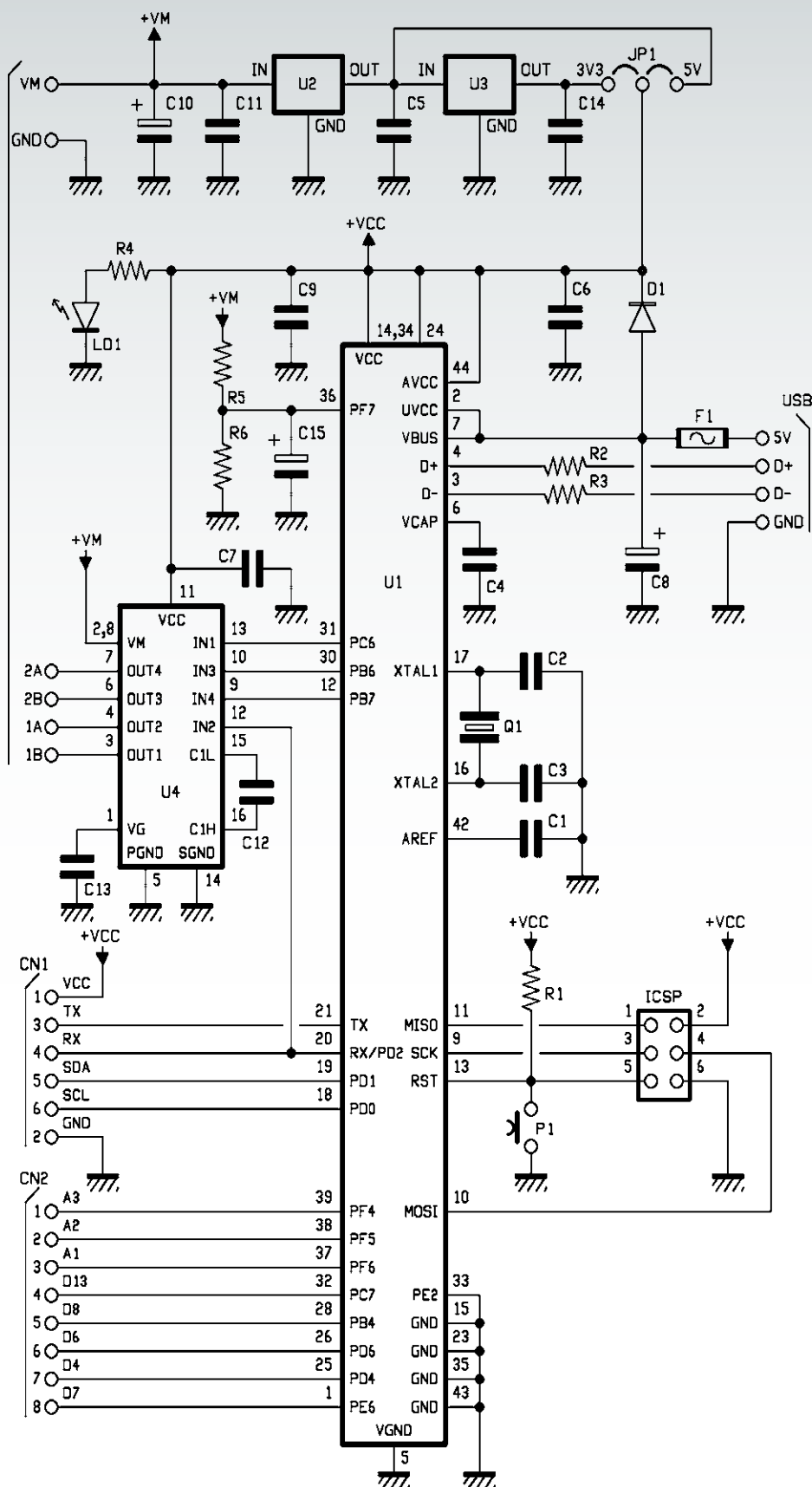


[schema ELETTRICO]

- tensione della sezione di potenza: $3 \div 15V$;
- tensione di alimentazione della logica: $3 \div 5V$;
- corrente di uscita: 1,4A continua (2,5A picco) per motore;
- interfacciamento USB, seriale, I²C-Bus;
- ingressi compatibili 3,3V e 5V;
- alimentazione selezionabile tra interna ed esterna a 3,3V o 5V.

Siccome volevamo un controller molto flessibile, abbiamo curato particolarmente la sezione di alimentazione; volevamo un driver compatibile con logiche a 3,3V e 5V che potesse sia essere alimentato dalla stessa tensione della logica di comando, sia alimentare una logica esterna, così da lavorare con gli stessi livelli di alimentazione.

Nello schema elettrico del circuito (lo trovate qui accanto...) potete vedere che sono stati previsti due stabilizzatori di tensione di tipo low-drop-out (a bassa caduta di tensione ingresso-uscita) utilizzati per ricavare le tensioni di alimentazione a 3,3V (se ne occupa U3) e 5V (se ne occupa U2); la selezione della necessaria tensione di alimentazione avviene unendo con una goccia di stagno le piazzole corrispondenti del jumper JP1. Saldando la piazzola centrale con la 3V3 si imposta la tensione di alimentazione a 5V, mentre collegandola a 3V3 si fissa la tensione a 3,3 volt. Come da specifiche dell'integrato ATmega32U4, il funzionamento a 3,3 volt è garantito solo con un quarzo da 8MHz, al quale va abbinato il corrispondente bootloader. La tensione scelta risulta anche disponibile sul connettore CN1 e può essere utilizzata per alimentare una logica esterna, come una scheda Arduino (si veda la **Tabella 6**). L'alimentazione principale per tutto il circuito



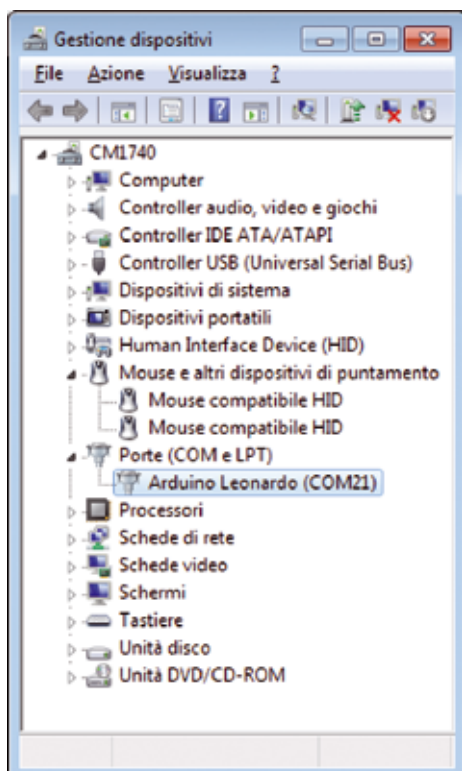


Fig. 1 - L'OMC21 viene rilevato come periferica seriale Arduino Leonardo.

viene prelevata dalla linea di alimentazione dei motori VM. Un'ulteriore possibilità prevede di non saldare alcuna piazzola del connettore CN3 in questo modo l'alimentazione VM verrà usata per alimentare i soli motori mentre l'alimentazione per l'ATmega32U4 dovrà essere applicata esternamente dalla logica di comando tramite il connettore CN1; in questo caso sarà necessario che la logica di comando venga alimentata separatamente. Per quando riguarda il connettore di alimentazione, esso prevede sei contatti di cui due per l'alimentazione denominati VM e GND, due per il primo motore (1A e 1B) e altrettanti per il secondo motore (2A e 2B). L'alimentazione VM dovrà ovviamente essere compatibile con la tensione di alimentazione dei motori e dovrà fornire sufficiente corrente per alimentare i motori. Il connettore ICSP è il connettore per la programmatore,

ne, spiegheremo più avanti come utilizzare questo connettore per caricare il bootloader all'interno dell'ATmega32U4.

Il CN1 è il connettore per l'alimentazione della logica e per la comunicazione seriale e I²C. Quanto al connettore micro USB, è utilizzato sia per impartire comandi al controller, sia per la programmazione del microcontrollore tramite l'IDE di Arduino. Per caricare il bootloader dovrete necessariamente procedere così: non collegate nulla sul connettore CN2 e lasciate dissaldato il connettore per la selezione dell'alimentazione, oppure lasciate impostata una tensione di 5V. Se disponete di un apposito programmatore per dispositivi Atmel conoscete già la procedura; altrimenti munitevi di una scheda Arduino Uno, che userete come programmatore.

Il connettore ICSP del controller OMC21 dovrà essere cablatto pin to pin con il connettore ICSP di Arduino utilizzato come programmatore, ovvero pin 1 con pin 1 e così via; fa eccezione il pin 5 (pin di reset) del connettore del controller, che deve essere connesso al pin10 di Arduino. A questo punto collegate la scheda Arduino alla porta USB; il LED di alimentazione di OMC si accenderà ad indicare la sua corretta alimentazione.

Caricate su Arduino lo sketch *Esempi-ArduinoISP*, il quale abiliterà Arduino alla funzione di programmatore; ora aprite il menu *Strumenti-Tipo di Arduino* e modificate il target di programmazione in Arduino Leonardo; selezionate il programmatore *Arduino as ISP* e quindi avviate la programmazione del bootloader cliccando su *Scrivi il bootloader*. Attendete circa un minuto per il completamento delle operazioni. L'integrato ATmega32U4 dispo-

ne di una seriale principale facente capo al modulo USB interno; a livello di programmazione, nulla cambia rispetto ad Arduino, se non per il fatto che non serve un convertitore USB/seriale esterno perchè già implementato all'interno. Grazie al bootloader, non appena OMC21 viene inserito nella USB esso verrà riconosciuto come periferica e creata una seriale virtuale etichettata come Arduino Leonardo alla quale viene assegnato un nome: nel caso di **Fig. 1**, COM21.

A questo punto il vostro OMC21 può essere gestito completamente via USB ed è possibile caricare, tramite l'IDE di Arduino lo sketch di gestione denominato *OMC21.ino*; descriviamo ora come funziona questo software di gestione, in particolare il controllo del ponte ad H. Iniziamo col dire che la gestione dei quattro interruttori statici rappresentati dai transistor può essere fatta in modo differente a seconda della funzione che si vuole ottenere. Prendiamo a riferimento la **Fig. 2** che rappresenta, in modo semplificato, i quattro interruttori del ponte ad H, che nel nostro driver sono implementati con dei MOSFET. Leggiamo attentamente il datasheet del driver LV8405V-D e ricaviamo la tabella di verità (**Tabella 1**) relativa allo stato degli interruttori (S1-S4) e delle uscite (OUT1 e OUT2) per ogni

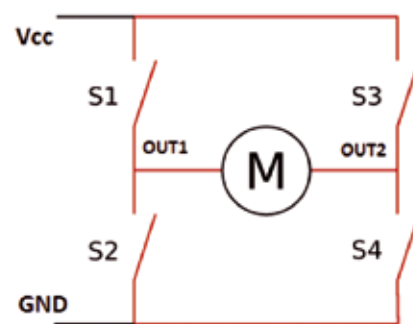


Fig. 2 - Schematizzazione del ponte ad H.

possibile combinazione degli ingressi; la lettera L indica un livello logico basso e la lettera H un livello logico alto. Analizzando la **Tabella 1** possiamo capire più in dettaglio le diverse modalità di funzionamento a seconda di come pilotiamo i due ingressi IN1 e IN2. Nel primo caso (IN1=H e IN2=H) entrambe le uscite si trovano in alta impedenza; in questa situazione è come se il motore fosse scollegato dal driver e libero di ruotare. Se invece ci poniamo nel quarto caso (IN1=L e IN2=L) è come se il motore si trovasse in cortocircuito; cosa cambia all'atto pratico tra i due casi? Se utilizziamo i motori per far avanzare un robot che sta avanzando a velocità sostenuta, nel primo caso, privando i motori dell'alimentazione, otterremo un arresto graduale del robot, che si fermerà in uno spazio dipendente dalla propria inerzia. L'arresto deriverà dai vari attriti negli organi meccanici e da quelli tra il robot e la superficie su cui si muoverà. Per comprendere cosa succede nella seconda ipotesi, occorre considerare che un motore in corrente continua è una macchina reversibile, ovvero se gli forniamo energia elettrica il suo motore ruota, ma se siamo noi a ruotarlo il motore fornirà energia

Tabella 1 - Tabella di verità del driver LV8405.

IN1	IN2	OUT1	OUT2	S1	S2	S3	S4	funzione
H	H	Z	Z	open	open	open	open	standby
H	L	L	H	open	close	close	open	reverse
L	H	H	L	close	open	open	close	forward
L	L	L	L	open	close	open	close	brake

elettrica, comportandosi, cioè, da generatore elettrico; in questo secondo caso, per il fenomeno della reazione d'indotto, lo sforzo richiesto per ruotare l'albero dipenderà da quanta corrente verrà prelevata dagli estremi degli avvolgimenti del motore. Trasferendo questo concetto al nostro caso, diciamo che se quando il robot è in movimento togliamo tensione al motore, questo continuerà a girare per inerzia, ma se il ponte mette in cortocircuito i suoi terminali, la corrente che ne deriva nell'avvolgimento causa a sua volta un campo elettromagnetico tale da opporsi alla causa che l'ha generato; essendo tale causa la rotazione dell'albero, quest'ultimo verrà frenato fino a fermarsi, allorché mancando la corrente smetterà anche l'azione frenante. Se provate entrambi i casi, anche solo facendo girare a mano una ruota, noterete in pratica l'effetto. In molti shield commerciali queste due funzioni non sono selezionabili, mentre nel nostro circuito abbiamo previsto la possibilità di selezionare

(via software) in quale modalità utilizzare le uscite.

I rimanenti due casi descritti nella **Tabella 1** sono invece più facili da comprendere: se poniamo l'ingresso IN1 a livello basso e comandiamo l'ingresso IN2 con un segnale PWM, abbiamo la possibilità di comandare il motore in direzione forward con una potenza direttamente proporzionale al valore del duty-cycle, il che permette di regolare la velocità di avanzamento del robot (quando il duty-cycle sarà zero fermeremo il robot con l'effetto freno precedentemente descritto). Se invece applichiamo il segnale PWM all'ingresso IN2, ma poniamo l'ingresso IN1 a livello alto, quando il duty-cycle ha valore massimo siamo nel caso delle uscite ad alta impedenza ed il robot si fermerà senza alcun effetto frenante.

Invertendo IN1 con IN2 avrem-



Due...uno...Arduino Zero!

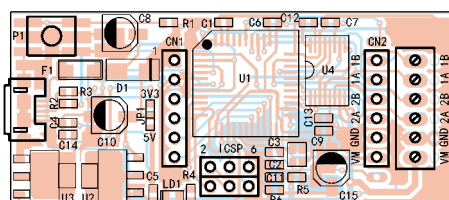
Dopo il divorzio in casa Arduino, Arduino srl (la ex Smart Project che realizzava le board ufficiali) ha annunciato Arduino Zero Pro, scheda a 32 bit che rappresenta un'estensione della Uno. Dotata di un micro Atmel SAMD21 basato su core ARM Cortex M0+ a 32 bit, nasce per applicazioni che vanno dai dispositivi

indossabili, all'Internet of Things, dall'automazione high tech, alla robotica. La scheda è dotata dell'Embedded Debugger (EDBG) di Amtel, che fornisce un'interfaccia di debug completa senza bisogno di software aggiuntivo e che supporta sia la COM virtuale che il tradizionale bootloader. Arduino Zero Pro ha una Flash da

256 kb e 32 kb di SRAM e può essere facilmente collegata a periferiche esterne, quali telefoni, tablet, macchine fotografiche digitali.

Elenco Componenti:

R1, R6: 10 kohm (0603)
 R2, R3: 22 ohm (0603)
 R4: 470 ohm (0603)
 R5: 100 kohm (0603)
 C1: 100 nF multistrato (0603)
 C2: 15 pF ceramico (0603)
 C3: 15 pF ceramico (0603)
 C4: 1 μ F ceramico (0603)
 C5÷C7: 100 nF multistrato (0603)
 C8: 10 μ F 10 VL elettrolitico (Ø 4mm)
 C9, C11: 100 nF multistrato (0603)
 C10: 10 μ F 35 VL elettrolitico (Ø 4mm)
 C12: 10 nF multistrato (0603)
 C13, C14: 100 nF multistrato (0603)
 C15: 4,7 μ F 10 VL elettrolitico (Ø 4mm)
 D1: GF1M
 F1: Fusibile 500mA auto ripristinante
 LD1: LED verde (0805)
 U1: ATMEGA32U4-AU (MF1173)
 U2: LD1117S50CTR



U3: LD1117AS33TR
 U4: LV8405V-TLM-E
 P1: Pulsante SMD basso profilo
 Q1: Quarzo 16MHz

Varie:

- Connettore micro-USB
- Strip maschio 3 vie (2 pz.)
- Strip maschio 6 vie (1 pz.)
- Strip maschio 8 vie (1 pz.)
- morsetto 6 poli passo 2,54mm
- Circuito stampato

dei motori con quella di lettura degli ingressi analogici di Arduino, la tensione viene ridotta di un fattore undici e quindi resa disponibile all'ingresso A0 di Arduino. Conoscere la tensione permette, nel caso di alimentazione a batteria, di evitare la scarica completa, ovvero, in carica, di impedire la sovra-scarica (sempre dannosa). Al connettore CN2 fanno capo ulteriori pin di Arduino, ovvero i rimanenti pin digitali non usati dal driver e tre pin analogici. In pratica è come se disponessimo di una scheda Arduino Leonardo con a bordo un driver per motori.

IL FIRMWARE

Occupiamoci ora di descrivere il programma che sarà caricato sull'ATmega32U4. Lo sketch dovrà occuparsi di gestire la comunicazione USB, quella seriale e quella I²C, interpretando i comandi letti e mettendoli in pratica comandando adeguatamente gli ingressi del driver. Lo sketch prevede quindi una routine di ascolto sulle tre porte di comunicazione, in attesa di dati in arrivo; fortunatamente il piccolo ATmega32U4 dispone

mo le stesse funzioni, ma con il motore che ruoterà in senso inverso. Vedete quindi come, giocando sugli ingressi, abbiamo la possibilità di scegliere direzione e velocità del motore e di attivare o meno l'effetto frenante.

