

Elettronica In

www.elettronica.in.it

€ 7,50 - Bimestrale, anno XXX - n. 279 - GIUGNO / LUGLIO 2024

Centralina smart per irrigazione

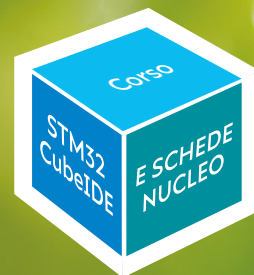
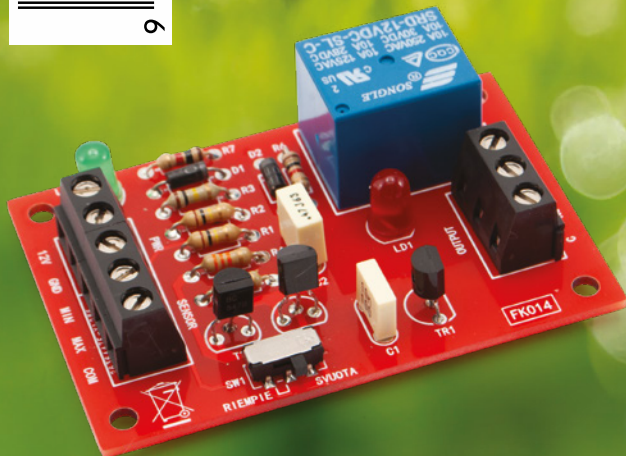


Piattaforma DSP audio



- ▶ Barra a LED psichedelica
- ▶ PONG Game 2D
- ▶ Amplificatore acustico
- ▶ Modulo ESP32 Low-Power
- ▶ Impianto audio completo
- ▶ Guida agli encoder
- ▶ Energy meter
- ▶ Breakout per driver motore

Stabilizzatore di livello liquidi



ARROW ELECTRONICS WORLD

20 giugno 2024

Arrow Electronics World sarà nuovamente lo scenario di incontro tra produttori, partner e terze parti nel mercato elettronico italiano.

Sarà un'occasione unica per condividere informazioni sulle opinioni di mercato e sulle tendenze, in un formato che combina business e svago.

Scansiona il
QR Code e
scarica il tuo
biglietto gratuito!



Palaverdi
Parma Convention Center
Viale delle Esposizioni 393/A
43126 Parma

221e[®] 3M ABLIC ABRACON
ARTESYN ALBANY ANALOG DEVICES
AEC BONA FIDE
CREE LED CUDVICES
20YS EAS EATON ELEMMASTER
embit ENTITY EXOR FAE TECHNOLOGY
gread ergad Honeywell infineon innoscience ITT cannon
ISSI JOHNSON KIOXIA kyocera
LANTRONIX LATTICE LIFE LIFE Littelfuse Lumberg
LUMIX MXIC MAS Melexis MICROCHIP
Micron MIPOT moxlex muRata MW.FEP
NES nexperia nichicon NVIDIA NXO
OMRON onsemi Panasonic PHOENIX CONTACT QUECTEL
IR RECOM reLoc REPLY samtec
SILICON LABS SYSTEM ST STALTE S
tastitalia TDK TE TECNO SYSTEM Telit
TOSHIBA TRACO POWER trueense ublox Variscite
VERI VICOR VISHAY WAGO Western Digital
Wolfspeed XU YAGEO

Servizi Arrow
ARROW ELECTRONIC COMPONENTS
ENGINEERING SOLUTION CENTER
ENTERPRISE COMPUTING SOLUTIONS
INTELLIGENT SOLUTIONS
SUPPLY CHAIN

CONVERGE[®] AN ARROW COMPANY infochips SILICONEXPERT
Media partner
EETimes POWER ELECTRONICS NEWS ELETTRONICA[®] Elettronica In
FARE ELETTRONICA elettronica tecniche nuove

Con la partecipazione di In collaborazione con Partner organizzativo
ANIF COMPONENTI ELETTRONICI ROME Maker Faire THE EUROPEAN EDITION
p.i. Camera di Commercio ennevolte



Il nuovo Idroelettrico: meno acqua, più energia

In un contesto globale che pone l'energia rinnovabile al centro delle strategie per uno sviluppo sostenibile, emerge la startup londinese RheEnergyse (www.rheenergyse.com). Questa azienda ha sviluppato una tecnologia idroelettrica che utilizza un fluido speciale, denominato R-19, che presenta una densità 2,5 volte superiore a quella dell'acqua. Questo sistema idroelettrico ad alta densità, grazie ai ridotti costi di installazione e al minimo

impatto ambientale, rappresenta una soluzione ottimale per l'accumulo di energia. Il funzionamento di questa tecnologia si basa su principi simili a quelli dei sistemi di pompaggio idroelettrico classici: l'energia in eccesso viene utilizzata per spostare il fluido da un serbatoio a quota inferiore a uno superiore, rilasciandolo

L'idroelettrico ad alta densità senza acqua offre un metodo innovativo per gestire l'intermittenza delle fonti rinnovabili.

successivamente quando necessario per generare energia. L'idroelettrico ad alta densità senza acqua offre un metodo innovativo per gestire l'intermittenza delle fonti rinnovabili e per aumentare la resilienza dei sistemi energetici. Mentre in Italia si discutono politiche restrittive sull'installazione del fotovoltaico in aree agricole, startup innovative come RheEnergyse continuano a progredire e a sviluppare nuove tecnologie, dimostrando che le barriere normative non possono arrestare il progresso tecnologico.

ON LINE

elettronica.in

SI RINGRAZIA

Arrow
Futura Elettronica
Microchip
Mouser



In copertina:
Una centralina di irrigazione a basso consumo energetico con controllo Bluetooth, che ci evita di irrigare il prato manualmente, ideale per godersi l'estate in totale relax.

Boris Landoni

boris.landoni@elettronica.in

Elettronica In

Anno XXX n. 279
GIUGNO / LUGLIO 2024

EDITORIALE

Direttore Responsabile

Boris Landoni

boris.landoni@elettronica.in

Redazione

Stefano Garavaglia, Paolo Gaspari,
Davide Scullino, Gabriele Daghetta

redazione@elettronica.in

Ufficio Pubblicità

Monica Premoli (0331-752668)

monica.premoli@elettronica.in

Ufficio Abbonamenti

Tel. 0331-752668

abbonati@elettronica.in

Direzione, Redazione, Pubblicità

FUTURA GROUP srl - Divisione Editoriale

via Adige 11 - 21013 Gallarate (VA)

Tel. 0331-752668

Abbonamenti

Annuo 6 numeri Euro 39,00

Annuo 6 numeri digitale Euro 29,00

Le richieste di abbonamento vanno inviate a:

FUTURA GROUP srl

via Adige 11, 21013 Gallarate (VA)

Tel. 0331-752668

oppure tramite il sito

<https://futuranet.it/abbonamenti>

Distribuzione per l'Italia

SODIP srl

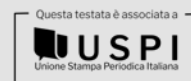
via Bettola 18 - 20092 Cinisello Balsamo (MI)

Tel. 02-660301 Fax 02-66030320

Stampa

ROTO3 Spa - Via Turbigo, 11/b

20022 CASTANO PRIMO (MI)



Rivista registrata presso il Tribunale di Milano
con il n. 245 il 03/05/1995.

Prezzo di copertina Euro 7,50.

Gli arretrati nei formati cartaceo e digitale (pdf) sono acquistabili
sul sito della rivista al prezzo di Euro 7,50.

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 02/07/2013.

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 (per maggiori informazioni
scrivere a redazione@elettronica.in). 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.

© 2024 Futura Group srl

**Futura
Group**
Edizioni



Contenuti

01

RUBRICHE

- 01 Editoriale
- 05 T&T Tips & Tricks
- 156 Scienza & Tecnologia

Progetto di Copertina



SOUND & LIGHT Barra a LED psichedelica

**11**

ARDUINO PONG Game 2D

**18**

AUDIO Amplificatore acustico

**27****11**

ARTICOLI

SOUND & LIGHT

11 Barra a LED psichedelica

Sposta un punto luminoso su una striscia di 10 LED a una velocità determinata dal ritmo della musica.

ARDUINO

18 PONG Game 2D

Rivisitiamo il classico Pong con un progetto DIY che combina Arduino Nano e LED NeoPixel, trasformando il leggendario gioco in una versione interattiva e moderna.

AUDIO

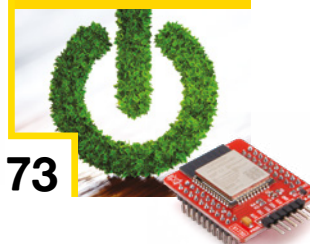
27 Amplificatore acustico

Amplifica voci, suoni e rumori captati nell'ambiente da una capsula microfonica e li rende udibili in un auricolare stereo; utilizzabile anche come apparecchio acustico.

DIDATTICA

33 Piattaforma per sviluppo DSP Audio

Realizziamo un modulo con ESP32 per lo sviluppo di applicazioni audio con effetti DSP personalizzabili in tempo reale.

DIDATTICA**Piattaforma
DSP Audio****33****AUTOMAZIONE****Stabilizzatore di
livello liquidi****47****AUDIO****Impianto audio
completo****55****ARDUINO****Modulo ESP32
Low-Power****73****AUTOMAZIONE****47 Stabilizzatore di livello liquidi**

Attraverso i contatti di un relé e di un doppio deviatore consente di mantenere il livello del liquido contenuto in una vasca o serbatoio al valore desiderato, intervenendo su una pompa o elettrovalvola.

AUDIO**55 AUDIO PRO: un impianto
audio completo**

Assembliamo il nostro impianto audio completo per regolarlo e verificarne il funzionamento. Terza e ultima puntata.

ARDUINO**73 Modulo ESP32 Low-Power**

Il meglio dei moduli con SoC Espressif senza il consumo che implicherebbe: scoprite come abbiamo fatto.

DOMOTICA**83 IdroBot la centralina di irrigazione
smart BLE e low-power**

Realizziamo una centralina di irrigazione low-power, alimentata a batteria e programmabile con smartphone tramite Bluetooth Low Energy.

DIDATTICA**93 Giochiamo
con gli Encoder**

Realizziamo una unità encoder indipendente che lavori in modalità indiretta e renda il nostro dispositivo una periferica seriale o IIC. Seconda e ultima puntata.

MISURE**111 Energy meter duale**

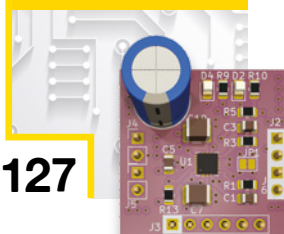
Analizziamo la Demo Board per l'Energy Meter progettata per facilitare connessioni alla rete elettrica e per fare analisi sui carichi grazie alla configurazione versatile e all'interfaccia USB per PC. Terza e ultima puntata.

I PROGETTI DI **127 Driver motore da 5A**

Realizziamo una scheda driver per motori a corrente continua (DC) da 5A.

DIDATTICA**135 STM32CubeIDE e schede Nucleo**

Iniziamo ad analizzare le principali periferiche integrate nei microcontrollori STM32. Esploreremo, utilizzando strumenti di debug, le funzionalità della periferica dedicata alla conversione da analogico a digitale e i Timer. Terza Puntata.

DIDATTICA**Giochiamo
con gli Encoder****93****MISURE****Energy meter
duale****111********Driver motore
da 5A****127****DIDATTICA****STM32CubeIDE
e schede Nucleo****135**



Alimentiamo l'elettificazione di ogni cosa

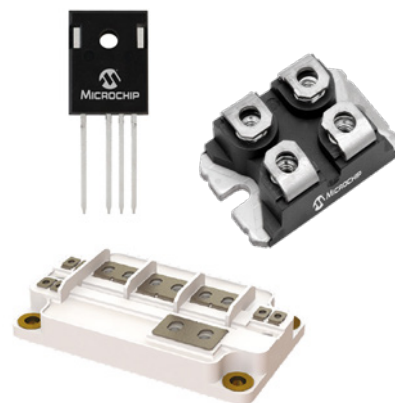
Sperimenta anche tu una rivoluzione nelle prestazioni ed efficienza dei sistemi di potenza industriale

I moderni sistemi industriali di potenza ad alta tensione richiedono prestazioni, efficienza e affidabilità più elevate per soddisfare le crescenti esigenze. I sistemi tradizionali al silicio e IGBT possono non risultare all'altezza, accelerando la necessità di una soluzione innovativa, in particolare una che offra robustezza e resilienza superiori in ambienti difficili.

Il Carburo di Silicio (SiC) funziona a temperature, tensioni e frequenze di commutazione più elevate, riducendo notevolmente le perdite di commutazione e contribuendo ad aumentare la densità di potenza e l'efficienza del sistema. Le soluzioni SiC riducono al minimo i requisiti di ridondanza e gestione termica dei dispositivi, contribuendo a ridurre le dimensioni, il peso e i costi del sistema rispetto alle soluzioni tradizionali.

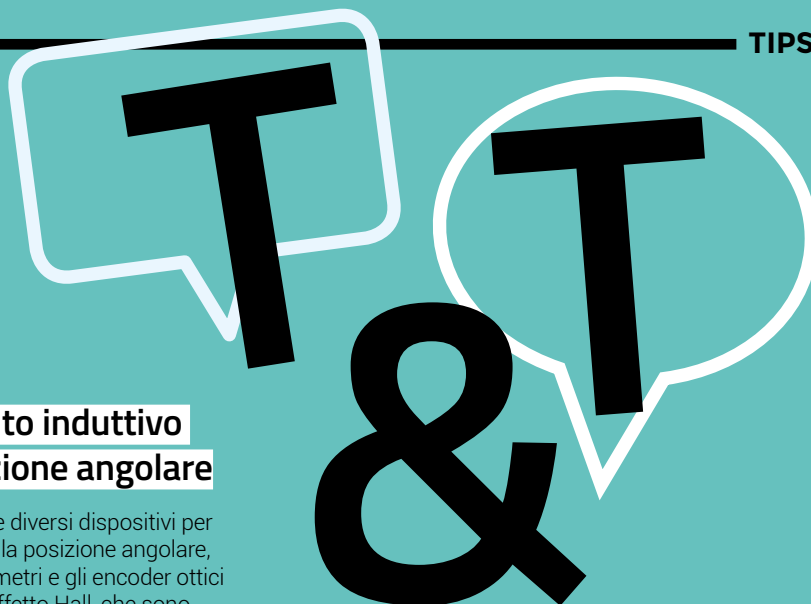
Frutto di oltre due decenni di esperienza nello sviluppo, nella progettazione e nel supporto di dispositivi SiC e soluzioni di alimentazione, il nostro ampio portfolio di soluzioni mSiC™ comprende prodotti con robustezza e prestazioni senza pari supportate dal più basso costo di sistema, dal più rapido dal time-to-market e dal rischio più basso.

Scopri come possiamo aiutarti ad adottare SiC con facilità, velocità e sicurezza.



microchip.com/SiC

Il nome e logo Microchip e il logo Microchip sono marchi industriali registrati di Microchip Technology Incorporated negli U.S.A. e in altri Stati. Tutti gli altri marchi appartengono ai rispettivi titolari. © 2023 Microchip Technology Inc. Tutti i diritti riservati.
MEC2517A-ITA-09-23



Rilevamento induttivo della posizione angolare

L'elettronica offre diversi dispositivi per il rilevamento della posizione angolare, come i potenziometri e gli encoder ottici e magnetici ad effetto Hall, che sono solo le soluzioni tradizionali. Negli ultimi anni, si è affermata una nuova tecnica basata sull'induzione elettromagnetica. Questo metodo è stato impiegato nello sviluppo del sensore di posizione angolare CIPOS, realizzato da Hella e viene utilizzato anche dalla AMS-Osram in circuiti integrati come l'AS5715R e da Onsemi, anch'essa con uno specifico integrato siglato NCV77320.

Il sensore Hella basato sulla tecnologia CIPOS (acronimo di Contactless Inductive Position Sensor) è costituito da uno statore che comprende una bobina di eccitazione, bobine riceventi e un'elettronica di valutazione, oltre che da un rotore formato da una spirale chiusa con una determinata geometria, applicata a un mozzo solidale con un perno.

Il rotore può essere un ulteriore PCB sul quale sono ricavate delle piste in rame che costituiscono la bobina ricevente avente la sagoma della forma descritta; questo secondo stampato è fissato al perno o mozzo che permette il collegamento meccanico all'elemento di cui il sensore deve rilevare la posizione angolare.

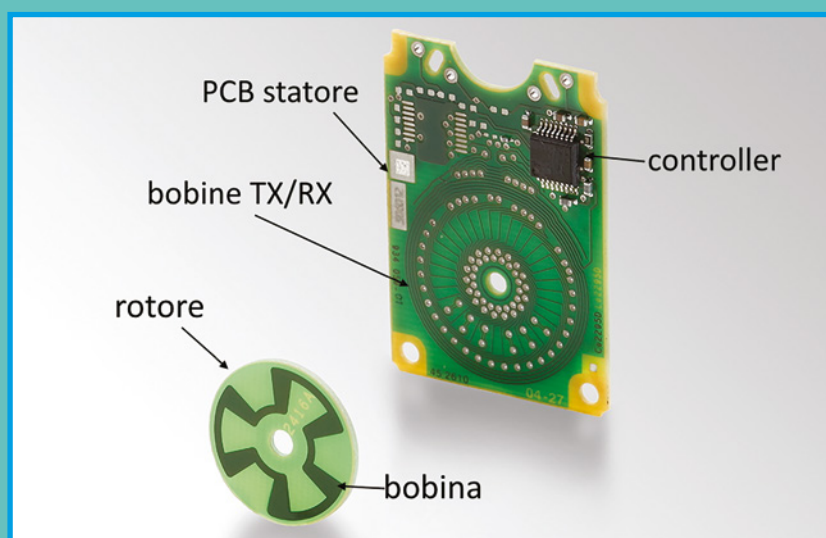
La struttura del sensore CIPOS è essenzialmente quella proposta nella figura qui accanto, la quale mostra i due circuiti stampati dello statore con elettronica e del rotore, privati dell'involucro.

Il controller della scheda applica una tensione alternata alle bobine trasmettenti che creano un campo magnetico che induce tensioni nel rotore, il quale è sagomato a spicchi con tre espansioni e altrettante tacche. Essendo questo di metallo, fa da bobina ricevente ed essendo in cortocircuito, la tensione indotta determina una corrente che viene cortocircuitata e a sua volta va a sviluppare un campo elettromagnetico che a sua volta interferisce con il PCB e nello specifico induce tensione nelle bobine riceventi (in

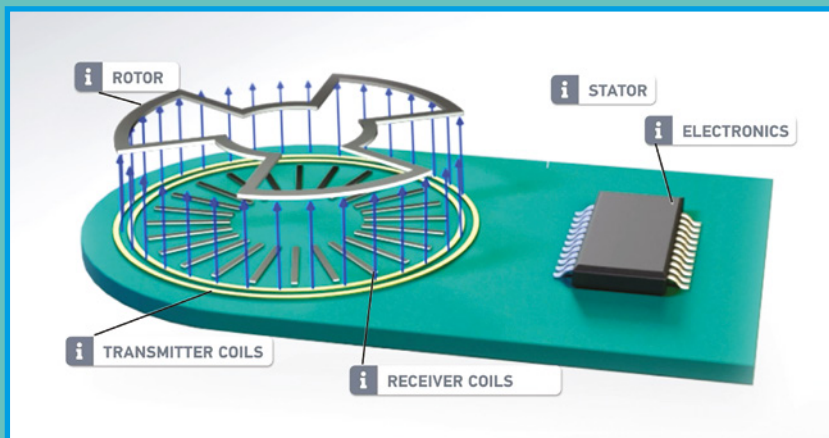
Tips & Tricks sono i nostri suggerimenti in pillole e i piccoli trucchi riguardanti sia l'hardware che il firmware, che sperimentatori, tecnici e progettisti elettronici vorrebbero conoscere per sviluppare meglio la propria passione: quella per l'Elettronica applicata. Ecco quelli che abbiamo preparato questo mese per voi.

realità interferisce anche con il circuito trasmittente). L'interazione tra i campi elettromagnetici provoca una variazione di corrente rilevata, influenzata dalla forma specifica a settori a spicchio del rotore. Nelle zone dove sono presenti i denti del rotore, l'intensità del flusso magnetico aumenta rispetto alle cave. Questo è dovuto sia alla minore distanza tra i denti e le bobine riceventi sullo statore, sia alla maggiore angolazione delle linee di forza del campo magnetico rispetto alla

superficie del PCB che ospita le bobine statoriche riceventi. Infatti l'induzione magnetica dovuta a una bobina alimentata a corrente variabile determina, ai capi del conduttore immerso nel campo, una tensione che varia in analogia con la corrente stessa. L'induzione determina una tensione indotta, il cui andamento nel tempo dipende dall'orientamento della spirale rotorica rispetto alle linee di forza del campo elettromagnetico; questo perché il flusso magnetico dipende



↑ Struttura del sensore di posizione angolare CIPOS a tecnica induttiva.



↑ Schematizzazione del funzionamento dei sensori di posizione angolare.

dall'intensità dell'induzione magnetica e dalla superficie entro il quale lo si considera.

Siccome il rotore è libero di ruotare, mantenendosi sullo stesso piano a una distanza fissa, a seconda della posizione angolare assunta rispetto alle bobine riceventi, nello statore vengono generati valori di tensione poi elaborati dall'ASIC, che valuta i segnali derivanti e genera un segnale di uscita analogico (tensione continua o PWM) o digitale (su Bus LIN). Le caratteristiche del segnale di uscita in tensione o di quello digitale dipendono dalla rotazione.

L'interazione tra i campi elettromagnetici coinvolti nel funzionamento del sensore di posizione CIPOS viene esemplificata dalla figura in alto in questa pagina. Tale tecnologia consente lunga durata e maggiore precisione rispetto al potenziometro, uniti all'insensibilità alla temperatura e alla stabilità della lettura della posizione angolare durante tutto il ciclo di vita del sensore. Queste caratteristiche rendono la tecnologia di rilevamento induttiva ideale per ambienti difficili come quello industriale e l'automotive, dove temperature e disturbi non sono trascurabili.

In campo automobilistico, la tecnologia induttiva trova applicazione sia nel rilevamento della posizione angolare di alberi di sensori e attuatori, valvole a farfalla, turbocompressore e sterzo, sia anche nel rilevamento di posizione e rilevamento dei giri dei motori elettrici di trazione delle auto elettriche ed ibride; il controllo del motore induttivo rileva la posizione elettrica all'interno di una coppia di poli di un albero rotante e può essere utilizzato per la commutazione di un motore in CC senza spazzole (BLDC). In questo ambito spiccano i prodotti del-

la AMS-Osram, tra cui gli integrati della famiglia AS5715A/AS5715R, sensori induttivi con uscita differenziale SIN/COS (segnale sinusoidale/cosinusoidale) capaci di rilevare la posizione angolare ad alta velocità.

Come i dispositivi CIPOS descritti, gli AS5715A/AS5715R necessitano di bobine trasmettenti e riceventi realizzate tipicamente (per rendere l'insieme più compatto) su circuito stampato. L'integrato alimenta la bobina trasmettente con un segnale elettrico ad alta frequenza e l'anello di cortocircuito posto sul rotore, sagomato con un certo numero di tacche, viene investito dalle linee di forza del campo magnetico generando a sua volta un campo elettromagnetico che ritorna alla bobina ricevente in misura dipendente dalla posizione dell'anello sul rotore. Da ciò dipende la tensione indotta nella bobina ricevente, poi rilevata, demodulata e amplificata dagli stadi interni.

Se il rotore è opportunamente sagomato, l'integrato fornisce alle uscite SIN e COS due segnali sinusoidali in opposizione di fase tra loro.

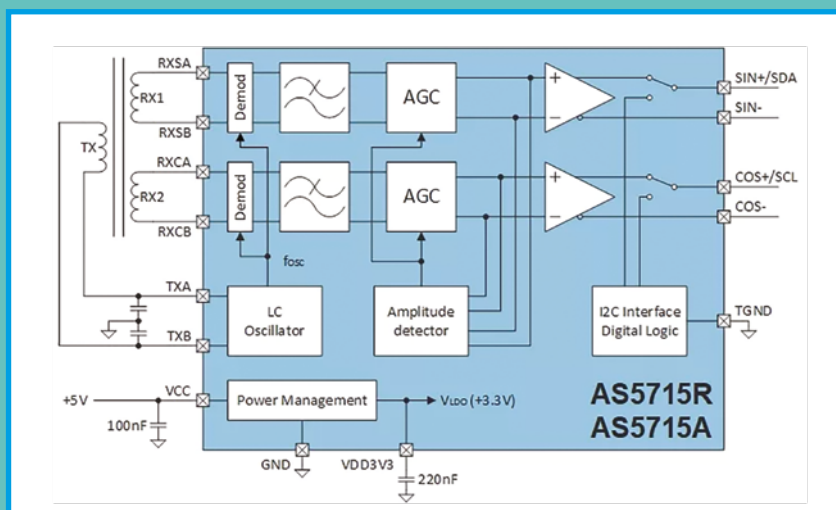
La ECU (Electronic Control Unit) può calcolare la posizione angolare del rotore applicando una funzione matematica $\text{atan2}()$ (arcotangente 2) ai segnali forniti dalle uscite SIN e COS.

Nell'applicazione ai motori elettrici, con la corretta disposizione delle bobine e del target, entrambi sincronizzati con il numero di coppie polari del motore, il segnale d'uscita del circuito integrato è proporzionale alla posizione angolare all'interno di un segmento di coppie polari con maggiore risoluzione e prestazioni elevate.

Gli integrati menzionati rappresentano un'alternativa a basso costo al risolutore, un tipo di encoder ampiamente utilizzato nei motori elettrici. Inoltre, possono essere impiegati per applicazioni sia in asse che fuori asse. L'accuratezza di misura, per esempio con un motore a 4 coppie di poli, è $<0,075$ gradi meccanici e $<0,3$ gradi elettrici. Il funzionamento è garantito fino a velocità angolari di 480.000 giri/min.

Tensione assoluta e raziometrica

Tra le caratteristiche del segnale di uscita dei sensori analogici viene sovente specificata l'eventuale dipendenza dell'ampiezza di detta tensione da quella di alimentazione. Un segnale in tensione la cui ampiezza dipende sia dal risultato della trasduzione (conversione in segnale elet-



↑ Schema a blocchi degli integrati AMS-Osram della famiglia AS5715.

trico della grandezza fisica che il sensore rileva) sia dalla tensione di alimentazione è detto "raziometrico" e la sua escursione viene definita come percentuale (per esempio dal 10% al 90%) del potenziale di alimentazione.

L'uscita in tensione assoluta, invece, fornisce una differenza di potenziale che dipende solo dalla relazione che lega la grandezza fisica da trasdurre a quella di uscita e la sua escursione non è affatto influenzata né dipendente dall'alimentazione, fatto salvo il caso, evidentemente, in cui la stessa non sia insufficiente a mantenere il corretto funzionamento del sensore.

Se un trasformatore non basta

Può capitare di aver bisogno di una tensione alternata con cui alimentare un alimentatore lineare e di non riuscire a trovare un trasformatore il cui secondario fornisca il valore esatto, oppure di avere in casa trasformatori la cui tensione è metà di quella che occorre.

In casi del genere è possibile utilizzare due trasformatori da rete collegando in parallelo i loro primari (così che entrambi ricevano il valore esatto, oppure di avere in casa trasformatori la cui tensione è metà di quella che occorre).

Per esempio se servono 21 volt si possono mettere in serie i secondari di due trasformatori, uno da 12V e l'altro da 9V; oppure se occorrono 24V ma in casa sono disponibili solo trasformatori da 12V, è sufficiente porre i secondari in serie tra loro.

Nel fare quanto esposto va però considerato un aspetto: per quanto funzionino in corrente alternata, i trasformatori presentano sul secondario una polarità istantanea che va rispettata, altrimenti invece che la somma delle tensioni di uscita si otterrà la differenza.

Per identificare questa "polarità dinamica" è sufficiente conoscere il capo di inizio e quello di fine del secondario di ciascuno dei trasformatori da porre in serie; se i capi non sono riconoscibili, è sufficiente procedere in maniera empirica unendo un terminale del secondario di un trasformatore con un capo dell'altro, misurando poi con un tester predisposto alla misura di tensioni alternate (con fondo scala pari almeno a 1,5 volte la somma teorica delle tensioni dei singoli secondari) il valore ottenuto fra i due capi rimasti liberi.

Ebbene, se il collegamento è corretto,

vale a dire in fase, le tensioni si sommano, mentre in caso contrario leggerete la differenza; nessun problema, perché se ciò accade è sufficiente invertire i collegamenti di uno dei secondari.

Ultima nota: collegare in serie i secondari di due trasformatori è qualcosa che va fatto tenendo presente la condizione che i due devono poter erogare la stessa corrente, altrimenti la corrente prelevabile sarà sempre pari a quella del secondario che può erogare di meno, pena un calo di tensione tanto più marcato quanto più elevata è la differenza tra la corrente di targa e quella che si va a richiedere alla serie.

Un po' THT, un po' SMD

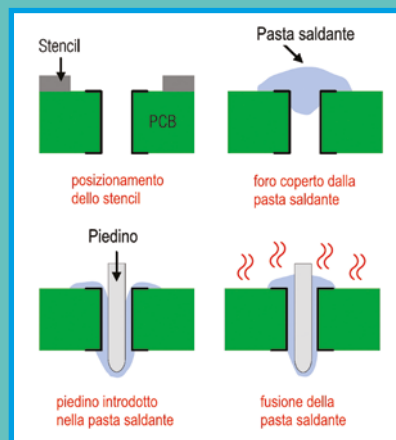
Oggigiorno la tecnica di realizzazione più diffusa delle schede elettroniche è quella SMT e a livello di produzione gli SMD vengono saldati con il reflow soldering, una metodica che consente di superare gli ostacoli posti da quei circuiti integrati che hanno i contatti a lato come i QFN o sotto al corpo come gli LGA e i BGA.

Tuttavia, quando una scheda elettronica prevede il montaggio sia di componenti SMD, sia di componentistica tradizionale da saldare a foro passante (THT, cioè Through Hole Technology) si pone il dilemma di dover svolgere due passaggi distinti (prima reflow soldering e poi saldatura a onda sul lato opposto, quando questo non presenti altri componenti SMD) oppure di saldare tutti gli elementi saldabili con il reflow soldering e poi stagnare a mano i componenti THT.

Quest'ultima opzione è praticabile se i componenti da montare a foro passante sono pochissimi.

L'ideale sarebbe poter saldare con la tecnica del reflow soldering anche i componenti THT, in modo automatizzato; questo in realtà è possibile, almeno per alcuni tipi di componenti che la tecnologia più recente offre in un formato innovativo noto come THR (Through Hole Reflow) ossia un Through Hole ma saldabile attraverso il reflow soldering e quindi in un unico passaggio insieme ai componenti SMD, perché è un ibrido fra montaggio superficiale e montaggio a foro passante.

I THR sono componenti elettronici per montaggio e saldatura a foro passante, quindi hanno i reofori da introdurre nelle piazzole come se dovessero essere stagnati manualmente, però sono costruiti per sopportare le temperature del forno di rifusione e per essere montati con la



↑ Schematizzazione della saldatura con la tecnica PIP per i componenti THR.

tecnica del reflow soldering, grazie anche al fatto che hanno i reofori sporgenti, seppure più corti.

Prima di eseguire il reflow soldering viene deposta una certa quantità di pasta saldatrice nei fori che ospiteranno i terminali, quindi i componenti vengono spinti fino a penetrare la pasta.

Quando si avvia la saldatura, la pasta si scioglie e la lega saldatrice una volta divenuta liquida per effetto della temperatura si ritrae nelle vie a causa delle forze di bagnatura e capillarità e forma il giunto di saldatura, che assicura la connessione elettrica e meccanica per ciascun reforo.

La tecnica di montaggio adottata per i THR è prevalentemente la PIP (Pin in Paste) della quale la figura in cima a questa colonna schematizza le fasi salienti: la pasta saldatrice viene deposta nelle aree scoperte di una dima metallica.

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.elettronicaain.it/webforum



ABBONAMENTO
ANNUALE



Abbonati a Elettronica In!
Realizza i **Progetti** descritti sulla rivista,
rimani aggiornato con i nostri **Corsi**,
approfondisci la conoscenza delle
tecniche e dei componenti
più avanzati con i **Tutorial**
e le nostre **News**.



Visita: www.elettronica.in.it

6 NUMERI
A SOLI

€29

ANZICHÉ € 45,00

ABBONAMENTO ANNUALE PER RIVISTA DIGITALE

In **OMAGGIO** avrai
il **10% DI SCONTO** per i tuoi ordini
su **Futurashop.it**

COME ABBONARSI

→ **on-line**

nella pagina "**Abbonati o rinnova**"
del nostro sito www.elettronicaain.it

→ **e-mail**

scrivendo a
abbonati@elettronicaain.it

→ **telefono**

telefonando a
+39-0331-752668

NUOVO FORMATO 160 PAGINE CON PIÙ CONTENUTI

ABBONAMENTO ANNUALE PER RIVISTA **CARTACEA**

In **OMAGGIO** avrai
il **10% DI SCONTO** per i tuoi ordini
su **Futurashop.it**

GRATIS 1 VOLUME A SCELTA della collana
"L'elettronica per tutti" oppure

BUONO SCONTO DA 5€ valido 12 mesi
da utilizzare su **Futurashop.it** nelle categorie "Libri" o "Riviste"

La possibilità di **LEGGERE GRATUITAMENTE**
la versione digitale di ELETTRONICA IN

6
NUMERI
A SOLI

€39

ANZICHÉ € 45,00

INDISPENSABILI IN LABORATORIO!



Tappeto in silicone per saldatura 550x350 mm

PER SALDARE

Tappeto in silicone antistatico multifunzione per saldatura resistente al calore fino a una temperatura di 500°C. Estremamente flessibile, anche dopo piegato ritorna sempre alla sua forma originale. Dispone di 3 piccoli contenitori e di 5 aree magnetiche.



cod. AS12

€ 35,00



Tappetino antistatico 550x300x2 mm

Tappetino antistatico particolarmente indicato per posizionare e manipolare componenti o schede elettroniche. Viene fornito completo di cavo di collegamento messa terra. Resistività superficiale: $10^7 \sim 10^{10}$ ohm/m², tempo decadimento carica statica: <0,03 secondi.

cod. AS13

€ 13,50

ANTISTATICO



Tappetino da taglio 300x220x2,8 mm

PER TAGLIARE

Tappetino da taglio realizzato in PVC (Polivinil-cloruro) ideale come piano di lavoro per tagliare carta, cartoncino, cartone pressato, fogli di vinile, ecc. Dispone di griglia di misurazione 1x1 cm e goniometro che consente all'utente di eseguire segni e tagli precisi.



cod. CUTMAT30X22

€ 9,00



Scopri la gamma completa su www.futurashop.it

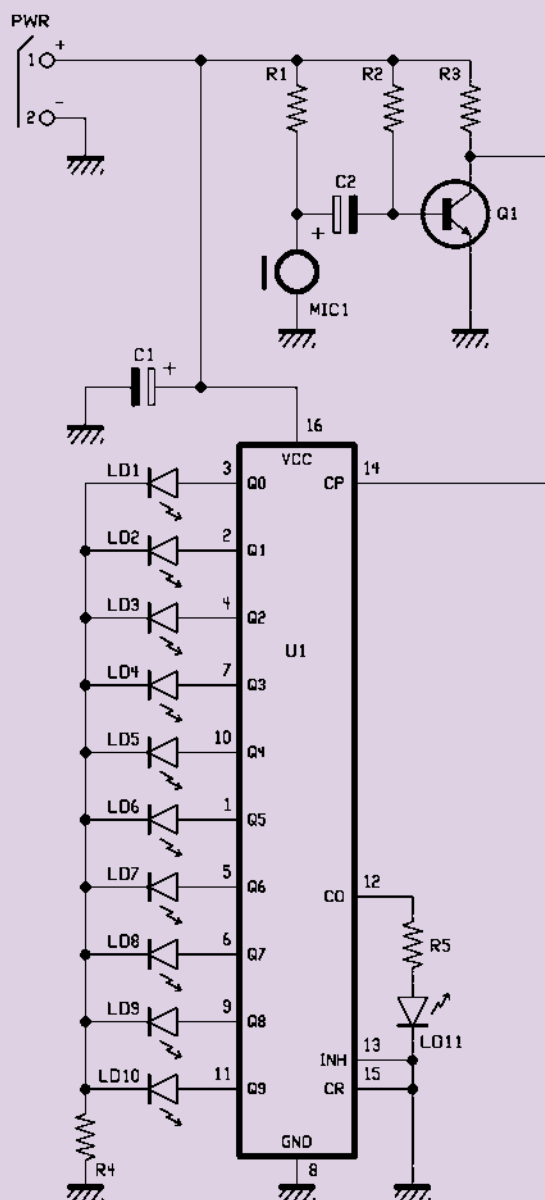
BARRA A LED PSICHEDELICA

di **DAVIDE SCULLINO**

Sposta un punto luminoso su una striscia di 10 LED a una velocità determinata dal ritmo della musica.

E

ffetti luminosi per animare esibizioni musicali, feste in casa o semplicemente abbellire degli apparati stereo portatili ne esistono di ogni genere: dalle luci psichedeliche a quelle stroboscopiche, dai level-meter più o meno graduati alle semplici lampade, fino ad arrivare ai LED che pulsano in base al ritmo della musica o delle sole percussioni. In questo articolo vogliamo proporvi qualcos'altro, pur sempre in tema di giochi di luce associati alla musica, che con un hardware essenziale e tradizionale (quindi niente microcontrollori, programmazione ecc.) consente di far scorrere su una striscia di 10 LED bianchi o multicolore, un punto luminoso, da sinistra a destra, la cui velocità di spostamento dipende dal ritmo della musica. Per come è fatto, il circuito ben si presta ad essere abbinato



a un riproduttore musicale, magari uno stereo portatile, per accompagnare l'ascolto della musica e scandirlo con l'accensione e lo spegnimento di un punto luminoso in movimento che avanza più o meno velocemente lungo una striscia. Il circuito è molto semplice e realizzarlo può anche essere l'occasione per esercitarsi nei montaggi elettronici da parte di chi inizia; infatti i componenti utilizzati sono tutti di facile reperibilità, perché molto comuni: si tratta di un contatore tipo CD4017, contornato da un transistor, una capsula microfonica e pochi elementi passivi (condensatori

e resistenze). Tutti rigorosamente a montaggio tradizionale, per la gioia di chi vede gli SMD fuori dalla propria portata.

Andiamo dunque a vedere bene di cosa si tratta, analizzando lo schema elettrico del nostro dispositivo.

SCHEMA ELETTRICO

Il nostro gadget, perché di questo sostanzialmente si tratta, è composto da un generatore di impulsi facente capo al transistor NPN Q1, la cui base è pilotata dal segnale fornito da una capsula microfonica electret-condenser; gli impulsi pilotano un contatore decimale CMOS di tipo CD4017 e un visualizzatore composto da dieci LED, che sul circuito stampato sono stati disposti in linea a comporre una striscia. A questi se ne aggiunge uno collegato all'uscita carry-out del contatore.

Il generatore di impulsi serve a dare il clock al contatore, il cui avanzamento determina l'accensione dei LED uno alla volta e in sequenza.

Tutto risulterà più chiaro analizzando lo schema elettrico passo dopo passo, a partire dal contatore CD4017, che è il cuore del circuito.

Vediamo dunque cosa accade in questo contatore decimale, premettendo che tale integrato conta gli impulsi in ingresso a decadi, attivando di volta in volta l'uscita corrispondente al numero di impulsi di clock contati in una decade, ossia quella corrente dall'ultimo reset.

Il 4017 dispone di 10 uscite, di peso crescente da 1 a 10, che nello schema elettrico sono nell'ordine Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9. Partendo dall'istante del reset (che può essere forzato ponendo a 1 logico il piedino 15 -CR- oppure automatico all'accensione) ogni volta che viene rilevato uno stato logico alto (ovvero un passaggio 0/1 logico) da parte dell'ingresso di clock (CP) lo stato del contatore incrementa di 1, nel senso che si porta a livello alto l'uscita di peso decimale immediatamente successivo a quello dell'uscita attualmente attiva.

Quindi, partendo dalla condizione di contatore resettato, inizialmente è a livello alto l'uscita 0; un primo impulso di clock porta ad 1 logico l'uscita 1, un secondo pone a livello alto la 2 e via di seguito, fin quando non si conta il decimo impulso, che attiva la 0, oltre il quale il conteggio riprende da 1. Le uscite assumono lo stato alto una sola per volta. Il contatore dispone di un'ulteriore uscita, localizzata al piedino 12, chiamata carry-out (CO), la quale fornisce un impulso a livello logico alto quando il CD4017 riceve sul clock un'ulteriore impulso quando ad essere attiva è l'uscita decimale di peso 10

(Q0); tale uscita serve per triggerare un contatore in cascata, nei circuiti in cui si debbano contare impulsi secondo potenze di 10, ma nel nostro caso la impieghiamo per accendere un ulteriore LED quando viene raggiunto il decimo impulso all'ingresso CP. In condizioni normali, il piedino 12 si porta da zero ad 1 logico quando il contatore conta un impulso oltre l'undicesimo, quindi si attiva l'uscita Q0.

Potete comprendere meglio il funzionamento del contatore CD4017 guardando la **Fig. 1**, che ne propone sia lo schema logico interno, sia il diagramma di temporizzazione.

Il nostro contatore decimale dispone altresì della funzione di abilitazione del clock, utilizzabile ad esempio quando si deve decidere l'istante in cui far partire il conteggio; nel nostro caso il clock è sempre attivo e ciò perché abbiamo posto a massa il piedino 13 (INH).

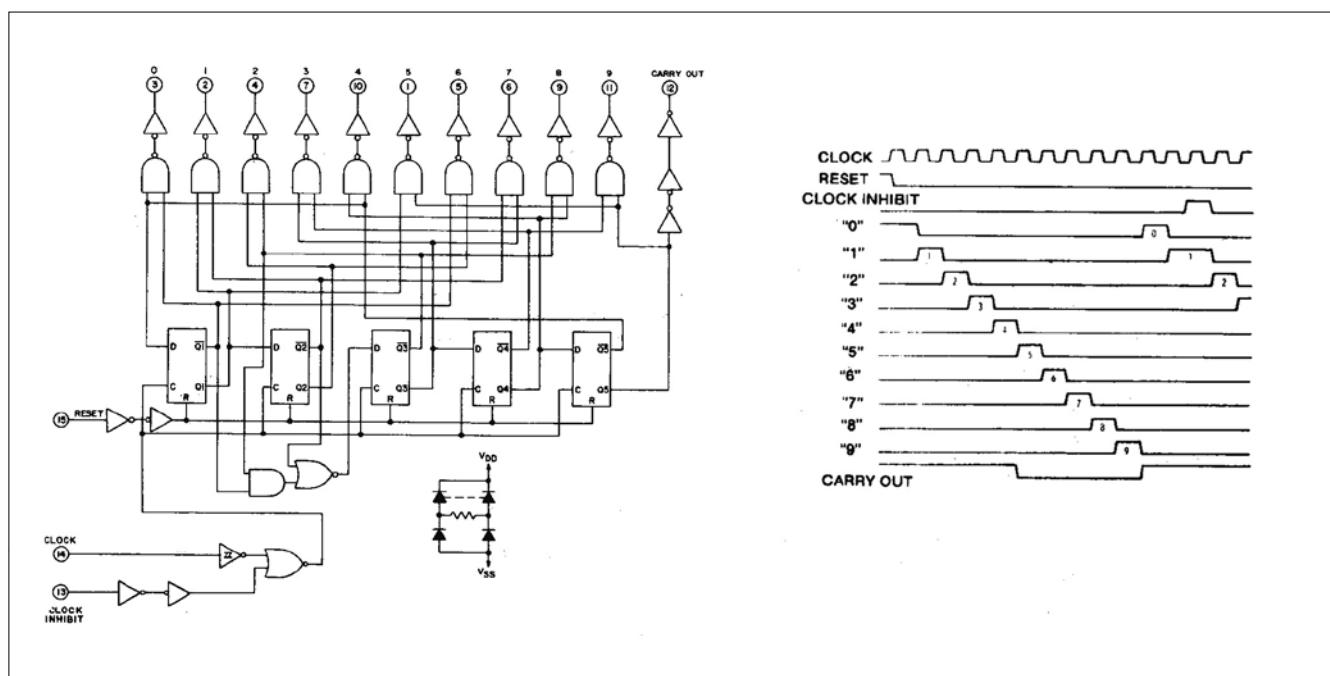
Per come abbiamo collegato il contatore, ossia con il piedino di reset (15) a massa (il che equivale a disattivare il reset esterno) il nostro CD4017 conta fino a dieci e viene azzerato automaticamente all'undicesimo impulso, per riprendere da capo il conteggio. Ad ogni impulso si porta a livello l'uscita di peso decimale superiore a quella attiva fino all'istante precedente, la quale torna a zero logico. Siccome ciascuna delle uscite Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8 e Q9 pilota un LED si accenderanno in sequenza ed uno solo alla volta, i 10 LED, dando la sensazione che un punto luminoso scorra su

una barra da un lato all'altro, per poi riprendere una volta arrivato a fine percorso.

I diodi luminosi che il 4017 pilota hanno tutti il catodo in comune, terminante a massa con un resistore da 330 ohm, che nello schema elettrico è siglato R4; questo perché, dovendosi illuminare uno solo alla volta, non serve una resistenza per ciascuno ma è possibile disporre una sola, calcolata per la corrente diretta di un singolo LED.

Bene, a questo punto facciamo un passo indietro e torniamo al blocco del circuito che genera gli impulsi per il contatore: la capsula microfonica electret-condenser capta i suoni nell'ambiente e trasforma la vibrazione della propria membrana impressa dalla pressione acustica in un segnale elettrico in tensione, che internamente viene amplificato da un jFET connesso a source comune, il cui drain fa capo al terminale + (il - va a massa). La capsula electret viene polarizzata attraverso un resistore collegato al positivo di alimentazione del circuito, che nello specifico è R1 e funge anche da resistenza di drain del jFET interno; tra il positivo e la massa della capsula MIC1 è quindi disponibile un segnale dell'ampiezza di qualche decina di millivolt efficaci, che tramite il condensatore elettrolitico C2 (utilizzato per disaccoppiare la componente continua e separare i circuiti di polarizzazione della capsula e del Q1) giunge alla base del transistor NPN 9014, qui utilizzato come amplificatore di segnale in configurazione a emettitore comune

Fig. 1
Schema a blocchi e diagramma di funzionamento del decade-counter 4017.

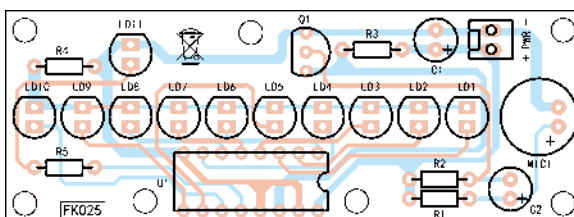
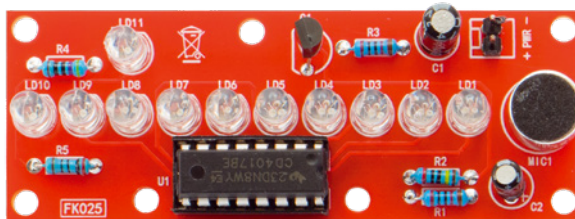


Elenco Componenti:

R1, R3: 20 kohm
R2: 2 Mohm
R4: 470 ohm
R5: 200 ohm
Q1: 9014
C1: 100 μ F 16 VL elettrolitico
C2: 1 μ F 16 VL elettrolitico
U1: CD4017
MIC1: Microfono 10 mm
LD1 ÷ LD11: LED 5 mm cambiacolore

Varie:

- Zoccolo 8+8
- Pin strip maschio 2 vie
- Circuito Stampato FK025 (77x28 mm)



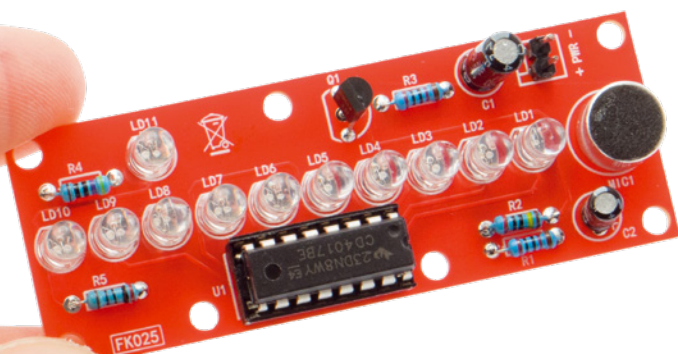
e polarizzazione in base operata attraverso il resistore R2. Q1 amplifica la tensione in arrivo dal microfono di circa 200 volte, ottenendo dai picchi del segnale audio impulsi idealmente sinusoidali, ma unidirezionali, la cui ampiezza può raggiungere i 5 ed anche 10V, urtando contro l'alimentazione; tali impulsi sono sufficienti a triggerare l'ingresso di clock (CP) del contatore, sebbene la loro forma d'onda non sia rettangolare: infatti i fronti di salita non rettilinei né verticali caratteristici degli impulsi ricavati dai segnali audio vengono squadrati da una porta logica NOT a trigger di Schmitt, che il 4017 ha all'ingresso di clock e che permette di ottenere impulsi perfettamente rettangolari con cui eccitare

Il circuito è semplice e utilizza soltanto un contatore.

la logica che segue, vale a dire la cascata di flip-flop componenti il contatore decimale. Ogni volta che il segnale audio amplificato supera l'ampiezza richiesta dall'ingresso di clock del 4017, si ha una commutazione e l'attivazione dell'uscita successiva a quella che l'istante prima si trovava a livello alto. L'effetto visivo che si ottiene, quindi, è una rapida successione dell'accensione dei LED disposti in cerchio fin quando si tiene premuto il pulsante S1, quindi il punto luminoso comincia a vedersi scorrere sempre più lentamente fino a fermarsi, come accadrebbe alla pallina della roulette. Concludiamo la descrizione con l'alimentazione del circuito, che si applica tra i contatti PWR (Vcc è il positivo e GND il negativo) ai quali va applicata una tensione continua di valore compreso fra 3 e 9 volt, prelevata da una fonte in grado di erogare almeno 60 mA: va benissimo una pila a secco di tipo "transistor" ossia da 9 volt, ma anche un portatile contenente 4 stilo in serie.

I diodi luminosi utilizzati nel circuito possono essere bianchi, ma nel prototipo sono in realtà dei LED cambiacolore RGB da 5 mm e a differenza dei comuni LED bianchi, questi possono cambiare colore durante il periodo in cui rimangono accesi, dando al gioco di luci un effetto ancor più suggestivo. Il LED cambiacolore si presenta come un comune LED a due terminali: il più lungo è il positivo (equivarrebbe all'anodo) ed il più corto è il negativo (catodo) che sta dalla parte smussata del contenitore.

Le lunghezze d'onda emesse sono: rosso 620 nm, verde 520 nm, blu 460 nm, mentre per i tre colori l'intensità luminosa (a una corrente diretta di 20



mA) sono per il rosso 750 mcd, per il verde 1,5 cd e per il blu 750 mcd.

Quando viene alimentato, il LED cambia il colore della luce emessa in modo automatico, partendo dalla tonalità luce rossa, per virare poi verso il lilla, il viola, il blu, il verde, il giallo e poi di nuovo il rosso, ricominciando il ciclo. Ciò viene ottenuto grazie a un minuscolo chip inserito nello stesso package del componente, a ridosso della giunzione PN, che pilota il LED vero e proprio.

Chiaramente la possibilità di vedere tutte le variazioni cromatiche è legata al tempo per cui ciascun diodo luminoso rimane acceso, ovvero al ritmo della musica: più è veloce, meno è probabile che per ciascun LED siano visibili tutti i mutamenti di colore.

REALIZZAZIONE PRATICA

Bene, ora che abbiamo descritto lo schema elettrico possiamo passare alle note costruttive, partendo dal circuito stampato che abbiamo disegnato allo scopo e del quale rendiamo disponibile, nelle pagine di presentazione di questo numero della rivista (su www.elettronica.in.it) le tracce lato rame per ottenere la pellicola necessaria a procedere per fotoincisione; ricordiamo che questo PCB è a doppia ramatura e va quindi impressionato in due passaggi: prima un lato e poi l'altro, centrando la relativa pellicola mediante due fori praticati in altrettante piazzole alle estremità.

Se non avete l'attrezzatura per preparare il circuito stampato, non preoccupatevi, perché con pochi soldi e in breve tempo potrete ottenerlo tramite il servizio realizzazione PCB della Futura Elettronica, accessibile on-line alla pagina web <https://ftrlink.it/pcb>.

Una volta inciso e forato il circuito stampato dovete iniziare il montaggio dei componenti richiesti, i quali sono tutti a foro passante e quindi facilmente gestibili anche da chi ha poca esperienza ed un'attrezzatura minima limitata al classico saldatore, al filo di lega saldante e al tronchesino per elettronica.

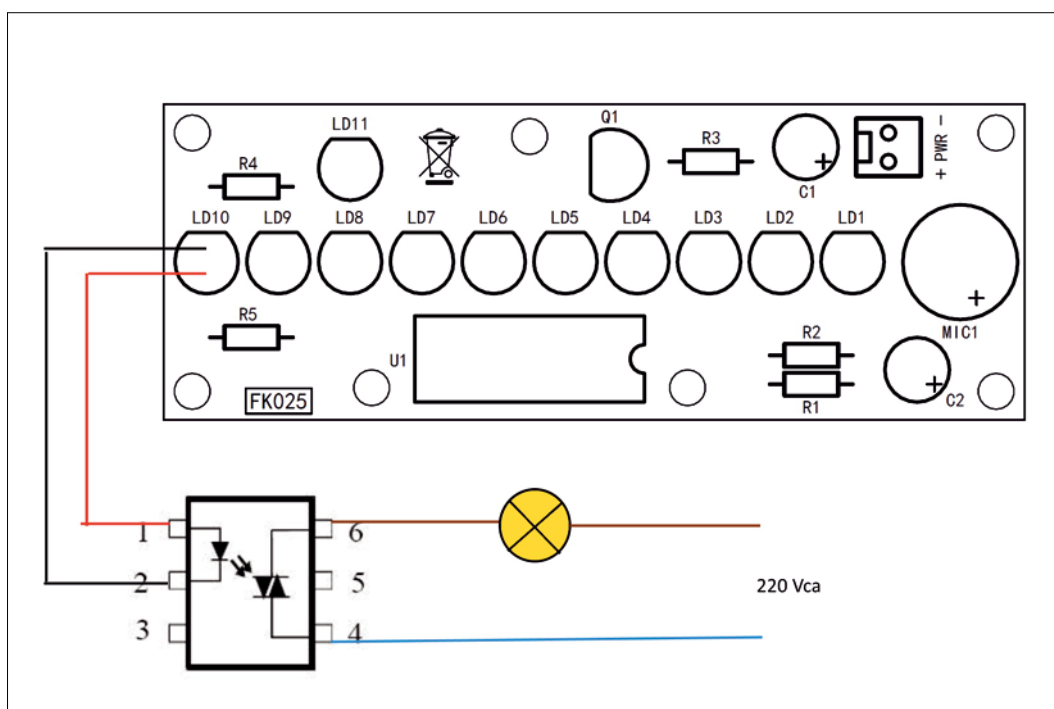
Entrati in possesso di tutto il necessario, iniziate il montaggio con le resistenze e il pulsante miniatura, poi con condensatori, prestando attenzione alla polarità dei due elettrolitici, quindi procedete con lo zoccolo dip da 8+8 piedini per il CD4017, cercando di allineare la tacca di riferimento come indicato nel piano di montaggio. In alternativa è possibile saldare direttamente il contatore, però bisogna fare attenzione a non surriscaldarli; invece se optate per il montaggio su zoccolo, inserirete l'integrato dopo che avrete saldato tutti gli altri componenti. Passate quindi al transistor, da orientare con la parte arrotondata rivolta alla resistenza R3, poi agli 11 LED, ricordando che per il loro corretto orientamento, il catodo è l'elettrodo che si trova dal lato smussato del contenitore; completate il montaggio saldando nelle piazzole per l'alimentazione (PWR) due fili di colore differente (idealmente rosso per il positivo e nero per il negativo) o il cavetto bipolare di una presa volante "a strappo" di quelle per le pile da 9V.

Per l'orientamento dei componenti polarizzati (quindi LED, elettrolitici e integrati) fate riferimento al piano di montaggio proposto in queste pagine. Una volta completato il montaggio e verificate l'esattezza, il vostro circuito è pronto per l'uso, dato che non richiede alcuna operazione di taratura preliminare; basta alimentarlo e vedrete subito accendersi casualmente un LED, perché il



Il prototipo montato con a bordo la capsula electret che rileva il suono.

Fig. 2
Cablaggio del
circuito in caso di
pilotaggio di
lampade a 220Vca.



contatore 4017 non viene resettato all'accensione e pertanto presenta un'uscita attiva a caso. Per far funzionare il circuito fornitegli un'alimentazione continua a tensione di valore compreso fra 9 e 12 volt; la corrente richiesta è dell'ordine dei 50 milliampere, tuttavia conviene che l'alimentatore utilizzato allo scopo sia in grado di erogare qualcosa in più, per poter far fronte ai picchi di assorbimento che inevitabilmente si verificheranno. In luogo dell'alimentatore potrete utilizzare una pila da 9V o un gruppo di 6 stilo in apposito portatile.

CONCLUSIONI

Il circuito che vi abbiamo proposto in queste pagine è un gadget luminoso, quindi un effetto visivo collocabile ovunque: non solo all'interno di uno stereo

portatile ma anche di un giocattolo o anche di una decorazione per vetrine o banchi di supermercato: in questo caso mostrerà i propri giochi di luce in base alle voci e ai rumori prodotti da chi si avvicina al microfono. Naturalmente può essere anche pensato come controller di qualcosa di più grande e appariscente, come ad esempio delle luci per spettacoli musicali, feste e via di seguito; in questo caso bisognerà collegare le uscite del contatore 4017 ciascuna all'ingresso di un driver per LED di potenza, che a sua volta piloterà un LED di vario colore, il quale funzionerà da faretto.

Si può anche restare sul classico e pilotare faretto funzionanti a 220 Vca, ma in questo caso sarà opportuno interporre dei fotoaccoppiatori di tipo fotoTRIAC come i MOC3040, il cui ingresso andrà pilotato con un'uscita ciascuno del contatore 4017 o, meglio ancora, in parallelo a ciascun LED del circuito, lasciando la resistenza R4 di valore tale e quale quello riportato nell'elenco componenti.

Lo schema di collegamento è quello esemplificato nella **Fig. 2**, che propone la connessione ad un LED (la stessa si può applicare a tutti e 11); dall'immagine vedete che i piedini 1 e 2 vanno connessi al posto del LED oppure in parallelo ai suoi terminali, rispettivamente di anodo e catodo.

L'uscita del fotoTRIAC andrà a pilotare il faretto o lampadina a 220 Vca, per una potenza massima di 40 W per canale.



COSA OCCORRE ?

Codice	Descrizione	Prezzo € IVA inclusa
FK025	Luci Psichedeliche a LED con controllo sonoro	8,50
ER09W09V	Alimentatore switching 9V / 1A	7,00

IL MATERIALE VA RICHIESTO A:

Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 - www.futurashop.it

La chiave per l'accesso illimitato

Esplorate milioni di componenti per il vostro prossimo progetto



Secret Archives
Vatican City

Potreste non accedere mai agli archivi segreti Vaticani, ma con noi avete accesso illimitato a milioni di componenti elettronici. Da Mouser siete sempre liberi di controllare i componenti che desiderate.

mouser.it



**MOUSER
ELECTRONICS**

PONG GAME 2D



Rivisitiamo il classico Pong con un progetto DIY che combina Arduino Nano e LED NeoPixel, trasformando il leggendario gioco in una versione interattiva e moderna.





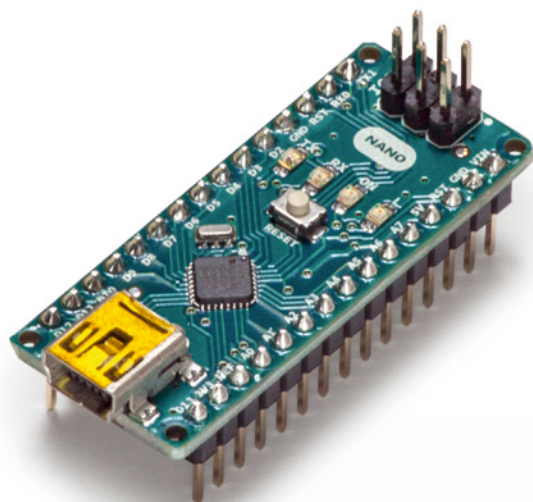
di **ROBERTO TESTI**

Chi tra voi ha qualche annetto sulle spalle sicuramente ricorda i primi videogame degli anni '80 del secolo scorso, arrivati in Italia da America e Giappone, disponibili non solo nei bar e nelle sale

giochi ma anche sotto forma di console per l'utilizzo domestico; notissime erano le Atari, Intellivision e ColecoVision, i cui primi giochi, molto semplici, includevano il ping-pong, noto come Pong. In questo semplice gioco dalla grafica spartana, due racchette rappresentate da righe verticali poste ai lati dello schermo, potevano scorrere su e giù per battere o ribattere un punto luminoso a rappresentare la pallina. Naturalmente il tutto era rigorosamente in bianco e nero. Tutte si avvalevano di un televisore come schermo, nelle prime versioni era collegato all'antenna TV tramite un modulatore, su un canale da sintonizzare dopo l'installazione. Pochi anni ed è cambiato tutto, fino ad arrivare alle Nintendo, a colori e dotate di computer potenti, ben di più di quelle primordiali che avevano dentro un Z80 o un Motorola 68010.

Senza voler andare ai livelli delle odierne console, dotate anche di game pad con accelerometri in grado di percepire i movimenti e trasmetterli, abbiamo pensato di attingere alle risorse dell'immane Arduino per implementare una versione semplificata del Pong, dove il visualizzatore è una classica barra di LED NeoPixel. Ai lati verranno mostrati alcuni punti luminosi corrispondenti alla distanza alla quale può trovarsi il giocatore che ribatte la palla (ovvero quanto il giocatore attende prima di ribattere) mentre al centro un certo numero di punti luminosi del colore dei giocatori segnerà il punteggio raggiunto nel gioco in corso. Per giocare, ogni giocatore ha a disposizione un pulsante che deve essere premuto quando la palla, rappresentata da un punto luminoso scorrevole, si trova tra il LED più interno e quello più esterno. Se un giocatore preme il pulsante mentre la palla è fuori da questa zona, perde il turno e l'avversario guadagna un punto. La palla accelera allontanandosi dal centro, aumentando la difficoltà del gioco. Tuttavia, se la palla viene colpita più vicino al centro, la sua velocità diminuisce. Accelerare il moto della pallina permette al giocatore

Fig. 1
La scheda Arduino Nano è il cervello del gioco.



che ribatte di avere più possibilità di guadagnare il punto perché rende più difficile la respinta da parte dell'avversario, tuttavia comporta dei rischi, perché più si attende che la pallina arrivi all'estremità, più si rischia di perdere il game.

Siccome è possibile il solo movimento simulato in una direzione, abbiamo chiamato questo gioco Pong 2D, per distinguerlo da quello tradizionale a video dove il giocatore può spostarsi sia verticalmente che orizzontalmente.

L'HARDWARE DEL PROGETTO

Per implementare il gioco elettronico del Pong qui proposto abbiamo fatto uso di pochissimi componenti e dispositivi elettronici che sono sostanzialmente una scheda Arduino Nano o compatibile (**Fig. 1**), due pulsanti "game" normalmente aperti (robusti e precisi come richiesto dal nostro gioco) con base da tavolo che abbiamo stampato in PLA colorato mediante la nostra stampante 3D, uno spezzone di strip a LED RGB Neopixel e un alimentatore da 5V, 1A per far funzionare l'insieme. Il tutto è collegato secondo lo schema di cablaggio che vedete in queste pagine. Nella scheda Arduino viene eseguito uno sketch che legge lo stato dei pulsanti e l'alternarsi delle commutazioni tra i due relativi concorrenti, attraverso due pin digitali configurati come input: 5 per il pulsante del giocatore 2 (blu) e 8 per il tasto del giocatore 1 (rosso). Tramite la linea di I/O D3 Arduino pilota, inviando in forma seriale le stringhe in formato NeoPixel, la strip a LED, la quale riceve i dati mediante un filo (l'ingresso dei dati è chiamato DI ed è ripetuto sul DO a fine stringa) rispetto a massa; le stringhe inviate prevedono il controllo di 54 LED RGB (tanti compongono la barra a LED costituente il terreno di gioco e il segnapunti centrale) e quindi conterranno

tanti byte quanti sono i LED, così da impostare per ciascuno luminosità e colore.

Nella fase iniziale, quindi subito dopo l'accensione (alimentazione) e l'inizializzazione, il firmware caricato in Arduino Nano fa sì che questa comandi i LED Neopixel della strip in modo che ai lati della strip si accendano 6 LED ciascuno, di colore verde a indicare che il gioco è pronto; trascorso un breve intervallo necessario ai concorrenti di prepararsi, il gioco può avere inizio.

Questa condizione rimane fin quando la gara avrà inizio, ovvero finché uno dei pulsanti viene premuto, ma non uno a caso, bensì quello del giocatore di sinistra ovvero quello associato al pulsante rosso (P1). Quando si preme tale pulsante, il firmware inizialmente fa aumentare la luminosità dei gruppi di sei LED ai lati della strip NeoPixel, quindi fa scorrere il punto luminoso bianco che rappresenta la palla verso il lato dell'avversario, il quale deve premere il proprio pulsante in un arco di tempo che va da quando il punto tocca il LED acceso più interno a quando raggiunge l'ultimo, ossia il più a lato. Se vi riesce, il punto luminoso cambia direzione e torna al primo giocatore, che deve ribattere la palla. Per capire dove si trova la pallina, il punto luminoso in movimento cambia colore, passando da bianco



Il prototipo completo e in funzione.

al colore del giocatore del lato (ossia dei 6 LED inizialmente verdi) dove si trova.

Ad esempio se arriva alla racchetta del blu, il punto nel percorrere i sei LED passa da bianco a blu e questo avviene sia nello spostamento verso l'estremità della striscia di LED, sia nel movimento di ribattuta, verso il centro.

In base al momento in cui viene rilevata la pressione del pulsante nella direzione del movimento della palla, il firmware valuta se il contatto con la racchetta del giocatore che respinge avviene correttamente o meno, nel qual caso il gioco si ferma e viene assegnato a chi ha battuto o respinto per ultimo.

La determinazione avviene sincronizzando l'indirizzamento dei LED sulla striscia NeoPixel con la pressione del pulsante, ovvero, per fare un esempio, se 27 LED fanno metà strip e il giocatore può ribattere negli ultimi 6, il firmware conta quanti indirizzi di avanzamento della palla dà e verifica se dal 22° al 27° viene rilevata la pressione del pulsante del giocatore che si trova da quella parte. Quindi, in caso affermativo la palla riparte nella direzione opposta e in caso contrario viene assegnato il punto (e la ripresa del gioco) a chi ha battuto per ultimo, riprendendo la routine di gioco da capo. Ogni volta che un giocatore riesce a ribattere, il cicalino emette una nota acustica.

È importante notare che la sincronizzazione tra l'avanzamento dei LED NeoPixel e l'attivazione del pulsante corrisponde alla direzione verso cui si

In commercio da tempo e molto utilizzati soprattutto in ambiente Arduino, i NeoPixel sono LED RGB "intelligenti" ossia dotati di controller incorporato, facilmente integrabili nell'ambiente Arduino grazie ad apposite librerie. Una particolarità dei LED NeoPixel è che possono essere collegati in cascata in modo che la linea dati da uno passi al successivo, il che è un vantaggio in termini di semplicità circuitale, ma impone un limite al numero di LED collegabili quando si desidera visualizzare effetti ad alta velocità. Questa limitazione non riguarda il nostro progetto, perché ci limitiamo a visualizzare punti o barre colorate. Ogni LED RGB può essere gestito individualmente tramite un apposito comando inserito nella stringa seriale e può produrre fino a 256 tonalità del proprio colore, determinando un totale di 16.777.216 combinazioni di colore.

Il protocollo di comando del sistema NeoPixel prevede l'invio di gruppi di tre byte in una stringa di 24 bit, ognuna delle quali con-

tiene lo stato di illuminazione di ciascun colore base (prima gli otto bit del verde, poi quelli del rosso e infine quelli del verde).

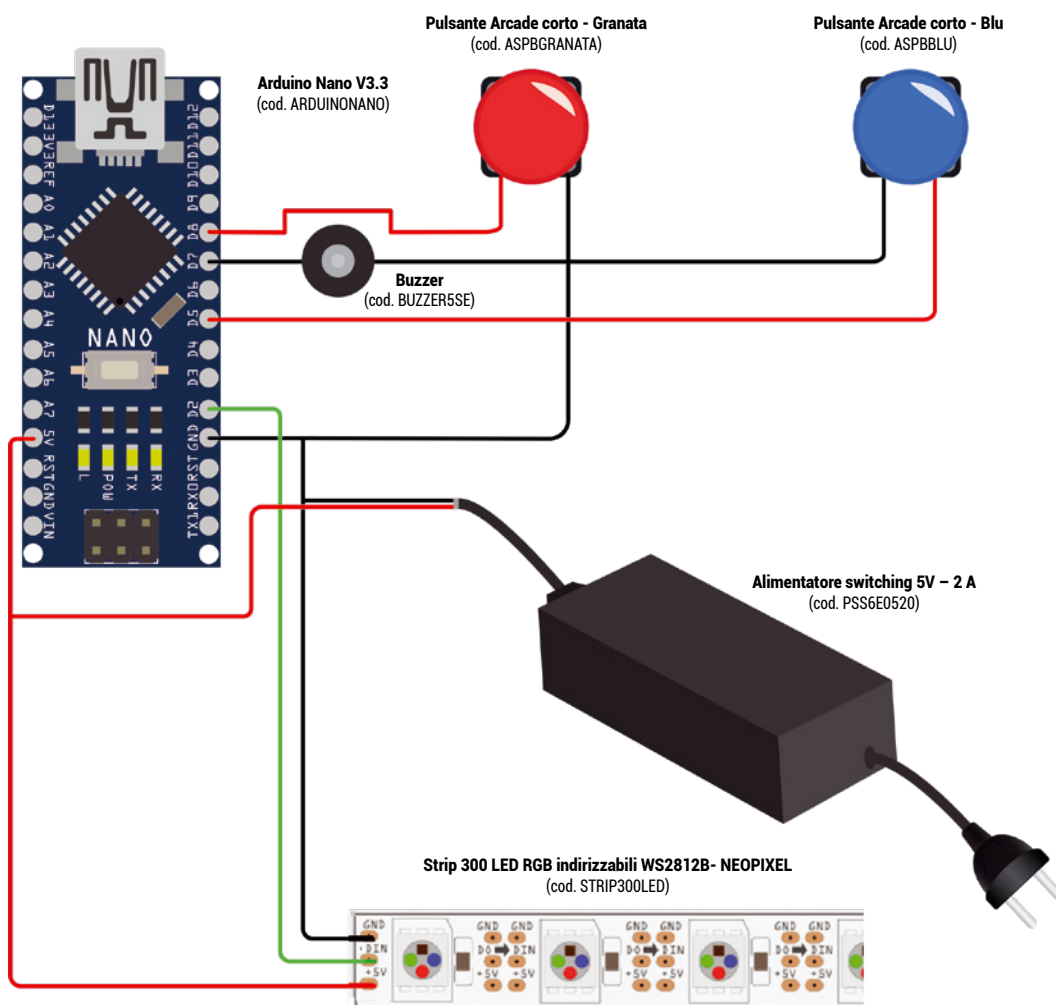
Il canale dati usato per la comunicazione con i LED NeoPixel e quindi con le strip è simile al tipo one-wire. L'alimentazione prevista per i LED NeoPixel è a tensione continua di 5 volt e la comunicazione avviene a un massimo di 800 kbps.

Per ogni strip di LED è possibile impostare a piacimento la frequenza di refresh, in modo da rendere impercettibili determinati giochi di luce.

Più strip possono essere collegati in cascata o in parallelo per realizzare vari effetti, ma in questo caso tale configurazione non interessa; sappiate comunque che più strip si connettono a un unico canale dati, più, fermo restando il massimo data-rate consentito, verrà limitata la frequenza di refresh. In breve, la frequenza di refresh e quindi la velocità di accensione/spegnimento dei singoli LED, è inversamente proporzionale alla quantità di LED da gestire.



Fig. 2
Schema
di cablaggio.



muove il punto luminoso. Questa sincronizzazione inizia sempre dal LED dove è avvenuta l'ultima ribattuta. Quando uno dei due giocatori non riesce a premere il pulsante in corrispondenza della zona illuminata del suo lato, la manche è terminata e un punto luminoso del colore del giocatore vincente si accende dal suo lato; inoltre questo LED lampeggia. Riprendendo il gioco il LED si accende a luce fissa ma con bassa luminosità e il punteggio rimane sulla strip durante le successive partite. Raggiunti i tre punti, ci si aggiudica un set e si vince il gioco, condizione evidenziata dal suono del cicalino piezo e dal lampeggio dei LED indicanti il punteggio; ora tutto ritorna come dal principio, nel senso che si può giocare un altro game o manche che dir si voglia. Ogni fase cruciale del gioco è accompagnata dall'emissione di una nota acustica da parte del cicalino piezoelettrico, che suonerà ogni volta che

un giocatore ribatterà. Non è prevista la nota acustica per l'aggiudicazione di una manche, perché a ciò provvede già la segnalazione visiva sulla strip NeoPixel. Invece all'aggiudicazione di un set, ossia al raggiungimento dei tre punti, la metà dei LED (27) sul lato del vincitore, si illumina ad arcobaleno, lampeggiando per alcuni secondi.

IL FIRMWARE

Lo sketch che implementa il nostro gioco del ping-pong fa uso della libreria di Adafruit *NeoPixel.h*, che viene inclusa all'inizio, prima della definizione dei pin e delle variabili, con la riga:

```
#include <Adafruit_NeoPixel.h>
```

Per il corretto funzionamento vengono incluse anche le librerie *FastLED.h* (`#include <FastLED.h>`) per il controllo dei LED della strip e la *PinButton.h*

con la quale si legge lo stato dei pulsanti. Subito dopo vengono definiti il pin di controllo della strip e il numero di LED da cui sarà composta la stessa, oltre che dichiarate le variabili:

```
#define NUM_LEDS 54           // Number of LEDs
#define DATA_PIN 3          // Data-Pin
#define BUZZER_PIN 7         // Pin per il buzzer

int maxBright      =      155; // max brightness

byte playerBtnPin[] = {5, 8}; // Pins for buttons
byte playerLedPin[] = {6, 9}; // Pins for LEDs

int gameSpeedMin    =      50; // min game-speed
int gameSpeedMax    =      5;  // max game-speed
int gameSpeedStep   =      1;  // fasten up when
                        // change direction
int ballSpeedMax     =      1;  // max ball-speed
int ballBoost0       =     25;  // superboost on
                        // last position
int ballBoost1       =     15;  // boost on forelast
                        // position
byte playerColor[]  = {0, 96}; // red & green
int winRounds       =     10;  // x winning rounds
                        // for winning game
int endZoneSize     =      6;  // size endzone
int endZoneColor    =    160;  // color endzone
```

Si definiscono quindi gli ingressi digitali di Arduino Nano destinati ai pulsanti e si imposta la libreria dei NeoPixel per la modalità di controllo voluta e la frequenza di refresh, che nel nostro progetto è di 800 kHz; la relativa stringa è la seguente:

```
Adafruit_NeoPixel neopixel (NUM_LEDS, 2, NEO_GRB +
NEO_KHZ800);
```

Si entra quindi nel `setup()` da dove viene inizializzata la libreria `neopixel.h` e si imposta l'attività delle periferiche (pulsanti, cicalino, variabili ecc.) quindi si avvia il loop, che implementa il gioco e lo fa svolgere nel modo che abbiamo descritto qualche paragrafo indietro.

Il codice corrispondente allo sketch da caricare nell'Arduino Nano è descritto parzialmente (per ragioni di spazio sulla rivista) nel **Listato 1**. Per l'esattezza nel **Listato 1** riportiamo il loop, che sostanzialmente si limita a richiamare la funzione `menu`; quest'ultima determina le azioni correlate alla pressione dei pulsanti il cui significato cambia in base alle fasi e condizioni di gioco. trovate anche la funzione `bounce` che implementa

il debouncing dei pulsanti da firmware, semplicemente verificando se `digitalwrite` è a LOW e successivamente considera un nuovo livello basso solo dopo che sono trascorsi (delay) 5 ms. Spiegheremo in fondo a questo articolo come caricarlo dal Personal Computer ed eseguirlo; ma ora passiamo agli aspetti pratici del progetto.

REALIZZAZIONE PRATICA

Il circuito elettronico che permette di implementare il gioco elettronico del Pong si realizza in modo semplice per cablaggio dei pochi elementi che lo compongono, utilizzando jumper Arduino e fili in guaina di piccolo diametro.

Costruire questo gioco è molto semplice e praticamente alla portata di tutti: basta procurarsi una strip NeoPixel da 54 LED, alimentata a 5V, quindi una base di appoggio cui incollarla con nastro biadesivo o colla per plastica (ma anche silicone sigillante); i tre poli di un capo della strip andranno collegati così:

- DI al pin D3 della scheda Arduino Nano;
- 5V all'omonimo pin della Arduino Nano, ovvero al 5V dell'alimentazione;
- GND all'omonimo pin della Arduino Nano, ovvero al negativo dell'alimentazione.

Si devono quindi collegare i due pulsanti, assegnando il colore rosso alla linea D8 di Arduino Nano e il blu all'I/O digitale D5 della stessa; ricordate che il giocatore 1 è associato al colore blu e il giocatore 2 al rosso. I pulsanti devono avere ciascuno un polo connesso alla linea GND di Arduino che farà da massa comune.

Bisogna anche collegare il cicalino che servirà a dare le segnalazioni acustiche ai giocatori; per l'esattezza il trasduttore elettroacustico verrà pilotato attraverso la linea digitale D7 di Arduino Nano, pertanto dovrete collegarne il positivo a tale pin e il negativo alla massa comune, insieme ai pulsanti.

Tutta l'elettronica deve essere cablata in modo volante e magari prevedendo di racchiudere la Arduino Nano in un piccolo contenitore in plastica che la protegga dai contatti accidentali.

Anche la strip NeoPixel sarebbe buona cosa se trovasse posto in un contenitore, magari una canalina da impianti elettrici con il coperchio trasparente o un profilato di alluminio ad U al quale applicare una striscia di plexiglass trasparente che renda visibili le luci ma impedisca l'entrata della polvere. Insomma, vedete un po' voi cosa riuscite a procurarvi e naturalmente quel che più si avvicina al



▼ Listato 1

```
void loop()
{
  menu();
}

// Menu (set brightness and start game)
void menu()
{
  digitalWrite(playerLedPin[0], HIGH); // activate both LEDs
  digitalWrite(playerLedPin[1], HIGH);

  GeneratePlayField(maxBright);        // show play-field
  FastLED.show();

  // set brightness (button player 0)
  if (buttonBounce(0, 5) == true)
  {
    maxBright += 50;
    if (maxBright > 255)
    {
      maxBright = 55;
    }
    GeneratePlayField(maxBright);
    FastLED.show();

    while (buttonBounce(0, 5) == true) {} // wait for button release
  }

  // start game (button player 1)
  if (buttonBounce(1, 5) == true)
  {
    digitalWrite(playerLedPin[0], LOW); // deactivate both LEDs
    digitalWrite(playerLedPin[1], LOW);
    activeGame = true;

    while (buttonBounce(1, 5) == true) {} // wait for button release

    game();
  }
}

// function to debounce button (true == pressed, false == not pressed)
boolean buttonBounce(byte button, byte bounceTime) // bounce the button
{
  boolean result = false;

  if (digitalRead(playerBtnPin[button]) == LOW)
  {
    delay (bounceTime);
    if (digitalRead(playerBtnPin[button]) == LOW)
    {
      result = true;
    }
  }
  return result;
}
```

vostro gusto.
Per il cablaggio seguite il disegno proposto nella **Fig. 2**, rammentando che i fili occorrenti possono essere di piccola sezione: anche 0,5 mmq.
Una volta collegato il tutto, verificate con lo

schema che ogni filo sia al posto giusto e che tutto risulti collegato, quindi l'hardware sarà pronto.
L'alimentazione al circuito può essere fornita dalla presa microUSB di Arduino Nano o meglio ancora direttamente al pin Vin e massa della scheda



Arduino, tramite una presa jack femmina volante adatta allo spinotto dell'alimentatore che userete, che dev'essere da 5V stabilizzati, 1A (va bene un alimentatore generico tipo wall-cube). Le basi di appoggio dei pulsanti possono essere stampate in 3D seguendo i file .STL scaricabili dal nostro sito <https://ei.futuranet.it> insieme agli altri file di questo numero della rivista, ma anche ricavate da un bicchiere in plastica un po' robusto o da un cono sempre in plastica oppure, magari, da un piccolo imbuto rovesciato al quale tagliare il "becco" di uscita. Tutti i componenti necessari a realizzare il gioco qui proposto sono reperibili presso la Futura Elettronica, anche on-line sul sito futurashop.it.

LA PROGRAMMAZIONE

Affinché il nostro gioco elettronico funzioni, va caricato nella memoria Flash della board Arduino Nano lo sketch che abbiamo scritto appositamente per gestirlo e che rendiamo disponibile sul nostro sito Internet <https://ei.futuranet.it> insieme agli altri file del progetto.

Il caricamento va effettuato mediante l'apposito IDE Arduino eseguito su Personal Computer e attraverso la connessione USB. Per programmare l'Arduino Nano, connettetelo al computer tramite un cavo USB A-mini USB. Assicuratevi che il computer sia dotato dell'ultima versione dell'IDE Arduino. Una volta collegato e riconosciuto, Arduino sarà alimentato direttamente via USB, eliminando la necessità di una fonte di alimentazione esterna. Procedete aprendo lo sketch desiderato selezionando "File > Apri" dall'IDE. Successivamente, accedete al menù "Strumenti" e, sotto il comando "Scheda", scegliete "Arduino Nano" per impostare correttamente la scheda.

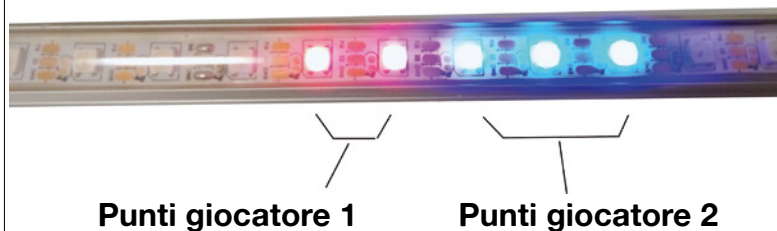
Successivamente cliccate sul pulsante "Carica", rappresentato da una freccia verso destra situata nella parte superiore della barra degli strumenti dell'IDE, sotto la barra dei menu.

Attendete fino al completamento del caricamento dello sketch. Una volta terminato, potete chiudere l'IDE Arduino, scollegare il cavo dall'Arduino e procedere a testare il circuito per assicurarvi che funzioni correttamente.

Provate dunque a giocare la vostra prima partita: attendete che la strip di LED NeoPixel si stabilizzi con le due estremità colorate di rosso e di blu, quindi date inizio al gioco ricordando che il giocatore abilitato a iniziare è sempre il numero 1, vale a dire dalla pressione del pulsante rosso; se si preme il blu, nulla accade.

Tenete presente che un set si conclude quando

AREA CENTRALE DELLA STRIP



uno dei giocatori raggiunge tre punti. Questo sarà indicato dall'illuminazione di tre LED nella zona centrale: rossi per il giocatore 1 (P1) se sarà lui a vincere, o blu per il giocatore 2 (P2) se sarà lui a trionfare nel gioco.

L'immagine proposta nella **Fig. 3** vi delucida a riguardo, perché in essa vedete quello che mostra l'area centrale della strip NeoPixel quando ad aggiudicarsi il gioco è il giocatore 2 (pulsante blu) e invece il giocatore 1 ha totalizzato 2 punti. Bene, giunti a questo punto riteniamo di avervi fornito tutte le informazioni necessarie per non solo realizzare ma anche sfruttare al meglio questo entusiasmante gioco elettronico, capace di rendere ancora più vivaci e divertenti i vostri momenti di svago. Vi invitiamo a visitare il nostro sito web per scaricare il codice sorgente del gioco, con la possibilità di personalizzarlo secondo le vostre preferenze. Questa potrebbe essere l'occasione perfetta per mettere alla prova la vostra creatività e ampliare ulteriormente le funzionalità del vostro progetto. ■

Fig. 3
Punteggio visualizzato in caso di vittoria del giocatore blu.



COSA OCCORRE ?

Codice	Descrizione	Prezzo € IVA inclusa
ARDUINONANO	Arduino Nano	29,00
STRIP300LED	Strip 300 LED RGB indirizzabili	49,00
PSS6E0520	Alimentatore switching 5V – 2 A	12,50
ASPBGRANATA	Pulsante Arcade colore granata	2,50
ASPBBLU	Pulsante Arcade colore blu	2,50
BUZZER5SE	Buzzer senza elettronica 5V	0,50

IL MATERIALE VA RICHIESTO A:

Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 - www.futurashop.it



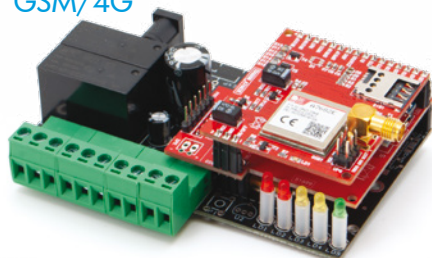
Gestisci e controlla la tua casa con i TELECONTROLLI GSM

Attiva e controlla da remoto qualsiasi dispositivo elettrico o elettronico: dalla centralina antifurto all'impianto di irrigazione, dalle luci di casa al sistema di condizionamento e raffrescamento. Configurazione via SMS/PC con predisposizione per gruppo di continuità.

4G
LTE



COMBINATORE TELEFONICO GSM/4G

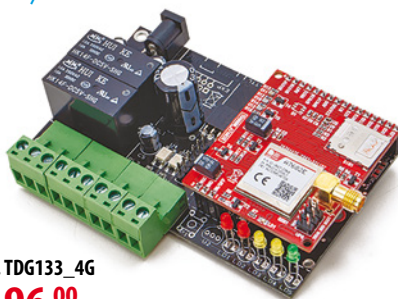


cod. TDG135MP3_4G
€125,00

- 2 Ingressi e 2 Uscite
- Menu vocale in Italiano
- Messaggi di allarme vocali
- Invio segnalazioni di allarme ingressi

Disponibile anche in versione 2G cod. TDG135MP3 • Euro 99,00

TELECONTROLLO BICANALE GSM/4G

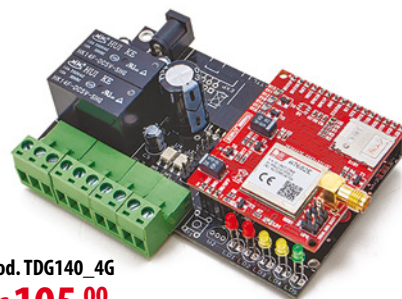


cod. TDG133_4G
€96,00

- 2 Ingressi e 2 Uscite
- Funzione Apricancello
- Invio segnalazioni di allarme ingressi

Disponibile anche in versione 2G cod. TDG133 • Euro 79,00

TELECONTROLLO GSM/4G CON COMANDI DTMF

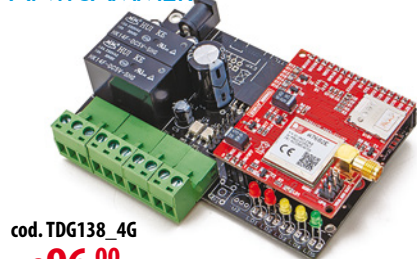


cod. TDG140_4G
€105,00

- 2 Ingressi e 2 Uscite
- Menu gestibile via DTMF
- Invio segnalazioni di allarme ingressi

Disponibile anche in versione 2G cod. TDG140 • Euro 88,00

TELEALLARME GSM/4G CON ANTI JAMMER

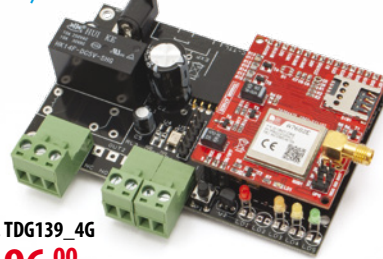


cod. TDG138_4G
€96,00

- 2 Ingressi e 2 Uscite
- Funzione anti Jammer o Telecontrollo
- Invio segnalazioni di allarme ingressi e Jammer

Disponibile anche in versione 2G cod. TDG138 • Euro 79,00

TERMOSTATO CON CONTROLLO GSM/4G



cod. TDG139_4G
€96,00

- Controllo riscaldamento
- Manuale/Automatich/Asservito Termostato
- Allarmi temperatura configurabili
- Invio segnalazioni di allarme temperatura e blocco caldaia

Disponibile anche in versione 2G cod. TDG139 • Euro 79,00

APRICANCELLO GSM/4G



cod. TDG134_4G
€92,00

- 1 Ingresso
- Funzione Apricancello

Disponibile anche in versione 2G cod. TDG134 • Euro 75,00

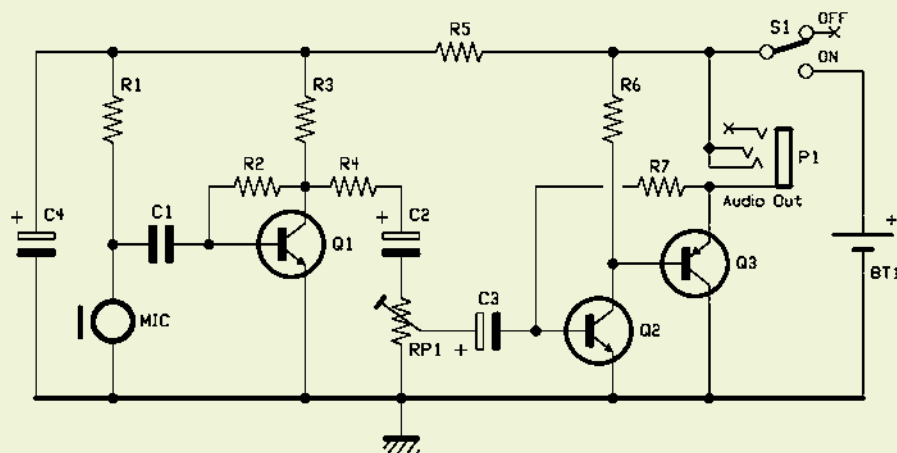


AMPLIFICATORE ACUSTICO

di ALESSANDRO SOTTOCORNOLA

Amplifica voci, suoni e rumori captati nell'ambiente da una capsula microfonica e li rende udibili in un auricolare stereo; utilizzabile anche come apparecchio acustico.

Può capitare di trovarsi in condizioni in cui le capacità del proprio udito non sono sufficienti o nelle quali si desidera ascoltare il rumore che proviene da una certa zona lontana o una voce da lontano; in casi del genere potrebbe essere utile un dispositivo come quello proposto in queste pagine, che è sostanzialmente un microfono fortemente amplificato con uscita di potenza sufficiente a pilotare una coppia di auricolari, ovvero una piccola cuffia stereofonica. Grazie a questo progetto sarà possibile amplificare qualsiasi suono presente nell'ambiente, dalla voce ad un comune rumore di oggetti e/o musica, in modo da ascoltarlo amplificato in cuffia.



Sarà inoltre possibile regolare l'amplificazione tramite un piccolo trimmer in modo da ottenere il volume d'ascolto desiderato, ovvero in maniera da aumentare entro certi limiti l'intensità del suono o rumore captato nell'ambiente, che si desidera ascoltare in cuffia.

Le qualità del circuito sono tali da poterlo utilizzare anche da parte di chi soffre di ipoacusia, ovvero è debole di udito, giacché si comporta da vero e proprio apparecchio acustico esterno, seppure tecnicamente e a livello elettromedicale non possa essere considerato tale; però in alcuni casi può svolgere le veci dell'apparecchio acustico, anche solo per periodi limitati o ad esempio quando l'effettivo apparecchio acustico sia guasto.

IL PROGETTO

Andiamo dunque a vedere di cosa si tratta aiutandoci con lo schema elettrico proposto in queste pagine, che meglio descrive il circuito. Come vedete, si tratta sostanzialmente di un amplificatore ad elevato guadagno al cui ingresso è applicata una capsula microfonica del tipo electret-condenser. L'amplificatore è monofonico, anche se alla sua uscita è possibile collegare cuffie e auricolari stereo, perché in realtà i due canali L ed R della presa jack stereo sono cortocircuitati tra loro. Partiamo con l'analisi circuitale: il microfono è una capsula di tipo electret (basata quindi su un condensatore avente un'armatura polarizzata a permanenza) a due fili con preamplificazione a jFET, il cui drain è collegato al polo positivo e viene polarizzato attraverso la resistenza R1. Il condensatore C1 blocca la componente continua del circuito di polarizzazione lasciando transitare il solo segnale audio e separando la polarizzazione

della capsula da quella dello stadio a transistor che segue.

Il segnale elettrico derivante dall'audio captato dalla membrana della capsula microfonica, viene portato dal C1 alla base del transistor Q1, un NPN che opera nella configurazione a emettitore comune e che viene retroazionato secondo lo schema parallelo-parallelo.

Questo stadio amplificatore è di tipo invertente e guadagna circa 120 volte in tensione, quindi restituisce sul collettore del transistor un segnale BF 120 volte più ampio di quello presente sul condensatore C1, però in opposizione di fase, giacché la configurazione adottata è di tipo invertente.

Il segnale così amplificato passa attraverso la resistenza R4 e il condensatore elettrolitico C2, raggiungendo il trimmer RP1, che ha la funzione di controllare il volume d'ascolto parzializzando il segnale che dal proprio cursore va allo stadio amplificatore seguente, al quale giunge tramite l'accoppiamento capacitivo implementato dall'elettrolitico C3, la cui funzione è, al pari degli altri condensatori di disaccoppiamento presenti nel circuito, lasciar passare il segnale audio e bloccare la componente continua degli stadi che unisce.

Lo stadio amplificatore che segue è formato da due transistor in cascata: Q2 e Q3, rispettivamente NPN e PNP; il primo è configurato a emettitore comune e riceve in base il segnale prelevato dal cursore del trimmer (montato a potenziometro) del volume RP1, quindi lo amplifica presentandolo sul proprio collettore invertito di fase, dove è collegata la base del Q3.

Quest'ultimo (un PNP configurato a collettore comune) amplifica in corrente il segnale già amplificato in tensione da Q2 e con il proprio emettitore

alimenta i trasduttori dell'auricolare attraverso i contatti della presa jack siglata Audio Out nello schema elettrico.

Il blocco amplificatore formato da Q2 e Q3 amplifica il segnale in potenza (Q2 principalmente in tensione, mentre Q3 solo in corrente perché è un inseguitore di emettitore) ed ha un'unica rete di retroazione che riporta alla base dell'NPN una porzione del segnale di uscita prelevata dall'emettitore del PNP.

Il complesso è retroazionato secondo lo schema parallelo-parallelo ed è invertente, cosicché la fase del segnale in uscita rispecchierà quella di ingresso. L'inversione di fase rispetto al segnale preamplificato da Q2 è stata voluta per assicurare una certa coerenza di fase fra ciò che verrà udito in auricolare e l'audio che viene captato nell'ambiente dalla capsula microfonica, il che è importante perché parte del suono nell'ambiente può essere udito sovrapposto all'audio dell'auricolare, cosa che diventa ancor più rilevante se si utilizza un auricolare singolo invece di uno stereo.

La rete di retroazione dello stadio di uscita comprende non solo la resistenza R7 ma anche, in vero, parte della resistenza equivalente del trimmer vista dalla base del Q2 e che varia in base alla posizione del cursore dello stesso RP1.

Mediante il guadagno dello stadio formato dai transistor Q2/Q3 è intorno alle 60 volte in tensione, il che determina un'amplificazione complessiva del circuito di circa 720 volte, quando il cursore del trimmer è tutto verso l'estremo collegato al

condensatore elettrolitico C2.

Notate che nel circuito non si trova alcun filtro che tagli o attenui frequenze della banda audio, pertanto qualsiasi suono verrà amplificato indistintamente.

In questo, il circuito differisce da un apparecchio acustico, nel quale alcune bande vengono filtrate o trattate per ottenere la migliore risposta possibile dell'orecchio della persona ipoudente.

Chiudiamo l'analisi dello schema elettrico del nostro circuito con l'alimentazione, che viene fornita all'intero circuito da una pila a bottone di tipo CR2032 e che può essere applicata o interrotta mediante il deviatore a slitta S1, qui utilizzato da interruttore, cablando quindi il contatto comune (verso l'interno del circuito) e uno degli estremi. Portando il comune sull'estremo ON, la pila alimenta il circuito, ovvero direttamente lo stadio di uscita facente capo a Q2 e Q3, mentre la tensione necessaria allo stadio di ingresso, che è molto suscettibile ai disturbi propagabili sulla linea di alimentazione quando la sezione di potenza pilota i trasduttori dell'auricolare, viene filtrata dalla cella RC facente capo al resistore R5 e al condensatore elettrolitico C4.

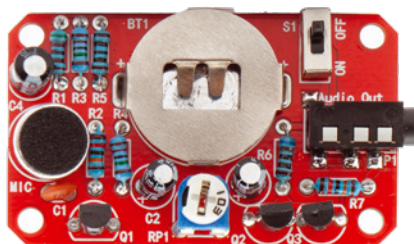
REALIZZAZIONE PRATICA

Bene, descritto il circuito possiamo procedere con le note costruttive, premettendo che per questo dispositivo è stato disegnato un apposito circuito stampato a doppia faccia le cui tracce lato rame potete trovare e scaricare dalla pagina di presenta-

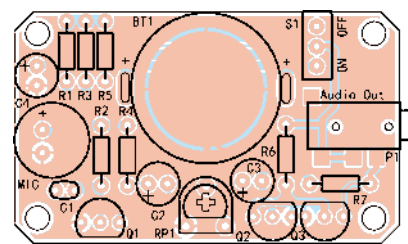
piano di MONTAGGIO

Elenco Componenti:

R1: 5,1 kohm
R2, R7: 120 kohm
R3, R4: 1,5 kohm
R5: 100 ohm
R6: 680 ohm
C1: 100 nF ceramico
C2, C3: 4,7 μ F 16 VL elettrolitico
C4: 100 μ F 16 VL elettrolitico
Q1, Q2: S9014
Q3: S9012
RP1: Trimmer 10 kohm
S1: Deviatore a slitta



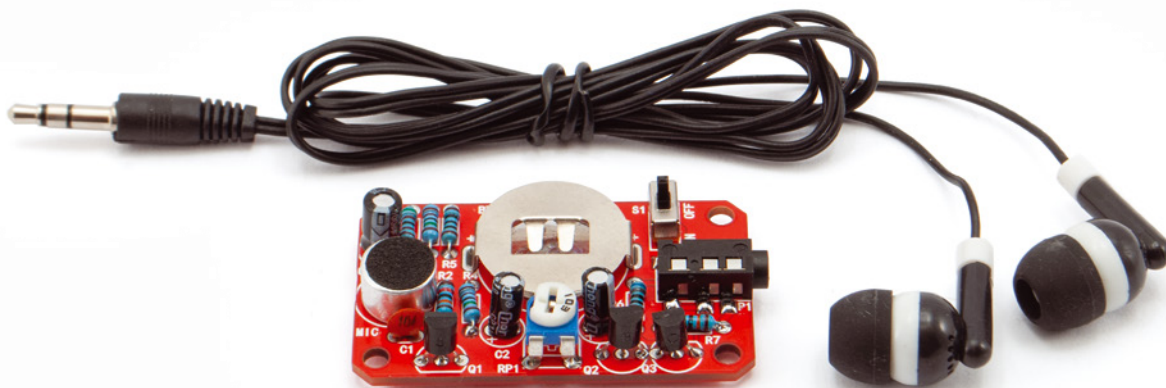
MIC: Microfono
BT1: Porta batteria per CR2032



Varie:
- Circuito Stampato FK019 (52x31 mm)



Il circuito si abbina a un auricolare stereofonico.



zione di questo numero della rivista.

Il PCB potete realizzarlo utilizzando il servizio di Futura Elettronica PCBPRODUCTION: è sufficiente caricare il file gerber e scegliere il colore della basetta (nelle foto di questo articolo rossa). Ottenuto il PCB potete procedere con la saldatura dei componenti, partendo da quelli a più basso profilo, quindi le resistenze fisse e il trimmer miniatura orizzontale, per poi proseguire con i condensatori (prima il C1, che è non polarizzato, poi gli elettrolitici, ognuno da orientare come indicato nel piano di montaggio) e i tre transistor, che hanno un verso di inserimento e vanno disposti come mostrato nel piano di montaggio che trovate in queste pagine.

Proseguite con il connettore jack da 3,5 mm per le cuffie, il deviatore a slitta con terminali dritti da circuito stampato, il portapila per la batteria CR2032 e infine la capsula microfonica electret-condenser, la quale ha un preciso verso di inserimento da rispettare: ricordate, a riguardo, che il terminale negativo (quello di massa) i riconosce perché visibilmente collegato all'involucro metallico della capsula.

Terminato il montaggio dovete verificare attentamente che tutto sia stato saldato correttamente e soprattutto di non aver causato dei cortocircuiti in fase di saldatura (baffi di stagno) nel qual caso con l'aiuto della treccia dissaldante dovete rimuovere lo stagno in eccesso e verificare nuovamente il tutto.

UTILIZZO E REGOLAZIONI

Bene, arrivati a questo punto, se il montaggio è tutto a posto pensate ad un piccolo contenitore in plastica (come ad esempio l'SC-704 disponibile sul sito di Futura Elettronica) dove potrete collocare il circuito stampato, lavoratelo per rendere accessibile la presa jack dell'auricolare nonché per affacciare all'esterno la membrana della capsula microfonica (ossia la parte superiore coperta dal feltrino) e la levetta dell'interruttore a slitta, quindi sistematevi il PCB e inserite la pila a bottone nel proprio alloggiamento accertandovi che sia nel giusto verso; a questo punto spostate l'interruttore a slitta nella posizione ON dopo aver inserito nella presa jack lo spinotto di un auricolare o una cuffietta stereofonica con impedenza di $16+32$ ohm, quindi andate a regolare il trimmer con un piccolo cacciavite a lama per regolare il livello d'ascolto fino a raggiungere il volume desiderato. Chiudete pure il contenitore, perché il vostro dispositivo è pronto per essere utilizzato. Ricordiamo che in virtù della forte amplificazione di cui è capace, questo circuito può trovare varie applicazioni, tra le quali fare da "supermicrofono". In questo caso, collocandolo nella bocca di un cono o di una tromba può captare suoni e rumori da una specifica zona, funzionando da captatore direttivo, consentendo di sentire anche da molto lontano. □



COSA OCCORRE ?

Codice	Descrizione	Prezzo € IVA inclusa
FK019	Amplificatore acustico in scatola di montaggio	9,00
SC-704	Contenitore plastico per amplificatore (da lavorare)	1,50

IL MATERIALE VA RICHIESTO A:

Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 - www.futurashop.it

Lampade da tavolo per ogni esigenza!

A LED, con lente, con ricarica wireless...

Lampada LED con lente d'ingrandimento



€94,00

Cod. TP246

- » Lente d'ingrandimento: 8 diottrie
- » Alimentazione: 220 VAC - 240 VAC / 50 Hz
- » Consumo: 9 W
- » Flusso luminoso: 685 lumen
- » Temperatura colore: 6000-7000 K
- » Montaggio: morsetto metallico per il fissaggio al banco

Lampada da tavolo a LED con lente d'ingrandimento



€39,00

Cod. TP206

- » Lente d'ingrandimento: 3 + 12 diottrie
- » Ingrandimento: 1,75 x + 4 x
- » Alimentazione: 230 VAC / 50 Hz - 5,5 W
- » Flusso luminoso: 400 lumen
- » Temperatura colore: 6000-7000 K
- » Diametro lente: 85 mm
- » Lunghezza braccio di supporto: 250 mm

Lampada LED con lente d'ingrandimento



€72,00

Cod. TP197

- » Lente d'ingrandimento: 3 diottrie
- » Flusso luminoso: 650 lumen
- » Temperatura colore: 6000K-7000K
- » Alimentazione: 230 VAC / 50 H
- » Consumo: 9 W
- » Montaggio: morsetto metallico per il fissaggio al banco

DISPONIBILE ANCHE CON
FLUSSO LUMINOSO DI 700 LUMEN
COD. TP193 - € 94,00

Lampada da tavolo a LED con lente d'ingrandimento



€49,50

Cod. TP248

- » Lente d'ingrandimento: 3 + 12 diottrie
- » Ingrandimento: 1,75 x + 4 x
- » Alimentazione: 230 VAC / 50 Hz - 6,5 W
- » Flusso luminoso: 520 lumen
- » Temperatura colore: 6500 K
- » Diametro lente: 90 mm
- » Lunghezza braccio di supporto: 290 mm

Lampada LED touch dimmerabile con ricarica wireless per smartphone



€34,00

Cod. EFL60

- » Temperatura colore: 2800 K - 3400 K - 4500 K - 5600 K
- » Flusso luminoso: 250 lumen
- » Alimentazione: 5 VDC / 2,5 A - a corredo cavo USB e adattatore da rete 220 VAC (opzionale)
- » Consumo: 5 watt
- » Ricarica wireless: 5 V / 1 A
- » Dimensioni (mm): 126x45x440

Lampada LED professionale con lente d'ingrandimento

- » Lente d'ingrandimento: 3 diottrie
- » Flusso luminoso: 850 lumen
- » Temperatura colore: 6500 K
- » Fissaggio: morsetto da tavolo
- » Lunghezza del braccio: 790 mm
- » Consumo: da 0,5 a 12 W
- » Alimentazione: 230 VAC / 50 Hz - 5,5 W
- » Montaggio: morsetto metallico per il fissaggio al banco



€134,00

Cod. TP243

Lampada professionale a LED dimmerabile - 880 lumen



€98,00

Cod. TP233

- » Alimentazione: 220 VAC - 240 VAC / 50 Hz
- » Consumo: da 1,5 a 14,5 W (dipende dalla luminosità)
- » Flusso luminoso: 880 lumen
- » Temperatura colore: 6000-7000 K
- » Montaggio: morsetto metallico per il fissaggio al banco
- » Dimensioni (mm): 450 x 80 x 400 (braccio)

Prezzi IVA inclusa.

SFRUTTA AL MEGLIO L'ENERGIA DEL SOLE!



**600
WATT**

cod. FR861

€ 340,00

MICROINVERTER SOLARE WIFI AD IMMISSIONE DIRETTA IN RETE

Inverter di tipo "Plug&Play" ideale per realizzare mini impianti fotovoltaici, dove si desidera immettere direttamente in rete l'energia prodotta dall'impianto solare, andando a ridurre l'energia prelevata. Con una potenza massima di picco di 450W per canale (due canali disponibili), è in grado di gestire elevati livelli di potenza. Inoltre, grazie al WiFi integrato, è possibile monitorare il microinverter da remoto tramite un'app gratuita (NEPViewer), consentendo una gestione del sistema fotovoltaico semplice e intuitiva.

Risponde alla norma CEI 0-21 per la realizzazione di mini impianti fotovoltaici.

ATS (AUTOMATIC TRANSFER SWITCH) COMMUTATORE AUTOMATICO DI ALIMENTAZIONE

L'ATS è progettato per commutare automaticamente tra l'alimentazione fornita dai pannelli solari e quella proveniente dalla rete elettrica tradizionale. Se l'energia solare è sufficiente per coprire le esigenze energetiche della casa, l'ATS utilizza automaticamente l'elettricità prodotta dai pannelli solari. Nel caso in cui la produzione di energia elettrica solare non sia sufficiente, l'ATS commuta automaticamente sulla rete elettrica tradizionale garantendo in questo modo l'alimentazione senza interruzioni.



cod. FR863

€ 194,00

PIATTAFORMA PER SVILUPPO DSP AUDIO

di LUCA FREGONESE E MICHELE FRODELLA



Realizziamo un modulo con ESP32 per lo sviluppo di applicazioni audio con effetti DSP personalizzabili in tempo reale.

S

e siete alla ricerca di una soluzione professionale per lo sviluppo di applicazioni audio, la lettura di questo articolo farà al caso vostro! Si tratta di una piattaforma studiata per realizzare elaborazione digitale del suono (DSP), ovvero convertire e processare audio in tempo reale con una potenza di calcolo tale da riuscire ad intervenire e modificare il segnale campione per campione. Intervenire a questo livello significa poter realizzare qualsiasi applicazione inerente al campo musicale: dal semplice player ad effetti complessi come modulazioni, distorsioni, delay e riverberi. Con questa piattaforma avremo quindi la possibilità di costruire i nostri effetti e applicazioni con un livello di qualità pari a quella dei sistemi audio profes-

Fig. 1
Protocollo VoidX
e controllo dei
parametri da
Smartphone/PC.

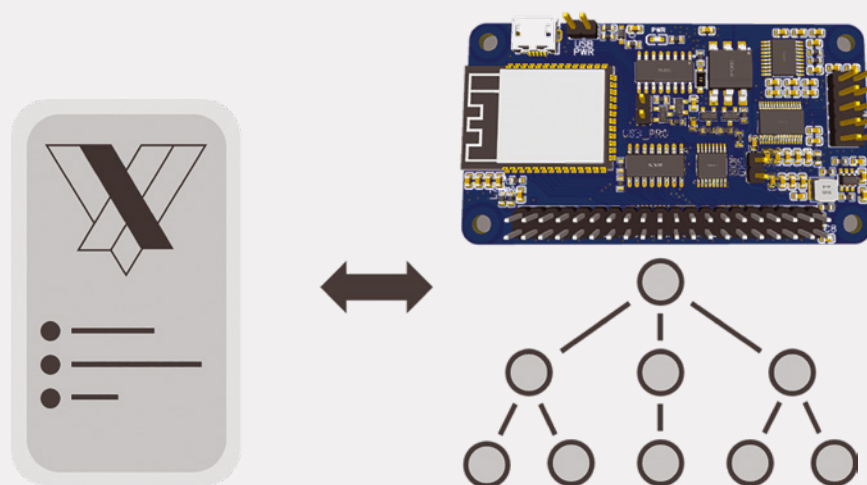


Fig. 2
VoidX-
DSPModule:
pinout.

sionali. Il cuore di questo progetto è VoidX DSP Module (**Fig. 1**), un modulo hardware nato con l'idea di fornire allo sviluppatore tutti gli strumenti per lavorare con il suono. Rispetto ad altri sistemi, hobbistici o professionali, questo lavoro si pone ad un livello decisamente diverso. Si tratta infatti di un'integrazione in un'unica scheda di CPU (ESP32-S3), convertitori audio (ADC e DAC) e periferiche accessorie (MIDI, USB e numerose interfacce I/O) le cui prestazioni audio sono paragonabili o supe-

riori a quelle dei prodotti commerciali professionali (>110dB SNR).

Nota l'architettura fisica, abbiamo creato un contesto software in cui vengono integrati e configurati tutti i driver necessari al funzionamento del sistema. Il risultato di questo lavoro è VoidX Library, un framework software di sviluppo per applicazioni audio real-time con il quale potrete concentrarvi esclusivamente sulla "business logic" della vostra applicazione.

MAIN CONNECTOR

USB D+	1	2	USB D-	
MIDI IN +	3	4	MIDI IN -	
MIDI OUT-	5	6	MIDI OUT+	
GPIO14 / MIDI UART TX	7	8	GPIO13 / MIDI UART RX	
GPIO19 / JTAG D-	9	10	GPIO9	
GPIO10	11	12	GPIO20 / JTAG D+	
GPIO1	13	14	3.3V OUT	
GPIO4	15	16	GPIO5	
GPIO6	17	18	GPIO8	
GPIO15 / I2C SCL*	19	20	GPIO16 / I2C SDA*	
SR IN0	21	22	SR IN1	
SR IN2	23	24	SR IN3	
SERIAL IN	25	26	GPIO21 / SERIAL CLOCK	
GPIO47 / SERIAL CS	27	28	SERIAL OUT	
SR OUT0	29	30	SR OUT1	
SR OUT2	31	32	SR OUT3	
SR OUT4	33	34	SR OUT7	
SR OUT6	35	36	SR OUT5	
GPIO11	37	38	GPIO12	
GND	39	40	POWER IN **	

AUDIO CONNECTOR

NC	10	9	NC
OUT L	8	7	OUT R
OUT GND	6	5	POWER IN **
IN L+	4	3	IN L-
IN R+	2	1	IN R-

	Pin connesso direttamente all'ESP32-S3 senza interferenze
	Pin che si interfacciano ad un circuito interno al modulo
	Pin USB (JTAG o USB-serial)
	Ingresso audio differenziale
	Uscita audio pseudo-differenziale
	Alimentazione positiva
	Alimentazione negativa

* I2C condivisa con codec audio, pull-up interne al modulo

** Alimentazione tra 5V and 12V DC.

pin denominati GPIOx sono direttamente connessi al pin IOx del chip ESP32-S3

pin denominati SR OUTx sono connessi al bit x dello shift register di uscita incluso nel modulo

pin denominati SR INx sono connessi al bit x dello shift register di ingresso incluso nel modulo

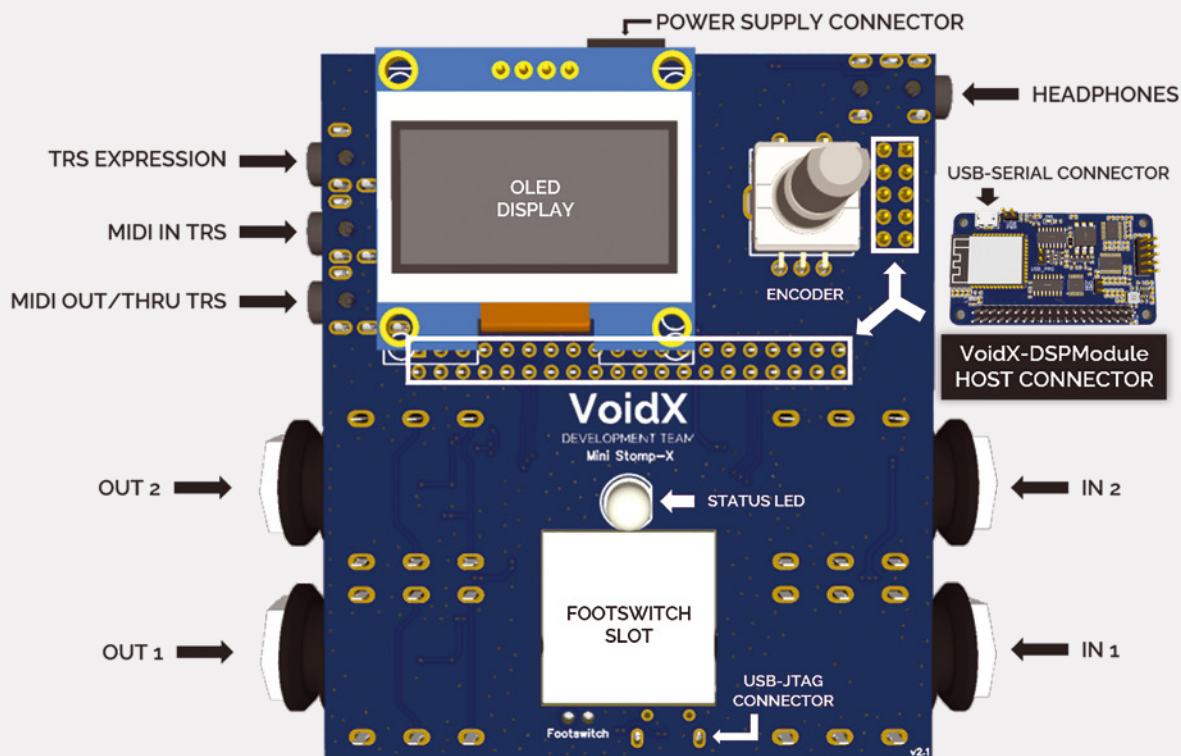


Fig. 3
Mini Stomp-X
Dev-Board,
schema
a blocchi.

Infatti, l'elaborazione dei segnali non è l'unico aspetto da curare nel contesto di un progetto reale. Potrebbe essere necessario dover modificare alcuni parametri (come ad esempio il volume), aggiornare il firmware, collegarsi al PC e magari allo smartphone. Il framework risponde a queste esigenze e fornisce una serie di primitive (nodi) per manipolare dati e parametri rendendoli disponibili attraverso un protocollo di comunicazione appositamente studiato. Abbiamo quindi sviluppato l'app VoidX Control, disponibile gratuitamente per Android e iOS, che ne permette il controllo remoto. Per comprendere le potenzialità di questo lavoro vi presentiamo un caso pratico: la realizzazione di un equalizzatore grafico a 7 bande per chitarra in formato pedale. Al modulo principale, che verrà descritto successivamente, abbiamo affiancato una scheda di espansione con i collegamenti necessari. Con questa scheda potrete, a partire dall'esempio fornito, divertirvi a realizzare il vostro effetto preferito.

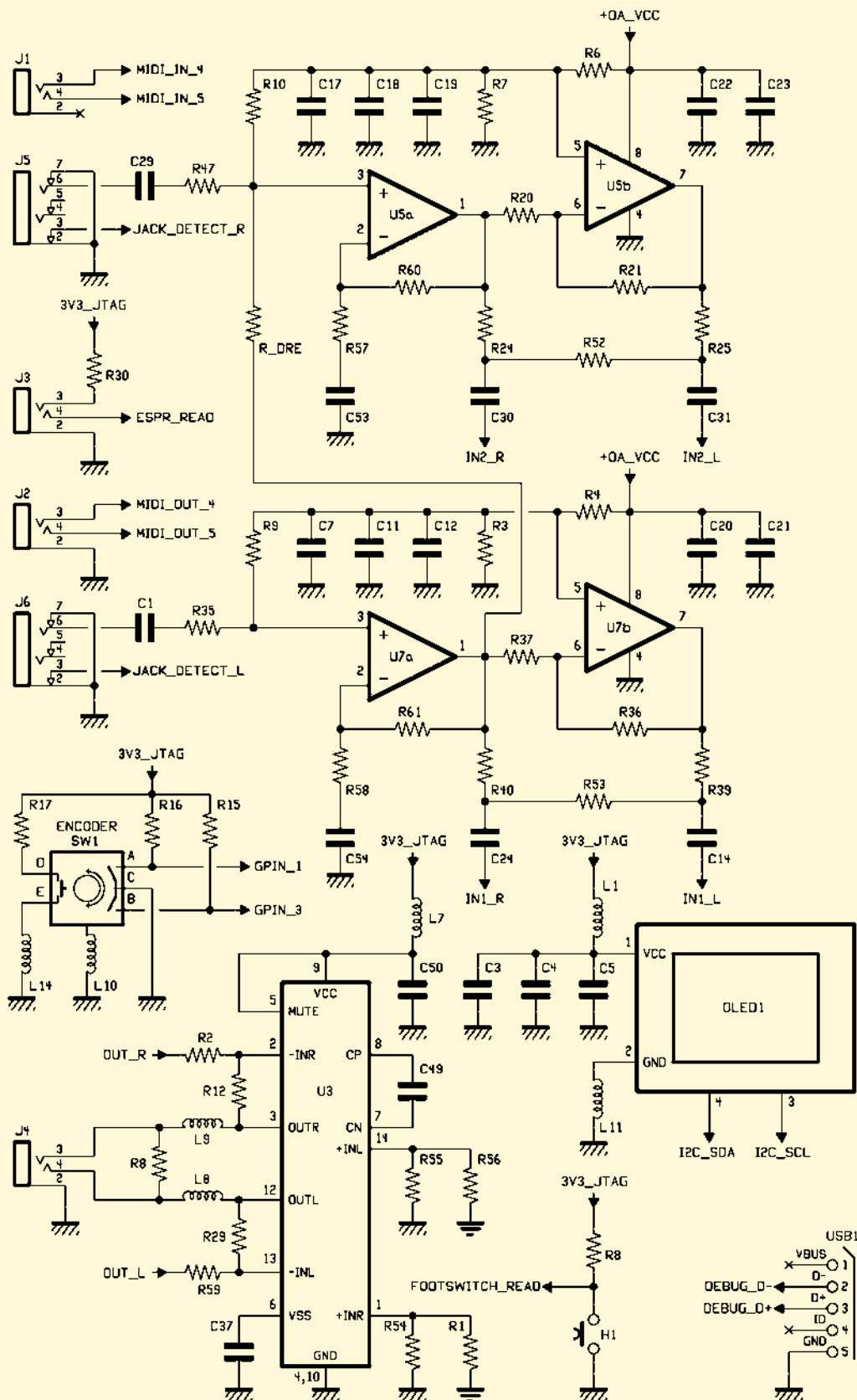
VOIDX DSP MODULE

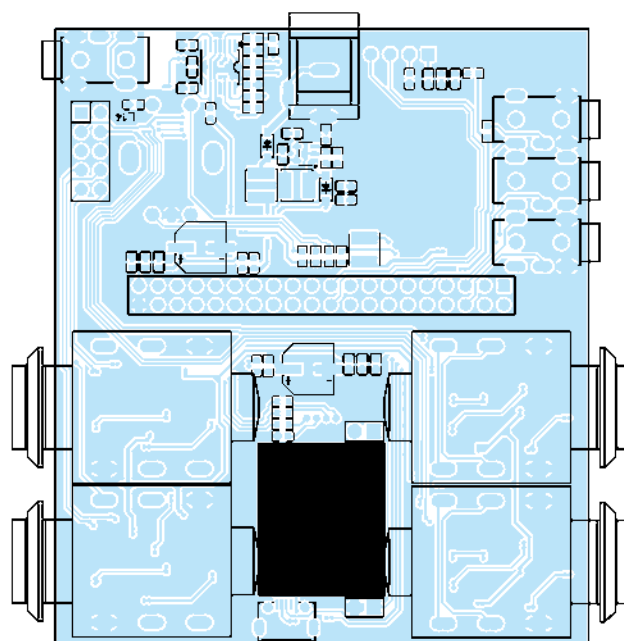
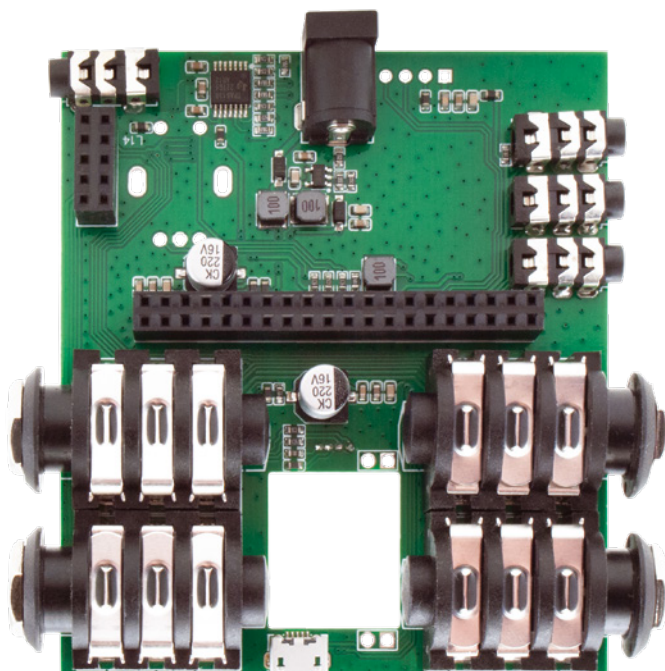
Per costruire la nostra applicazione andiamo a vedere, nel dettaglio, come si presenta il modulo

CARATTERISTICHE TECNICHE

- ❖ **Dimensioni:**
39 x 70 mm
- ❖ **Uscite analogiche pseudo-differenziali ad alta risoluzione (112 dB SNR):**
2
- ❖ **MIDI IN/OUT:** il modulo include tutta la circuiteria MIDI standard, permettendo attacco diretto ai connettori MIDI senza elettronica di contorno
- ❖ **RAM:**
2MB
- ❖ **CPU:**
Espressif ESP32-S3
- ❖ **Ingressi analogici differenziali ad alta risoluzione (110 dB SNR):** 2
- ❖ **Ampio set di interfacce di I/O:**
GPIOs, pin di lettura analogica, PWM, I2C, SPI, UART, interfaccia di espansione I/O tramite shift registers, USB-JTAG
- ❖ **Connettore USB per alimentazione, comunicazione seriale con host (fino a 2Mbps) e programmazione tramite Arduino IDE o ESP-IDF**
- ❖ **Flash:**
8MB







Elenco Componenti:

R1, R2, R56, R59: 56 kohm (0603)
 R3, R4, R6, R7, R33: 22 kohm (0603)
 R34, R43, R44: 22 kohm (0603)
 R5: 4,7 kohm (0603)
 R8, R24, R25, R28, R39, R40, R41, R42, R51: 680 ohm (0603)
 R9, R10: 1 Mohm (0603)
 R11, R30: 180 ohm (0603)
 R12, R19, R27, R29, R45, R50, R54, R55: 47 kohm (0603)

R13, R14, R15, R16, R17: 470 ohm (0603)
 R18, R23, R31, R32: 100 kohm (0603)
 R20, R21, R36, R37, R62: 11 kohm (0603)
 R22, R26, R46, R49: 33 kohm (0603)
 R35, R47, R52, R53, R57, R58: 3,3 kohm (0603)
 R38: 240 ohm (0603)
 R48: 3,3 kohm (0603)
 R60, R61: 0 ohm (0805)

C1, C4, C9, C11, C16, C18, C20, C23, C29, C32: 100 nF ceramico (0603)
 C40, C42, C45: 100 nF ceramico (0603)
 C3, C10, C12, C17, C25, C27, C35, C39, C44, C53, C54: 10 µF ceramico 80603)
 C5, C7, C8, C13, C14, C15, C19, C21, C22, C24, C26, C28, C30, C31, C33, C34, C36, C37, C38, C41, C43, C46,

DSP. Si tratta di una scheda di dimensioni ridotte che espone due connettori: il main connector da 2x20 pin per l'alimentazione e le interfacce I/O e l'audio connector da 2x5 pin per l'interfacciamento audio ad alta fedeltà. L'alimentazione può avvenire mediante USB oppure essere fornita attraverso il connettore principale.

All'interno è presente un alimentatore switching che accetta tensioni da 5V a 15V. In **Fig.2** riportiamo il pinout della scheda e le caratteristiche più importanti. Per ulteriori dettagli potete visitare il sito web www.voidxdevteam.com.

SCHEDA HOST: REALIZZAZIONE HARDWARE

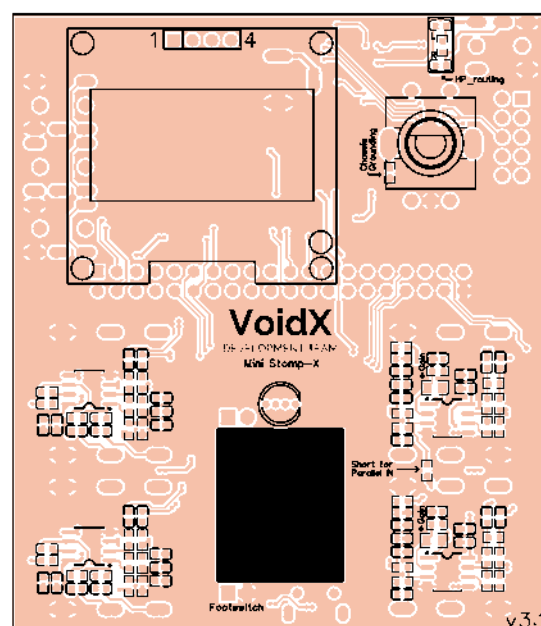
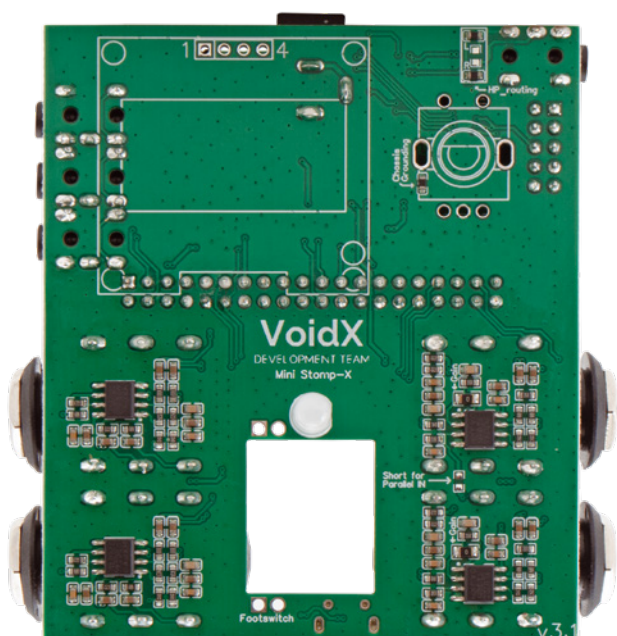
In questa sezione andremo ad analizzare Mini Stomp-X Dev-Board, scheda che, ospitando VoidX DSP Module, ci permetterà di realizzare il nostro equalizzatore stereo a 7 bande con controllo di volume in formato stomp-box per chitarra (**Fig. 3**).

Per questa board abbiamo anche realizzato una enclosure metallica.

I sorgenti hardware completi sono disponibili alla seguente repository:

<https://github.com/voidxdevteam/VoidX-Mini-Stomp-X>

Mini Stomp-X Dev-Board contiene la circuiteria analogica per l'interfacciamento ad ADC e DAC interni a VoidX DSP Module, espone connettori MIDI (IN e OUT/THRU), un'uscita cuffie e un ingresso per pedale d'espressione, tutto in formato jack TRS 3.5mm. La scheda può essere alimentata via USB o tramite il classico DC plug da 9V, tipico degli stomp-box. L'interfaccia utente è composta da un encoder per la navigazione, uno schermo OLED 128x64, un footswitch e un led RGB. Chiaramente è possibile controllare il pedale da remoto tramite l'app VoidX Control.



C47, C48, C49, C50, C51, C52: 1 μ F ceramico (0603)

C6, C55: 220 μ F 6,3 VL elettrolitico ($\varnothing$ 6,6mm)

D1, D2: B5819W (SOD-123)

OLED1: Display OLED 1,3"

LED1: LED RGB TJ-L5FYTXHMCSLCRGRB-A5

U1: Modulo DSP-PRO

U3: TPA6138A2PWR (TSSOP-14)

U2, U4, U5, U7: TL072IDR (SOIC-8)

U9: AP3012KTR-G1 (TSOT-23-5)

L1, L2, L3, L4, L6, L7: 600 ohm (0603)

L8, L9, L10, L11, L14: 600 ohm (0603)

L5, L12, L13: 10 μ H ($\varnothing$ 4mm)

J1, J2, J3, J4: Presa Jack 3,5 mm PJ313D

J5, J6, J7, J8: Presa Jack 6,3 mm PJ-609BA

USB1: Connettore micro-USB

P1: Pin stip Femmina 2x20 vie passo 2,54 mm

H2: Pin stip Femmina 2x5 vie passo 2,54 mm

DC2: Presa DC alimentazione DC-470-2.1GP

Varie:

- Circuito Stampato S1717 (68x78mm)

DESCRIZIONE DELL'APPLICAZIONE

L'applicativo presentato in questo articolo è un equalizzatore stereo a 7 bande con controllo di volume e curva di equalizzazione è comune ai due canali (Left e Right). Per ognuna delle 7 bande è presente un filtro di tipo PeakingEQ. Tali filtri sono caratterizzati da 3 parametri fondamentali (**Fig. 4**):

- Frequenza centrale: la frequenza attorno alla quale il filtro amplifica/attenua;
- Ampiezza: amplificazione (o attenuazione) in dB attorno alla frequenza centrale;
- Q (fattore di qualità): selettività del filtro.

Per l'applicazione in oggetto abbiamo selezionato un set di frequenze da 100Hz a 6.4kHz a distanza di ottave (100Hz, 200Hz, 400Hz, 800Hz, 1.6kHz, 3.2kHz, 6.4kHz).

Il Q è per tutti i filtri pari a 1.4, scelta classica quando si tratta di banchi octave-spaced (**Fig. 5**).

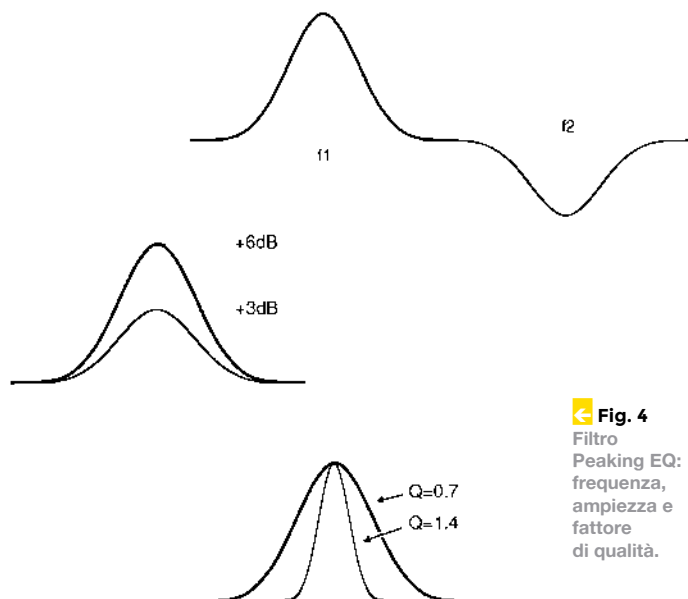
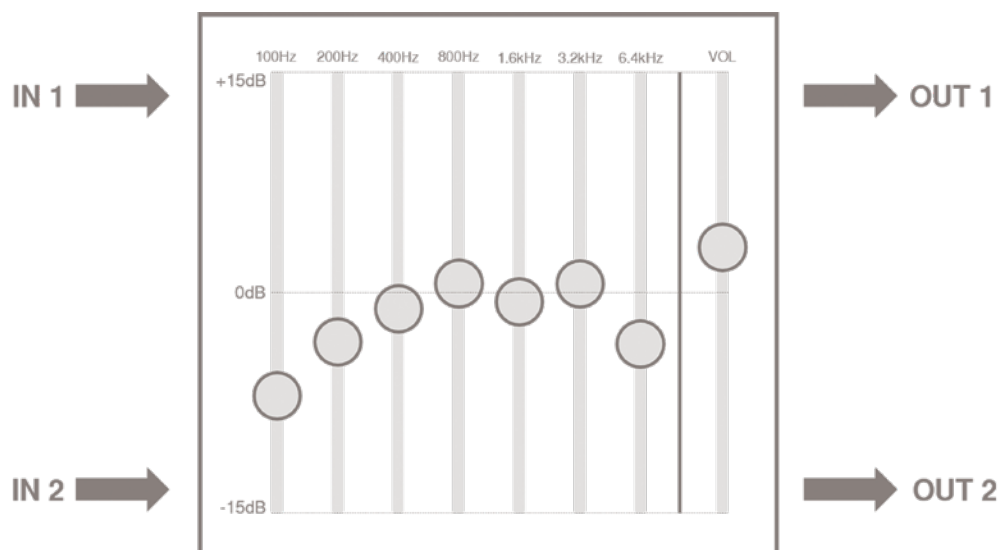


Fig. 4
Filtro
Peaking EQ:
frequenza,
ampiezza e
fattore di
qualità.

→ **Fig. 5**
Equalizzatore
stereo a sette
bande.



CODING DELL'APPLICAZIONE

Il framework utilizzato è ESP-IDF, ambiente integrato di sviluppo per la famiglia di microcontrollori ESP32 basato sul celebre sistema operativo real-time FreeRTOS. VoidX Library si configura come ESP-IDF Component, di fatto una libreria statica e un insieme di header istanziabili all'interno di un progetto. L'applicazione completa, sviluppata utilizzando come IDE Visual Studio Code, unita ad istruzioni dettagliate per il linking alla libreria è reperibile alla seguente repository Github:

<https://github.com/voidxdevteam/voidx-public-app>

Lo scheletro dell'applicazione è fornito dalla libreria VoidX Library, un framework che permette di sviluppare applicativi completi in maniera rapida

e intuitiva, fornendo un'interfaccia semplice ma completa e gestendo in modo trasparente vari aspetti critici di basso livello, uno tra tutti il threading audio real-time ad altissima priorità. Inoltre, VoidX Library fornisce numerosi driver hardware. Iniziamo a seguire il flusso del codice partendo dal main (**Listato 1**).

All'inizio vengono inizializzati i driver hardware: espansione I/O tramite shift register, I2S e I2C per comunicazione con i convertitori e periferica LCD. Si procede poi all'avvio del framework VoidX:

- `system_initialize()` avvia i task di sistema e crea i nodi fondamentali come il nodo root e i nodi che espongono nome e autore dell'app.
- `syst->adoptApplication(*)` associa al sistema l'applicazione, contenuta nell'oggetto `Eq7()`. `syst` rappresenta l'oggetto globale di sistema.

↓ Listato 1

```
void app_main(void)
{
    /* inizializzazione drivers */
    io_array_initialize();
    Codec::initialize();
    lcd_initialize();

    /* definizione della struttura dei nodi */
    system_initialize();
    syst->adoptApplication(Eq7());

    /* avvio driver BLE per comunicazione con VoidX Control */
    new Bluetooth(syst->pathToNode("root\\sys\\_name"));
}
```

▼ Listato 2

```
#define SAMPLING_CHANNELS    2
#define SAMPLING_FRAME      16

class Eq7: public AudioBlock, public Listener {
public:
    Eq7();
    void compile();
    void exec(float data[SAMPLING_CHANNELS][SAMPLING_FRAME]);
    bool draw();
    bool onEvent(int event);

private:
    /* Nodi esposti dall'applicazione (Parametri) */
    NodeFloat * band_level[7]; //livelli di EQ delle 7 bande
    NodeFloat * gain; //volume finale
    NodeEnum * mode; //modo (stereo o mono)
    NodeEnum * onoff; //on-off

    /* Istanziamento delle strutture relative ai 7x2 filtri PeakingEQ */
    BiquadExecData biquad_data_1_l;
    BiquadExecData biquad_data_2_l;
    BiquadExecData biquad_data_3_l;
    BiquadExecData biquad_data_4_l;
    BiquadExecData biquad_data_5_l;
    BiquadExecData biquad_data_6_l;
    BiquadExecData biquad_data_7_l;
    BiquadExecData biquad_data_1_r;
    BiquadExecData biquad_data_2_r;
    BiquadExecData biquad_data_3_r;
    BiquadExecData biquad_data_4_r;
    BiquadExecData biquad_data_5_r;
    BiquadExecData biquad_data_6_r;
    BiquadExecData biquad_data_7_r;

    /* volume finale */
    float gain_data;
};

/*Struttura di un filtro PeakingEQ */
typedef struct {
    float d_state[2]; //memoria del filtro
    float b_coeffs[3]; //coefficienti al denominatore
    float a_coeffs[3]; //coefficienti al numeratore
} BiquadExecData;
```

Una volta istanziato il framework, viene dato il via al task per la comunicazione Bluetooth con VoidX Control; come parametro viene fornito il nome da esporre come identificativo Bluetooth.

Lo sviluppatore non dovrà fare altro che lavorare all'interno di Eq7, classe madre dell'applicazione e figlia delle superclassi AudioBlock e Listener: dalla prima viene ereditata la gestione audio, dalla seconda la gestione degli eventi.

Nel **Listato 2** è descritto l'header Eq7.h.

Nel costruttore vengono creati i nodi relativi all'applicazione (**Listato 3**).

All'inizio vengono impostati i nodi relativi al nome e all'autore dell'applicazione, tali nodi sono reperiti utilizzando il loro path, noto a priori all'interno

dell'albero di sistema. Sotto il nodo app, figlio di root, vengono creati i nodi specifici dell'applicazione, ognuno dei quali ha un nome di sistema (ad esempio "on_off") e un display name (ad esempio "Enable").

I Nodi onoff e mode sono di tipo Enum, ovvero contenenti un numero finito di opzioni, in questo caso "ON"/"OFF" e "Mono"/"Stereo", default "OFF" e "Stereo".

I Nodi band_level[*] e gain sono di tipo Float, ovvero contenenti un numero reale, il loro range va da -15 a +15 con default 0, con unità di misura "dB", la curva del parametro è lineare.

Nelle ultime righe i nodi vengono aggiunti al motore di Storage: in questo modo il loro valore verrà

↓ Listato 3

```
Eq7::Eq7() : AudioBlock(){
    /* impostazione dei nodi di sistema Nome e Autore, la cui locazione è nota a priori */
    NodeItem * name = (NodeItem *)syst->pathToNode("root\\sys\\_name");
    if(name != NULL) name->setValue("Equalizer 7");
    NodeItem * author = (NodeItem *)syst->pathToNode("root\\sys\\_author");
    if(author != NULL) author->setValue("VoidX");

    /* creazione nodi specifici sotto "root/app" */
    NodeItem * parent = (NodeItem *)syst->pathToNode("root\\app");
    if(parent == 0) return;

    /* Nodo di ON/OFF */
    this->onoff = new NodeEnum(parent, "on_off", "Enable", {"ON", "OFF"}, "OFF");

    /* Nodi di equalizzazione */
    this->band_level[0] = new NodeFloat(parent, "band100", "100 Hz", -15.0f, 15.0f, 0, "dB", 0.5f, false);
    this->band_level[1] = new NodeFloat(parent, "band200", "200 Hz", -15.0f, 15.0f, 0, "dB", 0.5f, false);
    this->band_level[2] = new NodeFloat(parent, "band400", "400 Hz", -15.0f, 15.0f, 0, "dB", 0.5f, false);
    this->band_level[3] = new NodeFloat(parent, "band800", "800 Hz", -15.0f, 15.0f, 0, "dB", 0.5f, false);
    this->band_level[4] = new NodeFloat(parent, "band1600", "1600 Hz", -15.0f, 15.0f, 0, "dB", 0.5f, false);
    this->band_level[5] = new NodeFloat(parent, "band3200", "3200 Hz", -15.0f, 15.0f, 0, "dB", 0.5f, false);
    this->band_level[6] = new NodeFloat(parent, "band6400", "6400 Hz", -15.0f, 15.0f, 0, "dB", 0.5f, false);

    /* Nodo di volume */
    this->gain = new NodeFloat(parent, "gain", "Gain", -15.0f, 15.0f, 0, "dB", 0.5f, false);

    /* Nodo di "mode" (stereo o mono) */
    this->mode = new NodeEnum(parent, "mode", "Mode", {"Stereo", "Mono"}, "Stereo");

    /* Aggiunta dei nodi al motore di storage */
    Storage::addItem(this->onoff);
    Storage::addItem(this->band_level[0]);
    Storage::addItem(this->band_level[1]);
    Storage::addItem(this->band_level[2]);
    Storage::addItem(this->band_level[3]);
    Storage::addItem(this->band_level[4]);
    Storage::addItem(this->band_level[5]);
    Storage::addItem(this->band_level[6]);
    Storage::addItem(this->gain);
    Storage::addItem(this->mode);

    /* inizializzazioni di carattere generale */
    ...
}
```

mantenuto al riavvio. Si procede poi ad inizializzazioni di carattere generale che non è opportuno approfondire ora.

A seguito dell'inizializzazione la struttura dei nodi risulta essere come in **Fig. 6**.

L'albero ha come radice il nodo root, sotto sys sono presenti i nodi di sistema creati in system_initialize() e comuni ad ogni applicazione VoidX, mentre sotto app sono definiti i nodi specifici.

I nodi di tipo Float sono indicati in blu, quelli di tipo Enum in verde, in giallo quelli appartenenti ad altri tipi. Il framework permette di modellare l'informazione in modo molto flessibile; è ad esempio possibile dar vita ad un sistema di preset save and recall semplicemente istanziando opportuni nodi.

La rappresentazione dei nodi sotto sys non è esaustiva, sono presenti anche altri elementi di importanza per ora marginale (colore primario dell'applicazione, versione, device ID ecc.).

L'app VoidX Control esplora i nodi e crea un'interfaccia utente "semi-custom" su smartphone o PC. L'aspetto dell'applicazione su Smartphone Android è visibile in **Fig. 7**:

Procedendo con la classe Eq7, le funzioni ereditate sono:

- exec(*): ereditata da AudioBlock. La funzione viene eseguita 3000 volte al secondo con altissima priorità e ha lo scopo di processare 16 campioni alla volta alla frequenza di campionamento di 48kHz, e ciò garantisce elaborazio-

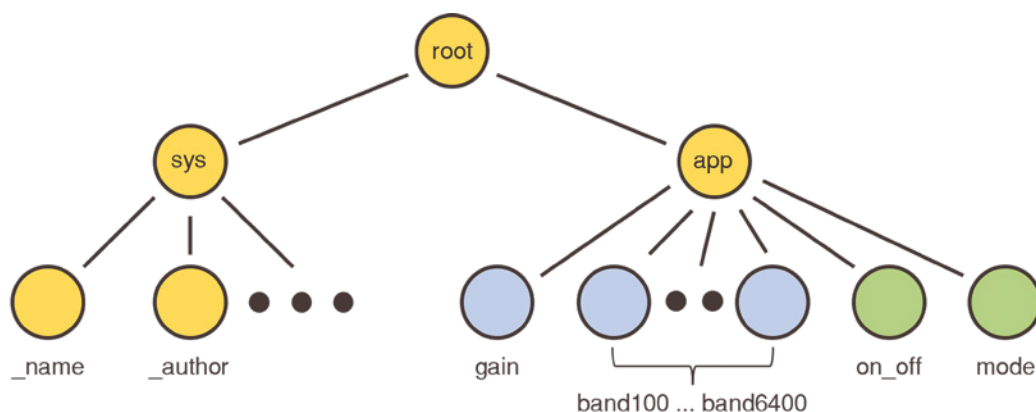


Fig. 6
Struttura
a nodi di
un'applica-
zione VoidX.

ne audio a bassissima latenza.

È all'interno di `exec(*)` che va eseguito il codice DSP in modo più efficiente possibile. Il parametro di ingresso punta allo stream I/O di campioni audio (array 2x16 di float).

- `compile()`: ereditata da `AudioBlock`. La funzione è eseguita ciclicamente con priorità medio-alta. Il suo scopo primario è quello di tradurre i parametri utente, ovvero i valori all'interno dei nodi, in coefficienti a livello DSP. Tale ruolo non può essere svolto direttamente da `exec(*)` per motivi di efficienza.
- `onEvent(*)`: ereditata da `Listener` ed eseguita in seguito al verificarsi di un evento, il quale è identificato da un numero intero. Lo scopo è quello di gestire i vari eventi ed aggiornare di conseguenza il modello interno. Non approfondiremo questo aspetto.
- `draw()`: ereditata da `Listener`. La funzione è eseguita ciclicamente con priorità minore rispetto a `compile()`; al suo interno va il codice relativo all'interfaccia utente. Non approfondiremo questo aspetto.

In un'ottica MVC (Model-View-Control), il Model è rappresentato dalla struttura dei nodi, la View è a carico di `draw()`, il Control a carico di `onEvent(*)` e `compile()`.

L'utente ha la facoltà di creare altri task (ad esempio per l'implementazione di nuovi driver) che andranno a comunicare con il framework VoidX tramite la logica degli eventi.

Nel **Listato 4** è descritta l'implementazione di `compile()`.

Le chiamate a `BiquadExecCompile(*)` vanno a ricavare i coefficienti DSP relativi ai filtri Peak-ingEQ partendo dalle loro caratteristiche di alto livello (frequenza centrale, ampiezza e Q). Le frequenze centrali (100Hz, 200Hz, ...6.4kHz) e i

fattori di qualità (1.4) sono noti a priori, le ampiezze sono dedotte dal valore dei nodi. L'ultima riga va a ricavare il valore lineare del guadagno, calcolato a partire dal valore contenuto nel nodo di volume, espresso in dB.

`BiquadExecCompile(*)` calcola i coefficienti dei filtri, operazione standard in ambito DSP (**Listato 5**).