Gli ingressi del driver IN1 e IN2 per il primo motore e IN3 e IN4 per il secondo motore, sono connessi a quattro uscite PWM dell'integrato ATmega32U4, facilmente gestibili con l'istruzione *analogWrite* disponibile nell'IDE di Arduino. Per il collegamento dei motori e dell'alimentazione di potenza è stato previsto il connettore CN2, sul quale possono essere saldati sia uno strip maschio o femmina, sia un connettore con morsetti a vite. Abbiamo previsto anche la possibilità di leggere la tensione

di alimentazione VM utilizzando un semplice partitore di tensione; per rendere compatibile il livello della tensione di alimentazione

Tabella 2 - Sequenza di byte per il comando del controller.

byte1	byte2	byte3	byte4
start byte = '\$'	modo e direzione	velocità del motore 1	velocità del motore 2

Tabella 3 - Funzioni relative al byte2 della sequenza di comando.

bit	funzione	condizione dei motori
0	motor 1 direction	0=foward, 1=reverse
1	motor 2 direction	0=foward, 1=reverse
2	motor 1 brake	0=brake, 1=standby
3	motor 2 brake	0=brake, 1=standby
4	read VM	0=null, 1=request motor volage
5	unused	-
6	unused	-
7	unused	-

di due moduli UART ed una porta I²C hardware, perfette per il nostro progetto. La seriale principale (Serial) fa capo al modulo USB e dal lato software si comporta esattamente come la seriale di Arduino Uno, perciò la ricezione di dati su questa porta può essere ottenuta con l'istruzione *Serial.available()* che riporta il numero di caratteri presenti sul buffer di lettura. In questa modalità l'OMC21 è controllabile tramite un PC o una qualsiasi periferica che disponga di una USB host. La dichiarazione di utilizzo di questa seriale avviene con la riga di codice *Serial.begin(9600)* dove 9600 è la velocità di comunicazione (modificabile a piacimento secondo le vostre necessità). Abbiamo usato 9600 come impostazione predefinita perché è la velocità più usata; volendo ridurre i tempi di lettura dei comandi, potete incrementare questo valore sino al massimo di 115.200 bps. Se sono presenti caratteri nel buffer del modulo UART, vengono letti tramite l'istruzione *Serial.readBytes(inByte, maxinByte)* che permette di leggere un numero prefissato di byte in un lasso di tempo prestabilito. Il protocollo di comunicazione che abbiamo pensato di utilizzare prevede l'utilizzo di quattro byte, di cui il primo è solo il carattere '\$', necessario per riconoscere l'inizio della sequenza; questo, assieme al numero di caratteri inviati, rappresenta il controllo di correttezza nella lettura del comando. Terminata la lettura dei quattro byte, si verifica se ve ne siano altri; in ogni caso il buffer viene svuotato. Può capitare che inviando i comandi si aggiungano inavvertitamente dei caratteri non necessari, come ad esempio un fine riga o un ritorno a capo; nella nostra applicazione questi vengono ignorati e cancellati dal

Listato 1

```
if(Serial.available() > 0)
{
  char inByte[maxinByte];
  int numByte = Serial.readBytes(inByte, maxinByte);
  //reads until maxinByte are reads or if TimeOut is occurred
  byte extraByte; // clear the buffer if extra bytes is arrived
  while (Serial.available())
  extraByte = Serial.read();
  //if data starting with '$' and lenght is correct...
  if (inByte[0]=='$' && numByte==maxinByte)
  {
    dirM1 = inByte[1];
    dirM1 = dirM1 & 0x01;
    dirM2 = inByte[1];
    dirM2 = (dirM2 & 0x02)>>1;
    modeM1 = inByte[1];
    modeM1 = (modeM1 & 0x04)>>2;
    modeM2 = inByte[1];
    modeM2 = (modeM2 & 0x08)>>3;
    speedM1 = inByte[2];
    speedM2 = inByte[3];
    setMotor();
    byte readVM = inByte[1];
    readVM = (readVM & 0x10)>>4;
    if (readVM==1)
    {
      int VM = analogRead(VM_pin)*36;
      VM = VM>>6;
      Serial.println(VM); //VM*10 [volt] (0-255)
    }
  }
}
```

buffer. I rimanenti tre byte dopo il carattere '\$' rappresentano i comandi per il controller: il secondo byte contiene informazioni sulla modalità di funzionamento (brake o standby) e sulla direzione di marcia dei motori, mentre i byte tre e quattro contengono l'informazione della velocità,

rispettivamente, per il motore 1 e 2. In questo modo, inviando una sequenza di quattro byte si comandano contemporaneamente entrambi i motori (**Tabella 2**). Il byte2 contiene informazioni sia sulla modalità di funzionamento che sulla direzione dei motori (**Tabella 3**). Il quinto bit del byte2,

Listato 2

```
void I2CreceiveEvent(int byteIn)
{
  while( Wire.available()>0) // loop until available byte on wire
  {
    char acknowledge = Wire.read();
    if (acknowledge=='$')
    {
      char mode = Wire.read(); // read mode byte
      speedM1 = Wire.read(); // read speed byte
      speedM2 = Wire.read(); // read speed byte
      dirM1 = mode;
      dirM1 = dirM1 & 0x01;
      dirM2 = mode;
      dirM2 = (dirM2 & 0x02)>>1;
      modeM1 = mode;
      modeM1 = (modeM1 & 0x04)>>2;
      modeM2 = mode;
      modeM2 = (modeM2 & 0x08)>>3;
      setMotor();
    }
  }
}
```

se posto a uno consente di richiedere la lettura della tensione di alimentazione; è l'unico caso in cui è necessaria la comunicazione seriale bidirezionale, altrimenti la sola linea di invio dati al controller è sufficiente.

Nel **Listato 1** trovate le righe di codice dello sketch che svolgono le funzioni descritte. Se volete comandare il controller tramite la seriale di un microcontrollore, dovete utilizzare la seconda porta seriale (Serial1) del microcontrollore ATmega32U4 i cui pin TX e RX sono disponibili nel connettore CN1. La connessione prevede che la linea TX del microcontrollore sia connessa alla linea RX di OMC21 e se richiesta la lettura della tensione di alimentazione, anche la linea RX del microcontrollore connessa alla linea TX di OMC21. La riga di codice

Tabella 4 - Costruzione del byte2 per l'esempio indicato.

bit	funzione	valore
0	motor 1 direction forward	0
1	motor 2 direction reverse	1
2	motor 1 brake ON	0
3	motor 2 brake OFF	1
4	read VM	0
5	unused	0
6	unused	0
7	unused	0

Tabella 5 - Sequenza di byte per l'esempio indicato.

formato	byte1	byte2	byte3	byte4
binario	B00011000	B00001010	B10000000	B01000000
decimale	D36	D10	D128	D64
esadecimale	0x24	0x0A	0x80	0x40

che abilita questa seriale è *Serial1.begin(9600)*; anche in questo caso la velocità di trasmissione può essere modificata sino a un massimo di 115.200 bps. Per il resto la gestione a livello software è

Tabella 6 - Funzioni del connettore CN1.

	byte4
1	+5V
2	GND
3	TX serial
4	RX serial
5	SDA
6	SCL

Listato 3

```
// OpenMotorControl21
// Example to use Processing for control OMC21 USB

import processing.serial.*;

Serial myPort; // Create object from Serial class

void setup()
{
  size(200,200);
  background(0);
  String portName = Serial.list()[0]; //change the 0 to a 1 or 2 etc.
  println("portName: ", portName);
  myPort = new Serial(this, portName, 9600);
}

void draw()
{
}

void mouseClicked() // if click on background
{
  int dirM1 = 0; // set direction for Motor1 0=forward 1=reverse
  int dirM2 = 0; // set direction for Motor2 0=forward 1=reverse
  int modeM1 = 0; // set mode for Motor1 0=brake 1=standby
  int modeM2 = 0; // set mode for Motor2 0=brake 1=standby
  int mode = (modeM2<<3) + (modeM1<<2) + (dirM2<<1) + dirM1;
  char modedir=char(mode); // convert integer to char
  char speedM1=128; // set speed for Motor1 at 50%
  char speedM2=128; // set speed for Motor2 at 50%
  myPort.write('$'); // write start char
  myPort.write(modedir); // write mode-dir byte
  myPort.write(speedM1);
  myPort.write(speedM2);
}
```

la stessa della seriale principale. Per quanto riguarda la porta di comunicazione I²C questa viene abilitata tramite l'istruzione *Wire.onReceive(I²CReceiveEvent)* in cui la label *I²CReceiveEvent* indica la procedura richiamata dall'interrupt all'arrivo del primo byte. Per il dettaglio di come avviene l'elaborazione dei dati ricevuti, riferitevi al **Listato 2**.

Ad ogni arrivo di dati su questa porta viene richiamata la procedura che si preoccupa di verificare se la sequenza è compatibile con lo standard OMC21 e quindi ne estrapola i valori. Lo sketch per OMC21 è quindi predisposto per monitorare continuamente l'invio dei byte di comando su tutte e tre le porte di comunicazione: USB, seriale e I²C. Non ha importanza su quale porta arriveranno i dati, al limite potrebbero arrivare anche in contemporanea,

anche se sarebbe abbastanza inutile farlo, il vantaggio è comunque quello di caricare un unico sketch, evitando di impazzire con la programmazione. Il nostro controller OMC21 una volta programmato è pronto ad azionare i motori in base ai comandi ricevuti, vediamo con degli esempi come poterlo fare sfruttando le diverse porte di comunicazione.

GESTIONE TRAMITE USB

Appena programmato, l'OMC21 è ancora connesso alla USB e viene subito voglia di testarlo senza ulteriori cablaggi, utilizzando il PC per l'invio dei comandi. A titolo di esempio vediamo come costruire i tre byte dati per il controllo dei motori nel caso volessimo il motore 1 con direzione forward con potenza del 50% e funzione frenatura attivata ed il motore 2 con direzione reverse con potenza del 25% senza frenatura (standby). Essendo la velocità espressa come valore numerico compreso da 0 (0% della potenza) a 255 (100% della potenza), ai due valori 50% e 25% corrispondono rispettivamente i valori decimali 128 e 64. Il byte2 dovrà essere costruito bit per bit come nella **Tabella 4**.

I tre byte di comando saranno quindi quelli riportati in **Tabella 5**. Se il nostro controller è con-

nesso alla seriale difficilmente potremmo controllarlo utilizzando la funzione SerialMonitor di Arduino, per il semplice motivo che i primi 31 caratteri del codice ASCII non sono stampabili, in quanto usati come comandi relativamente alla comunicazione con le stampanti. Se vogliamo inviare i quattro byte così costruiti è necessario utilizzare un monitor seriale che permetta di gestire anche i caratteri non stampabili; nel nostro caso abbiamo usato un software gratuito, distribuito in

versione *portable* di nome SSCOM (Fig. 3), ma tanti altri sono disponibili in rete. Nell'esempio sono stati inviati (cerchio rosso) i byte 0x24 0x10 0x8F 0x00 corrispondenti a motore 1 fermo e motore 2 forward al 56% e richiesta tensione di alimentazione VM. Evidenziati con un cerchio blu, i dati di risposta relativi ad una tensione di alimentazione di circa 6,8 volt. Se invece volete crearvi un software di gestione tutto vostro, vi consigliamo di utilizzare Processing, che permette con semplicità di utilizzare una comunicazione seriale via USB, un esempio di sketch è riportato nel **Listato 3**. In questo esempio, semplicemente viene creato un oggetto della classe *serial*, quindi all'avvio dello sketch viene aperta la porta e non appena viene cliccato su un qualsiasi punto vengono costruiti ed inviati i quattro byte necessari al comando dell'OMC21. Non avendo previsto un menu a tendina per la scelta della COM, la porta viene specificata con la riga di codice

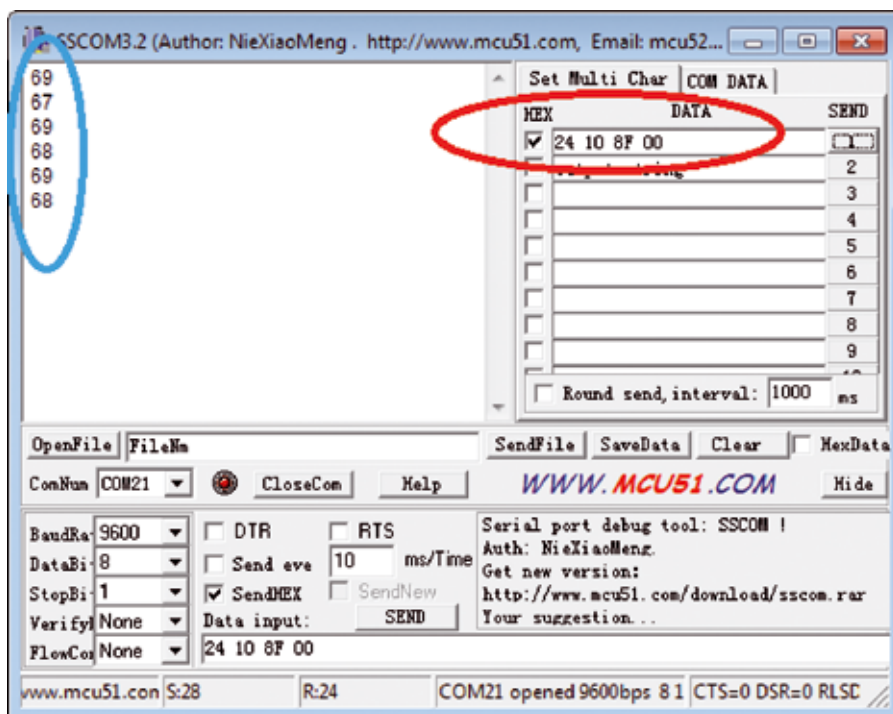


Fig. 3 - Monitor Seriale SSCOM per inviare i comandi ad OMC21.

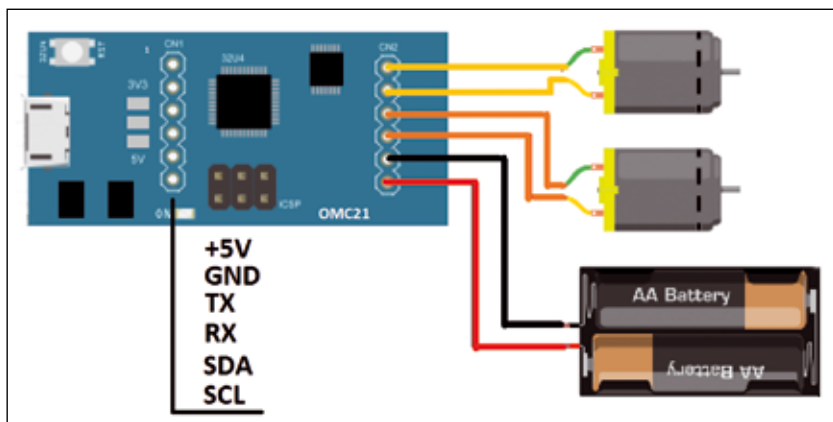


Fig. 4 - Collegamento di motori e alimentazione al controller OMC21.

Listato 4

```
// OpenMotorControl21
// Example to use Arduino for control OMC21 Softserial Serial

#include <SoftwareSerial.h>
SoftwareSerial OMCSerial(2, 3); // RX, TX

void setup()
{
  Serial.begin(9600); //use Hardware Serial to communicate with PC
  OMCSerial.begin(9600); //use software serial to communicate with OMC21
}

void loop()
{
  byte dirM1 = 0; // set direction for Motor1 0=forward 1=reverse
  byte dirM2 = 0; // set direction for Motor2 0=forward 1=reverse
  byte modeM1 = 0; // set mode for Motor1 0=brake 1=standby
  byte modeM2 = 0; // set mode for Motor2 0=brake 1=standby
  byte mode = (modeM2<<3) + (modeM1<<2) + (dirM2<<1) + dirM1;
  mode = mode+16; //if request Motor voltage
  char speedM1=128; // set speed for Motor1 at 50%
  char speedM2=128; // set speed for Motor2 at 50%
  OMCSerial.write('$');
  OMCSerial.write(mode);
  OMCSerial.write(speedM1);
  OMCSerial.write(speedM2);
  delay(1000);
}
```

`portName = Serial.list()[0]` in cui è necessario indicare l'indice relativo all'elenco del COM installate nel PC. Se, come spesso accade, non disponete di COM hardware ed al PC è connesso solo Arduino,

nella lista trovate una sola COM il cui indice è zero.

GESTIONE VIA SERIALE

Se vogliamo comandare l'OMC21 tramite una scheda Arduino,

conviene utilizzare una seriale software, utilizzando una sola linea digitale da connettere alla linea RX del connettore CN1; in questo modo la seriale hardware rimarrà libera di comunicare con il PC.

Lo sketch di esempio è riportato nel **Listato 4**, dove viene mostrato come sia possibile costruire il byte di controllo (byte2) partendo dalle impostazioni dei singoli bit di direzione e frenatura. Sarà la riga di codice `int mode = (modeM2<<3) + (modeM1<<2) + (dirM2<<1) + dirM1` a ricostruire il byte completo traslando e sommando i singoli bit. Utilizzeremo successivamente quattro comandi `Serial.write` per inviare i singoli byte contenenti il carattere di inizio e i tre byte di dati.

GESTIONE TRAMITE I²C

Se volete controllare l'OMC21 tramite la porta TWI di Arduino, non dovete far altro che connettere le due linee SDA e SCL di entrambe le porte tra loro e collegare i GND di entrambe le schede. Non sono necessarie le resistenze di pull-UP per il semplice motivo che sono già presenti internamente al microcontrollore sia in Arduino Uno che sull'ATmega32U4 ed abilitate in automatico non appena si richiama la funzione `Wire.begin()`. Lo sketch da caricare in Arduino Uno per comandare l'OMC21 è contenuto nel **Listato 5**, che si presenta molto simile ai già descritti precedenti listati. C'è però un'importante differenza con il modo di comunicare tramite seriale che riguarda il sistema di indirizzamento delle periferiche. Mentre in una comunicazione seriale lo scambio di dati avviene punto a punto, ovvero solo due periferiche dialogano tra loro, nel Bus I²C possono coesistere sino a 127 periferiche di tipo Slave, tutte

Listato 5

```
// OpenMotorControl21
// Example to use Arduino for control OMC21 I2C

#include <Wire.h>
const int OMC21address=4; //address of OMC21

void setup()
{
  Wire.begin(); // join i2c bus (address optional for master)
}

void loop()
{
  byte dirM1 = 0; // set direction for Motor1 0=forward 1=reverse
  byte dirM2 = 0; // set direction for Motor2 0=forward 1=reverse
  byte modeM1 = 0; // set mode for Motor1 0=brake 1=standby
  byte modeM2 = 0; // set mode for Motor2 0=brake 1=standby
  byte mode = (modeM2<<3) + (modeM1<<2) + (dirM2<<1) + dirM1;
  char speedM1=128; // set speed for Motor1 at 50%
  char speedM2=128; // set speed for Motor2 at 50%
  Wire.beginTransmission(OMC21address); // transmit to OMC21
  Wire.write('$');
  Wire.write(mode);
  Wire.write(speedM1);
  Wire.write(speedM2);
  Wire.endTransmission(); // stop transmitting

  delay(1000);
}
```


in comunicazione con l'unico master. Il master in questo caso è Arduino Uno (o la logica di comando) mentre gli slave sono le schede OMC21; il vantaggio evidente è quello di poter utilizzare più controller e quindi più motori, tutti gestiti da un'unica scheda Arduino utilizzando una sola porta TWI.

Ricordiamo che in Arduino la porta TWI utilizza a tutti gli effetti il protocollo I²C, che però è proprietario della Philips (che lo ha inventato).

Nel **Listato 5** è infatti presente la riga di codice `const int OMC21address=4` che ha lo scopo di specificare verso quale slave avviene la comunicazione; ovviamente sul BUS TWI potranno coesistere anche altri slave non necessariamente controller motori OMC21. Nello sketch *OMC21.ino* è stato specificato come indirizzo predefinito di periferica il valore 4, ma questo può essere modificato a piacimento se intendete usare sul bus una periferica che ha già questo indirizzo.

Se usate più controller OMC21 è necessario che ciascuno abbia un indirizzo univoco; dovrete quindi caricare su ciascun controller uno sketch in cui avrete specificato un indirizzo differente. Nella stesura del programma di controllo dovrete poi tenere conto dei vari indirizzi assegnati, per assicurare il corretto transito dei dati.

UTILIZZO DEL CONTROLLER

Quanto ai collegamenti e all'utilizzo pratico, sono possibili numerose combinazioni per l'alimentazione della logica di controllo e della sezione di potenza dell'integrato driver LV8405V-D; la **Fig. 4** mostra il collegamento generale dei due motori in cc (per lo stepper-motor bipolare ogni avvolgimento va al posto di

un motore) e dell'alimentazione della scheda.

Invece, nella **Fig. 5** viene illustrato come impostare il ponticello 3V3/5V (in questo caso va lasciato completamente aperto) di selezione dell'alimentazione del microcontrollore e della logica del driver LV8405V-D affinché venga prelevata direttamente dalla connessione USB; nella stessa figura i motori vengono alimentati da VM.

La **Fig. 6** mostra come impostare il ponticello se si desidera alimentare il circuito di potenza con VM e il microcontrollore e la logica del driver con i 3,3 volt ricavati dalla stessa VM tramite il regolatore U3 (il ponticello va realizzato chiudendolo con una goccia di stagno).

Nella **Fig. 7** è invece proposta l'impostazione del ponticello ammettendo di alimentare il microcontrollore e la logica del driver con i 5 volt che il regolatore U2

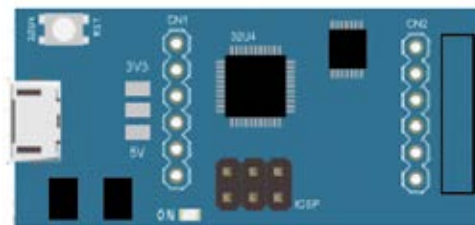


Fig. 5 - Motori alimentati da VM e ATmega32U4 alimentato esternamente dalla logica di controllo.

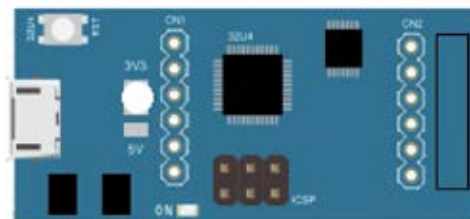


Fig. 6 - Motori alimentati da VM e ATmega32U4 alimentato a 3,3V da VM.