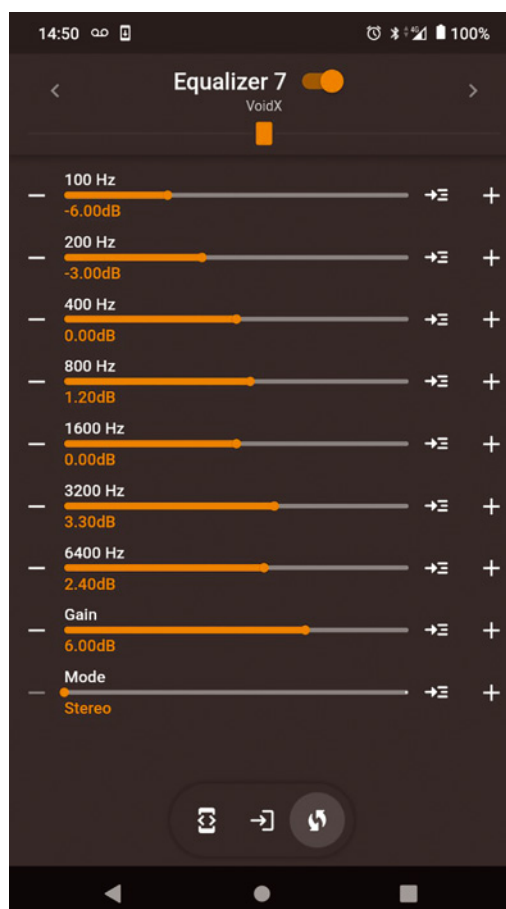


Fig. 7
Controllo
tramite
smartphone/
PC con
VoidX-
Control.

↓ Listato 4

```
void Eq7::compile(){
    //canale Left
    BiquadExec_compile(&this->biquad_data_1_l, this->band_level[0]->getValue(), 100.0f, 1.4f);
    BiquadExec_compile(&this->biquad_data_2_l, this->band_level[1]->getValue(), 200.0f, 1.4f);
    BiquadExec_compile(&this->biquad_data_3_l, this->band_level[2]->getValue(), 400.0f, 1.4f);
    BiquadExec_compile(&this->biquad_data_4_l, this->band_level[3]->getValue(), 800.0f, 1.4f);
    BiquadExec_compile(&this->biquad_data_5_l, this->band_level[4]->getValue(), 1600.0f, 1.4f);
    BiquadExec_compile(&this->biquad_data_6_l, this->band_level[5]->getValue(), 3200.0f, 1.4f);
    BiquadExec_compile(&this->biquad_data_7_l, this->band_level[6]->getValue(), 6400.0f, 1.4f);

    //canale Right
    BiquadExec_compile(&this->biquad_data_1_r, this->band_level[0]->getValue(), 100.0f, 1.4f);
    BiquadExec_compile(&this->biquad_data_2_r, this->band_level[1]->getValue(), 200.0f, 1.4f);
    BiquadExec_compile(&this->biquad_data_3_r, this->band_level[2]->getValue(), 400.0f, 1.4f);
    BiquadExec_compile(&this->biquad_data_4_r, this->band_level[3]->getValue(), 800.0f, 1.4f);
    BiquadExec_compile(&this->biquad_data_5_r, this->band_level[4]->getValue(), 1600.0f, 1.4f);
    BiquadExec_compile(&this->biquad_data_6_r, this->band_level[5]->getValue(), 3200.0f, 1.4f);
    BiquadExec_compile(&this->biquad_data_7_r, this->band_level[6]->getValue(), 6400.0f, 1.4f);

    //volume
    this->gain_data = powf(10,( this->gain->getValue() / 20.0f));
}
```

Nel **Listato 6** è visibile l'implementazione di `exec(*)`.

La condizione sul nodo `onoff` garantisce il bypass in caso di "OFF"; in caso di "ON", vengono eseguiti i filtri e l'amplificazione/attenuazione.

Alla fine se il parametro `mode` è impostato a "Mono" le uscite vengono rese uguali.

Le chiamate a `BiquadExec_exec(*)` eseguono il filtraggio, prendendo come argomento la memoria dei vari filtri e i loro coefficienti (**Listato 7**).

↓ Listato 5

```
#define SAMPLING_FREQ 48000.0f

void BiquadExec_compile(BiquadExecData * item, float value_db, float freq, float q){
    float parameter_geqf_a0;
    float parameter_geqf_a1;
    float parameter_geqf_a2;
    float parameter_geqf_b0;
    float parameter_geqf_b1;
    float parameter_geqf_b2;

    float a = sqrtf(powf(10,(value_db / 20.0f)));
    float w = 2 * M_PI * freq/SAMPLING_FREQ;
    float ws = sinf(w);
    float wc = cosf(w);
    float alpha = ws / (2.0f * q);

    parameter_geqf_b0 = 1+alpha*a;
    parameter_geqf_b1 = -2*wc;
    parameter_geqf_b2 = 1-alpha*a;
    parameter_geqf_a0 = 1+alpha/a;
    parameter_geqf_a1 = -2*wc;
    parameter_geqf_a2 = 1-alpha/a;

    item->a_coeffs[1] = parameter_geqf_a1 / parameter_geqf_a0;
    item->a_coeffs[2] = parameter_geqf_a2 / parameter_geqf_a0;
    item->b_coeffs[0] = parameter_geqf_b0 / parameter_geqf_a0;
    item->b_coeffs[1] = parameter_geqf_b1 / parameter_geqf_a0;
    item->b_coeffs[2] = parameter_geqf_b2 / parameter_geqf_a0;
    item->a_coeffs[0] = 1.0f;
}
```


Listato 6

```
void Eq7::exec(float data[SAMPLING_CHANNELS][SAMPLING_FRAME]){
    if(this->onoff->isValue("ON")){
        //eq L channel
        BiquadExec_exec(data[0], this->biquad_data_1_l.b_coeffs, this->biquad_data_1_l.a_coeffs, this->biquad_data_1_l.d_state, data[0]);
        BiquadExec_exec(data[0], this->biquad_data_2_l.b_coeffs, this->biquad_data_2_l.a_coeffs, this->biquad_data_2_l.d_state, data[0]);
        BiquadExec_exec(data[0], this->biquad_data_3_l.b_coeffs, this->biquad_data_3_l.a_coeffs, this->biquad_data_3_l.d_state, data[0]);
        BiquadExec_exec(data[0], this->biquad_data_4_l.b_coeffs, this->biquad_data_4_l.a_coeffs, this->biquad_data_4_l.d_state, data[0]);
        BiquadExec_exec(data[0], this->biquad_data_5_l.b_coeffs, this->biquad_data_5_l.a_coeffs, this->biquad_data_5_l.d_state, data[0]);
        BiquadExec_exec(data[0], this->biquad_data_6_l.b_coeffs, this->biquad_data_6_l.a_coeffs, this->biquad_data_6_l.d_state, data[0]);
        BiquadExec_exec(data[0], this->biquad_data_7_l.b_coeffs, this->biquad_data_7_l.a_coeffs, this->biquad_data_7_l.d_state, data[0]);
        //eq R channel
        BiquadExec_exec(data[1], this->biquad_data_1_r.b_coeffs, this->biquad_data_1_r.a_coeffs, this->biquad_data_1_r.d_state, data[1]);
        BiquadExec_exec(data[1], this->biquad_data_2_r.b_coeffs, this->biquad_data_2_r.a_coeffs, this->biquad_data_2_r.d_state, data[1]);
        BiquadExec_exec(data[1], this->biquad_data_3_r.b_coeffs, this->biquad_data_3_r.a_coeffs, this->biquad_data_3_r.d_state, data[1]);
        BiquadExec_exec(data[1], this->biquad_data_4_r.b_coeffs, this->biquad_data_4_r.a_coeffs, this->biquad_data_4_r.d_state, data[1]);
        BiquadExec_exec(data[1], this->biquad_data_5_r.b_coeffs, this->biquad_data_5_r.a_coeffs, this->biquad_data_5_r.d_state, data[1]);
        BiquadExec_exec(data[1], this->biquad_data_6_r.b_coeffs, this->biquad_data_6_r.a_coeffs, this->biquad_data_6_r.d_state, data[1]);
        BiquadExec_exec(data[1], this->biquad_data_7_r.b_coeffs, this->biquad_data_7_r.a_coeffs, this->biquad_data_7_r.d_state, data[1]);
        //gain
        for(int i=0; i<SAMPLING_FRAME; i++){
            data[0][i] *= -this->gain_data;
            data[1][i] *= -this->gain_data;
        }
    }
    //mode
    if(this->mode->isValue("Mono")){
        for(int i=0; i<SAMPLING_FRAME; i++){
            data[1][i] = data[0][i];
        }
    }
}
```

Listato 7

```
void BiquadExec_exec(float * x, float *b, float *a, float * delay, float * y){
    for (int i = 0 ; i < SAMPLING_FRAME ; i++) {
        float d0 = x[i] - a[1] * delay[0] - a[2] * delay[1];
        y[i] = b[0] * d0 + b[1] * delay[0] + b[2] * delay[1];
        delay[1] = delay[0];
        delay[0] = d0;
    }
}
```

CONCLUSIONE E SVILUPPI FUTURI

Lo scopo di questo articolo è stato quello di presentarvi l'ecosistema VoidX, dando un'idea dei vantaggi che il lettore può trarre da quest'ultimo, sia in termini di piattaforme hardware che in termini di sviluppo software. L'applicativo "Equalizzatore" è stato scelto per rendere lo scopo del progetto comprensibile a tutti pur sapendo che, purtroppo, il mondo DSP non è di immediata comprensione ai non addetti ai lavori.

In futuri articoli andremo maggiormente nel dettaglio, addentrandoci nell'hardware e nella struttura dei nodi, nei vari tipi di parametri, e nello standard di comunicazione.



COSA OCCORRE ?

Codice	Descrizione	Prezzo € IVA inclusa
VOIDXDSP	Modulo con ESP32 per applicazioni audio	44,00
DMSTOMPX	Demo board per il modulo VOIDXDSP	39,00

IL MATERIALE VA RICHIESTO A:

Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 - www.futurashop.it



Makers LAB e dintorni!

Starter Kit con Micro:bit V2 e accessori

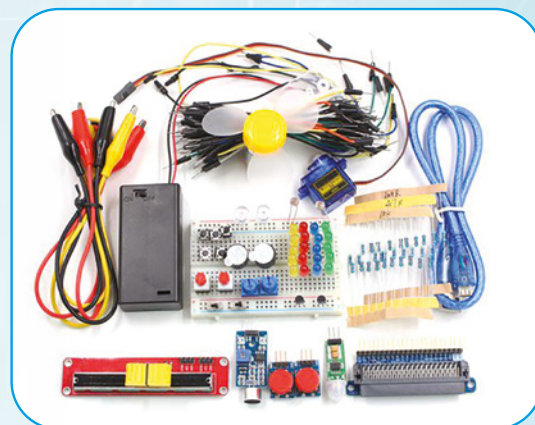
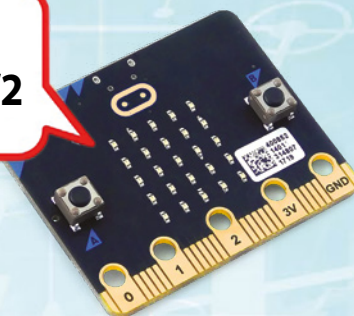
micro:bit V2 è una piccola scheda di sviluppo che integra un microcomputer a 32 bit; può essere utilizzata per imparare le basi della programmazione ed è adatta a tutte le età. Utilizzala subito con lo starter kit!



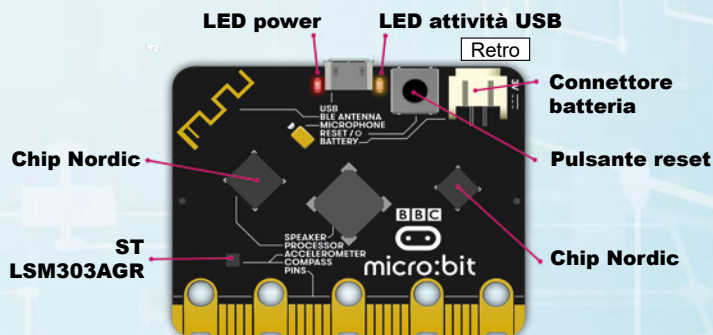
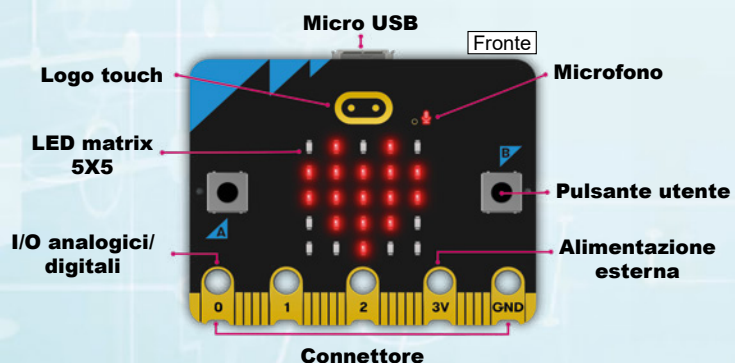
Il kit contiene
micro:bit versione V2

cod. MICROBITKITVER2

€ 79,00



La scheda micro:bit V2 nel dettaglio



STABILIZZATORE DI LIVELLO LIQUIDI

di DAVIDE SCULLINO

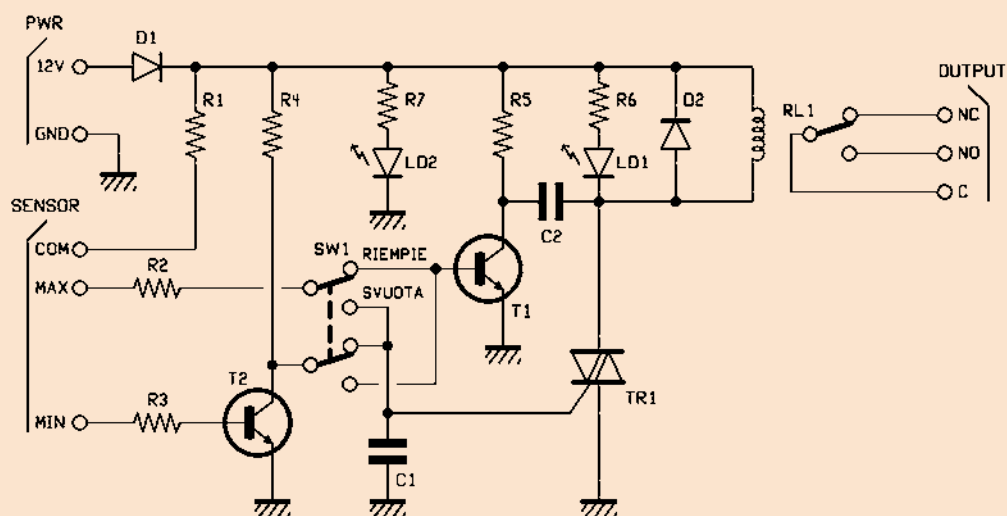


Attraverso i contatti di un relé e di un doppio deviatore consente di mantenere il livello del liquido contenuto in una vasca o serbatoio al valore desiderato, intervenendo su una pompa o elettrovalvola.

Q

uando si immagazzina del liquido in una vasca ed ancor più in un serbatoio chiuso, è determinante conoscerne il livello, ovvero la quantità residua, sia per evitare di rimanere a secco, sia per prevenire la tracimazione. Di solito allo scopo ci si limita al monitoraggio attraverso gli innumerevoli indicatori di livello liquidi

dei quali anche noi abbiamo proposto in questi anni varie declinazioni, salvo che non sia disponibile una finestrella spia (view-stripe) o non risulti semplice l'ispezione visiva diretta. Un operatore, ossia una persona, provvede poi ad agire di conseguenza, eventualmente aprendo un rubinetto per ripristinare il livello o un tappo per abbassarlo. In queste pagine abbiamo voluto proporre qualcosa che va oltre, vale a dire che



non solo monitora il livello, ma può automaticamente agire su di esso intervenendo, attraverso lo scambio di un relé a bordo, su elettropompe ed elettrovalvole in grado, opportunamente installate, di incrementare il livello oppure di ridurlo così da farlo rientrare nella finestra desiderata. Più esattamente, il circuito qui proposto permette di mantenere il liquido di una cisterna/vasca tra un livello minimo e uno massimo, offrendo la possibilità di portare a livello minimo al raggiungimento del livello massimo (funzione svuotamento) e viceversa di portarla al livello massimo al raggiungimento del livello minimo (funzione riempimento). Per scegliere una di queste due funzioni, sulla scheda è stato previsto un doppio deviatore che permette di attivare il relé al superamento del livello rilevato, oppure, al contrario, quando lo stesso scende sotto la soglia desiderata. Comprenderemo meglio ciò ed anche in che modo è stata implementata la relativa funzionalità, analizzando lo schema elettrico del circuito, proposto in alto in questa pagina; il tutto, con la premessa che il progetto trova applicazione nel controllo del livello di liquidi elettricamente conduttivi e non infiammabili.

SCHEMA ELETTRICO

Il circuito di controllo è, come vedete, abbastanza semplice, peraltro formato da soli componenti discreti; per agevolarne l'analisi scomponiamo lo schema elettrico in blocchi, identificando lo stadio di ingresso per il rilevamento del livello attuale della vasca o serbatoio, facente capo alle resisten-

ze R2-R3, nonché al transistor NPN T2, al doppio deviatore SW1 e al transistor T1 (un secondo NPN) poi il relé di uscita e l'SCR (Silicon Controlled Rectifier) TR1 utilizzato per comandare il relé di uscita RL1. L'SCR ci serve perché consente di semplificare l'attuatore del relé, giacché permette di mantenere una condizione di conduzione stabile anche dopo essere stato triggerato con un semplice impulso e quindi sostituisce circuitazioni più complesse come il bistabile (che nella versione più semplice richiede due transistor BJT) o il flip-flop.

Il doppio deviatore SW1 è l'elemento che dice al circuito come deve considerare la lettura del livello della vasca o serbatoio: se impostato su "SVUOTA" permette di svuotare la vasca appena sotto il livello minimo se il liquido ha raggiunto il livello massimo, oppure se impostato su "RIEMPIE" di riempire la vasca appena il liquido scende sotto il livello minimo.

Ma vediamo di spiegare nel dettaglio il funzionamento di questo circuito, con la premessa che all'interno del serbatoio o vasca del caso devono essere posizionati due elettrodi alla distanza desiderata, che dovrà corrispondere alla differenza tra il minimo (contatto MIN del circuito) e il massimo (elettrodo MAX del circuito) livello attesi; non si possono collocare gli elettrodi alla stessa altezza perché in tal caso il circuito non funzionerebbe, ovvero continuerebbe a commutare attivando impulsivamente e inutilmente l'elettrovalvola o elettropompa.

In contatto con il liquido dovrà trovarsi il contatto comune (COM) che nel circuito viene alimentato

attraverso la resistenza R1 (questa serve ad evitare danni in caso di cortocircuito accidentale nei cablaggi) dal positivo di alimentazione del circuito a valle del diodo di protezione dall'inversione di polarità, in modo che la presenza di liquido che lambisce almeno l'elettrodo COM e uno tra MIN e MAX determini una resistenza elettrica e il conseguente flusso di corrente dalla linea positiva alla base dei transistor. L'elettrodo COM dovrà essere posizionato all'interno della vasca o serbatoio più in basso dell'elettrodo del minimo (MIN) altrimenti il circuito non funzionerà correttamente.

Quando il livello del liquido è minore del minimo il circuito è in stato di quiete e bisogna riempire la vasca o serbatoio abbastanza da raggiungere il livello minimo, ovvero fare in modo che sia COM che MIN siano toccati dal liquido; questo può essere ottenuto spostando il doppio deviatore SW1 in posizione RIEMPIE. Se è posizionato su SVUOTA, il circuito rimane inattivo.

Analizziamo dunque il funzionamento nelle due distinte condizioni, iniziando con SW1 su RIEMPIE; se il livello del liquido è sotto il minimo, l'NPN T2 è interdetto e il suo collettore viene mantenuto a livello alto (il potenziale di alimentazione del circuito) dal resistore R4, cosicché il gate del TRIAC riceve corrente e il diodo controllato conduce tra anodo (A) e catodo (K) determinando la polarizzazione del LED rosso (attraverso la relativa resistenza di caduta R6) e l'alimentazione della bobina del relé, cosicché lo scambio di RL1 si chiude tra C ed NO. Se tra questi due contatti è collegato il filo di alimentazione di un'elettropompa che spinge o un'elettrovalvola (se trattasi di circuito collegato a una tubazione di rifornimento sotto pressione) il contenitore inizia a riempirsi.

Quando il livello del liquido raggiunge il contatto MIN, la resistenza R3 permette di polarizzare e mandare in conduzione tra collettore ed emettitore il transistor T2, il cui collettore si porta a un potenziale che non è comunque sufficiente a spegnere l'SCR, il quale rimane innescato; man mano che il livello del liquido cresce, arriva a toccare il contatto MAX, allorché la corrente che fluisce in R2 va, tramite il deviatore SW1, a polarizzare la base del T1, il cui collettore si porta a un potenziale di circa 0,5V e tale condizione scarica quasi istantaneamente il condensatore C2, che fino a prima era carico con polarità positiva sul collettore stesso ad opera della resistenza R5.

La rapida scarica di tale condensatore determina un impulso di tensione (e di corrente) negativo sull'anodo dell'SCR, che è sufficiente a bloccarne

la conduzione, giacché il gate del RT1 non è più polarizzato sufficientemente da quando il liquido aveva lambito l'elettrodo MIN. Il C2 si carica quindi con polarità opposta finché l'NPN T1 non torna interdetto.

La condizione suddetta fa ricadere lo scambio del relé e spegnere il LED rosso LD1. In questa fase il diodo D2, seppure potrebbe non essere strettamente necessario, taglia l'extratensione inversa che la bobina del relé genera quando viene bruscamente privata dell'alimentazione, ostacolando l'eventuale re-innesco dell'SCR.

Bene, va da sé che quando il livello scende sotto al massimo ma rimane sopra il minimo, nulla cambia nello stato del TR1, che rimane interdetto; tornerà in stato di ON quando il livello del liquido nella cisterna o serbatoio tornerà sotto al minimo, allorché il transistor T2 non potrà più essere mantenuto in conduzione, si interdirà e il suo collettore assumerà un potenziale pari a quello della linea positiva di alimentazione, il che determinerà il nuovo innesco del gate dell'SCR e la conduzione di quest'ultimo, che sarà tale da riattivare il relé e con esso la pompa elettrica o l'elettrovalvola alimentata dallo scambio del RL1.

Quindi, riassumendo, quando SW1 è spostato in posizione RIEMPIE il circuito aziona un'elettrovalvola o un'elettropompa mediante lo scambio del relé quando il liquido scende sotto il livello minimo, per spegnerla poi quando supera il livello massimo e viceversa; tra minimo e massimo in riempimento e tra massimo e minimo durante lo svuotamento, il relé non muta il proprio stato e il circuito risulta praticamente inerte. La distanza tra gli elettrodi MAX e MIN determina l'isteresi del funzionamento del relé.

CARATTERISTICHE TECNICHE

Tensione di alimentazione:
12 Vcc

Corrente assorbita:
60 mA

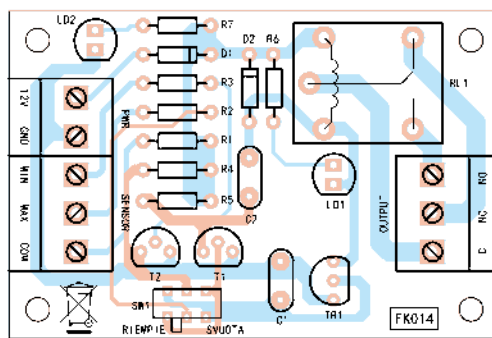
Uscita a relé:
C-NC-NO

Tensione commutabile in uscita:
max 230Vca

Corrente commutabile in uscita:
10A

Funzionamento:
in riempimento e svuotamento





Elenco Componenti:

R1, R2, R3: 100 kohm

R4: 33 kohm

R5: 10 kohm

R6, R7: 1 kohm

C1: 10 nF 63 VL poliestere

C2: 470 nF 63 VL poliestere

D1, D2: 1N4007

T1, T2: BC547

TR1: P0102DA 2AL3

LD1: LED 5 mm rosso

LD2: LED 5 mm verde

RL1: Relè 12V SRD-12VDC-SL-C

SW1: Deviatore slitta DPST

Varie:

- Morsetto 2 vie passo 5.08 mm
- Morsetto 3 vie passo 5.08 mm (2 pz.)
- Circuito stampato FK014 (65x44 mm)

LE MODALITÀ OPERATIVE

Andiamo ora a spiegare come funziona il tutto se il doppio deviatore SW1 è in posizione SVUOTA: in questa modalità operativa lo scopo non è riempire il serbatoio o la cisterna fino al massimo e aspettare che ritorni al minimo, ma fare il contrario, ossia operare in maniera tale da tenere il livello del liquido al minimo. Vediamo come ciò avviene, ipotizzando che il livello di partenza sia tale da toccare l'elettrodo MAX; come già visto, fluisce corrente attraverso la resistenza R2, solo che questa volta attraverso i contatti del doppio deviatore va a polarizzare il gate dell'SCR, causando l'innesco di tale componente e l'eccitazione del relé, con la conseguenza di attivare l'elettrovalvola di svuotamento o la pompa aspirante. Quando il livello si abbassa fino a rimanere comunque tra MIN e MAX, il contatto MIN porta corrente alla base del T2, che pertanto rimane in conduzione interdicendo T1, giacché il suo collettore ha un potenziale non sufficiente a mantenere la giunzione base-emettitore del T1 in conduzione. Pertanto in questa condizione il transistor T1 è irrilevante ai fini dello stato di funzionamento dell'SCR.

Se il liquido scende ancora di livello fino a non toccare più né il contatto MAX né tantomeno quello MIN, nel momento in cui ciò avviene il transistor

T2 commuta in interdizione, il suo collettore viene portato a livello alto dalla resistenza R4, cosicché la base del T1 viene polarizzata e quest'ultimo passa dall'interdizione alla saturazione, trascinando a livello basso l'armatura destra del condensatore C2, il quale determina il già descritto impulso negativo sull'anodo dell'SCR, causando l'interdizione di quest'ultimo come già visto. In conseguenza di ciò, il relé RL1 torna a riposo e il suo scambio C-NO si apre, sospendendo l'alimentazione dell'elettropompa o elettrovalvola di svuotamento. Quando il serbatoio o cisterna si riempirà nuovamente e il liquido tornerà a lambire il contatto MIN, T2 riprenderà a condurre e porterà a circa 0,5 volt la base del T1, il quale ora andrà in interdizione lasciando nuovamente caricare il condensatore C2, ma ciò non avrà alcun effetto sul circuito; non appena il liquido toccherà anche il contatto MAX, la corrente che attraverserà R2 andrà a triggerare il gate dell'SCR, mandando nuovamente in stato di ON questo diodo controllato e facendo nuovamente eccitare il relé, il cui scambio alimenterà una volta ancora l'elettropompa o elettrovalvola di svuotamento.

Bene, questo è quanto riguarda il funzionamento del nostro controllo elettronico; andiamo ora ad analizzare i componenti trascurati nell'analisi con-

dotta sinora a iniziare dal condensatore C1, la cui funzione è cortocircuitare gate e catodo dell'SCR nel momento in cui viene applicata l'alimentazione al circuito, per evitare che il tiristore venga innescato accidentalmente, specie se all'accensione con SW1 in posizione RIEMPIE i contatti MIN e MAX sono entrambi a secco o se con SW1 in posizione SVUOTA i predetti sono immersi nel liquido. Il diodo D1 (situato in serie alla linea positiva dell'alimentazione) serve a proteggere il circuito nel caso in cui l'alimentazione, che va applicata tra i contatti PWR (12V e GND), venga fornita per errore con la polarità invertita; infine, il LED LD2 (verde) indica che la scheda è alimentata, giacché viene polarizzato attraverso la resistenza R7 dalla linea positiva a valle del diodo D1.

La tensione prevista per alimentare il nostro circuito è di 12V in continua, mentre l'assorbimento massimo (con relé eccitato) si attesta intorno ai 60 milliampere.

REALIZZAZIONE PRATICA

Bene, dopo un'accurata descrizione dello schema siamo giunti agli aspetti costruttivi: il circuito qui proposto si realizza su un circuito stampato da noi progettato, che potete preparare utilizzando il servizio PCBPRODUCTION di Futura Elettronica,

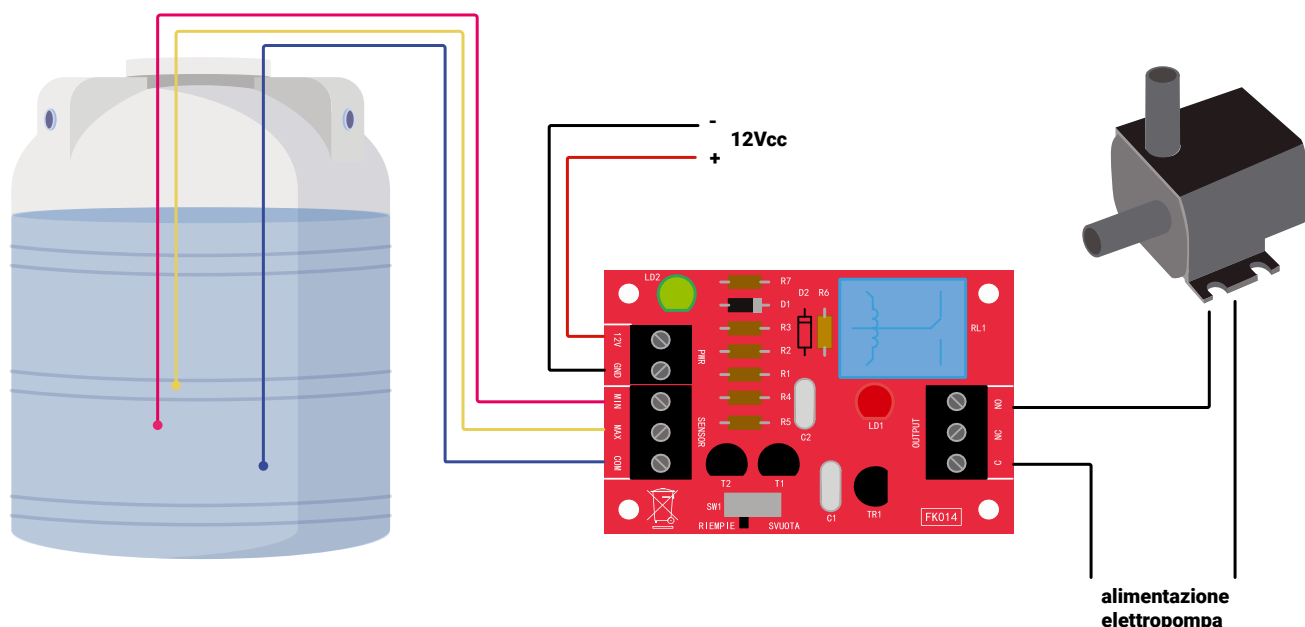
utilizzando il gerber scaricabile dalle pagine di presentazione di questo numero della rivista. Una volta ricevuto il PCB, potete montarvi i pochi componenti occorrenti a partire dalle resistenze e dai diodi al silicio, per poi passare al doppio deviatore miniatura a slitta, ai condensatori e ai transistor e all'SCR, tutti da orientare come indicato nel piano di montaggio che trovate illustrato in queste pagine.

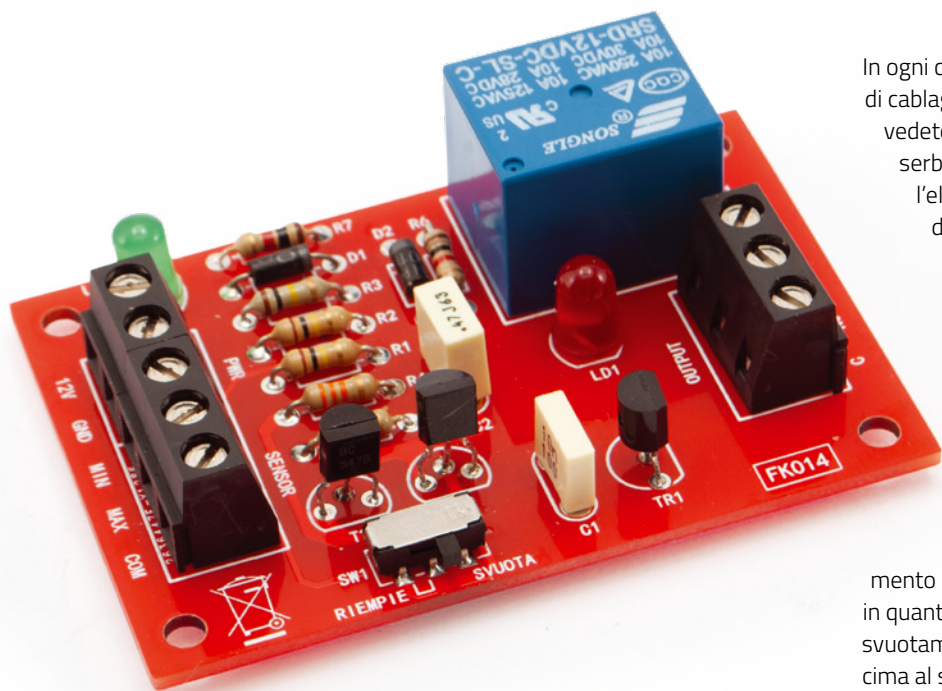
Quindi inserite e saldate i due LED, ricordando che il terminale di catodo è quello che si trova dalla parte dello smusso, mantenendoli all'altezza che desiderate; procedete saldando le morsettiere a passo 5 mm, una bipolare e una tripolare dal lato dell'alimentazione ed una tripolare dalla parte del relé, componente che monterete per ultimo e che entrerà nel circuito stampato nel verso giusto perché ha un solo verso d'inserimento.

Controllato il montaggio e verificatane l'esattezza, nonché l'assenza di baffi di stagno che potrebbero unire accidentalmente piste adiacenti, il circuito è pronto per l'utilizzo; procuratevi un contenitore in plastica dove inserirlo, quindi collegate i morsetti MAX, MIN e COM con dei cavi unipolari in guaina ai rispettivi elettrodi che potrete posizionare nel modo che meglio credete, all'interno della vasca o del serbatoio da monitorare.

Fig. 1
Schematizzazione dei collegamenti e della disposizione degli elettrodi sensori di livello.

| schema di CABLAGGIO





 Il prototipo del circuito montato.

Ricordiamo che l'elettrodo COM dovrà trovarsi più in basso di MAX e MIN e che quest'ultimo dovrà essere al disotto dell'elettrodo MAX.

Per gli elettrodi potete utilizzare delle barrette di metallo, magari acciaio inossidabile, ma anche dei semplici bulloni da fissare all'interno del serbatoio, sulla parete; quelli di MIN e MAX devono essere isolati dal punto di partenza del cavo che porta alla scheda, altrimenti la loro collocazione falserà la lettura. L'ideale sarebbe introdurre gli elettrodi da un lato (quello comune può tranquillamente essere deposto sul fondo) ma ciò implicherebbe di dover adottare un morsetto in gomma o altro tipo di guarnizione per evitare fuoriuscite di liquido dal punto di introduzione degli elettrodi stessi. Fate un po' voi, come vi viene più facile e in base alla condizione in cui vi troverete a operare e al materiale di cui disponete.

In ogni caso potete prendere spunto dallo schema di cablaggio proposto nella **Fig. 1**, nella quale vedete come vanno posizionati gli elettrodi nel serbatoio o vasca e in che modo collegare l'elettropompa o elettrovalvola del caso; va da sé che se impostate il circuito affinché funzioni in riempimento (switch su RIEMPIE) l'elettrovalvola o pompa dovranno essere poste in alto rispetto al serbatoio, mentre se volete che operi in svuotamento (SW1 in posizione SVUOTA) l'elettrovalvola o pompa dovranno stare idealmente in basso. In realtà questi posizionamenti vanno rispettati sostanzialmente se riempi-

mento e svuotamento li affidate ad elettrovalvole, in quanto adottando delle elettropompe per lo svuotamento si potrebbero collocare anche in cima al serbatoio o vasca. È da evitare, per una questione di sforzo e di prevalenza, il riempire dal basso il serbatoio o vasca con una pompa, perché dovrebbe vincere il peso della colonna di liquido. Comunque sia, una volta realizzato il cablaggio potete pensare all'alimentazione del circuito, che va fornita in continua con un alimentatore in grado di erogare 12V stabilizzati e una corrente di 70-80 mA per stare un po' "larghi" e sostenere eventuali transienti di assorbimento. Resta inteso che avendo a che fare con liquidi, è opportuno tenere il circuito al riparo, magari in un contenitore a tenuta stagna tipo quelli da esterno per impianti elettrici, ovvero posizionare lo stampato lontano dal liquido, ben riparato (lo stesso vale per la fonte di alimentazione).

Fate comunque in modo da rendere visibile dall'esterno almeno il LED rosso, che indica quando il circuito sta intervenendo e il relé sta comandando la pompa o elettrovalvola; se è il caso, portate i LED all'esterno della basetta collegandoli con sottili spezzoni di filo in rame con guaina, per esempio da 2x0,3 mmq e applicandoli ad un pannello o al contenitore che avete scelto.

Poter avere una segnalazione visiva dello stato del relé vi consente di capire se il sistema di riempimento o svuotamento sta funzionando o meno e di diagnosticare eventuali problemi laddove il funzionamento non sia quello atteso.

Bene, con questo abbiamo terminato la descrizione di un circuito al tempo stesso semplice e molto utile, che risolve con pochissimi componenti discreti situazioni reali senza ricorrere a microcontrollori né programmazione. 



COSA OCCORRE ?

Codice	Descrizione	Prezzo € IVA inclusa
FK014	Controllo liquido serbatoio	13,50
ER12W12V	Alimentatore 12V 1A	7,50

IL MATERIALE VA RICHIESTO A:

Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 - www.futurashop.it

Non farti sfuggire nulla!

Con le mini telecamere riprendi tutto,
in ogni momento
e in ogni situazione.

Action Camera Wi-Fi 4K

Action camera a colori con sensore da 5 Megapixel in grado di effettuare riprese da 4k visualizzabili sul display a colori da 2". Dispone di connessione Wi-Fi e USB, uscita video micro HDMI per visualizzare direttamente le immagini e i video registrati su un televisore, obiettivo grandangolare 140°, microfono integrato ad alta sensibilità e slot per micro SD card. Viene fornita con batteria ricaricabile da 1050 mAh, custodia a tenuta stagna (IPX68) e varie staffe di fissaggio. Tramite l'App "DENVER ACK-8062W" scaricabile per Android o iOS, e attivando il Wi-Fi dell'Action camera, è possibile, tramite Wi-Fi, vedere sul proprio smartphone le immagini riprese in diretta. I video real time e le immagini vengono salvate su una memoria micro SD card HC (max. 64 GB - non inclusa) da inserire nell'apposito slot.



Full HD
1080

cod. ACK8062W

€ 69,00

CON CONTENITORE
ERMETICO
PER RIPRESE
SUBACQUEE

Penna con telecamera full HD

Penna dal design elegante di colore nero-alluminio dotata di telecamera ad alta risoluzione in grado di effettuare riprese audio video o scattare fotografie di alta qualità. Registrazione tramite micro SD Card (Acquistabile separatamente).

Full HD
1080

cod. FR830

€ 38,00



Telecamera a bottone con telecomando

Telecamera CMOS a colori da 2 Megapixel con batteria ricaricabile integrata. Indossabile e di piccole dimensioni è facilmente occultabile grazie all'obiettivo a forma di bottone. Utilizzabile anche come webcam, dispone di slot per micro-SD card, interruttore on/off e radiocomando che permette di avviare o fermare la registrazione video, senza che nessuno se ne accorga. È in grado di effettuare registrazioni video (AVI) con risoluzione 1920x1080 pixel e salvarle direttamente su micro-SD card max. 128 GB (acquistabile separatamente).



cod. FR825

€ 80,00

Full HD
1080



cod. CP756

€ 19,90

Full HD
1080

Mini telecamera action sport

A colori con audio con angolo di ripresa di 140°, 5 LED IR per la visione notturna, funzione Motion Detection, attacco per cinghia e staffa di supporto ruotabile di 360°. Può essere utilizzata anche come Webcam.



SOLO
23x23
x23
mm

Mini telecamera Action sport Full HD

A colori con audio. Può essere utilizzata per riprese sott'acqua, grazie all'involucro ermetico trasparente fornito in dotazione. La telecamera integra un magnete che consente di fissarla facilmente a qualsiasi superficie metallica e dispone di staffa di fissaggio ruotabile di 360°.

SOLO
26x26
x26
mm



cod. FR834

€ 33,00

Full HD
1080

CON CONTENITORE
ERMETICO PER RIPRESE
SUBACQUEE



Saldare diventa più facile!

Supporto per saldatura con 6 bracci flessibili e accessori



cod. **HR2120**
€ 39,00

Composta da lampada a 16 LED con lente d'ingrandimento 3X, terza mano, braccio flessibile da 25 cm con clip a coccodrillo, porta saldatore e porta rocchetto di stagno. Viene fornita completa di pasta per la saldatura (10g) e di spugnetta per rimuovere lo stagno dalla punta del saldatore.

Supporto per saldatura composto da 6 bracci flessibili con clip a coccodrillo, lente d'ingrandimento 3x, ventola da 30X30 mm per assorbire il fumo della saldatura con alimentazione USB.

Postazione multifunzione



cod. **TIP019**
€ 48,00

Supporto per saldatura a 4 bracci flessibili con clip a coccodrillo



cod. **SHH4ARMS**
€ 23,90

Ideale per saldare e dissaldare componenti elettronici o per eseguire piccoli lavori di precisione. Con spugnetta e un pulisci punta per saldatore con paglietta in ottone.

Terza mano con lente, luce e portasaldatore



cod. **TP232**
€ 17,00

Terza mano con luce LED, lente d'ingrandimento 3,5X e supporto per saldatore

Terza mano con lente d'ingrandimento



cod. **VTHH**
€ 7,90

Terza mano con lente d'ingrandimento 2,5X e supporto per saldatore

Supporto per serraggio scheda

Permette di sorreggere e bloccare, tramite due morsetti che possono ruotare di 360°, una scheda elettronica con dimensioni max 20X14 cm.



cod. **VTHH6**
€ 12,00

Prezzi IVA inclusa

AUDIO PRO: UN IMPIANTO AUDIO COMPLETO



del Prof. **MICHELE MENNITI**

Assembliamo il nostro impianto audio completo per regolarlo e verificarne il funzionamento. Terza e ultima puntata.

Nelle prime due puntate abbiamo illustrato tutti gli schemi elettrici dei circuiti che compongono il nostro Audio PRO stereo e visto il montaggio dei numerosi PCB necessari. Vedremo ora come assemblare tutti i componenti all'interno di quattro contenitori, come realizzare i collegamenti tra i vari elementi, eseguire le tarature e infine condurre il test finale per garantire il corretto funzionamento dell'impianto audio.

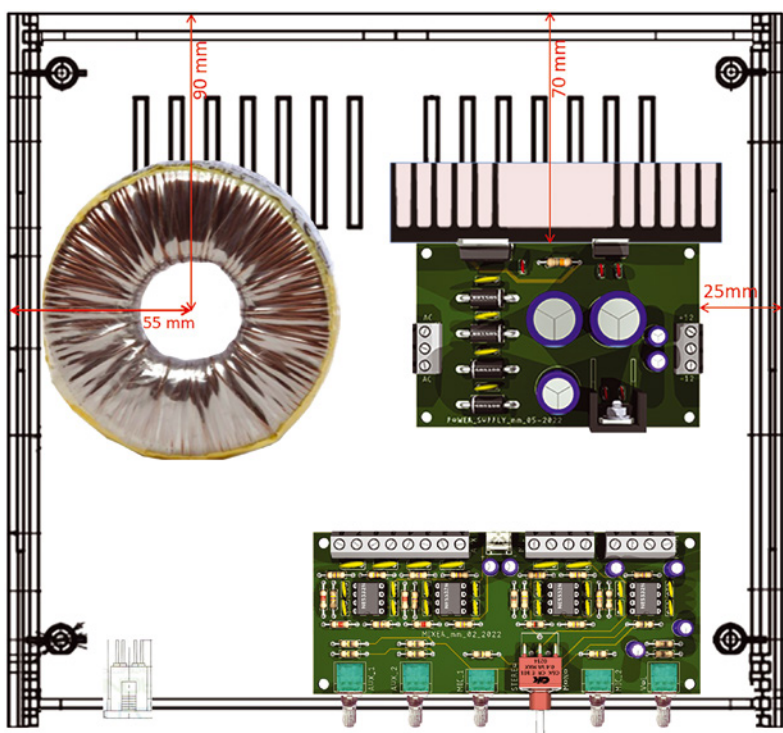
Fig. 1
Vista frontale dei
quattro contenitori
dell'Audio PRO.



MONTAGGIO NEI CONTENITORI

Per dare un aspetto esteticamente gradevole al nostro impianto Audio PRO, abbiamo montato tutti i circuiti in quattro contenitori in ABS: due per equalizzatore e visualizzatore di spettro (cod. KZ15) e due per alimentatore generale e amplificatore finale (cod. KZ17W). Poiché è stato necessario forare sia i pannelli frontali che quelli posteriori di ogni contenitore, e considerando che non è stato un lavoro semplice, abbiamo pensato di approntare dei file STL (scaricabili dalla sezione download), che permetteranno di ricrearli in 3D (forati con la serigrafia in rilievo) o, meglio ancora, prepararli mediante incisione laser (scelta da noi preferita come visibile in **Fig. 1**).

Fig. 2
Quote per il
montaggio di PCB
e trasformatore
toroidale di mixer
e alimentatore
generale.



Risolta la parte più complessa, con la preparazione dei pannelli, il lavoro residuo consiste nel forare la base dei contenitori, per il fissaggio dei PCB mediante dei distanziali in plastica; è un lavoro relativamente semplice ma, in alcuni casi, richiede estrema precisione ed alcuni accessori extra che potranno essere preparati, sempre mediante stampa in 3D.

Naturalmente nulla vieta che si possa decidere di realizzare 4 contenitori in legno, in perfetto stile retrò, oppure usarne altri completamente differenti, provvedendo poi alla preparazione artigianale degli otto pannelli.

Cercheremo ora di dare il maggior numero di informazioni possibili, per facilitare il compito della lavorazione dei contenitori.

Alcune informazioni riguardo i collegamenti filari: per l'alimentazione va bene un normale cavetto guainato a tre fili, di colore diverso, per evitare errori di inversione che risulterebbero disastrosi. Per i collegamenti audio serve necessariamente del cavo schermato specifico, del tipo singolo (centrale più calza) oppure doppio (due poli più due calze) che ha il vantaggio di avere i poli di colore diverso (tipicamente bianco e rosso), per attribuirli univocamente ai due canali destro e sinistro. Infine per i collegamenti delle spie luminose, dei LED di clipping e dei vari deviatori si può ricorrere a del cavo guainato sottile.

NOTA BENE: entrambi i tipi di contenitori sono costituiti da due gusci (superiore ed inferiore) apparentemente identici e, naturalmente dai due pannelli già descritti. In realtà la base, rispetto al coperchio, presenta i quattro fori per le viti di chiusura dell'intero contenitore; inoltre sia base che coperchio hanno la parte anteriore diversa rispetto a quella posteriore: ai lati della parte anteriore ci sono due scanalature, mentre sulla parte posteriore ce n'è solo una.

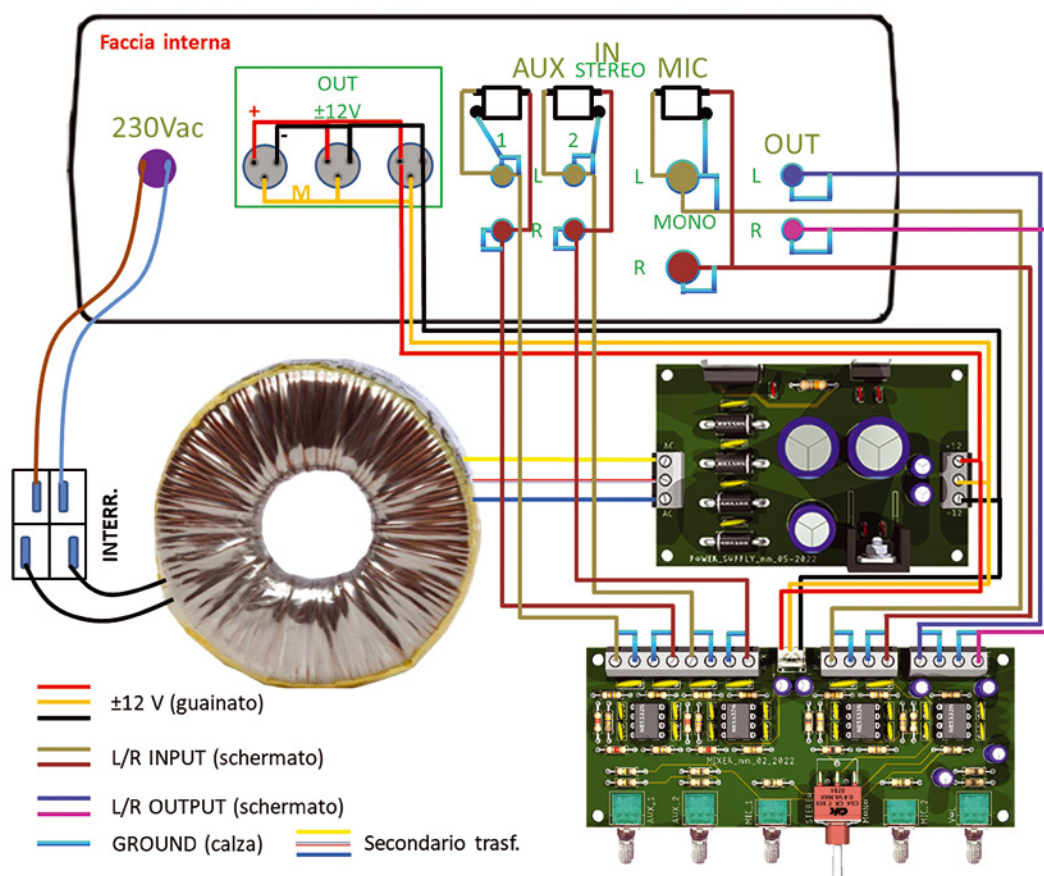


Fig. 3
Schema di
cablaggio
di mixer e
alimentatore
generale.

Infine, le misure che forniremo, per le forature di fissaggio di PCB e trasformatori, saranno sempre riferite ai bordi esterni del contenitore e non ai pannelli. Bisognerà fare quindi molta attenzione alla foratura della base, altrimenti si rischierà di commettere degli errori.

KZ17W PER ALIMENTATORE E MIXER

Nel primo dei due KZ17W montiamo due PCB ed un trasformatore toroidale (**Fig. 2**).

Il PCB del mixer ha il posizionamento obbligato, occorre fissarlo su 4 distanziali lunghi esattamente 33 mm e poi infilare i perni dei potenziometri nei rispettivi fori del pannello anteriore; a questo punto si possono segnare i punti in cui i distanziali toccano la base e quindi procedere con la foratura. Il trasformatore toroidale, se ha le stesse dimensioni del nostro (diametro 93 mm), va centrato ad una distanza di 90 mm dal bordo posteriore e di 55 mm dal bordo laterale sinistro. Infine il PCB dell'alimentatore generale va fissato su 4 distanziali lunghi circa 20 mm e posizionato indicativamente a 25 mm dal bordo laterale destro e a 70 mm (non considerando il dissipatore) dal bordo posteriore.

Dopo aver fissato questo PCB si può procedere con l'aggiunta del dissipatore grande per il TIP2955 ed il 7812, centrandolo rispetto al PCB e facendolo aderire contemporaneamente sia al bordo del PCB stesso che alla base del contenitore, in modo da impedire movimenti del dissipatore, che potrebbero portare alla rottura dei pin dei due componenti; inoltre in questo modo, con le dimensioni che abbiamo fornito quando abbiamo visto la preparazione del PCB, non ci saranno problemi in altezza. L'operazione successiva è quella di montare l'interruttore sul pannello anteriore e tutte le varie prese sul pannello posteriore, quindi si può procedere con i collegamenti.

La **Fig. 3** mostra lo schema di cablaggio del mixer e dell'alimentatore generale.

Nel disegno abbiamo riportato, per comodità, le serigrafie che in realtà si trovano sulla faccia esterna del pannello. Le tre prese XLR12 (o XLR12 mini) vanno collegate in parallelo tra loro e poi all'uscita dell'alimentatore; ribadiamo la necessità di fare molta attenzione a rispettare colori e collegamenti, perché un'inversione dell'alimentazione provocherebbe danni enormi!

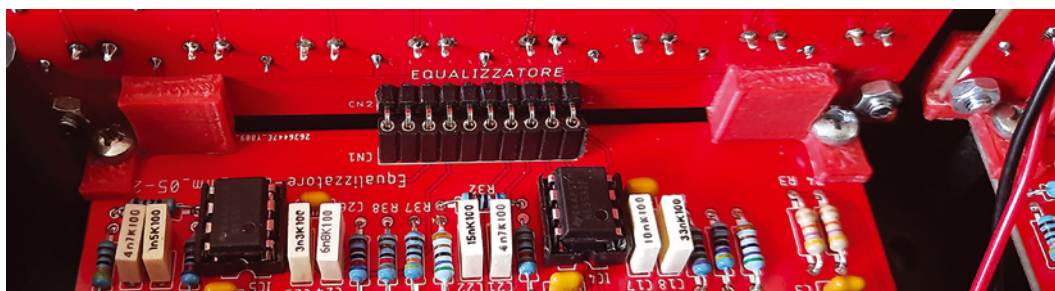
→ **Fig. 4**
Interno del pannello
posteriore di mixer
e alimentatore
generale.



→ **Fig. 5**
Mixer e alimentatore
generale montati
nel contenitore



→ **Fig. 6**
PCB base e slider
innestati con i
supporti stampati
in 3D.



Sempre all'uscita dell'alimentatore va collegato il cavetto con terminazione MK a tre pin, da innestare sul connettore J3 del mixer; poiché in genere questi cavetti hanno colori casuali, occorre fare attenzione alle serigrafie dei simboli "+" e "-" accanto a J3, che indicheranno con certezza assoluta la polarità dei due pin laterali, mentre quello centrale è sempre la massa comune, come del resto sul morsetto a vite J2 del PCB dell'alimentatore generale. La figura comunque è inequivocabile. Per quanto riguarda il collegamento alla rete elettrica serve un cavo bipolare, con spina standard, da far passare nello specifico foro del pannello posteriore, possibilmente usando un qualsiasi fermacavo; i due fili dovranno essere collegati all'interruttore doppio mediante due spinotti faston isolati; agli altri due poli dell'interruttore bisognerà collegare i due fili d'ingresso del trasformatore toroidale (nel nostro caso sono di colore nero). Trattandosi di un trasformatore di tipo duale, le uscite sono costituite da 4 fili colorati; occorre fare attenzione a collegarli correttamente in quanto due di questi fili saranno uniti tra loro e rappresenteranno la massa comune, in caso di errore si avrebbero risultati inattesi, con possibili danni alla circuiteria. Il trasformatore usato da noi, come rappresentato nello schema di cablaggio, va collegato nel seguente modo:

- cavetto giallo ad una estremità del morsetto J1;

- cavetto blu all'altra estremità del morsetto J1;
- cavetti rosso e grigio, uniti tra loro, al centrale del morsetto J1.

Tutto il resto dei collegamenti deve essere effettuato mediante cavetto schermato: la regola generale è che, lato PCB, i fili centrali vanno collegati agli esterni dei morsetti a vite, mentre le rispettive calze a quelli interni. Lato pannello i fili centrali vanno saldati ai pin centrali delle prese RCA e JACK mono da 6,3 mm, quindi, come da schema, vanno prolungati ai due pin laterali delle prese JACK stereo da 3,5 mm. Le calze invece vanno saldate sulle pagliette in dotazione prese RCA, sui pin di massa delle prese JACK mono e sui pad più grandi delle prese JACK stereo. Il risultato finale sarà quello visibile nelle **Fig. 4 e Fig. 5**.

KZ15 PER EQUALIZZATORE

Nel primo dei due KZ15 montiamo i quattro PCB relativi all'equalizzatore, infatti per ogni canale servono un PCB con i filtri ed uno con gli slider. I PCB dell'equalizzatore, come quello del mixer, hanno il posizionamento obbligato, per il fatto che quelli con gli slider devono essere attaccati al pannello anteriore e sono collegati a quelli dei filtri mediante una coppia di connettori strip-header. In questo caso specifico abbiamo disegnato e stampato in 3D due coppie di supporti ad angolo, che hanno la funzione di stabilizzare il contatto tra

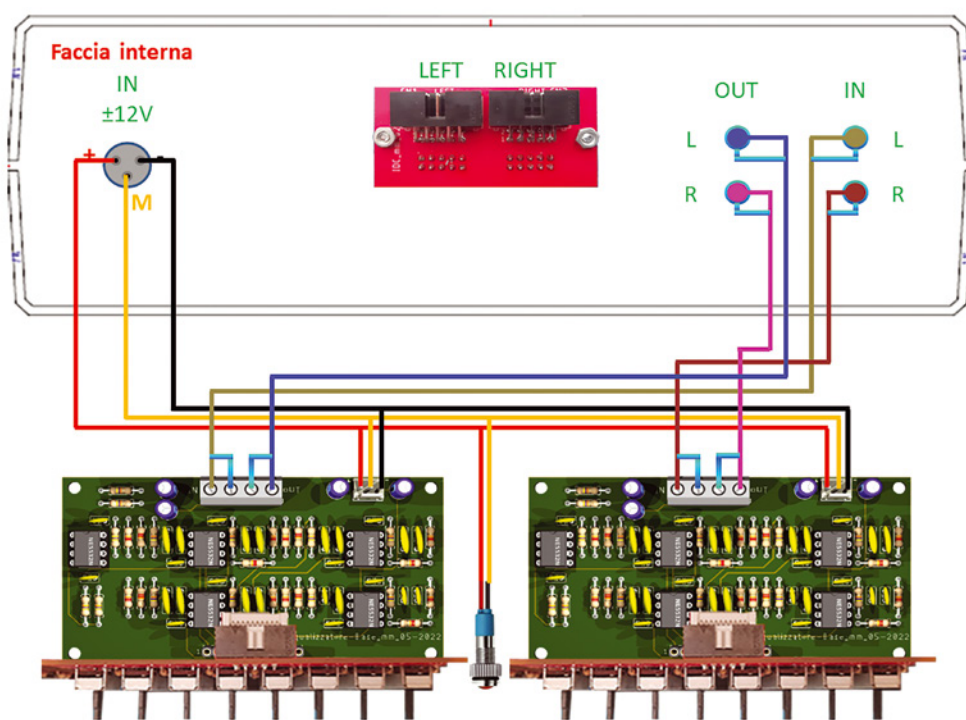


Fig. 7
Schema di
cablaggio
dell'equalizzatore.

- ±12 V (guainato)
- ±12 V (guainato)
- L/R INPUT (schermato)
- L/R OUTPUT (schermato)
- GROUND (calza)

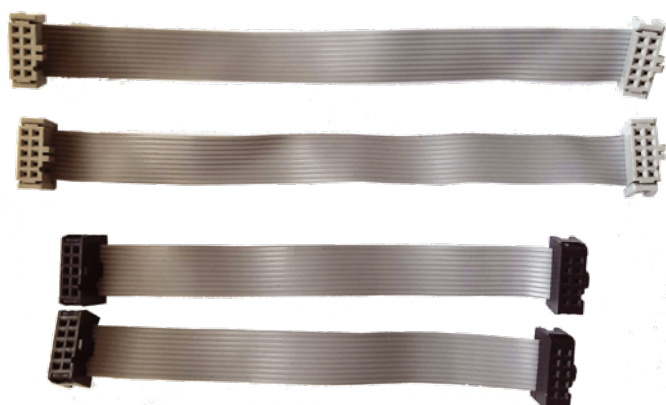


Fig. 8
I cavetti IDC per
i collegamenti
interni ed esterni
dell'equalizzatore.

i PCB e garantire la necessaria rigidità degli slider. Nella **Fig. 6** si vedono i due supporti ad angolo sinistro e destro ed una delle due coppie di PCB. Dopo aver collegato tra loro le due coppie di PCB, occorre fissare quelli di base su 4 distanziali lunghi 12 mm e poi infilare le levette degli slider nelle corrispondenti aperture del pannello anteriore; a questo punto si possono segnare i punti in cui i distanziali toccano la base del contenitore e quindi procedere con la foratura.

Sul pannello anteriore andrà fissata anche la spia luminosa a 12/24 V (www.futurashop.it 1606-HR2242BLU).

Per quanto riguarda lo schema di cablaggio dell'equalizzatore, vale quanto già spiegato per quello del mixer; bisogna seguire la **Fig. 7**.

Come visibile nello schema, la spia luminosa va collegata tra +12 V e la massa comune, il tipo da noi usato in realtà potrebbe essere collegato tra +12 V e -12 V, potendo funzionare anche a 24 V, ma poiché la maggior parte delle spie lavora a 12 V, conviene rispettare il nostro disegno.

Vanno poi realizzati quattro cavetti con una piastrina da 10 poli e otto spinotti IDC 10 pin femmina (**Fig. 8**); due (lunghezza totale circa 15 cm) servono per collegare internamente il PCB degli slider con l'adattatore IDC montato sul pannello posteriore, gli altri due (lunghezza in base alla distanza tra i due apparecchi) serviranno per portare esternamente il segnale equalizzato al visualizzatore di spettro. Nella fase di crimpatura della piastrina occorre fare attenzione ad orientare gli spinotti nella stessa direzione. Una volta completato il montaggio e tutti i collegamenti il nostro equalizzatore apparirà come in **Fig. 9**.

Fig. 9
Equalizzatore mon-
tato nel contenitore.

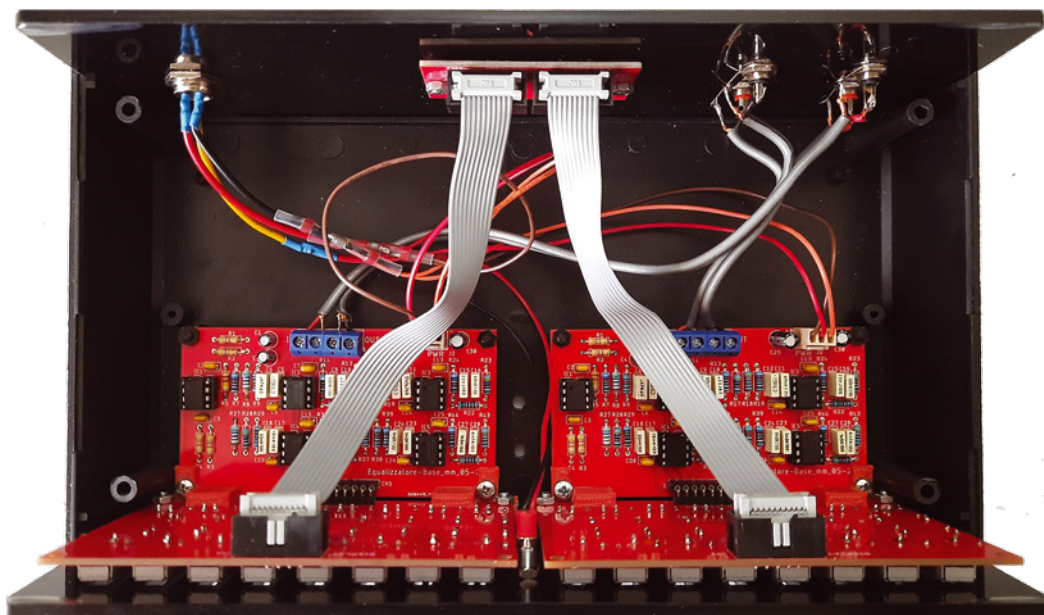


Fig. 10
Barra stampata in
3D per il fissaggio
dei PCB con le
barre di LED al
pannello anteriore.



KZ15 PER VISUALIZZATORE DI SPETTRO AUDIO

Nel secondo KZ15 montiamo i quattro PCB relativi al visualizzatore di spettro audio; anche in questo caso, per ogni canale, servono un PCB con i circuiti di amplificazione e raddrizzamento del segnale ed uno con le barre di LED. I PCB di condizionamento del segnale possono essere posizionati in un punto qualsiasi del contenitore, e fissati alla sua base mediante quattro distanziali da 12-20 mm.

I PCB con le barre di LED invece vanno fissati sul retro del pannello anteriore in modo da far fuoriuscire le barre dai riquadri corrispondenti. Poiché il pannello non presenta punti di ancoraggio e dato che per questi PCB non c'è praticamente alcun margine di tolleranza nel contenitore, abbiamo stampato una barra in 3D, contenente 4 fori, nei quali andranno inseriti altrettanti dadi da incasso per 3D (www.futurashop.it 2846-CH46222) con misure M3 x 10 mm (**Fig. 10**).

Dalla sezione download è possibile scaricare il file STL per stamparla con una qualsiasi stampante 3D. Essa andrà incollata sul retro del pannello ante-

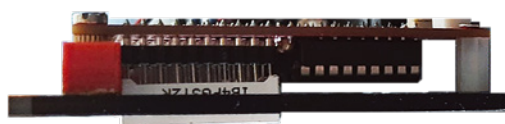


Fig. 11
Distanziali inferiori del PCB con le barre di LED.

riore, in corrispondenza dei fori superiori dei PCB, nei quali andranno poi inserite quattro viti M3; ai quattro fori inferiori dei due PCB andranno invece fissati dei distanziali da 10 mm, in modo da garantire lo stesso spessore sia in alto che in basso (**Fig. 11**). Al centro del pannello anteriore andranno inseriti l'interruttore doppio per la commutazione BAR/DOT e la spia luminosa a 12/24V. Per lo schema di cablaggio bisogna seguire la **Fig. 12**.

Servono solo normali fili guainati, rispettando al solito i colori per l'alimentazione, mentre il collegamento del deviatore doppio non richiede alcuna polarità. Come si vede nello schema, l'alimentazione duale va portata ai PCB base, mentre per quelli delle barre di LED e per la spia luminosa servono solo +12 V.

Come per l'equalizzatore, vanno realizzati quattro cavetti con una piattina e otto spinotti IDC femmi-

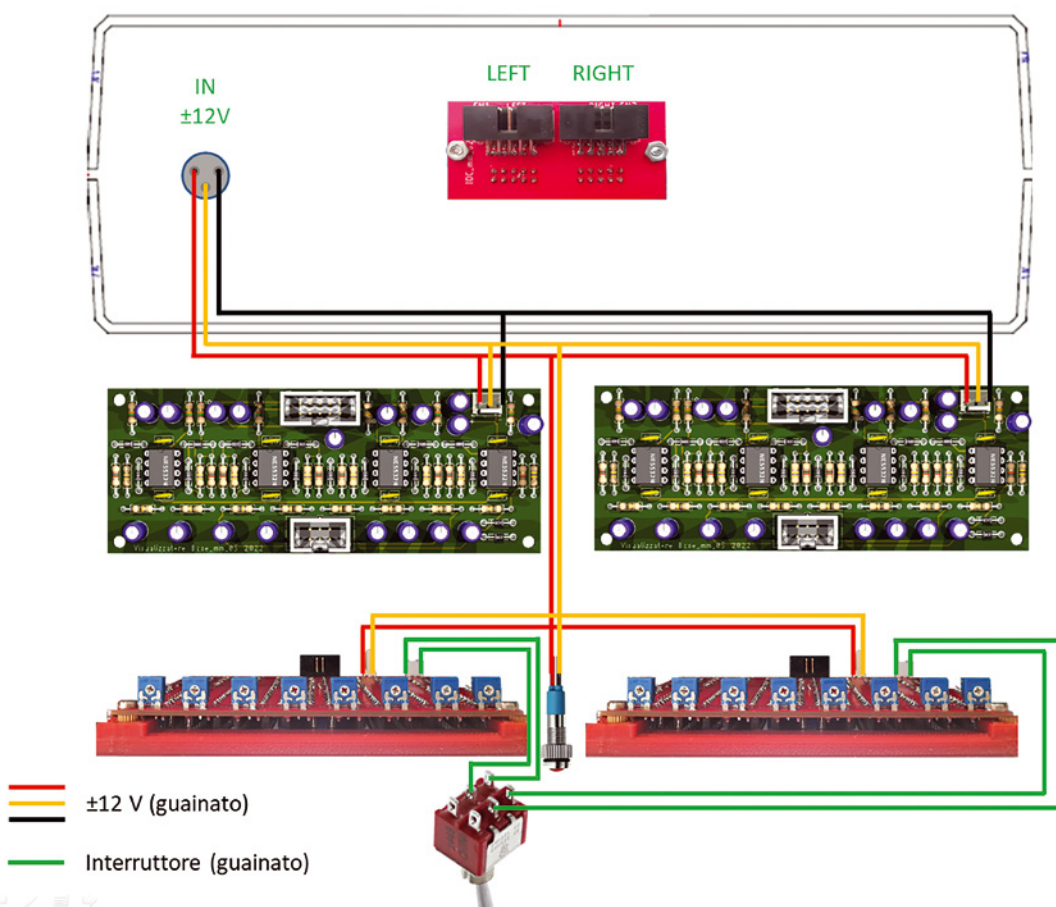


Fig. 12
Schema di cablaggio del visualizzatore di spettro audio.

Fig. 13
Visualizzatore di
spettro audio montato
nel contenitore.

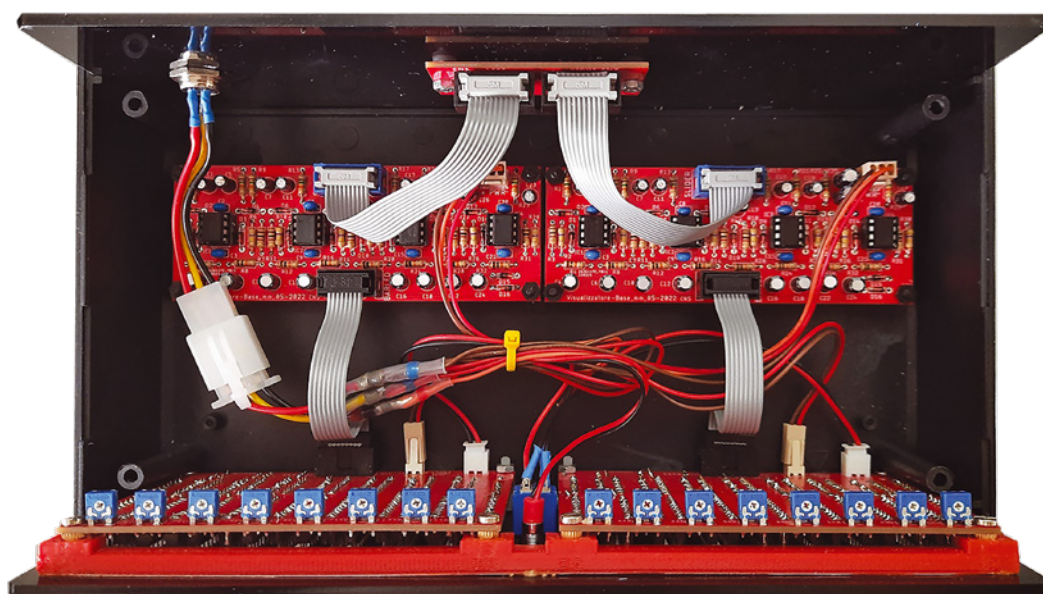
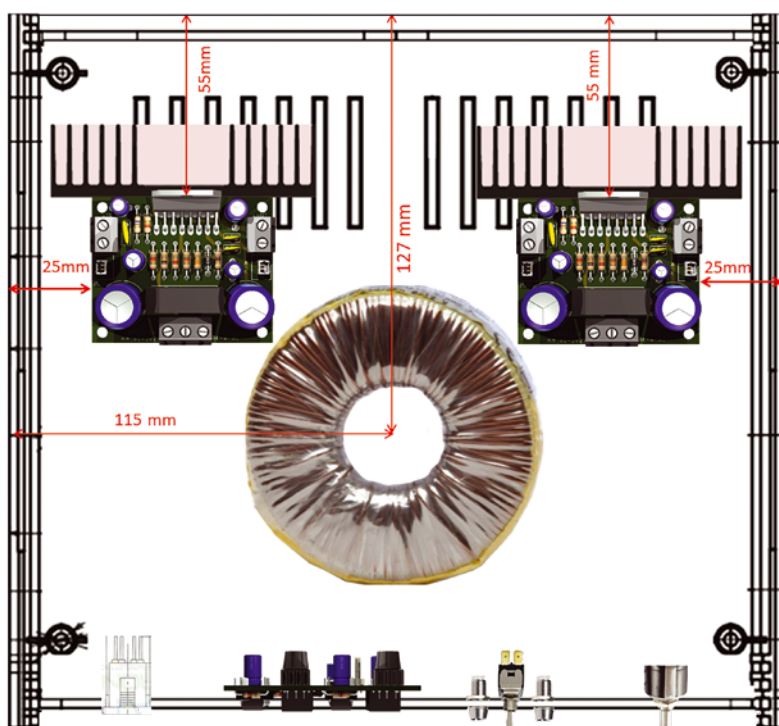


Fig. 14
Quote per il mon-
taggio dei PCB e
del trasformatore
toroidale dell'ampli-
ficatore finale.



na, tutti lunghi circa 10 cm, ma in questo caso due sono da 10 poli, e servono per collegare internamente il PCB base con l'adattatore IDC montato sul pannello posteriore, gli altri due invece sono da 8 poli e portano il segnale dalla base alle barre di LED.

In **Fig. 13** l'interno del visualizzatore di spettro audio dopo aver completato montaggio e collegamenti.

KZ17W PER AMPLIFICATORE FINALE E VU-METER

Nel secondo KZ17W montiamo tre PCB ed un trasformatore toroidale (**Fig. 14**).

Ricordiamo che le misure indicate sono relative al trasformatore toroidale usato da noi, ma in generale consigliamo di posizionare i due PCB (con i dissipatori montati) ed il trasformatore e trovare la soluzione migliore e solo dopo segnare i punti di foratura. Conviene anche montare prima tutti gli accessori sul pannello posteriore e anteriore, oltre al PCB del VU-METER, perché bisogna tener conto anche delle loro sporgenze interne oltre a quelle dei vari cavetti di collegamento.

Sul pannello anteriore vanno montati: interruttore generale, due portaled 3 mm, un deviatore doppio ed un potenziometro doppio; per quanto riguarda il PCB del VU-METER abbiamo realizzato in 3D un riquadro con quattro torrette nelle quali occorre infilare altrettanti dadi da incasso per 3D con misure M3 x 8+10 mm (usare una misura che permetta alle due barre di LED di trovarsi a filo con il pannello) (**Fig. 15**).

Nella sezione download è possibile scaricare il file STL per stampare il riquadro con una qualsiasi stampante 3D. Esso andrà incollato sul retro del pannello anteriore; il fissaggio del PCB andrà effettuato mediante quattro viti M3.

Per quanto riguarda lo schema di cablaggio dell'amplificatore e del VU-METER, bisogna seguire la **Fig. 16**.

I cavetti schermati saldati alle due prese RCA portano il segnale proveniente dall'equalizzatore, le

loro estremità opposte vanno saldate ai pin laterali del potenziometro doppio di regolazione del volume; con un'ulteriore coppia di cavetti schermati, saldando le calze in comune, i pin centrali dei potenziometri vanno collegati agli ingressi J1 dei due PCB dell'amplificatore finale, rispettando la polarità (il filo centrale va collegato al + del connettore). Le uscite J3 dei due amplificatori, sempre mediante cavetto schermato, vanno saldate ai morsetti per gli altoparlanti; da J3, o anche dai morsetti degli altoparlanti, mediante normale cavetto guainato sottile, si preleverà il segnale da portare agli ingressi J1 del VU-METER.

Le due prese J2 degli amplificatori, mediante cavetto guainato sottile, vanno collegate al deviatore doppio, per attivare o disattivare i comandi di MUTE/STAND-BY.

Le due prese J5 degli amplificatori, mediante cavetto guainato sottile, vanno collegate ai due LED per la segnalazione del clipping, anche in questo caso va rispettata la polarità: i due pin dei connettori

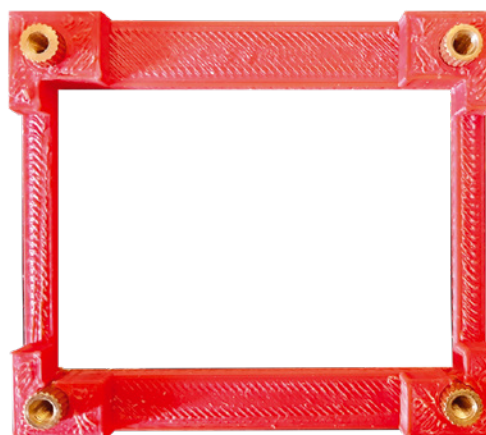


Fig. 15
Riquadro stampato in 3D per il fissaggio del VU-METER.

sono contrassegnati con A e K, il primo va collegato all'anodo del LED (reoforo più lungo), mentre l'altro va al catodo dello stesso LED.

Il VU-METER richiede la sola alimentazione di +12 V, bisogna prelevare il polo positivo e la massa comune dallo spinotto XLR12 (o XLR12 mini) e

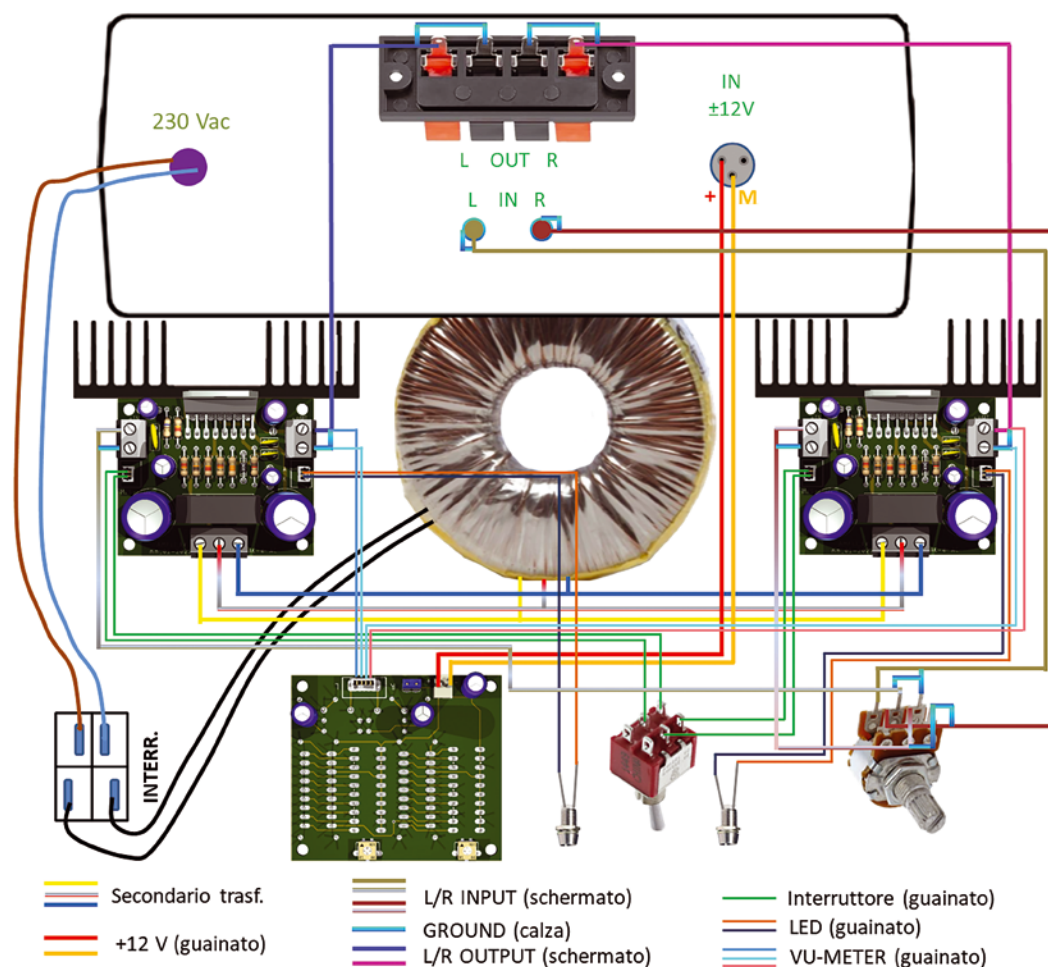


Fig. 16
Schema di cablaggio di amplificatore finale e VU-METER.

Fig. 17
Schema di realizzazione dei tre cavi di alimentazione esterna.



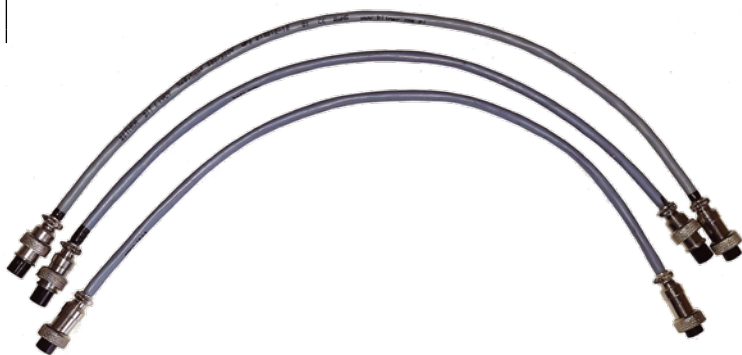
portarli alla presa MK J2, rispettando la polarità indicata dai "+" e "-" serigrafati sul PCB. Per quanto riguarda il collegamento alla rete elettrica serve un cavo bipolare, con spina standard, da far passare nello specifico foro del pannello posteriore, possibilmente usando un qualsiasi fermacavo; i due fili dovranno essere collegati all'interruttore doppio mediante due spinotti faston isolati; agli altri due poli dell'interruttore bisognerà collegare i due fili d'ingresso del trasformatore toroidale (nel nostro caso sono di colore nero). Trattandosi di un trasformatore di tipo duale, le uscite sono costituite da 4 fili colorati; occorre fare attenzione a collegarli correttamente in quanto due di questi fili saranno uniti tra loro e rappresenteranno la massa comune, in caso di errore si avrebbero risultati inattesi, con possibili danni alla circuiteria.

Il trasformatore usato da noi, come rappresentato nello schema di cablaggio, va collegato nel seguente modo (fare riferimento allo schema di cablaggio):

- il cavetto giallo va sdoppiato, saldandovi due fili, ognuno dei quali sarà collegato ad una estremità di uno dei due morsetti J4;
- il cavetto blu va sdoppiato, saldandovi due fili, ognuno dei quali sarà collegato ad una estremità di uno dei due morsetti J4;
- i cavetti rosso e grigio vanno prima uniti tra loro, e poi sdoppiati, saldandovi due fili, ognuno dei quali sarà collegato al centrale di uno dei due morsetti J4.

Nota bene: in pratica i cavetti rosso e grigio vanno portati entrambi ad ognuno dei due PCB,

Fig. 18
I tre cavi XLR12 per l'alimentazione.



se vengono erroneamente separati si fornirà ad ognuno dei due PCB un solo ramo di alimentazione, con probabile rottura del circuito integrato finale.

COLLEGAMENTI ESTERNI

L'ultima operazione da fare, una volta completato il montaggio interno dei contenitori, è quella di effettuare i collegamenti esterni; oltre ai due cavetti piattina ed ai tre cavi per l'alimentazione, serviranno due coppie di cavetti audio RCA maschio-maschio, per collegare l'uscita del mixer all'ingresso dell'equalizzatore e l'uscita di quest'ultimo all'ingresso dell'amplificatore finale. Servirà inoltre una fonte audio, che inizialmente dovrebbe essere rappresentata da un generatore di funzioni, per la taratura dei 18 trimmer dei visualizzatori di spettro audio e del VU-METER. Servirà inoltre una fonte musicale che potrà essere rappresentata da un lettore CD portatile, o anche da un lettore MP3, in entrambi i casi servirà un ulteriore cavo audio stereo che termini con due RCA maschio o con un JACK 3,5 mm stereo. Ancora, ai fini di un completo collaudo, si dovrebbe disporre di almeno un microfono professionale, dotato di un cavo terminante con uno spinotto jack maschio mono da 6,3 mm, meglio se sono due per testare la modalità stereo. Infine occorrerà una coppia di casse passive, possibilmente a tre vie, cioè dotate di un woofer per i toni bassi, un midrange per i toni medi ed un tweeter per i toni alti, solo così si potrà apprezzare la qualità del nostro impianto Audio PRO. Naturalmente le casse dovranno avere potenza adeguata a quella che otterremo da nostro amplificatore finale, in base al tipo di alimentazione che avremo deciso di fornire. Nel nostro caso saranno più che sufficienti delle casse da 30-40 watt l'una. Le tre prese XLR 12 (o XLR12 mini) maschio presenti sul retro del contenitore che ospita mixer e alimentatore generale, vanno collegate, mediante altrettanti cavi XLR femmina-femmina, alle singole prese presenti su equalizzatore, visualizzatore di spettro audio e amplificatore finale. Per la realizzazione dei tre cavi esterni, che servono per portare l'alimentazione di ± 12 V dall'alimentatore generale agli altri tre apparecchi, servono sei spinotti femmina (XLR12 o XLR12 mini) dello stesso tipo usato per i maschi montati sui pannelli posteriori, e del cavo guainato tripolare, di lunghezza idonea. Anche se all'amplificatore finale deve arrivare il solo ramo positivo per l'alimentazione del VU-METER, conviene che il cavo sia comunque tripolare, così non ci saranno problemi

di intercambiabilità.

I cavi devono essere realizzati seguendo rigorosamente lo schema della **Fig. 17**, in caso di errori si faranno sicuramente gravi danni ai circuiti.

La vista dell'immagine è lato saldature, in pratica è un collegamento pin-to-pin, cioè ogni pin del primo spinotto va collegato al corrispondente pin dell'altro spinotto. Il risultato finale sarà quello della **Fig. 18**.

Le due prese RCA "OUT L & R" del mixer vanno collegate, mediante un cavetto audio stereo RCA maschio-maschio alle prese RCA "IN L & R" dell'equalizzatore. Le due prese RCA "OUT L & R" dell'equalizzatore vanno collegate, mediante un cavetto audio stereo RCA maschio-maschio alle prese RCA "IN L & R" dell'amplificatore finale. Le due prese IDC esterne dell'equalizzatore vanno collegate alle corrispondenti prese esterne del visualizzatore di spettro audio.

Vediamo ora i collegamenti delle fonti al mixer.

Un lettore CD ed uno MP3 si possono collegare contemporaneamente, uno all'ingresso AUX1 e l'altro all'ingresso AUX2, o viceversa; il collegamento può avvenire o sulle prese RCA "AUX IN L&R", mediante un cavo audio stereo RCA, oppure sulle prese jack stereo 3,5 mm; naturalmente ogni ingresso AUX può ospitare un solo apparecchio, quindi se si usano le prese RCA non si può usare la jack stereo, e viceversa; è invece possibile collegare un lettore alla presa jack stereo AUX1 ed un altro alle prese RCA AUX2, o viceversa.

Due microfoni mono possono essere collegati alle due prese jack mono 6,3 mm MIC L e MIC R; un microfono stereo può essere collegato alla presa jack stereo 3,5 mm. Naturalmente non sarà possibile usare contemporaneamente uno o due ingressi mono e l'ingresso stereo; nella parte relativa alla descrizione dello schema elettrico del mixer abbiamo già chiarito la funzionalità del deviatore STEREO/MONO.

Il fatto di aver realizzato un mixer invece del tipico preamplificatore con selettore, ci permetterà di miscelare più fonti d'ingresso, per esempio, si potrebbe collegare ad un ingresso AUX una base musicale da miscelare con l'audio di un microfono, in modo da ottenere un effetto karaoke.

TEST FINALE DELL'IMPIANTO CON STRUMENTAZIONE DI LABORATORIO

Per questo test sono necessari un generatore di funzioni e un oscilloscopio digitale; disponendo di questi due strumenti sarà possibile eseguire tutte le operazioni che descriveremo di seguito, in caso



Fig. 19
Collegamento delle sonde del DSO.

contrario si può direttamente alla sezione successiva per il test con il simulatore on line.

Il test finale del nostro impianto Audio PRO si divide in due fasi: la prima coinvolge mixer, equalizzatore e visualizzatore di spettro audio, regolarmente collegati tra loro, come spiegato in precedenza, per cui l'amplificatore finale dovrà essere scollegato e tenuto spento. Alla seconda fase si passerà dopo aver terminato la prima, bisognerà collegare l'amplificatore finale all'equalizzatore e le due casse ai relativi morsetti. In questa fase non bisogna ancora montare i coperchi dei tre contenitori, in quanto il test prevede una serie di misure da confrontare con quelle rilevate da noi e riportate in alcune tabelle riepilogative.

Inizialmente bisogna posizionare l'interruttore dell'alimentatore generale su OFF e collegare la spina di alimentazione alla presa di rete elettrica, dopo aver posizionato tutti i potenziometri, rotativi e slider, al minimo (perni ruotati completamente verso sinistra e cursori in basso). A questo punto si può portare l'interruttore generale su ON, si accenderanno le due spie luminose e potrebbero accendersi per un istante i primi LED delle barre del visualizzatore di spettro audio.

Frequenza	Vin AUX	Vin MIC	Vout
125	20 mVpp	9 mVpp	450 mVpp
250			444 mVpp
500			444 mVpp
1k			444 mVpp
2k			448 mVpp
4k			444 mVpp
8k			428 mVpp
16k			380 mVpp

Tabella 1
Misure di test degli ingressi.

TEST INGRESSI AUX

Il generatore di funzioni dovrà essere impostato con questi parametri: segnale sinusoidale, frequenza 1 kHz, ampiezza 20 mVpp, offset 0 V.

Nota bene: tutte le nostre misure e tarature si basano sulla corrispondenza del valore applicato in ingresso.

Il segnale dovrà essere applicato contemporaneamente ai due ingressi "AUX1 L & R" (l'operazione dovrà essere poi ripetuta per AUX2). Disponendo di un generatore di funzioni a due canali, è possibile impostarli in modo identico usando un canale per ogni ingresso, ma non è indispensabile.

I potenziometri AUX1 e VOLUME del mixer dovranno essere posizionati al massimo (tutti gli altri al minimo), le due sonde del DSO dovranno essere collegate all'interno del pannello posteriore del mixer, sulle prese "OUT L & R", con le pinzette sulla calza dei cavetti schermati (**Fig. 19**).

Nota bene: È importante che siano collegati, e di conseguenza accesi, sia l'equalizzatore che il visualizzatore di spettro audio, in caso contrario si otterrebbero valori diversi da quelli da noi indicati. In queste condizioni si dovrà osservare un segnale sinusoidale di circa 440 mVpp, una differenza di ± 20 mVpp sarebbe giustificata dalle tolleranze dei componenti usati. Se invece si dovesse riscontrare una differenza notevole su uno o entrambi i canali destro e sinistro, bisognerà verificare la componentistica attorno agli integrati IC1 e IC4. La misura va ripetuta variando la frequenza del segnale, ed usando quelle previste per l'equalizzatore, da 125 Hz a 16 kHz, in modo da verificare il comportamento della banda passante dei due stadi AUX e degli stadi finali del mixer. Confrontando i valori ottenuti con quelli riportati nella **Tabella 1** non si dovrebbero notare particolari differenze, eccetto quanto appena spiegato.

TEST INGRESSI MIC

Il generatore di funzioni dovrà essere impostato con questi parametri: segnale sinusoidale, frequen-

za 1 kHz, ampiezza 9 mVpp, offset 0 V.

Il segnale dovrà essere applicato contemporaneamente ai due ingressi "MIC L (1) e R (2)"; disponendo di un generatore di funzioni a due canali, essi possono essere impostati in modo identico usando un canale per ogni ingresso, ma non è indispensabile; il deviatore S1 andrà posizionato su STEREO, in modo da testare i due ingressi separatamente.

I potenziometri MIC1, MIC2 e VOLUME del mixer dovranno essere posizionati al massimo (tutti gli altri al minimo), le due sonde del DSO dovranno essere collegate come per il test degli ingressi AUX. In queste condizioni si dovrà vedere un segnale sinusoidale di circa 440 mVpp, una differenza di ± 20 mVpp sarebbe giustificata dalle tolleranze dei componenti usati. Se invece si dovesse riscontrare una differenza notevole su uno o entrambi i canali destro e sinistro, bisognerà verificare la componentistica attorno agli integrati IC3 e IC4.

La misura va ripetuta variando la frequenza del segnale, ed usando quelle previste per il centro banda dell'equalizzatore, da 125 Hz a 16 kHz, in modo da verificare il comportamento della banda passante dei due stadi AUX e degli stadi finali del mixer.

Non si dovrebbero notare particolari differenze confrontando i valori ottenuti con quelli riportati nella precedente tabella riepilogativa.

Se, come previsto, i due ingressi AUX ed i due ingressi MIC, con i segnali consigliati, daranno lo stesso risultato, o comunque molto simile, per il prosieguo delle prove basterà mantenere il generatore costantemente collegato agli ingressi "AUX1 L & R".

TEST EQUALIZZATORE

Questo test serve per verificare la corretta risposta degli otto filtri attivi che compongono ognuno dei due canali dell'equalizzatore stereo.

Il test consiste nel rilevare l'ampiezza del segnale in uscita, agendo su una frequenza per volta, e posizionando il cursore del relativo slider prima al minimo, poi in posizione intermedia ed infine al massimo, mentre tutti gli altri cursori devono restare sempre al minimo.

Il segnale applicato agli ingressi "AUX1 L & R" del mixer deve avere queste caratteristiche: frequenza da 125 Hz a 16 kHz (il centro banda previsto) da applicare in sequenza, ampiezza 20 mVpp, offset 0 V. Le sonde del DSO vanno collegate come in precedenza, ma questa volta sugli RCA "OUT L & R" dell'equalizzatore, misurando prima un canale e poi l'altro.

Anche in questo caso abbiamo rilevato le misure

Tabella 2
Misure di test
dell'equalizzatore.

Frequenza Hz	Signal IN mVpp	SLIDER (mVpp)			GAIN (dB)		
		min	med	max	min	med	max
125	450	130	704	1440	-10,8	3,887	10,1
250	444	138	732	1480	-10,2	4,343	10,46
500	444	144	726	1470	-9,78	4,271	10,4
1k	444	148	756	1470	-9,54	4,623	10,4
2k	448	148	748	1510	-9,62	4,452	10,55
4k	444	144	728	1450	-9,78	4,295	10,28
8k	428	136	698	1420	-9,96	4,248	10,42
16k	380	116	600	1210	-10,3	3,967	10,06

di ampiezza sul nostro prototipo e le abbiamo riportate nella **Tabella 2**.

La tabella è così strutturata: ogni riga riguarda una delle otto frequenze di centro banda, riportate nella prima colonna, la seconda colonna contiene i valori della tabella precedente, misurati quindi all'uscita del mixer (o all'ingresso dell'equalizzatore); in terza, quarta e quinta colonna abbiamo riportato i valori da noi misurati, in mVpp, sull'uscita dell'equalizzatore, a seconda della posizione del cursore dello slider; le ultime tre colonne riportano il guadagno in dB corrispondente alle posizioni dei cursori per le singole frequenze.

Nota bene: Come si vede chiaramente nella tabella, la posizione dei cursori a metà corsa non corrisponde ad un guadagno di 0 dB (come riportato sulla serigrafia del pannello) che rappresenta un segnale di uscita uguale a quello di ingresso; questo valore si ottiene con i cursori ad $\frac{1}{4}$ della loro corsa.

Il comportamento dei due canali dovrà essere molto simile, oltre che sufficientemente coerente con i valori da noi riportati, in caso contrario bisognerà verificare i circuiti dell'equalizzatore. Naturalmente occorre considerare sempre una certa tolleranza, si tratta pur sempre di circuiti analogici, con i comportamenti strettamente legati ai valori dei componenti passivi, noi abbiamo usato resistori con una tolleranza dell'1%, quindi se sono stati usati valori corretti, non ci saranno particolari influenze, mentre i condensatori possono arrivare anche al 20% di tolleranza. Solo quando si sarà sicuri del buon funzionamento dell'equalizzatore si potrà passare al test successivo.

TARATURA VISUALIZZATORE DI SPETTRO

Il test di questo circuito è insito nella taratura che andremo ad effettuare, in quanto eventuali problemi saranno chiaramente visibili in base al comportamento dei LED.

Le sonde del DSO non servono per questa operazione, bisogna invece ripetere tutta la procedura di test dell'equalizzatore, applicando il solito segnale agli ingressi "AUX1 L & R" del mixer, con il corrispondente potenziometro al massimo, così come quello del volume master.

Tutti gli slider dei due canali dell'equalizzatore dovranno essere posizionati al minimo, e si inizierà con la frequenza di 125 Hz.

In queste condizioni tutti i LED delle barre dovranno essere spenti, in caso contrario occorrerà agire sui trimmer corrispondenti, ruotandoli verso sinistra fino ad ottenere lo spegnimento dei LED blu.

A questo punto bisogna portare a metà corsa i cursori dei due slider che regolano la frequenza di 125 Hz e ruotare i due trimmer in modo da ottenere l'accensione dei primi cinque LED di ognuna delle due barre; portando ora i cursori al massimo dovranno risultare accesi tutti i dieci LED. L'operazione naturalmente andrà ripetuta per le altre sette frequenze.

Nota bene: solo con tutti gli slider al minimo dovranno risultare spenti tutti i LED, mentre quando si alzano i cursori sarà normale che si accendano i primi LED delle frequenze adiacenti, in quanto le bande passanti si incrociano agli estremi.

Da tener presente che, in assenza di segnale, nel caso l'impianto capti i 50 Hz della frequenza di rete, potrebbero risultare accesi uno o più LED delle bande 125 Hz e 250 Hz; l'effetto dovrebbe scomparire appena si applicherà un segnale all'ingresso AUX1, però converrà valutare bene, successivamente, con le casse collegate, l'eventuale presenza di un ronzio di bassa frequenza che ovviamente dovrà essere filtrato, per evitare fastidiosi disturbi durante la riproduzione di audio musicali.

Se durante la taratura si dovessero riscontrare comportamenti anomali, è chiaro che bisognerà controllare i valori dei componenti coinvolti nel condizionamento del segnale della frequenza sotto test, quindi sia sul PCB base del visualizzatore che su quello delle barre di LED.

Una valida verifica finale consiste nell'impostare il generatore di funzioni per effettuare uno sweep andata e ritorno, durata totale 4 secondi, con frequenza iniziale 60 Hz e frequenza finale 20 kHz, sempre con ampiezza di 20 mVpp, e tutti i cursori

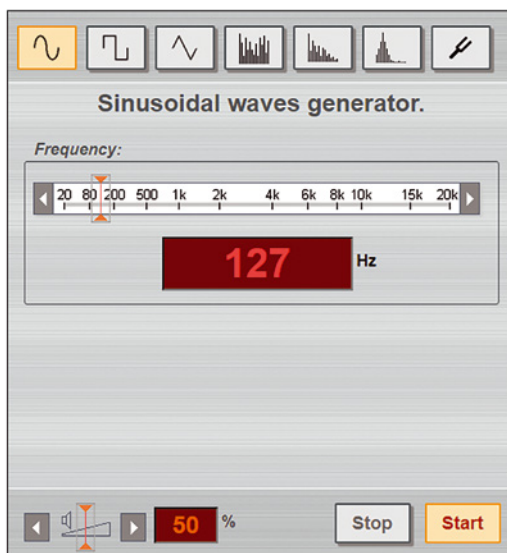
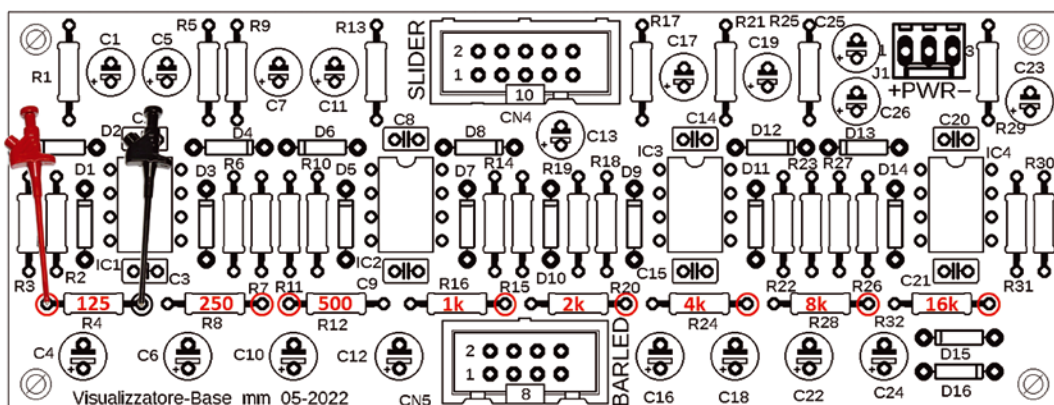


Fig. 20
Generatore di
forme d'onda
on line.



Fig. 21
Collegamento
puntali multimetro.



degli slider a metà corsa. Se lo strumento lo prevede bisogna impostare un andamento logaritmico dello sweep, in tal modo si otterrà una visualizzazione quasi costante per ogni barra di LED, con un effetto molto bello di ondulazione visiva.

TEST FINALE DELL'IMPIANTO CON UN SIMULATORE ON LINE

Bisogna andare all'indirizzo <http://www.claredot.net/en/sec-Sound/audio-signal-generator.php> e si apre la schermata visibile nella **Fig. 20**.

Questo strumento virtuale è molto semplice da usare, bisogna selezionare l'onda sinusoidale, impostare la frequenza tramite il cursore, ricorrendo alle 2 freccette laterali per la regolazione fine; non è semplice riuscire ad impostare una frequenza precisa ma, considerando il funzionamento dei filtri del nostro equalizzatore, differenze dell'ordine di qualche Hz sulle frequenze più basse o di qualche decina di Hz su quelle più alte, non rappresentano affatto un problema.

Per il settaggio iniziale del simulatore conviene usare la frequenza di 1 kHz, il volume al 10% e poi cliccare sul tasto START.

Negli altoparlanti collegati al PC si sentirà istan-

taneamente una nota fissa, con frequenza pari a quella impostata; abbiamo effettuato delle misure e possiamo garantire che essa è molto precisa.

Nota bene: nel momento in cui lo abbiamo testato abbiamo scoperto un piccolo bug, se attivandolo non si dovesse sentire niente negli altoparlanti del PC sarà sufficiente cambiare lingua, cliccando sulla bandierina in alto a destra, appena si sente il suono è possibile settare la lingua preferita, italiana o inglese.

Il volume dell'audio del PC va abbassato fino a sentire il suono molto fiavole, appena percettibile, altrimenti si rischia di mandare in saturazione lo stadio d'ingresso del mixer, falsando il test. Chiaramente, non disponendo di un DSO non potremo fare tutto quanto descritto in precedenza, ma potremo comunque verificare il buon funzionamento dell'impianto tramite un comune multimetro digitale.

Bisogna effettuare tutti i collegamenti tra mixer, equalizzatore e visualizzatore di spettro audio, lasciando scollegato e spento il solo amplificatore finale. Per il collegamento tra PC e mixer serve un cavetto audio con entrambe le terminazioni di tipo jack 3,5 mm stereo (8228-WT69106 reperibile sul sito www.futurashop.it) che sarà utile anche per il test degli ingressi microfonici.

Lato PC lo spinotto andrà inserito al posto di quello degli altoparlanti, mentre lato mixer andrà collegato alla presa jack dell'ingresso AUX1. I potenziometri AUX1 e volume master del mixer vanno posizionati al massimo, mentre i cursori degli slider devono restare al minimo.

I puntali del multimetro digitale vanno collegati in questo modo: negativo su un punto di massa della base del visualizzatore di spettro audio e positivo sul resistore di riferimento della frequenza di test. In **Fig. 21** abbiamo riportato il collegamento dei puntali per il test con 125 Hz.

Tabella 3
Misure di test del
visualizzatore di
spettro audio.

Frequenza	Vraddr AUX	Vraddr MIC
125	1,35	2,90
250	1,43	3,07
500	1,36	2,92
1k	1,39	2,99
2k	1,37	2,94
4k	1,30	2,80
8k	1,29	2,76
16k	1,06	2,25

Il cerchietto nero è il punto in cui si potrà collegare il puntale negativo, ed è valido per tutte le otto misure, mentre i cerchietti rossi sono i punti di ancoraggio del puntale positivo per le varie frequenze. Il multimetro dovrà essere impostato per la lettura di Volt in corrente continua, portata 2 V.

La taratura iniziale consiste nell'usare la frequenza di 1 kHz e rilevare il valore di tensione continua sul resistore R16, con lo slider a metà corsa; a questo punto bisogna regolare il volume del simulatore e dell'audio del PC, fino a leggere sul multimetro una tensione di circa 1,5 V, quindi bisognerà prendere nota del volume attuale, per la successiva taratura. Una volta ottenuto il risultato richiesto, sempre tollerando senza particolari problemi piccole differenze, si può procedere, senza più toccare i volumi del simulatore, del PC e del mixer, con le altre sette frequenze, spostando il solo puntale positivo del multimetro e posizionando a metà corsa il cursore del solo slider corrispondente alla frequenza di test. Abbiamo realizzato la **Tabella 3** per fornire un riferimento di massima, con i valori da noi rilevati, dopo aver applicato un segnale da 20 mVpp, con un generatore di funzioni professionale, agli ingressi "AUX1 L & R" e agli ingressi "MIC L (1) e R (2)". Quando si testeranno gli ingressi microfonici occorrerà portare a zero i volumi degli ingressi AUX1 e AUX2, posizionare il deviatore su "STEREO" e portare al massimo i due volumi MIC1 e MIC2. Come si può facilmente notare ci sono delle differenze che, fino a 4 kHz, potrebbero dipendere anche dalle tolleranze dei componenti, mentre per le frequenze più alte c'è sicuramente un leggero minore guadagno dei filtri.

Non si dovrà cercare di ottenere a tutti i costi questi valori, sarà sufficiente che siano abbastanza simili, mentre differenze notevoli potrebbero indicare qualche errore nella circuiteria della relativa frequenza.

Terminati i test con un canale bisogna ripetere tutto collegando i puntali sui punti di ancoraggio del PCB dell'altro canale, sempre lasciando inalterate le regolazioni dei vari volumi.

Naturalmente, se su qualche frequenza, dell'uno e/o dell'altro canale non si otterranno valori simili a quelli da noi indicati, occorrerà verificare i vari componenti della porzione di circuito interessata. Per quanto riguarda gli ingressi "AUX2 L & R" sarà sufficiente fare una misura con la sola frequenza di 1 kHz, giusto per verificare il relativo stadio, se è a posto non serve ripetere tutto.

La prova con la singola frequenza di 1 KHz andrà fatta anche per gli ingressi MIC1 e MIC2, sempre

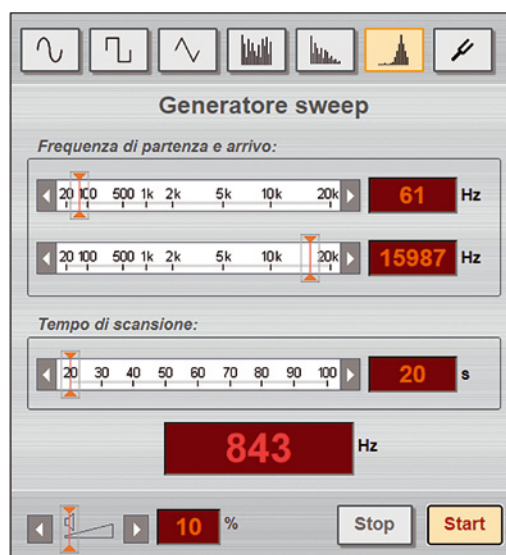


Fig. 22
Simulatore on line in modalità Sweep.

con i volumi del simulatore e dell'audio PC inalterati, ma in questo caso per i valori bisognerà far riferimento all'ultima colonna della tabella.

Una volta completate tutte le operazioni di verifica si potrà passare alla taratura dei trimmer del visualizzatore di spettro audio. I puntali del multimetro non servono più, bisogna applicare il segnale nuovamente agli ingressi "AUX1 L & R". Tutti gli slider dei due canali dell'equalizzatore dovranno essere posizionati al minimo, e si inizierà con la frequenza di 125 Hz.

In queste condizioni tutti i LED di tutte le barre dovranno essere spenti, in caso contrario occorrerà agire sui trimmer corrispondenti, ruotandoli verso sinistra fino ad ottenere lo spegnimento dei LED blu. A questo punto bisogna portare a metà corsa i cursori dei due slider che regolano la frequenza di 125 Hz e ruotare i due trimmer in modo da ottenere l'accensione dei primi cinque LED di ognuna delle due barre; portando ora i cursori al massimo dovranno risultare accesi tutti i dieci LED.

L'operazione naturalmente andrà ripetuta per le altre sette frequenze, sempre badando a mantenere completamente abbassati gli slider delle altre sette frequenze.

Nota bene: solo con tutti gli slider al minimo dovranno risultare spenti tutti i LED, mentre quando si alzano i cursori sarà normale che si accendano i primi LED delle frequenze adiacenti, in quanto le bande passanti si incrociano agli estremi.

Se durante la taratura si dovessero riscontrare comportamenti anomali dei LED, è chiaro che bisognerà controllare i valori dei componenti coinvolti nel condizionamento del segnale della frequenza

Fig. 23
L'impianto Audio
PRO montato.



sotto test, quindi sia sul PCB base che su quello delle barre di LED.

A fine taratura si potrà eseguire un ultimo test con l'intera banda audio, ricorrendo alla funzione sweep del simulatore. Bisogna impostare il simulatore come visibile nella **Fig. 22**.

La frequenza di partenza deve essere circa 60 Hz, quella di arrivo circa 16.000 Hz, il tempo di scansione al minimo (20 secondi), sempre lasciando tutti i volumi inalterati. Cliccando su "Start" si vedranno le barre di LED accendersi con un moto ondulato, partendo da sinistra e finendo a destra; il generatore effettua un solo ciclo, ma è possibile ripremere "Start" all'infinito, anche variando i livelli degli slider. A prescindere dalla modalità usata per effettuare i test, quando tutto funzionerà come abbiamo descritto sarà possibile scollegare il generatore di funzioni o l'uscita audio del PC e collegare tutti i cavi dell'amplificatore finale.

TEST AMPLIFICATORE FINALE E TARATURA VU-METER

Prima di accendere il nostro impianto audio (**Fig. 23**) occorre eseguire alcune operazioni preliminari: collegare le casse, posizionare il deviatore MUTE/STANDBY in posizione OFF, impostare il potenziometro volume generale al minimo.

A questo punto si può collegare una fonte audio (lettore CD o MP3) quindi si posiziona su ON l'interruttore generale dell'amplificatore finale, sempre tenendo il volume al minimo, e si regolano tutti i vari potenziometri e slider per ottenere l'accensione più o meno costante dei primi 5-7 LED di tutte le barre, o almeno di quelle delle 4-5 frequenze più interne, del visualizzatore di spettro audio, visto che le frequenze estreme non sempre raggiungono livelli elevati. Ora si può iniziare ad aumentare il volume leggermente e costantemente, fino ad intravedere i primi lampeggi dei due LED di clipping; in queste condizioni si possono regolare i due trimmer fino a far più o meno coincidere l'accensione del primo LED rosso di ognuna delle due barre con quella del corrispondente LED di clipping.

Finite le operazioni di taratura, impostando un volume generale apprezzabile si possono iniziare a fare tutte le prove desiderate con l'equalizzatore. Un ultimo consiglio importante è quello di lasciare sempre libere le prese di areazione dei due contenitori KZ17W (mixer e alimentatore generale e amplificatore finale) in quanto il calore dissipato deve trovare sfogo, in caso contrario la temperatura elevata potrebbe causare la rottura di qualche componente.

E con questo abbiamo completato la descrizione di questo bellissimo progetto, mettendo in luce ogni dettaglio e procedura necessaria. Ora non ci resta che augurarvi buon divertimento e speriamo che vi divertiate tanto quanto noi nel realizzarlo! ■



COSA OCCORRE ?

Codice	Descrizione	Prezzo € IVA inclusa
PCBPRODUCTION	Servizio realizzazione PCB (il costo dipende dalle dimensioni)	
KZ15	Contenitore plastico 90x148x250 mm	11,80
KZ17W	Contenitore plastico 92x217x235 mm	13,50
HR2242BLU	Segnalatore con LED Blu - 12-24 VDC	2,50
CH46222	Set 370 dadi da incasso per stampa 3D	22,00
WT69106	Cavo audio da jack 3,5 mm a jack 3,5 mm	3,50

I restanti componenti sono di facile reperibilità.

IL MATERIALE VA RICHIESTO A:

Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 - www.futurashop.it

LA TENSIONE DI RETE OVUNQUE TU SIA

INVERTER DA 12 VDC A 220 VAC CON FORMA D'ONDA D'USCITA SINUSOIDALE PURA O MODIFICATA DOTATI DI PROTEZIONE DA SOVRACCARICO, SOVRATENSIONE D'INGRESSO, CORTOCIRCUITO E ALLARME BATTERIA SCARICA. DISPONGONO DI USCITA USB SUPPLEMENTARE (5 V - 500 mA), PRESA DI TIPO SCHUKO E CAVI PER IL COLLEGAMENTO ALLA BATTERIA.

FORMA D'ONDA MODIFICATA



cod. FR670M300
€ 49,00

INVERTER DA 12VDC A 220VAC/300W CON PRESA USB E DISPLAY A LED

- Tensione di ingresso: 12 Vdc nominali
- Tensione di uscita: 220-240 VAC - 50 Hz
- Allarme batteria scarica: 10,5-11,5 Vdc
- Dimensioni (mm): 175x112x56
- Peso: 612 grammi



cod. FR670M600
€ 69,00

INVERTER DA 12VDC A 220VAC/600W CON PRESA USB E DISPLAY A LED

- Tensione di ingresso: 12 Vdc nominali
- Tensione di uscita: 220-240 VAC - 50 Hz
- Allarme batteria scarica: 10,5-11,5 Vdc
- Dimensioni (mm): 205x112x58
- Peso: 760 grammi



cod. FR670M1500
€ 148,00

INVERTER DA 12VDC A 220VAC/1500W CON PRESA USB E DISPLAY A LED

- Tensione di ingresso: 12 Vdc nominali
- Tensione di uscita: 220-240 VAC - 50 Hz
- Allarme batteria scarica: 10,5-11,5 Vdc
- Dimensioni (mm): 270x150x70
- Peso: 1.890 grammi

FORMA D'ONDA PURA



cod. FR668P300
€ 98,00

INVERTER DA 12VDC A 220VAC/300W CON PRESA USB

- Tensione di ingresso: 12 Vdc nominali
- Tensione di uscita: 220-240 VAC - 50 Hz
- Allarme batteria scarica: 10,5-11,5 Vdc
- Dimensioni (mm): 205x112x60
- Peso: 880 grammi



cod. FR668P600
€ 178,00

INVERTER DA 12VDC A 220VAC/600W CON PRESA USB E DISPLAY A LED

- Tensione di ingresso: 12 VDC nominali
- Tensione di uscita: 220-240 VAC - 50 Hz
- Allarme batteria scarica: 10,5-11,5 Vdc
- Dimensioni (mm): 220x115x55
- Peso: 980 grammi

IDEALE
SU CAMPER,
CAMION, ROULOTTE,
PER ALIMENTARE TV,
VCR, PC E MOLTO
ALTRO!

Per studenti di ogni età!

Entra nell'affascinante mondo di Arduino...

Libro "L'ABC di Arduino": una guida pratica per chi vuole scoprire il mondo di Arduino

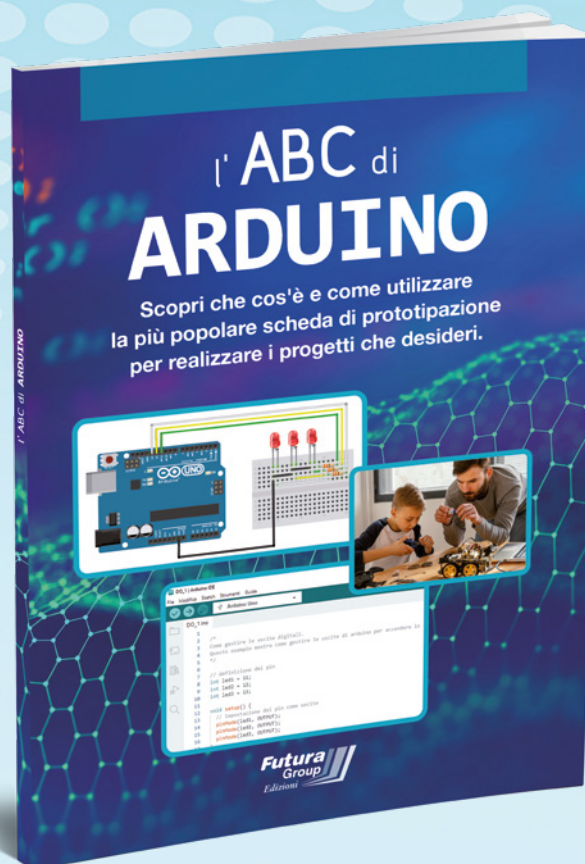
Un libro pensato per i principianti che si avvicinano per la prima volta all'elettronica e per gli hobbisti che vogliono imparare a programmare con la piattaforma Arduino. Comprende numerose esercitazioni didattiche che consentono di mettere in pratica le nozioni acquisite realizzabili con i componenti presenti nel kit "ARDUKITABC".

€ 12,90

cod. BOOKABC



**Guarda il video della
recensione nella scheda
on-line del libro**



...e impara ad utilizzarlo al meglio!

Starter kit con Arduino uno Rev3 e componenti

Starter kit con Arduino UNO REV3 contenente tutti i componenti utilizzati per realizzare gli esperimenti descritti nel libro L'ABC di ARDUINO.



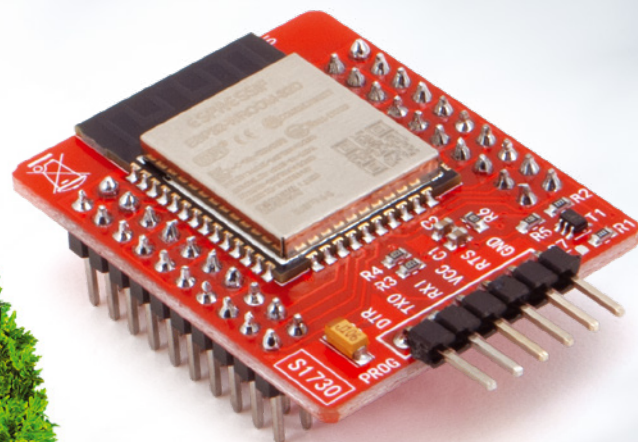
€ 58,00

cod. ARDUKITABC



MODULO ESP32

LOW-POWER



di **BORIS LANDONI**

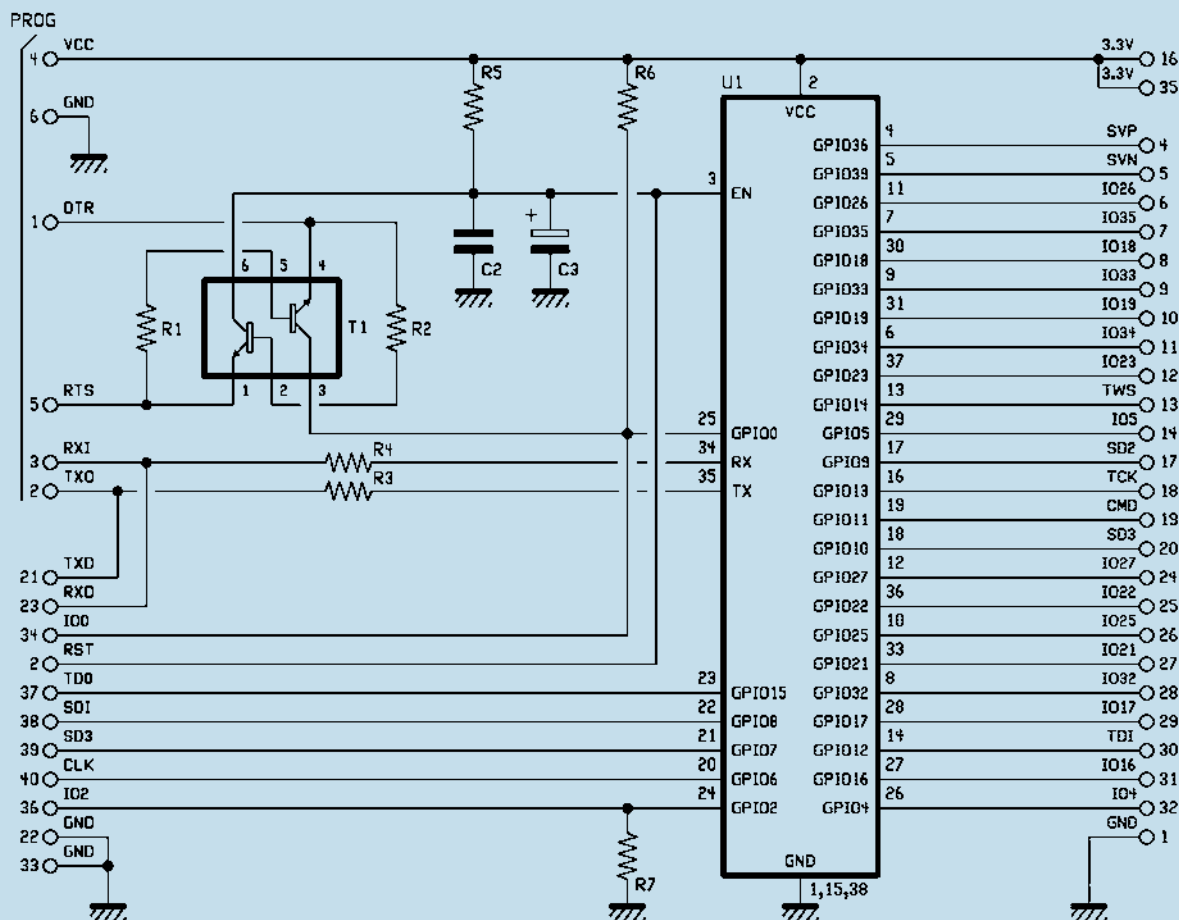
Il meglio dei moduli con SoC Espressif senza il consumo che implicherebbe: scoprite come abbiamo fatto.

S

i è conquistato un posto in prima fila nella progettazione di dispositivi connessi e la sua fama se la merita tutta: parliamo dell'ESP32, un SoC nato in casa Espressif che in poco tempo è diventato uno dei dispositivi preferiti per la prototipazione e la produzione di apparati connessi in wireless tramite

link WiFi e Bluetooth. In commercio si trovano svariati moduli basati su ESP32, di varie dimensioni e con dotazioni più o meno ricche, tuttavia nel panorama si sentiva la mancanza di una versione per applicazioni a basso consumo.

Per questa ragione ci siamo messi all'opera per realizzare un modulo che mantenesse la ricca dotazione di funzionalità e tutto l'hardware che il componente porta con sé, però capace di supportare applicazioni low-power.



Come prima cosa ci siamo chiesti dove andare a intervenire, rispetto agli schemi proposti dai produttori dei moduli già esistenti in commercio, per ridurre il consumo di elettricità. La risposta che ci siamo dati per prima è stata: sull'interfaccia USB, che infatti abbiamo tolto, giacché serve di fatto soltanto per la programmazione dell'ESP32 e che nelle applicazioni è ridondante.

Per programmare la scheda utilizzeremo quindi un'interfaccia USB to UART come quella reperibile presso Futura Elettronica con il codice prodotto 1686-USBTUART (on-line su www.futurashop.it) perché offre la possibilità di selezionare la gestione dei criteri CTS o RTS (noi usiamo l'RTS) dello standard RS232-C; meglio l'IDE Arduino gestendo opportunamente RTS e DTR va a "pilotare" il GPIO0 e EN per avviare la comunicazione durante la programmazione.

Ne risulta un modulo ESP32 low-power compatibile al 100% con la versione commerciale MINI ESP32 (cod. 8300-YB555MINI) ma che non montando la parte USB che assorbe potenza è ideale

per applicazioni a basso consumo energetico. Questa scheda è compatibile con l'IDE di Arduino, basata sull'ESP32 (ESP-WROOM-32) e dotata di un transceiver WiFi 802.11 b/g/n, nonché di interfaccia Bluetooth dual-mode (classico e BLE); sul link WiFi supporta una velocità di trasmissione dati fino a 150 Mbps, una potenza di uscita sull'antenna di 20,5 dBm per garantire la massima portata e un consumo in modalità sleep di soli 5 μ A.

La scheda dispone inoltre di interfacce per sensori di temperatura, touch sensor, SD-card, UART, SPI, SDIO, I²C, LED PWM, Motor PWM, I²S, IR, nonché del connettore micro USB standard e di un pulsante di reset.

QUALCOSA SULL'ESP32

Il componente chiave del progetto è un System On Chip nato per interfacciare tramite link wireless a basso consumo e ridotta portata, all'interno di PAN (Personal Area Network) hardware di vario genere, disponendo già a bordo di interfacce per sensori e attuatori, ADC e DAC, ma anche dei principali bus

ed interfacce di comunicazione utilizzati per comunicare con i moderni sensori digitali, con memorie, CODEC e dispositivi audio in generale.

Il successo del SOC della Espressif sta tutto qui, anzi no, perché a decretarlo è stata una progettazione che consente la programmazione e il debug del firmare attraverso lo stesso ambiente integrato di sviluppo (IDE) di Arduino, il che significa portabilità degli sketch e delle librerie, numerosissimi e facili da trovare nei forum e nella community Arduino.

Il core dell'ESP32 è un microprocessore dual-core Xtensa LX6 a 32 bit con capacità di calcolo fino a 600 DMIPS, dotato di proprie SRAM e ROM, della capacità rispettivamente di 520 kB e 448 kB, oltre a una memoria Flash da 4 MB e un RTC incorporato dotato di una sua RAM statica da 16 kB

Questo hardware è contornato da un controller Bluetooth in banda base compatibile BLE, un transceiver WiFi con MAC integrato, una sezione radio rice-trasmittente con antenna integrata su PCB (così da non incorrere in problematiche di accordo) oltre a BalUn (Balancer/Unbalancer) e commutatore d'antenna per i due link wireless.

Operando in wireless ed essendo stato pensato per le applicazioni moderne, l'ESP32 integra anche un acceleratore hardware per la gestione della crittografia, secondo i protocolli più utilizzati e sicuri come AES, SHA, RSA ed RNG.

Non gli mancano un sensore di temperatura digitale integrato, un gestore di quattro ingressi touch-sensor, oltre che l'hardware per i principali protocolli di comunicazione on-board (I²C-Bus, SPI, I²S) e per quelli esterni come CAN-Bus ed ethernet, come si vede dallo schema a blocchi dell'ESP32 proposto nella **Fig. 1**.

Grazie alle caratteristiche intrinseche implementate dalla Espressif, l'ESP32 può essere fatto

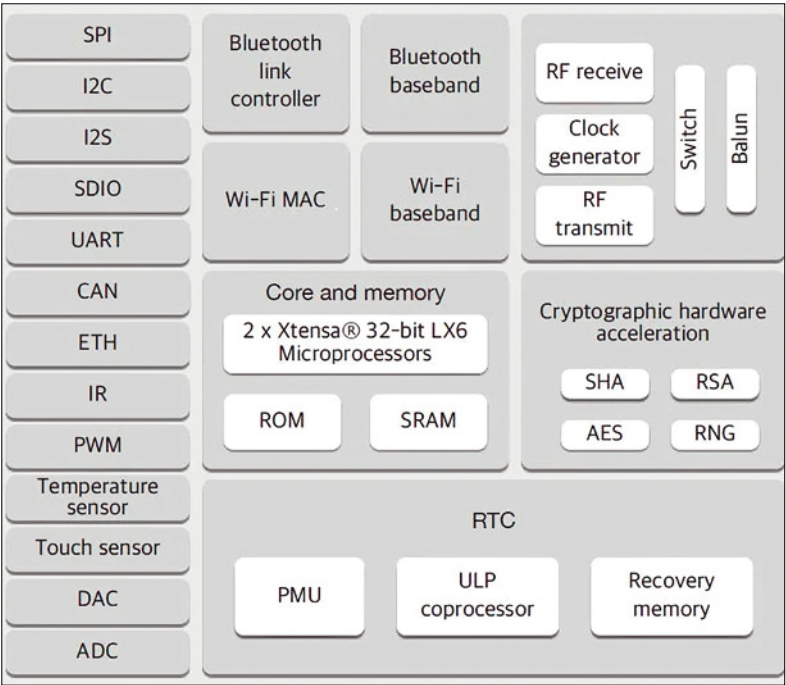


Fig. 1
Schema a blocchi del SoC Espressif ESP-32 WROOM.

funzionare in varie modalità di risparmio energetico andando a disattivare specifiche unità di quelle proposte dallo schema a blocchi, ma anche ponendo le CPU del core in modo low-power; verrebbe quindi da chiedersi perché per ottenere un modulo a basso consumo siamo andati a togliere il converter USB con la relativa elettronica.

Ebbene, la ragione è che in un modulo ESP32 anche essenziale, di quelli reperibili in commercio, quella parte è esterna al SoC ESP32, pertanto non può essere gestita dal firmware per essere disattivata quando serve il minimo consumo di energia. Togliendola, come vedrete dallo schema elettrico, a consumare resta praticamente il solo ESP32 e

Power mode	Description	Power consumption
Active (RF working)	Wi-Fi Tx packet 14 dBm ~ 19.5 dBm	Please refer to Table 10 for details.
	Wi-Fi / BT Tx packet 0 dBm	
	Wi-Fi / BT Rx and listening	
Modem-sleep	The CPU is powered on.	Max speed 240 MHz: 30 mA ~ 50 mA
		Normal speed 80 MHz: 20 mA ~ 25 mA
		Slow speed 2 MHz: 2 mA ~ 4 mA
Light-sleep	-	0.8 mA
Deep-sleep	The ULP co-processor is powered on.	150 μA
	ULP sensor-monitored pattern	100 μA @1% duty
	RTC timer + RTC memory	10 μA
Hibernation	RTC timer only	5 μA
Power off	CHIP_PU is set to low level, the chip is powered off	0.1 μA

Tabella 1
Modalità low-power supportate dall'ESP32.

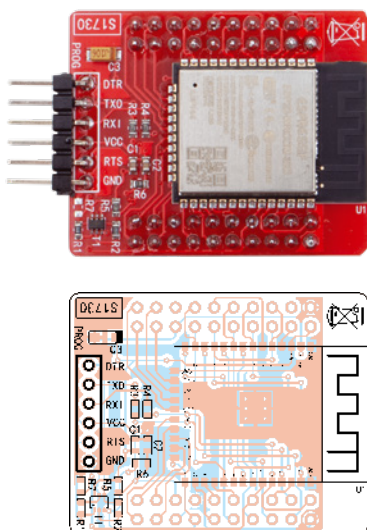


Elenco Componenti:

R3, R4: 22 ohm (0603)
 R1, R2: 2,2 kohm (0603)
 R5, R6: 10 kohm (0603)
 R7 -
 C1, C2: 100 μ F ceramico (0603)
 C3: 10 μ F 6.3V tantalum (A)
 T1: SBC846BDW1T1G
 U1: Modulo ESP32

Varie:

- Pin strip maschio 2x10 passo 2.54mm (2 pz.)
- Pin strip maschio 6 vie
- Circuito Stampato S1730 (40x32 mm)



in quello abbiamo tutti gli strumenti firmware per gestirci a piacimento il consumo energetico.

Far funzionare l'ESP32 in modalità attiva con le batterie non è l'ideale, poiché l'energia delle batterie si scaricherà molto rapidamente.

Le varie modalità di funzionamento a basso consumo sono dettagliate nella **Tabella 1**, che espone l'assorbimento di corrente in base alle modalità scelte e alle periferiche attive.

Attivando la modalità Deep Sleep implementata nell'ESP32 è possibile sospendere le attività che consumano più energia durante il funzionamento, ma lasciare in esecuzione quel poco che basta a riattivare il microprocessore quando si verifica un evento che lo richiede; a riguardo notate che si può utilizzare il timer, risvegliando l'ESP32 utilizzando periodi di tempo predefiniti, attraverso gli ingressi touch, eventi esterni o il coprocessore ULP.

In modalità Deep Sleep vengono disattivati sia le

CPU che il WiFi (ed anche il Bluetooth) ma il co-processore Ultra Low Power (ULP) può comunque essere acceso.

Mentre l'ESP32 è in modalità Deep Sleep rimane alimentata anche la memoria RTC, quindi è possibile scrivere un programma per il coprocessore ULP e memorizzarlo nella memoria RTC per accedere a dispositivi periferici, timer interni e sensori interni.

Questa modalità di funzionamento è utile se è necessario riattivare la CPU principale tramite un evento esterno, un timer o entrambi, mantenendo un consumo energetico minimo.

Durante il Deep Sleep, alcuni pin dell'ESP32 possono essere utilizzati dal coprocessore ULP, ovvero i pin RTC_GPIO e i pin Touch; il dettaglio è disponibile nel datasheet del SoC, che espone sia i pin interessati, sia le condizioni di utilizzo durante le varie modalità di esercizio del SoC. Può essere anche utile fare riferimento alla **Fig. 2**.

SCHEMA ELETTRICO

Andiamo dunque a vedere, attraverso l'analisi circuitale, come è composto l'hardware del nostro modulo ESP, che è sostanzialmente un piccolo PCB sul quale è saldato l'ESP32 insieme a un dual-transistor NPN-NPN e ai pochi passivi che servono, con tutte le connessioni riportate sui lati come è consuetudine dei moduli del genere.

Il cuore è, come si può immaginare, U1, che è il nostro ESP32-WROOM, del quale attraverso le piazzole da 1 a 40 (corrispondenti ad altrettanti pin-strip da montare sui lati della basetta) portiamo all'esterno tutti i GPIO, alcuni dei quali hanno più funzioni che si attivano in base alle impostazioni firmware.

I contatti 21 e 23 (rispettivamente TXD ed RXD) sono ripetuti e risultano in comune con il connettore PROG, destinato alla programmazione della scheda attraverso una connessione seriale TTL ed eventualmente un'interfaccia USB to UART dalla quale sia possibile gestire RTS e DTR; tali linee, ossia TXD ed RXD, nel normale utilizzo possono essere impostate come I/O, mentre in programmazione svolgono la funzione prevista dalla comunicazione seriale con l'ESP32.

Nella **Fig. 2** è proposto il dettaglio di tutti i GPIO e le linee di controllo della nostra scheda ESP32, con la premessa che molti dei pin possono essere configurati per svolgere alcune funzioni in alternativa ad altre, ovvero insieme.

La programmazione attraverso un convertitore USB/TTL (come il modello 1686-USBTOUART)

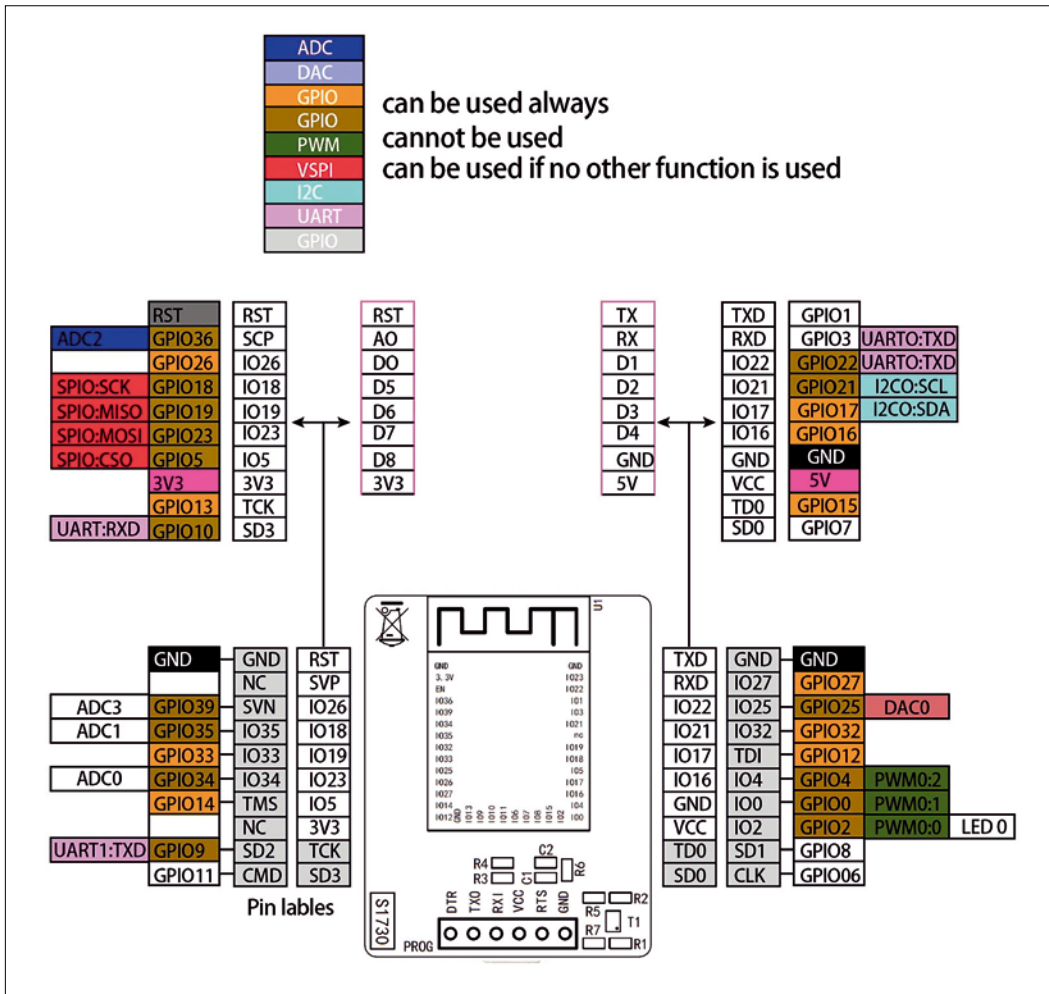


Fig. 2
Pin-out del
modulo
ESP32 da noi
proposto.

permette il facile caricamento del firmware. L'USB/TTL agisce come un ponte tra i segnali USB del computer e i segnali seriali TTL compatibili con l'ESP32. Questa operazione sfrutta principalmente due segnali di controllo - DTR (Data Terminal Ready) e RTS (Request To Send) - per automatizzare il processo di messa in programmazione del modulo.

Il connettore PROG sul nostro circuito è il punto di ingresso per il convertitore USB/TTL. Questo connettore fornisce i segnali seriali TX e RX per la trasmissione dati e i segnali DTR e RTS per il controllo del flusso e dello stato del modulo. Quando il convertitore è collegato al PC, viene creata una porta seriale virtuale attraverso la quale



Fig. 3
Il modulo
ESP32 da
noi proposto
a confronto
con quello
standard.

il modulo riceve i dati.

Ma per iniziare la programmazione, l'ESP32 deve essere posto in una modalità speciale che rende possibile scrivere nella sua memoria flash.

Questo è ottenuto manipolando i suoi pin EN (Enable) e GPIO0.

Normalmente, il pin EN deve essere alto (connesso a VCC) per operare, mentre il pin GPIO0 deve essere basso (connesso a GND) per entrare in modalità di programmazione.

Per questo motivo il nostro modulo è dotato di un doppio transistor NPN T1, collegato in modo tale che i segnali DTR e RTS possano controllare questi due pin. Il transistor funge da interruttore elettronico che, quando attivato, può connettere i pin al GND, cambiando il loro stato.

Vediamo come. L'IDE Arduino (o il software utilizzato per la programmazione) inizia il processo abbassando il segnale DTR. Questo, a sua volta, attiva T1 che collega il pin EN a RTS che in questa fase si trova a GND, resettando il modulo. Con il modulo in reset e GPIO0 basso, il segnale DTR viene rilasciato, facendo risalire EN e avviando l'ESP32 mentre GPIO0 rimane basso. Questo fa entrare l'ESP32 in modalità di programmazione. A questo punto, il firmware può essere inviato attraverso i pin RXD e TXD.

Dopo che il firmware è stato caricato, il segnale RTS viene rilasciato, permettendo a GPIO0 di risalire attraverso la resistenza di pull-up. Un ulteriore impulso di reset viene inviato tramite DTR per avviare il modulo con il nuovo firmware.

Tutto il processo di programmazione è automatizzato da software, rendendo il processo semplice e trasparente per l'utente. La gestione automatica dei segnali DTR e RTS rimuove la necessità di mettere manualmente il modulo in modalità di programmazione, accelerando e semplificando lo

sviluppo.

Notate anche che inizialmente, vale a dire all'accensione, il piedino EN viene portato per un istante a zero logico attraverso il condensatore C2, che nell'istante di applicazione dell'alimentazione è scarico e che in pochi istanti si carica attraverso la resistenza R2, collegata al positivo di alimentazione generale a 3,3V. Il pin EN ha anche la funzione di reset esterno, da utilizzare durante il normale funzionamento della scheda per riavviare il programma caricato nella Flash EPROM del modulo ESP32.

L'intero circuito prevede un'alimentazione a tensione continua di 3,3V ed assorbe una corrente intorno ai 600 mA, che dipende dalle periferiche utilizzate e dalle interfacce wireless in uso; tale corrente è la massima richiesta, ma chiaramente impostando la modalità a basso consumo si riduce, soprattutto quando il circuito viene posto nella modalità sleep, dove vengono spente tutte le periferiche non indispensabili.

L'INTERFACCIAMENTO AL PC

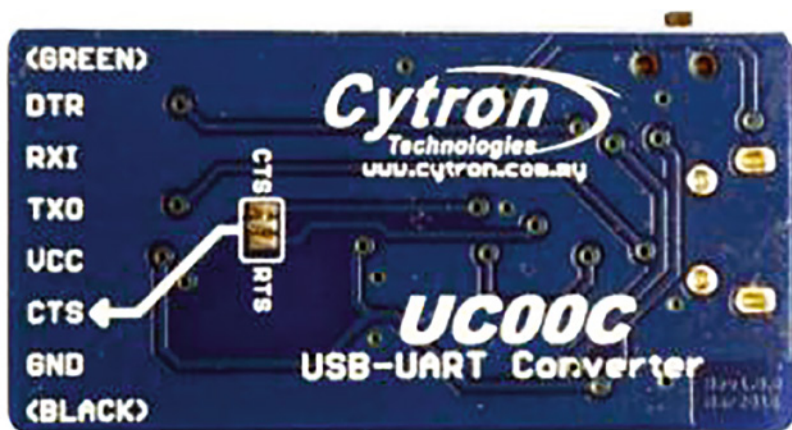
Come accennato, il nostro modulo ESP low-power è predisposto per l'interfacciamento al Personal Computer tramite una porta di comunicazione seriale a livelli 0/3,3V utilizzando le linee RXD e TXD dell'UART interno; siccome i moderni PC hanno solo la USB come forma di interfacciamento seriale, per implementare la comunicazione ai fini della programmazione e del debug firmware occorre interporre un converter USB to UART operante con segnali lato seriale non a 5V ma a 3,3 volt, pur secondo il protocollo RS232-C.

Nel nostro caso abbiamo adottato il modulo 1686-USBTOUART, prodotto dalla Cytron e commercializzato dalla Futura Elettronica, perché dispone di un'interfaccia USB che nei Personal Computer con sistema operativo Windows, previa installazione dei driver forniti, viene associata a una porta COM virtuale e può quindi essere gestita nell'IDE Arduino con facilità, semplicemente impartendo, dal menù Strumenti, il comando Porta Seriale, quindi scegliendo nel sottomenù la porta creata di driver stessi.

Altra caratteristica di questo converter è che sul lato seriale permette sia di gestire RTS, CTS e DTR, sia di impostare i livelli logici scegliendo fra 3,3V e 5 V; nel nostro caso bisognerà spostare il microswitch a bordo scheda nella posizione 3,3V, dato che il SoC ESP32 opera con livelli logici del tipo 0/3,3 volt.

Come mostra la **Fig. 3**, che propone la faccia infe-

Fig. 3
Il jumper con cui selezionare la gestione di CTS o RTS, posto sulla faccia inferiore del modulo USB/TTL.



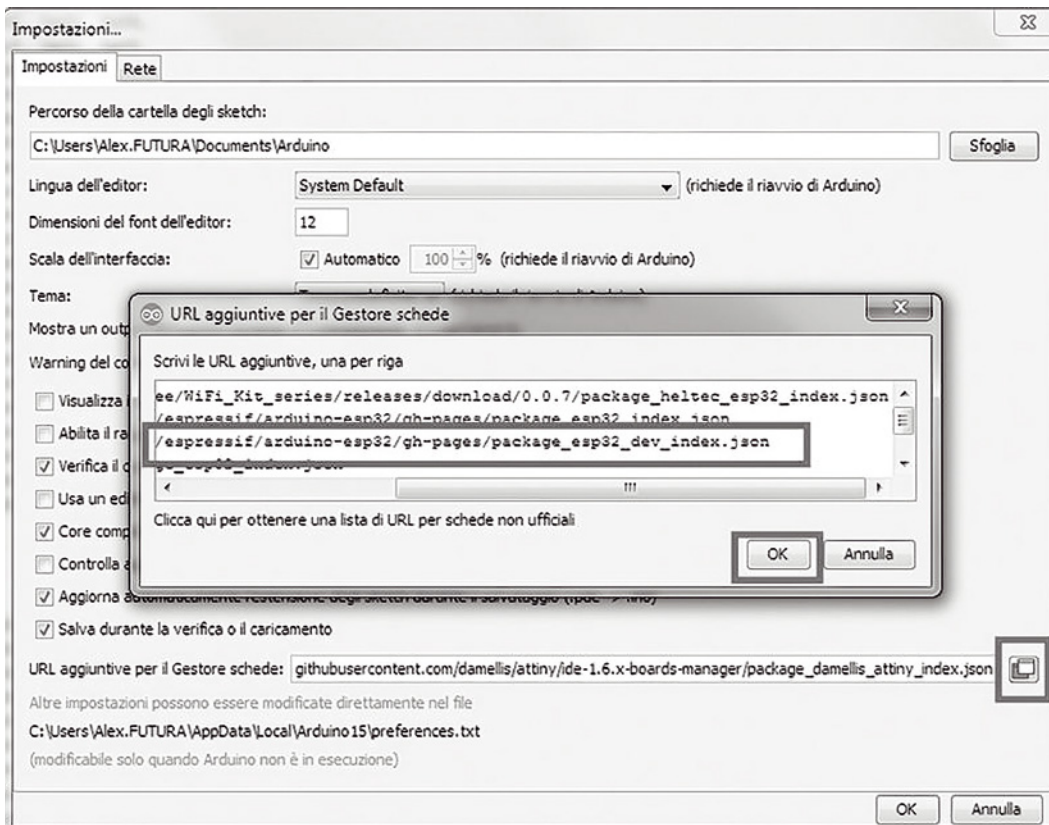


Fig. 4
Aggiunta dei percorsi on-line dove l'IDE reperirà i driver della scheda ESP32.

riore del modulo converter, un jumper ci permette di scegliere se portare all'uscita seriale e quindi gestire, opzionalmente il CTS (Clear To send) che è il segnale con cui il dispositivo di comunicazione risponde al RTS, ovvero quest'ultimo; nel nostro caso, dato che durante la programmazione il

bootloader dell'ESP32 e l'IDE Arduino possono gestire l'RTS, occorre chiudere il jumper su RTS, altrimenti la programmazione non potrà avviarsi.