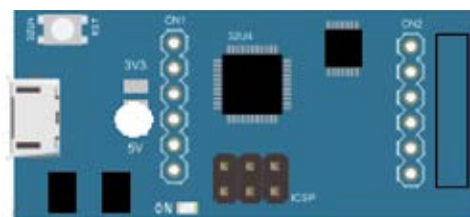


Fig. 7 - Motori alimentati da VM e ATmega32U4 alimentato a 5V da VM.

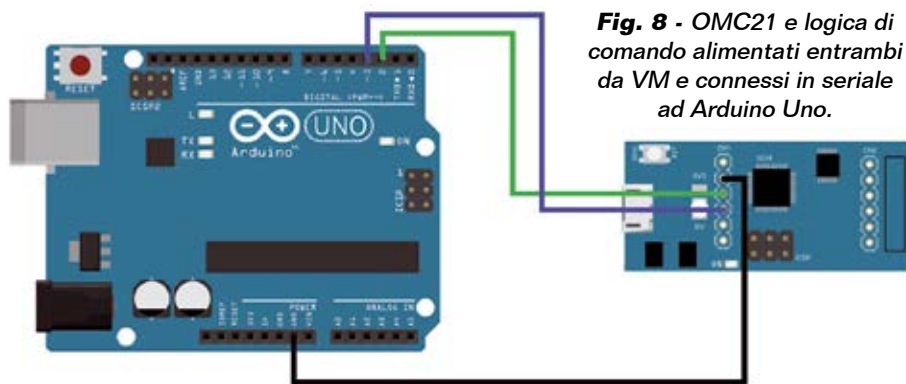


Fig. 8 - OMC21 e logica di comando alimentati entrambi da VM e connessi in seriale ad Arduino Uno.

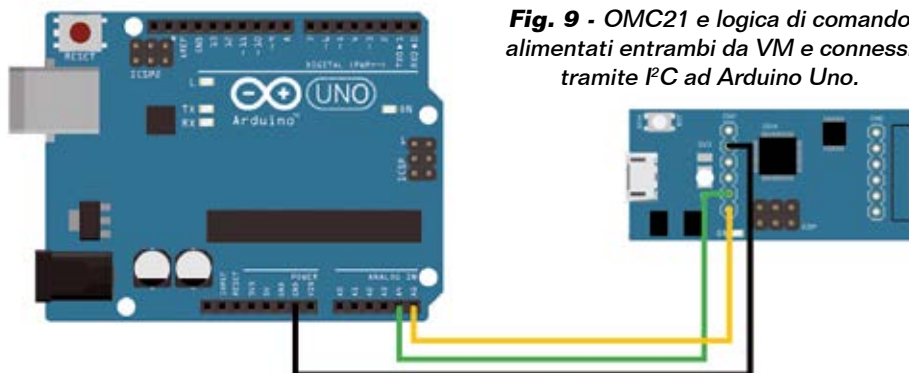


Fig. 9 - OMC21 e logica di comando alimentati entrambi da VM e connessi tramite PC ad Arduino Uno.

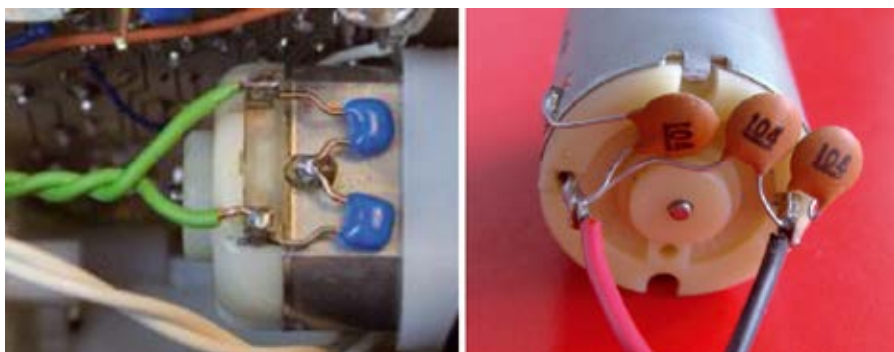


Fig. 10 - Utilizzo dei condensatori come soppressori dei disturbi su un motore a spazzole.

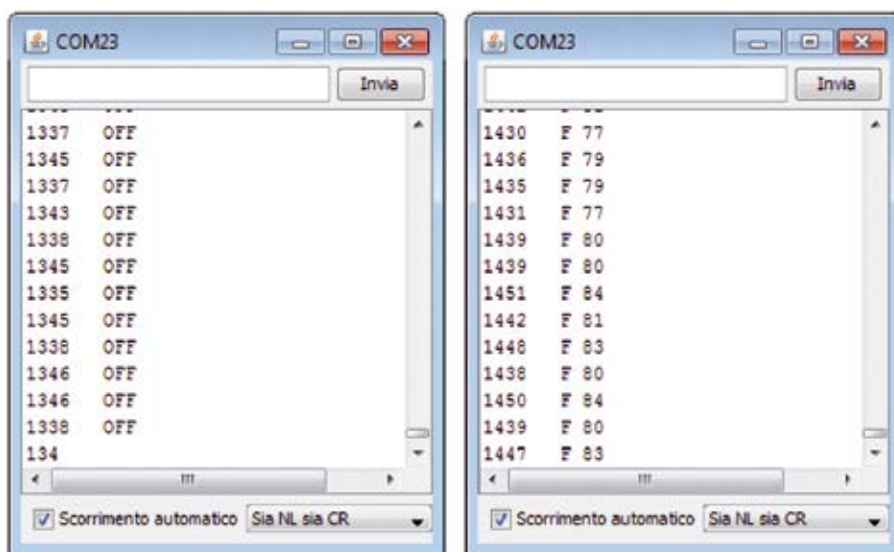


Fig. 11 - Dati ricevuti dalla ricevente RC.

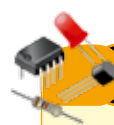
ricava da VM (tensione d'ingresso del circuito e di potenza). E veniamo alle connessioni con Arduino Uno per il controllo seriale: le **Fig. 8** e **Fig. 9** riguardano rispettivamente la connessione via seriale TTL e quella tramite I²C-Bus. Aggiungiamo ora alcune raccomandazioni per il corretto utilizzo di questo controller, tenendo conto di tutte le problematiche che si possano incontrare all'atto pratico nell'utilizzo dei motori a spazzole. È buona norma prima di collegare un motore verificare se la massima corrente assorbita (di solito chiamata corrente di stallo) non eccede il valore massimo consentito. Un'altra buona norma è di saldare direttamen-

te sui contatti del motore tre condensatori ceramici del valore compreso tra 10 nF e 100 nF, con lo scopo di sopprimere i disturbi derivati dallo strisciamento delle spazzole sul collettore. I condensatori vanno saldati tra i due terminali e la carcassa ed eventualmente anche uno tra i due terminali di alimentazione (vedere **Fig. 10**).

Per concludere l'articolo, riportiamo anche un esempio pratico di utilizzo del nostro controller utilizzando in sostituzione della circuiteria elettronica guasta in una piccola automobilina RC. In questo esempio utilizziamo l'OMC21 per leggere i segnali provenienti da un ricevitore RC, di quelli usati in ambito model-

listico, e comandare il motore dell'automobilina. La ricevente è alimentata dallo stesso OMC grazie allo stabilizzatore di tensione interno impostato sui 5V. Il servo per lo sterzo è controllato direttamente dalla ricevente RC. L'alimentazione giunge da una piccola batteria LiPo connessa direttamente ad OMC21. Il programma da inserire sul controller si chiama *esc_rc_car2.ino* e prevede anche una funzione di diagnostica grazie alla quale su serial monitor è possibile visualizzare i dati ricevuti dalla ricevente RC (**Fig. 11**).

Il programma è molto semplice e si basa sulla lettura della durata dell'impulso ricevuto dalla ricevente RC grazie alla funzione *pulseIn* di Arduino; da questa lettura è poi sufficiente ricavare il giusto valore del duty-cycle per il comando dei motori. Per il resto sono state usate le funzioni già precedentemente descritte, con in aggiunta la lettura della tensione della batteria, impostando lo spegnimento motore (cut-off), non appena questa scenda al di sotto dei 6V, al fine di salvaguardare la batteria di alimentazione. ■



per il MATERIALE

Il controller OpenMotorControl (cod. FT1173M), viene fornito assemblato, con il bootloader precaricato ed è disponibile presso Futura Elettronica al prezzo di 37,00 Euro. Il prezzo si intende 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>

Proponiamo tre circuiti dedicati al tema "power", realizzati con altrettanti integrati della Linear Technology: un selettore automatico della fonte di alimentazione e due DC/DC converter.



TRE SOLUZIONI PER L'ALIMENTAZIONE

..... dell'ing. VINCENZO MENDOLA

Negli ultimi anni abbiamo assistito all'esplosione del fenomeno dei maker, gli appassionati di tecnologia e di autocostruzione che in tutto il mondo hanno dato libero sfogo alla realizzazione pratica delle loro molteplici idee, spaziando dai dispositivi più semplici a quelli più complessi. A contribuire in maniera significativa a questo processo, imprimendo una forte spinta alla diffusione di quella che viene considerata una vera e propria cultura, sono state le innumerevoli board di prototipazione disponibili in maniera sempre più capillare e

a prezzi sempre più contenuti tanto che ormai oggi sono diventate accessibili a tutte le tasche, comprese quelle degli studenti. Tra esse, quella che ha dato la spinta più energica a questa nuova passione per l'elettronica, l'informatica e la tecnologia in genere, è stata sicuramente Arduino, che a tutt'oggi rimane la più conosciuta e diffusa insieme a Raspberry Pi. La piccola scheda a microcontrollore nata in Italia, ha dato origine ad un fitto substrato di "accessori" e sensoristica "pronta all'uso" che ha fatto la fortuna di varie e notissime aziende americane (ma non

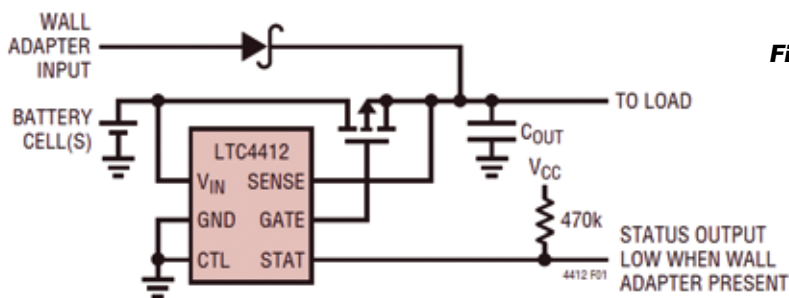


Fig. 1

solo), che hanno saputo cogliere le necessità di chi iniziava a muovere i primi passi in questo mondo, fornendo tutta una serie di schede più o meno piccole e semplici (non solo "shield" ma sempre più spesso vere e proprie schede stand-alone cui interfacciare i microcontrollori e quindi non necessariamente ed esclusivamente Arduino), molto spesso dotate di un singolo sensore contornato dai pochissimi componenti necessari a farlo funzionare correttamente e dotate di connessioni ideate per semplificare notevolmente l'interfacciamento alla scheda di prototipazione. Sono nate così le "breakout" board, come vengono chiamate dagli americani, ossia piccole schede elettroniche utili per risolvere in maniera rapida ed economica problematiche potenzialmente complesse per chi non è dotato dell'attrezzatura o anche più semplicemente della manualità per saldare componentistica SMD, oggi sempre più presente nei dispositivi portatili. Infatti i componenti SMD sono i più utilizzati dai produttori di semiconduttori, specialmente nel caso dei trasduttori più complessi come ad esempio gli accelerometri, che spesso sono disponibili esclusivamente racchiusi in contenitori praticamente impossibili da saldare a mano (ed in effetti non sono stati concepiti per questo scopo) a causa della frenetica corsa alla ricerca della massima

compattezza e miniaturizzazione. A questo gran numero di schede abbiamo deciso di aggiungere qualcosa anche noi, pubblicando, a partire dai numeri precedenti, i progetti di breakout board dalle funzionalità più utili ed interessanti che secondo noi potranno aiutare chi di voi necessita di dispositivi da poter interfacciare in maniera semplice e rapida alla propria MCU o scheda di prototipazione, per estenderne le funzionalità.

Oltre alla sensoristica di base da interfacciare alle schede a microcontrollore disponibili sul mercato, di comprovata utilità sono i piccoli alimentatori e i dispositivi in grado di gestire varie sorgenti di alimentazione applicate ad un solo carico.

Ecco perché abbiamo deciso di pubblicare, in queste pagine, i progetti di tre breakout board

destinate all'alimentazione, la prima delle quali è un circuito dai molteplici impieghi: un selettore automatico di sorgente di alimentazione, utile per chi ha la necessità di commutare tra due alimentazioni indipendenti, garantendo al contempo basse perdite energetiche.

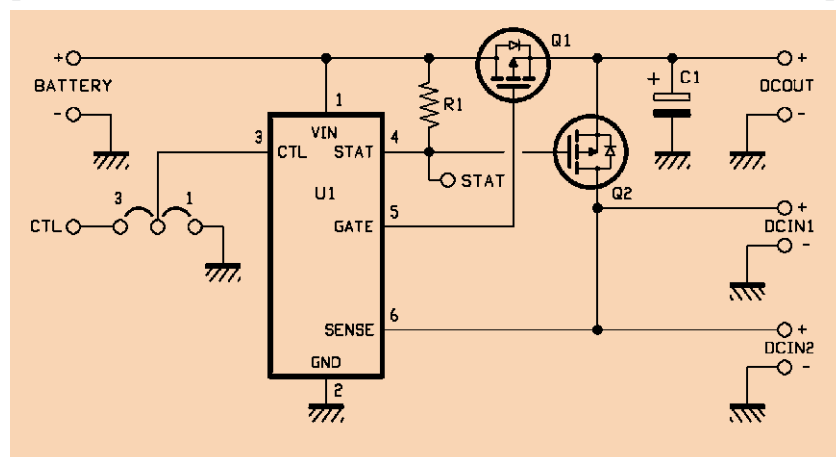
Per ottenere con semplicità questo risultato, abbiamo scelto un integrato della Linear Technology: si tratta dell'LTC4412 (<http://www.linear.com/docs/1527>) che è un selettore automatico che abilita una delle fonti di alimentazione e che, con pochissimi componenti esterni, ci permette di raggiungere lo scopo desiderato (Fig. 1).

SELETTORE DI ALIMENTAZIONE

Come si può vedere dallo schema elettrico, data l'estrema semplicità circuitale non è stato necessario scostarci dagli esempi applicativi forniti dal data-sheet: è stata scelta la configurazione a due MOSFET a canale P in quanto è la più efficiente, garantendo perdite di energia molto basse, grazie alla bassissima resistenza che i MOSFET presentano in conduzione (ON).

Gli altri due componenti indicati dal costruttore come necessari

[schema SELETTORE di ALIMENTAZIONE]



per il corretto funzionamento del sistema, sono una resistenza di pull-up da 470 kohm e il condensatore d'uscita.

Una manciata di componenti è sufficiente per avere un sistema completo e funzionante, la cui applicazione più comune è quella in cui la sorgente primaria è un comune adattatore di rete con uscita in continua a bassa tensione e l'altra è costituita da una cella al litio ricaricabile; nulla vieta di cambiare la configurazione per interfacciare differenti tipologie di alimentatori (ad esempio uno primario e uno secondario o di backup) oppure servirsi del circuito per commutare tra diverse sorgenti a seconda della loro disponibilità (celle fotovoltaiche e batterie) con lo scopo comune di assicurare in uscita una tensione ininterrotta, adeguata al proprio carico.

Il terminale denominato GATE, come facilmente intuibile pilota il gate del MOSFET a canale P, che viene portato in conduzione (gate a livello basso) quando viene a mancare la sorgente primaria, la cui presenza è monitorata tramite il piedino SENSE; il pin STAT è un'uscita open drain che può essere utile per monitorare, tramite un microcontrollore, la presenza della tensione primaria. Il pin CTL permette di controllare il funzionamento del dispositivo: quando è a livello basso, l'alimentatore connesso

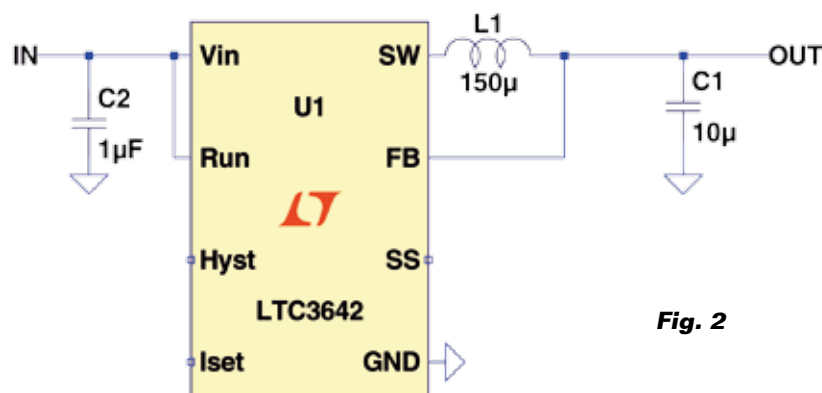


Fig. 2

alla rete o primario alimenta il carico mentre quando CTL è a livello alto, è la sorgente secondaria ad essere connessa all'uscita, indipendentemente dal valore dell'altra alimentazione. Per poter utilizzare queste funzionalità tramite un microcontrollore esterno, sul PCB sono state previste le connessioni ai pin CTL e STAT; per quanto riguarda CTL, sul PCB sono state realizzate le piazzole con cui selezionare, tramite un punto di saldatura, se connetterlo a massa oppure ad una piazzola accessibile dall'esterno.

E ORA, DUE DC/DC CONVERTER

I circuiti a commutazione o switching che dir si voglia, sono entrati prepotentemente all'interno delle apparecchiature elettroniche, qualunque sia la fonte di alimentazione; questo per via dell'elevata efficienza e versatilità offerte. Infatti essi permettono di ridurre, elevare e invertire la polarità della tensione d'ingresso, mantenendo sempre un'incomparabile efficienza rispetto ai regolatori lineari, utilizzati sempre meno frequentemente. L'evoluzione della componentistica integrata ha reso disponibile un gran numero di semiconduttori dalle ottime prestazioni, a prezzi sempre più ridotti, con frequenze di commutazione in continuo aumento, permettendo la riduzione delle dimensioni dell'intero dispositivo.

La tecnologia degli alimentatori switching ha raggiunto, negli ultimi anni, un grado di maturità veramente considerevole e la realizzazione di convertitori DC-DC di piccola potenza è una pratica diventata relativamente semplice grazie al gran numero di integrati disponibili sul mercato, realizzati da molte case produttrici di semiconduttori.

Vogliamo quindi presentare due alimentatori estremamente compatti e con un ottimo rendimento, utili per varie applicazioni, basati su due dispositivi ancora della Linear Technology: il primo è l'LTC3642 e il secondo l'LTC3400.

IL REGOLATORE BUCK

Analizziamo il circuito con l'LTC3642: questo integrato è un regolatore buck incapsulato in contenitore MSOP (a montaggio superficiale) a 8 pin, che può erogare in uscita una corrente massima di 50 mA. Il circuito con l'LTC3642 (trovate

[schema DC/DC BUCK]

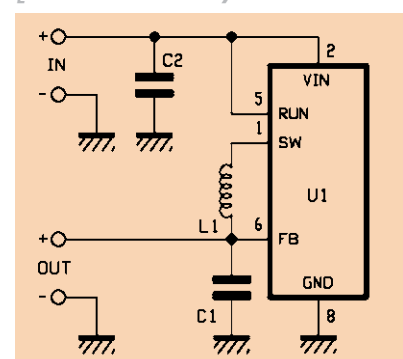
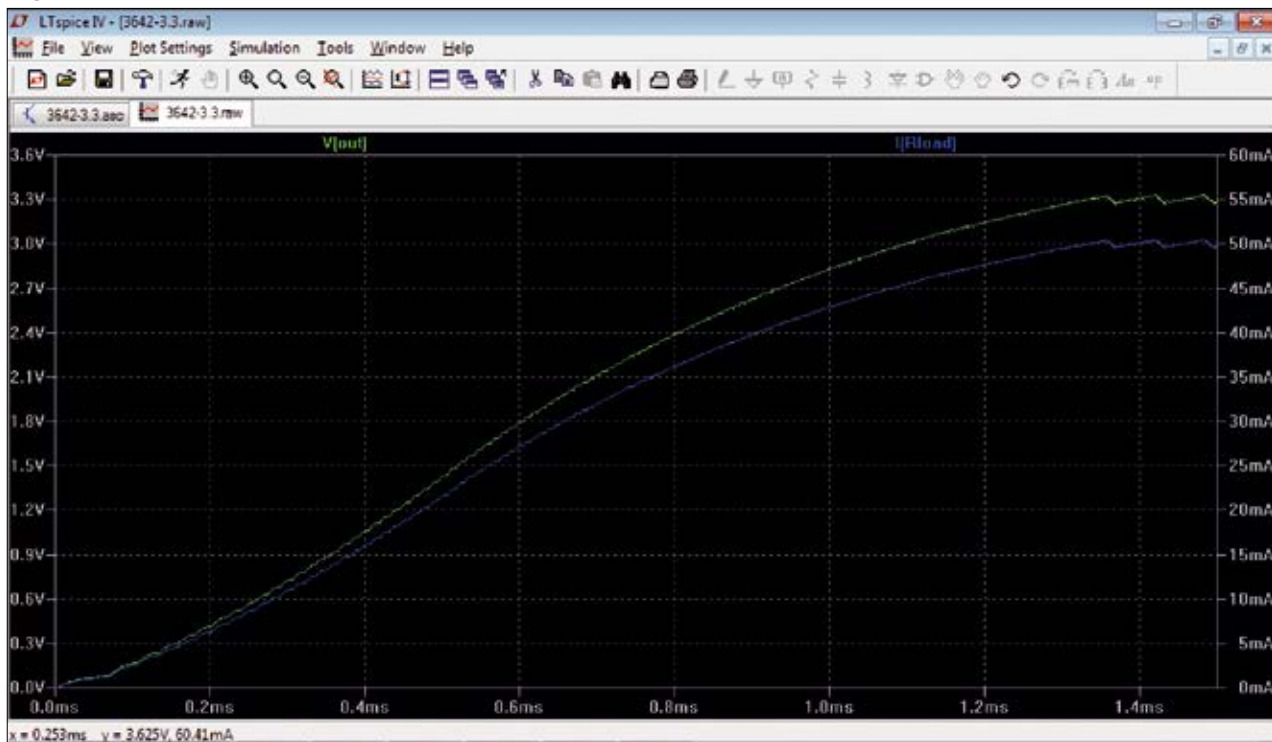


Fig. 3



il data-sheet all'indirizzo web www.linear.com/docs/26983 è derivato dall'application circuit mostrato nelle Fig. 2 ed il suo comportamento è descritto dal grafico di Fig. 3, ottenuto dal sof-

tware di simulazione LTSpice IV della Linear Technology, che vi abbiamo presentato nel numero 166 della rivista.

Il circuito permette di avere in un'uscita una corrente massima di 50mA, più che sufficiente per molte applicazioni basate su microcontrollori e su dispositivi di bassa potenza.

L'integrato accetta in ingresso una tensione compresa tra 4,5 e 45 V ed è disponibile sia con uscite fisse da 3,3 V e 5 V, sia in versione con tensione d'uscita regolabile tramite due resistori (Fig. 4). La semplicità ed il nume-

ro dei componenti esterni richiesti, rende l'LTC3642 molto valido nella sostituzione di piccoli regolatori LDO, quando è richiesto un elevato rendimento anche con tensioni d'ingresso relativamente alte. In pratica bastano solamente tre componenti esterni (due condensatori, uno all'ingresso e l'altro all'uscita, preferibilmente ceramici per via della loro bassa ESR, ed un'induttanza di piccolo valore) per avere una tensione di 3,3 V o di 5V con un'elevata efficienza di conversione.

Oltre che dei terminali che vedete utilizzati nello schema elettrico, l'integrato LTC3642 dispone di altri tre contatti che nella nostra applicazione abbiamo volutamente lasciato flottanti, così come riportato anche nell'esempio presente nel data-sheet della Linear Technology; il primo di questi è ISET, che permette di fissare la corrente di picco erogata in uscita tramite una resistenza: se lasciato flottante, in uscita viene erogata la massima corrente di picco ammessa

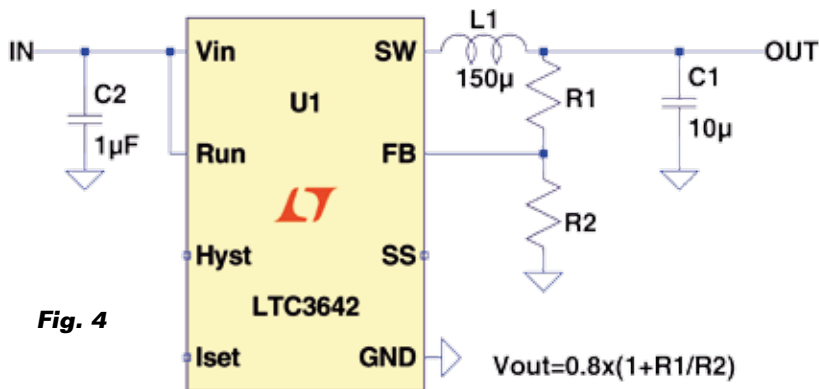
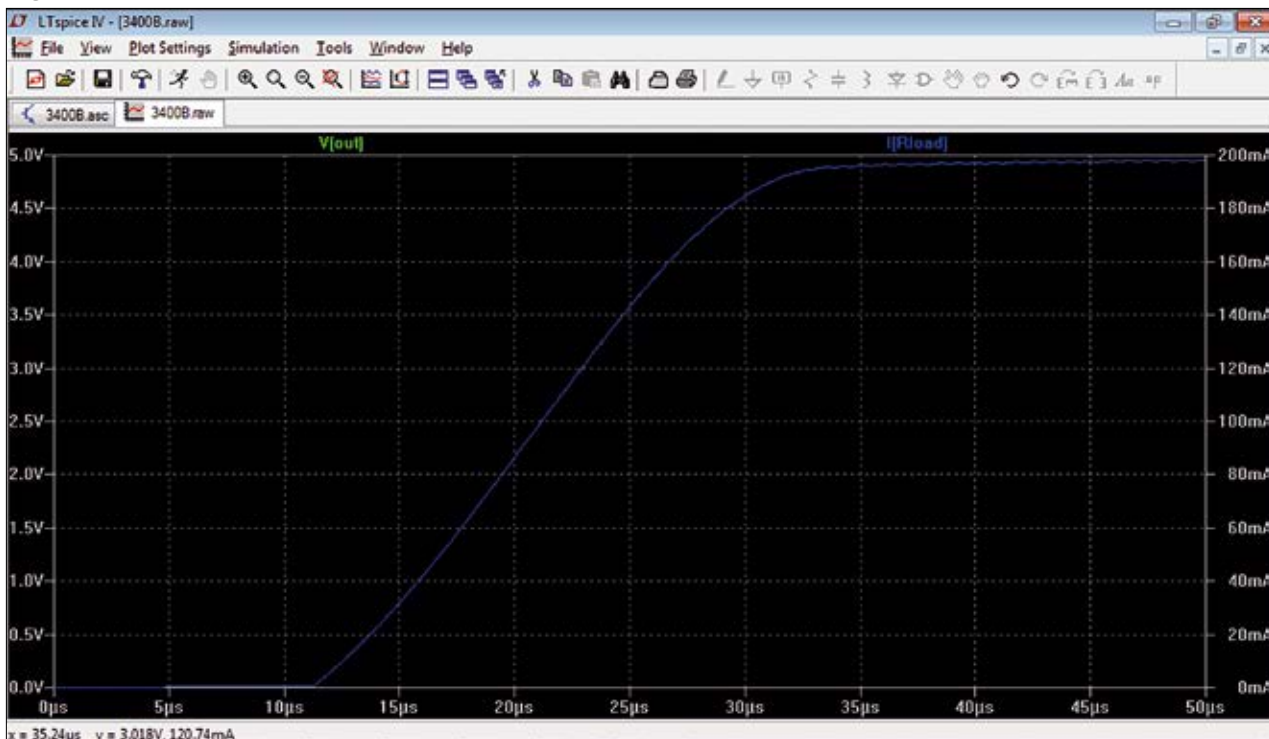


Fig. 4

Fig. 5



(115 mA) mentre se il pin viene collegato a massa, la corrente di picco disponibile prima che intervenga la protezione è ridotta a 25mA per l'LTC3642. Il secondo pin è SS e costituisce l'ingresso di soft start, il quale permette di fissare -tramite un condensatore collegato verso massa- la velocità con cui la massima corrente è resa disponibile in uscita dopo l'accensione dell'integrato; se il pin SS viene lasciato flottante, il tempo trascorso è di 0,75 mS. Abbiamo infine il pin HYST, che permette di controllare ed "aggiustare", qualora fosse necessa-

rio, l'isteresi del terminale RUN; a questo deve essere applicata una tensione di valore superiore a 1,2 V per abilitare l'integrato al normale funzionamento, mentre un valore di RUN inferiore a 0,7 V spegne il convertitore, riducendone il consumo a circa 3 μ A.

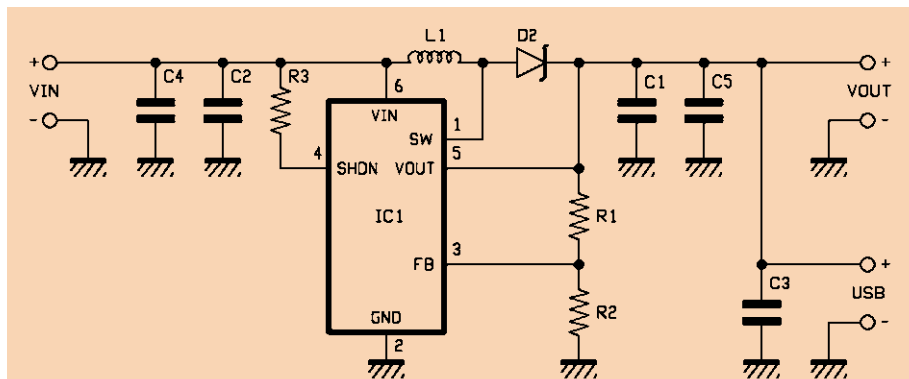
IL REGOLATORE BOOST

Un'altra tipologia di convertitore switching molto utile, specialmente nel caso di sistemi portatili, è quella boost o step-up, che permette di elevare la tensione

in ingresso ad un valore più alto, disponibile in uscita. Un tale sistema ci permette, ad esempio, di alimentare la nostra scheda di prototipazione a microcontrollore che richiede un'alimentazione a 5 V, tramite una cella al litio da 3,6 V, anche di dimensioni e di capacità piuttosto piccole, ovviamente adeguando di conseguenza l'autonomia del dispositivo connesso.

Ci sono in commercio innumerevoli circuiti integrati che effettuano questa funzione; per questo

[schema **REGOLATORE** con LTC3400]



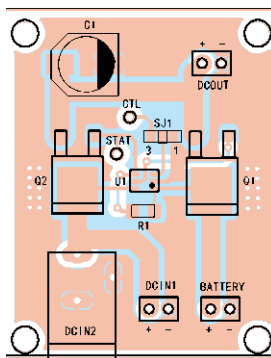
[SELETTORE di ALIMENTAZIONE]

Elenco Componenti:

R1: 470 kohm (0805)
C1: 220 μ F 12 VL elettrolitico ($\varnothing$ 8mm)
U1: LTC4412IS6
Q1, Q2: FDD4141

Varie:

- Plug alimentazione
- Strip maschio 2 vie (3 pz.)
- Strip maschio 1 via (2 pz.)
- Circuito stampato



[DC/DC BUCK]

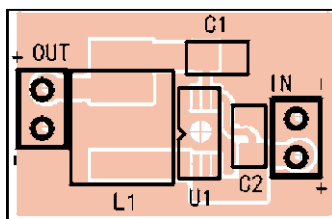
Elenco Componenti:

C1: 1 μ F ceramico (0805)
C2: 10 μ F ceramico (1206)
U1: LTC3642
L1: 150 μ H

Varie:

- Strip maschio 2 vie (2 pz.)
- Circuito stampato

ingrandito del 200 %



articolo abbiamo scelto l'LTC3400 della Linear Technology, che racchiude in un piccolissimo contenitore SOT-23 a 6 pin, un intero regolatore switching sincrono, internamente compensato per ridurre al minimo i componenti esterni, caratterizzato da un'elevata efficienza che arriva fino al 92% e da una bassissima tensione di accensione, tipicamente 0,85 V, valore che scende fino a 0,5 V una volta avviato l'oscillatore, permettendo all'integrato di funzionare fino alla completa scarica, qualora richiesto, delle celle da 1,5 V. Il comportamento del circuito è mostrato nel grafico tracciato con LTspice IV in Fig. 5.

Il data-sheet del componente (www.linear.com/docs/3957) ci mostra come scegliere i pochissimi componenti esterni richiesti per realizzare un efficientissimo e miniaturizzato elevatore di tensione. Essenzialmente questi consistono nell'induttanza che immagazzina

l'energia, nei due condensatori di filtro all'ingresso e all'uscita, nelle due resistenze per la selezione di tensione d'uscita e, opzionalmente, nel diodo rettificatore Schottky; tale componente è opzionale in quanto l'LTC3400 ha già al proprio interno un diodo

raddrizzatore.

La Linear Technology distribuisce l'integrato in due versioni: LTC3400 e LTC3400B: il primo è caratterizzato dalla capacità di passare automaticamente nel funzionamento burst mode per ridurre i consumi con bassi cari-

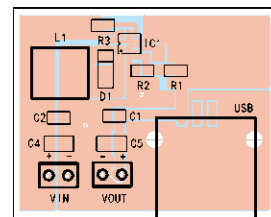
[REGOLATORE con LTC3400]

Elenco Componenti:

R1, R3: 1 Mohm (0805)
R2: 330 kohm (0805)
C1 $\div$ C3: 100 nF ceramico (0805)
C4, C5: 4,7 μ F ceramico (1206)
D1: MBRS130LT3
IC1: LTC3400ES6#TRMPBF
L1: Bobina 10 μ H

Varie:

- Connettore USB-A da CS
- Strip maschio 2 vie (2 pz.)
- Circuito stampato



chi modificando la frequenza di commutazione; il secondo invece lavora a frequenza fissa anche per carichi ridotti. La frequenza di commutazione relativamente elevata ci permette di utilizzare induttanze piuttosto compatte, riducendo ulteriormente i costi, pesi e dimensioni dei circuiti finali. La **Fig. 6**, estratta dal data-sheet, mostra la correlazione tra il valore della corrente in uscita e quello dell'induttanza L1; notiamo come il miglior compromesso tra valore massimo di corrente in uscita e dimensioni si ottenga con 10 μH , che è il valore da noi scelto. Per quanto riguarda la tensione in uscita, questa è data da:

$$V_{\text{out}} = 1,23 \times (1 + R1/R2)$$

con i valori scelti di R1 (1 Mohm) e R2 (330 kohm) ammonta a 5V.

REALIZZAZIONE PRATICA

I tre circuiti sono stati realizzati come breakout board per agevolare l'integrazione in altri sistemi. Per ridurre le dimensioni e facilitarne l'utilizzo in qualunque applicazione, le unità sono state realizzate con componentistica SMD. Chi vorrà realizzare questi dispositivi in autonomia, potrà farlo utilizzando i file con le tracce

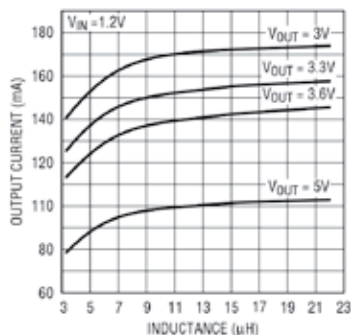


Fig. 6

per la foto incisione sul nostro sito Internet (www.elettronica.in.it).

Come per tutti i montaggi SMD, è richiesta una buona manualità ed esperienza in questo tipo di realizzazioni, nonché l'attrezzatura necessaria, consistente in un saldatore di buona qualità (si consiglia una stazione saldante con stilo a bassa tensione, tipicamente 24V) dotato di punta di dimensione adatta alla tipologia di componenti da saldare, set di pinzette con caratteristiche adatte all'impiego con componenti a montaggio superficiale e flussante (come ad esempio il Kester flux 951 con dosatore di tipo a "penna", disponibile presso la Futura Elettronica di Gallarate). Una volta in possesso del materiale necessario, si consiglia di iniziare a saldare i componenti passivi, per poi passare agli integrati.

In questo caso l'uso del flussante vi aiuterà ad evitare di saldare erroneamente tra di loro i piedini vicini. Utilizzate la pinzetta per mantenere in posizione il dispositivo e iniziate a saldare due piedini diametralmente opposti per mantenere fermo l'integrato. Procedete poi a saldare i piedini rimanenti. Prestate attenzione a non scaldare eccessivamente (oltre il necessario) i componenti, specialmente gli integrati, che altrimenti potrebbero subire danni irreversibili.



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono di facile reperibilità. Il regolatore boost (cod. LTC3400ES6) costa Euro 9,80. Il regolatore buck (cod. LTC3642) è disponibile a Euro 11,00, mentre il selettore di carico (cod. LTC4412IS6) è in vendita a Euro 5,90. Tutti i prezzi si intendono IVA compresa.

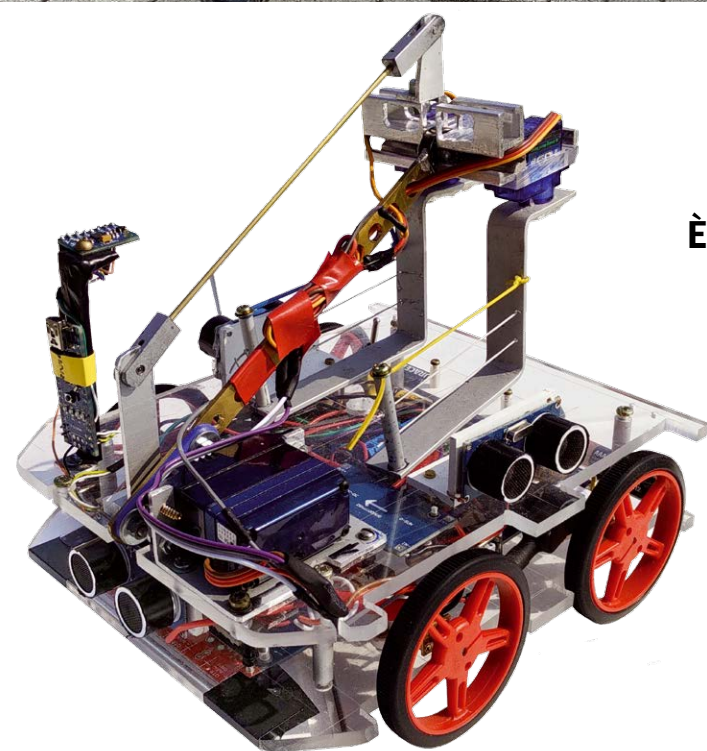
Il materiale va richiesto a:
Futura Elettronica, Via Adige 11,
21013 Gallarate (VA)
Tel: 0331-799775 • Fax: 0331-792287
<http://www.futurashop.it>

**Tutti i nostri progetti,
i master, i firmware
e tanto altro ancora su**
www.elettronica.in.it
Collegati subito!



Per la prima volta in una finale nazionale della Robocup Jr i robot autocostruiti basati su Arduino battono quelli realizzati con la piattaforma Lego. È successo durante la settima edizione della Robocup Jr Italia che si è appena conclusa a Busto Arsizio.

di ARSENIO SPADONI



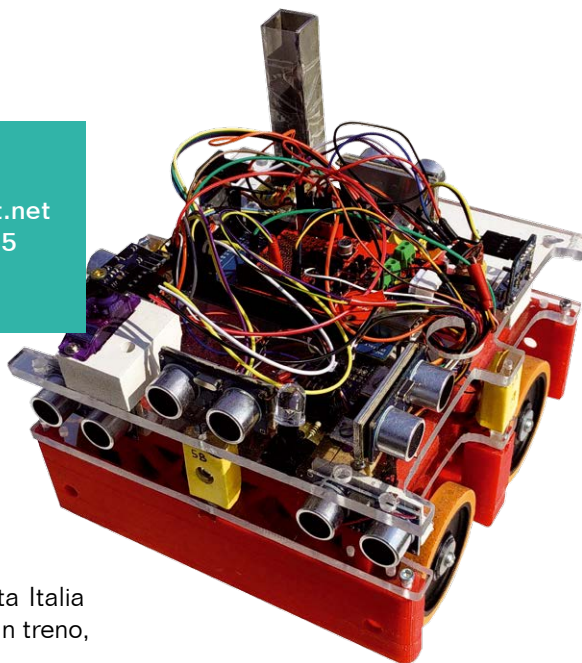
Si è conclusa a Malpensa Fiere, in provincia di Varese, la settima edizione della Robocup Jr Italia 2015, l'evento più importante legato alla Robotica Educativa che si svolge nel nostro paese e che vale anche – per alcune categorie –

quale gara di qualificazione per l'accesso ai campionati mondiali che quest'anno si svolgeranno a Hefei, in Cina. La manifestazione – di cui eravamo main sponsor – nonché parte attiva nel comitato organizzativo locale, ha visto la presenza di



Amico Robot

www.amicorobot.net
il 21 maggio 2015
a Milano.



quasi 140 team provenienti da tutta Italia che sono arrivati a Malpensa Fiere in treno, bus, aereo e auto.