CONFIGURARE LA SCHEDA NELL'IDE

La scheda che vi proponiamo di realizzare è a

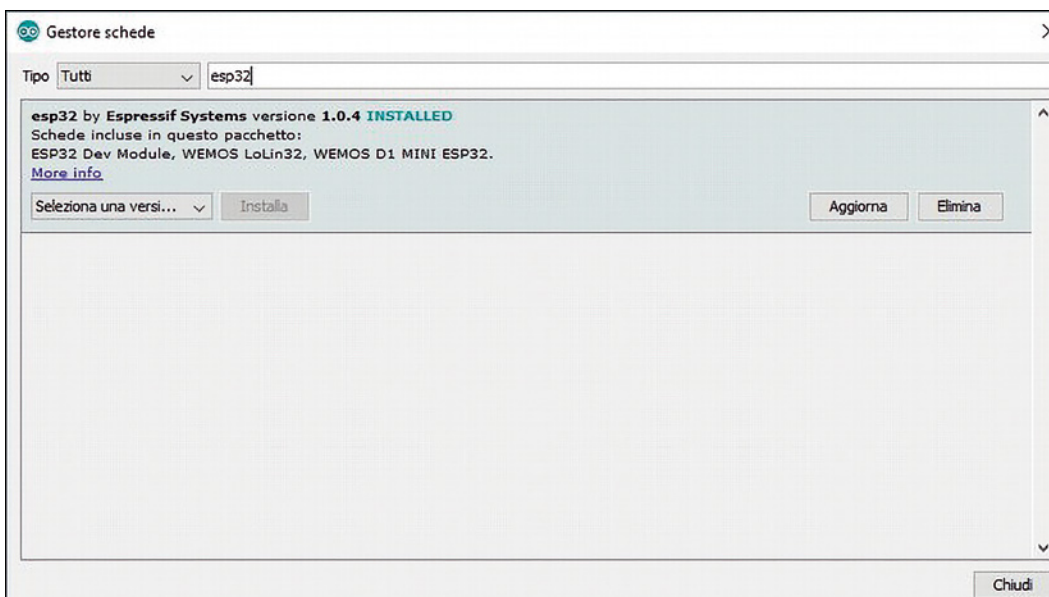
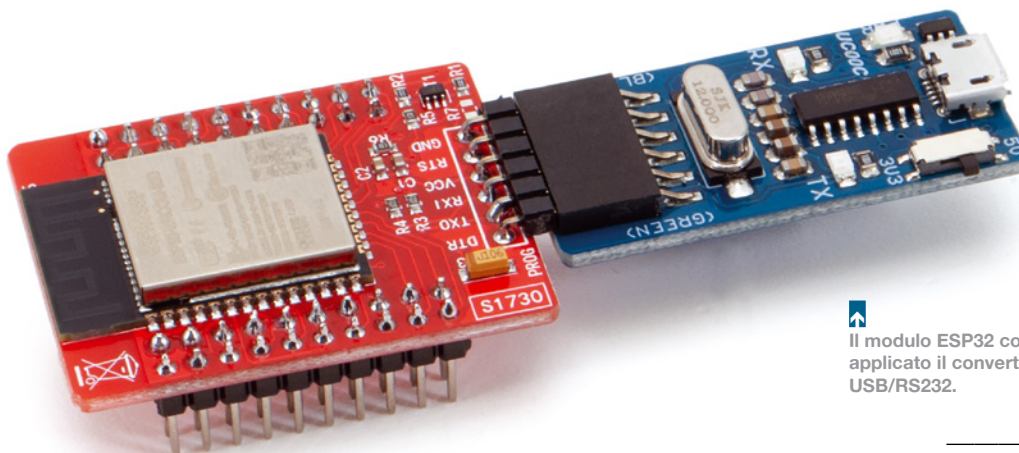


Fig. 5
Installazione del pacchetto software per la board ESP.



Il modulo ESP32 con applicato il converter USB/RS232.

tutti gli effetti una board ESP32, quindi per poter caricare il codice e gestirla tramite l'IDE Arduino (raccomandiamo di lavorare con una versione che vada dalla 1.6.0 in poi) dovranno essere eseguite alcune operazioni preliminari così da installare le librerie e caricare i driver di terze parti della board stessa, che altrimenti non sarà presente nel menù Strumenti.

Se non l'avete già fatto, nel PC dovete avviare l'IDE Arduino, quindi dal comando di menù "File > Impostazioni" accedere alla finestra di dialogo corrispondente e nella casella di testo "URL aggiuntive..." cliccare sul pulsante in fondo alla casella stessa (quello evidenziato nella cornice in rosso nella **Fig. 4**) e inserire in una nuova riga quanto di seguito indicato:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json

Fatto questo, dovete fare clic sul pulsante "OK" per confermare la configurazione.

Fatto ciò bisogna installare il pacchetto software ESP nell'IDE Arduino: allo scopo, dalla finestra di

dialogo cui si accede impartendo il comando menù "Strumenti > Scheda > Gestore schede", cercate "ESP32" e attendete che appaiano i risultati, quindi selezionate il risultato "ESP32 by Espressif Systems" in versione 2.0.7 o successiva, cliccando infine sul pulsante "Installa" per procedere con l'installazione dei driver della scheda (**Fig. 5**). Al termine dell'operazione, se avvenuta con successo la relativa sezione della finestra di dialogo mostrerà il messaggio "Installed".

Ad installazione terminata potete cliccare sul pulsante "Chiudi", quindi verificare che l'installazione sia avvenuta correttamente; allo scopo, basta verificare che nella finestra di dialogo cui si accede con il comando di menù Strumenti > Scheda siano presenti ulteriori board ESP.

È ora il momento di scegliere la scheda ESP32 che l'IDE considererà come target per le operazioni di programmazione: dal menu "Strumenti > Scheda" selezionate "WEMOS D1 MINI ESP32" o "ESP32 Wrover Module".

CARICARE GLI SKETCH NEL MODULO

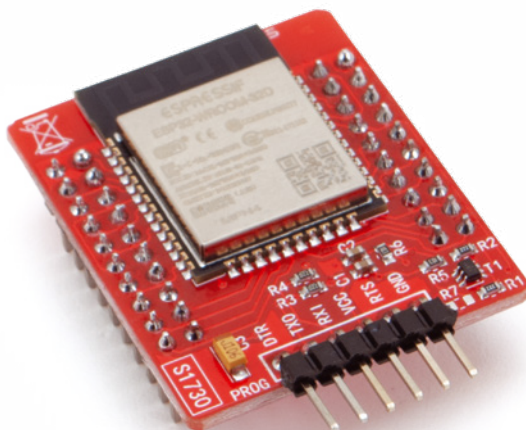
Per caricare il firmware nella nostra board è sufficiente aprire l'IDE, quindi selezionare il file sorgente (.ino) dall'elenco proposto, ovvero dalla cartella d'origine, cui si accede con il comando File>Apri. Il caricamento dello sketch si esegue come su qualsiasi scheda Arduino, ossia cliccando sul pulsante di programmazione che si trova nella toolbar in alto nell'interfaccia utente dell'IDE. Il software gestirà, come abbiamo visto, RTS e DTR per far partire la programmazione (**Fig. 6**).

REALIZZAZIONE PRATICA

Bene, a questo punto possiamo passare alla parte pratica del progetto e vedere come si realizza la nostra scheda ESP32: al solito abbiamo previsto il montaggio su circuito stampato a doppia faccia,



Il modulo ESP32 low power.




```
//*****
//***** Pin ESP per definizione SD-Card
*****

Caricamento completato
Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.2 seconds (effective 404.5 kbit/s)...
Hash of data verified.
Compressed 868512 bytes to 555230...
Writing at 0x00010000... (2 %)
Writing at 0x0001bf57... (5 %)
Writing at 0x000271e0... (8 %)
Writing at 0x000370c4... (11 %)
Writing at 0x0003ebf5... (14 %)
Writing at 0x00044871... (17 %)
Writing at 0x0004alc9... (20 %)
Writing at 0x0004f33f... (23 %)
Writing at 0x00054d74... (26 %)
Writing at 0x0005a7a2... (29 %)
Writing at 0x0005ff56... (32 %)
Writing at 0x00065771... (35 %)
Writing at 0x0006a7f5... (38 %)

23 ESP32S3 Dev Module su COM197
```

Fig. 6
Tipico Log
che indica
l'esecuzione
della pro-
grammazione
nei moduli
ESP32.

realizzabile scaricando i file Gerber dalla pagina di presentazione di questo numero, quindi utilizzando il servizio PCBPRODUCTION di Futura Elettronica. Ottenuto il PCB potete disporvi i pochi componenti richiesti, che sono tutti SMD, il che significa che la realizzazione implica la manualità e l'attrezzatura necessari nella gestione del montaggio superficiale.

Per riuscire nell'operazione conviene spalmare della pasta saldante nelle piazzole riservate ai componenti, specie l'ESP32, quindi procedere con un saldatore a punta molto sottile (lo stilo di una stazione saldante è l'ideale...) sui lati del modulo, badando di non unire con lo stagno i contatti attigui, quindi procedere con i pochi passivi e con il dual-transistor T1. In alternativa potete utilizzare una stazione saldante ad aria calda, dotata di ugello di piccolo diametro, dirigendo il getto ai lati del modulo ESP fin quando non vedrete brillare le saldature, poi saldare i componenti discreti con il saldatore a stilo. In tutti i casi vi sarà utile una lente d'ingrandimento per vedere bene mentre state saldando e una pinzetta per posizionare i passivi ed il T1 ben centrati nelle rispettive piazzole.

Un'altra opzione consiste nello spalmare della pasta flussante sulle piazzole prima del montaggio, quindi nello stagnarle tutte e poi procedere con il saldatore come già descritto.

A saldature terminate andate a verificare l'esattezza della realizzazione aiutandovi con il piano di montaggio proposto in queste pagine; il vostro modulo ESP ora è pronto per essere program-

mato utilizzando un'interfaccia esterna, che una volta caricato lo sketch andrà sconnessa per poi destinare il modulo all'applicazione target a basso consumo.

CONCLUSIONI

Bene, con questo possiamo ritenere conclusa la descrizione del progetto. In queste pagine vi abbiamo proposto una nuova scheda ESP32 che va ad arricchire il panorama delle board basate sull'omonimo SoC della Espressif, portando un hardware nel quale si va ad ampliare il concetto di low-power, sfruttando non solo le relative funzionalità di cui il componente dispone, ma andando anche a lavorare sull'hardware di contorno, che da firmware non può essere gestito, almeno riguardo al consumo energetico. ■



COSA OCCORRE ?

Codice	Descrizione	Prezzo € IVA inclusa
MINIESP32LP	Mini scheda Low Power con ESP32	12,90
USBTUART	Convertitore USB/TTL	8,00

IL MATERIALE VA RICHIESTO A:

Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 - www.futurashop.it



L'ENERGIA PULITA che ti fa risparmiare!

Pannelli solari monocristallini semirigidi 12V



Sono composti da singoli e sottilissimi strati di cristallo di silicio che opportunamente trattati aumenta la loro conduzione. Le celle che compongono il pannello monocristallino sono facilmente riconoscibili grazie al loro colore molto scuro. La purezza del silicio che compone le celle monocristalline permette di ottenere la massima efficienza energetica e grazie a questa caratteristica è possibile installare un numero di pannelli inferiore rispetto ai tradizionali, permettendo così di occupare superfici meno estese. L'alta qualità del materiale utilizzato permette di avere una durata maggiore del pannello.

Pannelli solari monocristallini rigidi 12V



Regolatori di carica per pannelli fotovoltaici a 12 e 24V



Prezzi IVA inclusa.

IDROBOT

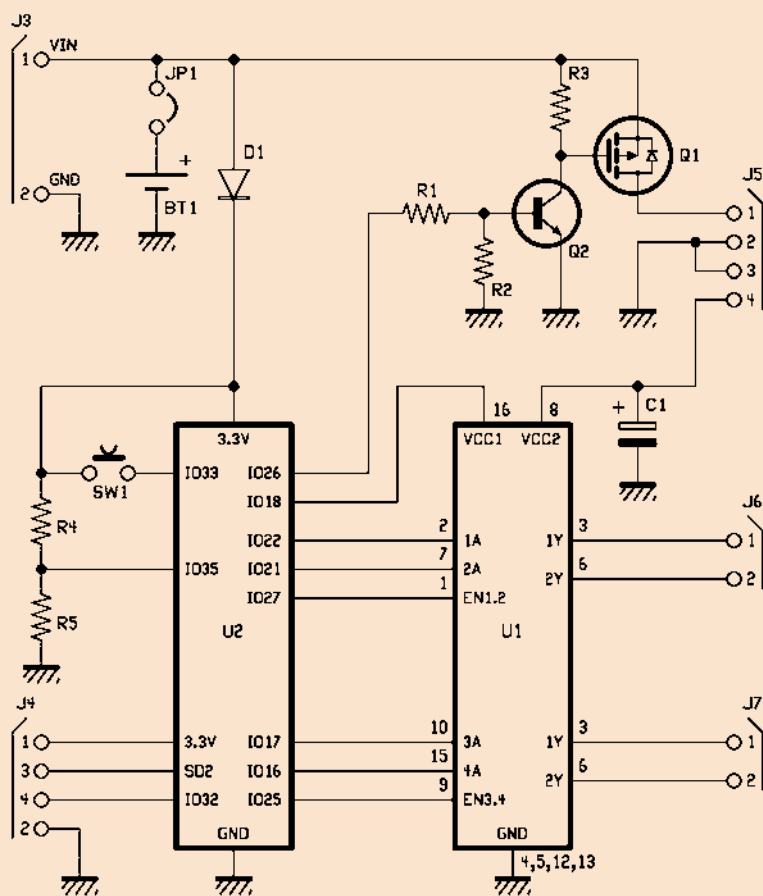
LA CENTRALINA DI IRRIGAZIONE SMART BLE E LOW-POWER

di LUCA BELLAN



Realizziamo una centralina di irrigazione low-power, alimentata a batteria e programmabile con smartphone tramite Bluetooth Low Energy

Chiunque abbia un giardino, o anche una serie di vasi dotati di irrigazione automatica, dispone quasi sicuramente di una centralina che, una volta programmata, è in grado di erogare acqua in momenti precisi della giornata, con una durata preimpostata. Sul mercato esistono diverse tipologie di centraline. Alcune sono piuttosto semplici ed economiche: per programmarle, è sufficiente regolare dei selettori per determinare la frequenza e la durata dell'irrigazione. Esistono poi modelli intermedi, dotati di pulsanti e display, che consentono di impostare programmi di irrigazione personalizzati per ciascun giorno della settimana. Infine, ci sono centraline più avanzate che possono essere programmate tramite un'applicazione su smartphone e che addirittura si integrano con i sistemi di domotica, dimostrand



do come la tecnologia possa facilitare la gestione delle risorse quotidiane.

Questo approccio tecnologico apre nuove possibilità per gli appassionati di elettronica e fai-da-te, i quali spesso cercano di implementare soluzioni innovative e personalizzate per risolvere problemi comuni, sfruttando le nozioni e le idee trovate su riviste o sul web. Ed è proprio per questo motivo che nel corso degli anni sulla nostra rivista sono stati presentati diversi progetti che riguardano i sistemi di irrigazione e la gestione di una delle risorse più preziose sul nostro pianeta, ovvero l'acqua.

Continuando in questa direzione, in questo articolo vi presentiamo una centralina di irrigazione capace di controllare due valvole con solenoidi bistabili, alimentata da una batteria ricaricabile 18650, predisposta per la connessione di sensori per la pioggia e l'umidità, programmabile tramite app Android e soprattutto dai consumi elettrici estremamente ridotti, grazie alle ottimizzazioni sia lato software che lato elettronica.

Prima di esplorare il dettaglio del nostro progetto, è utile comprendere i fondamenti degli impianti di

irrigazione. Tipicamente, tali sistemi si compongono di una fonte d'acqua (che può essere il collegamento alla rete idrica cittadina o un serbatoio autonomo) e di una o più elettrovalvole controllate dalla centralina. Queste valvole sono essenziali per regolare il flusso d'acqua nelle diverse linee, o 'zone', dell'impianto di irrigazione.

Esistono diverse tipologie di elettrovalvole, ciascuna con le proprie caratteristiche e applicazioni specifiche:

- **Elettrovalvole a motore DC:** queste elettrovalvole utilizzano un motore DC per controllare il flusso d'acqua, hanno una notevole affidabilità e consentono un controllo preciso sul flusso;
- **Elettrovalvole a solenoide:** le elettrovalvole a solenoide sono tra le più comuni nei sistemi di irrigazione; utilizzano un solenoide per aprire o chiudere il passaggio dell'acqua, sono semplici da installare e da utilizzare e sono disponibili in diverse dimensioni e portate;
- **Elettrovalvole a pistone:** sono caratterizzate da una costruzione robusta e sono adatte per applicazioni ad alta pressione; mediante il pistone possono gestire grandi portate d'acqua e sono spesso impiegate in sistemi di irrigazione agricola o industriale;
- **Elettrovalvole a membrana:** queste elettrovalvole utilizzano una membrana flessibile per controllare il flusso d'acqua e sono spesso utilizzate in applicazioni dove è richiesta una maggiore resistenza alla corrosione e all'usura;
- **Elettrovalvole a controllo di flusso:** queste elettrovalvole incorporano un regolatore di pressione o un riduttore di portata. Consentono una regolazione precisa del flusso d'acqua in base alle esigenze specifiche dell'impianto di irrigazione.

Ora che abbiamo visto le diverse tipologie di elettrovalvole, è facile comprendere perché la scelta della centralina dipende strettamente da loro. Una delle discriminanti per la scelta della centralina è senz'altro la disponibilità di alimentazione elettrica nel punto in cui si trovano le valvole: infatti molte centraline sono posizionate all'interno di pozzetti nel giardino o comunque in posti difficilmente raggiungibili da una linea di alimentazione fissa; in questo caso la scelta ricade sulle centraline alimentate a batteria che comandano valvole con solenoidi bistabili.

Grazie alla presenza di magneti permanenti, il solenoide bistabile (chiamato anche "latching solenoid") può rimanere in due stati stabili anche senza alimentazione, a differenza di un solenoide normale che

richiede un'alimentazione continua: questo consente un notevole risparmio energetico in quanto, il comando di apertura o di chiusura della valvola richiede un impulso elettrico di pochi millisecondi.

Il cuore della nostra centralina è un modulo ESP32 in formato "mini": il famoso microcontrollore a basso consumo energetico, dotato di funzionalità avanzate di connettività come Bluetooth e WiFi. Una delle tecniche di risparmio energetico più importanti offerte da ESP32 è il "deep sleep": quando la centralina non è attivamente coinvolta in processi di controllo o comunicazione, può entrare in questa modalità, riducendo significativamente il consumo energetico. Durante questa fase, la maggior parte dei componenti del modulo viene disattivata, consentendo di prolungare la durata della batteria.

Proprio per questo progetto è stato realizzato un modulo ESP32 (presentato in questo numero della rivista) che sfrutta al massimo la caratteristica low-power del controllore eliminando tutti gli elementi di maggior consumo e ottenendo così un assorbimento di corrente nell'ordine dei microampere (μA).

SCHEMA ELETTRICO

Il cuore del circuito è il modulo ESP32 alimentato da una batteria 18650 attraverso un jumper (JP1) che permette di isolare la batteria in caso di manutenzione alla scheda o di caricamento del firmware. La tensione della batteria (che a carica completa può arrivare a 4,2 Volt) danneggerebbe irrimediabilmente il modulo ESP32 se dovessimo alimentarlo direttamente, per questo abbiamo inserito il diodo D1 per ottenere un voltaggio compatibile con le specifiche di alimentazione del modulo (massimo 3,6 Volt); in questo caso, non è stato utilizzato un convertitore DC/DC step-down perché l'obiettivo primario è stato quello di massimizzare la durata della batteria.

Per raggiungere questo scopo, è stato essenziale minimizzare il consumo di corrente di ogni singolo componente del sistema.

Un partitore di tensione composto da R4 e R5, posizionato dopo il diodo, ci consente di misurare la carica della batteria grazie a un pin ADC: anche in questo caso le resistenze del partitore sono state dimensionate per fare fluire la minima quantità di corrente necessaria per la misura della tensione. SW1 serve per risvegliare il modulo ESP32 (normalmente in Deep Sleep) e farlo entrare in modalità connessione BLE per permettere la programmazione della centralina tramite app su

smartphone; questo pulsante è connesso al GPIO 33 che è anche un pin touch (in grado di rilevare il tocco umano).

Sul lato destro della scheda troviamo il connettore J3 che consente di alimentare la scheda utilizzando un alimentatore al posto della batteria, a patto che eroghi una tensione compresa tra i 3,6 e i 4,5 Volt. Sul connettore J4 troviamo i 3,3V generati dal modulo EPS32, GPIO 13, GPIO 32 (dotati delle funzionalità touch e ADC) e il GND del circuito. Questi pin non sono utilizzati nella versione attuale della centralina ma restano a disposizione per eventuali espansioni (ad esempio un sensore di pioggia o un sensore di umidità del terreno). Per elevare la tensione della batteria al voltaggio richiesto dal solenoide utilizziamo un convertitore DC/DC posizionato sulla strip J5; in commercio esistono differenti solenoidi bistabili che richiedono un voltaggio specifico come ad esempio 9, 12 o 24 Volt: andando a sostituire il convertitore DC/DC con uno adatto presente in catalogo, possiamo agevolmente adattare la nostra centralina di irrigazione qualora dovessimo cambiare le elettrovalvole. Nel nostro caso abbiamo utilizzato il convertitore 2846-DCDC12V in grado di elevare la tensione della batteria ai 12V (con una corrente di circa 280mA) necessari all'elettrovalvola da noi utilizzata.

Il componente L293D (posizionato in U1) è il popolare driver di motori DC pilotato da ESP32 che, grazie a due ponti H, ci consente di inviare impulsi a polarità invertita ai due solenoidi; nella variante che abbiamo selezionato questo componente include circuiti di protezione da sovracorrenti per evitare eventuali surriscaldamenti.

Gli impulsi ai solenoidi impongono l'erogazione di una quantità elevata di corrente in pochi millisecondi che il convertitore DC/DC fatica a fornire: per questo abbiamo inserito il condensatore C1 da 4700uF (come da specifica tecnica del solenoide da noi utilizzato). Il fatto che il condensatore sia in grado di sopportare una tensione massima di 25 Volt ci consente di utilizzare la maggior parte dei solenoidi di mercato (che operano ad un voltaggio massimo di 24 Volt) senza sostituire questo componente.

In questo progetto, ridurre il consumo di corrente per aumentare l'autonomia della batteria è fondamentale. A tale scopo, è stato sviluppato un circuito che sfrutta un paio di transistor, Q1 e Q2. Questo circuito è progettato per spegnere completamente il convertitore DC/DC quando non è in uso, eliminando così il suo consumo di energia.



Il processo inizia quando il modulo ESP32 attiva il transistor Q2, tramite il pin 26, portandolo in saturazione. Questo collega il collettore di Q2 al gate del mosfet Q1, il quale viene quindi portato a potenziale di terra, permettendo alla corrente di fluire dal source al drain.

Questa corrente alimenta il convertitore DC/DC, che carica il condensatore C1. Dopo il breve lasso di tempo necessario a caricare il condensatore, il modulo ESP32 imposta la polarità corretta sulle uscite dell'integrato L293D e invia l'impulso di attivazione al solenoide attraverso il pin ENABLE. Anche l'integrato L293D è un componente che consuma energia a riposo, per questo il pin VCC1 (alimentazione della parte logica) non è stato connesso ai 3,3 Volt dell'ESP32, bensì al pin 18 che fornisce tensione solamente quando l'integrato deve essere utilizzato.

Il circuito di accensione del convertitore, l'accensione controllata dell'integrato L293D e la realizzazione di un modulo ESP32 completamente low-power, uniti alla modalità software "deep

sleep" riducono al minimo i consumi di corrente consentendo alla centralina di irrigazione un funzionamento continuo per diversi mesi.

MONTAGGIO

Il PCB della centralina è stato disegnato in modo da essere inserito e fissato, tramite le quattro viti M3 agli angoli, nel contenitore in ABS 8220-G31 1MF (disponibile su www.futurashop.it) che è dotato di un grado di protezione dalla polvere e dall'acqua IP65/NEMA4 (grazie alla guarnizione in neoprene).

Per il suo assemblaggio facciamo riferimento al piano di montaggio e incominciamo a saldare sul PCB le cinque resistenze (da R1 a R5), il diodo D1, il pulsante SW1 (solo se vogliamo usarlo al posto del pin touch) e il jumper JP1.

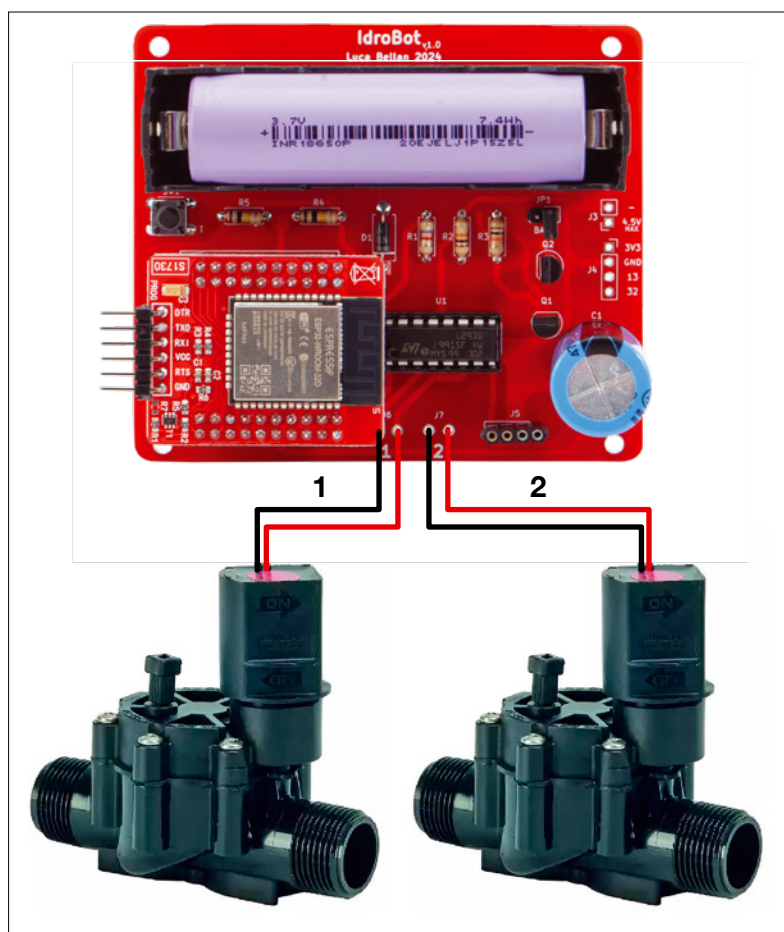
Proseguiamo con il transistor BC547 (Q2), il mosfet BS250 (Q1), lo zoccolo 8+8 per l'integrato L293D (U1) e per ultimi gli strip femmina con passo 2,54mm: 1x4 vie in J5, 2x10 sia in J1 che in J2. Terminiamo l'installazione dei componenti più ingombranti con il condensatore C1 e il porta batteria BT1 facendo estrema attenzione a rispettare le polarità.

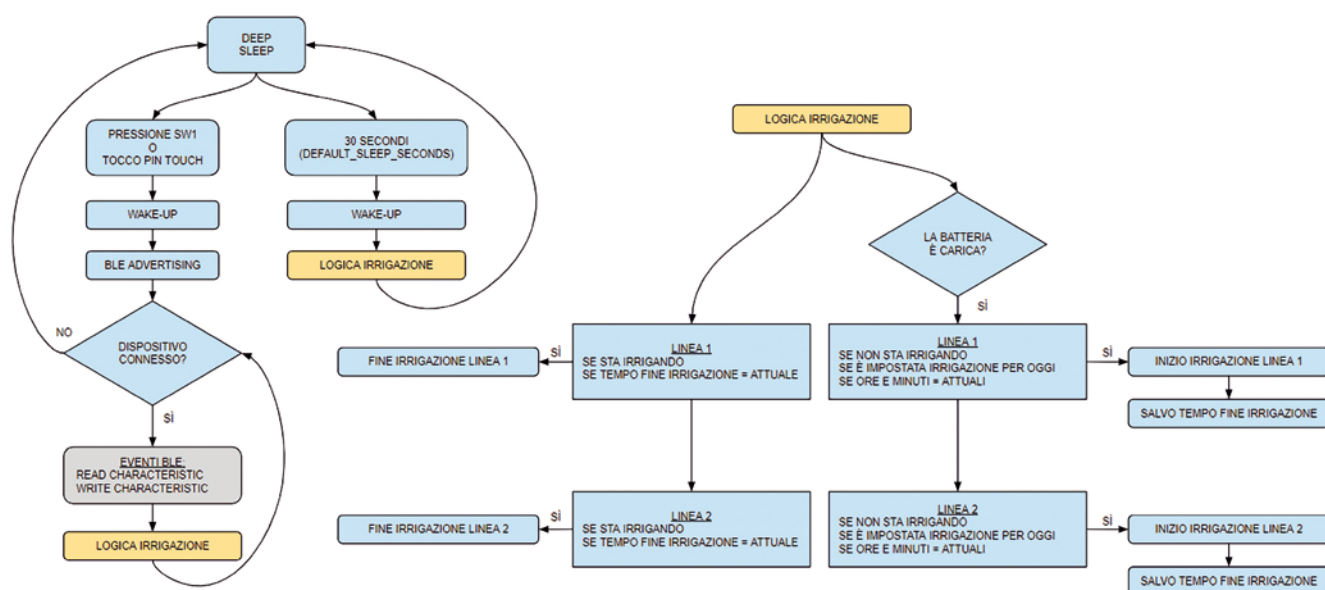
Nel caso in cui avessimo scelto di utilizzare il pin touch per fare entrare in modalità programmazione la centralina possiamo saldare un cavo sulla piazzola di SW1 contrassegnata dalla lettera "T" per poi farlo uscire dal contenitore. In questo modo, ci basterà toccare il cavo per attivare la modalità di programmazione della centralina, rendendo il processo più semplice e veloce rispetto a svitare tutte le volte il coperchio del contenitore per premere il pulsante.

Suggeriamo un ulteriore metodo che consente di utilizzare il pin touch senza portare fuori il cavo: saldiamo sempre il cavo alla piazzola del pulsante e facciamo in modo che tocchi, dall'interno, una delle quattro viti di fissaggio del coperchio; in questo modo ci basterà toccare la vite sul coperchio del contenitore per attivare la modalità di programmazione.

Per il collegamento tra la centralina e i solenoidi (**Fig. 1**), consigliamo di saldare quattro pezzi di cavo da 1mm² ai connettori J6 e J7. La lunghezza dei cavi dovrebbe essere tale da coprire la distanza tra la centralina e i solenoidi. Successivamente, passeremo questi cavi attraverso quattro piccoli fori che realizzeremo sul bordo del contenitore in plastica, all'altezza delle scritte "1" e "2" sul PCB. È importante sigillare i bordi dei fori con del silicone per evitare infiltrazioni d'acqua.

Fig 1
Collegamento tra scheda ed elettrovalvole.





Uniamo ora i cavi in uscita dal contenitore della centralina con quelli dei solenoidi utilizzando dei morsetti elettrici con gel: in questo modo, non sarà più necessario aprire la scatola della centralina e dissaldare i cavi dal PCB nel caso in cui dovessimo sostituire o semplicemente scollegare i solenoidi. Fissiamo il PCB (senza modulo ESP32) sul fondo del contenitore con le quattro viti M3 in dotazione e passiamo al caricamento e all'analisi del firmware.

FIRMWARE

Per prima cosa otteniamo il codice sorgente navigando all'indirizzo

<https://github.com/open-electronics/idrobot>, facciamo click sul pulsante verde "Code" e successivamente su "Download ZIP". Se il nostro PC dispone di GIT possiamo scaricare il sorgente anche tramite il comando `git clone` da terminale.

Apriamo il file `idrobot_v1.ino` con Arduino IDE (noi abbiamo utilizzato la versione 2.3.0) e assicuriamoci di aver installato le schede "esp32 by Espressif Systems" e la libreria "ESP32Time".

Passiamo ora ad analizzare il codice per capirne il funzionamento (possiamo aiutarci con lo schema in **Fig. 2**); dopo l'import di alcune librerie troviamo la sezione dei parametri di configurazione, nel dettaglio:

- **USE_TOUCH_PULSE_MS**: durata dell'impulso in millisecondi dato al solenoide; questo valore cambia a seconda del solenoide impiegato, anche se riteniamo che 100 sia un valore corretto per i diversi modelli che abbiamo testato.

- **DEFAULT_SLEEP_SECONDS**: intervallo di tempo in cui la nostra centralina "dorme" tra un controllo dell'irrigazione e l'altro, consigliamo di lasciare inalterato questo valore poichè diminuendolo andremo ad aumentare i consumi della batteria e settandolo ad un valore più alto di 30 secondi rischieremmo di saltare il minuto di irrigazione impostato durante la programmazione;
- **TOUCH_THRESHOLD**: è la soglia sopra cui il tocco sul pin touch viene rilevato, possiamo modificare leggermente questo valore nel caso in cui la centralina non andasse in modalità programmazione dopo aver toccato il pin touch;
- **DISCONNECT_AFTER_SEC**: intervallo di tempo in secondi dopo di cui l'ESP32 si disconnette automaticamente dallo smartphone e torna in modalità "deep sleep" per risparmiare la batteria;
- **CAPACITOR_CHARGE_MS**: intervallo di tempo in millisecondi utile a consentire il caricamento completo del condensatore prima di dare l'impulso di apertura o di chiusura al solenoide;
- **SOLENOID_IMPULSE_MS**: durata dell'impulso in millisecondi dato al solenoide; questo valore cambia a seconda del solenoide impiegato, anche se riteniamo che 100 sia un valore corretto per i diversi modelli che abbiamo testato.

Dopo aver configurato opportunamente la nostra centralina, avanzando nella lettura del codice, troviamo le definizioni dei pin utilizzati sul nostro PCB e gli UUID per i servizi e le caratteristiche del BLE. Nella tecnologia BLE (Bluetooth Low Energy) le caratteristiche sono attributi che definiscono i dati

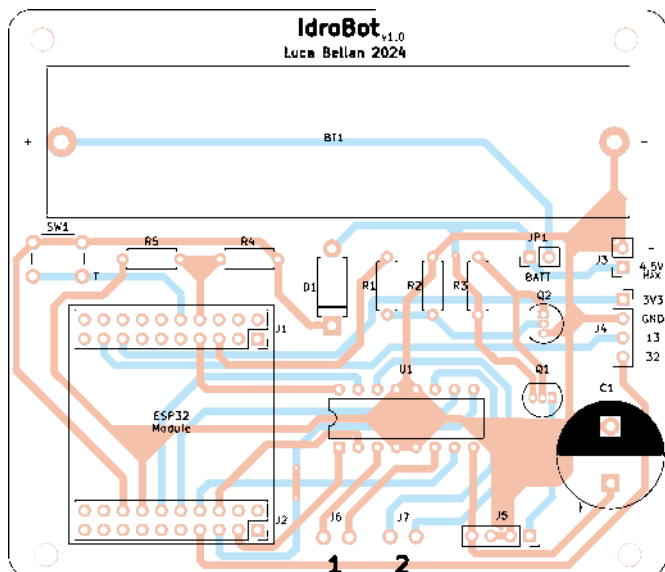
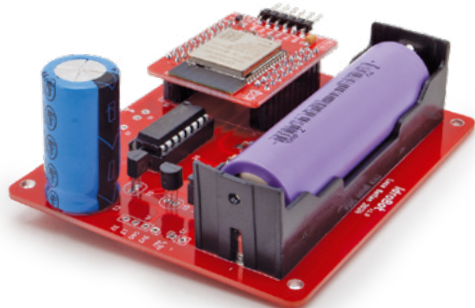
Fig. 2
Schema
della logica
software.

Elenco Componenti:

R1: 4.7 kohm
 R2, R3: 10 kohm
 R4, R5: 100 kohm
 D1: 1N4007
 C1: 4700 µF 16 VL elettrolitico
 Q1: BS250
 Q2: BC547
 U1: L293D (cod. L293D)
 U2: Breakout ESP32 (cod. MINIESP32LP)
 SW1: Microswitch (cod. KRS0610)

Varie:

- Zoccolo 8+8
- Jumper
- Strip Maschio 2 vie passo 2.54 mm
- Strip Femmina 10 vie passo 2.54 mm (2 pz.)
- Strip Femmina 4 vie passo 2.54 mm
- Porta batteria 1x18650 da C.S. (cod. 1X18650CS)
- Circuito Stampato S1745 (87x75 mm)



che possono essere scambiati con gli altri dispositivi; i servizi invece raggruppano logicamente le caratteristiche.

Per questo progetto abbiamo creato un servizio dedicato a tutte le caratteristiche (impostazioni) della linea 1, un servizio dedicato a tutte le caratteristiche della linea 2 e altri due servizi per impostare l'ora e per leggere la carica batteria.

I servizi e le caratteristiche devono essere identificati mediante un "indirizzo" univoco, ecco quindi spiegato l'utilizzo di UUID (Universally Unique Identifier) per ciascuno di questi elementi.

Come vedremo a breve, tramite la scrittura e

lettura dei dati sulle varie caratteristiche, l'app sarà in grado di impostare la programmazione della centralina.

L'ultima sezione delle dichiarazioni è rappresentata dalle variabili: normalmente i dati contenuti nelle variabili sono memorizzati in RAM, nel nostro caso ciò significa che tutti i valori andrebbero persi durante la modalità "deep sleep" in cui molti componenti di ESP32 vengono disalimentati per risparmiare energia: i core, la memoria, le periferiche e i modem (Bluetooth e WiFi).

Per ovviare a questo problema abbiamo anteposto "RTC_DATA_ATTR" alla dichiarazione delle variabili,

questa istruzione sta ad indicare al compilatore che quelle variabili non dovranno essere memorizzate in RAM, bensì nella memoria dell'RTC che rimarrà sempre attivo, anche durante il "deep sleep". Così facendo riusciremo a tenere memorizzate le informazioni di programmazione che provengono dall'app ma anche lo stato corrente di irrigazione delle due linee.

Nello specifico, per ciascuna linea, abbiamo dichiarato un array *waterDays* che indica per ogni giorno della settimana se è attiva l'irrigazione (valore 0/1), un array di struct *waterTime* che contiene le ore e minuti di partenza irrigazione per ciascun giorno della settimana e infine un array *waterDurations* che indica, sempre per ciascun giorno della settimana, la durata in minuti dell'irrigazione.

Proseguendo con l'analisi del codice troviamo le funzioni *callback* del BLEServer e delle BLECharacteristics: le prime indicano al sistema quando un dispositivo (l'app su smartphone) si è connesso o disconnesso, mentre la seconda riceve tutti i parametri di programmazione della centralina, li verifica e li inserisce nelle relative variabili.

Sono disponibili due caratteristiche per controllare manualmente l'apertura e la chiusura delle linee di irrigazione nel caso in cui dovessimo eseguire dei test sull'impianto.

L'ultima caratteristica che viene scritta dall'app serve per impostare l'RTC interno dell'ESP32 alla data e ora attuali; la sincronizzazione dell'RTC a bordo è estremamente importante per eseguire in modo corretto il programma di irrigazione. Tutta la logica di funzionamento della centralina è eseguita tramite la funzione *setup*: infatti, questa funzione non solo viene richiamata all'inizio del programma, ma anche ogni volta che l'ESP32 esce dalla modalità "deep sleep".

Per prima cosa vengono inizializzati e settati tutti i pin del convertitore DC/DC e di L293D a LOW per mantenerli disattivati, poi viene richiamata la funzione *RunIrrigation* che, come vedremo più avanti, gestisce la logica dell'irrigazione.

A questo punto viene verificata la causa del risveglio: infatti se il risveglio è dovuto alla pressione del pulsante o al tocco del pin touch (in base alla nostra configurazione), la centralina andrà in modalità programmazione, attivando la parte BLE e rendendosi disponibile alla connessione con l'app per 5 secondi; mentre se il risveglio è dato dal timer di controllo irrigazione (impostato con `DEFAULT_SLEEP_SECONDS`) l'ESP32, dopo aver eseguito le logiche di controllo dell'irrigazione, tornerà semplicemente in modalità "deep sleep"

richiamando la funzione *GoToSleep*.

La funzione *StartBLE* inizializza e configura tutti i servizi e le caratteristiche del BLE, associando a ciascuna di esse il valore della relativa variabile contenuta in memoria RTC: questa operazione sarà utile all'app per recuperare tutti i parametri di programmazione memorizzati nella centralina. Infine, viene attivato l'advertising che permette all'ESP32 di propagare, mediante la radio del Bluetooth, il proprio nome e altri dati utili agli altri dispositivi per connettersi.

La funzione *GoToSleep* imposta il timer per il risveglio del modulo ESP32, successivamente configura la modalità di wake-up aggiuntiva in base alla nostra configurazione, disattiva i modem Bluetooth e WiFi e, infine, attiva la modalità "deep sleep". Come abbiamo accennato precedentemente, durante questa modalità molti componenti vengono disalimentati; restano però attivi l'RTC (con la propria memoria) e il co-processore ULP (Ultra Low Power) che, attraverso dei pattern di monitoraggio a basso impatto energetico, è in grado di risvegliare il dispositivo in caso di variazioni allo stato dei pin.

La funzione *RunIrrigation* recupera la data e ora attuali dall'RTC (incluso il giorno della settimana corrente) e, successivamente, verifica per ciascuna linea le seguenti condizioni:

- Se non sta già irrigando e il giorno corrente deve irrigare e le ore/minuti coincidono con quelle correnti: attiva l'irrigazione della linea e salva in una variabile il tempo di fine irrigazione ottenuto sommando la durata di irrigazione al tempo corrente;

↓
Il nostro
prototipo
montato.



- Se la linea sta irrigando ed è impostato un tempo di fine irrigazione e questo tempo coincide con l'orario attuale: disattiva l'irrigazione della linea.

L'attivazione o la disattivazione di una linea viene effettuata tramite le funzioni *StartIrrigation* e *StopIrrigation*: la prima funzione attiva il convertitore DC/DC e il componente L293D, attende l'intervallo di tempo impostato per caricare il condensatore e configura le uscite del ponte-H per poi dare l'impulso con la polarità corretta al solenoide; infine disattiva sia L293D che il convertitore DC/DC.

La funzione *StopIrrigation* esegue i medesimi step dando però l'impulso al solenoide con la polarità invertita: in questo modo la valvola si chiude e il flusso d'acqua viene interrotto.

Ora che abbiamo configurato e analizzato il codice, colleghiamo il modulo ESP32 alla USB tramite il convertitore USB-seriale FTDI e carichiamo il firmware: a caricamento avvenuto potremo inserire l'ESP32 sul PCB e, subito dopo, la batteria 18650. Nel caso in cui dovessimo caricare altre volte il firmware ricordiamoci di isolare la batteria dal circuito rimuovendo il jumper JP1 prima di connettere il convertitore USB-seriale.

APP

Lo strumento che abbiamo utilizzato per creare l'app della nostra centralina è MIT App Inventor: un potente quanto semplice editor online, presentato in passato sulla pagine di questa rivista, che consente a tutti di creare la propria app per Android e recentemente anche per iOS (https://appinventor.mit.edu/ios_tips).

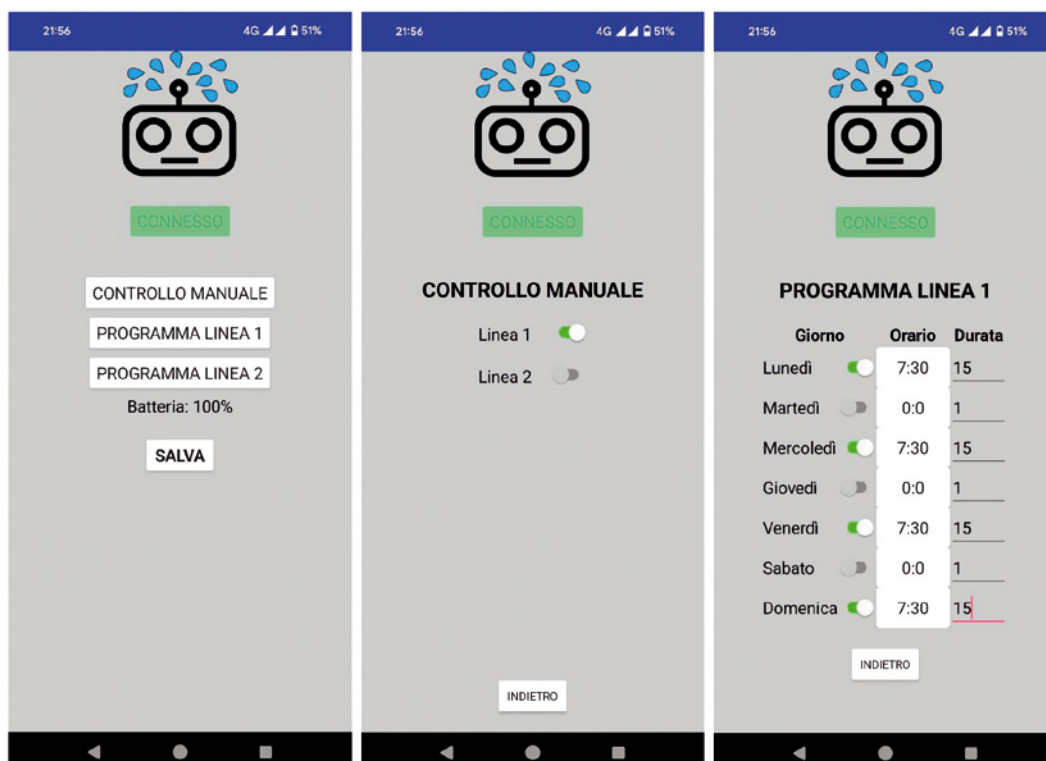
Per poter procedere è necessario registrarsi tramite il sito <https://appinventor.mit.edu> facendo click sul pulsante arancione "Create Apps!"; dopodichè, a login effettuato, mediante il menu "Projects" e la voce "Import project (.aja) from my computer..." carichiamo il file dell'app *IdroBot.aia* dalla cartella dei sorgenti che abbiamo scaricato all'inizio.

La schermata dell'editor si divide principalmente in due sezioni: Designer e Blocks; la prima sezione consente, in maniera visuale, la creazione dell'interfaccia grafica dell'applicazione mediante la collocazione di pulsanti, caselle di controllo, aree di testo, immagini e altri elementi.

La seconda sezione consente di programmare, tramite un editor a blocchi, la logica software di tutti i controlli posizionati sull'interfaccia grafica.

Il primo elemento che visualizziamo quando l'app viene avviata è il pulsante "CONNETTI": la pressione di questo pulsante avvia una scansione della

Fig 3
Schermate
dell'applicazione
per smartphone.



durata di 10 secondi in cui tutti i dispositivi BLE nei paraggi vengono trovati uno ad uno. Quando il dispositivo BLE individuato ha come nome "IdroBot" la ricerca si interrompe e avviene la connessione alla centralina di irrigazione.

A connessione effettuata, l'app va a leggere le caratteristiche BLE che contengono le informazioni sui giorni, gli orari e le durate dell'irrigazione di tutte e due le linee; inoltre viene letto anche il valore di carica della batteria. Queste informazioni sono poi riportate sull'interfaccia grafica nelle relative schermate. A questo punto, viene mostrato il menu di navigazione dove possiamo andare a controllare manualmente le valvole oppure pianificare le irrigazioni di ciascuna linea.

Nella schermata di controllo manuale abbiamo a disposizione due interruttori a leva: agendo su questi interruttori andremo a comunicare alla centralina di aprire o chiudere immediatamente le valvole. Le schermate di programmazione linea consentono invece, per ciascun giorno della settimana, di attivare l'irrigazione inserendo ora e minuti di inizio e la durata espressa in minuti: la durata di un'irrigazione non può mai essere inferiore a 1 minuto. Quando abbiamo impostato correttamente i programmi di irrigazione possiamo premere il pulsante "SALVA" nella schermata principale: tutti i parametri verranno scritti nelle relative caratteristiche BLE della centralina e l'app si disconetterà in automatico per permettere alla centralina di tornare in modalità "deep sleep". Consigliamo di limitare al minimo il tempo di connessione tra app e centralina in quanto la trasmissione BLE, essendo una comunicazione radio, risulta dispendiosa in termini di energia e andrebbe quindi ad influire negativamente sulla durata della batteria. Ora siamo pronti a compilare il file apk che ci servirà per installare l'app sul nostro smartphone (noi abbiamo utilizzato uno smartphone Android per comodità); andiamo quindi sul menu "Build" e facciamo click sulla voce "Android App (apk)". Dopo qualche secondo di attesa per la compilazione, potremo scaricare il file sul PC o direttamente sullo smartphone inquadrando il codice QR con la fotocamera. Per installare l'app tramite il file apk ricordiamoci di abilitare, nelle impostazioni di sicurezza del nostro smartphone, l'installazione di app da origini sconosciute. Ad installazione conclusa apriamo l'app, mandiamo in modalità programmazione la centralina e connettiamoci: eseguiamo quindi dei test di apertura e chiusura manuale sulle valvole per verificare che tutto funzioni correttamente. Nel caso in cui le valvole facessero fatica ad

aprirsi o a chiudersi possiamo agire, nel firmware della centralina, sui parametri CAPACITOR_CHAR-GE_MS e SOLENOID_IMPULSE_MS per andare a regolare i tempi di carica del condensatore e la durata dell'impulso dato al solenoide.

PERSONALIZZAZIONE

Giunti alla fine di questo articolo vorremmo darvi alcuni spunti e sfide per modificare, sia a livello hardware che software, il nostro progetto. Come abbiamo accennato all'inizio, i GPIO 12 e 32 sono a disposizione per future espansioni: ad esempio è possibile collegare un sensore di pioggia (come il sensore 7200-RN01 in catalogo Futura Elettronica) e modificare il firmware per disabilitare, fino ad alcuni giorni, l'irrigazione nel caso in cui sia rilevato un livello di pioggia sopra a una soglia predefinita.

Se invece volessimo controllare con precisione la quantità di acqua erogata in un vaso, potremmo inserire nel terreno un sensore di umidità (codice 2850-MOISTURESENSOR) e andare ad attivare l'irrigazione solamente nel momento in cui il livello di umidità scende sotto a una certa soglia.

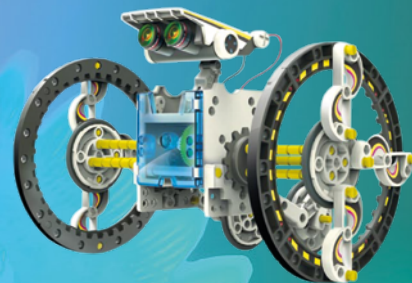
Modificando le strutture dati che vengono memorizzate nella memoria RTC e sviluppando ulteriormente l'app con MIT App Inventor sarà possibile aggiungere più irrigazioni schedate in una singola giornata. Non dimentichiamoci inoltre che il nostro ESP32 è dotato di connettività WiFi: possiamo collegarlo alla nostra rete di casa (dotando la centralina di un'alimentazione fissa, a causa dei consumi più elevati) e controllarla tramite un bot Telegram oppure con MQTT integrato ESPHome (<https://esphome.io>).

E ora non ci resta che augurarvi: buona irrigazione! 

 COSA OCCORRE ?		
Codice	Descrizione	Prezzo € IVA inclusa
MINIESP32LP	Mini scheda Low Power con ESP32	12,90
USBTUART	Convertitore USB/TTL	8,00
L293D	IC Driver per motori	0,70
1X18650CS	Portabatteria da C.S. per 1 pila tipo 18650	0,90
IL MATERIALE VA RICHIESTO A:		
Futura Elettronica, Via Adige 11, 21013 Gallarate (VA) Tel: 0331-799775 - www.futurashop.it		



SERVOMOTORI, INDISPENSABILI PER GLI APPASSIONATI DI ROBOTICA!



INGRANAGGI IN METALLO

cod. MC946R

€ 8,50



SERVO ANALOGICO AD ALTE PRESTAZIONI CON ROTAZIONE DI 180°

- Coppia di torsione: 10,5kg/cm (4,8V), 13kg/cm (6V)
- Velocità a 4,8V: 0,20sec/60° senza carico, velocità a 6V: 0,17sec/60° senza carico
- Alimentazione: da 4,8 a 7,2V
- Dimensioni: 40,7x19,7x42,9mm
- Peso: 55g

cod. SERVO213

€ 11,00



SERVO ANALOGICO CON ROTAZIONE DI 180°

- Coppia di torsione: 13kg/cm (4,8V) - 15kg/cm (6V)
- Velocità a 4,8V: 0,20sec/60° senza carico, velocità a 6V: 0,16sec/60° senza carico
- Alimentazione: da 4,8 a 6V
- Dimensioni: 40,6x39,5x20mm
- Peso: 55g

cod. MC996R

€ 9,00



SERVO ANALOGICO CON ROTAZIONE DI 360°

- Coppia di torsione: 9,4kg/cm (4,8V) - 11kg/cm (6V)
- Velocità a 4,8V: 0,19sec/60° senza carico, velocità a 6V: 0,15sec/60° senza carico
- Alimentazione: da 4,8 a 6V
- Dimensioni: 42,7x40,9x20mm
- Peso: 55g

cod. SERVO860M9C

€ 9,00



MINI SERVO ANALOGICO CON ROTAZIONE CONTINUA DI 360°

- Coppia di torsione: 1,8kg/cm (4,8V)
- Rapporto di riduzione: 320:1
- Velocità a 4,8V: 0,12sec/60° senza carico
- Alimentazione: da 4,8 a 6V
- Dimensioni: 22,6x32x12mm
- Peso: 12g

cod. SERVO09METAL

€ 8,00



MINI SERVO ANALOGICO CON ROTAZIONE DI 180°

- Coppia di torsione: 1,8kg/cm (4,8V)
- Rapporto di riduzione: 320:1
- Velocità a 4,8V: 0,12sec/60° senza carico
- Alimentazione: da 4,8 a 6V
- Dimensioni: 22,6x11,8x26,7mm
- Peso: 12g

INGRANAGGI IN PLASTICA

cod. SERVO214

€ 8,50



SERVO ANALOGICO CON ROTAZIONE DI 120°

- Coppia di torsione: 13kg/cm (4,8V) - 15kg/cm (6V)
- Velocità a 4,8V: 0,20sec/60° senza carico, velocità a 6V: 0,16sec/60° senza carico
- Alimentazione: da 4,8 a 6V
- Dimensioni: 54,2x39,5x20mm
- Peso: 55g

cod. SERVO860LC

€ 9,90



SERVO ANALOGICO CON ROTAZIONE CONTINUA DI 360°

- Coppia di torsione: 5,5kg/cm (4,8V)
- Velocità a 4,8V: 0,22sec/60° senza carico
- Alimentazione: da 4,8 a 6V
- Dimensioni: 40x39,1x20mm
- Peso: 38g

cod. SERVO209

€ 8,00



SERVO ANALOGICO CON ROTAZIONE DI 180°

- Coppia di torsione: 5,5kg/cm (4,8V) - 6,5kg/cm (6V)
- Velocità a 4,8V: 0,19sec/60° senza carico, velocità a 6V: 0,15sec/60° senza carico
- Alimentazione: da 4,8 a 6V
- Dimensioni: 40,6x39,5x20mm
- Peso: 42,5g

cod. SERVO207

€ 9,90



MINI SERVO ANALOGICO CON ROTAZIONE DI 180°

- Coppia di torsione: 2,3kg/cm
- Velocità: 0,18 sec/60°
- Alimentazione: da 4,8 a 6V
- Dimensioni: 28,5x13,3x30,6mm
- Peso: 17g

cod. SERVO206

€ 7,50



MINI SERVO ANALOGICO CON ROTAZIONE DI 180°

- Coppia di torsione: 1,2kg/cm (4,8V) - 1,5kg/cm (6V)
- Velocità a 4,8V: 0,12sec/60°, velocità a 6V: 0,11sec/60°
- Alimentazione: da 4,8 a 6V
- Dimensioni: 22,5x12x29mm
- Peso: 9g

GIOCHIAMO CON GLI ENCODER

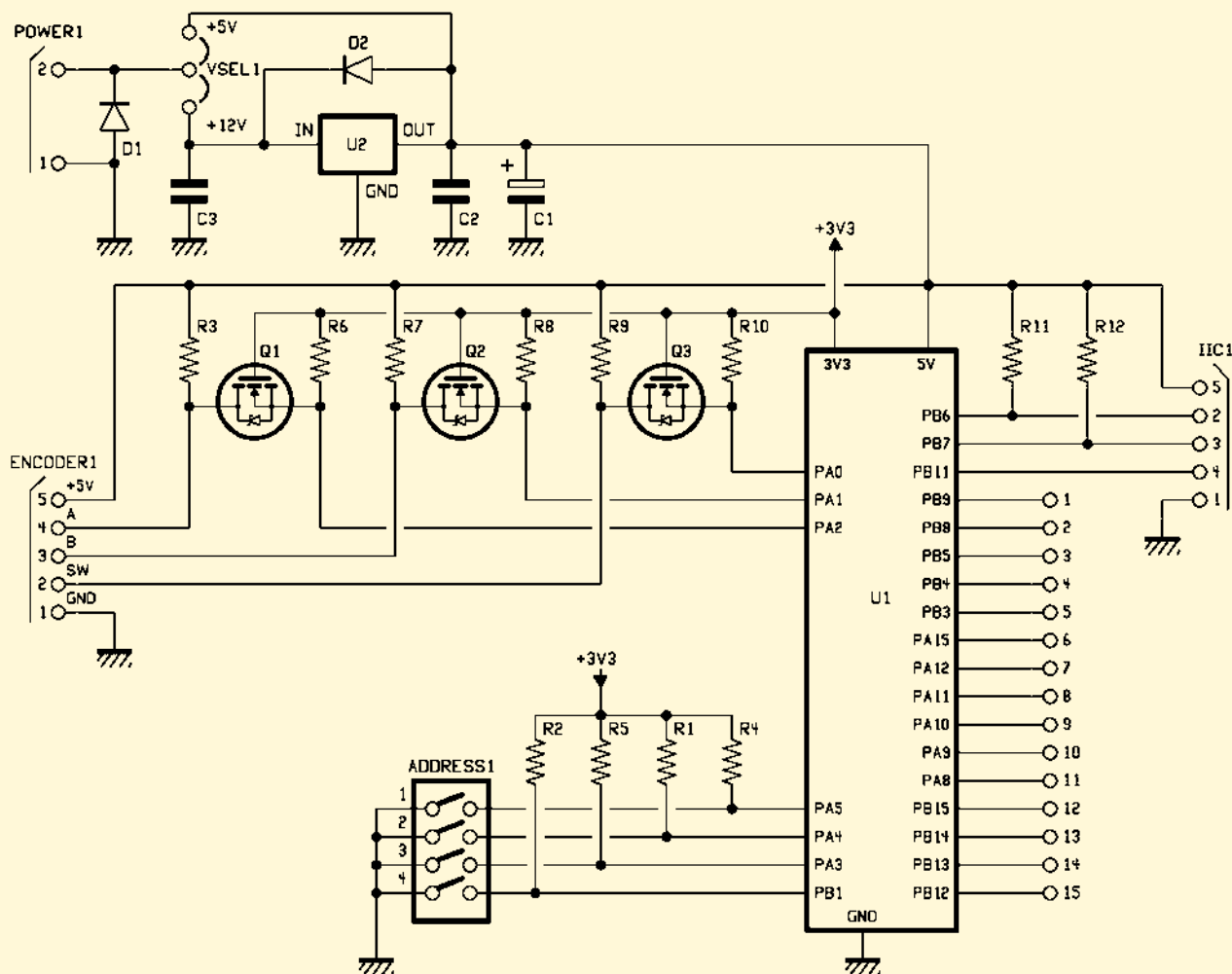
di LEONARDO SAMMARTANO



Realizziamo una unità encoder indipendente che lavori in modalità indiretta e renda il nostro dispositivo una periferica seriale o IIC. Seconda e ultima puntata.

N

el numero precedente (il n. 278) abbiamo parlato di MCU, encoder, timer e librerie limitandoci ad una chiacchierata sull'argomento senza toccare fisicamente l'hardware e il saldatore. In questa seconda parte passeremo dalla teoria alla pratica realizzando una unità encoder autonoma utilizzabile come periferica in grado di colloquiare con PC o altri dispositivi tramite l'interfaccia seriale (emulazione RS232 su porta USB) o tramite il bus IIC. Il progetto prevede di gestire un encoder in modalità indiretta ovvero utilizzando un timer interno della MCU il quale verrà configurato come un quadrature counter a 16 bit. Ricordiamo che con questa configurazione la conta dei tick avverrà senza mai interrompere l'esecuzione del programma.



L'esperienza è divisa in due fasi: nella prima fase realizzeremo l'interfaccia dell'encoder e caricheremo nella MCU un firmware il quale farà dialogare la nostra Encoder Unit tramite la porta USB simulando un flusso seriale (RS232). Per i test utilizzeremo un PC e il simulatore di terminale presente nell'IDE di Arduino.

Nella seconda parte aggiungeremo all'Encoder Unit la gestione del dialogo tramite il bus IIC. Normalmente i PC non sono dotati di porte IIC esterne per cui per il test sul bus IIC realizzeremo un protocollo converter seriale/IIC utilizzando per lo scopo un secondo microcontrollore. Questa seconda MCU opportunamente programmata farà da convertitore RS232/IIC/RS232 permettendoci di leggere e scrivere sul bus IIC da PC.

COSA SERVE

Per iniziare la nostra esperienza il minimo necessario è: un PC con installato un IDE (Integrated Development Environment) compatibile con la piattaforma Arduino, la libreria STM32Encoder,

una board basata sulla MCU STM32F103, un programmatore ST-Link V2, un encoder relativo, due resistenze di pull-up, dei ponticelli, una breadboard e infine un cavetto USB che ci permetta di collegare la board con l'STM32 al PC.

Il PC e l'IDE

L'IDE è quell'insieme di strumenti software che ci permette partendo da un testo (listato) di produrre un codice binario (compilato) e di caricare questo nella memoria flash delle nostre MCU (programmazione o upload). Ne esistono svariati ma, considerando che intendiamo lavorare all'interno dell'ambiente Arduino, sicuramente il primo riferimento è l'Arduino IDE; per le nostre prove abbiamo utilizzato la versione 2.3.2. Una valida alternativa è l'applicazione PlatformIO all'interno dell'ambiente Visual Studio Code. Per chi scegliesse di utilizzare quest'altro IDE la funzione import disponibile all'interno della PIO Home provvederà alle principali azioni necessarie per trasportare il codice descritto di seguito nel formato per l'utilizzo con PlatformIO.

Agli utenti Arduino IDE ricordiamo che dovranno impostare come Board una "Generic STM32F1 series", come Board part number "BluePill F103CB (or C8 with 128k", come Upload method "STM32CubeProgrammer (SWD)" e come USB support "CDC Generic 'Serial' supersede U(S)ART"). Queste impostazioni vanno verificate prima di ogni compilazione.

On-line si trovano un certo numero di ambienti di sviluppo e simulatori per l'ambiente Arduino ma per l'uso che dobbiamo farne noi questi non sono indicati o comunque comportano una serie di restrizioni per cui non sono stati considerati fra il materiale utilizzabile per questi esperimenti.

La libreria STM32Encoder

Per rendere l'attività di programmazione meno impegnativa si è scelto di appoggiarsi alla libreria STM32encoder sviluppata da Gianni Carbone e rilasciata sotto licenza GPL-3.0 (vedi approfondimento). Troverete molte altre librerie indirizzate alla gestione degli encoder; la nostra scelta è ricaduta su questa per vari motivi non ultima la semplicità con cui è possibile configurare una catena per la decodifica indiretta e per la buona leggibilità che offre a chi dovesse scegliere di leggerne i contenuti. Considerando che al momento sulle versioni di IDE testate non risulta disponibile l'installazione automatica per questa libreria, troverete le indicazioni per scaricarla da GitHub nel box "Come scaricare la libreria STM32Encoder". Per come si deve procedere per installare una libreria in formato ZIP in Arduino IDE si rimanda ai numerosi tutorial disponibili in rete.

La libreria ci permette di configurare, leggere e manipolare un quadrature counter con pochi e semplici comandi; sarà lei a farsi carico di tutti i passaggi a basso livello necessari affrancandoci da una parte tutt'altro che semplice del lavoro che dovremmo affrontare se decidessimo di procedere noi con la scrittura del codice necessario alla configurazione e uso dei timer come quadrature counter.

La board STM32F103

La scelta è ricaduta su una Blue Pill (codice STM32ARMCORE sul sito www.futurashop.it) principalmente per disponibilità, fattore di forma e costo contenuto ma, anche in questo caso, sarà possibile utilizzare altre board a condizione che utilizzino la MCU STM32F103 nella configurazione C8T6 o superiore e che i pin e i timer usati per la nostra esperienza siano disponibili.

Il programmatore ST-Link V2

L'ST-Link V2 è il programmatore che consigliamo di utilizzare fra l'IDE e le MCU della famiglia STM32F per le operazioni di caricamento del firmware. L'ST-Link V2 accede alla MCU tramite la porta JTAG (vedi box JTAG) predisposta dal costruttore della MCU come porta di debug e programmazione. Nel caso specifico delle board Arduino compatibili basate su MCU della STMicroelectronics troviamo 4 pin con la dicitura GND, SWCLK (Clock), SWDIO (Data) e 3.3; questi sono i quattro piedini della porta JTAG semplificata (forma ridotta della JTAG nota come SWD ovvero Serial Wire Debugging) che utilizzeremo per la programmazione della flash tramite ST-Link V2. Nel caso della Blue Pill il connettore JTAG è sul lato opposto al connettore USB.

Prima di poter utilizzare il programmatore ST-Link si dovrà procedere all'installazione dello stesso sul PC che si intende utilizzare. Le istruzioni sono reperibili sul sito della STMicroelectronics dove sono presenti anche i link per scaricare i file necessari all'installazione dei driver e dell'applicativo STM32 Cube Programmer.

In commercio sono reperibili versioni clone di questo dispositivo: possiamo solo dire che ad oggi tutte quelle provate funzionano.

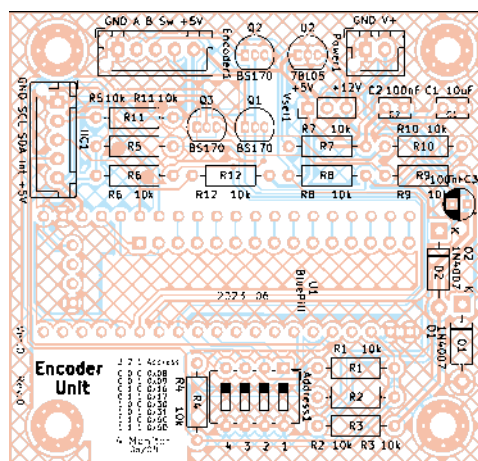
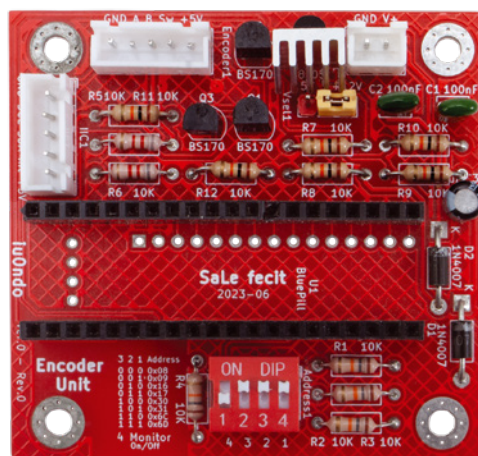
Se utilizzate una scheda della serie Nucleo 64 il programmatore ST-Link V2 non è necessario in quanto le Nucleo 64 vengono fornite già complete di programmatore. Il PCB di queste schede è diviso in due parti, la scheda vera e propria e una appendice del PCB generalmente più piccola contenente il programmatore. La programmazione di queste schede si effettua tramite la connessione USB posta sulla board del modulo programmatore. La possibilità di scrivere la flash tramite la porta USB della scheda è stata scartata preferendo l'adozione di un programmatore fisico che elimina eventuali problemi di compatibilità fra IDE e scheda MCU.

L'encoder

Per quanto riguarda la scelta dell'encoder non ci sono particolari restrizioni, tutti gli encoder con uscite in quadratura di tipo meccanico possono essere utilizzati senza problemi collegando le uscite dell'encoder alla MCU e applicando ad ogni uscita una resistenza di pull-up verso i 3.3 V (**Fig. 1**).

Se invece la nostra scelta si è rivolta verso un encoder ottico bisogna fare qualche verifica prima di collegarlo alla MCU. Gli encoder ottici spesso si presentano come device attivi ovvero richiedono





Elenco Componenti:

R1, R2, R3, R4 R5, R6, R7, R8, R9,
R10, R11, R12: 10 kohm
C1: 10 μ F 63 VL: elettrolitico
C2, C3: 100 nF
ceramico
D1, D2: 1N4007
Q1, Q2, Q3: BS170
U1: BlueBill
U2: 78L05
Encoder1: Encoder Rotativo
Address1: Dip-Switch 4 vie

Varie:
- Connettore Inserzione 2 vie
passo 2.54 mm
- Connettore Inserzione 5 vie
passo 2.54 mm (2 pz.)
- Strip maschio 3 vie passo 2.54 mm
- Jumper
- Strip Femmina 15 vie passo
2.54 mm (2 pz.)
- Circuito Stampato S1748
(63x60 mm)

un'alimentazione e le loro uscite possono fornire una tensione o essere del tipo Open-Collector. Nel caso in cui le uscite siano del tipo Open-Collector possiamo considerarle alla stregua di contatti meccanici riferiti a massa (negativo) e utilizzare lo stesso schema di connessione già

indicato per gli encoder meccanici a prescindere dal valore della tensione di alimentazione richiesta. Nel caso in cui le uscite siano attive ovvero siano loro ad erogare una tensione, dovremmo verificarne il valore in uscita e se questo risulta superiore ai 3.3 V dovremo disporre fra le uscite dell'encoder e la MCU un level shifter per ogni uscita così da evitare di applicare agli ingressi della MCU tensioni che la potrebbero danneggiare (**Fig. 2**). Alcuni encoder sono dotati di un pulsante e/o di altri segnali; per il nostro scopo non sono rilevanti per cui li ignoreremo.

Resistenze di Pull-up

Ne servono due, una per ogni segnale in uscita dall'encoder così che sia garantito il livello alto (3.3 V) quando il contatto è aperto e limitino la corrente quando il contatto è chiuso (livello basso). Non sono critiche come valore, l'importante è che garantiscano un passaggio di corrente che sia compreso indicativamente fra i 100 μ A e 1 mA per cui tutti i valori compresi fra 3.3 k e 33 k vanno bene. Queste due resistenze potrebbero essere eliminate settando opportunamente i piedini della MCU ma per semplicità si è preferito ricorrere a dei pull-up esterni piuttosto che utilizzare quelli interni della MCU.

IL PRIMO CODICE

Prima di iniziare con il codice, è necessaria una precisazione. Lo scopo di questo articolo è fornire spunti per sperimentare e, possibilmente, adottare soluzioni diverse su problemi già noti. Il codice prodotto non ha le caratteristiche necessarie né la pretesa di essere considerato un prodotto industriale o professionale. Anche in questo caso si è cercato di fornire spunti e soluzioni funzionanti ma principalmente comprensibili lasciando spazio alle personalizzazioni. Affrontiamo ora il codice che potrete scaricare nella sua forma completa dal sito elettronica.in.it dalla pagina di presentazione di questo numero. Lo tratteremo per blocchi e per passi successivi così da dover affrontare un solo problema alla volta. Nella versione stampata riportiamo solo le istruzioni e qualche commento mentre nella versione scaricabile troverete una quantità maggiore di commenti. Troverete le cartelle di progetto del software dall'apposita area della rivista. Assicuriamoci che nell'IDE sia installata la libreria STM32encoder sviluppata da Gianni Carbone; se non è presente procediamo con l'installazione (vedi punto sulla libreria). A questo punto la soluzione più semplice è quella

di prelevare le cartelle di progetto dal sito della rivista e una volta scaricate le posizionate nell'area a vostra scelta all'interno del PC. Diversamente dovrete creare un progetto nuovo e trascrivere al suo interno i listati che trovate di seguito. Inizieremo con l'analizzare il listato contenuto nella cartella progetto "Encoder-Step1-Slave". Nella prima parte (**Listato 1**) dopo aver incluso la libreria per la gestione degli encoder si passa a definire alcuni sinonimi (#define) e la variabile pubblica posRaw. La parte relativa alle premesse si conclude con la dichiarazione della routines instruction() e la creazione dell'oggetto enc associandolo al Timer2. Si rende necessario dichiarare la routines in testa al listato in quanto per una ragione puramente di comodo è stato scelto di posizionare le routines di servizio alla fine del listato ovvero dopo la loop(). Nella setup() si procede all'inizializzazione del LED interno e della seriale. È stato inserito un loop il quale attende che la porta USB sia collegata: in effetti ha principalmente lo scopo di evitare che il firmware proceda alla scrittura di messaggi sulla porta in assenza di un dispositivo collegato. Segue l'invio del messaggio di conferma dell'attivazione del firmware e le istruzioni d'uso. In conclusione della setup() viene inizializzato a 0 il registro del quadrature counter e azzerato il flag isUpadated() il quale ha lo scopo di informare il firmware sullo stato dell'encoder (mosso / non mosso) dall'ultima consultazione del quadrature counter. Durante la setup() il LED della scheda viene utilizzato per segnalare il completamento del ciclo di inizializzazione e attivazione del canale seriale di comunicazione.

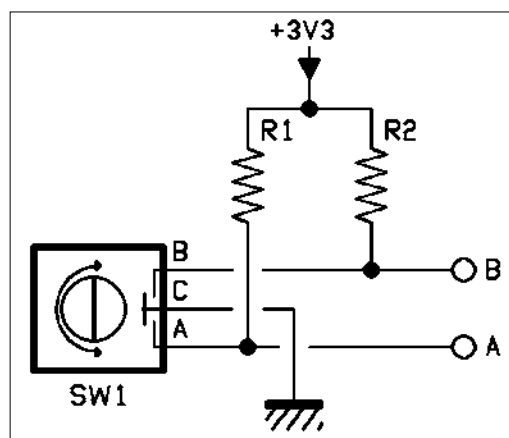


Fig. 1
Esempio di pull-up per encoder.

Nella seconda parte (**Listato 2**) troviamo la loop() all'interno della quale troviamo il loop di attesa di comandi in entrata sulla porta seriale e successivamente la parte relativa alla decodifica ed interpretazione dei due comandi ammessi: R legge la posizione; Z legge la posizione e la azzerata dopo la lettura. Di seguito alla loop() troviamo la sub instruction() che provvede a mandare verso il client il riepilogo delle istruzioni per la consultazione del quadrature counter. Come vedete, nel codice non sono stati inseriti cicli di lettura dei piedini collegati all'encoder nè sono stati attivati degli interrupt. Il codice accede al registro del quadrature counter solo quando il client ne fa richiesta. La gestione dell'encoder è totalmente affidata al Timer2 e all'hardware ad esso associato. A questo punto lanciamo la compilazione e se non riceviamo nessuna segnalazione di errore possiamo passare alla fase di caricamento nella MCU.

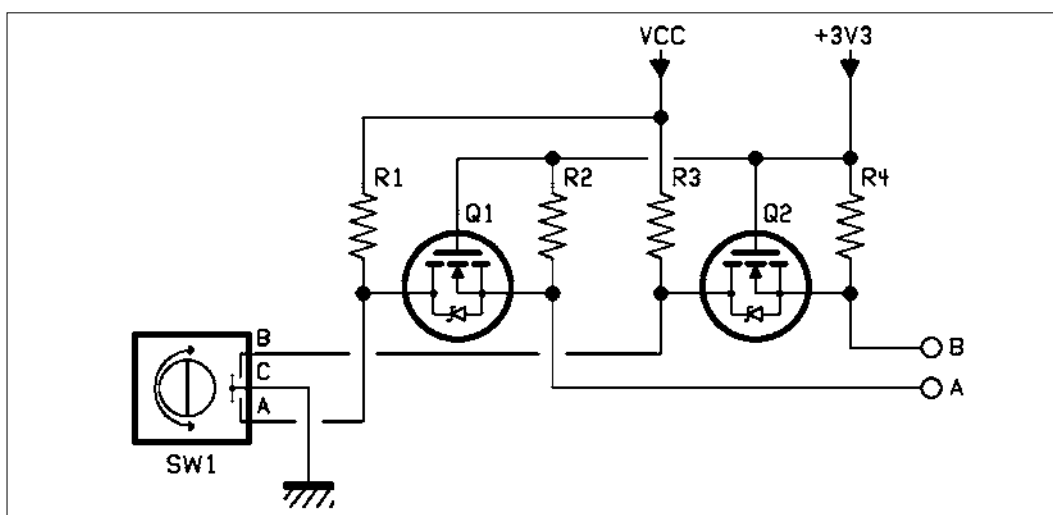


Fig. 2
Esempio di encoder collegato ad un Level Shifter.

↓ Listato 1

```
#include <STM32encoder.h> // https://github.com/gianni-carbone/STM32encoder

#define Enc1A PA0 //Definisce il pin dedicato al segnale A dell'encoder
#define Enc1B PA1 //Definisce il pin dedicato al segnale B dell'encoder

#define updateOff 1
#define updateOn 0

int16_t posRaw;

void instruction();

STM32encoder enc(TIM2);

void setup()
{
    uint32_t appo = 0; //Crea una variabile di appoggio

    pinMode(LED_BUILTIN, OUTPUT);
    digitalWrite(LED_BUILTIN, updateOn); //Accende il LED

    Serial.begin(115200);

    while (!Serial)
    {
        appo++; //Conta i cicli di attesa. La conta è solo finalità dimostrative
    }
    digitalWrite(LED_BUILTIN, updateOff); //Spegne il LED così da segnalare che la porta USB è connessa

    Serial.print("porta USB connessa dopo ");
    Serial.print(appo);
    Serial.println(" cicli di attesa");

    instruction(); //Presenta le istruzioni

    enc.pos(int32_t(0)); //Azzera il contatore dell'encoder
    enc.isUpdated(); //Azzera il flag che indica un movimento dell'encoder
}
```

Se non lo abbiamo fatto in precedenza provvediamo a collegare il cavo USB fra la board e il PC, resettiamo la MCU e avviamo il monitor seriale dell'IDE. A questo punto dovremmo vedere il messaggio di avvio. In questo messaggio vengono indicati i cicli di attesa che la MCU ha fatto prima che la porta USB fosse riconosciuta dal PC e le istruzioni per la consultazione dell'Encoder Unit. I due comandi (R e Z) devono essere digitati come lettere maiuscole altrimenti il codice le classificherà come comandi non riconosciuti. Adesso cominciate a ruotare l'alberino dell'encoder in senso orario e antiorario a piacimento e, sempre a vostro piacimento, inviate tramite il terminale i comandi all'Encoder Unit osservandone il comportamento.

QUANDO IL GIOCO SI FA DURO

Per quanto interessante quello che abbiamo realizzato non può essere definito un prodotto o almeno,

non può essere considerato una periferica al pari di molte altre che siamo abituati ad utilizzare nelle nostre applicazioni per cui adesso cercheremo di trasformare il nostro esperimento in qualcosa di più simile ad un prodotto.

Analizzando il lavoro sin qui fatto appare chiaro che la parte relativa all'interfacciamento dell'encoder e all'attivazione e gestione del quadrature counter è decisamente un lavoro di buon livello.

Un punto dove siamo un pò limitati è l'interfaccia, la quale essendo una seriale simulata su USB sicuramente permette un facile utilizzo del dispositivo da parte di un PC ma, se pensiamo di integrare la nostra Encoder Unit come periferica di una MCU le cose cambiano. Dover dedicare 2 piedini per la gestione di un solo device non è un bel punto di partenza, soprattutto se poi consideriamo la limitata disponibilità di porte seriali di alcune MCU. Provate ad immaginare cosa potrebbe significare



collegare su un Arduino Uno un dispositivo come il nostro: servirebbe una porta seriale da dedicare al colloquio con l'Encoder Unit ma l'ATMEGA328 (la MCU dell'Arduino Uno, appunto) a livello nativo ha una sola porta seriale che normalmente è dedicata alla programmazione della MCU e alle comunicazioni seriali via USB. Esistono gli strumenti software per gestire il problema ma richiedono spazio nel firmware e piedini disponibili. Guardando i periferici che normalmente troviamo sul mercato noteremo come molto spesso utilizzano il protocollo di comunicazione IIC (vedi il box "l'IIC e le resistenze di pull-up"). Questo protocollo basato sul modello master-slave utilizza come media due fili sui quali si svolge una comunicazione seriale bidirezionale e dove i diversi device presenti sul bus sono

identificati da un indirizzo a 7 bit che li rende unici. Le MCU che normalmente troviamo disponibili per lo standard Arduino sono sempre dotate di almeno una interfaccia IIC e la libreria per la gestione di questo protocollo è parte integrante del pacchetto Arduino (**Fig. 3**). A questo punto sembra chiara la direzione da intraprendere: conservando le funzionalità sulla porta seriale, dotiamo il nostro Encoder Unit di un client IIC così che possa essere raggiunto tramite questo versatile protocollo.

PREPARIAMO L'AMBIENTE

Come nella fase precedente non ci limiteremo a realizzare qualcosa ma metteremo su un vero e proprio banco di prova. Nell'esperienza precedente sul banco avevamo due oggetti, il Pc e la scheda

⬇ Listato 2

```
void loop()
{
    digitalWrite(LED_BUILTIN, updateOff); //Spegne il LED prima di processare il comando in ingresso

    while (!Serial.available())
    {
        if(enc.isUpdated()!=0)
        {
            digitalWrite(LED_BUILTIN, updateOn); //Accende il LED così da segnalare che l'encoder è stato mosso
        }
    }

    char USBbuffer = Serial.read();

    Serial.print(USBbuffer);

    switch (USBbuffer)
    {
        case 'R':
            Serial.print(" -----> Posizione rispetto allo zero ");
            Serial.println(int16_t(enc.pos())); //Legge il contatore
            break;
        case 'Z':
            Serial.print(" -----> Posizione rispetto allo zero ");
            Serial.print(int16_t(enc.pos())); //Legge il contatore
            enc.pos(uint32_t(0)); //Azzera il contatore
            Serial.println(" - Contatore azzerato");
            break;
        default:
            Serial.println ();
            Serial.println("=====> COMANDO NON RICONOSCIUTO");
            break;
    }
    instruction(); //Presenta le istruzioni
}

void instruction()
{
    Serial.println("Per leggere lo stato dell'encoder");
    Serial.println("senza azzerare il contatore digitare R");
    Serial.println("azzerando il contatore digitare Z");
    Serial.print(">"); //Presenta il prompt
}
```

Nella libreria STM32encoder, utilizzeremo alcune funzionalità principali, come la creazione dell'oggetto Quadrature Counter, le funzioni di lettura e impostazione del contatore e l'interrogazione dei registri per verificare se l'encoder è stato spostato. Per chiarire come la libreria aiuti a mantenere attivo il sottosistema hardware all'interno della MPU e lo renda accessibile al codice, esamineremo i moduli che la compongono. Questo ci permetterà di capire meglio come la libreria agevoli la comunicazione tra il software e l'hardware.

Creazione dell'oggetto Quadrature Counter: L'oggetto che creeremo per noi è rappresentato dal nome simbolico

che ci permetterà di rivolgerci alle funzioni contenute nella libreria e alle risorse hardware utilizzate (Timer) trattandole come se fossero membri di una famiglia ai quali chiediamo cose o assegniamo compiti. Creando l'oggetto di riferimento (nome simbolico scelto da noi) non solo ci si semplifica la vita rendendo il nostro codice più leggibile ma ci permette (in un uso più evoluto) di gestire più quadrature counter contemporaneamente senza che nascano "confusioni" fra un counter e l'altro.

Funzione di lettura/settaggio del contatore: è il metodo con cui interroghiamo il nostro oggetto (enc) facendoci restituire il contenuto del timer/counter all'istante

della consultazione. Questo metodo è totalmente asincrono rispetto al processo di conta del quadrature counter e non ne influenza il processo.

Lo stesso metodo, se usato valorizzando l'interno delle parentesi tonde, opera come processo di settaggio (inizializzazione a n) del quadrature counter ovvero, se noi invochiamo il metodo inserendo 0 all'interno della parentesi tonde otterremmo il settaggio a valor 0 (azzeramento) del counter.

Parimenti, se al posto dello 0 inseriamo un numero questo verrà comunque scritto nel contatore il quale partirà a contare da quel valore. Importante, il valore di inizializzazione del counter deve essere contenuto nel dominio

del contatore, nel nostro caso da -32768 a +32767, diversamente verranno effettuati dei troncamenti.

Interrogazione dei registri del quadrature counter: la catena hardware che trasforma il timer counter in un quadrature counter è composta oltre che dal multiplexer da una serie di registri di stato e logiche di controllo, fra queste vi è il flag che viene settato ogni qualvolta il contenuto del counter si incrementa o decrementa di uno step. Questo flag può essere utilizzato per molteplici scopi, nel nostro caso lo usiamo come indicatore letto tramite il codice senza attivare interrupt. Il comando che provvede a leggere lo stato del

Blue Pill nel ruolo di Encoder Unit. Il PC collegato tramite la porta USB all'Encoder Unit ci ha fornito l'interfaccia utente tramite il programma termi-

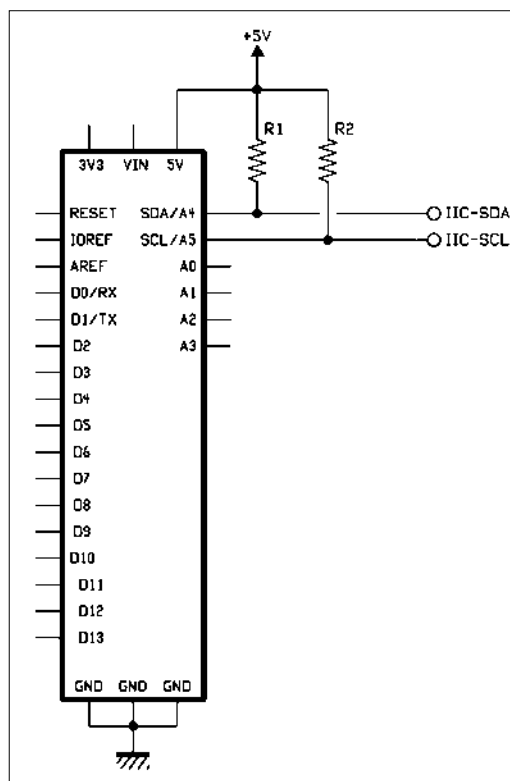


Fig. 3
Arduino Uno con re-
sistenze di Pull-Up

nale (Serial Monitor in Arduino IDE o Monitor in PlatformIO). In questa nuova configurazione gli oggetti sul banco saranno tre, il PC che come nel caso precedente farà da interfaccia utente tramite il terminale, la scheda Blue Pill che farà da Encoder Unit e un terzo oggetto che si collegherà fra il PC e l'Encoder Unit svolgendo il ruolo di convertitore seriale/IIC/seriale. Il protocollo di comunicazione in IIC è complesso e articolato e normalmente i desktop non sono dotati di porte IIC disponibili per l'utente (quando presenti sono interne e gestite a basso livello) né sono disponibili programmi generici che svolgano la funzione di terminale utilizzando come tramite degli adattatori USB-IIC. Il modo più semplice per aggirare questo scoglio è quello di continuare ad usare il PC come un terminale seriale via USB frapponendo un convertitore di protocollo fra il PC e l'Encoder Unit. Questo tipo di soluzione risulta perfettamente in linea con il nostro percorso in quanto è realizzabile tramite l'uso di una MCU nella quale caricheremo un firmware che si fa carico di tutte le operazioni necessarie per accedere e colloquiare con il periferico IIC ricevendo comandi e fornendo risposte sulla porta seriale (lato PC). In pratica una scheda Arduino compatibile (per le prove abbiamo utilizzato un Arduino Uno) dove carichiamo quello che sarà il codice che dovrà scrivere chiunque voglia usare il nostro dispositivo aggiungendo la gestione di un minimo di interfaccia sulla porta seriale.

flag provvede anche a resettarlo per cui, dopo la lettura il flag sarà basso sino a quando il contatore non rileva un nuovo tick sugli ingressi.

Creazione dell'oggetto Quadrature Counter

```
class STM32encoder {
public:
    STM32encoder(TIM_TypeDef
        *Instance, u8 _ICxFilter = 0,
        u16 _pulseTicks = 3);
```

Definisce una classe che viene associata al simbolo "STM32encoder" e vi associa un costruttore (il codice che provvederà a inizializzare un oggetto "classe"). La libreria prevede due modalità di definizione della classe,

noi usiamo la prima delle due ovvero quella indicata nel codice precedente.

Funzione di lettura/settaggio del contatore

```
i32 pos(void);
// get the current position
void pos(i32 _p);
// set the absolute position
to the given value
```

Il metodo pos è definito due volte, la prima è quella che effettua la lettura (getter) del quadrature counter e non prevede nessun parametro nella chiamata e restituisce un numero intero a 32 bit (i32 pos(void)). La seconda dichiarazione non prevede alcun valore restituito (void)

mentre richiede un parametro che sarà trattato dalla libreria come un numero intero a 32 bit (void pos(i32)) e utilizzato per caricare nel counter il nuovo valore corrente (setter) in maniera indipendente da quanto sta accadendo sugli ingressi dello stesso.

Interrogazione dei registri del quadrature counter

```
bool isUpdated(void);
// return a tick is done since
last call. Function call resets
the flag
```

Il metodo isUpdated è definito come una funzione che restituisce uno stato logico il quale sarà TRUE (HIGH o 1) se dall'ultima volta che abbiamo consultato il metodo il contatore ha ricevuto segnali sugli ingressi in quadra-

tura mentre restituirà un FALSE (LOW o 0) se non sono arrivati impulsi dall'ultima consultazione. L'azione di consultazione del metodo esegue il reset del registro per cui, dopo averlo letto il suo valore sarà FALSE.

Le parti di codice riportate sono un estratto del file STM32Encoder.h.

La libreria si compone di questo file e del file STM32Encoder.cpp. In questo secondo file troverete il codice vero e proprio che viene eseguito quando si invocano i metodi ma, nella nostra disamina abbiamo preferito fermarci al livello delle dichiarazioni (STM32encoder.h) essendo il valore aggiunto dalla lettura del file.cpp (e del resto del file.h) irrilevante per i nostri scopi.

▼ Listato 3

```
#include <Wire.h>
#include <STM32encoder.h> // https://github.com/gianni-carbone/STM32encoder

#define Enc1A PA0 //Definisce il pin dedicato al segnale A dell'encoder
#define Enc1B PA1 //Definisce il pin dedicato al segnale B dell'encoder
#define updateOff 1
#define updateOn 0
#define SDA1 PB7 //Definisce il pin dedicato al segnale SDA dell'IIC
#define SCL1 PB6 //Definisce il pin dedicato al segnale SCL dell'IIC
#define MONITOR_ON PB1 //Dip Switch 4 Abilitazione Monitor
#define ADDRESS_A PA5 //Dip Switch 3 Address bit A
#define ADDRESS_B PA4 //Dip Switch 3 Address bit B
#define ADDRESS_C PA3 //Dip Switch 3 Address bit C

int16_t posRaw;

#define DipSwitchNum 4 //Numero dei dip switch gestiti
#define DipSwitchMonitorPin 3 //Definisce il puntatore per lo Switch del Monitor

uint8_t DipSwitchPin[DipSwitchNum] = {ADDRESS_A, ADDRESS_B, ADDRESS_C, MONITOR_ON};
uint8_t I2C_ADDR;//Indirizzo IIC corrente

#define addressBitNum 3 //definisce il numero di bit gestiti per la selezione dell'indirizzo IIC
#define addressNum 8 //Numero degli indirizzi possibili

uint8_t addressTable[addressNum] = {0x08, 0x09, 0x16, 0x17, 0x30, 0x31, 0x6C, 0x6D};
char IICCommandAppo = ' ';

void instruction();
void i2c_setup();
void requestEvent();
void receiveEvent(int howMany);

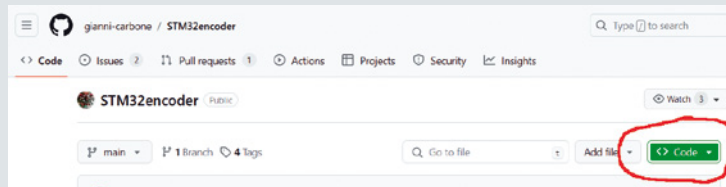
TwoWire Wire1(SDA1,SCL1);

STM32encoder enc(TIM2);
```

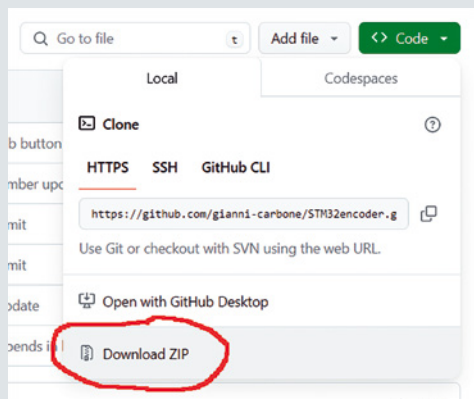

COME SCARICARE LA LIBRERIA STM32ENCODER

Se dal vostro IDE non è disponibile la libreria prodotta da Gianni Carbone (attenzione ne esistono altre con lo stesso nome o nomi molto simili ma non compatibili con il nostro codice) dovrete scaricarla dal repository al link di seguito <https://github.com/gianni-carbone/STM32encoder>.

Una volta aperto il link in alto a destra troverete il bottone verde "code"



Cliccandoci sopra si apre la tendina



Cliccate "Download ZIP", per scaricarla sul vostro computer.

ENCODER UNIT VERSIONE AGGIORNATA

Il nuovo dispositivo che stiamo per assemblare è un oggetto composto dalla MCU Blue Pill e da un certo numero di componenti come visibile dallo schema elettrico.

Come si può notare sono presenti alcuni blocchi

circuitali e per l'esattezza una unità alimentatore, un level shifter a tre canali e un gruppo di switch che permettono di modificare i parametri di funzionamento del firmware il tutto con al centro la MPU. Volendo realizzare una vera e propria unità periferica abbiamo sviluppato un PCB dove assemblare il tutto dotato di tre connettori che ne semplificano l'inserimento nelle nostre realizzazioni.

Unità Alimentatore

La scheda nasce per essere alimentata a 5V ma, se dovesse essere necessario, offre la possibilità di essere alimentata con una tensione compresa fra i 7,5V e i 12V. Le due modalità di funzionamento sono selezionabili tramite il jumper Vsel1.

Quando si opta per l'alimentazione con tensione superiore ai 5V, la riduzione della tensione è garantita da un regolatore a tre piedini il quale, se l'installazione non prevede condizioni gravose, può essere un 78L05 montato direttamente sulla scheda purché dotato di dissipatore (vanno bene i dissipatori da 10x10 mm generalmente usati sui ponti H delle schede di controllo delle stampanti 3d incollato con una goccia di colla cianacrilica) altrimenti dovrà essere un 7805 in contenitore TO220 montato esternamente alla scheda e dotato di un piccolo dissipatore.

Dal punto di vista delle protezioni è stato inserito il diodo D1 come ant'inversione di polarità.

La scheda può essere alimentata anche fornendogli i 5V dal connettore dell'encoder o dal connettore dell'IIC.

La protezione contro l'inversione di polarità funziona solo quando l'interruttore Vsel1 è impostato su 5V.

In caso contrario, il diodo di protezione contro l'inversione viene scollegato dalla linea da 5V.

ADDRESS BIT

2 ² (switch 3)	2 ¹ (switch 2)	2 ⁰ (switch 1)	IIC Address (nn)	Possibile corrispondenza
0	0	0	0x08	??
0	0	1	0x09	??
0	1	0	0x16	LM25066
0	1	1	0x17	LM25066
1	0	0	0x30	SAA2502
1	0	1	0x31	SAA2502
1	1	0	0x6C	PCA9685
1	1	1	0x6D	PCA9685

Tabella. 1
Tabella degli indirizzi potenzialmente liberi

Listato 4

```
void setup()
{
  uint32_t appo = 0; //Crea una variabile di appoggio
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, updateOn); //Accende il LED

  for (uint8_t ii= 0; ii < addressNum; ii++)
  {
    pinMode(DipSwitchPin[ii], INPUT);
  }

  Serial.begin(115200);
  MonitorSeriale = digitalRead(DipSwitchMonitorPin);
  if(0==MonitorSeriale)
  {
    while (!Serial)
    {
      appo++; //Conta i cicli di attesa. La conta Ã solo finalitÃ dimostrative
      if ((appo % 500) == 0)
      {
        digitalWrite(LED_BUILTIN, !(digitalRead(LED_BUILTIN)));
      }
    }
    Serial.print("porta USB connessa dopo ");
    Serial.print(appo);
    Serial.println(" cicli di attesa");
    instruction(); //Presenta le istruzioni
  }
  else
  {
    delay(2500); //Se non Ã attivo il monitor attende un tempo arbitrario prima di proseguire
  }
  digitalWrite(LED_BUILTIN, updateOff);

  enc.pos(int32_t(0)); //Azzerare il contatore dell'encoder
  enc.isUpdated(); //Azzerare il flag che indica un movimento dell'encoder

  i2c_setup();
}
```

Il level shifter

Come accennato il nostro sistema internamente lavora a 3.3V ma alcuni encoder possono richiedere tensioni diverse per il loro funzionamento; da qui l'esigenza di rendere sicuri i segnali che vanno dall'encoder fisico alla MCU. Questa sicurezza è ottenuta inserendo fra i due dispositivi un circuito in grado di operare con due tensioni diverse in ingresso ed in uscita e di trasferire in maniera bidirezionale i segnali (level shifter).

La scelta di realizzare i level shifter con componenti discreti è stata preferita a quella di utilizzare dei chip specializzati per questioni di reperibilità dei componenti e per offrire un ulteriore spunto per chi non si sia mai trovato ad avere a che fare con questi simpatici dispositivi.

I canali previsti sulla scheda sono tre ovvero due per i segnali in quadratura provenienti dall'encoder (A e B) e un terzo per un eventuale pulsante o per lo switch di cui sono dotati alcuni encoder.

Il dip switch

Introducendo il canale di comunicazione IIC nasce il problema di definire l'indirizzo al quale risponderà la nostra Encoder Unit.

Replicando la soluzione già largamente adottata da altri produttori di device IIC abbiamo previsto la possibilità di selezionare l'indirizzo da assegnare tramite dei ponticelli esterni alla MCU e non resistendo al fascino che emana dai dip switch (la presenza di anche un solo dip switch dà alla scheda che lo ospita un che di misterioso e professionale, una calamita per gli occhi) abbiamo inserito un dip switch a quattro canali di cui tre dedicati alla selezione dell'indirizzo e uno per definire il livello di interazione della scheda. Tutte le funzioni dei dip switch sono definite all'interno del firmware.

I connettori

La scheda prevede tre connettori e rispettivamente: Encoder1 mette a disposizione 5 fili per la con-



Nel quinquennio fra il 1985 e il 1990 alcuni fra i principali costruttori di chip sentirono l'esigenza di definire uno standard comune per l'accesso ai loro chip digitali finalizzato al collaudo, al test e come supporto allo sviluppo dei dispositivi elettronici basati su questi chip. Fra i fondatori di questo consorzio vi erano AT&T, DEC, IBM, Nixdorf, Philips, Siemens, Texas Instruments. Dal loro lavoro scaturì lo standard IEEE 1149.1 "*Standard Test Access Port and Boundary-Scan Architecture*" confidenzialmente noto come JTAG. Nel 1990 la Intel rilascia il microprocessore 80486 dotato di porta JTAG, probabilmente il primo dispositivo VLSI commerciale dotato di questa porta di comunicazione.

nessione di encoder passivi, attivi e di un eventuale pulsante; IIC1 il quale oltre all'alimentazione e ai segnali dell'IIC (SCL e SDA) ha un segnale indicato come "Int" destinato a consentire la gestione di accessi da o verso i device esterni in modalità Interrupt ed infine Power1 il quale consente di alimentare la scheda a 5V o con tensione superiore (vedi dettaglio sull'unità alimentatore).

AGGIORNAMO IL FIRMWARE DELL'ENCODER UNIT

Pronti al salto di qualità? Per semplicità si è scelto di creare una nuova cartella di progetto anche se buona parte del codice è quello che già abbiamo conosciuto e usato nella fase precedente della sperimentazione per cui, cominceremo con lo scaricare il nuovo firmware che troviamo nella cartella progetto "Encoder-Step2-Slave" per poi copiarla nella nostra area di lavoro sul PC e aprirla con Arduino IDE (per chi intende trascrivere di suo pugno il lista-to vale quanto già detto in precedenza). Ricordiamo agli utenti Arduino IDE che devono verificare la corretta configurazione dei parametri di compilazione che abbiamo già esposto in precedenza. Cominciando dall'alto avremo le nuove premesse (**Listato 3**). In questa parte troverete già inserite le inizializzazioni relative alla gestione della selezione dell'indirizzo da assegnare al device sul bus IIC. Le dichiarazioni delle routines che saranno utilizzate ora includono la parte che riguarda la gestione del traffico sul canale IIC ed è stata aggiunta la dichiarazione che istanza l'oggetto Wire1 che utilizzeremo per gestire le attività sulla porta IIC. Nel creare l'oggetto Wire1 definiamo anche quali piedini fra quelli riservati all'interfaccia per la prima porta IIC dell'STM32F103 utilizzeremo come SDA e SCL; questo si rende necessario in quanto le MCU della serie STM32F1 prevedono più di un canale di uscita per ogni interfaccia seriale e in fase di inizializzazione del canale è opportuno se non necessario indicare quale coppia di piedini intendiamo utilizzare. Nella parte alta della setup() (**Listato 4**) troviamo l'inizializzazione dei piedini che utilizzeremo per

leggere lo stato dei dip switch. A seguire troviamo il loop di attesa della connessione sulla porta USB. Questo loop è eseguito solo se il dip switch 4 (Monitor On/Off) è attivo ovvero se si vuole che l'Encoder Unit oltre a ricevere comandi e rispondere sulla porta seriale utilizzi quest'ultima per fornirci le istruzioni per l'uso e per aggiornarci su quello che succede all'interno del firmware. Se il dip switch 4 è settato su Off questo loop viene saltato e nel corso del funzionamento potremo utilizzare la porta seriale solo per interrogare l'Encoder Unit. Nella loop() (**Listato 5**) in buona sostanza ritroviamo quanto già visto nella versione precedente; semplicemente le parti relative alla stampa dei commenti sono state assoggettate al controllo del dip switch 4 così da sapere se siamo in modalità verbose o se ci si deve limitare a fornire l'eco dei comandi e le eventuali risposte là dove i comandi siano stati riconosciuti.

L'ultima parte del firmware contiene le routines di servizio. Qui troveremo la I2c_setup() la quale provvede ad inizializzare il canale IIC e a collegare agli eventi .onRequest() e .onReceive() le rispettive routine di gestione (requestEvent() e receiveEvent()). Di fatto tutta la gestione del colloquio fra il nostro device e il master del bus IIC è gestito da queste tre routines e dalla libreria Wire.h. Anche all'interno di questa parte di codice troviamo le parti che ci informeranno tramite la porta seriale su quanto accade sul canale IIC se il dip switch 4 è settato su On.

Una nota a parte la merita la trasmissione del dato (numero di tik contati del quadrature counter) sul bus IIC. Il nostro counter ha una ampiezza di 16 bit ovvero 2 parole da 8 bit; in tutto il codice noi trattiamo questo valore come associato ad una variabile intera da 16 bit e questo ci permette di vederlo come un unico oggetto ma, quando dobbiamo inviarlo sul bus IIC, si presenta un piccolo problema: il bus IIC ragiona con parole da 8 bit e, così come lo stiamo usando noi, non sa gestire le parole di lunghezza superiore per cui dobbiamo provvedere noi a dividere i sedici bit in due parole da 8. La parte bassa (gli otto bit meno significativi) la estraiamo convertendo a 8 bit la nostra variabile tramite la funzione int8_t() la quale provvede a "scartare" gli otto bit più significativi. Per quanto riguarda la parte alta (gli otto bit più significativi) abbiamo prima fatto ruotare verso destra di otto bit il contenuto della nostra variabile (>>8) ottenendo l'inversione fra la parte alta e quella bassa per poi tagliare gli otto bit alti ancora una volta con la funzione int8_t().

▼ Listato 5

```
void loop()
{
  digitalWrite(LED_BUILTIN, updateOff); //Spegne il LED prima di processare il comando in ingresso
  MonitorSeriale = digitalRead(DipSwitchPin[DipSwitchMonitorPin]);
  while (!Serial.available())
  {
    if(enc.isUpdated()!=0)
    {
      digitalWrite(LED_BUILTIN, updateOn); //Accende il LED così da segnalare che l'encoder è stato mosso
    }
  }
  char USBbuffer = Serial.read();
  Serial.print(USBbuffer);
  switch (USBbuffer)
  {
    case 'R':
      if(0==MonitorSeriale)
      {
        Serial.print(" -----> Posizione rispetto allo zero ");
      }
      Serial.println(int16_t(enc.pos())); //Legge il contatore
      break;
    case 'Z':
      if(0==MonitorSeriale)
      {
        Serial.print(" -----> Posizione rispetto allo zero ");
      }
      Serial.print(int16_t(enc.pos())); //Legge il contatore
      enc.pos(uint32_t(0)); // Azzera il contatore
      if(0==MonitorSeriale)
      {
        Serial.println(" - Contatore azzerato");
      }
      break;
    default:
      if(0==MonitorSeriale)
      {
        Serial.println ();
        Serial.println("===== COMANDO NON RICONOSCIUTO");
      }
      break;
  }
  if(0==MonitorSeriale)
  {
    instruction(); //Presenta le istruzioni
  }
}

void instruction()
{
  Serial.println("Per leggere lo stato dell'encoder");
  Serial.println("senza azzerare il contatore digitare R");
  Serial.println("azzerando il contatore digitare Z");
  Serial.print(">"); //Presenta il promp
}
```

Di questa modalità di trasmissione dell'informazione se ne dovrà tenere conto nello scrivere la procedura di consultazione della nostra Encoder Unit in quanto il dato ricevuto dovrà essere ricostruito con un processo simile che vedremo quando affronteremo il convertitore di protocollo.

A questo punto non rimane che compilare, verificare di non avere segnalazioni di errore e caricare

il firmware nella scheda Blue Pill dopo di che possiamo effettuare una prima serie di test utilizzando la porta seriale simulata sulla USB per verificare il corretto funzionamento del firmware sul canale seriale, la funzione Monitor (dip switch 4) e la selezione degli indirizzi per la porta IIC. Prima di passare ai test del canale IIC provvediamo a realizzare il convertitore di protocollo.



Il bus IIC necessita di una resistenza di pull-up per ognuna delle due linee. Queste resistenze sono indispensabili perché parte della comunicazione fra il master e gli slave avviene tramite la messa a massa di una linea di comunicazione e se le linee fossero alimentate da uscite attive (alimentate dai piedini dei microcontrollori), mettendole a massa si metterebbe in corto l'uscita che in quel momento sta alimentando la linea in questione invece, rendendo la linea riferita al positivo tramite una resistenza di pull-up, se la si mette a massa si farà scorrere solo la corrente del pull-up senza mettere in corto nessuna uscita. Secondo le specifiche la corrente minima che deve essere garantita in un bus IIC è di 1 mA per cui assumendo come riferimento i 5 V il valore commerciale più prossimo è 4,7 k. In effetti questo valore è abbastanza indicativo e tutti i valori compresi fra 10 k e 2,2 k in genere funzionano benissimo. Per quanto riguarda il riferimento normalmente si usano i 5 V ma, se usiamo tutti dispositivi funzionanti a 3,3 V va bene riferire i Pull-up ai 3,3 V. Nel nostro caso, utilizzando come protocol converter un ATMEGA328 alimentato a 5 V riferiremo i pull-up della IIC ai 5 V (Errore: sorgente del riferimento non trovata). Per il bus IIC non sono necessari adattatori di livello (level shifter) in quanto la specifica del protocollo IIC prevede che le porte IIC siano tutte 5V tolerant.

IL CONVERTITORE DI PROTOCOLLO

Abbiamo già anticipato che realizzeremo noi il convertitore di protocollo e che questo sarà un oggetto separato e distinto dalla Encoder Unit. La scelta di farlo con una scheda diversa dalla Blue Pill è solo per rendere a colpo d'occhio la distinzione fra i due dispositivi ma nulla vieta che venga realizzato con una scheda diversa dall'Arduino Uno e magari proprio su un'altra Blue Pill. D'altronde, lo scopo finale del nostro progetto è quello di produrre un device IIC e dunque un oggetto che possa parlare con altri sistemi e altri mondi (ad esempio diventare il periferico di un Raspberry).

Se la vostra scelta in merito alla MCU che intendete utilizzare come protocol converter sarà diversa dalla ATmega328 verificate semplicemente che sia comunque disponibile una porta seriale emulata su USB e una porta IIC e che sia utilizzabile con il framework Arduino.

Il codice che realizzeremo sarà completamente sganciato dalla scelta della MCU in quanto, operando all'interno del framework Arduino, le due librerie che utilizzeremo sono parte integrante del framework stesso e dunque trasparenti per quanto riguarda il nostro lavoro.

Ancora una volta dobbiamo scaricare una cartella progetto; questa volta sarà la "Encoder-Protocol-Converter" che come per le altre la copieremo nella nostra area di lavoro per poi aprirla con Arduino IDE. I settaggi della scheda dovranno rispecchiare la vostra scelta, nel nostro caso la scheda selezionata è la "Arduino Uno".

Una parte decisamente rilevante di questo codice è costituita dai messaggi che vengono inviati verso il monitor seriale per informare e guidare l'operatore

nei test. Le soluzioni possibili in questi casi sono molte e vanno dal caricare tutte le frasi da usare all'interno di array di stringhe (molto facile, veloce ma consuma grandi quantità di RAM che, in linea generale scarseggia nell'ATmega328). Un'altra soluzione è quella di allocare delle costanti nell'area di memoria normalmente riservata al codice: non consuma RAM e generalmente non crea problemi di spazio per il codice. E poi c'è la soluzione di mezzo ovvero, scrivere direttamente le stringhe all'interno dei comandi .print(), poco elegante ma molto semplice da gestire soprattutto per chi intende trascrivere il codice a mano o per chi volesse personalizzare i testi in uscita dal converter. Questa soluzione ha lo svantaggio di generare un copioso numero di righe nel codice per cui nella versione stampata abbiamo ridotto al minimo le righe di stampa lasciando la versione integrale nel progetto scaricabile.

Prima del setup() troverete una struttura del tutto simile a quanto già visto negli altri due listati per cui nulla di nuovo o di rilevante.

Segue la loop(). La struttura che abbiamo scelto è per certi versi atipica in quanto all'interno della loop() abbiamo collocato due processi, uno per ognuna delle due funzioni di questo firmware, i quali costituiscono di fatto altri due loop uno di seguito all'altro. Si è arrivati a questa scelta per permettere lo svolgimento di diversi test senza dover necessariamente resettare l'Arduino Uno.

I due processi sono rispettivamente uno scanner IIC il quale cerca ed elenca i dispositivi presenti sul bus IIC e il processo che gestisce il colloquio con l'Encoder Unit.

Lo scanner IIC (IICScanner()) chiama in sequenza i 126 indirizzi possibili sul bus IIC (l'indirizzo 0 è il master). Con questa scansione vengono interrogati anche gli indirizzi riservati. Il risultato è una lista dove per ogni indirizzo viene indicata l'apertura del test, l'avvenuta chiusura del dialogo e il risultato ("No devices"; "Unknown error"; "IIC device found"). In fondo alla lista viene fornito il numero di device trovati. Se il numero di device trovati è maggiore di zero è possibile inserire l'indirizzo del device con il quale si vuole colloquiare. La procedura di inserimento dell'indirizzo (IICAddressAssign()) verifica che l'indirizzo immesso sia scritto nel formato corretto (esadecimale) e compreso nell'intervallo possibile (0x01 – 0x7F). In fase di digitazione è possibile inserire l'indirizzo di un device assente; il controllo verrà comunque effettuato alla fine dell'inserimento dell'indirizzo e, se non risulta presente, il processo riprende dall'inizio. Immesso un indirizzo formalmente corretto e corrispondente

ad un device attivo si passa alla seconda fase della loop(). Dopo aver inviato sul terminale la conferma di aver agganciato il device, si passa alla seconda fase ovvero quella dove possiamo inviare dei comandi all'Encoder Unit e leggerne le risposte. Da questo secondo anello si esce digitando il comando 'Q' (Quit). Il colloquio con l'Encoder Unit di fatto è gestito dalle due routines leggiEncoderUnit() e AzzerEncoder(). La leggiEncoderUnit() manda una richiesta per l'invio di 2 parole (da otto bit) all'indirizzo specificato (.requestFrom()). Questa richiesta è inoltrata ogni qual volta viene inviato un carattere sulla porta serale ovvero tutte le volte che tramite il monitor seriale inviamo un carattere.

Dopo l'invio della richiesta il firmware raccoglie i due byte e li mette in xl e xh che sono due variabili a 16 bit. Al momento i due byte ricevuti occupano gli otto bit meno significativi di entrambe le variabili. A questo punto prima di ricomporre il valore ricevuto, si sposta verso sinistra di otto bit (<<8) il valore contenuto nella xh. A questo punto avremo che le due parole ricevute dal device chiamato sono: la prima nella parte bassa di xl e la seconda nella parte alta di xh. A questo punto basta fondere insieme le due variabili per ottenere nuovamente il valore a 16 bit inviato dal client (EncoderPos = xh | xl;) e il gioco è fatto.

Se il comando che inviamo dal monitor seriale è la lettera Z (maiuscola) il firmware, dopo aver chiamato la leggiEncoderUnit() e presentato il valore corrente del quadrature counter, invoca la AzzerEncoder() la quale trasmette il comando al client (.write()) dopo aver trasformato il carattere 'Z' nel corrispondente numero intero a 8 bit ((uint8_t) "Z"). A questo punto il controllo ritorna alla loop() la quale effettua nuovamente l'interrogazione del client prelevando il valore attuale che deve essere zero e ce lo presenta.

Per quanto riguarda i collegamenti fra il convertitore di protocollo e l'Encoder Unit è necessario solo connettere insieme i negativi delle alimentazioni delle due schede e collegare il piedino SDA del protocol converter (se realizzato con Arduino Uno è il A4) con il piedino SDA dell'Encoder Unit (piedino 3 del connettore IIC1) e il piedino SCL del protocol controller (se realizzato con Arduino Uno è il A5) con il piedino SCL dell'Encoder Unit (piedino 2 del connettore IIC1). Non sarà necessario inserire le resistenze di pull-up in quanto sono già presenti all'interno della scheda Encoder Controller.

Come accennato all'inizio del nostro discorso, alcuni encoder sono corredati con un pulsante il quale si aziona se si preme l'alberino, dal punto di

vista del nostro excursus il pulsante non costituiva un elemento rilevante per cui lo abbiamo solo citato ma, per completezza nel circuito abbiamo previsto anche l'hardware per gestirlo. Chi volesse sperimentarne l'uso lo può fare aggiungendo alcune righe di codice. Anche in questo caso ci sono almeno due strade per affrontare la gestione del pulsante, scrivere da zero la parte del firmware o appoggiarsi alla libreria di Carbone la quale prevede alcune funzioni proprio per la gestione (monitoraggio e lettura) del pulsante.

Nello schema l'encoder che utilizzeremo è indicato come Enc0, troverete anche le diciture Enc1 ed Enc2, questo perché con lo STM32F103 è possibile attivare tre timer in configurazione quadrature counter e dunque gestire con una sola MCU tre encoder in modalità indiretta contemporaneamente. In questa versione è stata sviluppata solo la parte relativa all'encoder zero per semplicità ma nulla vieta che voi possiate implementare uno o entrambe gli altri due encoder ovviamente apportando le opportune modifiche al firmware.

Per quanto riguarda il PCB e il montaggio dei componenti, considerando le correnti e la tipologia dei segnali in gioco non ci sono particolari attenzioni da prestare.

CONCLUSIONI

Se siete arrivati sin qui o siete veramente tenaci o magari siamo riusciti ad incuriosirvi. Speriamo che questa chiacchierata abbia stimolato la vostra curiosità e che abbia ampliato almeno in parte la vostra conoscenza su un elemento di larga diffusione che a volte ci limitiamo ad usare dandolo un pò per scontato. 

COSA OCCORRE ?		
Codice	Descrizione	Prezzo € IVA inclusa
ENCODER12MM	Encoder rotativo 12 mm con manopola	3,50
YC093	Breakout con Encoder rotativo 12 mm	5,00
STM32ARMCORE	Board con Microcontrollore STM32F103C8T6	14,80
IL MATERIALE VA RICHIESTO A:		
Futura Elettronica, Via Adige 11, 21013 Gallarate (VA) Tel: 0331-799775 - www.futurashop.it		



ENTRA NEL MONDO DELL'ELETTRONICA

KIT FOR MAKERS

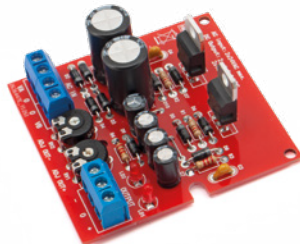
Le scatole di montaggio di Futura Elettronica sono semplici da montare e da utilizzare, ideali per avvicinarsi al fantastico mondo dell'elettronica. Non utilizzano microcontrollori e sono fornite con schemi elettrici e istruzioni.

Alimentatore stabilizzato da 1,5 a 35V



cod. FK003 € 10,50

Alimentatore stabilizzato duale e regolabile



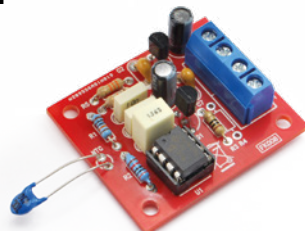
cod. FK004 € 14,00

Scheda di controllo automatico per ventilazione



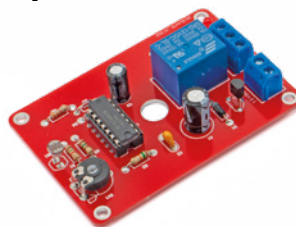
cod. FK007 € 14,00

Regolatore di velocità per ventole



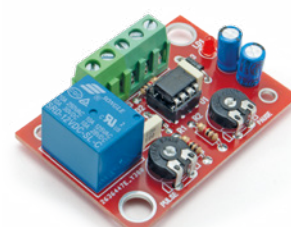
cod. FK008 € 17,80

Interruttore crepuscolare



cod. FK001 € 9,00

Modulo timer a intermittenza



cod. FK002 € 12,00

Timer Start e Stop



cod. FK005 € 14,00

Interruttore battimani



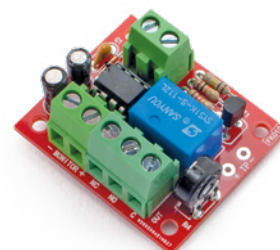
cod. FK006 € 14,00

Luce psichedelica a LED



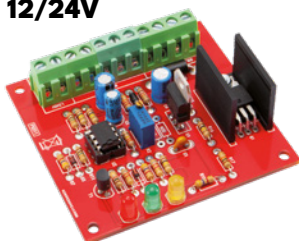
cod. FK009 € 12,00

Protezione bassa tensione



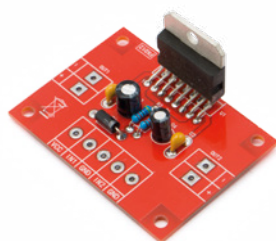
cod. FK010 € 11,50

Regolatore PWM per pannelli fotovoltaici 12/24V



cod. FK011 € 18,00

Amplificatore audio stereo 2x15



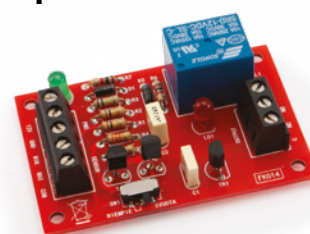
cod. FK012 € 8,50

Automatismo per tende e tapparelle motorizzate



cod. FK013 € 13,50

Stabilizzatore livello liquidi



cod. FK014 € 13,50

Prezzi IVA inclusa

CON I KIT DI FUTURA ELETTRONICA

Indicatore di livello liquido per vasche



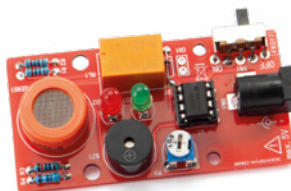
cod. FK015 € 8,50

Allarme batteria scarica



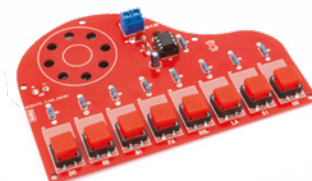
cod. FK016 € 8,00

Alcol tester



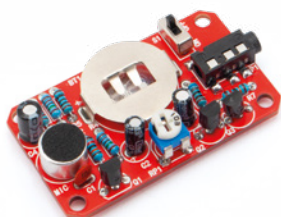
cod. FK017 € 9,00

Mini piano elettronico con NE555



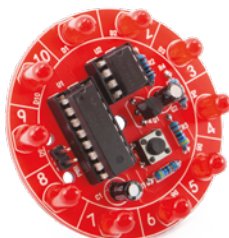
cod. FK018 € 8,00

Amplificatore acustico



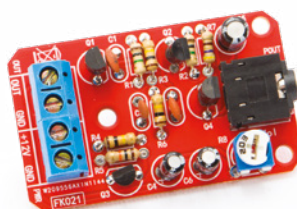
cod. FK019 € 9,00

Ruota della fortuna a LED



cod. FK020 € 8,00

Generatore di rumore bianco



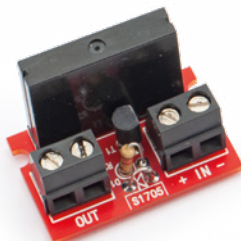
cod. FK021 € 8,00

Candela a LED con effetto fiamma



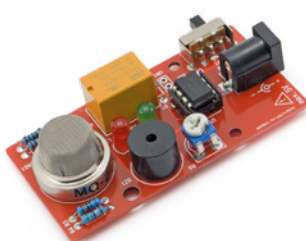
cod. FK022 € 8,00

Scheda con relè allo stato solido



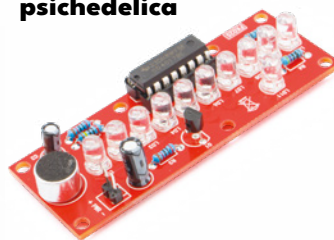
cod. FK023 € 5,50

Scheda con sensore di gas GPL, propano e idrogeno



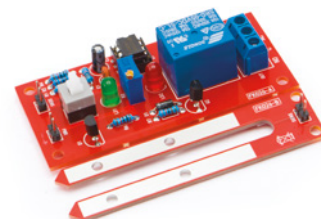
cod. FK024 € 11,00

Barra a LED psichedelica



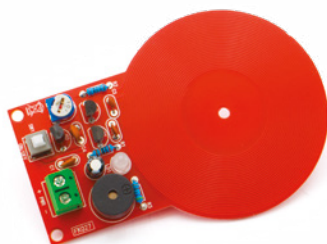
cod. FK025 € 8,50

Sensore di umidità



cod. FK026 € 13,00

Mini Metal Detector



cod. FK027 € 7,50

Lampeggiatore a LED



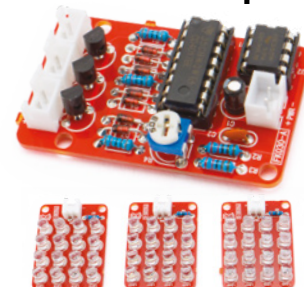
cod. FK028 € 6,50

Campanello



cod. FK029 € 8,00

Luci LED stroboscopiche



cod. FK030 € 14,00

Impara facilmente l'elettronica!

EDUKIT 01

11

**FACILI PROGETTI
PER IMPARARE
L'ELETTRONICA
DI BASE**

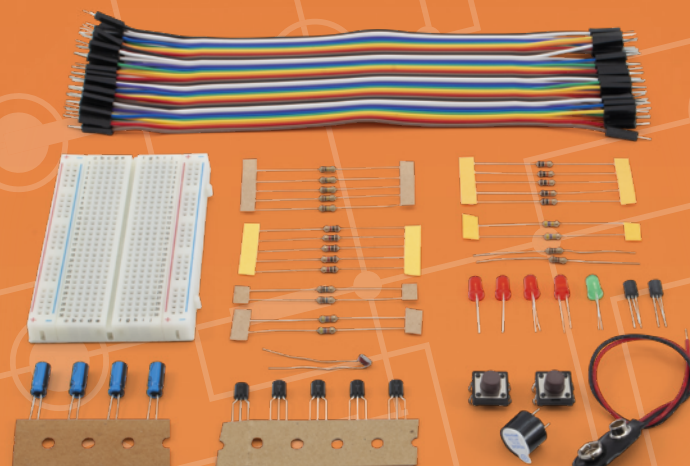
Kit di 11 esperimenti di elettronica base

Divertente, sicuro ed educativo, non richiede alcuna saldatura. Il kit contiene una breadboard 400 contatti con base autoadesiva e una serie di componenti per realizzare 11 progetti di elettronica di base. Tra i progetti: LED lampeggianti, sensore di luce, tester di polarità, circuito di start-stop, timer e molti altri.



€ 19,90

cod. EDUKIT01



Kit di 10 esperimenti con energia solare

Avvicinati facilmente al mondo dell'energia solare apprendendo e mettendo in pratica semplici circuiti. Per la realizzazione dei 10 progetti non serve alcuna attrezzatura, nemmeno il saldatore, in quanto il sistema a breadboard permette ogni genere di collegamento elettrico tra i componenti forniti, mediante l'innesco degli stessi nella piastra; basta seguire le immagini presenti nel manuale in dotazione corredato di schemi elettrici e di descrizione tecnica.

EDUKIT 02

10

**ESPERIMENTI
CON DIFFERENTI
GRADI DI
DIFFICOLTÀ**



€ 17,90

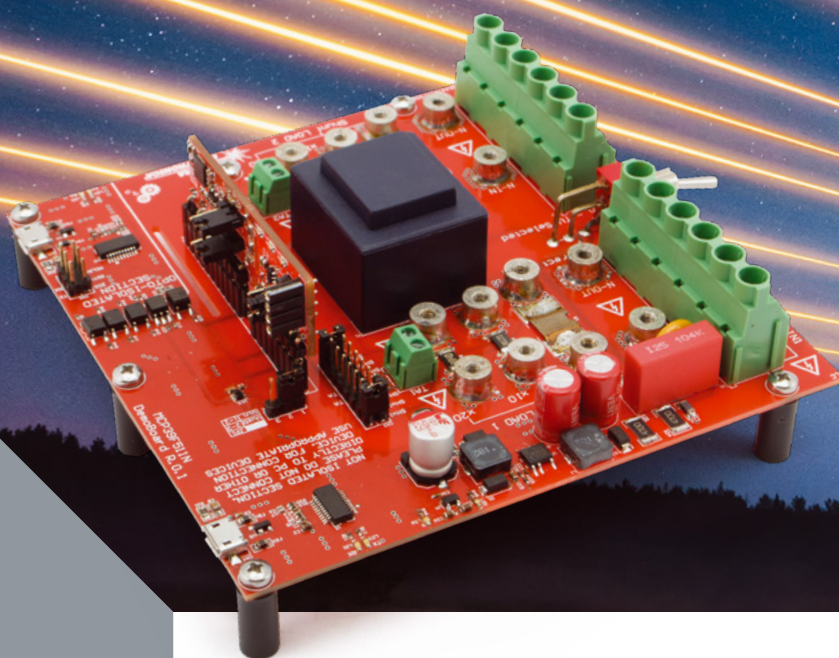
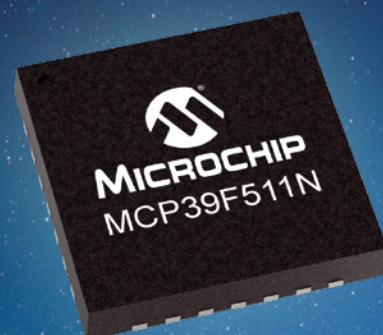
cod. EDUKIT02



Prezzi IVA inclusa.

ENERGY METER DUALE

di MATTEO DESTRO

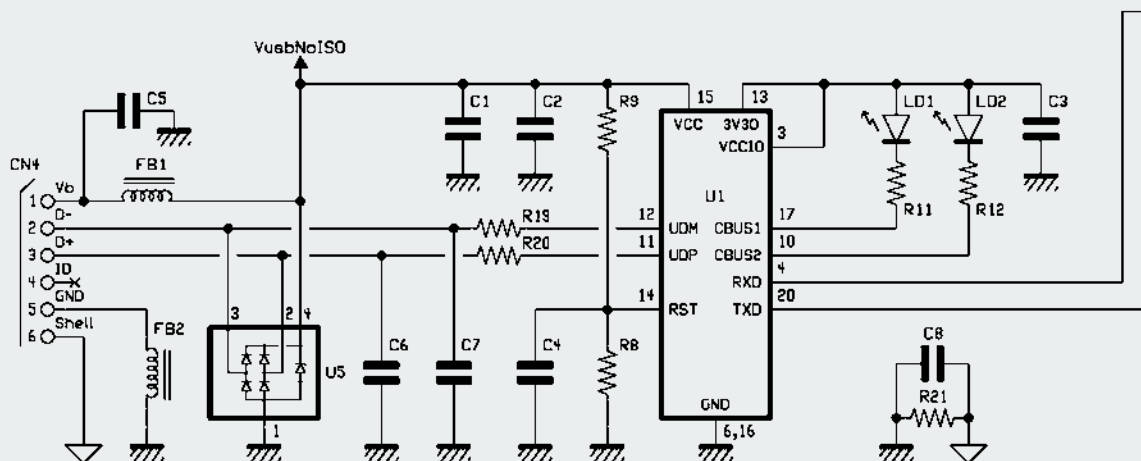
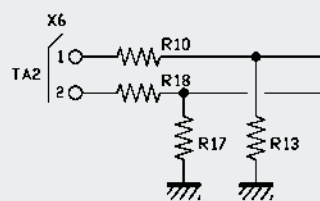
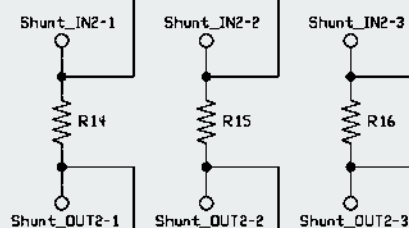
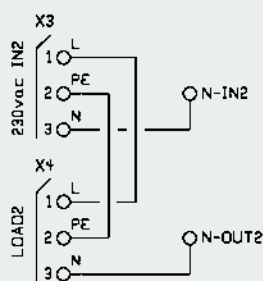
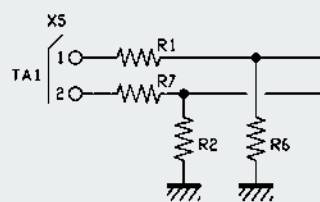
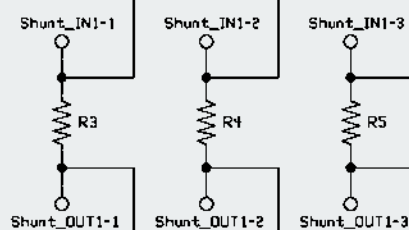
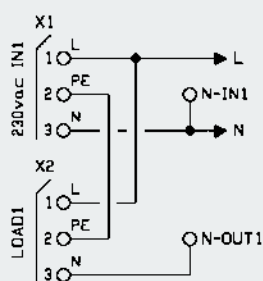
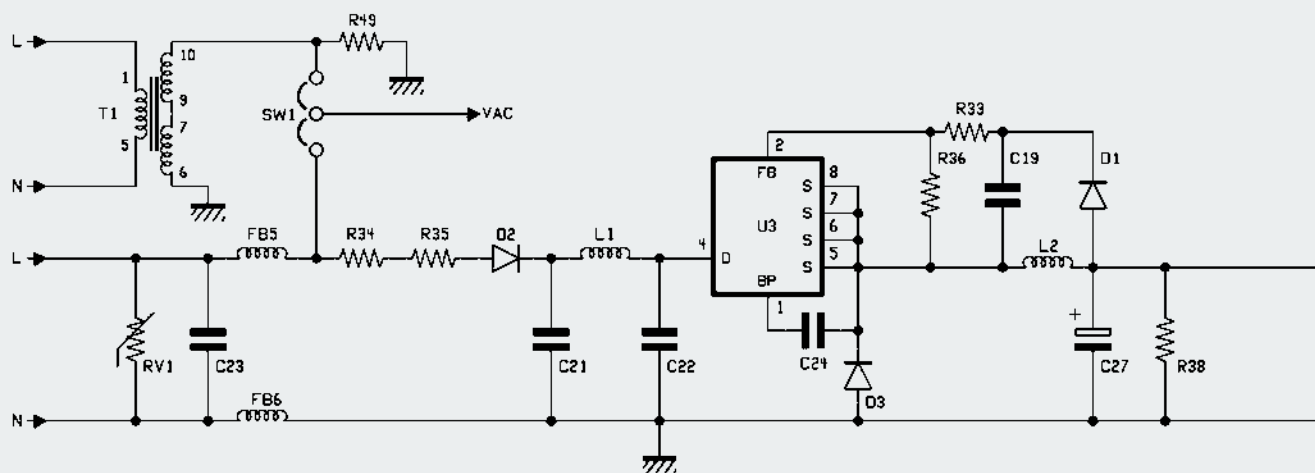


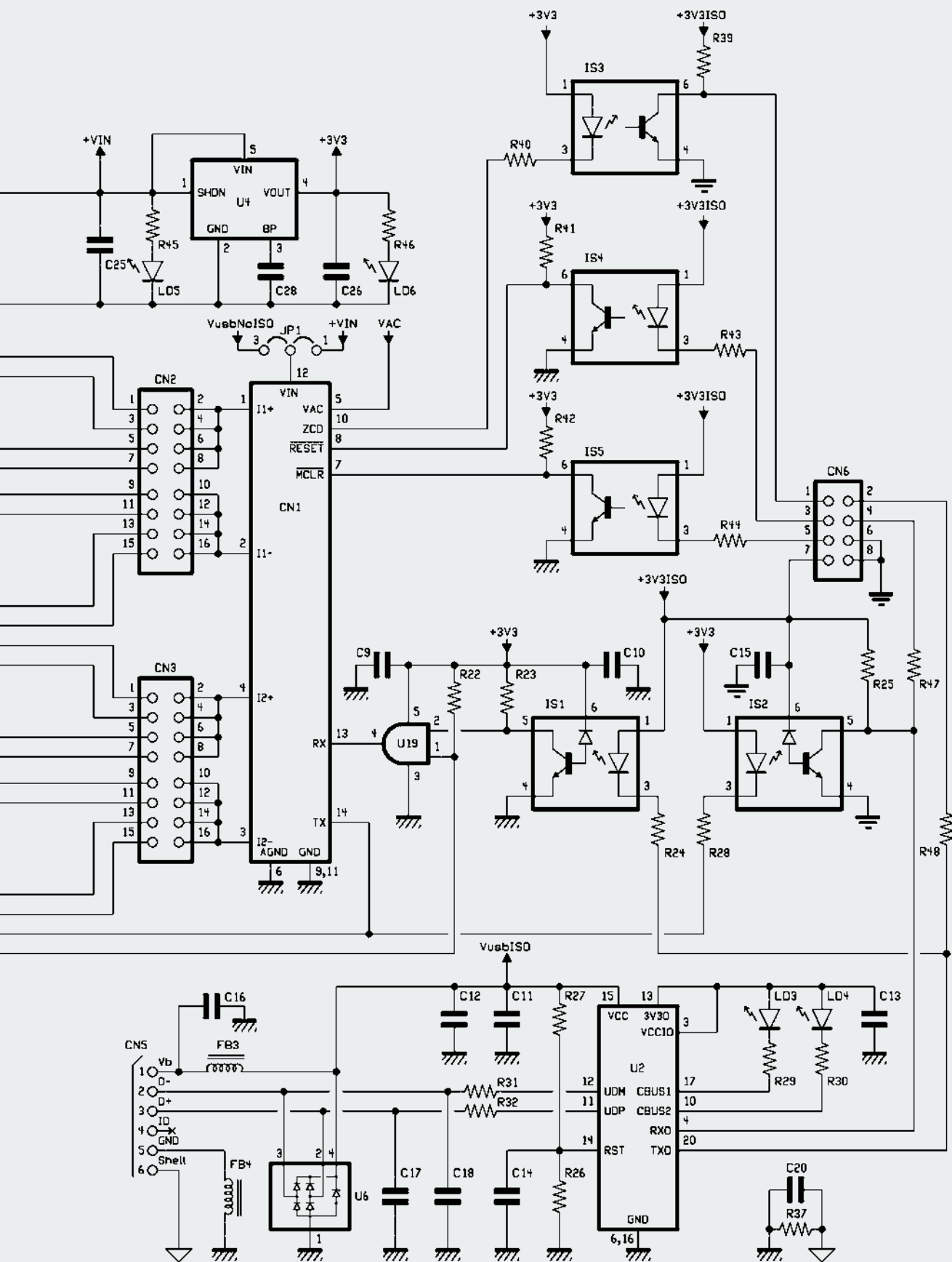
Analizziamo la Demo Board per l'Energy Meter progettata per facilitare connessioni alla rete elettrica e per fare analisi sui carichi grazie alla configurazione versatile e all'interfaccia USB per PC. Terza e ultima puntata.

N

el precedente articolo avevamo introdotto l'integrato MCP39F511N e la relativa schedina di supporto siglata FT1663M la quale permette di interfacciare l'integrato, tramite interfaccia UART, direttamente a un microcontrollore o se si desidera a un PC.

In questo numero proporremo una Demo Board da utilizzare per imparare a configurare e utilizzare l'integrato MCP39F511N nonché la schedina FT1663M. La Demo Board permetterà una certa flessibilità di configurazione al fine di poter utilizzare sia le configurazioni con TV e TA sia le configurazioni con misura diretta della tensione e della corrente tramite shunt resistivo.





La Demo Board si presenta con un fattore di forma di 122mm x 122mm sulla cui superficie sono distribuiti i componenti necessari alla connessione della stessa alla tensione di rete 230V e al collegamento dei due possibili carichi su cui eseguire le misure elettriche. Trova ovviamente posto anche la sezione a bassa tensione compreso di opto-isolamento dove necessario.

Sulla scheda sono presenti due interfacce μ USB, di cui una totalmente opto-isolata, alle quali possiamo collegare un PC, una Raspberry Pi o altro.

Sulla parte opto-isolata trova posto anche un connettore a 8 poli, passo 2.54mm, utile all'interfacciamento verso schede a microcontrollore.

SCHEMA ELETTRICO DEMO BOARD

Iniziamo la trattazione dello schema elettrico partendo dalle morsettiere X1, X2, X3 e X4 tutte da 3 poli.

La morsettiera X1 serve sia per alimentare l'elettronica della Demo Board direttamente dalla tensione di rete 230V che per portare la fase alla morsettiera X2 necessaria per collegare il carico 1. Quindi ai pin 1 delle morsettiere X1 e X2 abbiamo la fase mentre sui pin 3 di entrambe le morsettiere abbiamo il neutro. Il pin 2 serve per il collegamento della messa a terra. Le morsettiere X3 e X4, indipendenti dalle prime due, servono per la gestione del carico 2, quindi alla morsettiera X3 va portata la tensione di rete 230V come per la morsettiera X1 mentre alla morsettiera X4 va collegato il carico 2. La fase e il neutro, della morsettiera X1, vengono portati alla sezione di alimentazione la quale è stata realizzata sfruttando l'integrato LNK304DG-TL (U3) della Power Integrations. Con questa soluzione si realizza una alimentazione non isolata partendo dalla tensione di rete 230V.

Quindi analizzando lo schema si vede che, prima di arrivare all'integrato U3, tra la fase e il neutro è interposto un varistore RV1 seguito da un condensatore classe X2 nominato C23. Il varistore serve per filtrare eventi transitori di sovratensione. Il suo comportamento può essere rappresentato come un resistore non lineare che, superata la tensione di soglia, abbassa bruscamente la sua resistenza in modo che il transitorio venga fortemente attenuato. Il condensatore C23 serve invece per attenuare le emissioni EMI/RFI che possono essere causate dal rumore di modo differenziale causato dalle alimentazioni AC/DC.

Seguono, sia sulla fase che sul neutro, delle ferriti di filtro nominate FB5 e FB6 sempre per la soppressione delle interferenze EMI.

A questo punto la fase viene portata sia all'interruttore a leva SW1, per decidere se eseguire una misura diretta della tensione di rete o sfruttare il TV SVL10201 presente sulla Demo Board, sia alla sezione di ingresso dell'alimentazione. In quest'ultimo caso abbiamo due resistenze serie da 470R (R34 e R35) seguite da un diodo rettificatore D2 così da lavorare solo sulla semi onda positiva della fase. Segue quindi un filtro passa basso a π composto dai condensatori C21, C22 e dalla induttanza L1 la cui frequenza di taglio è data da:

$$f_t = \frac{1}{\pi\sqrt{LC}}$$

Che con i valori assegnati si attesta a circa 4.6kHz. A questo punto si può parlare dell'integrato U3 il quale, in un unico dispositivo, implementa un MOSFET che può lavorare ad alta tensione e una logica di controllo dell'alimentazione. Diversamente dai convenzionali controllori PWM questo usa un semplicissimo circuito di controllo ON/OFF per regolare la tensione di uscita. La **Fig. 1** mostra lo schema a blocchi interno dell'integrato il quale per funzionare correttamente necessita di:

- Oscillatore interno che lavora a una frequenza di 66kHz
- Sezione di feedback
- Sezione di bypass nel caso di alimentazione sotto soglia
- Protezione per temperatura di lavoro sopra soglia
- Frequency jettering
- Circuito limitatore di corrente
- MOSFET per alte tensioni fino a 700V

Di seguito facciamo un paio di considerazioni su alcuni blocchi funzionali dell'integrato LNK304. L'oscillatore interno è impostato per lavorare a una frequenza media di 66kHz; questo genera due segnali il primo è detto DC_{MAX} (massimo duty-cycle) e il secondo è il segnale di clock che indica l'inizio di ogni ciclo. L'oscillatore incorpora anche una sezione di circuito che introduce una piccola quantità di jitter, tipicamente intorno ai 4kHz picco-picco, per minimizzare le emissioni EMI.

La **Fig. 2** mostra quanto appena detto.

La sezione di feedback interna, collegata al pin FEEDBACK dell'integrato, serve per gestire in modo corretto l'accensione del MOSFET per ogni ciclo di clock (gestione sul fronte di salita).

Quindi se la corrente che entra nel pin è inferiore a 49 μ A il MOSFET viene acceso sul fronte di salita del clock altrimenti rimane spento fino al prossi-

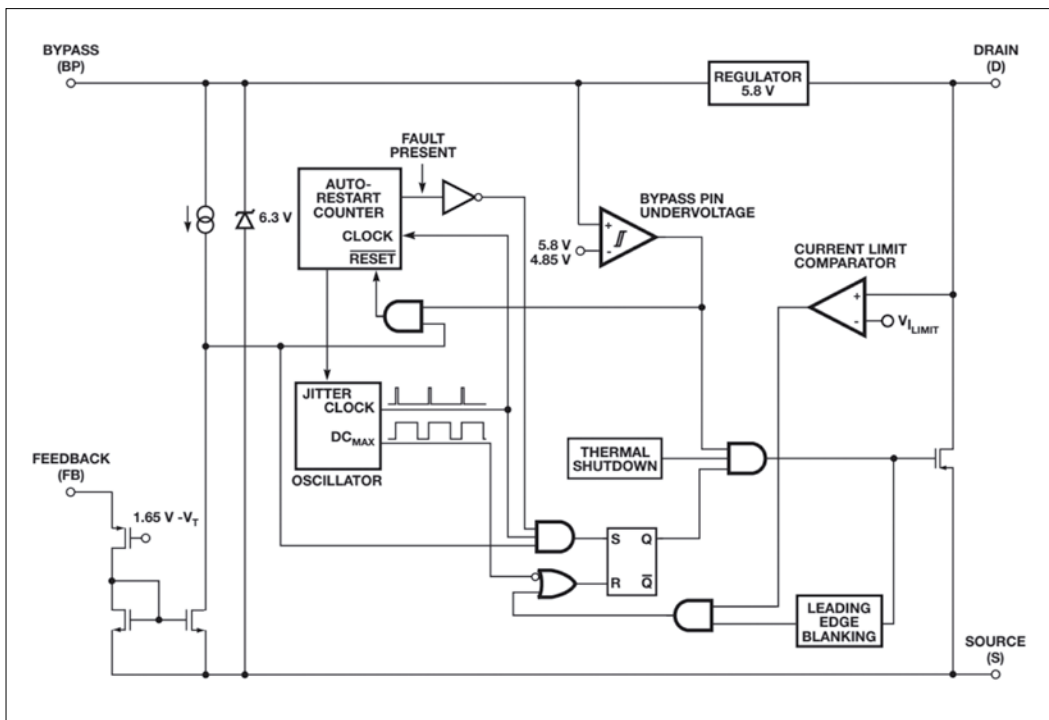


Fig. 1
Schema a
blocchi inte-
grato LNK304.

mo ciclo di clock. Configurando correttamente le resistenze R_{BIAS} e R_{FB} si va a impostare la tensione di uscita V_O desiderata.

La formula da applicare per ottenere la tensione di uscita V_O è la seguente:

$$R_{FB} = \frac{(V_O - V_{FB}) * R_{BIAS}}{V_{FB} + I_{FB} * R_{BIAS}}$$

Dove V_O desiderata è 12V, R_{BIAS} è settata a 2K, V_{FB} è fissa a 1,65V e la corrente I_{FB} è anch'essa un valore fisso a 49uA. Quindi sostituendo i valori nella formula si ottiene che R_{FB} è pari a 11K842. Ovviamente si sceglie il valore più prossimo che è 12K. Date le due resistenze appena impostate possiamo applicare la formula inversa e vedere a quanto ammonta la tensione di uscita:

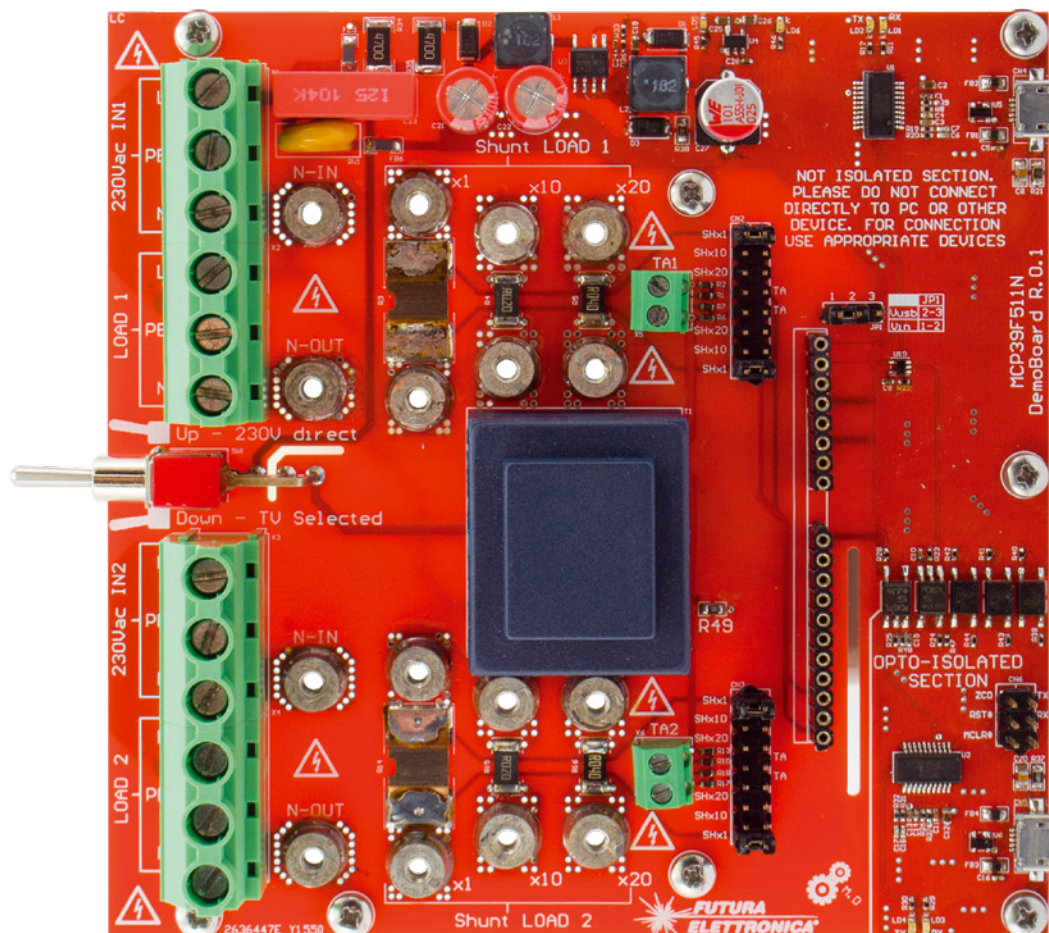
$$V_O = R_{FB} * \left(\frac{V_{FB} + I_{FB} * R_{BIAS}}{R_{BIAS}} \right) + V_{FB}$$

Inserendo i valori calcolati si ottiene che la tensione di uscita è pari a 12,13V.

Per quanto riguarda il limitatore interno di corrente si ha che questo monitorizza la corrente che passa nel MOSFET e quando questa corrente supera I_{LIMIT} (valore intorno ai 300mA) il MOSFET stesso viene spento fino alla fine del ciclo in corso.