Un record assoluto di presenze che ha superato persino quello di Pescara 2013. Una formula sicuramente da ripensare che ha messo a dura prova la struttura, gli organizzatori nonché le stesse squadre. Da anni si parla di gare di selezione a livello regionale (che per la verità in alcuni casi si svolgono, come nel Veneto), ma finora i numeri della finale nazionale continuano a crescere, specie in alcune specialità come nella Rescue Line U19 (ex Rescue A) e nella Rescue Maze (ex Rescue B). Quest'anno, poi, da gennaio sono cambiate molte

regole che hanno, di fatto, costretto a modificare radicalmente i campi di gioco utilizzati nella Rescue Line. Per riuscire a portare a termine la manifestazione nei

Robot & Makers

6 e 7 giugno 2015
a Novogro (MI).



tempi previsti si è dovuto fare ricorso a ben 11 campi per la Rescue Line e 2 per la Rescue Maze.

Dal punto di vista dei risultati, ancora una volta le squadre del Triveneto hanno dominato le specialità più tecniche (Rescue e Soccer) mentre molto significativi sono stati i risultati delle squadre del Centro Sud nelle gare Theatre & Dance.

Comunque è dal punto di vista tecnico che l'edizione di Malpensa Fiere ha segnato un punto di svolta nelle gare di robotica con la vittoria, per la prima volta in assoluto, di un robot autocostruito nella specialità più importante e con maggior numero di partecipanti, la Rescue Line. Non solo, il robot era basato su una italianissima scheda Arduino (precisamente una Mega). Un robot agile e ben fatto, realizzato dal team

Ritorna Officine Robotiche - 23 e 24 Maggio - a Roma, presso i nuovi spazi di SPQwoRk, un evento gratuito a cui tutti possono partecipare, dedicato alla Robotica amatoriale e non solo, per tutti coloro che hanno voglia di incontrare persone con la loro stessa passione, conoscere il mondo della Robotica, scoprire nuovi progetti, seguire workshop sul tema e partecipare a competizioni robotiche di Line Follower.



Espositori - Officine Robotiche 2015 - 23 e 24 maggio



Workshop - Officine Robotiche 2015 - 23 e 24 maggio

Espositori

Workshop

Officine Robotiche
<http://2015.officinerobotiche.it>
il 23 e 24 maggio a Roma.

RescueLine_Galilei del Liceo Scientifico "Galileo Galilei" di Trento, composto da Francesco Ballerin, Daniele Gottardini, Emanuele Pallaoro con la supervisione del docente Andrea Cristofori.

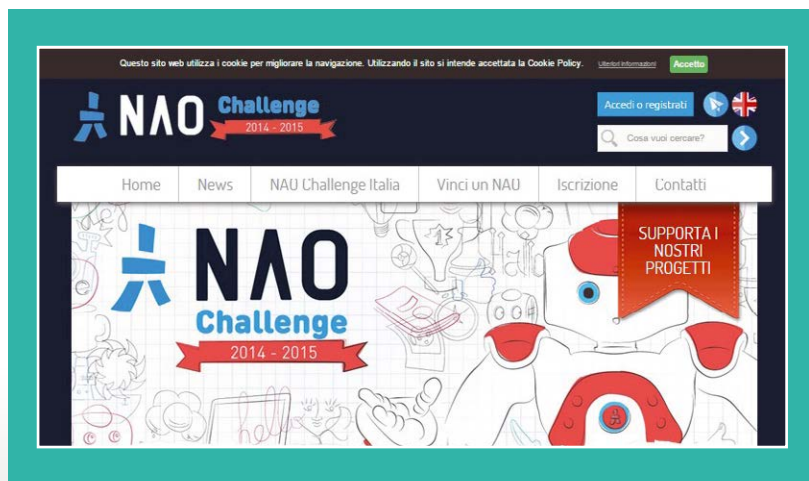




Oltre che aggiudicarsi il diritto di accedere alla finale in Cina, il team del Galilei ha vinto anche il primo premio offerto dalla nostra società, una stampante 3D Vertex con doppio estrusore.

Ricordiamo che in questa categoria il robot deve – in maniera completamente autonoma – seguendo una linea, raggiungere una stanza nella quale sono presenti delle vittime rappresentate da palline rivestite di stagnola. Nel percorso, il robot incontra ostacoli di varia natura che deve identificare e superare correttamente, deve affrontare salite e discese, fino a raggiungere l'area di recupero dove si trovano le vittime, identificarle e raccoglierle una ad una, per poi individuare la zona della stanza dove metterle in salvo. Realizzare un robot in grado di compiere una missione di questo tipo non è assolutamente banale, specie per i ragazzi delle scuole medie superiori: sono necessarie infatti approfondite conoscenze di meccanica, elettronica, programmazione e... anche un po' di strategia durante la gara. Per questo motivo, in passato, hanno sempre vinto i robot realizzati con la piattaforma Lego dove le problematiche legate alla meccanica, all'elettronica e alla sensoristica sono già risolte in partenza.

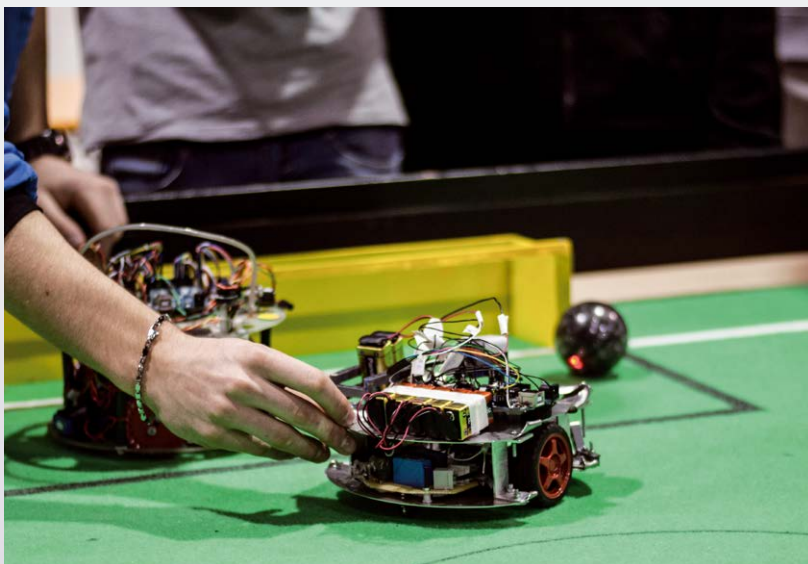
Ma quest'anno il Galilei di Trento ha superato se stesso. Infatti il team RescueMaze_Galilei formato da Andrea Dalla Costa, Luca Donini, Stefano Huber, Francesco Pedrotti, Filippo Quattrocchi ha vinto anche nella categoria Rescue Maze, sempre con un robot autocostruito e basato su Arduino. Un robot sicuramente meno bello del primo ma non per questo meno efficiente. Anche per il team della Maze del Galilei si sono spalancate le porte della Cina: anche questa gara era infatti valida come accesso alla finale mondiale.



NAO Challenge
<http://nao.scuoladirobotica.it/>
 il 23 e 24 maggio
 a Bologna.



Da Trento partiranno per la Cina anche i ragazzi della Rescue Line U14: infatti al primo ed al secondo posto di questa categoria si sono piazzate rispettivamente le squadre Bronzetti Again e Bronzetti Plus della scuola media Bronzetti-Segantini di Trento. In questo caso (come del resto in tutte le competizioni U14) i robot utilizzati sono dei Lego, modificati e programmati



quel tanto che basta per sconfiggere gli avversari.

Per quanto riguarda la categoria Soccer (unica non valida come qualificazione per i mondiali), dopo una serie di partite molto accese (in tutti i sensi), hanno vinto i ragazzi del Volterra S1, la squadra dell'ITIS Vito Volterra di San Donà di Piave (VE), guidata dal nostro collaboratore e docente di quella scuola Mirco Segatello.

Questa è l'unica specialità della Robocup Jr dove le qualificazioni si svolgono durante la Rome Cup.

Nella categoria Dance U19 si è classificato al primo posto la squadra del Liceo Democrito di Roma, team che in tutte le precedenti edizioni della Robocup Jr si è sempre piazzato ai primi posti. Anche in questo caso la vittoria a Malpensa Fiere vale il visto per i mondiali in Cina.



RoboGame

Il 9 maggio a Livorno presso l'ITI Galilei.

ROBOGAME

2015

sabato 9 Maggio

ITI "G. Galilei"
Livorno

ore 8.30	registrazioni e prove
ore 10.00	gare di Minisumo, Labyrinth, Line-Follower e Rescue Line
ore 13.00	premiazioni

Per quanto riguarda la Dance U14, si sono piazzati al primo posto ex-equo le squadre Zumbini Team (IC Zumbini di Cosenza) e Happydancers dell'IC Japigia1 Verga di Bari. Proprio la città di Bari ha avanzato la propria candidatura per la prossima finale italiana della Robocup Jr: per il momento ragazzi e docenti debbono "accontentarsi" dei mondiali in Cina.

Numerosi gli altri premi e i riconoscimenti vari offerti dall'organizzazione e dagli sponsor; sono stati premiati anche i bimbi della scuola dell'Infanzia e delle elementari provenienti da Mortara e Tortona.

Da segnalare anche come quest'anno le interviste alle squadre, previste dal regolamento internazionale, siano state condotte con metodo e con criteri scientifici da alcuni team della vicina Università LIUC-Carlo Cattaneo che ha inserito questa attività nel più ampio ambito di un progetto europeo di ricerca sul "fai da te digitale" per capire come queste nuove tendenze potranno migliorare la nostra società, valorizzando la creatività degli individui e le sempre più diffuse possibilità di fare innovazione. □



Roboval

www.roboval.it,
23 e 24 maggio
a Verona.



Progetto Galileo: in orbita due nuovi satelliti

La costellazione del sistema europeo di navigazione satellitare Galileo cresce con il lancio di ulteriori due satelliti (il settimo e l'ottavo) avvenuto con successo base europea di Kourou (Guyana Francese) con una Soyuz.

Alle 22:46 – ora italiana – del 27/3 dalla base ESA di Kourou è puntualmente cominciato il viaggio di FM03 e FM04 a bordo del vettore Soyuz ST-B (Fregat), appositamente configurato. Esattamente 3 ore e 48 minuti dopo il lift-off la quarta coppia del sistema satellitare di navigazione europeo Galileo veniva posizionata nella propria orbita, a circa 23.500 km di quota.

Con questo lancio, il primo della serie di tre prevista nel 2015 (gli altri sono in agenda per fine settembre e per dicembre, secondo quanto è stato fissato a fine gennaio alla Space Policy Conference di Bruxelles) sale dunque a 4 il numero complessivo dei satelliti pienamente

operativi (FOC, Full Operational Capability) della costellazione.

Il successo di questo lancio con il corretto posizionamento di FM03 e FM04 è quello che ci voleva dopo l'anomalia occorsa nello scorso agosto ai satelliti 5 e 6, finiti su un'orbita ellittica diversa rispetto a quella prevista.

Le operazioni di recupero della coppia 5-6 erano iniziate lo scorso ottobre, quando fu deciso di ottimizzare per quanto possibile l'orbita dei due satelliti al fine di poterli testare e successivamente decidere del loro utilizzo. A fine novembre, il Sat-5 è stato posto su un'orbita migliorata con

perigeo innalzato a circa 19.000 km consentendo la ricezione del segnale e l'attivazione delle apparecchiature di bordo.

A metà gennaio, sono state quindi avviate le manovre e le attività di test in orbita (IOT) per il Sat-6, l'operazione di recupero si è poi conclusa con successo lo scorso 13 marzo. La completa funzionalità dei satelliti 5 e 6 avverrà in modo graduale nel corso del quadriennio 2015-2018, passando attraverso tre scenari consecutivi: due prevedono la produzione dei messaggi di navigazione senza informazioni di almanacco mentre il terzo e ultimo vedrà l'aggiunta delle informazioni.

www.asi.it



Anche in Italia i mobili IKEA per la ricarica wireless

Dopo aver annunciato alcuni mesi fa che avrebbe commercializzato mobili con sistema di ricarica wireless per smartphone ed aver iniziato la distribuzione di questi prodotti negli Stati Uniti e in altri Paesi, finalmente i sistemi di ricarica wireless arrivano anche nel nostro Paese.

Per gli smartphone che supportano già la ricarica wireless basta scegliere il o i mobili che si desidera tra una vasta scelta di modelli; per quei cellulari, invece, che ne sono privi, è disponibile una speciale cover (VI-TAHULT) compatibile con la maggior parte degli smartphone in commercio.

Alcuni mobili hanno già al loro interno il sistema di ricarica wireless mentre per creare una postazione di ricarica personalizzata esiste il caricabatteria wireless JYSSEN che può essere inserito in qualsiasi mobile esistente.

Esistono anche delle basi di ricarica con un design semplice e lineare che si adatta a qualsiasi ambiente della casa: possono



essere appoggiate sul davanzale di una finestra, sulla cassetta in camera da letto e persino sul piano di lavoro della cucina. IKEA ha scelto i trasmettitori Wireless Power di Integrated Device Technology (IDT) per realizzare i caricabatteria wireless; nella fattispecie sono stati utilizzati i trasmettitori a induzione magnetica P9030 conformi alle specifiche tecniche Qi definite da WPC (Wireless Power Consortium).

www.ikea.it

Ai Fori Imperiali i LED illuminano la storia dell'Impero Romano

Uno dei siti archeologici più importanti al mondo, simbolo di una storia che ancora oggi, dopo 2.000 anni, non finisce mai di sorprendere e conquistare, si arricchisce di nuovo fascino anche nelle ore notturne grazie ad un innovativo sistema di illuminazione a LED che ne esalta la bellezza e la maestosità.

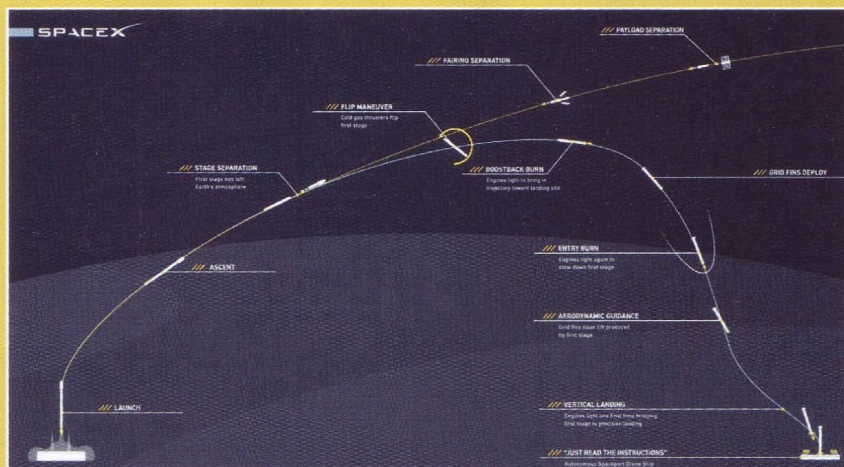
L'area compresa tra il Campidoglio, il Palatino e il Quirinale è infatti protagonista di un grande intervento d'illuminazione e valorizzazione progettato dal maestro Vittorio Storaro, vincitore di tre premi Oscar per la fotografia con capolavori come "Apocalypse Now", "Reds" e "L'ultimo Imperatore" e dall'architetto Francesca Storaro che ha realizzato anche il progetto illuminotecnico.

L'intervento è stato voluto dal Comune di Roma Capitale, curato e realizzato con il sostegno di Acea spa e di Unilever per la parte progettuale. L'obiettivo è stato quello di creare un impianto d'illuminazione artistica permanente per raccontare, attraverso la luce, le gesta dei vari Imperatori e la grande storia dell'Impero Romano. In particolare l'intervento coinvolge il Foro di Augusto, il Foro di Nerva e il Foro di Traiano.

Nell'illuminazione dei Fori Imperiali sono stati utilizzati solo apparecchi con tecnologia LED e in particolare il progettista ha individuato nei corpi illuminanti ERCO gli strumenti ideali alla realizzazione dell'ideazione luministica.

www.erco.com

Falcon9 & Dragon, lancio riuscito, recupero fallito

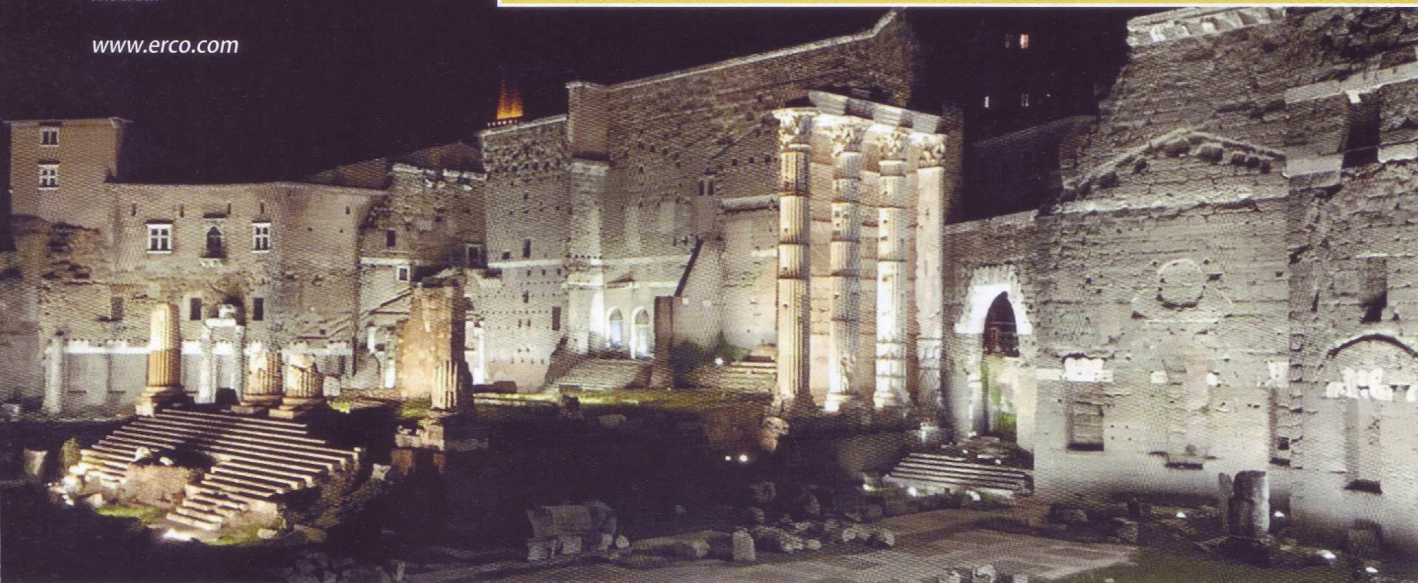


La nuova missione spaziale che ha lo scopo di rifornire la Stazione Spaziale Internazionale (ISS) con materiale destinato a nuovi esperimenti, dopo il rinvio di 24 ore a causa delle avverse condizioni meteo, è partita regolarmente alle ore 22:10 del 14 aprile dalla base di Cape Canaveral in Florida. Il razzo vettore Falcon 9 ha lasciato nell'orbita corretta la navetta Dragon che col suo carico di circa 2 tonnellate è ora in viaggio verso la ISS. Si tratta del sesto lancio con il vettore Falcon 9 di SpaceX, la società americana che ha firmato un accordo con la NASA per l'invio di rifornimenti alla ISS per almeno 12 missioni. Anche questa volta è stato tentato il recupero del razzo vettore Falcon 9, che nei programmi dell'azienda Space X guidata da Elon Musk, dovrebbe essere riutilizzato per altri lanci. Purtroppo anche in questo caso, pur centrando l'apposita

piattaforma che lo attendeva al largo delle coste americane, all'ultimo momento il vettore si è rovesciato e ha preso fuoco. Rispetto ai tentativi precedenti, grazie anche alla nuova piattaforma molto più stabile, tutto è filato liscio fino a pochi istanti prima della conclusione del tentativo quando il razzo non è riuscito a fermarsi in posizione verticale.