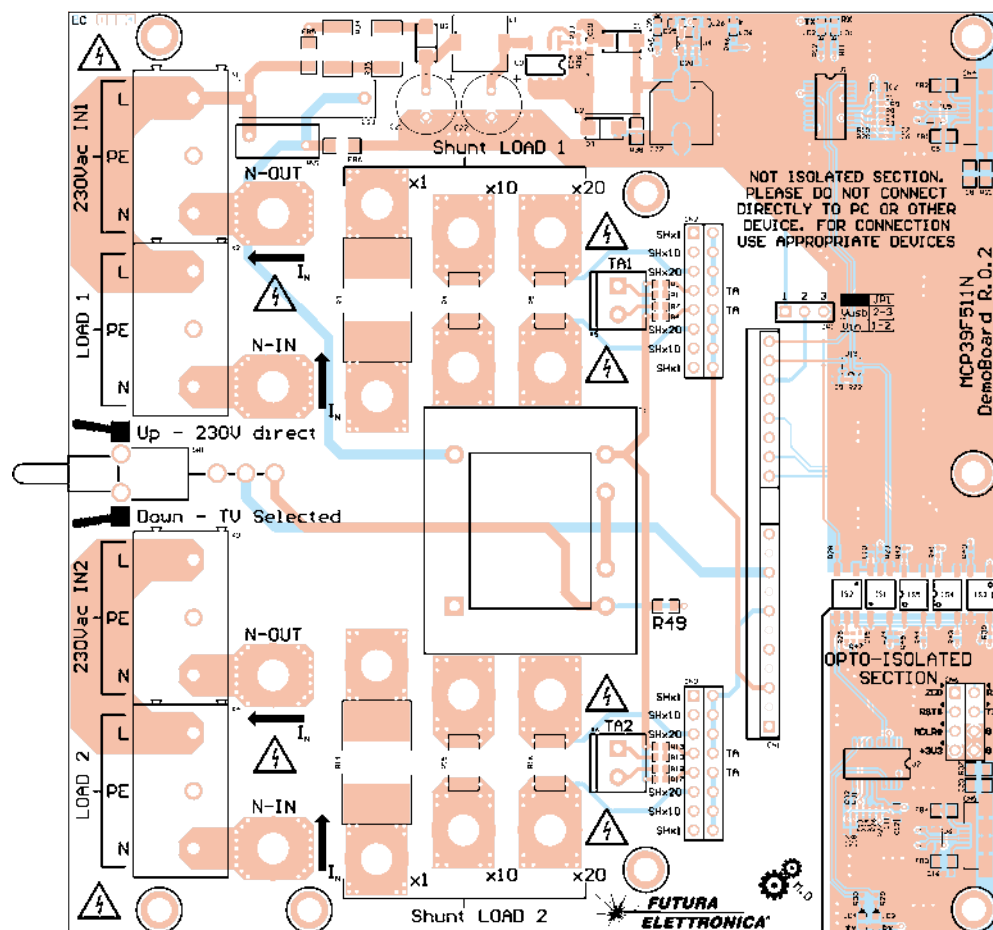
Il circuito limitatore di corrente ha anche una sezione detta "**leading edge blanking**" la quale inibisce il comparatore di corrente per un breve periodo di tempo detto t_{LEB} (valore tipico impostato a 200 nS circa) dopo che il MOSFET è stato acceso.

Ora prestiamo attenzione al diodo D3 detto di "**freewheeling**" il quale deve essere della famiglia ultra-fast con un tempo di recupero uguale o inferiore ai 75nS. Quello utilizzato nello schematico è un ES1G-13-F il quale ha un tempo di recupero di soli 25nS assolutamente compatibile con la nostra applicazione. Segue una induttanza L2 da 1mH da 200mA e un condensatore C27 detto di filtro da 100uF la cui funzione primaria è limitare il ripple sulla tensione di uscita. La tensione di ripple in uscita è fortemente dipendente dalla ESR del condensatore più che dalla sua capacità. Infine sull'uscita è prevista una resistenza R38 detta di pre-carico da 3K3 per mantenere regolata la tensione di uscita quando la corrente assorbita dal carico è inferiore a 3mA. La resistenza viene quindi scelta affinché ci sia sempre un assorbimento maggiore di 3mA. Nel nostro caso è di 3,6mA. La tensione di alimentazione di +12V viene identificata nello schematico come +V_{IN}, tale alimentazione viene portata sia a un secondo regolatore di tensione LDO per generare i +3V3 necessari al funzionamento della Demo Board sia alla schedina



Elenco Componenti:

R1, R7, R10, R18: 16.9 ohm	R5, R16: 0.04 ohm
R19, R20, R31, R32: 22 ohm	R34, R35: 470 ohm
R47, R48: 100 ohm	R3, R14: 0.002 ohm
R24, R28, R40, R43, R44: 180	RV1: Varistore 430V 3.5 kA cod. 820412711
R36: 2 kohm	C5, C16: 10 nF 50 VL ceramico (0402)
R22, R23, R25, R39, R41, R42: 2.2 kohm	C6, C7, C17, C18: 47 pF 16 VL ceramico (0402)
R11, R12, R29, R30, R46: 3.3 kohm	C4, C14: 220 pF 16 VL ceramico (0402)
R9, R27, R45: 4.7 kohm	C1, C3, C9, C10, C11, C13, C15, C24, C28
R8, R26: 10 kohm	100 nF 50 VL ceramico (0402)
R33: 12 kohm	C2, C12, C19, C25, C26: 10 µF 25 VL ceramico (0603)
R2, R6, R13, R17: 8.2 ohm	C8, C20: 1 nF 1 kV ceramico (0805)
R38: 3.3 kohm	C27: 10 µF 25 VL elettrolitico (F)
R49: 10 kohm	C21, C22: 4.7 µF 400 VL ceramico
R21, R37: 1 Mohm	C23: 100 nF 275 VL ceramico
R4, R15: 0.02 ohm	C27: 100 µF 25 VL elettrolitico



D1, D2: MRA4005T1G
 D3: ES1G
 LD1, LD3: LED Giallo (0603)
 LD2, LD4, LD5, LD6: LED Rosso (0603)
 U1, U2: FT231XS-U
 U3: LNK304DG-TL
 U4: LK112M33TR
 U5, U6: PRTR5V0U2X
 U19: 74LVC1G08GW,125
 FB1, FB2, FB3, FB4: EMI Suppression
 BLM21SP111BH1D
 FB5, FB6: Chip Beads per power line
 (0R150@800mA)
 IS1, IS2: PC457L0NIP0F
 IS3, IS4, IS5: TLP185(SE)

L1, L2: 1mH@0A2
 SW1: Microswitch 108-0020-EVX
 T1: SVL101201
 JP1: Pin Strip Maschio 3 vie passo 2.54 mm
 CN6: Pin Strip Maschio 4x2 vie passo 2.54 mm
 CN2, CN3: Pin Strip Maschio 8x2 vie
 passo 2.54 mm (2 pz.)
 CN4, CN5: Micro USB Type A/B 10104111-0001LF (2 pz.)
 CN1: Breakout Board MCP39F511N
 X1, X2, X3, X4: Morsetto 3 poli pasos 7.62 mm (4 pz.)
 X5, X6: Morsetto 2 poli pasos 3.81 mm (2 pz.)

Varie:
 - Circuito Stampato S1744 (118x118mm)

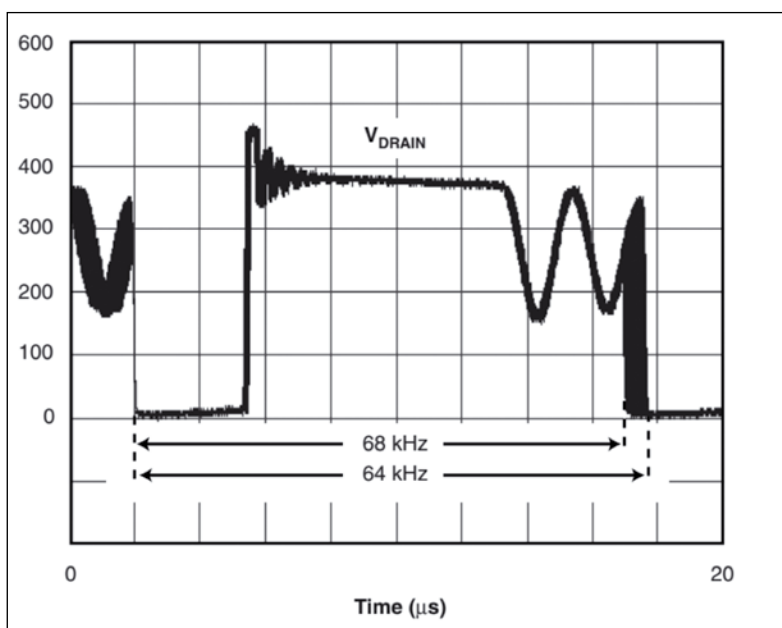


Fig. 2
LNK304
clock jitter.

di misura FT1663M siglata come CN1.

Il regolatore di tensione secondario è siglato U4 ed è un LK112M33TR della ST Microelectronics il quale, come detto prima, è un LDO a basso rumore con funzione di shutdown integrata in case SOT23-5L. La massima corrente erogabile in uscita è di 200mA più che sufficiente per la nostra applicazione.

Come si vede dallo schematico la quantità di componenti che necessita per il suo funzionamento è davvero esigua; infatti sono presenti due condensatori da 10uF/25V (C25 e C26), uno sull'ingresso e uno sull'uscita e un condensatore C28 da 100nF/50V sul pin 3 (BP) detto di bypass che serve per mitigare il rumore come evidenziato dalla **Fig. 3**.

Come si vede dall'immagine il rumore è riportato sulle ordinate del grafico, ed è espresso in

$$\frac{nV}{\sqrt{Hz}}$$

mentre sulle ascisse viene riportata la frequenza in kHz. A frequenze più basse, il rumore risulta molto più alto in assenza del condensatore di bypass rispetto alla stessa condizione con il condensatore montato. Il pin 1 di shutdown, attivo basso, viene collegato direttamente a +V_{IN} cosicché l'integrato sia sempre attivo. Concludendo la sezione di alimentazione abbiamo due led dove il primo indica la presenza della +V_{IN} (LD5) mentre il secondo indica la presenza della tensione di alimentazione +3V3 (LD6), entrambi sono di colore rosso.

Analizziamo ora la sezione di misura di tensione e corrente. Per misurare la tensione di rete ci sono due opzioni: la prima è quella diretta tramite partitore resistivo. In questo caso si deve impostare lo switch SW1 sui contatti 1-2, di conseguenza la tensione di rete 230V verrà portata direttamente al pin 5 del connettore CN1. Questo vuol dire che la scheda di misura FT1663M deve essere impostata per accogliere la tensione di rete 230V e quindi i jumper JP2 e JP3, presenti sulla scheda, **DEVONO** essere **TASSATIVAMENTE** impostati sui contatti 2 e 3. Se così non fosse l'elettronica della scheda di misura verrà danneggiata in modo irreparabile.

La seconda opzione prevede di sfruttare il TV SVL101201 della Itacoil (T1), in questo caso lo switch SW1 deve essere impostato sui contatti 2-3. Di conseguenza i jumper JP2 e JP3 sulla scheda di misura dovranno essere impostati sui contatti 1 e 2. In questo caso se si sbaglia a impostare i jumper sulla scheda non succede nulla di irreparabile, semplicemente vedrete delle misure di tensione errate nulla di più.

Il secondario del TV è configurato per ritornare un valore pari a 10V nel caso in cui in ingresso ci siano 230V. Il range di tensioni di ingresso per misure di precisione va da 100V a 275V mentre il range consentito va da 0V a 300V. La **Fig. 4** mostra l'errore massimo che introduce il TV in quattro possibili casi. L'errore tipico è sempre sotto il $\pm 0.50\%$.

Per quanto riguarda le misure di corrente, ricordiamo che possiamo farne due indipendenti, anche in questo caso abbiamo due possibilità: la prima è una misura diretta tramite shunt resistivo sul neutro del carico mentre la seconda viene fatta utilizzando un TA SBT002 o SBT005 della Itacoil. Le misure di corrente fatte con lo shunt resistivo sono previste sul neutro per entrambi i carichi collegati e come resistenza di shunt si è usata una resistenza da 0R002/8W (R3 e R14) per una corrente massima misurabile di circa 60A_{RMS}.

Tuttavia le morsettiere usate per la Demo Board consentono una corrente massima al morsetto di 30A_{RMS} quindi se si volesse arrivare ai 60A_{RMS} si dovrebbero usare dei collegamenti diversi verso il carico bypassando le morsettiere (possibilità non prevista dal progetto della Demo Board). Ricordiamo tuttavia che anche le piste di collegamento sulla scheda sono dimensionate per i 30A_{RMS}.

Oltre allo shunt sopra menzionato sono stati predisposti altri due valori di shunt per simulare correnti superiori a 30A_{RMS} ovvero uno da 0R02/3W (R4 e R15) e l'altro da 0R04/3W (R5 e R16).

Per chiarire lo scopo di questi ulteriori shunt proviamo a fare un esempio: supponiamo di volere misurare una corrente di 10A_{RMS} e quindi di circa 14A_{PICCO}

- se si usasse lo shunt R3 si avrebbe che la caduta di tensione sulla resistenza sarebbe pari a 28.284mV
- Ora se invece di usare lo shunt R3 si usasse lo shunt R4 da 0R02 si avrebbe che la caduta di tensione sulla resistenza sarebbe pari a 282.842mV che equivale a fare un x10 sul valore reale di corrente e quindi la nostra schedina di misura ritornerebbe un valore di corrente moltiplicato appunto x10
- Se invece si opta per la resistenza R5 da 0R04 si avrebbe che la caduta di tensione sulla resistenza sarebbe pari a 565.685mV che equivale a fare un x20 sul valore reale di corrente e quindi la nostra schedina di misura ritornerebbe un valore di corrente moltiplicato appunto x20

Questo trucco è usato per testare la schedina FT1663M sulle misure di correnti elevate anche se in realtà la corrente assorbita dal carico è molto più bassa. Ovviamente questo trucchetto oltre a moltiplicare il valore di corrente misurato moltiplica anche l'errore di misura e quindi non si ottiene mai un x10 o x20 "pulito" ma con una certa tolleranza accettabile per lo scopo.

Altra considerazione riguarda la potenza delle resistenze di shunt ausiliarie che sono da 3W; con una corrente di 10A_{RMS} la potenza dissipata da R3 diventa 2W, pienamente dentro il range consentito, mentre per la R5 diventa 4W che è sopra il valore massimo. Quindi per R5 la massima corrente tollerabile sarà di circa 8,5A_{RMS} per una potenza di 2,89W. La **Fig. 5** mostra i tre possibili collegamenti degli shunt resistivi per la misura della corrente del carico 1. In particolare abbiamo che:

- In azzurro inserimento shunt resistivo da 0R002/8W (R3). Per chiudere il collegamento verso la schedina di misura i due jumper sulla destra devono essere inseriti sulla posizione **Shx1** come indicato dalla serigrafia sul PCB
- In rosso inserimento shunt resistivo da 0R02/3W (R4). Per chiudere il collegamento verso la schedina di misura i due jumper sulla destra devono essere inseriti sulla posizione **Shx10**
- In giallo inserimento shunt resistivo da 0R04/3W (R5). Per chiudere il collegamento verso la schedina di misura i due jumper sulla destra devono essere inseriti sulla posizione **Shx20**

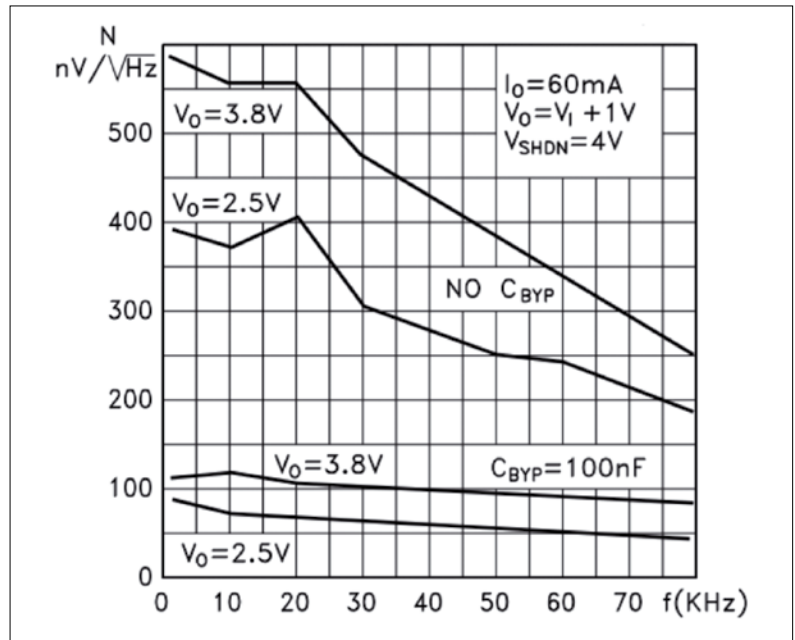
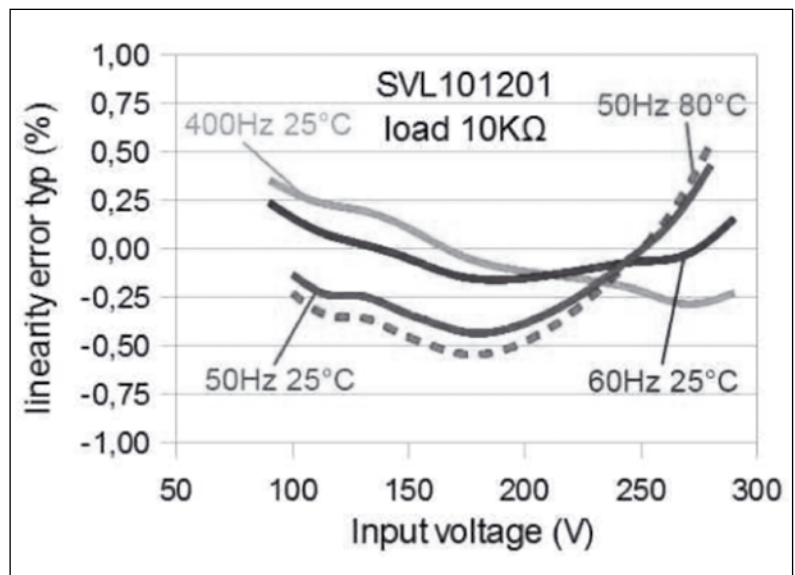


Fig. 3
Noise Spectrum
con e senza
condensatore di
bypass da 100nF.

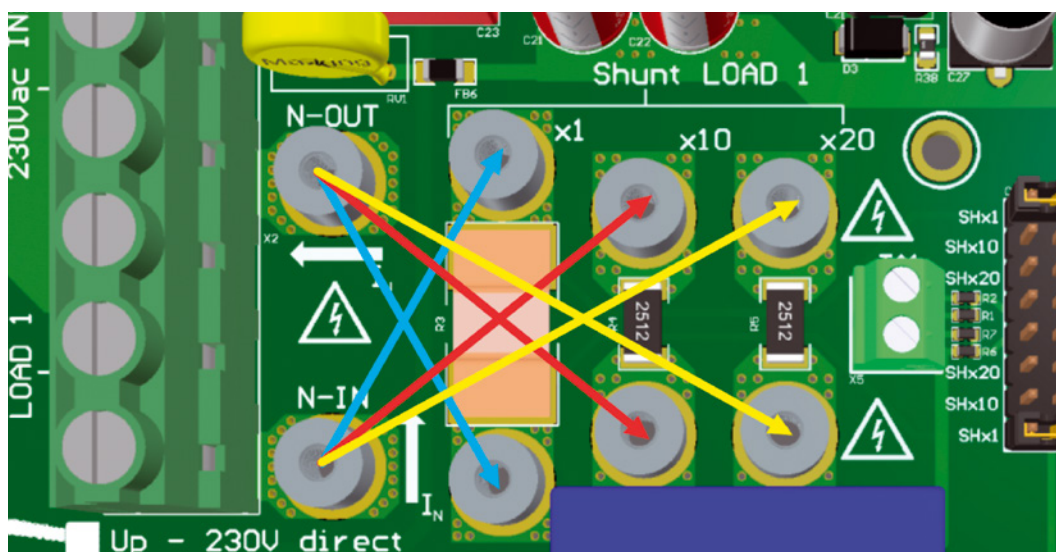
Come si può vedere dalla figura i collegamenti elettrici per ogni shunt resistivo sono di tipo incrociato questo perché la corrente di neutro arriva dal carico (N-IN) e va verso il neutro della morsettiera di alimentazione (N-OUT). Quindi dato che la misura di corrente è di tipo differenziale si ha che il polo positivo di misura I⁺ è in alto rispetto allo shunt e il polo negativo I⁻ è in basso.

Da qui nasce la necessità di incrociare i collegamenti. Questa logica di collegamento vale anche per gli shunt resistivi del carico 2.

Fig. 4
Errore di
linearità
SVL101201.



➔ **Fig. 5**
Collegamento degli
shunt resistivi per il
carico 1.



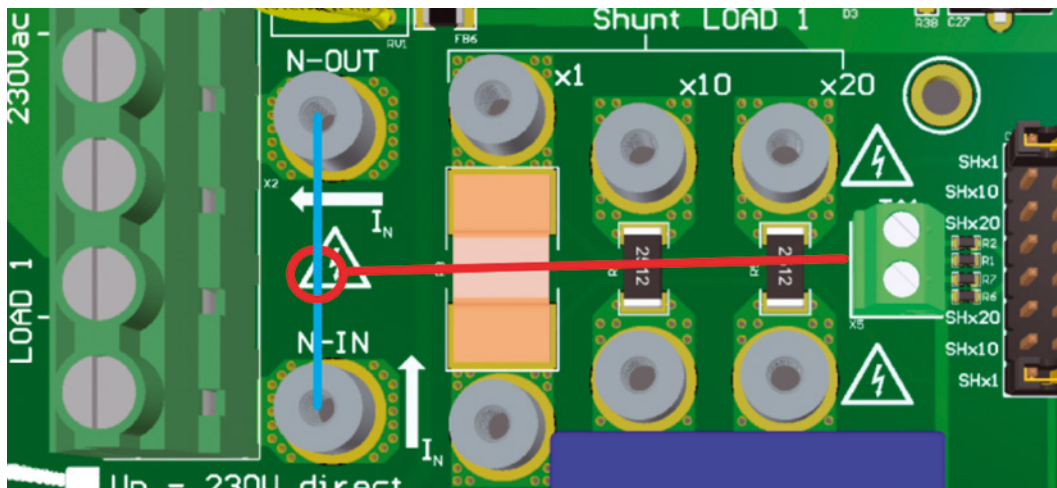
I collegamenti elettrici degli shunt di corrente al neutro vengono fatti tramite cavo elettrico crimpato inserito su appositi connettori WURTH codice 7466003R facenti parte della famiglia WP-SMBU da 50A massimi. Il cavo crimpato viene collegato al connettore tramite vite di fissaggio rendendo agevole il collegamento di uno shunt rispetto a un altro.

Se invece si vuole eseguire la misura di corrente sfruttando il TA Itacoil SBT002 o SBT005 si deve eseguire il collegamento come indicato dalla **Fig. 6** anche in questo caso si fa uso di cavo crimpato come per gli shunt. Come si può vedere in azzurro il collegamento del carico 1 sul neutro mentre in rosso il collegamento del TA ovvero il primario (primario rappresentato da un cerchio rosso) sul neutro e il secondario sulla morsettiere a destra.

In questo caso i jumper devono essere entrambi inseriti nella posizione **TA**. Come si vede alla destra della morsettiere ci sono le resistenze di Burden che nello schematico sono indicate con R1, R2, R6 e R7. Tutto quello che è stato scritto per il TA del carico 1 è ovviamente valido anche per il TA del carico 2.

I collegamenti appena descritti, sia alle morsettiere di alimentazione che per gli shunt di corrente e i TA, devono essere fatti in totale sicurezza e quindi con la tensione di rete scollegata! Anche la movimentazione dei jumper sopra descritta deve essere assolutamente fatta con la tensione di rete scollegata. I segnali appena descritti vengono quindi portati al connettore CN1 nel quale viene inserita la schedina di misura FT1663M; questa ha un unico verso di inserimento dovuto al fattore di forma della scheda stessa.

➔ **Fig. 6**
Collegamento
del TA al carico 1.



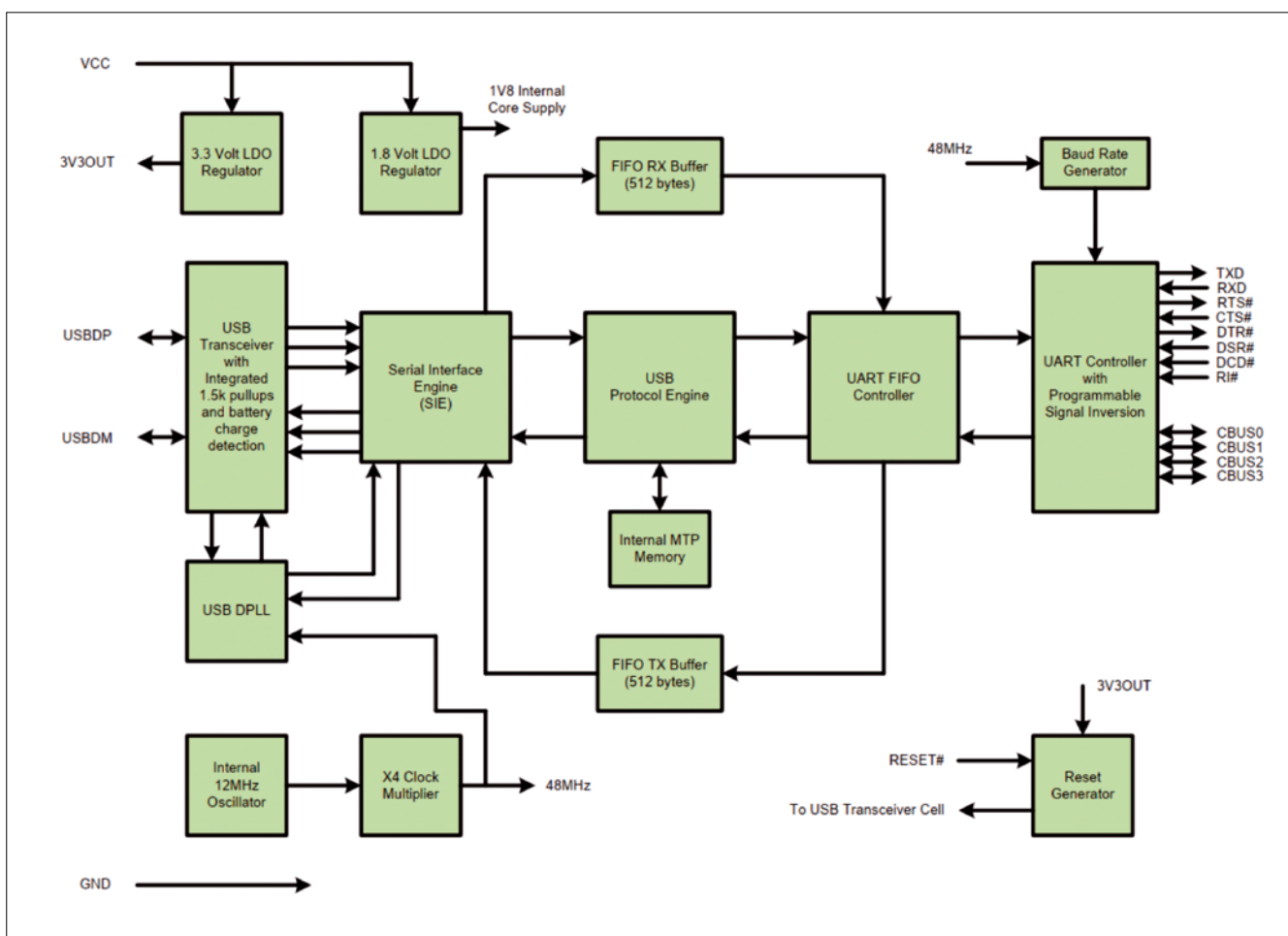


Fig. 7
Schema
a blocchi
FT231XS-U.

Al pin 12 del connettore viene portata l'alimentazione e questa, tramite il jumper JP1, può essere scelta tra quella generata dall'integrato U3 (+V_{IN}), pari a circa +12V, e la tensione di alimentazione fornita dal connettore µUSB CN4 pari a +5V. Sulla Demo Board sono state previste due connessioni dati tramite convertitore RS232/USB, una totalmente isolata mentre l'altra no. I +5V di alimentazione di cui sopra provengono dalla connessione **NON** elettricamente isolata ("**Vusb_Nolso**").

La sezione di comunicazione non isolata viene gestita dall'integrato U1, FTDI FT231XS-U, il quale semplicemente converte il segnale TTL RS232 proveniente dal connettore CN1 in un segnale USB 2.0 Low Speed per la connessione a un PC, RaspberryPi o altro. Se collegato a un PC, con sistema operativo Windows™, è possibile interfacciare la schedina di misura con il nostro software "**EnergyMeter R.1.1.0.0**", o versioni successive, col quale si possono eseguire tutte le configurazioni

dell'integrato MCP39F511N presente sulla scheda di misura nonché visualizzare tutte le misure elettriche in corso.

Se si opta per un collegamento non elettricamente isolato si devono prendere alcune precauzioni tra cui usare il TV e il TA per le misure e interporre una interfaccia di isolamento tra la scheda demo e il PC. In caso contrario il PC da collegare **DEVE** essere messo in sicurezza tramite una connessione isolata elettricamente. Sulla Demo Board è comunque predisposto un secondo collegamento tramite interfaccia µUSB, connettore CN5, il quale sfrutta sempre un integrato FTDI FT231XS-U (U2). In questo caso le linee TX e RX della RS232 vengono opto-isolate tramite IS1 e IS2. Anche le linee ZCD (Zero Cross Detect), RESET e MCLR vengono opto-isolate rispettivamente tramite IS3, IS4, IS5 e portate al connettore CN6.

Questo viene utile nel caso si voglia collegare una scheda elettronica a microcontrollore per gestire la nostra schedina di misura.

Al connettore CN6 vengono portate anche le linee TX e RX a livello TTL.

Quindi:

Pin 1: ZCD_ISO (Microcontrollore ← FT1663M)
 Pin 2: TX_ISO (Microcontrollore → FT1663M)
 Pin 3: RESET_ISO (Microcontrollore → FT1663M)
 Pin 4: RX_ISO (Microcontrollore ← FT1663M)
 Pin 5: MCLR_ISO (Microcontrollore → FT1663M)
 Pin 6: GND
 Pin 7: +3V3
 Pin 8: GND

La freccia indica la direzione del segnale.

Le due interfacce µUSB, partendo dai connettori CN4 e CN5, vengono entrambe protette dalle possibili sovratensioni dovute a delle scariche elettrostatiche tramite l'integrato PR-TR5V0U2X,215 (U5 e U6), sono semplicemente dei diodi rail-to-rail, mentre sulle linee di alimentazione vengono interposte in serie delle ferriti di filtro tipo BLM21SP111BH1D. Quindi le due linee differenziali D+ e D- vengono portate, tramite resistenza serie da 22R, ai pin 11 e 12 dell'integrato FT231XS-U. Ovviamente CN4 è associato a U1 mentre CN5 a U2.

Se il cavo µUSB non è inserito, discorso valido per entrambi gli integrati, si ha che il convertitore RS232/USB viene mantenuto in stato di reset in quanto c'è un pull-down sul pin 14 che lo forza a

massa (infatti il pin di RESET è attivo basso).

Nel momento in cui viene inserito un cavo µUSB in uno dei due connettori, si ha che l'integrato FT231XS-U viene tolto dalla condizione di reset perché la tensione Vbus, presente sul connettore, viene portata tramite partitore resistivo al pin 14 dell'integrato ma anche al pin 15 di alimentazione accendendone quindi l'elettronica.

La Fig. 7 mostra lo schema a blocchi dell'integrato dal quale si nota il regolatore LDO interno che genera i +3V3 sul pin di uscita 13 detto 3V3OUT. Questa tensione viene quindi riportata sul pin di ingresso 3 detto VCCIO il quale determina a quale tensione devono lavorare i pin di I/O come ad esempio TX e RX della seriale.

Sempre dallo schema a blocchi è possibile vedere il secondo LDO necessario a generare la tensione interna di +1V8 necessaria ad alimentare l'elettronica dell'integrato.

Ai pin 10 (CBUS2) e 17 (CBUS1) sono collegati, tramite resistenza serie da 470R, rispettivamente il led rosso che indica la trasmissione dei dati e il led giallo che ne indica la ricezione.

Tutti gli altri pin di I/O non sono utilizzati e quindi non connessi.

Dato che abbiamo messo due interfacce di comunicazione abbiamo quanto segue:

- Il segnale **TX_MCP** che esce dal pin 14 del connettore CN1 viene portato a entrambi i pin 4 dei due integrati U1 e U2 con la differenza che per U2 prima viene interposto un opto-isolatore IS2
- Il segnale **RX_MCP** che entra nel pin 13 del connettore CN1 arriva dalla porta logica AND (U19) in quanto tale segnale viene generato dalla combinazione logica dei segnali **RX_NO_ISO** e **RX_ISO** uscenti dai pin 20 dei due integrati U1 e U2. Il segnale **RX_ISO** prima di raggiungere la porta logica vede interposto l'opto-isolatore IS1
- Se invece si ipotizza di usare il connettore CN6, a cui si connette una apposita elettronica a microcontrollore, allora entrambi i cavi µUSB devono essere scollegati

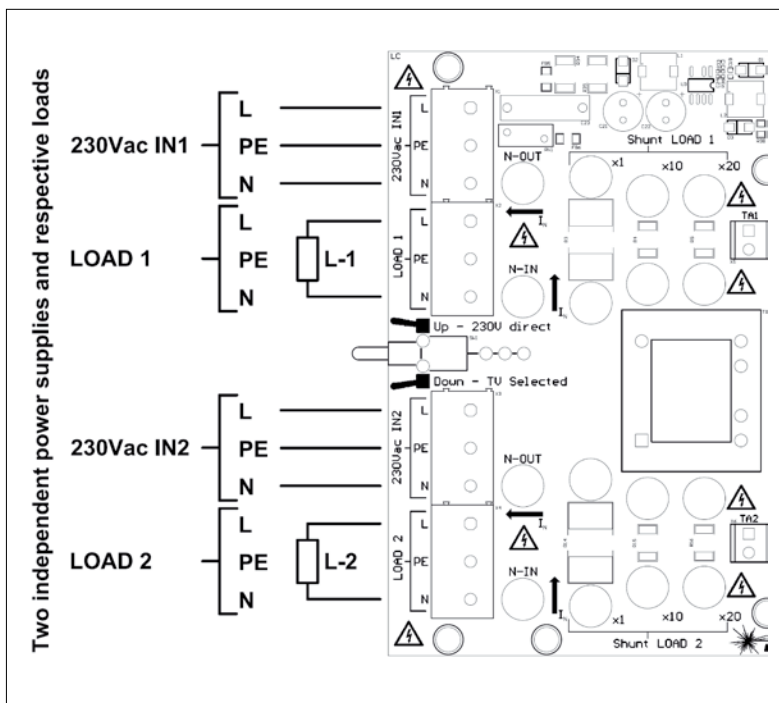
Con questa ultima osservazione riteniamo conclusa la trattazione dello schema elettrico della Demo Board.

SCHEMA ELETTRICO SCHEDA DI ESPANSIONE

Con questa Demo Board è possibile utilizzare anche la schedina di misura FT1346M, presentata sul n° 220 di Elettronica In, previo utilizzo apposito PCB adattatore.

Lo schema elettrico è a dir poco essenziale infatti

Fig. 8
Alimentazione
doppia per
carico 1 e 2.



prevede due connettori CN1 e CN3 per innestarsi nella Demo Board e un connettore CN2 per inserire la schedina di misura FT1346M con montato l'integrato MCP39F511.

Sulla scheda in esame trova posto anche un regolatore di tensione LDO LK112M50TR che partendo dalla tensione di alimentazione $+V_{IN}$ genera la tensione stabilizzata +5V.

Ovviamente per utilizzare FT1346M si devono prendere le seguenti precauzioni:

- Si possono eseguire le misure elettriche solo sul carico 1
- È obbligatorio usare il TV per le misure di tensione e il TA per le misure di corrente

Detto ciò, un volta fatti correttamente i collegamenti elettrici e inserita la scheda di espansione, potrete iniziare a parametrizzare l'integrato MCP39F511 con il nostro software Energy Meter Rev.1.1.0.0 o superiore.

Anche se è obbligatorio utilizzare il TV e il TA per le misure elettriche è comunque consigliato collegare il cavo uUSB al connettore CN5. Ovviamente è possibile anche sfruttare le connessioni presenti sul connettore CN6 per il collegamento di una scheda a microcontrollore.

Il connettore che ospita la schedina FT1346M deve essere montato sul Bottom Layer del PCB mentre sul Top Layer trovano posto i pochi componenti SMT utilizzati.

Il fattore di forma dato al PCB serve per non interferire con il TV montato sulla DemoBoard.

DESCRIZIONE PCB DI SUPPORTO PER I TA SBT002 E SBT005

Per dare un supporto meccanico ai due TA, SBT002 e SBT005, abbiamo progettato un piccolo PCB cosicché sia più facile collegare gli stessi alle apposite morsettiere due poli presenti sul PCB della DemoBoard (Morsettiere X5 per il TA del carico 1 e morsettiere X6 per il TA del carico 2). Dal punto di vista elettrico semplicemente si collegano i due terminali del secondario del TA ai due poli di un connettore berg passo 2.54mm 90°. Questo è valido sia per SBT002 che per SBT005. Per quanto riguarda i PCB siamo stati costretti a progettarne due in quanto le dimensioni meccaniche dei due TA sono incompatibili tra loro. Nel primario del TA si può fare passare sia la fase che il neutro. Con la nostra DemoBoard, supponendo di monitorare due carichi distinti e indipendenti, abbiamo la misura della corrente sul neutro quindi come da disegno inserire il filo come

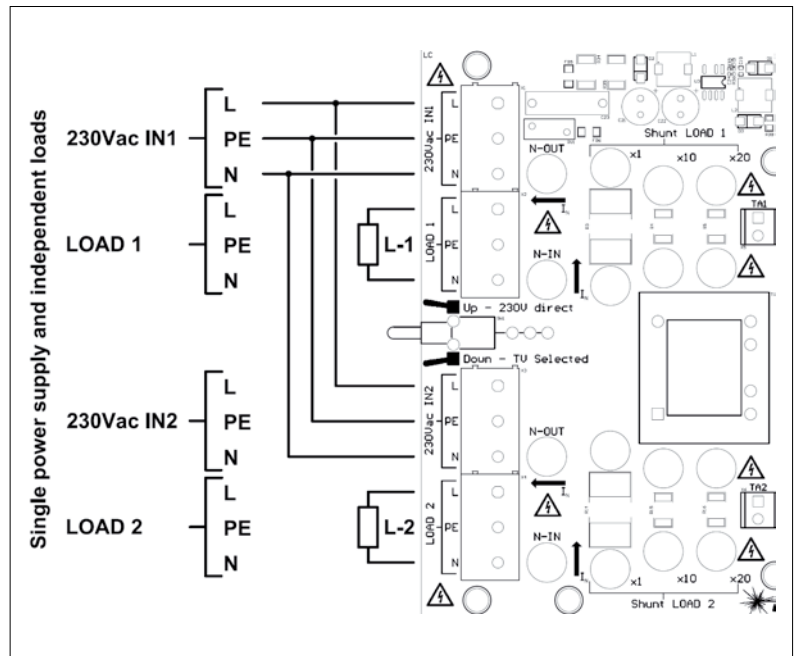


Fig. 9
Alimentazione
singola per
carico 1 e 2.

indicato affinché la corrente fluisca con il verso indicato. Nel caso in cui leggeste una potenza attiva negativa invertire il verso del filo di neutro o girare i collegamenti alla morsettiere X5 o X6.

COLLEGAMENTI ELETTRICI ALLA MORSETTIERA

Vediamo ora come effettuare i collegamenti elettrici alle morsettiere X1, X2, X3 e X4.

La **Fig. 8** mostra come eseguire i collegamenti elettrici al fine di alimentare i due carichi in modo indipendente. Quindi la tensione di rete collegata alla morsettiere X1 serve sia per alimentare la Demo Board sia per alimentare il carico 1 mentre la tensione di rete collegata alla morsettiere X3 serve unicamente per alimentare il carico 2.

Con questo tipo di collegamenti le misure di corrente possono essere fatte sia sfruttando gli shunt resistivi sia i TA.

La **Fig. 9** mostra una singola alimentazione di rete portata sia alla morsettiere X1 che X3.

Prestare attenzione ai collegamenti di fase e neutro sulle rispettive morsettiere.

Anche in questo caso è possibile eseguire le misure di corrente sia con gli shunt resistivi che con i TA.

La **Fig. 10** mostra come eseguire i collegamenti per una misura differenziale della corrente. In questo caso la tensione di rete viene portata solo alla morsettiere X1.

Sfruttando poi le morsettiere X2, X3 e X4 si crea un

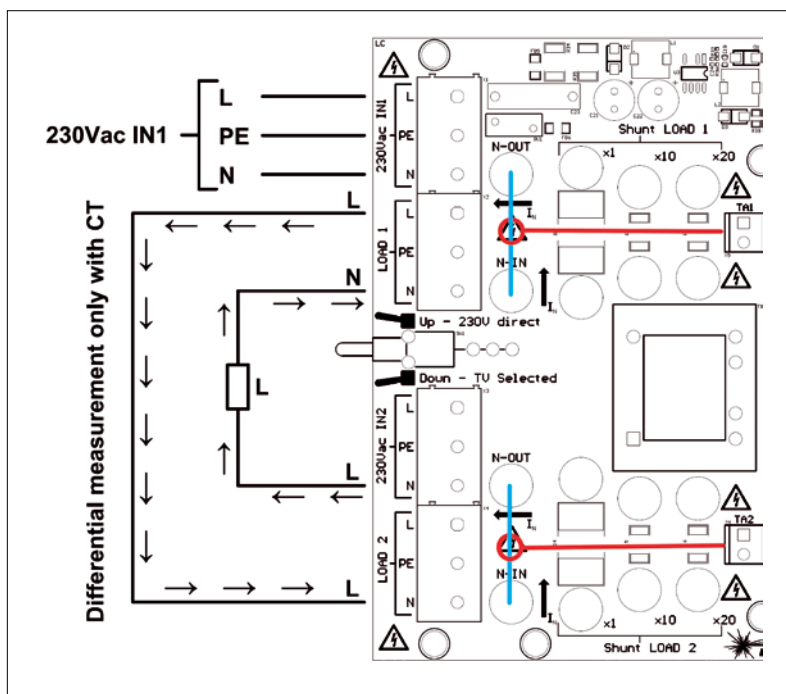


Fig. 10
Alimentazione
singola per misura
differenziale.

collegamento per un singolo carico con due misure di corrente, una sulla fase e una sul neutro. In altre parole si prende la fase della morsetteria X2, che è la stessa fase della morsetteria X1, e la si porta al neutro della morsetteria X4. Il neutro della morsetteria X3 andrà collegato alla fase del carico e il neutro del carico collegato al neutro della morsetteria X2. Come si vede dalla **Fig. 10** il circuito viene chiuso collegando tra loro i rispettivi N-IN e N-OUT con annesso TA per la misura della corrente. Con questa configurazione si possono usare **ESCLUSIVAMENTE** i TA per le misure di corrente. Quindi alla morsetteria X6 verrà collegato il TA per la misura della corrente di fase mentre alla morsetteria X5 il TA per la misura della corren-

te di neutro. Il software Energy Meter R.1.1.0.0 attualmente non gestisce la misura differenziale, semplicemente vedrete visualizzate le due misure di corrente come se fossero due carichi indipendenti.

Tuttavia sfruttando le connessioni opto-isolate messe a disposizione dal connettore CN6 potrete voi stessi sviluppare dei sistemi di misura differenziale a seconda dell'applicazione che intendete sviluppare.

COLLEGAMENTI INTERNI ALLA DEMO-BOARD

Per il collegamento dei vari shunt resistivi o dei TA si è fatto uso dei terminali a vite (Screw Terminals) della WURTH e più in particolare il codice 7466003R facente parte della serie RED-CUBE WP-SMBU.

Questa tipologia di terminali portano fino a 50A di corrente e servono per collegare a vite (con passo M3) dei cavi elettrici precedentemente crimpati, su entrambe le estremità, con capicorda di opportuno diametro.

Questo rende molto pratico e veloce il collegamento dei vari shunt resistivi o del primario del TA che si intende usare. Le **Fig. 5 e 6** precedentemente introdotte mettono in evidenza i collegamenti di cui si parla. Quindi quando si usano i TA è sufficiente un solo cavo crimpato per chiudere il circuito mentre nel caso si utilizzi uno dei tre possibili shunt servono due cavi.

CONCLUSIONI

Concludiamo quindi la trattazione con le seguenti considerazioni: prima di tutto qualora decideste di utilizzare questa Demo Board dovrete prestare massima attenzione nel maneggiarla e nella realizzazione dei collegamenti necessari al suo funzionamento come descritto nell'articolo. Inoltre ricordate sempre di togliere l'alimentazione 230V prima di cambiare configurazione alla scheda e prestate massima attenzione ai collegamenti alla morsetteria al fine di evitare di mischiare fase con neutro.

Rispettate sempre quello che vedete scritto in serigrafia e come descritto nell'articolo. Nel caso in cui decideste di utilizzare la scheda di misura FT1346M, con apposito PCB adattatore, ricordatevi che dovrete obbligatoriamente utilizzare la configurazione con TA e TV e quindi dovrete configurare in modo opportuno la Demo Board. Detto ciò non ci rimane che augurarvi buon lavoro nello sviluppo delle vostre applicazioni. □



COSA OCCORRE ?

Codice	Descrizione	Prezzo € IVA inclusa
MCP39F511N	IC di monitoraggio della potenza	8,00
FT782M	Modulo convertitore USB/seriale	15,50

IL MATERIALE VA RICHIESTO A:

Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 - www.futurashop.it

SERVIZIO DI PROGETTAZIONE

L'area R&D di Futura Group studia, progetta e realizza prototipi per differenti settori utilizzando tecnologie innovative. Il team Futura Group è composto, oltre che dal personale dei nostri laboratori tecnici, anche da un numero elevato di collaboratori esterni, ciascuno specializzato nel proprio settore.



SERVIZIO DI REALIZZAZIONE PCB

Grazie al Servizio di realizzazione PCB potrai ricevere i tuoi circuiti stampati semplicemente caricando sul nostro sito il tuo Gerber e scegliendo le dimensioni e le caratteristiche preferite. È possibile ordinare schede a una singola faccia oppure multistrato, scegliendo anche il colore del solder.



Trasformiamo le tue idee in realtà!

FUTURA ELETTRONICA È IL TUO PARTNER DI RIFERIMENTO PER MOLTEPLICI SERVIZI.
I NOSTRI TECNICI SONO A TUA DISPOSIZIONE PER FORNIRE TUTTE LE INFORMAZIONI
E UNA VALUTAZIONE PERSONALIZZATA IN BASE ALLE TUE NECESSITÀ.

SERVIZIO TAGLIO LASER

Futura Elettronica offre la possibilità di realizzare mediante taglio laser i tuoi progetti in plexiglass (o metacrilato) con spessore massimo di 10 mm e con dimensioni massime lavorabili di 130x90 cm. Carica sul nostro sito il tuo file .DXF, scegli le dimensioni e le caratteristiche preferite.



SERVIZIO STAMPA 3D

Futura Elettronica mette a disposizione le proprie stampanti 3D FDM (fused deposition modeling), per chi intende realizzare un oggetto tridimensionale con dimensioni fino a 40x40x40 cm partendo da un modello 3D digitale. Carica sul nostro sito il tuo file .STL, scegli le dimensioni e le caratteristiche preferite.



**FUTURA
ELETTRONICA®**

www.futurashop.it

Futura Group srl
Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775

Tutti a caccia di tesori nascosti!



Cercametalli base con sonda impermeabile

Rileva la presenza di piccoli oggetti metallici. Dispone di uscita per cuffia, VU-meter analogico. Sonda a tenuta stagna da 170 mm. Lunghezza dell'asta regolabile.

cod. CS100
€ 69,00



Cercametalli professionale con sonda impermeabile

cod. CS102
€ 138,00

Dotato di sonda a tenuta stagna da 215 mm, strumento a lancetta, discriminatore di oggetti metallici non significativi (funzione notch), discriminatore audio a toni, controllo della sensibilità, modalità 'all metal' in grado di rilevare tutti i tipi di metalli, uscita per cuffia e indicatore di batteria scarica. Lunghezza dell'asta regolabile.



Cercametalli digitale professionale

Cercametalli con sensibilità fino a 27 cm (in base al tipo di terreno), ampio display LCD (75 x 40 mm) e sonda impermeabile da 220 mm. Dispone inoltre di funzione pinpoint (ricerca accurata degli oggetti metallici rilevati) funzione notch (discrimina gli oggetti metallici non significativi), discriminatore audio a tre toni, controllo della sensibilità a 12 livelli, lettura di profondità, uscita per cuffia e asta regolabile.

cod. CS300
€ 228,00



Prezzi IVA inclusa

DRIVER MOTORE DA 5A

Scheda driver per motori a corrente continua (DC) da 5A.

di **EDWIN VAN DEN OETELAAR**

I maker e gli studenti di oggi spesso sono di fretta oppure non hanno le competenze per progettare e saldare circuiti elettronici. Preferiscono "shield" già pronte all'uso. Perciò ho sviluppato una scheda di breakout per il driver di motore Monolithic Power Systems MP6619, che può essere utilizzata efficacemente per i propri progetti basati su motori.

Q

uesto articolo è un viaggio alla scoperta della breakout board (BoB) dell'integrato MP6619. Questo chip è un nuovo H-bridge per il controllo degli economici motori DC a spazzole e solenoidi. Questi motori sono i più accessibili e convenienti per gli hobbisti che si

dedicano alla costruzione di piccoli robot, al modellismo e a progetti creativi. Questa è la storia del percorso educativo che ho dovuto affrontare per sviluppare e testare questa scheda.

L'AVVENTURA INIZIA

Un giorno ho ricevuto un'email da un distributore di componenti riguardo a un nuovo componente H-bridge: l'MP6619. La **Fig. 1** mostra il suo diagramma a blocchi interno. Questo piccolo "miracolo" di soli 3x3 mm con specifiche impressionanti come 24 V e 5 A, oltre ad altri aspetti, ha catturato la mia attenzione (vedi il riquadro **Caratteristiche del Chip**). Come può essere possibile? Così tanta corrente, nessun dissipatore di calore, e tutte le funzionalità di protezione in un pacchetto così piccolo? Ho pensato che fosse solo pubblicità. Volevo sapere se fosse vero e se sarebbe stato utile per i miei

progetti. Così ne ho ordinati alcuni. Dopo soli due giorni, erano sulla mia scrivania: wow, erano minuscoli! Solo 3x3 mm con 0,2 mm tra i contatti e senza piedini.

In cosa mi sono cacciato?

Mi sono immerso a capofitto in questa sfida. Nonostante il prezzo basso di circa €2,50, le piccole dimensioni e il footprint insolito dell'MP6619 si sono rivelate non direttamente applicabili ai miei progetti. Era necessaria una BoB che rendesse l'MP6619 più gestibile e adatto ai test e all'integrazione. Tuttavia, non c'era una BoB disponibile per questo minuscolo footprint QFN-19. Così, ho dovuto realizzare la scheda da solo. È iniziata così la mia avventura con l'H-bridge MP6619!

PROGETTATO CON KICAD

Per sviluppare una scheda breakout MP6619, ho iniziato studiando il datasheet e disegnando lo schema della BoB in KiCad. Prima ho disegnato il simbolo e il footprint (questi non erano disponibili nella libreria o su SnapEDA), e poi il layout. Ho prestato particolare attenzione alla disposizione della resistenza shunt R(i-sense) e alla connessione dei pad di alimentazione. Questa resistenza shunt viene utilizzata per misurare la corrente dei driver MOSFET. Questo è cruciale per

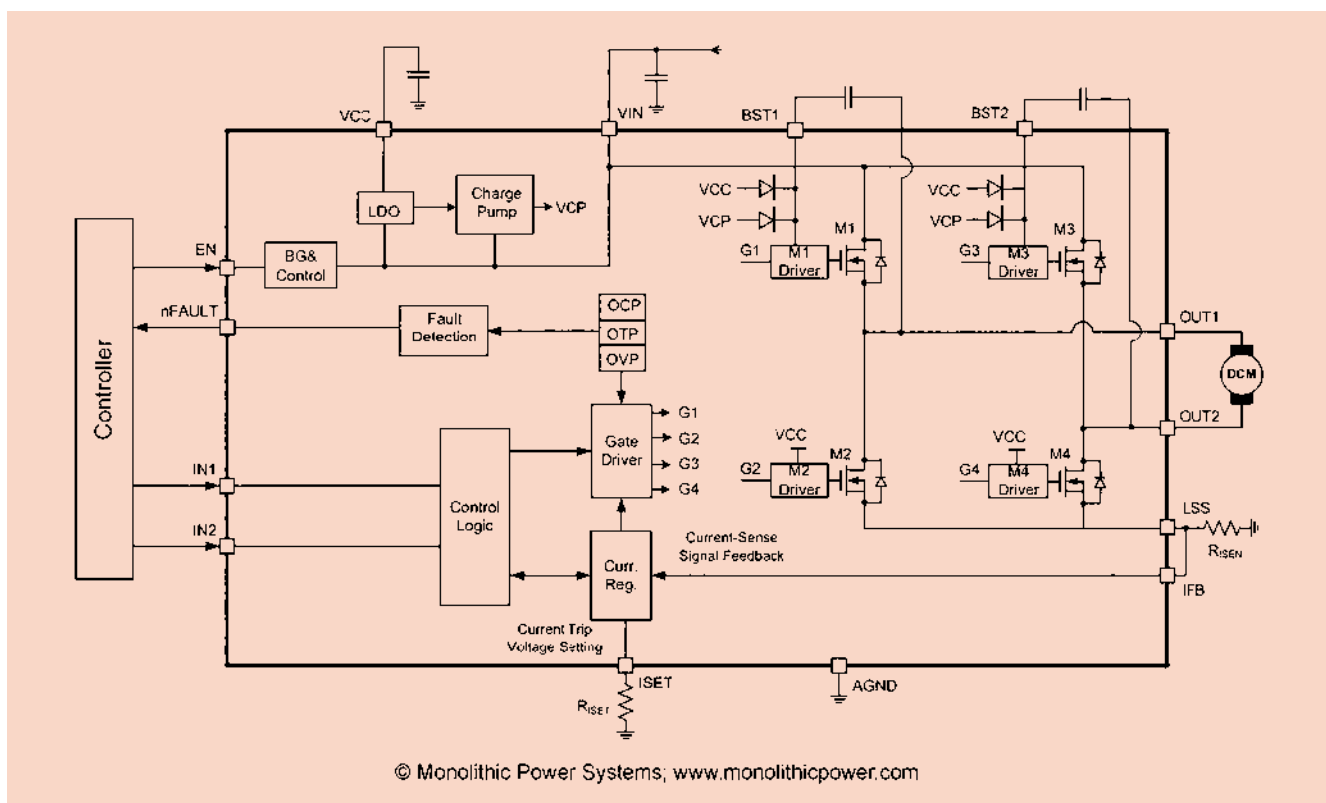


Fig. 1 Diagramma a blocchi.

Caratteristiche del Chip

- Non è solo una questione di marketing: il chip può effettivamente gestire 24 V e 5 A. È ben protetto contro sovraccarichi, cortocircuiti e si riprende affidabilmente da situazioni di guasto.
- L'aggiunta di una bobina o di un filtro migliora significativamente le prestazioni.
La necessità dipende dall'applicazione e dal motore.
- I MOSFET a canale N integrati hanno una bassa $R_{ds(on)}$, di 65 mΩ, essenziale per limitare la dissipazione. I circuiti di boost necessari per pilotare i loro gate sono anch'essi integrati. I loro condensatori sono collegati esternamente.
- Il chip MP6619 è prodotto senza fili di bonding per ottenere una buona prestazione termica in un pacchetto QFN-19.
- Le isole del footprint sono particolarmente lunghe per dissipare calore dal chip al PCB.
- Il PCB deve essere progettato con cura, tenendo conto non solo dei requisiti elettrici, ma anche di quelli termici.

il corretto funzionamento della scheda breakout. Anche la caduta di tensione lungo le tracce deve essere considerata durante il processo di layout del PCB. La protezione contro le sovracorrenti (OCP) è regolabile con una resistenza esterna. Senza di essa, il chip si spegne quando la tensione attraverso la resistenza shunt raggiunge 200 mV. Il datasheet suggerisce una resistenza shunt di 0,04 Ω, che limita la corrente massima a 5 A ($V = I \times R$). Aggiungendo una resistenza $R(i-set)$ di 80 kΩ, il limite è fissato a 100 mV, consentendo di ridurre il limite di corrente quando si utilizza la stessa resistenza shunt. Sono possibili anche altri valori di resistenza. Ho scelto una resistenza shunt di 0,05 Ω per il mio primo test perché disponibile nel mio cassetto. Questo porta a un limite di corrente in uscita di 4 A.

Ho realizzato rapidamente la prima versione del layout PCB della BoB senza uno studio approfondito, poiché volevo iniziare i test subito, ma mi sono rallegrato troppo presto.

La prima BoB includeva due chip MP6619, LED, resistori, condensatori e connettori. In quel momento, pensavo ancora che i terminal block del PCB potessero essere utili. Sono stati necessari

quattro strati di rame, ho posizionato molti vias, un piano di massa e ho utilizzato le tracce più larghe possibili a causa delle alte correnti. È stato un lavoro di un giorno. Una settimana dopo aver inviato i Gerber a un produttore di PCB, avevo le schede sul mio banco di lavoro. Saldare il QFN è stata la sfida più grande, poiché i chip erano più piccoli di quanto fossi abituato. Per posizionare correttamente il QFN con tutte le connessioni, sono stati necessari un microscopio stereoscopico, un buon saldatore a temperatura controllata, un saldatore ad aria calda, molto fluxante e tanta pazienza. Una volta posizionato il QFN e asciugato il sudore dalla fronte, gli altri componenti SMD "normali" erano un gioco da ragazzi; sembravano giganti rispetto al QFN. L'isola di saldatura per il terminal block del PCB è più grande dell'intero chip QFN (**Fig. 2**)!

Ho pubblicato questa prima versione sul sito web Elektor Labs (<https://tinyurl.com/299prw8n>).

PRIMI TEST

I primi test sono stati effettuati con un piccolo motore DC che avevo sul mio banco di lavoro. Era un modello semplice da 12 V con una corrente di stallo inferiore a 5 A (**Fig. 3**).

L'MP6619 è costituito internamente da due mezzi ponti, ognuno dei quali ha un ingresso (IN1, IN2). Il chip stesso assicura il corretto timing dei MOSFET lato alto e lato basso in modo che non si verifichino cortocircuiti tra le linee di alimentazione. Il chip è protetto da tutti i problemi tipici. Integra UVLO (sottotensione), OVP (protezione da sovratensione), OCP (protezione da sovracorrente) e OTP (protezione da temperatura eccessiva) che dovrebbero gestire in modo sicuro quasi tutte le situazioni di errore, come vedremo!

CONNESSIONE DI IN1, IN2 ED EN

Gli ingressi sono dotati di resistori pull-down. Non appena l'ingresso EN riceve un livello alto, il chip viene attivato e il regolatore di tensione interno inizia a funzionare. In quel momento, l'uscita VCC del MP6619 fornisce una tensione di 5 V.

Ho utilizzato un piccolo condensatore di disaccoppiamento da 100 nF per stabilizzare questa tensione. Se i 5 V sono presenti, anche i MOSFET a canale N diventano attivi. A seconda dei livelli degli ingressi IN1 e IN2, vengono attivati i MOSFET sul lato alto o sul lato basso. I livelli logici vanno da 3,3 V a 5 V, quindi non c'è problema a collegare la scheda a un ESP32 o a un Arduino.

Per i miei test, ho utilizzato interruttori per applicare 3,3 V agli ingressi.

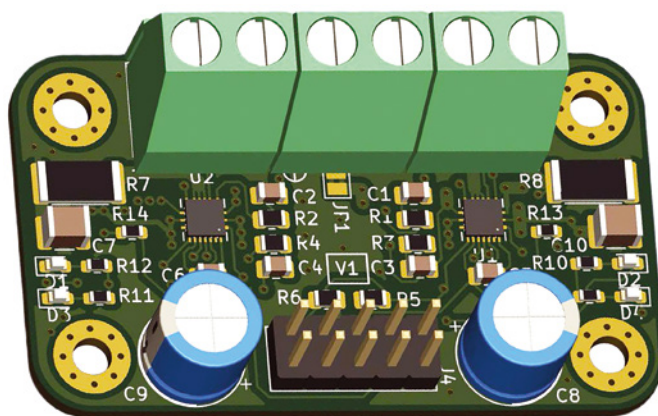


Fig. 2 Modello in 3D del primo PCB.

SUCCESSO PARZIALE

Il mio motore non voleva girare, l'uscita nFault è andata immediatamente a livello basso (l'indicatore LED di errore si è acceso), ma non avevo un cortocircuito all'uscita. Riducendo significativamente la tensione di ingresso alla fine ho permesso di far ruotare lentamente il motore in senso antiorario e orario. È importante notare che in questa fase non ho previsto nessun controllo di velocità PWM. Ho fatto prove riducendo la resistenza shunt e aumentando la resistenza Rset per far girare il motore, il che alla fine ha portato a un "errore umano" — 12 V su IN1, una BoB bruciata come risultato e un profondo sospiro di frustrazione.

Il PCB è andato in fumo e ho dovuto assemblare una nuova BoB con nuovi componenti.

UN'ALTRA BOB E ULTERIORI TEST

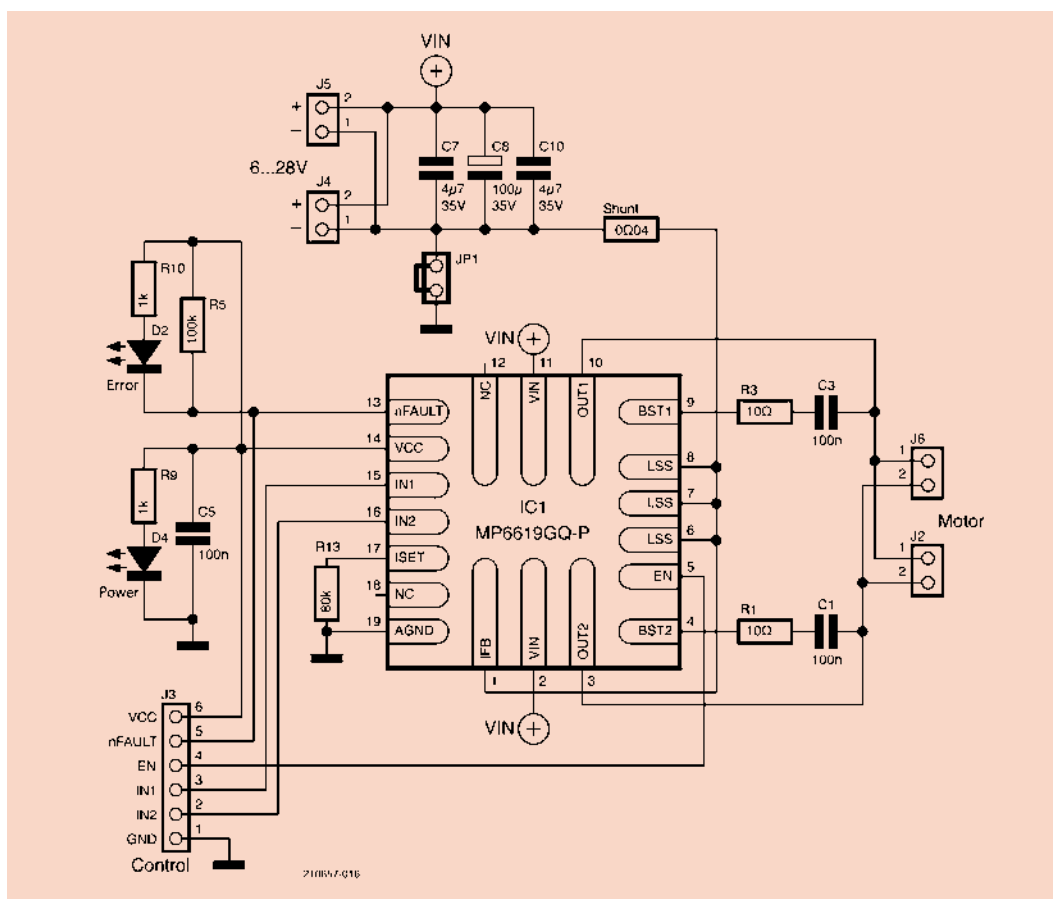
Alla fine, ho scoperto che il motore era dotato di condensatori di soppressione del rumore collegati in parallelo al commutatore.

Di conseguenza, il motore si comporta meno come



Fig. 3 I primi test sono stati effettuati con un motore a corrente continua con riduttore.

→ Fig. 4
Lo schema della BoB rivista.



induttore e più come un condensatore. Ciò fa sì che l'OCP intervenga immediatamente quando c'è un picco di corrente verso il motore. Questo "sovraccarico" causa un timeout di 1 ms prima che l'MP6619 ci riprovi. Questo processo si ripete ancora e ancora, facendo suonare il motore in maniera stridente.

La rimozione di questi condensatori aiuta a controllare il motore a tensioni leggermente più elevate. Nel frattempo, dopo aver esaminato più attentamente il datasheet, ho anche sviluppato un nuovo design della mia scheda.

NUOVO DESIGN E LAYOUT

Per la mia seconda versione, il design e il layout dovevano essere modificati.

Ho apportato le seguenti modifiche:

- Solo un chip MP6619.
- La posizione dello shunt è stata rivista.
- Un header pin da 0,1" invece dei morsetti; nessun foro di montaggio.
- Una serigrafia chiara; solo il numero minimo di componenti.

Ciò ha portato a un nuovo schema (**Fig. 4**) e successivamente anche a un nuovo PCB (**Fig. 5 e 6**) con quattro strati e un ingombro di circa 25x25 mm, prodotto in un pannello di produzione con finitura ENIG (electroless nickel immersion gold). Dopo aver popolato la scheda, ho ottenuto un modulo davvero compatto che, credo, possa essere controllato da quasi tutti i microcontrollori.

ULTERIORI TEST ED ESPERIMENTI

Ho utilizzato lo stesso motore del primo test. L'OVP e l'ULVO si sono comportati bene. L'ho provato con il controllo PWM per realizzare il controllo della velocità. Testando con frequenze da 1 kHz a 50 kHz ho ottenuto il comportamento previsto: il motore era ben controllabile. L'esempio del datasheet utilizza 20 kHz e ha funzionato bene a basse tensioni, ma a tensioni più elevate, ancora una volta l'OCP è intervenuto e il motore si è fermato. Aggiungendo un induttore extra di 220 μH in serie con il motore (come quello mostrato in **Fig. 7**), il comportamento è stato significativamente migliorato e l'intervallo di velocità è stato aumentato.

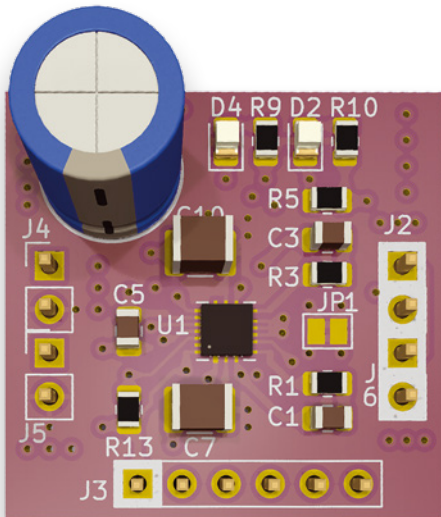


Fig. 5
Modello 3D della BoB da 25x25 mm (lato superiore) composto da quattro strati e finitura ENIG.

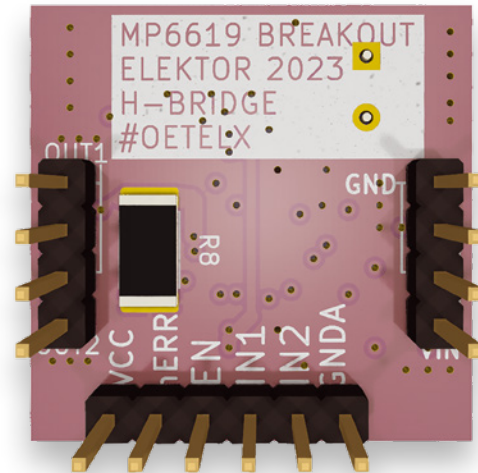


Fig. 6
Modello 3D della BoB (lato inferiore) con pin header.

Di conseguenza, il motore ha fornito più potenza e l'OCF è intervenuto meno frequentemente. Con questi induttori extra, la soluzione sembra essere adatta per l'uso nei progetti. In alcune condizioni, il motore restituisce energia, che torna all'H-bridge. Questa energia viene dissipata nelle linee di alimentazione attraverso i diodi dei MOSFET, poiché il motore funziona anche come generatore (la forza elettromotrice di un motore DC). Nei miei test, ho usato il condensatore elettrolitico C8 a bassa ESR in parallelo con la sorgente di alimentazione. Con quest'ultimo miglioramento la scheda era stabile.

OPTOISOLATORI

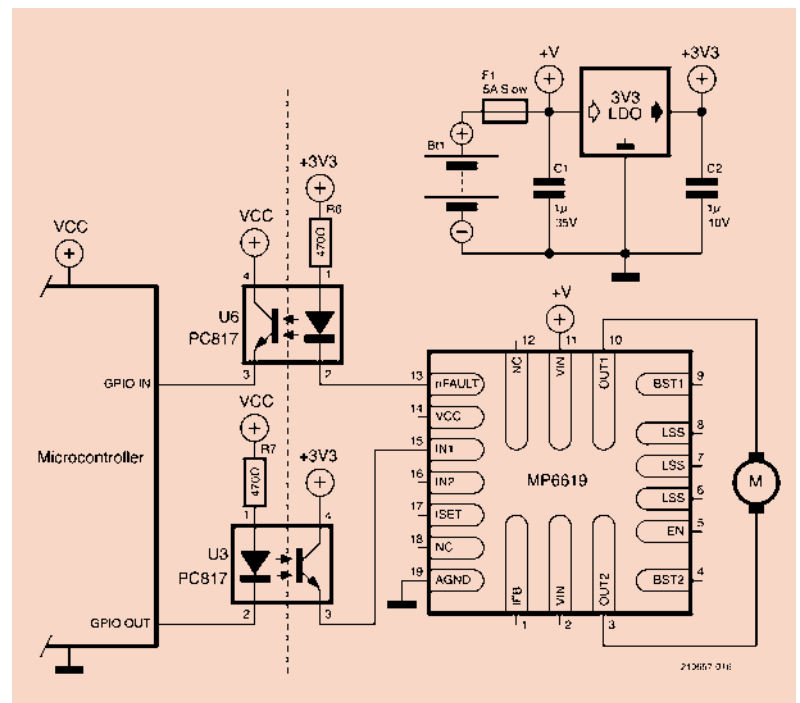
La **Fig. 8** mostra il mio circuito di test. Ho deciso di utilizzare optoisolatori per fornire isolamento

galvanico tra il lato di controllo e i driver del motore, poiché l'alimentazione di un motore di solito trasporta livelli di rumore considerevoli. Anche l'EMI può causare problemi. L'isolamento riduce il rischio di interferenze nei segnali a bassa tensione e impedisce che impulsi generati dal motore e dal suo driver raggiungano il microprocessore e altre componenti elettroniche. I LED di input degli optoisolatori possono essere pilotati direttamente da un pin di microprocessore utilizzando una resistenza da 470 Ω in serie. Sull'uscita, l'emettitore del fototransistor è collegato all'IN1 della BoB, e

Fig. 8
Il circuito con optoisolatori, tra il microcontrollore e la BoB, utilizzato per i test.



Fig. 7
Un induttore con nucleo toroidale in serie con il motore migliora il comportamento della BoB.



Resistenze

(SMD 0805)
 R1, R3 = 10 Ω
 R5 = 100 k Ω
 R8 = 0 Ω 04, 2 W, SMD 2512
 R9, R10 = 1 k Ω
 R13 = 80 k Ω

Condensatori

C1, C3, C5 = 100 nF, 50 V, SMD 0805
 C7, C10 = 4.7 μ F, 35 V, SMD 1210
 C8 = 100 μ F, 35 V, electrolytic, $\varnothing$ 8 mm

Semiconduttori

D2 = LED, red, SMD 0806
 D4 = LED, green, SMD 0805
 IC1 = MP6619GQ-P, QFN-19

Varie

J2, J4, J5, J6 = 2-pin pin header, 1/10" (2.54 mm)
 J3 = 6-pin pin header, 1/10" (2.54 mm)
 BoB PCB (vedi il testo)

quindi all'ingresso dell'MP6619, e il suo collettore è accoppiato a una tensione logica fornita da un regolatore di tensione LDO da 3,3 V alimentato dalla fonte di alimentazione del motore — nel mio caso una batteria. Questo è sufficiente per garantire un funzionamento affidabile e stabile.

La **Fig. 8** mostra anche come è possibile portare il segnale nFault dell'H-bridge (uscita Open-Drain) all'MCU tramite un optoisolatore nello stesso modo dell'altro percorso. In tal caso, il GPIO del microcontroller connesso avrà bisogno di una resistenza pull-down.

CASE QFN

I case FCQFN (**Fig. 9**) possono avere pad di forma irregolare, spesso disposti in lunghe strisce strette. A differenza dei pacchetti QFN regolari, il calore viene dissipato attraverso molti di questi pad anziché attraverso un grande pad centrale. Questo pone alcune sfide per il design del PCB poiché ci sono molti pad, tutti con segnali diversi, che devono essere collegati ad aree di rame.

Piccole vie possono essere posizionate all'interno delle aree dei pad (via in PAD). Sui PCB multistrato con piani di alimentazione e di massa, le vie possono collegare direttamente questi pad ai layers. In altri casi, il rame deve essere direttamente attaccato ai pad per dissipare il calore dall'IC a aree di rame più grandi. Una linea guida di layout è disponibile all'indirizzo <https://tinyurl.com/2p8ry9hd>. L'ostacolo maggiore che ho incontrato durante

questo progetto è stata la saldatura del chip MP6619. Per superare questo problema, ho sperimentato varie tecniche e alla fine ho avuto successo con molto flussante, stagno al piombo e aria calda, ma è rimasta una sfida posizionare il chip con precisione.

Una volta posizionato, la tensione superficiale dello stagno fuso assicura che il chip sia correttamente allineato con i pad. Si è rivelata necessaria molta pratica. La mia conclusione è acquistare una piastra calda per semplificare il processo di saldatura.