Tornando alla capsula Dragon, questa resterà agganciata alla ISS per circa 5 settimane e fornirà agli astronauti il materiale necessario per gli ultimi tre esperimenti della missione Futura. Nel dettaglio, oltre al materiale di supporto per gli astronauti, i tre preziosi payload per Samantha Cristoforetti sono: Cell Shape and Expression (Cytospace), ISSpresso e Nanoparticles and Osteoporosis (NATO).

www.spacex.com



L'orologio atomico più preciso al mondo

L'orologio atomico più preciso del mondo è stato messo a punto dal team di ricercatori coordinato da Jun Ye, del centro di ricerca americano JILA, del National Institute of Standards and Technology (Nist) e dell'università del Colorado a Boulder. L'orologio, realizzato con atomi di stronzio intrappolati in un reticolo di luce laser, è di una precisione inimmaginabile, visto che non perde né anticipa un secondo nell'arco di 15 miliardi di anni. Rispetto al precedente orologio ottenuto dal Nist, questo è tre volte più preciso. Oggi giorno poter disporre di orologi ato-

mici estremamente precisi riveste una notevole importanza, anche al fine di una misurazione delle costanti della natura. Molti sistemi tecnologici come i navigatori satellitari GPS di nuova generazione o i nuovi sensori per la gravità, necessitano di una precisione di calcolo nel tempo sempre maggiore. L'orologio atomico del Nist, che magari un giorno potrà persino testare la teoria della relatività di Einstein, è descritto su Nature Communications.

www.nist.gov

Sicral 2, il satellite italiano per comunicazioni militari è in orbita

Dopo due rinvii è stato lanciato con successo Sicral 2, il satellite per le telecomunicazioni frutto della cooperazione tra i ministeri della difesa francese e italiano. Il lancio è stato effettuato alle 22 (ora di Roma) del 26 aprile dalla base europea di Kourou in Guyana Francese. Il lancio era stato inizialmente fissato per il 14 aprile ed è stato rinviato a causa di un problema tecnico al vetto-

re Ariane 5, a bordo del quale era presente anche il satellite norvegese Thor 7 della Telenor che garantirà il servizio televisivo per l'Europa centrale e le comunicazioni a banda larga per il settore marittimo.

SICRAL (Sistema Italiano per Comunicazioni Riservate e Allarmi) è il sistema satellitare italiano per le comunicazioni militari, caratterizzato da flessibilità e versatilità di impiego. È in grado di garantire l'interoperabilità tra le reti della Difesa, della sicurezza pubblica, dell'emergenza civile e della gestione e controllo delle infrastrutture strategiche.

SICRAL 2 è un satellite geostazionario operante nelle bande UHF e SHF, in grado di potenziare le capacità di comunicazioni satellitari militari già offerte dalla coppia di satelliti SICRAL 1 e SICRAL 1B lanciati rispettivamente nel 2001 e nel 2009. In particolare SICRAL 2 supporterà le comunicazioni satellitari delle forze armate italiane e francesi, anticipandone tutte le esigenze di crescita e di sviluppo previste per i prossimi anni. Il satellite si posizionerà a 36.000 chilometri dalla Terra e avrà una vita operativa di 15 anni.

www.difesa.it

Supercomputer da un milione di operazioni al secondo per la ricerca italiana



Dopo un breve periodo di collaudo e test, è entrato in funzione a pieno regime Galileo, il nuovo supercomputer installato al Cineca di Bologna. L'investimento è il risultato di un'azione di sviluppo congiunto con INFN e Università di Milano-Bicocca, che hanno in parte cofinanziato i costi di acquisizione del nuovo sistema.

Galileo si affianca a Fermi, il supercomputer per la ricerca accademica più potente in Italia, su cui è stata avviata la collaborazione tra Cineca e INFN. L'accordo quadro, sottoscritto nel 2012, prevede lo svolgimento di attività di ricerca e sviluppo nel settore del calcolo scientifico ad alte prestazioni negli ambiti della Fisica delle Alte Energie, della Fisica Astroparticellare e della Fisica Nucleare.

I positivi risultati derivati dalla collaborazione in corso hanno indotto INFN a un ulteriore investimento a cui si è associata anche l'Università Milano-Bicocca.

Le risorse economiche congiuntamente mobilitate hanno consentito l'acquisizione di Galileo, un calcolatore IBM/Lenovo Nextscale configurato con 516 nodi per un totale di 8256 processori Intel Haswell, 768 acceleratori floating point Intel Phi e 80 acceleratori floating point nVidia K80; queste risorse consentono al nuovo supercomputer di esprimere una potenza computazionale di picco di oltre 1 PetaFlop/s (un milione di miliardi di operazioni al secondo).

Galileo è dedicato al calcolo scientifico e ingegneristico ed è a disposizione dei ricercatori afferenti agli istituti di ricerca e alle università italiane.

www.cineca.it

CORSO MPLAB X



di FRANCESCO FICILI
e VINCENZO GERMANO

Continuiamo il nostro viaggio alla scoperta di MPLab X, il nuovo ambiente di sviluppo integrato prodotto e distribuito da Microchip Technology, che soppianta il vecchio MPLab IDE. In questa puntata ci occupiamo di come realizzare applicazioni embedded multitasking con i PIC32 che sfruttano l'USB Device come protocollo di comunicazione.

Nelle puntate precedenti abbiamo esposto le caratteristiche ed il funzionamento dei microcontrollori PIC32 della Microchip e proposto esempi per utilizzare in pratica varie loro periferiche, come ad esempio SPI e UART, per poi passare all'analisi e all'implementazione di uno scheduler che ci permetterà di affrontare la programmazione embedded multitasking cooperativa. A questo punto abbiamo tutti gli strumenti e gli elementi per poter passare a spiegare, nel corso di questa puntata, funzionamento e utilizzo di una delle più comuni e utilizzate periferiche: l'USB Device.

LO STACK USB MICROCHIP

Lo standard di comunicazione seriale USB (Universal Serial Bus) venne progettato per consentire di collegare periferiche eterogenee ad un Personal Computer usando una sola interfaccia standardizzata, un solo tipo di connettore, ma anche per implementare la funzionalità "plug and play" consentendo di collegare/scollegare i dispositivi senza dover necessariamente riavviare il computer. Negli anni l'USB si è affermato come uno dei più diffusi e comuni standard di comunicazione e ad oggi è presente nella maggior parte dei dispositivi elettronici, come

v2013-02-15	Windows	Microchip Libraries for Applications	
v2013-02-15	Mac OS X	Microchip Libraries for Applications	
v2013-02-15	Linux	Microchip Libraries for Applications	
v2013-02-15		Help Files	
v2013-02-15		Release Notes ①	

Fig. 1 - File di download per le MLA.

mouse, tastiere, memorie di massa a stato solido e a disco rigido, scanner per immagini, macchine fotografiche digitali, stampanti, navigatori satellitari, televisori, cellulari, smartphone ecc.

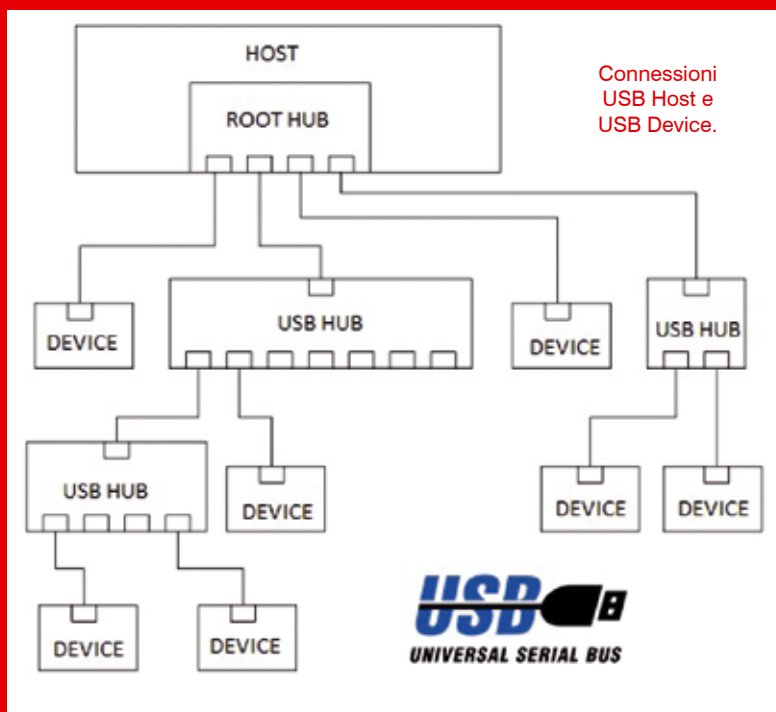
Una delle caratteristiche principali dell'USB è la sua **architettura asimmetrica** che collega un singolo Host e più periferiche Device mediante una struttura ad albero, grazie a

dei dispositivi chiamati hub (concentratori). In queste pagine affronteremo solo la parte riguardante la configurazione dell'USB Device (ricordiamo che l'USB prevede due dispositivi: l'Host e il Device; la loro funzione è spiegata nell'apposito riquadro "USB Host e USB Device") analizzando lo stack messo a disposizione dalla Microchip e integrandolo nella nostra applicazione; rimandiamo alla

USB HOST E USB DEVICE

La prima versione dell'USB venne realizzata nel gennaio del 1996 e supportava collegamenti a solo 1,5 Mbit/s (velocità adeguate per mouse, tastiere e altri dispositivi "lenti") e una lunghezza del cavo massima di 3 metri. Con il tempo la sua evoluzione ha visto vari aggiornamenti, fino ad arrivare all'odierno USB 3.0, che è in grado di trasferire dati a una velocità di trasferimento di 4,8 Gbit/s (l'equivalente di circa 600 MB/s). Venne progettato per consentire a più periferiche di essere connesse usando una sola interfaccia standardizzata e un solo tipo di connettore, ma anche per migliorare la funzionalità "plug and play" consentendo di collegare o scollegare i dispositivi senza dover riavviare il computer. Una delle caratteristiche principali dell'USB è la sua architettura asimmetrica che consiste in un singolo gestore (identificato come Host) e più periferiche (identificate come Device) collegate ad albero, attraverso dei dispositivi chiamati hub (concentratori), come mostrato nella figura a lato; perciò nel protocollo di comunicazione si possono identificare due entità: Host e Device. Il primo comunica con il dispositivo e riceve dati di ingresso da esso rispetto alle azioni eseguite dall'utente, mentre il secondo è l'entità che interagisce direttamente con l'utente e può essere

-ad esempio- una tastiera o un mouse. Nei computer, il driver analizza i dati e consente l'associazione dinamica dei dati di I/O con la funzionalità dell'applicazione. Queste caratteristiche/funzionalità hanno permesso una rapida innovazione e lo sviluppo di tale periferica, ma anche la diversificazione di nuovi dispositivi.



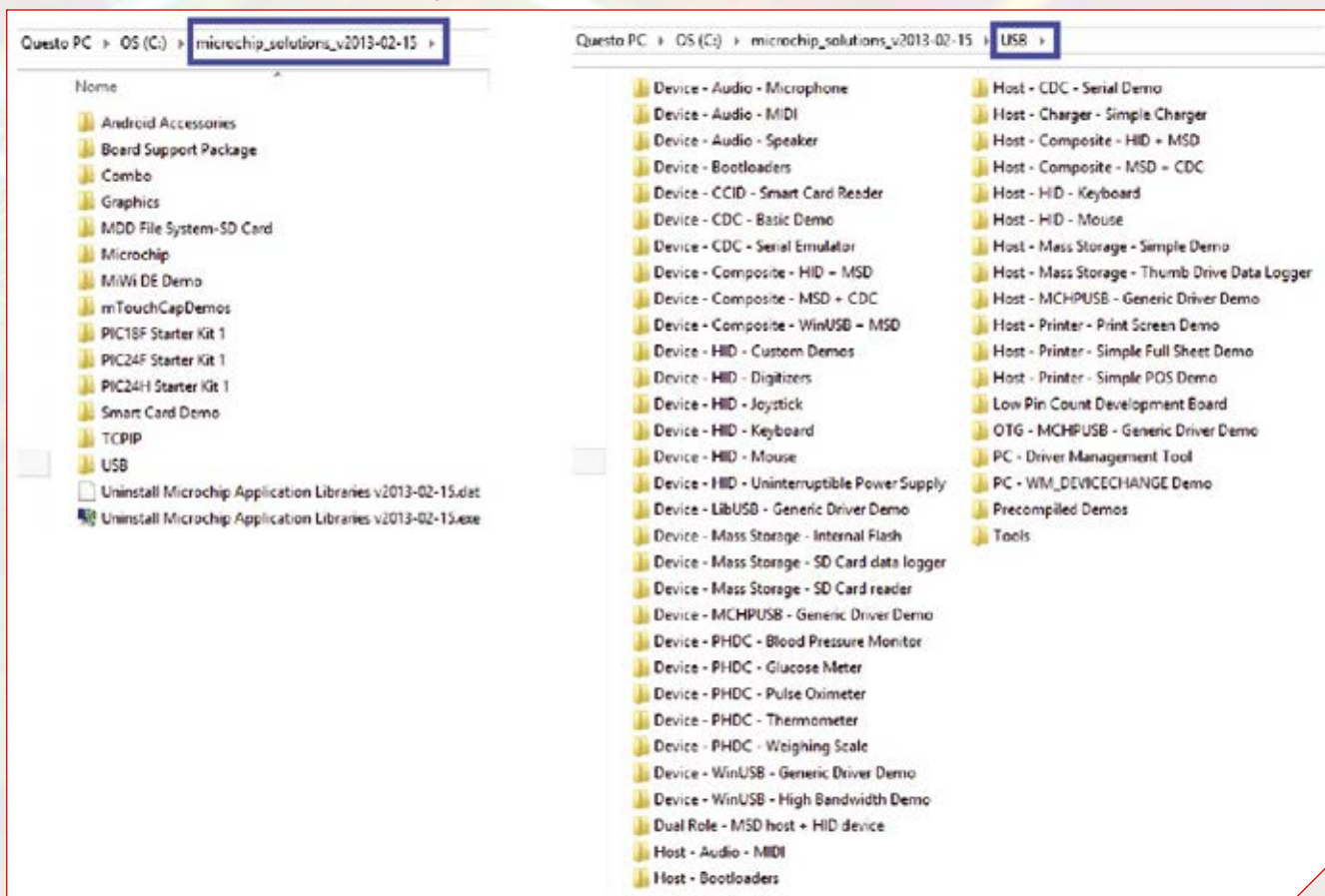
prossima puntata la trattazione e la configurazione dell'USB Host. Concentrando la nostra attenzione sullo stack Microchip, il primo passo da effettuare è scaricare dal sito ufficiale il pacchetto d'installazione delle "Microchip Libraries for Applications" (MLA) che sono le librerie di riferimento da cui poter attingere per importare tutto il necessario nei nostri progetti; esse migliorano l'interoperabilità tra le applicazioni che generalmente utilizzano più di una libreria e contengono gli strumenti base per lo sviluppo della maggior parte delle applicazioni (dal codice sorgente ai driver, dimostrazioni, documentazioni e utility), tutto preconfezionato e pronto per essere utilizzato. Per questa e le future puntate del corso, abbiamo preferito non utilizzare l'attuale versione (v2014-07-22) delle MLA perché non presenta il supporto d'interesse legato ai PIC32 (essendo nato un Framework dedicato "MPLAB® Harmony software suite") ma una delle versioni precedenti e nello specifico la

"v2013-02-15", come si vede dalla Fig. 1.

Un consiglio è quello di scaricare anche i file di "Help" perché molto utili per comprendere meglio la struttura del pacchetto complessivo delle MLA. In ogni caso, una volta scaricato e installato l'eseguibile, la situazione che ci si presenta nella cartella di destinazione selezionata è riportata in Fig. 2 (lato sinistro), da cui si evince l'organizzazione principale delle MLA. Si può notare come ci siano cartelle legate alla grafica, al TCPIP, USB e così via; all'interno di ogni cartella si possono trovare librerie, codici sorgenti, driver e utility, pronti per essere integrati nei nostri progetti. Notate che sebbene l'utilizzo di queste parti di codice messe a disposizione dalla Microchip possa sembrare immediato, non lo è, perché in ogni caso si deve comunque avere una buona conoscenza di quello che si vuole utilizzare.

La cartella che ci interessa per questa puntata del corso è ovviamente quella USB, visibile in

Fig. 2 - Struttura delle MLA e della cartella USB.



Listato 1

```
// Hardware Configuration
#define USB_PING_PONG_MODE    USB_PING_PONG__FULL_PING_PONG

// Peripheral Configuration
#define MY_VID                0x04D8
#define MY_PID                0x0000
#define USB_POLLING
#define USB_PULLUP_OPTION     USB_PULLUP_ENABLE
#define USB_TRANSCEIVER_OPTION USB_INTERNAL_TRANSCEIVER
#define USB_EP0_BUFF_SIZE     8
#define USB_MAX_NUM_INT       (0+1)
#define USB_MAX_EP_NUMBER     6
```

Listato 1 - Esempio d'impostazione USB.

Fig. 2 (lato destro) che contiene i file d'interesse per la configurazione di questa periferica e permette una creazione abbastanza rapida di applicazioni in cui è necessaria la periferica USB; al suo interno troviamo altre sottocartelle di progetti di riferimento che coprono una gran varietà di applicazioni e casi di utilizzo.

CONFIGURARE UN PROGETTO USB DEVICE

Per il progetto pratico che vi proponiamo abbiamo fatto riferimento a quanto presente nella cartella "Device - HID - Joystick"; in questa si trova un insieme di interfacce firmware modulari che gestiscono la maggior parte della comunicazione USB. In Fig. 3 viene mostrato un tipico flusso di un programma USB. Ogni progetto di riferimento MLA è scritto per avere un ambiente cooperativo multitasking, perciò nessuna funzione "bloccante" deve essere utilizzata (come ampiamente discusso nella puntata precedente di questo corso). Scendendo maggiormente nei dettagli dello stack USB Device della Microchip, per avere un'idea di come funzioni a basso livello, concettualmente possiamo notare quattro blocchi fondamentali (fate riferimento alla Fig. 3):

- *main.c* che contiene la funzione *main()*, cioè un ciclo infinito con servizi che attuano diversi compiti, che possono essere logicamente considerati sia come compiti USB, sia come task utente ma anche come altro;
- *USBDevice.c*, in cui sono inserite tutte le funzioni riguardanti l'USB, che vengono gestite dalla funzione *USBTasks()*, in linea con la filosofia del multitasking cooperativo spiegato nella puntata precedente;
- *hid.c* contenente le funzioni (macro) che possono essere usate da un utilizzatore umano per controllare il funzionamento del

sistema;

- *USBDescriptor.c*, che contiene le informazioni del descrittore USB per il particolare dispositivo e tali informazioni variano in base all'applicazione; a riguardo va detto che l'host non ha bisogno di capire intrinsecamente o analizzare il descrittore nello specifico, inoltre, generalmente esiste una "stringa descrittore" che fornisce una descrizione testuale del dispositivo.

Si può pensare all'*USBDescriptor.c* come a un array di byte di dati che descrivono i pacchetti del dispositivo, includendo il numero di pacchetti supportati dal dispositivo, la dimensione dei pacchetti e lo scopo di ogni byte/bit nel pacchetto.

Dopo aver installato l'eseguibile MLA, la cartella di destinazione relativa a questi file sorgente si trova nel path seguente: "*<Install Directory>\>\Microchip\USB*". Per poter utilizzare questi file bisogna importare nel proprio progetto i corrispondenti file ".h", come verrà mostrato successivamente nell'esempio pratico.

Tra i file da importare per il progetto, oltre ai quattro cardine appena illustrati, altri due meritano un'attenzione particolare: *l'HardwareProfile.h* e *l'usb_config.h*. Il primo dei due generalmente viene usato perappare diverse configurazioni hardware per una definizione comune di codice da utilizzare. Al suo interno vengono definite varie impostazioni come, per esempio, su quale porta utilizzare i LED, la velocità di clock della scheda, che pin usare per l'acquisizione dei segnali e via dicendo.

Da quanto appena detto risulta chiaro come *l'HardwareProfile.h* possa assumere un ruolo fondamentale in ogni progetto, anche per poter continuare a mantenere una struttura ordinata e coerente. Mentre per quanto riguarda il file di intestazione *usb_config.h*, come si può notare dal Listato 1 (un estratto del file), è un elemento chiave per la configurazione dello stack USB; infatti esso definisce vari parametri all'interno della pila, che determinano come lo stack deve funzionare e quali caratteristiche opzionali sono da includere. Ovviamente la modifica di queste opzioni può avere implicazione sulla dimensione del codice, utilizzo della RAM e throughput

dei dati. Come potete immaginare, senza un'opportuna definizione di questo file lo stack USB non funzionerà come previsto nella modalità da voi desiderata.

LA CLASSE USB HID

Dopo aver compreso quali sono le parti fondamentali dello stack Microchip e come esse collaborano tra loro, passiamo alla trattazione della classe *Human Interface Device* (HID), legata all'Universal Serial Bus (USB).

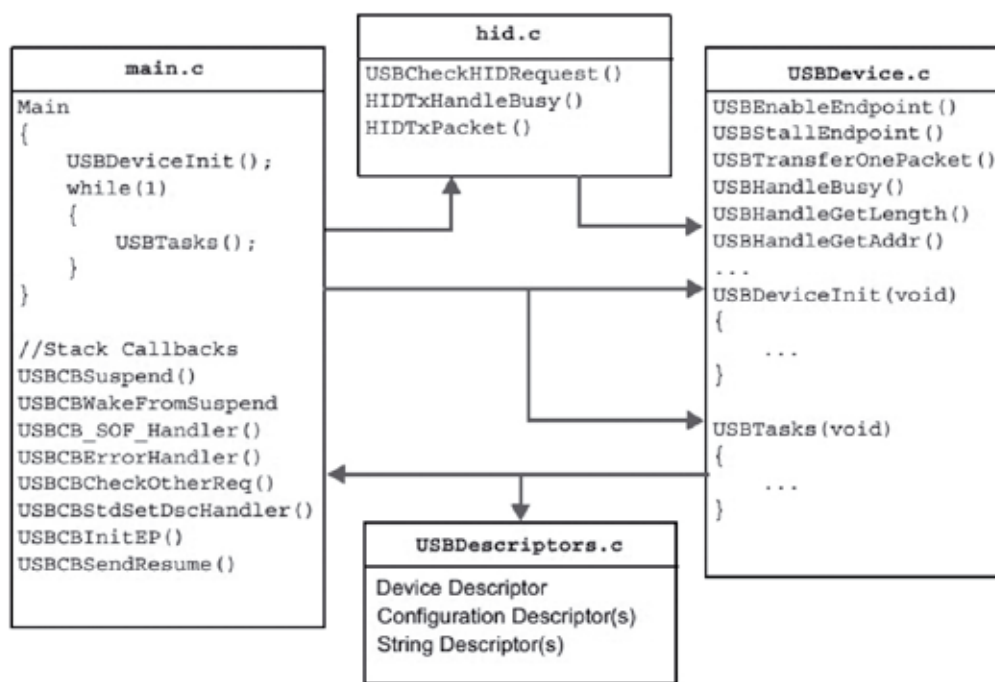
Lo standard HID, tra le altre cose, è stato adottato per consentire l'innovazione in dispositivi di input per computer e per semplificare il processo di installazione degli stessi, infatti oggi giorno la maggior parte dei dispositivi HID forniscono pacchetti di auto-installazione che possono contenere vari tipi e formati di dati. La classe HID è costituita da dispositivi che un essere umano può utilizzare per controllare il funzionamento dei sistemi come mouse, tastiera, joystick, pulsanti, interruttori e via dicendo.

È importante sapere che, oltre a fornire informazioni dalle interfacce umane, essa fornisce anche dati in output per indicare le azioni da parte del sistema operativo (computer). Per avere un'idea generale di come possa fun-

zionare una comunicazione di un dispositivo USB, possiamo affermare che è basata su *pipes* (tubi) o connessioni logiche dal controller host a un'entità logica, che si trova su un device e che viene identificata con il termine endpoint; visto che tali canali corrispondono 1-a-1 agli endpoint, i termini sono a volte usati come sinonimi.

Un dispositivo USB può avere fino a 32 endpoint (16 in, 16 OUT) ma è raro averne così tanti. Un endpoint è definito e numerato dal dispositivo durante la fase d'inizializzazione (il periodo immediatamente dopo il collegamento fisico noto come "enumeration") e quindi è permanente, mentre un canale può essere aperto e chiuso dopo una comunicazione. Utilizzando il firmware della Microchip appena presentato, la classe HID implementa le seguenti funzioni specifiche: USB Descriptor Table, Endpoint Configuration Table e Function Driver Table. Riguardo "*USB Descriptor Table*", ogni dispositivo USB deve sia fornire una serie di descrittori (strutture dati) che definiscono il dispositivo, sia informare l'host USB su quale driver di classe deve utilizzare. I descrittori dei dispositivi USB possono essere organizzati in tre gruppi: Device, Configuration e Strings.

Fig. 3 - Struttura concettuale di funzionamento dell'USB.



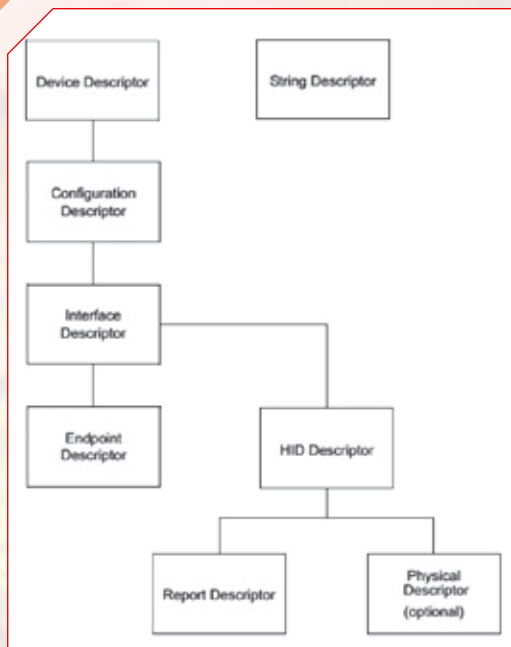


Fig. 4 - Descrittore USB per driver HID.

- *Device* identifica il tipo di dispositivo e fornisce il numero di possibili configurazioni.
- *Configuration* descrive i tipi di interfaccia e di endpoint utilizzati (inclusendo anche descrittori specifici della classe).
- *Strings*, che generalmente è opzionale, fornisce informazioni leggibili dall'utente che l'host può visualizzare.

La "Endpoint Configuration Table" viene utilizzata dallo stack USB per configurare correttamente tutti gli endpoint d'interfaccia e le impostazioni alternative, secondo quanto definito dalla tabella del particolare descrittore; identifica, inoltre, quali funzioni devono essere utilizzate per gestire l'evento che si verifica su ciascun endpoint. Infine, per quanto riguarda la "Function Driver Table", dal momento che un dispositivo può implementare più di una classe/driver della periferica USB di uno specifico fornitore, lo stack della Microchip usa una tabella per gestire l'accesso e supportare la funzione del driver. Ogni voce della tabella contiene le informazioni necessarie a gestire una singola funzione del driver. Ogni dispositivo USB ha una struttura descrittore a esso associata e può contenere più classi definite nel layer d'interfaccia, come ad esempio succede per l'HID. Una struttura descrittore ad albero è mostrata nella Fig. 4, la quale descrive un dispositivo di classe HID: come si vede, il descrittore HID indica quanti altri descrittori specifici seguono.

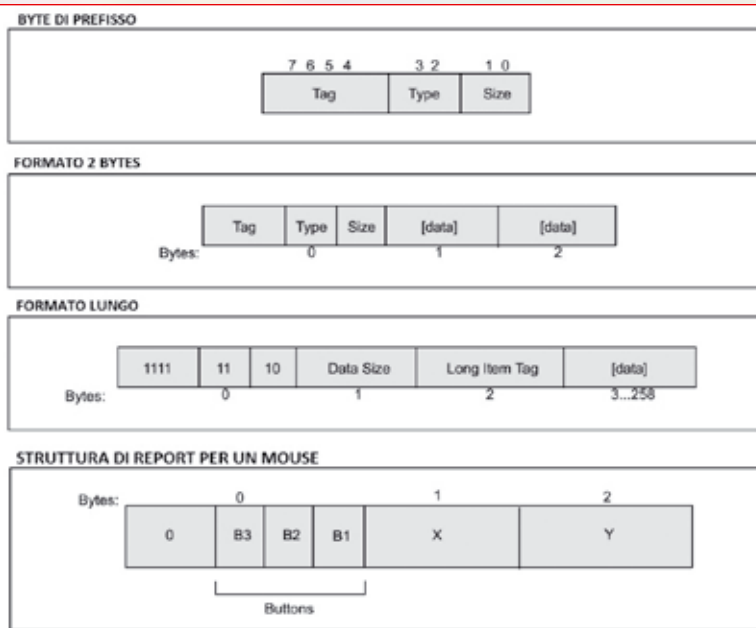


Fig. 5 - Formati dell'Item Report.

no. Notate che dev'essere presente almeno un "Report Descriptor", mentre quelli fisici sono opzionali. Un *Report Descriptor* descrive il formato e il significato di dati generati dal dispositivo, viene caricato dalla classe del driver HID dell'host utilizzando una richiesta specifica; dopo essere stato inizializzato, il dispositivo genera rapporti per indicare quando un utente interagisce con una periferica. È costituito da parti di informazione chiamate "Items" (elementi) e ognuno descrive un aspetto dei dati del report.

Generalmente un report di un item segue un formato costituito da un prefisso di un byte e un payload (informazioni utili); il primo contiene il tag, il tipo e la lunghezza del payload, come mostrato nella Fig. 5 (Byte di prefisso), nella quale sono anche riportati i due formati che può assumere un report: corto e lungo. Nell'esempio pratico che verrà mostrato successivamente verrà utilizzato il primo formato, mediante il quale sarà possibile inviare al computer gli spostamenti richiesti del mouse mediante l'interazione con un joystick (nel formato indicato come "struttura di report per un mouse").

È facile immaginare che i report di ingresso vengono utilizzati per inviare i dati sulle interazioni dell'utente con il dispositivo per l'host, mentre quelli di uscita vengono utilizzati dall'host per inviare dati al dispositivo di controllo, ad esempio un comando per accendere il LED.

Listato 2

```

/*****
* User Joystick
*****/
/* Data direction register */
#define JOY_UP_TRIS          TRISBbits.TRISB10
#define JOY_RIGHT_TRIS      TRISBbits.TRISB9
#define JOY_LEFT_TRIS       TRISBbits.TRISB12
#define JOY_DOWN_TRIS       TRISBbits.TRISB11
#define JOY_FIRE_TRIS       TRISBbits.TRISB13

/* Resource Alias */
#define JOY_UP               PORTBbits.RB10
#define JOY_RIGHT            PORTBbits.RB9
#define JOY_LEFT             PORTBbits.RB12
#define JOY_DOWN             PORTBbits.RB11
#define JOY_FIRE             PORTBbits.RB13

```

Listato 2 - Configurazione delle porte del microcontrollore legate al Joystick.

Infine, i report vengono usati dall'host per configurare correttamente il device.

LE MACRO USB

Con "Application Programming Interface" (API - Interfaccia di Programmazione di un'Applicazione), generalmente viene indicato un insieme di procedure disponibili al programmatore, raggruppate a formare un set di strumenti specifici per un determinato compito all'interno di un programma. La finalità è ottenere un'astrazione a più alto livello tra software a basso e quello ad alto livello semplificando così il lavoro di programmazione, permettendo di evitare ai programmatori di riscrivere ogni volta tutte le funzioni necessarie al programma dal nulla, ovvero dal basso livello, rientrando quindi nel più vasto concetto di riuso di codice.

Ci sono due livelli di API relative al HID USB: il livello USB e il livello di sistema operativo. Il primo è un protocollo utilizzato riguardante i device e serve per poter inviare al sistema operativo le loro caratteristiche/capacità e permettere a quest'ultimo di gestirle correttamente. Invece il livello del sistema operativo offre una visione al livello superiore delle applicazioni, che non hanno più bisogno di includere il supporto per i singoli dispositivi.

Nel nostro caso la funzione più rilevante che utilizzeremo anche nell'esempio pratico sarà la macro "HIDTxPacket" contenuta all'interno del file "usb_function_hid.h", che ha per definizione la seguente struttura:

```
#define HIDTxPacket USBTxOnePacket
```

Grazie a questa macro è possibile inviare dei dati specifici da un particolare endpoint e

Listato 3

```

/* Array of task to call */
void (*TaskArray[ACTIVE_TASK_NUMBER]) (void) =
{
    LedTask,
    UsbTask,
};

/* Array of task period timeouts */
UINT16 TaskPeriodTimeoutMs[ACTIVE_TASK_NUMBER] =
{
    LED_TASK_PERIOD_MS,
    USB_TASK_PERIOD_MS,
};

```

Listato 3 - Inserimento task USB.

Listato 4

```

/*****
* Function:                UsbTask
* Input:                   None
* Output:                  None
* Author:                  F.Ficili
* Description:             Manage Usb tasks and state machines
* Date:                    28/02/15
*****/
void UsbTask (void)
{
    switch (SystemState)
    {
        /* System Initialization Phase */
        case InitializationState:

            /* Initialize port as digital */
            AD1PCFG = 0xFFFF;
            /* Initialize JOYSTICK pins */
            JOY_UP_TRIS    = LINE_DIRECTION_INPUT;
            JOY_RIGHT_TRIS = LINE_DIRECTION_INPUT;
            JOY_LEFT_TRIS  = LINE_DIRECTION_INPUT;
            JOY_DOWN_TRIS  = LINE_DIRECTION_INPUT;
            JOY_FIRE_TRIS  = LINE_DIRECTION_INPUT;

            /* Init Usb device */
            USBDeviceInit();
            break;

        /* System Normal operation Phase */
        case RunningState:
            /* Usb stack task */
            USBDeviceTasks();
            /* Emulate joystick */
            JoystickTask();
            /* Manage joystick */
            JoystickHidSendTask();
            break;

        /* Default */
        default:
            break;
    }
}

```

Listato 4 - Implementazione del task USB.

quindi poter trasferire informazioni verso il sistema operativo. La macro accetta i seguenti parametri:

- BYTE ep, che è l'endpoint verso il quale inviare i dati;
- BYTE* data; è un puntatore ai dati che si

Listato 5

```
void JoystickTask (void)
{
    /* Joystick state variable */
    static JoystickStateType JoystickState = DoNothing;

    if (JOY_UP == 1)
    {
        /* Jump to MoveUP */
        JoystickState = MoveUp;
    }
    else if (JOY_RIGHT == 1)
    {
        /* Jump to MoveRight */
        JoystickState = MoveRight;
    }
    else if (JOY_LEFT == 1)
    {
        /* Jump to MoveLeft */
        JoystickState = MoveLeft;
    }
    else if (JOY_DOWN == 1)
    {
        /* Jump to MoveDown */
        JoystickState = MoveDown;
    }
    else if (JOY_FIRE == 1)
    {
        /* Jump to MoveDown */
        JoystickState = FireButton;
    }
    else
    {
        JoystickState = DoNothing;
    }

    switch (JoystickState)
    {
        case MoveUp:
            /* Move the joystick up */
            HidBuffer[HID_REPORT_BUTTON] = 0;
            HidBuffer[HID_REPORT_X_AXIS] = 0;
            HidBuffer[HID_REPORT_Y_AXIS] = JOY_SPEED*MOVE_UP;
            break;

        case MoveDown:
            /* Move the joystick down */
            HidBuffer[HID_REPORT_BUTTON] = 0;
            HidBuffer[HID_REPORT_X_AXIS] = 0;
            HidBuffer[HID_REPORT_Y_AXIS] = JOY_SPEED*MOVE_DOWN;
            break;

        case MoveLeft:
            /* Move the joystick down */
            HidBuffer[HID_REPORT_BUTTON] = 0;
            HidBuffer[HID_REPORT_X_AXIS] = JOY_SPEED*MOVE_LEFT;
            HidBuffer[HID_REPORT_Y_AXIS] = 0;
            break;

        case MoveRight:
            /* Move the joystick down */
            HidBuffer[HID_REPORT_BUTTON] = 0;
            HidBuffer[HID_REPORT_X_AXIS] = JOY_SPEED*MOVE_RIGHT;
            HidBuffer[HID_REPORT_Y_AXIS] = 0;
            break;

        case FireButton:
            /* Move the joystick down */
            HidBuffer[HID_REPORT_BUTTON] = FIRE_ON;
            HidBuffer[HID_REPORT_X_AXIS] = 0;
            HidBuffer[HID_REPORT_Y_AXIS] = 0;
            break;

        case DoNothing:
            /* Move the joystick down */
            HidBuffer[HID_REPORT_BUTTON] = 0;
            HidBuffer[HID_REPORT_X_AXIS] = 0;
            HidBuffer[HID_REPORT_Y_AXIS] = 0;
            break;

        default:
            break;
    }
}
```

Listato 5 - Implementazione del task USB.

vogliono inviare;

- WORLD len; è la lunghezza dei dati da trasferire.

La macro prevede come ritorno un valore che permette di comprendere lo stato del trasferimento ("USB_HANDLE"). Ne vediamo un'applicazione qui di seguito.

ESEMPIO PRATICO: MINI JOYSTICK USB

Passiamo ora a utilizzare le informazioni viste fino a questo punto presentando un esempio pratico di un joystick USB. Il codice è stato realizzato e testato sulla DemoboardPIC32 pensata, progettata e realizzata per questo corso; le parti d'interesse nel nostro caso sono evidenziate con un rettangolo blu in Fig. 6, e sono il Joystick e il connettore USB. Ovviamente mediante il connettore USB ci collegheremo al computer, mentre mediante il joystick avremo la possibilità di muovere il mouse come un "trackpoint" con l'aggiunta del tasto di "conferma" (tasto sinistro del mouse).

La struttura del progetto rimane quella dello scheduler presentato nella puntata precedente ma con un'architettura mostrata in Fig. 7, dalla quale si evince la funzionalità dello scheduler in background, il blocco dei driver che comunica a basso livello con l'hardware e infine i due task che vengono richiamati per le funzionalità d'interesse (in questo caso il LED e l'USB).

L'organizzazione globale del progetto Joystick USB è riportata in Fig. 8, nella quale è possibile vedere sulla parte sinistra i file header e nella parte destra i file sorgente. È consigliabile avere una struttura di progetto ben organizzata per poter mantenere e gestire meglio la realizzazione/manutenzione dello stesso. All'interno della cartella "Bsp" (header file) sono stati inseriti i file di configurazione, come i configuration bits e il file spiegato precedentemente per la configurazione del nostro hardware, "hw_profile.h"; mentre nella



Fig. 6 - Parti d'interesse della DemoboardPIC32.

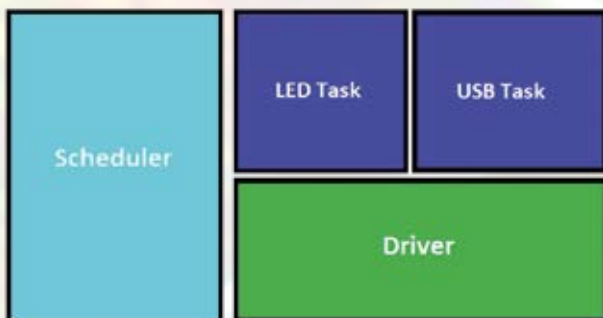


Fig. 7 - Architettura Software.

cartella equivalente del codice sorgente è stata inserita la gestione degli interrupt. Per quanto concerne la cartella "Stack", nella parte legata agli header sono stati inseriti i file per la configurazione dell'USB tra cui anche le funzioni HID (*usb_function_hid.h*); l'implementazione dell'USB è invece contenuta nella cartella "Stack" dei file sorgente, infatti al suo interno possiamo riconoscere i file precedentemente trattati. Le cartelle "Sched" e "Task" sono state ampiamente spiegate nel corso della puntata precedente.

Avendo compreso la struttura del progetto e come quest'ultimo funziona, passiamo ai dettagli implementativi partendo dalla configurazione delle porte d'interesse sulla DemoboardPIC32. Analizzando il data-sheet del PIC32MX e guardando le interconnessioni di esso con il joystick, possiamo inserire nel file header di configurazione (*hw_profile.h*) le definizioni visibili nel **Listato 2**.

Configurate le porte da cui poter leggere i segnali provenienti dal joystick, aggiungiamo un nuovo task legato all'USB come mostrato precedentemente nell'architettura, all'interno del file "*scheduler.c*" (oltre a quello del LED già presente) e visibile nel **Listato 3**.

Listato 6

```

/*****
* Function:    JoystickHidSendTask
* Input:      void
* Output:     void
* Author:     F.Ficili
* Description: Send HID data to Host
* Date:      07/03/15
*****/
void JoystickHidSendTask (void)
{
    /* Copy over the data to the HID buffer */
    HidReportIn[HID_REPORT_BUTTON] = HidBuffer[HID_REPORT_BUTTON];
    HidReportIn[HID_REPORT_X_AXIS] = HidBuffer[HID_REPORT_X_AXIS];
    HidReportIn[HID_REPORT_Y_AXIS] = HidBuffer[HID_REPORT_Y_AXIS];

    /* Send the 3 byte packet over USB to the host */
    LastTransmission = HIDTxPacket(HID_EP, (BYTE*)HidReportIn, 0x03);
}

```

Listato 6 - Funzione d'Invio dati mediante l'USB.

A questo punto creiamo un file dal nome "*usb_task.c*" che inizializza la periferica e successivamente la gestisce. Facendo riferimento al **Listato 4**, possiamo vedere come anche in questo caso è stato differenziato uno stato in cui viene inizializzata la periferica d'interesse (*case InitializationState*) e una fase di normale funzionamento (*case RunningState*). Nella fase d'inizializzazione sono state definite le porte come digitali (dato che per impostazione predefinita sono impostate come analogiche) e poi inizializzate come ingressi per poter consentire la lettura degli stati del joystick. Infine viene richiamata la funzione per inizializzare la periferica USB.

Sempre con riferimento al **Listato 4**, superato lo stato di inizializzazione, il codice che viene eseguito è quello di normale funzionamen-

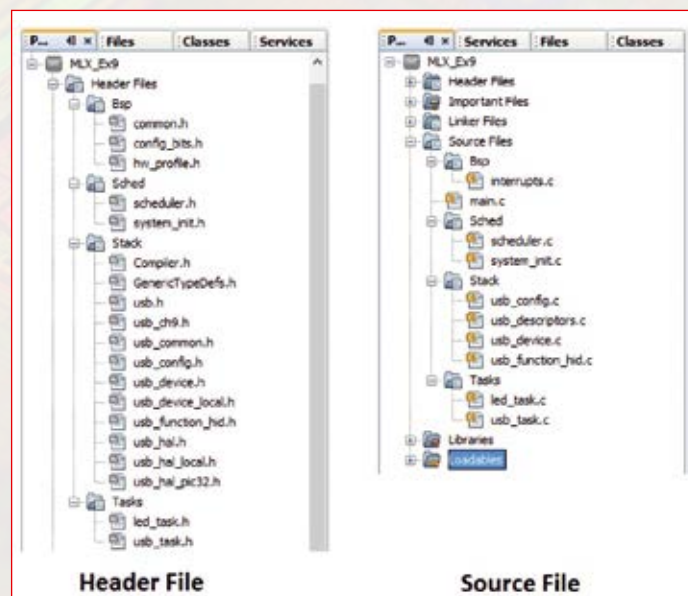


Fig. 8 - Struttura del programma Joystick USB.

Listato 7

```
//Product string descriptor
ROM struct{BYTE bLength;BYTE bDscType;WORD string[28];}sd002={
sizeof(sd002),USB_DESCRIPTOR_STRING,
{'J','o','y','s','t','i','c','k',' ','D','e','m','o',' ','\
'E','l','e','t','t','r','o','n','i','c','a',' ','I','n'}
};
```

to (case *RunningState*) e infatti si può notare come al suo interno siano presenti le seguenti funzionalità:

- leggere lo stato del joystick, che si ottiene con *"JoystickTask()"*;
- inviare il comando corrispondente al computer, mediante USB *"JoystickHidSendTask()"*.

Nel **Listato 5** viene mostrata l'implementazione della funzione *JoystickTask()* che come anticipato legge lo stato del Joystick e poi la memorizza nella variabile *"JoystickState"* che viene successivamente utilizzata per capire in che direzione far muovere in mouse tramite il blocco *"switch (JoystickState)"*.

Come si può vedere, oltre alla definizione dei quattro movimenti che il mouse può fare, sono stati inseriti anche altri due casi: il primo riguarda la pressione di un tasto (pulsante sinistro del mouse) e il secondo è legato a uno stato di attesa d'azione utente. Quest'ultimo caso è fondamentale perché altrimenti il nostro mouse rimarrebbe nell'ultimo stato memorizzato di movimento e continuerebbe a scorrere verso uno dei quattro bordi dello schermo. Infine l'ultimo passo da compiere è inviare al computer i dati d'interesse e quindi gli spostamenti del mouse. Per far ciò è stata definita la funzione *JoystickHidSendTask()*, riportata nel **Listato 6**. Come si può vedere, tale funzione non fa altro che copiare i dati del mouse all'interno del buffer HID e infine trasmetterli verso il computer mediante la macro analizzata precedentemente. A questo punto il programma è completo, non bisogna far altro che compilarlo, scaricarlo sulla DemoboardPIC32 e collegarla al computer per poter utilizzare il proprio *trackpoint* "fatto

in casa". La maggior parte dei sistemi operativi riconosce i dispositivi standard HID USB come tastiere/mouse, senza bisogno di driver speciali; infatti, una volta collegati, il computer li installa e successivamente appare sullo schermo un messaggio di conferma che il dispositivo conforme è stato riconosciuto/ installato correttamente e può essere utilizzato, come mostrato nella sequenza di messaggi d'installazione in **Fig. 9**. È doveroso precisare che quanto appena presentato come progetto pratico è compatibile con le versioni di Windows XP, Vista e 7, ma non con il più recente sistema operativo di Microsoft: Windows 8. Infatti in quest'ultimo caso la periferica non verrà riconosciuta e non sarà possibile utilizzarla in quanto è presente un'incompatibilità di driver. Come visibile nell'immagine di sinistra della **Fig. 9**, il descrittore può essere personalizzato a piacimento per poter far visualizzare, in fase d'installazione, il testo desiderato: nel caso specifico abbiamo impostato Joystick **Demo Elettronica In**. La personalizzazione del descrittore è fattibile modificando, all'interno della cartella "Stack" del progetto, il file *"usb_descriptors.c"*, come riportato nel **Listato 7**.

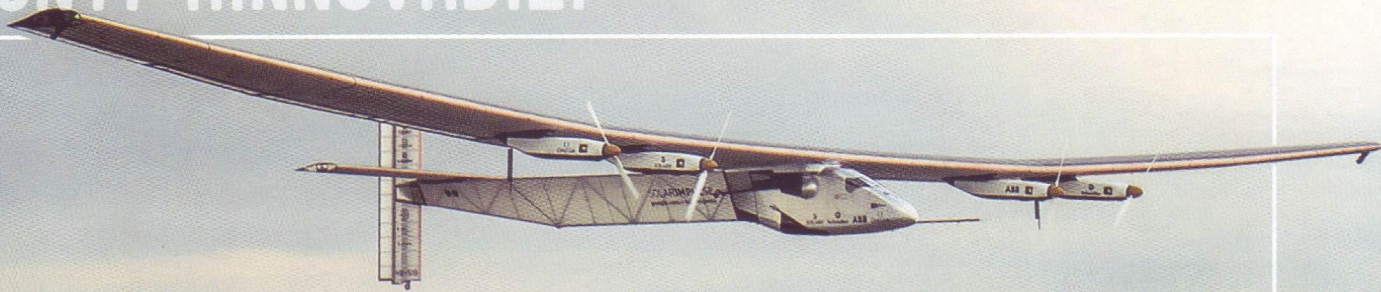
Fate attenzione al fatto che bisogna modificare anche la lunghezza della stringa da inviare, oltre al suo contenuto, altrimenti l'operazione non è possibile.

CONCLUSIONI

In questa puntata abbiamo introdotto uno dei più comuni e diffusi protocolli di comunicazione, ossia l'USB lato device; in più abbiamo analizzato lo stack della Microchip per poi applicarlo ad un esempio pratico legato alla DemoboardPIC32, realizzando un mouse mediante il joystick presente su tale scheda. Nella prossima puntata passeremo ad analizzare la parte USB Host, per completare la panoramica della trattazione riguardante l'USB. ■



Fig. 9 - Sequenza di messaggi d'installazione su Windows XP.



Solar Impulse 2 vola in Cina

Solar Impulse ha compiuto il primo volo in Cina coprendo una distanza di circa 1.400 chilometri in 16 ore; decollato da Chongqing, ha raggiunto Nanchino, sul delta del fiume Yangtze.

Il velivolo, alimentato da più di 17.000 pannelli fotovoltaici, pilotato dallo svizzero Bertrand Piccard, è alla sesta tappa delle dodici previste nell'affascinante viaggio, che ha fatto registrare alcuni passi falsi a causa delle avverse condizioni meteorologiche e in particolare a causa dei forti venti.

Piccard e il team di Solar Impulse hanno

trascorso le ultime settimane nella città cinese di Chongqing, dove hanno partecipato a conferenze ed eventi di sensibilizzazione sull'utilizzo delle energie rinnovabili.

Il velivolo, spinto da 4 motori da 10 hp cadauno muniti di elica a due pale, ha lasciato Abu Dhabi il 9 marzo, dopo aver sostato a Muscat (Oman), Ahmedabad e Varanasi (India), Mandalay (Birmania) e Chongqing (Cina).

Dopo la tappa di Nanchino, l'aereo solare attraverserà l'Oceano Pacifico, fermandosi alle Hawaii, per poi sosta-

re anche negli Stati Uniti e in Europa. Solar Impulse ha un'apertura alare che raggiunge i 72 metri, leggermente superiore a quella di un Airbus A380 ed è impegnato in un viaggio di 35 mila chilometri, che dovrebbe essere completato in cinque mesi.

La parte inferiore del velivolo contiene un insieme di batterie ai polimeri di litio che permettono di accumulare durante il giorno energia sufficiente a sostenere anche un volo notturno.

www.solarimpulse.com

L'energia dell'impianto di cogenerazione e l'energia creativa di Uppsala

Sarà una cupola colorata una delle prossime attrazioni di Uppsala, la quarta città della Svezia per numero di abitanti ma la più importante sotto il profilo culturale per essere la sede della prima Università svedese fondata nel 1477, per ospitare una delle più grandi cat-

edrali gotiche europee e per essere la città dove visse e insegnò Carlo Linneo, considerato il padre della moderna classificazione scientifica degli organismi viventi.

È stato forse questo background a guidare la mano degli architetti di BIG, lo studio danese che ha progettato il nuovo sito e che ospiterà, oltre alla centrale vera e propria, uno spazio al chiuso per eventi e manifestazioni, oltre ad un parco per attività outdoor durante la bella stagione.

Quando pensiamo ad una centrale elettrica o termi-

ca, immaginiamo un luogo industriale privo di fascino, poco adatto ad essere al centro di eventi culturali o semplicemente ludici. In questo caso la proposta di BIG trascende questa percezione e nei mesi estivi, durante il picco del turismo, la proposta "cristallina" si trasformerà in un luogo per eventi.

Il progetto di BIG fonde due archetipi industriali convenzionali in un ibrido: l'impianto industriale e la serra. Entrambi sono stati sviluppati per fornire una forma razionale ed efficiente di custodia, rispettivamente per la produzione e l'agricoltura.

Sfruttando le economie di scala associate con strutture a effetto serra, è possibile fornire una custodia trasparente al 100% per aggiungere una sagoma futuristica allo skyline di Uppsala con una leggerezza senza precedenti, consentendo ai cittadini di godere di scorci educativi di ciò che avviene all'interno, con la nuova cupola cristallina che funge da invito per l'esplorazione e l'istruzione: per la prossima generazione di energia creativa.

www.big.dk



le automobili è dunque la volta dei sistemi di accumulo domestici in grado di garantire (teoricamente) l'autonomia di una piccola abitazione. E se al momento questo risultato è ancora lontano, i sistemi di accumulo offerti da Tesla Energy sono sicuramente in grado di livellare i picchi di potenza assorbita, consentendo di prelevare energia dalla rete quando l'elettricità costa meno (load defection).

Powerwall utilizza batterie agli ioni di litio ed è disponibile in due modelli, da 7 kWh o 10 kWh. Entrambi possono essere facilmente collegati all'impianto solare esistente ed alla rete elettrica. Tipicamente i due sistemi di accumulo possono fornire energia sufficiente ad un'abitazione media durante l'arco delle 24 ore, evitando così di dover ricorrere alla rete.

Attualmente il prezzo di vendita è rispettivamente di 3.000 e 3.500 dollari. Queste le caratteristiche tecniche:

Montaggio: a parete Indoor / Outdoor

Energia: 7 kWh o 10 kWh

Potenza continua: 2 kW

Potenza di picco: 3,3 kW

Efficienza energetica: > 92%

Temperatura di esercizio: -20°C/+43°C

www.teslamotors.com

Fotovoltaico galleggiante per fornire energia a Malta

Malta cerca di abbandonare l'atavica dipendenza dal petrolio per la produzione di energia elettrica, puntando sul fotovoltaico galleggiante, anche per restare negli standard europei che prevedono di produrre almeno il 10% del proprio fabbisogno energetico da fonti rinnovabili entro il 2020.

Oltre all'energia eolica, al centro dell'attenzione di Malta è finita anche l'energia solare penalizzata sull'isola a causa della mancanza di terreni disponibili viste le limitate disponibilità per l'allocation degli impianti a terra. Per ovviare a tale problema, il Consiglio maltese per la Scienza e la Tecnologia sta considerando l'idea di realizzare

parchi solari galleggianti sul mare, finanziando con 200 mila euro lo studio su un progetto molto interessante.

Il progetto 'Solaqua', guidato dal professor Luciano Mulè Stagno, dell'Istituto per l'energia sostenibile dell'Università di Malta, vede la cooperazione di Pandia Energy e General Membrane Ltd. Il prossimo step è valutare la fattibilità tecnica e finanziaria dei pannelli solari galleggianti in mare aperto.

Il progetto 'Solaqua' è unico nel suo genere in quanto sarebbe il primo impianto galleggiante realizzato in mare e non solamente su bacini idrici.

www.generalmembrane.it/

Coms Ev la nuova minicar da città

Coms EV, è una minicar prodotta da Toyota Auto Body, divisione specializzata della casa automobilistica giapponese, che si propone come un mezzo elettrico ideale per la mobilità urbana.

Lunga 2,39 metri e larga 1,5 metri la minicar ha un bagagliaio funzionale, è pick-up con piano di carico e spondina posteriore abbassabile (Cargo Box) e vanta un raggio di sterzata di soli 3,2 metri.

La guida, posta al centro della monoposto, fa sembrare il veicolo come uno scooter di grossa cilindrata. La carrozzeria monoscocca è realizzata in materiali compositi, tanto da rendere il veicolo resistente ma leggero.

Coms EV ha brillantemente superato i crash test effettuati in Giappone. La trazione è posteriore e il motore elettrico da 5 W dispone di 250 Nm di coppia che devono spostare solo 400 kg.

Le batterie del veicolo sono nel pianale. Per la ricarica completa bastano 6 ore e si può utilizzare qualsiasi presa di corrente casalinga a 220 volt.

L'autonomia varia dai 60 ai 90 km e il costo di un pieno di energia è inferiore a un euro.

www.toyota-body.co.jp/



La Bachecca degli annunci

Vendo:

Da privato "soddisfatti o rimborsati", spedisco pagate alla consegna sconti per urgente realizzo:

- Radiocuffia stereo multifunzione multiaccessoriata Pioneer anche per uso investigativo.
 - Microspia americana FM 40,68MHz con auricolare ricevente e certificato di conformità (campionatura).
 - Campanello senza fili con funzioni manuali o automatiche (campionatura).
 - Proiettore laser a fascio visibile rosso + centralina multieffetti a 3 uscite con portafaretti + lampada di Wood (campionatura).
 - Accumulatore al gel a tracolla con multiuscite multitensioni ad uso contemporaneo accessorizzato tedesco.
 - Mini megafono compatto tipo torcia con altoparlante a membrana piatta (campionatura).
 - Variatore di velocità professionale fino a 1.320W + variatore universale fino a 1.800W.
 - Radiomicrofono con microricevitore a pile.
 - Microscopio tascabile tedesco da superficie 100 ingrandimenti con luce per microelettronica, micromeccanica, ecc... (campionatura).
 - Vibrografo a battute regolabili + micro incisore multialimentato accessorizzato.
 - Micro trapano con accessori multialimentato + saldatore ceramico rapido a temperatura regolabile.
- Possiedo inoltre utili dispense tecniche collezionabili dal 1900 in poi, che, invecchiandosi, si valorizzano. Contattare Pietro al numero di telefono 0371-30418.

Cerco:

Chassis valvolari da rottamare (no mobilia).
Prezzi da valutare. Spedizione in contro-assegno.
Spese postali a carico del destinatario.
Contattare Alberto al numero 065257175 oppure al 3382056777, ore pomeridiane (14-20).

Vendo:

- Cordless lunga portata 8/12Km;
 - 8000 metri cavo allarme 4*0.25 / 2*0.50+2*0.25 / 2*0.50+4*0.25 / 2*0.50+6*0.25;
 - Coppia ricetrasmittitori Alinco dj s800 nuovi;
 - Finale TV UHF 1W;
 - 100m cavo antenna RG213U (nuovo);
 - Rotore Jeager Supermount SMR 1224 reed HH per parabole fino a 3 metri;
 - Diversi rotori antenna ES CDE autorotor ed altri;
 - Centralino telefonico più citofono PBX15D 1 linea entrata 5 interni;
 - Centralino telefonico più citofono PBX2/8 2 linee entrata 8 interni;
 - Amplificatore Geloso valvolare G1/1070A° funzionante;
 - Alternatore 5KW trifase 4KW monofase mecc. alte.
- Contattare Giuseppe al numero di telefono 335-6257322 o inviare e-mail: giuseppe@spaziotempo.org

Offro:

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

Realizzo:

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

Eseguo:

- Riparazioni ed installazioni elettroniche, tecnico per assistenza.

Vendo:

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

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

Vendo:

Nuovo bromografo semi-professionale Delta Elettronica mai usato, quattro lampade, timer, area utile espositiva 40x27, corredato da basette presensibilizzate varie dimensioni. Il tutto a euro 80,00.
Contattare Gennaro al numero di telefono 335-398994.

Vendo:

- Ricetrasmittente sia per stazione fissa, che per mobile (auto, camper, barca, camion). Marca Electronica, Modello 360 da 120 Canali AM/SSB (lsb/am/usb) in 3 'shifts' - (A, B, C). Frequenza determinata con MC145106 PLL e 9 quarzi! Indicatore di modulazione a LED bicolore (verde come s-meter) e regolatore di guadagno sensibilità RF per DX o locali. In ottimo stato e perfettamente funzionante. Il tutto a 95 euro.
 - Amplificatore di linea AM/SSB ZG B150 da 100Watt per trasmettitore CB, che eroga 100 Watt in AM e 200 Watt in SSB. Adatto sia in casa che in auto o camion, roulotte, barca. Aumenterete la potenza di trasmissione in modo notevole! Ancora imballato, come nuovo e perfettamente funzionante. Il tutto a 65 euro.
 - Wattometro-Rosmetro ZG 500 Misuratore della Potenza in uscita e qualità dell'accoppiamento trasmettitore/antenna (r.o.s.) per stazione radiotrasmettitore CB. Ancora imballato, come nuovo, perfettamente funzionante. Il tutto a 40 euro.
 - Euro Ricetrasmittente CB Lafayette Texas Ricetrasmittente veicolare e/o fisso per banda CB, Marca Lafayette Modello Texas, omologato. 40 Canali AM/FM, Potenza in uscita 4Watt, frequenza PLL-2,5Watt audio, in ottimo stato e perfettamente funzionante.
- Contattare Roberto il numero di telefono 377-1398999 o inviare e-mail: serviceuk@libero.it.

Sviluppo:

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

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