CONCLUSIONE

In questo articolo, ho condiviso le mie esperienze nella progettazione e realizzazione di una scheda breakout per il driver motore H-bridge MP6619. Ho discusso l'importanza dell'utilizzo di optoisolatori per fornire isolamento elettrico e ho condiviso le mie intuizioni sul lavoro con case FCQFN e le sfide coinvolte nella loro saldatura.

La breakout MP6619 offre una soluzione molto compatta per la guida di motori DC a basso costo con spazzole, rendendola una scelta eccellente per gli hobbisti e i maker che costruiscono piccoli robot e altri progetti creativi. Tutti i file di progettazione e di produzione sono open source e possono essere trovati sulla mia pagina GitHub <https://github.com/edwin-oetelaar>. L'MP6619 è disponibile in piccole quantità presso distributori come Farnell, Mouser e direttamente da Monolithic Power Systems.

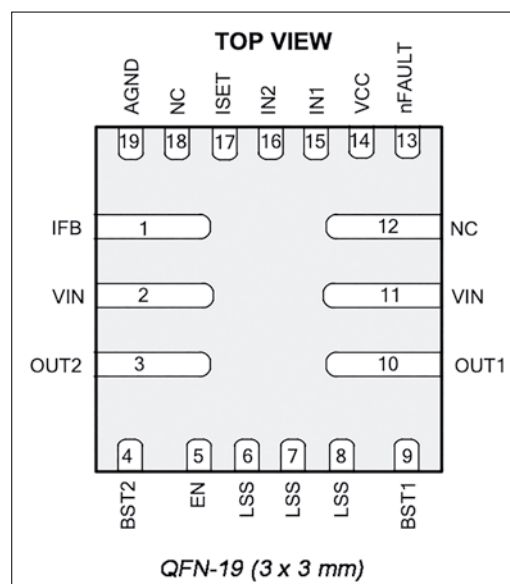


Fig. 9
 Pinout del pacchetto QFN-19 MP6619.

Per maggiori informazioni visita la pagina elektormagazine.com/magazine/elektor-312/62144

Coppia di ricetrasmittitori 8 canali PMR446

Comunica a costo zero con i nuovi ricetrasmittitori PMR 446.

Ideali per: tempo libero, escursione, lavoro, arrampicata, caccia, softair,
o come interfono all'interno dell'abitazione.

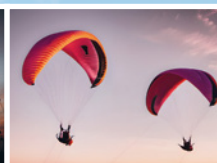
PORTATA
FINO A **5Km**
IN SPAZI APERTI



Coppia di ricetrasmittitori a 8 canali operanti sulla banda UHF. Appartengono alla categoria PMR446 e non necessitano di licenza d'uso. Dispongono di display retroilluminato, 10 tipi di suonerie di chiamata, VOX per comunicare a mani libere, funzione SCAN, tasto CALL di chiamata e mini torcia a led.



Escursionismo



Sport



Lavoro



Interfono

€ 35,⁵⁰

cod. WTA446

Prezzo IVA inclusa.



Ad ogni batteria la sua ricarica!



PER BATTERIE NI-CD / NI-MH / LI-ION

CARICABATTERIE UNIVERSALE CON DISPLAY LCD PER BATTERIE NI-MH, NI-CD, LI-ION, LIFEPO4



Caricabatterie con display LCD in grado di caricare batterie Ni-MH, Ni-Cd, Li-Ion e LiFePO4 dei formati più diffusi in commercio. Sul display vengono visualizzate: tensione, corrente, stato di carica e tipo di batteria. Il caricabatterie è alimentato tramite una porta d'ingresso USB Type-C.

cod. UC2
€ **28,00**

PER BATTERIE NI-CD / NI-MH

CARICABATTERIE CON DISPLAY LCD PER BATTERIE NI-MH E DISPLAY

cod. TP203
€ **24,00**



Caricabatterie intelligente controllato da microcontrollore, dotato di display LCD retroilluminato e in grado di ricaricare 4 batterie Ni-MH tipo AA o AAA. Rileva le batterie danneggiate, non ricaricabili o inserite male. Dispone di timer di sicurezza che interrompe la ricarica dopo 6,5 ore. Sul display LCD viene visualizzato lo stato di carica di ogni singola batteria.

CARICABATTERIE A 4 POSTI PER BATTERIE AA/AAA/9V

Permette di caricare da 1 a 4 batterie (Ni-Cd e Ni-MH) formato AA, AAA e 9V. Dotato di LED operativi che forniscono indicazioni sullo stato di ricarica delle batterie.



cod. WT40902
€ **19,00**

PER BATTERIE AL PIOMBO

CARICABATTERIE INTELLIGENTE 6-12V CON CORRENTE DI CARICA MAX 4A

Per batterie tipo WET, AGM, GEL, VRLA, MF, Ca / Ca da 6V e 12V di tensione. Controllato da un microcontrollore esegue in totale autonomia: la diagnosi, la ricostruzione (rigenerazione), la carica, il mantenimento e verifica l'efficienza della batteria.



cod. TP274
€ **48,00**

CARICABATTERIE INTELLIGENTE 6-12V CON CORRENTE DI CARICA MAX 1A

Per batterie tipo WET, AGM, GEL, VRLA, MF, Ca / Ca da 6V e 12V di tensione. Controllato da un microcontrollore esegue in totale autonomia: la diagnosi, la ricostruzione (rigenerazione), la carica, il mantenimento e verifica l'efficienza della batteria.



cod. TP221
€ **19,50**

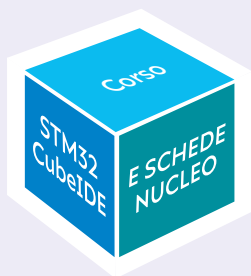
PER BATTERIE LI-ION

CARICABATTERIE A RICARICA RAPIDA CON CORRENTE DI CARICA MAX 2A

Compatibile con diverse tipologie di batterie come le 18650, 26650, 14500, 16340 e 18500. Fornisce una corrente di ricarica fino a 2A. Dispone di LED indicatori per monitorare lo stato di ricarica.



cod. LIONCH2
€ **12,00**



3

STM32CubeIDE E SCHEDA NUCLEO: LA POTENZA DELL'EMBEDDED A PORTATA DI MANO

di VALTER SARTORI

Iniziamo ad analizzare le principali periferiche integrate nei microcontrollori STM32. Esploreremo, utilizzando strumenti di debug, le funzionalità della periferica dedicata alla conversione da analogico a digitale e i Timer.

Con questa puntata andremo ad analizzare l'uso della periferica dedicata all'acquisizione analogico-digitale. Utilizzeremo tecniche di debug per osservare i dati raccolti, sia graficamente che numericamente. Cominciamo con una descrizione tecnica di questa periferica, ricordando che il numero di porte disponibili può variare in base al modello di microcontrollore utilizzato. In **Tabella 1** possiamo osservare le differenze tra le varie schede Nucleo. Ad esempio, la risoluzione, tipicamente a 12 bit, su alcuni tipi di schede è selezionabile, mentre su alcune schede è possibile avere ingressi analogici differenziali oltre alla configurazione single-ended usata comunemente.

Inoltre, sono disponibili anche alcuni canali interni: un canale per il sensore di temperatura interno (VSENSE), uno per la tensione di riferimento interna (VREFINT), uno per il monitoraggio dell'alimentazione VBAT esterna.

Per comprendere meglio come configurare le opzioni messe a disposizione dalla MCU, non resta che realizzare un nuovo progetto che ci porterà a scoprire nel dettaglio le proprietà di questa periferica.

CONFIGURAZIONE DELLE CONNESSIONI ESTERNE

Utilizzeremo, come abbiamo fatto finora, la scheda Nucleo-F401RE a cui collegheremo un trimmer resistivo del valore di 10 kΩ predisposto su una breadboard.

Scheda Nucleo	Periferiche ADC	Numero massimo di External Input	Numero massimo di Ingressi Differenziali	Risoluzioni disponibili
NUCLEO-G474RE	5	16	15	6/8/10/12 bit
NUCLEO-F446RE	3	16	-	
NUCLEO-F401RE	1	16	-	
NUCLEO-F303RE	4	14	9	
NUCLEO-F103RB	2	16	-	12 bit
NUCLEO-F072RB	1	16	-	6/8/10/12 bit
NUCLEO-L476RG	3	16	15	
NUCLEO-L152RE	1	23	-	
NUCLEO-L073RZ	1	16	-	
NUCLEO-F412ZG	1	16	-	12 bit

Tabella 1 Tipologie di periferiche ADC adottate sulle schede Nucleo

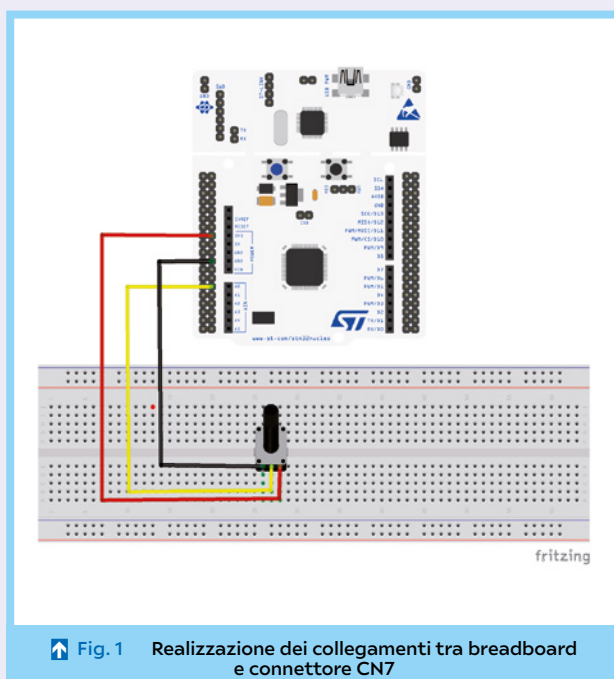


Fig. 1 Realizzazione dei collegamenti tra breadboard e connettore CN7

Le **Fig. 1 e 2** mostrano i collegamenti tra il trimmer e i connettori della scheda Nucleo e, per aiutarci ulteriormente, in **Fig. 3** possiamo osservare la disposizione dei pin sul connettore Morpho CN7.

Le frecce indicano i pin dedicati all'alimentazione 3.3V (posizione 1 pin 5), al GND (posizione 2 pin 8) e all'ingresso ADC1/0 (posizione 3 pin 28) corrispondente a PA0, che andremo successivamente a configurare.

Dopo aver realizzato i collegamenti passiamo a prendere in esame la configurazione hardware e la progettazione del software.

CONFIGURAZIONE DELLA PERIFERICA

Per gestire la conversione analogico/digitale in maniera corretta verrà attivato l'interrupt che segnala il termine della conversione. La lettura dei registri che contengono le letture relative alla conversione analogico digitale verrà sincronizzata con il completamento della conversione. Come già anticipato, il progetto verrà configurato selezionando direttamente il tipo di microcontrollore e non più la scheda Nucleo.

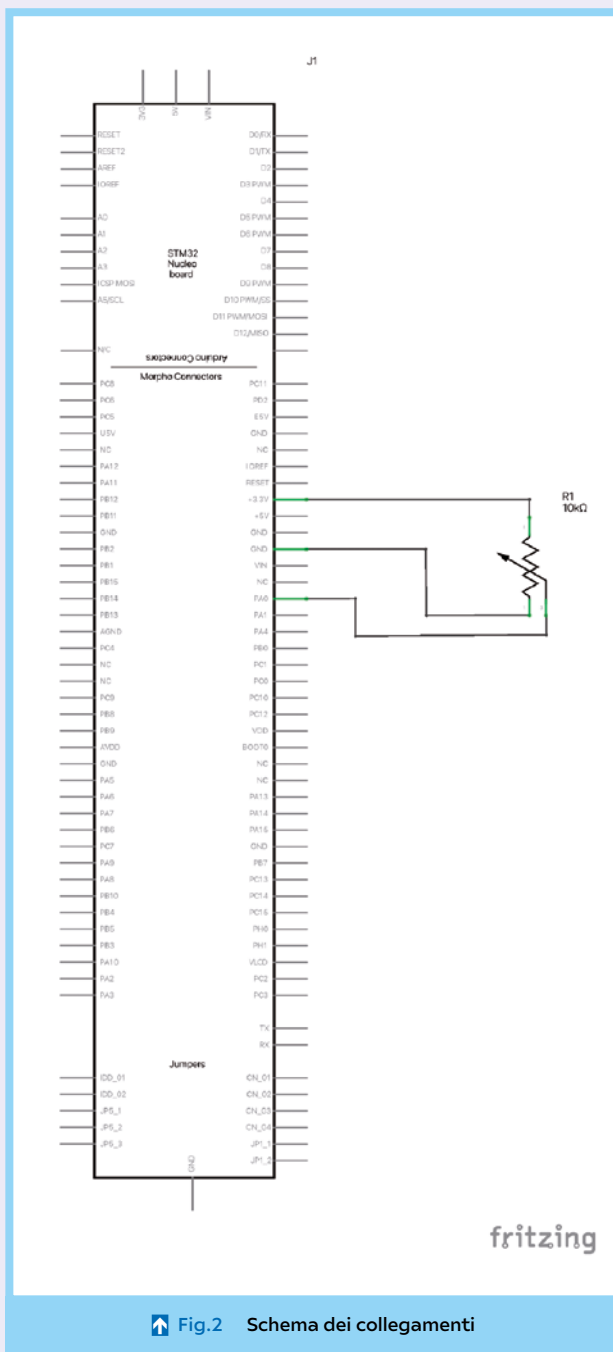


Fig. 2 Schema dei collegamenti

In questo modo potremo selezionare esclusivamente gli I/O utili ai nostri progetti ed ottenere un controllo completo di tutte le periferiche disponibili. Finora abbiamo utilizzato una preconfigurazione standard utile ad attivare l'hardware presente sulla scheda (led LD2 e pulsante B1), ma ora dovremo affrontare l'utilizzo delle periferiche disponibili nel processore, per cui dovremo realizzare collegamenti e circuiti esterni alla scheda.

Come visibile in Fig. 4, selezioniamo l'MCU STM32F401RET6 e premiamo il pulsante 'Next'.

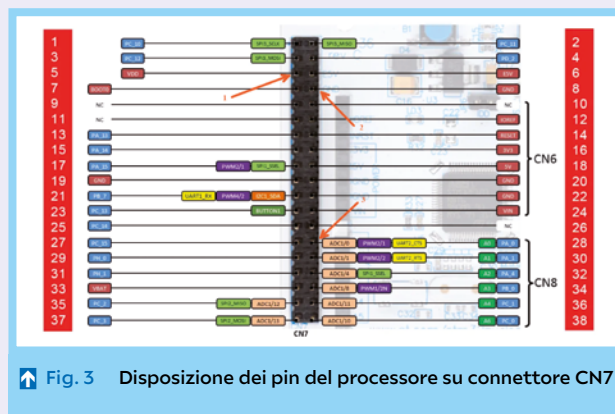


Fig. 3 Disposizione dei pin del processore su connettore CN7

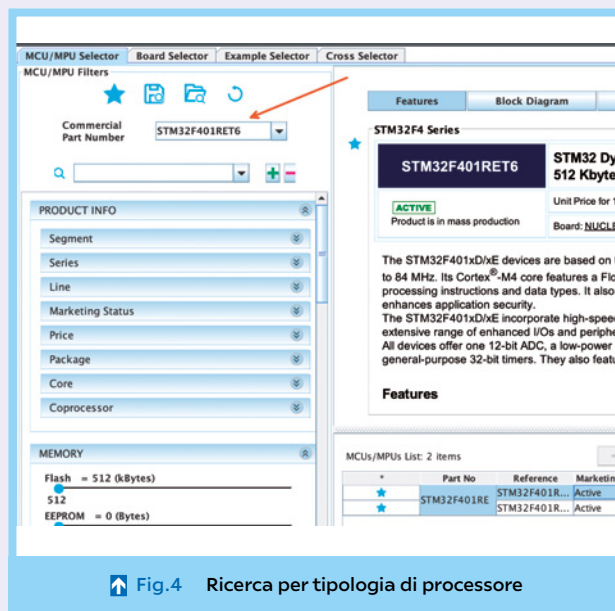


Fig. 4 Ricerca per tipologia di processore

Nel campo 'Project Name' inseriamo il titolo del progetto 'F401_ADC_IT' e selezioniamo il pulsante 'Finish'. STM32CubeIDE attiverà STM32CubeMX raffigurando la MCU come in Fig. 5. Come possiamo osservare, non sono presenti particolari configurazioni, ad eccezione di quelle base relative ad alcuni particolari funzioni del micro, le alimentazioni e il reset.

Andremo a selezionare ad una ad una le configurazioni utili al nostro progetto, utilizzando in questo modo solo gli I/O necessari.

Apriamo la 'System Core' come evidenziato nella Fig. 6, selezioniamo il campo 'RCC' (Reset Clock Control). Definiamo come generatore di clock HSE il quarzo esterno a 8 MHz X1 che, oltre a generare il clock per la STLink, opportunamente può fornirlo anche alla MCU e selezioniamo il campo 'Crystal/Ceramic Resonator'.

Passiamo a selezionare il campo 'SYS' e, come si può osservare in Fig. 7, attiviamo il Debug selezionando 'Serial

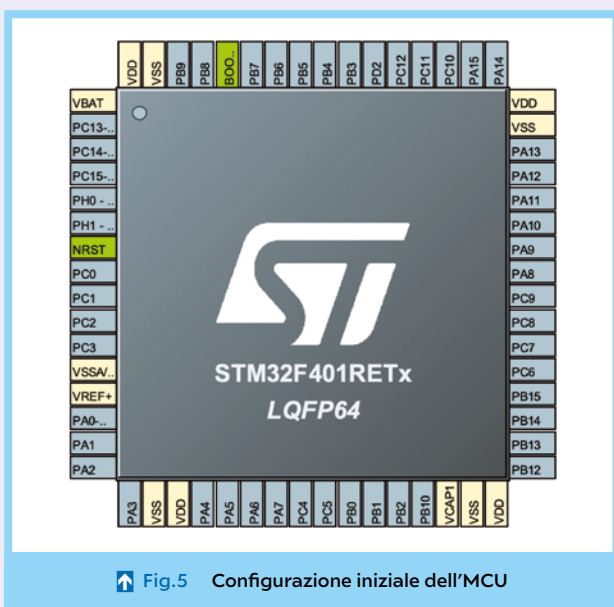


Fig.5 Configurazione iniziale dell'MCU

Wire', metodo di comunicazione della STLink e, come Time Base Source, definiamo la modalità 'SysTick' come si vede in **Fig. 8**. A questo punto passiamo a definire la 'Clock Configuration' (**Fig. 9**), impostando la sorgente di clock (input frequency) a 8 MHz, HCLK a 84 MHz, e selezionando i due radio button identificati dalle frecce. Premendo 'Enter' STM32CubeMX configurerà automaticamente lo stadio PLL in modo tale da ottenere la frequenza richiesta. Al termine non dovranno comparire blocchi colorati in rosso e le frequenze in uscita verso i bus e le periferiche dovranno avere valori di 84 e 42 MHz. Passiamo a configurare la sezione 'Analog' confermando con un click 'ADC1'. In **Fig. 10** osserviamo le configurazioni selezionate per il nostro caso. Innanzitutto, abbiamo attivato il canale IN0, spuntando il relativo checkbox. Questo ingresso è impostato in modalità single-ended, poiché la Nucleo-F401RE non dispone di ingressi differenziali. Nella finestra sottostante, è presente un elenco di parametri; per i più importanti forniremo una breve descrizione.

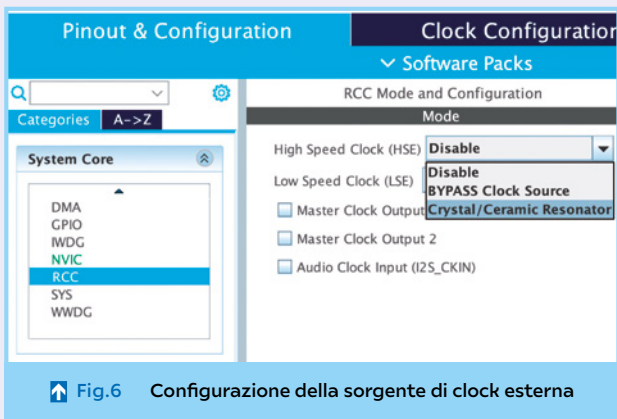


Fig.6 Configurazione della sorgente di clock esterna

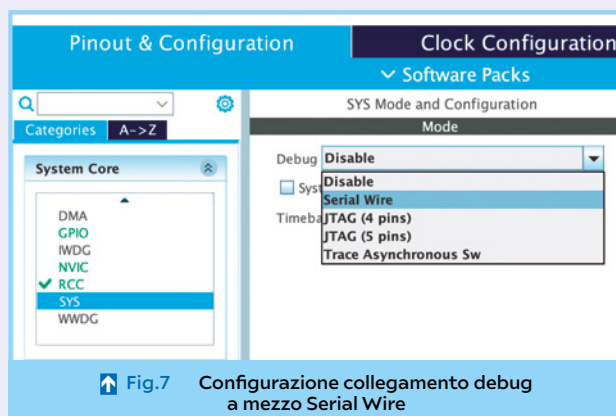


Fig.7 Configurazione collegamento debug a mezzo Serial Wire

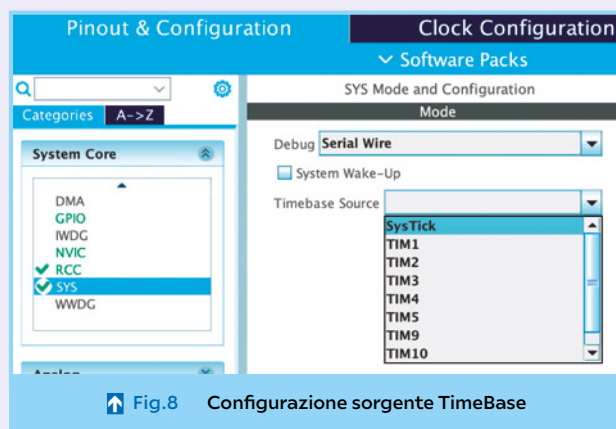


Fig.8 Configurazione sorgente TimeBase

Clock Prescaler

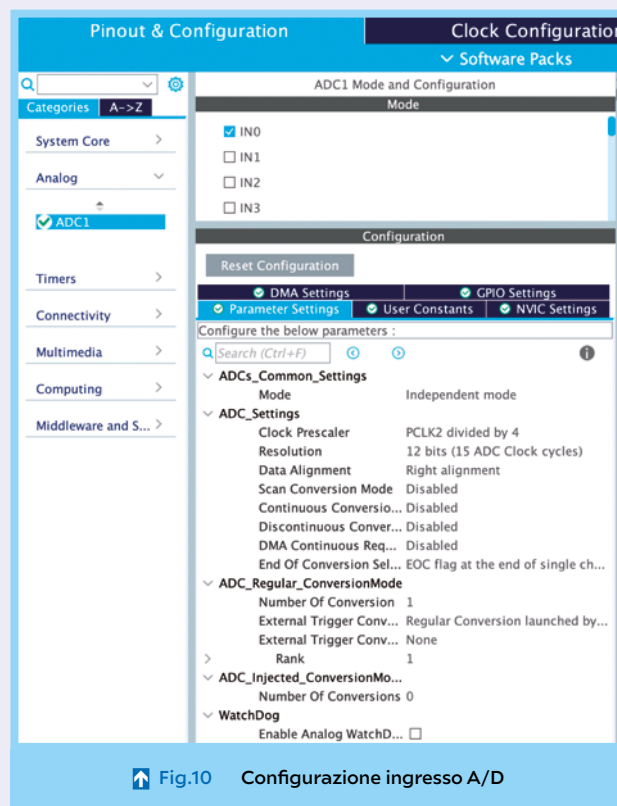
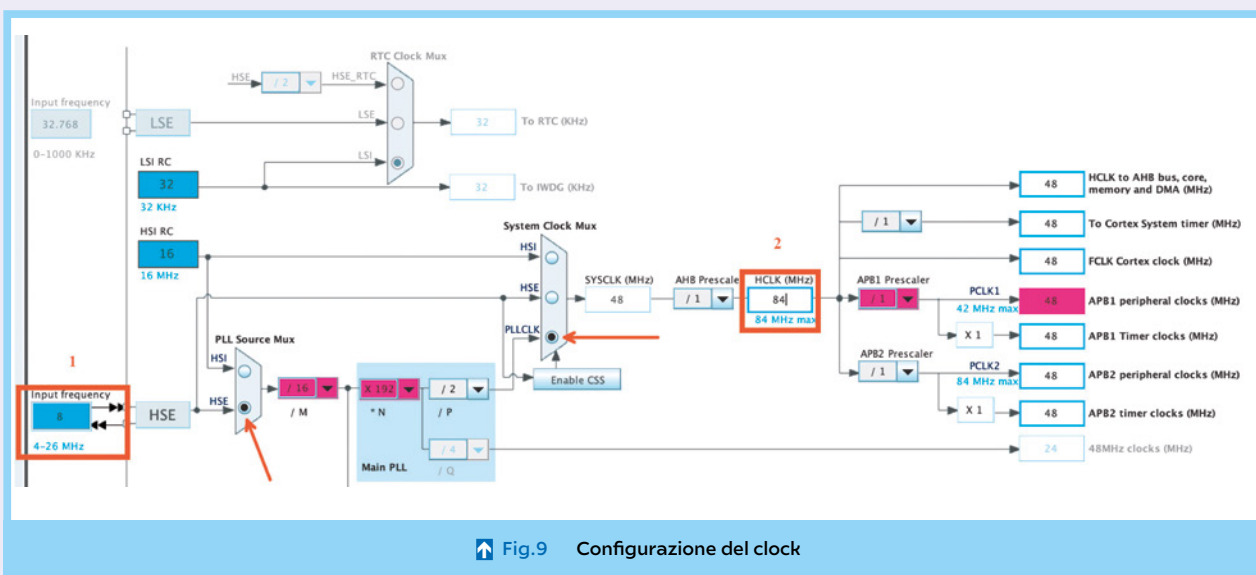
Definisce la velocità del clock (ADCCLK) utilizzato nei circuiti dell'ADC. L'ADCCLK stabilisce la velocità di questa unità di temporizzazione e ha un impatto sul numero di campioni al secondo, perché definisce la quantità di tempo utilizzata da ciascun ciclo di conversione.

Resolution

Utilizzando questo parametro è possibile definire la risoluzione della conversione A/D. La risoluzione può assumere un valore elencato nell'IDE. Maggiore è la risoluzione, minore è il numero di conversioni possibili in un secondo. Nel nostro caso adotteremo la risoluzione a 12 bit.

Data Alignment

Specifica l'allineamento dei dati convertiti. Il registro dei dati ADC è implementato come registro half-word register. Poiché solo 12 bit sono utilizzati per memorizzare la conversione, questo parametro stabilisce come questi bit sono allineati all'interno del registro. Nel nostro caso selezioneremo la modalità 'right alignment'.




```

1198  */
1199 void HAL_ADC_IRQHandler(ADC_HandleTypeDef* hadc)
1200 {
1201     uint32_t tmp1 = 0U, tmp2 = 0U;
1202
1203     uint32_t tmp_sr = hadc->Instance->SR;
1204     uint32_t tmp_cr1 = hadc->Instance->CR1;
1205
1206     /* Check the parameters */
1207     assert_param(IS_FUNCTIONAL_STATE(hadc->Init.ContinuousConvMode));
1208     assert_param(IS_ADC_REGULAR_LENGTH(hadc->Init.NbrOfConversion));
1209     assert_param(IS_ADC_EOCSelection(hadc->Init.EOCSelection));
1210
1211     tmp1 = tmp_sr & ADC_FLAG_EOC;
1212     tmp2 = tmp_cr1 & ADC_IT_EOC;
1213     /* Check End of conversion flag for regular channels */
1214     if(tmp1 && tmp2)
1215     {
1216         /* Update state machine on conversion status if not in error state */
1217         if (HAL_IS_BIT_CLR(hadc->State, HAL_ADC_STATE_ERROR_INTERNAL))
1218         {
1219             /* Set ADC state */
1220             SET_BIT(hadc->State, HAL_ADC_STATE_REG_EOC);
1221         }
1222
1223         /* Determine whether any further conversion upcoming on group regular */
1224         /* by external trigger, continuous mode or scan sequence on going. */
1225         /* Note: On STM32F4, there is no independent flag of end of sequence. */
1226         /* The test of scan sequence on going is done either with scan */
1227         /* sequence disabled or with end of conversion flag set to */
1228         /* of end of sequence. */
1229         if(ADC_IS_SOFTWARE_START_REGULAR(hadc) &&
1230            (hadc->Init.ContinuousConvMode == DISABLE) &&
1231            (HAL_IS_BIT_CLR(hadc->Instance->SQR1, ADC_SQR1_L) ||
1232             HAL_IS_BIT_CLR(hadc->Instance->CR2, ADC_CR2_EOCS) ) )
1233         {
1234             /* Disable ADC end of single conversion interrupt on group regular */
1235             /* Note: Overrun interrupt was enabled with EOC interrupt in */
1236             /* HAL_ADC_Start_IT(), but is not disabled here because can be used */
1237             /* by overrun IRQ process below. */
1238             __HAL_ADC_DISABLE_IT(hadc, ADC_IT_EOC);
1239
1240             /* Set ADC state */
1241             CLEAR_BIT(hadc->State, HAL_ADC_STATE_REG_BUSY);
1242
1243             if (HAL_IS_BIT_CLR(hadc->State, HAL_ADC_STATE_INJ_BUSY))
1244             {
1245                 SET_BIT(hadc->State, HAL_ADC_STATE_READY);
1246             }
1247         }
1248
1249         /* Conversion complete callback */
1250         #if (USE_HAL_ADC_REGISTER_CALLBACKS == 1)
1251             hadc->ConvCpltCallback(hadc);
1252         #else
1253             HAL_ADC_ConvCpltCallback(hadc);
1254         #endif /* USE_HAL_ADC_REGISTER_CALLBACKS */
1255     }

```

Fig. 14 File 'stm32f4xx_hal_adc.c' listato funzione HAL_ADC_IRQHandler()

```

main.c  stm32f4xx_hal_adc.c  F401_ADC_IT.ioc
1563  * @brief Gets the converted value from data register of regular channel.
1564  * @param hadc pointer to a ADC_HandleTypeDef structure that contains
1565  *         the configuration information for the specified ADC.
1566  * @retval Converted value
1567  */
1568  uint32_t HAL_ADC_GetValue(ADC_HandleTypeDef* hadc)
1569  {
1570      /* Return the selected ADC converted value */
1571      return hadc->Instance->DR;
1572  }
1573
1574  /**
1575   * @brief Regular conversion complete callback in non blocking mode
1576   * @param hadc pointer to a ADC_HandleTypeDef structure that contains
1577   *         the configuration information for the specified ADC.
1578   * @retval None
1579   */
1580  weak void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
1581  {
1582      /* Prevent unused argument(s) compilation warning */
1583      UNUSED(hadc);
1584      /* NOTE : This function Should not be modified, when the callback is needed,
1585       the HAL_ADC_ConvCpltCallback could be implemented in the user file
1586      */
1587  }
1588

```

Fig. 15 File 'stm32f4xx_hal_adc.c' listato funzione 'HAL_ADC_ConvCpltCallback()'

```

237  /* USER CODE BEGIN 4 */
238  void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc1)
239  {
240      end_conversion = TRUE; // Attiva la variabile essendo stata completata la conversione
241  }
242  /* USER CODE END 4 */

```

Fig. 16 File 'main.c' implementazione della funzione 'HAL_ADC_ConvCpltCallback()'

```

48  /* USER CODE BEGIN PV */
49  _Bool end_conversion = FALSE; // Al termine della conversione viene portata a livello 1
50  uint16_t adcResult;          // Risultato della conversione digitale
51  float voltage;               // Valore misurato dall'ADC convertito in tensione
52
53  /* USER CODE END PV */

```

Fig. 17 File 'main.c' definizione delle variabili

```

32  /* Private define -----*/
33  /* USER CODE BEGIN PD */
34  #define RIF_VOLTAGE 3.3      // Tensione di riferimento del convertitore 3.3 V
35  #define RESOLUTION_BIT 4095.0 // Risoluzione 12 bit
36  #define TRUE 1              // Valori booleani
37  #define FALSE 0
38  /* USER CODE END PD */

```

Fig. 18 File 'main.c' dichiarazioni delle costanti

```

67
68 /**
69  * @brief The application entry point.
70  * @retval int
71  */
72 int main(void)
73 {
74     /* USER CODE BEGIN 1 */
75     /* USER CODE END 1 */
76
77     /* MCU Configuration-----*/
78
79     /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
80     HAL_Init();
81
82     /* USER CODE BEGIN Init */
83
84     /* USER CODE END Init */
85
86     /* Configure the system clock */
87     SystemClock_Config();
88
89     /* USER CODE BEGIN SysInit */
90
91     /* USER CODE END SysInit */
92
93     /* Initialize all configured peripherals */
94     MX_GPIO_Init();
95     MX_ADC1_Init();
96     /* USER CODE BEGIN 2 */
97     HAL_ADC_Start_IT(&adc1); // attiva la routine di interrupt
98     /* USER CODE END 2 */
99
100    /* Infinite loop */
101    /* USER CODE BEGIN WHILE */
102    while (1)
103    {
104        /* USER CODE END WHILE */
105
106        /* USER CODE BEGIN 3 */
107
108        if(end_conversion == TRUE)
109        {
110            adcResult = HAL_ADC_GetValue(&adc1); // lettura registro e memorizzazione su variabile
111            voltage = ((float)adcResult) * RIF_VOLTAGE / RESOLUTION_BIT; // conversione in volt
112            end_conversion = FALSE; // reset variabile booleana
113            HAL_Delay(100); // attesa 100 ms
114            HAL_ADC_Start_IT(&adc1); // Reimposta le condizioni per un nuovo interrupt
115        }
116    }
117
118    /* USER CODE END 3 */
119 }
120
121

```

 Fig. 19 File 'main.c' listato del programma

più razionale delle risorse ben evidenziato in **Fig. 12**. Quindi vogliamo evidenziare che la scelta diretta della MCU e non della scheda Nucleo, permette di configurare le risorse necessarie e razionalizzare al meglio l'utilizzo di ogni singolo I/O.

Negli esempi che seguiranno utilizzeremo sempre questo approccio per la creazione dei nostri progetti.

Configurazione del software

Procediamo ora con la generazione del codice salvando le configurazioni impostate su STM32CubeMX e visualizziamo sull'IDE il file main.c. In **Fig. 13** osserviamo la finestra dedicata al 'Project Explorer' dove espandiamo la directory contenente gli HAL_Driver.

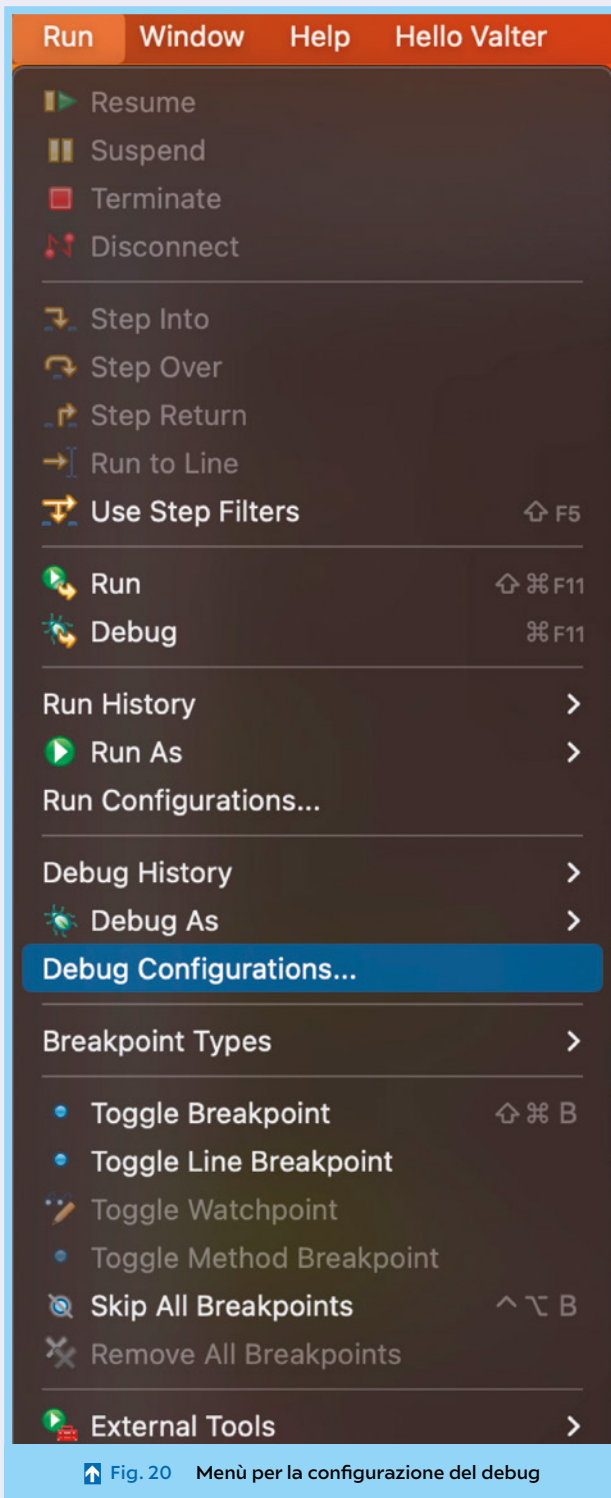
Selezioniamo il file 'stm32f4xx_hal_adc.c' dove all'interno troveremo le funzioni dedicate alla periferica ADC.

Nella finestra Outline dell'IDE, selezioniamo la funzione incaricata della gestione dell'interrupt request: HAL_ADC_IRQHandler().

In **Fig. 14**, osserviamo il listato della funzione 'HAL_ADC_IRQHandler' che richiama alla linea 1253 la funzione 'HAL_ADC_ConvCpltCallback'.

Alla riga 1580 troviamo la funzione 'HAL_ADC_ConvCpltCallback', come visibile in **Fig. 15**, da cui copieremo il testo evidenziato in figura evitando il suffisso '_weak'.

Questa funzione dovrà essere implementata nel file 'main.c' per gestire l'interrupt relativo al completamento della conversione analogico/digitale.



In **Fig. 16** osserviamo la funzione 'HAL_ADC_ConvCpltCallback' racchiusa nella sezione 'USER CODE BEGIN 4' e 'USER CODE END 4'.

All'interno effettueremo un cambio di stato alla variabile booleana 'end_conversion' che abbiamo dichiarato alla riga 49 (**Fig. 17**) ponendola allo stato FALSE.

Inoltre, sempre in **Fig. 17** osserviamo il listato relativo alle variabili utilizzate per la conversione analogico/digitale. La variabile intera a 16 bit senza segno 'adcResult' verrà utilizzata per la memorizzazione del valore della conversione analogico/digitale, mentre la variabile float 'voltage' verrà utilizzata per memorizzare il valore convertito in unità ingegneristiche della misura rilevata dal convertitore. Inoltre nell'area dedicata alle definizioni (**Fig. 18**) troviamo le dichiarazioni che verranno utilizzate per il calcolo della tensione e i valori abbinati agli stati booleani 'TRUE' e 'FALSE'.

Con 'RIF_VOLTAGE' definiamo il valore di tensione di riferimento a cui corrisponderanno 2^{12} bit pari a 3.3 V, e con la variabile 'RESOLUTION_BIT' definiamo la costante $2^{12} - 1 = 4095$.

La risoluzione è data dal rapporto tra 'RIF_VOLTAGE' e 'RESOLUTION_BIT' del valore di 806 $\mu\text{V/bit}$.

In **Fig. 19** osserviamo il listato del programma sviluppato nella funzione 'main'. Alla riga 97 dopo le inizializzazioni relative al GPIO e all'ADC1 osserviamo il richiamo alla funzione che attiva la conversione A/D in modalità interrupt: 'HAL_ADC_Start_IT()'.

Il convertitore ADC1 e l'interrupt ad esso annesso devono essere riattivati a valle di ogni conversione.

La gestione dell'interrupt si occupa di cambiare di stato la variabile booleana 'end_conversion' (**Fig. 16** riga 240).

Durante l'esecuzione del loop (**Fig. 19** riga 102) il firmware controlla lo stato della variabile 'end_conversion'. Nel caso il valore della variabile sia 'TRUE', il programma entra nella routine gestita dalla condizione 'if'.

Alla riga 110, sempre di **Fig. 19**, leggiamo il valore contenuto nel registro del convertitore ADC1 e, nella riga successiva, convertiamo questo valore nel rispettivo valore in volt, memorizzandolo nella variabile 'voltage'.

Alla riga 112 (**Fig. 19**) riportiamo la variabile booleana 'end_conversion' al valore FALSE e riattiviamo l'interrupt per la prossima conversione mediante la funzione 'HAL_ADC_Start_IT'.

La conversione analogico digitale viene riattivata e attiverà il flag di interrupt al termine del processo di lettura.

Quando il flag dell'interrupt risulterà attivo, l'IRQHandler richiamerà la funzione 'HAL_ADC_ConvCpltCallback' (**Fig. 16**) riportando la variabile booleana 'end_conversion' allo stato TRUE e permettendo così la gestione di una nuova lettura del convertitore A/D.

Al termine della stesura del software possiamo effettuare una 'build' per verificare la presenza di eventuali errori e al termine compilare e scaricare nella Nucleo il firmware onde completare il debug.

IL DEBUG GRAFICO

Il programma che abbiamo realizzato permette di apprendere un ulteriore metodo di gestione del debug in modalità grafica.



Tipologia Timer	Risoluzione (bit)	Tipologia di contatore	DMA	Canali	Canali Complementari
Avanzato	16	Up, Down, Center Aligned	Si	4	3
General Purpose	16/32	Up, Down, Center Aligned	Si	4	0
Basic	16	Up	Si	0	0
1 canale	16	Up	No	1	0
2 canali	16	Up	No	2	0
1 canale 1 uscita complementare	16	Up	Si	1	1
2 canali 1 uscita complementare	16	Up	Si	2	1
Alta risoluzione	16	Up	Si	10	10
Basso consumo	16	Up	No	1	0

Tabella 2 Tipi di Timer presenti nelle MCU STM32

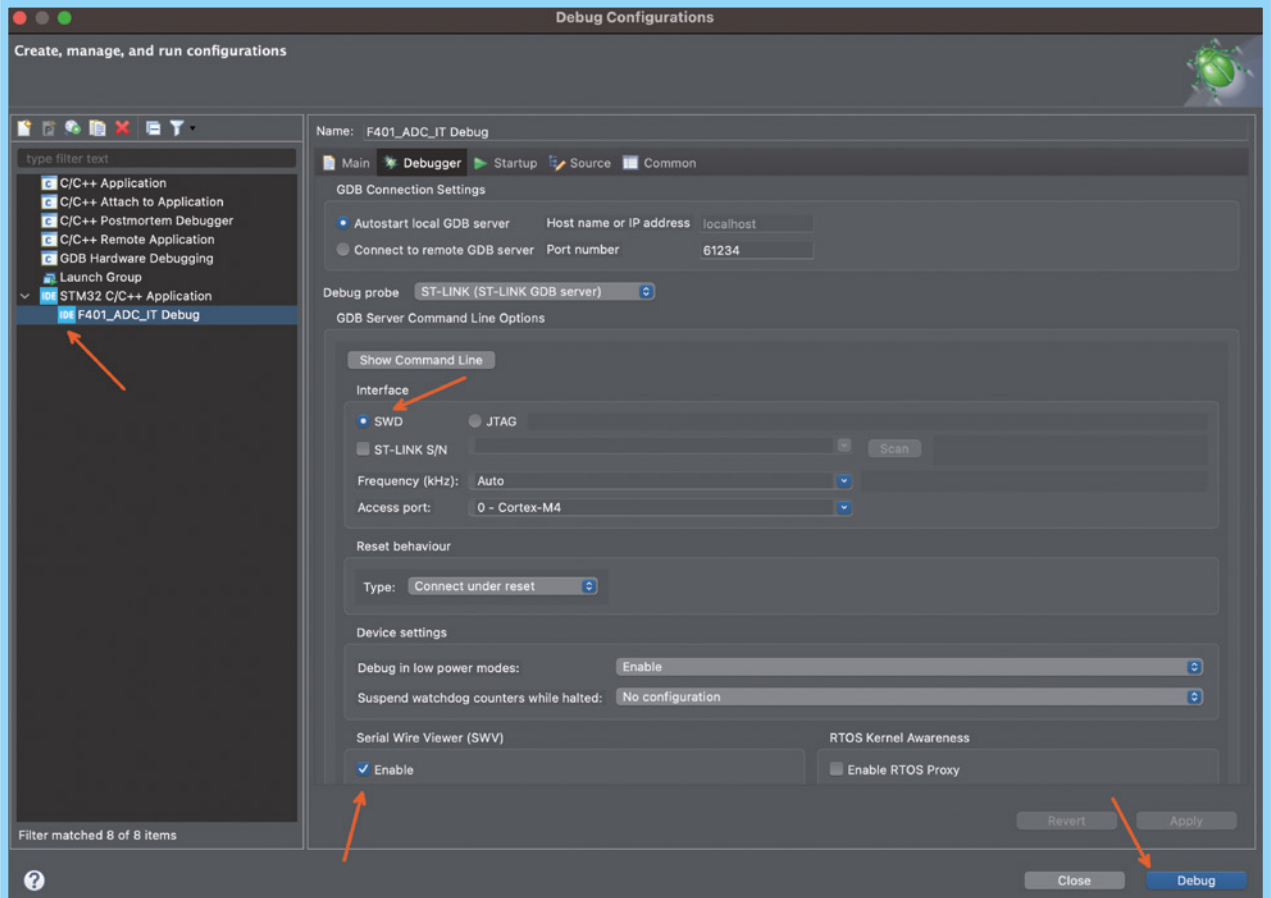


Fig. 21 Configurazione del debug per la visualizzazione su grafico

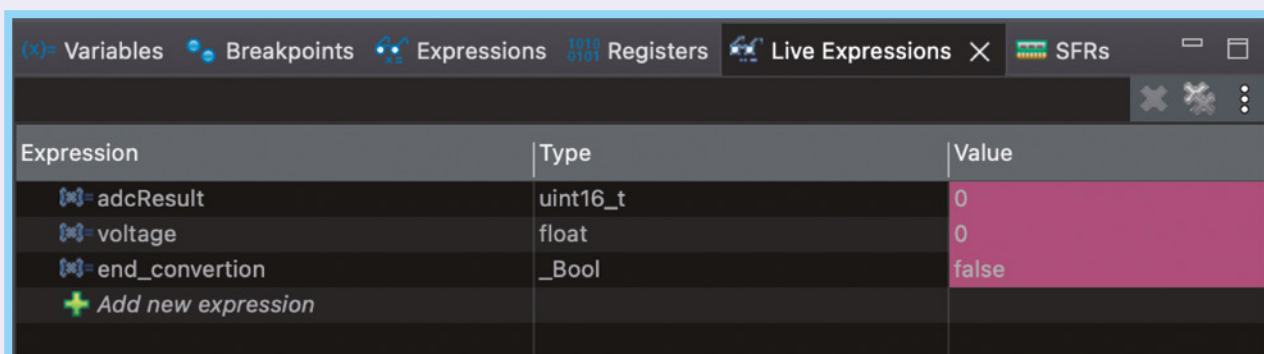


Fig. 22 Finestra 'Live Expressions' dove aggiungere le variabili di controllo

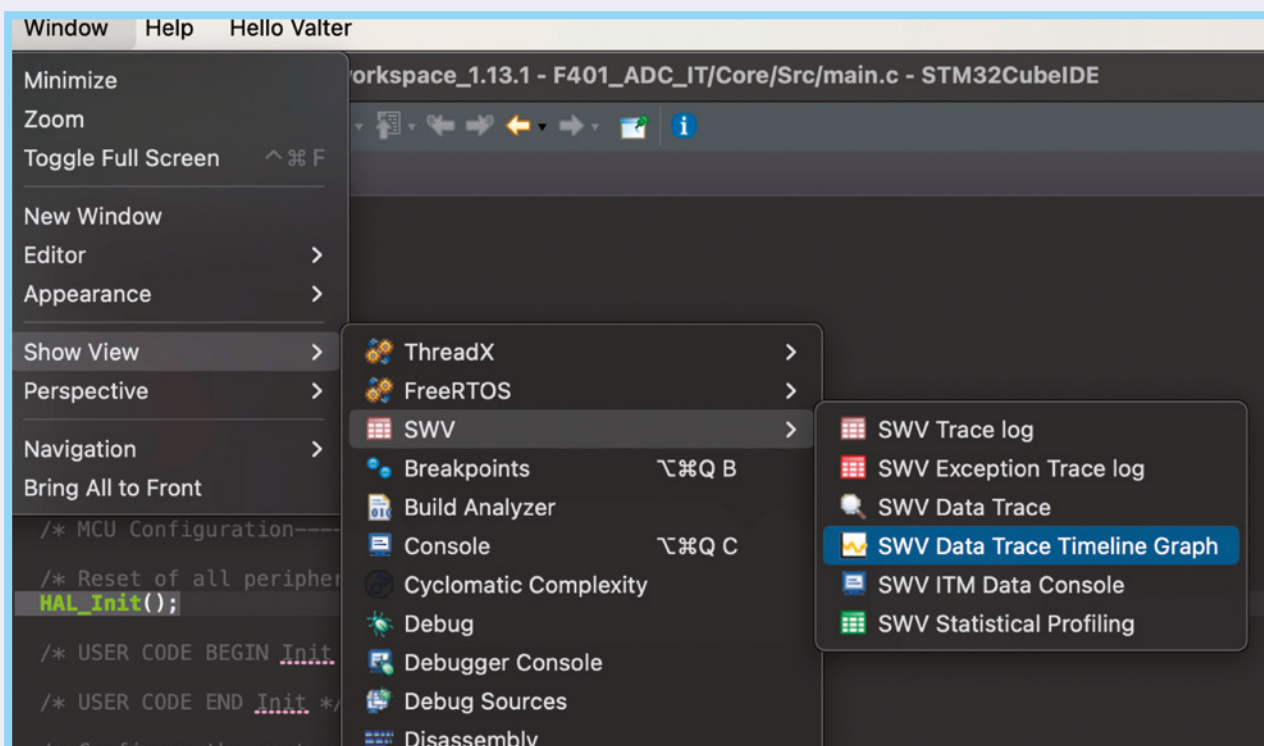


Fig. 23 Menù di attivazione della SVW Data Trade Timeline Graph



Fig. 24 Finestra del grafico

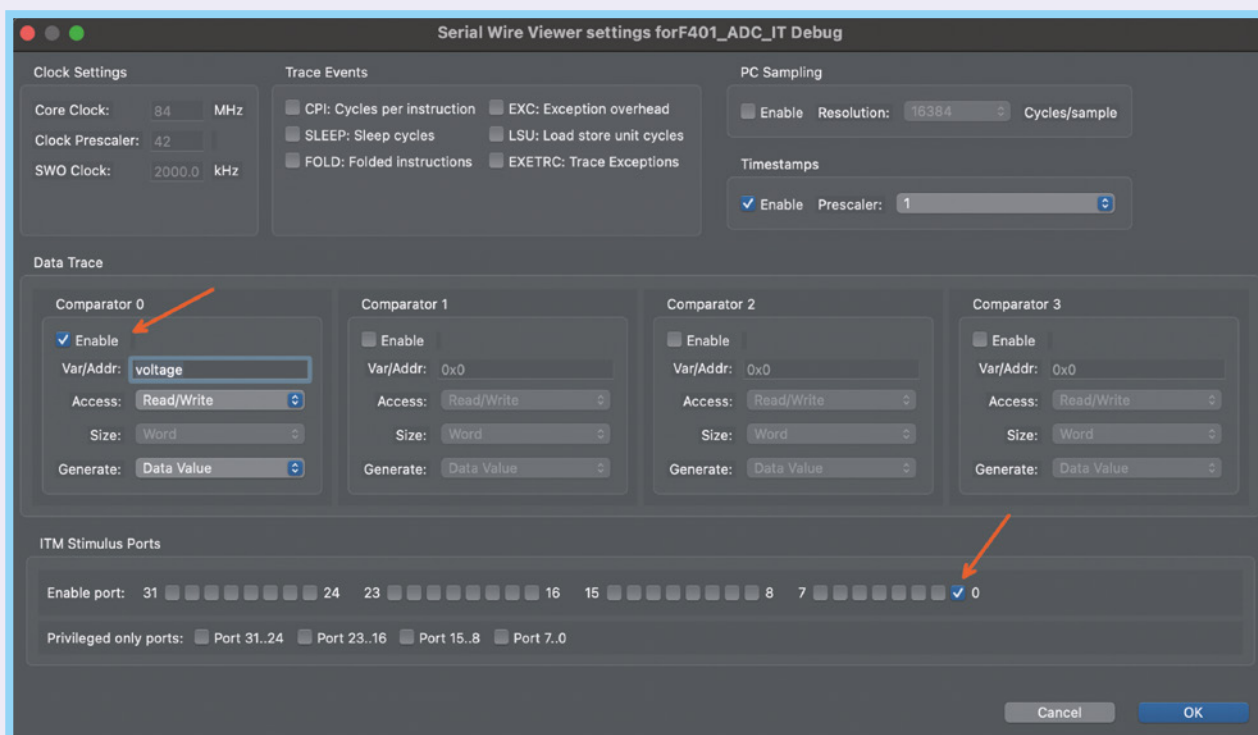


Fig. 25 Settaggio della SVW

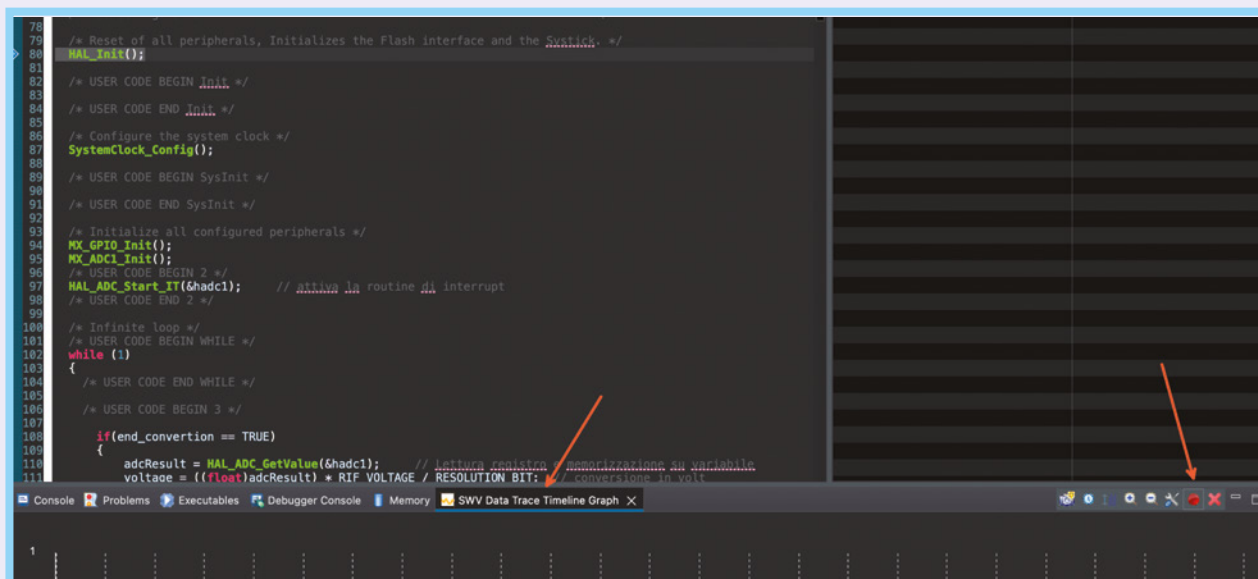


Fig. 26 Attivazione della start trace

Per raggiungere questo obiettivo, dalla barra menù, clicchiamo su 'Run' e successivamente su 'Debug Configurations' come mostrato in Fig. 20. Si aprirà la finestra di configurazione del debug come mostrato in Fig. 21.

Assicuriamoci di selezionare tutte le opzioni indicate dalle frecce prima di selezionare il tasto 'Debug'. Nella finestra 'Live Expressions' raffigurata in Fig. 22 aggiungiamo le variabili che desideriamo monitorare. Ad esempio abbiamo scelto le variabili utilizzate nella



Fig. 27 Grafico relativo alle letture del convertitore AD

Tipologia Timer	Timer	Risoluzione (bit)	Tipologia di contatore	Prescaler	Richiesta DMA	Canali	Uscita complementare	Max. Frequenza di interfaccia (MHz)	Max Frequenza Timer (MHz)
Avanzato	TIM1	16	Up, Down, Up/Down	1 ÷ 65536	Si	4	Si	84	84
General Purpose	TIM2 TIM5	32	Up, Down, Up/Down	1 ÷ 65536	Si	4	No	42	84
	TIM3 TIM4	16	Up, Down, Up/Down	1 ÷ 65536	Si	4	No	42	84
	TIM9	16	Up	1 ÷ 65536	No	2	No	84	84
	TIM10 TIM11	16	Up	1 ÷ 65536	No	1	No	84	84

Tabella 3 Timer presenti nell' STM32F401RE2

conversione analogico digitale 'adcResult' e 'voltage' e la variabile booleana 'end_conversion'.

Non ci resta che abilitare la SWV Data Trade Timeline Graph attraverso l'albero dei menù presentato in Fig. 23, ottenendo il risultato di Fig. 24.

Selezionando l'icona indicata dalla freccia in Fig. 24, verrà presentata la finestra di Fig. 25 dove andremo ad abilitare la sezione 'Comparator 0' ed inseriremo la variabile 'voltage' che il sistema implementerà graficamente. Per ultimo abilitiamo la porta zero come indicato dalla freccia sempre in Fig. 25.

Ora possiamo avviare il debug che attenderà un comando alla riga 80 per l'inizializzazione del sistema (Fig. 26).

Prima di avviare il debug dobbiamo assicurarci di selezionare la tab 'SWV Data Trade Timeline Graph' e attivare la 'start trace' come indicato dalle frecce in Fig. 26.

Avviamo il debug e ruotiamo il potenziometro da un estremo all'altro fino ad ottenere un grafico come raffigurato in Fig. 27.

L'obiettivo di questo semplice esempio vuole essere una graduale introduzione all'utilizzo della periferica ADC che può evolvere ancora in maniera esponenziale arrivando a memorizzare i dati acquisiti direttamente in DMA senza una diretta gestione della MCU come avviene nel nostro esempio.

LA PERIFERICA TIMER: UNA BREVE DESCRIZIONE

Le MCU STM32 sono provviste di più tipologie di timer prevedendo svariate situazioni di utilizzo.

I timer STM32 possono essere raggruppati in nove principali categorie, visibili in Tabella 2, dove troviamo:

Timer di base

Questa categoria di timer rappresenta la forma più semplice di timer utilizzati nei microcontrollori STM32.

Sono timer a 16 bit utilizzati come generatore di base temporale e non hanno pin di input/output.

I timer di base vengono utilizzati in abbinamento con la periferica DAC e possono essere utilizzati anche come "master" per altri timer.

Timer General Purpose

Sono timer a 16/32 bit (a seconda della serie STM32), forniscono le classiche funzionalità di un timer di un microcontrollore embedded moderno.

Vengono utilizzati in qualsiasi applicazione con il confronto in uscita (generazione di temporizzazioni e ritardi), modalità a impulso singolo, modalità di cattura in ingresso (per la misurazione della frequenza del segnale esterno), interfaccia sensore (encoder, sensore hall), ecc.

Ovviamente, un timer general-purpose può essere utilizzato come generatore di base temporale, come un timer di



base. I timer di questa categoria forniscono quattro canali di input/output programmabili.

Timer avanzati

Questi timer sono i più completi in un microcontrollore STM32. Oltre alle funzionalità presenti in un timer general-purpose, includono diverse funzionalità legate alle applicazioni di controllo motori e conversione di potenza digitale: tre segnali complementari con inserimento del tempo di ritardo, ingresso di interruzione di emergenza.

Timer ad alta risoluzione

Il timer ad alta risoluzione (HRTIM1) è un timer speciale implementato su alcuni microcontrollori della serie STM32F3/G4 (serie dedicate al controllo motori e alla conversione di potenza) e della serie STM32H7. Consente di generare segnali digitali con tempi di precisione elevata, come PWM o impulsi a fase sfalsata. È composto da 6 sottotimer, 1 master e 5 slave, per un totale di 10 uscite ad alta risoluzione, che possono essere abbinare a coppie. Include anche 5 ingressi di fault per scopi di protezione e 10 ingressi per gestire eventi esterni come la limitazione di

corrente, la commutazione a zero tensione o zero corrente. Il timer HRTIM1 è costituito da un kernel digitale con clock del core seguito da linee di ritardo.

Le linee di ritardo con controllo a loop chiuso garantiscono una risoluzione di 217 ps indipendentemente dalla tensione, dalla temperatura o dalla deviazione del processo di produzione chip-to-chip. L'alta risoluzione è disponibile sulle 10 uscite in tutti i modi di funzionamento: duty cycle variabile, frequenza variabile e tempo di ON costante.

Timer a basso consumo

I timer di questo gruppo sono appositamente progettati per le applicazioni a basso consumo. Grazie alla loro diversità di fonti di clock, questi timer sono in grado di funzionare con tutti i modi di alimentazione (eccetto la modalità Standby). Data questa capacità di funzionare anche senza una fonte di clock interna, i timer a basso consumo possono essere utilizzati come "contatore di impulsi". Hanno anche la capacità di riattivare il sistema dalla modalità a basso consumo.

Il processore STM32F401 include più timer. Utilizzando i timer è possibile generare forme d'onda, reagire a eventi esterni, misurare i parametri dei segnali tempo variabili in

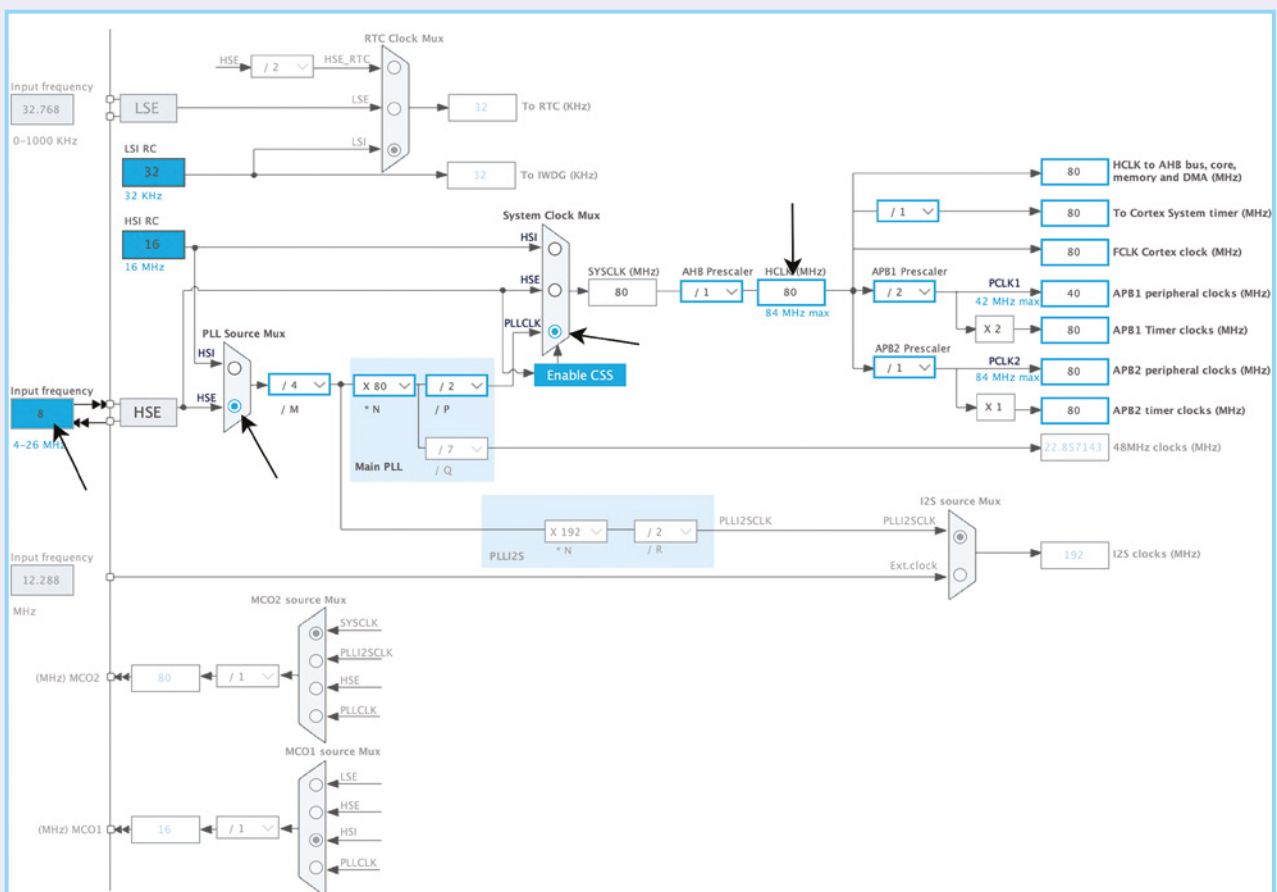


Fig. 28 Configurazione clock esterno

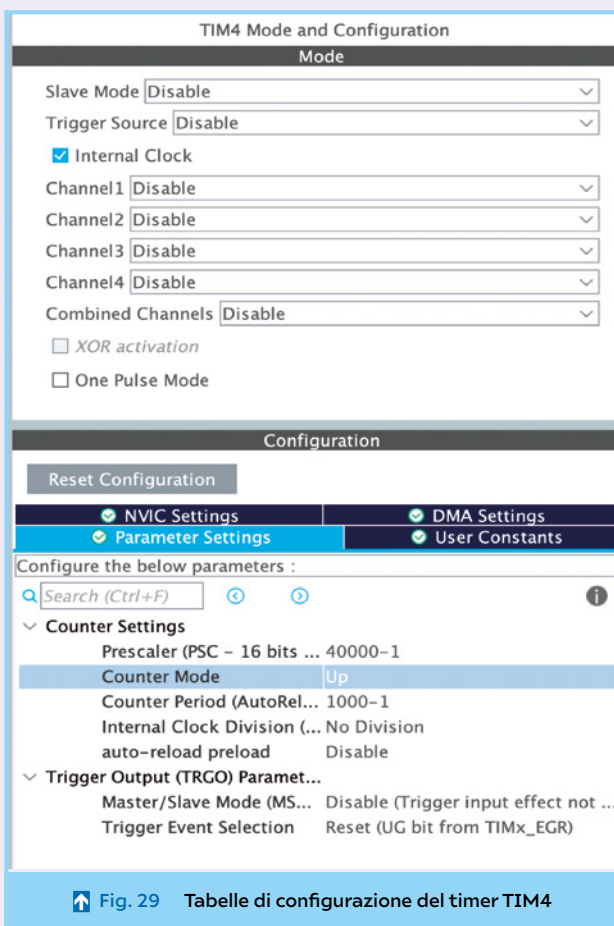


Fig. 29 Tabelle di configurazione del timer TIM4

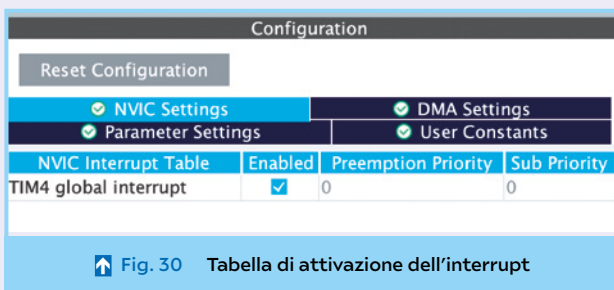


Fig. 30 Tabella di attivazione dell'interrupt

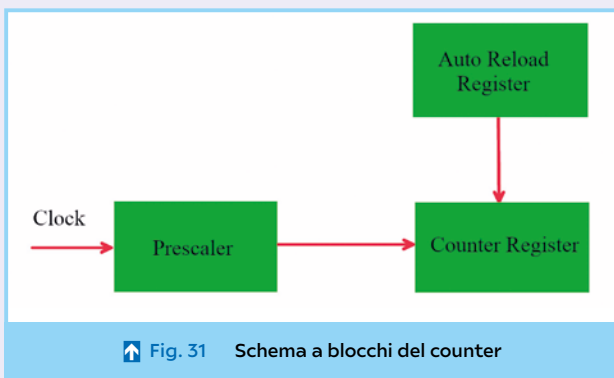


Fig. 31 Schema a blocchi del counter

ingresso e così via.

I timer funzionano in background, liberando così la CPU da compiti ripetitivi e critici dal punto di vista del tempo.

I timer sono tutti basati sulla stessa architettura e sono scalabili in base alla loro risoluzione (16 o 32 bit), al numero di canali (1 a 4), o alle loro caratteristiche (PWM, conteggio Up/Down, DMA).

I canali timer possono essere configurati come ingresso o uscita e più timer possono essere collegati tra loro e sincronizzati.

I timer possono essere utilizzati anche per fornire segnali di temporizzazione ad altri moduli all'interno del processore, come ad esempio l'ADC.

In **tabella 3** sono elencati i vari tipi di timer presenti nel processore STM32F401.

CONFIGURAZIONE IN MODALITÀ INTERRUPT

I Timer possono essere configurati in più modalità.

La modalità 'Interrupt' risulta la più efficiente quando si tratta di provocare un evento con un preciso ritardo e con un netto miglioramento dell'efficienza della MCU.

Nel prossimo esempio verrà introdotta questa tecnica di gestione del timer, che prevede l'utilizzo di alcuni parametri che andremo ad esaminare di seguito.

Per impostare i valori corretti dei parametri di seguito elencati occorre avvalersi delle seguenti formule:

$$Event_{Time} = \frac{(Prescaler + 1)(Period + 1)}{(Timer_{clock})}$$

dove:

- $Event_{Time}$ espresso in s rappresenta la periodicità dell'evento
- $Timer_{clock}$ espresso in Hz rappresenta la frequenza del bus a cui è collegato il timer
- $Prescaler$ espresso in ticks rappresenta il fattore di divisione
- $Period$ espresso in ticks rappresenta il coefficiente di periodicità del timer

Nel nostro esempio vogliamo ottenere un $Event_{Time}$ ogni 500 ms, impostando un $Timer_{clock}$ di 80 MHz per cui dovremo andare a definire $Prescaler$ e $Period$.

Definiamo un nuovo progetto nominandolo F401_Timer_IT seguendo la stessa procedura ormai acquisita e predisponiamoci ad effettuare una serie di modifiche che saranno fondamentali per ottenere una precisa commutazione. Innanzitutto utilizzeremo il quarzo a 8 MHz presente sulla scheda come generatore di clock e andremo ad impostare questa scelta nella tabella 'RCC Mode and Configuration' come già mostrata in **Fig. 6**.

Per completare le impostazioni selezioniamo 'Clock Configuration' e, come mostrato in **Fig. 28**, andremo a modifica-



```

69 int main(void)
70 {
71     /* USER CODE BEGIN 1 */
72
73     /* USER CODE END 1 */
74
75     /* MCU Configuration-----*/
76
77     /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
78     HAL_Init();
79
80     /* USER CODE BEGIN Init */
81
82     /* USER CODE END Init */
83
84     /* Configure the system clock */
85     SystemClock_Config();
86
87     /* USER CODE BEGIN SysInit */
88
89     /* USER CODE END SysInit */
90
91     /* Initialize all configured peripherals */
92     MX_GPIO_Init();
93     MX_USART2_UART_Init();
94     MX_TIM4_Init();
95     /* USER CODE BEGIN 2 */
96     HAL_TIM_Base_Start_IT(&htim4); // Start Timer in interrupt mode
97     /* USER CODE END 2 */
98

```

Fig. 32 File main.c: inizializzazione e attivazione del timer

re i parametri indicati dalle frecce.

Iniziamo impostando la frequenza di ingresso a 8 MHz, che è la frequenza fornita dal quarzo presente sulla scheda. Successivamente, scegliamo di utilizzare il percorso che collega l'ingresso HSE (High-Speed External) direttamente al blocco SYCLK (System Clock) del sistema. Per farlo, attiviamo gli ingressi HSE e PLLCLK (Phase-Locked Loop Clock). A questo punto non ci resta che riportare la frequenza del blocco HCLK a 80 MHz, digitando la cifra 80 all'interno del blocco e premendo il tasto Enter.

STM32CubeMX individuerà in automatico i parametri per ottenere la frequenza desiderata.

Il motivo per cui andiamo a modificare il valore della frequenza di clock è puramente didattico, utile per comprendere meglio i calcoli che andremo ad analizzare e osservare le modalità di programmazione della scheda Nucleo anche sotto questo aspetto.

In **Fig. 31** possiamo valutare in che modo i registri Prescaler e Counter Period Autoreload Register influiscono nel funzionamento del Timer.

Il Prescaler divide la frequenza di clock, mentre l'Auto Reload confronta il valore del counter con quello impostato nel suo registro e raggiunto lo reinizializza il counter.

Ipotizziamo di portare la frequenza all'ingresso del timer a

2 kHz impostando *Prescaler* a 39999 (40000-1) e valutiamo *Period*.

$$Period = \frac{Timer_{clock} * EventTime}{Prescaler + 1} - 1$$

$$Period = \frac{80E6 * 0.5}{39999 + 1} - 1 = 999$$

Il valore da impostare nel registro Counter Period (Auto Reload Register) sarà pari a 999 ticks (1000-1) e permetterà una commutazione del led ogni 0.5 s come nel precedente esempio.

In **Fig. 29** è mostrata la tabella 'Parameter Setting' dove introdurre i valori calcolati, riguardanti il Prescaler e il Counter Period. Nella parte 'Mode' occorre selezionare la spunta 'Internal Clock'.

Come mostrato in **Fig. 30** alla tab 'NVIC Setting' abilitiamo l'interrupt del timer, spuntando il campo 'Enabled'. Ricordiamo che il formato introdotto (40000-1 oppure 1000-1) è un modo per mettere in risalto il fatto che il valore 0 per questi due registri vale 1, come fattore matematico. Procediamo con l'analisi del software.

In **Fig. 32** alla riga 96 osserviamo la funzione 'HAL_TIM_Base_Start_IT()' che attiva il TIM4 e l'interrupt.

Nell'istante in cui il Counter Register raggiunge il valore

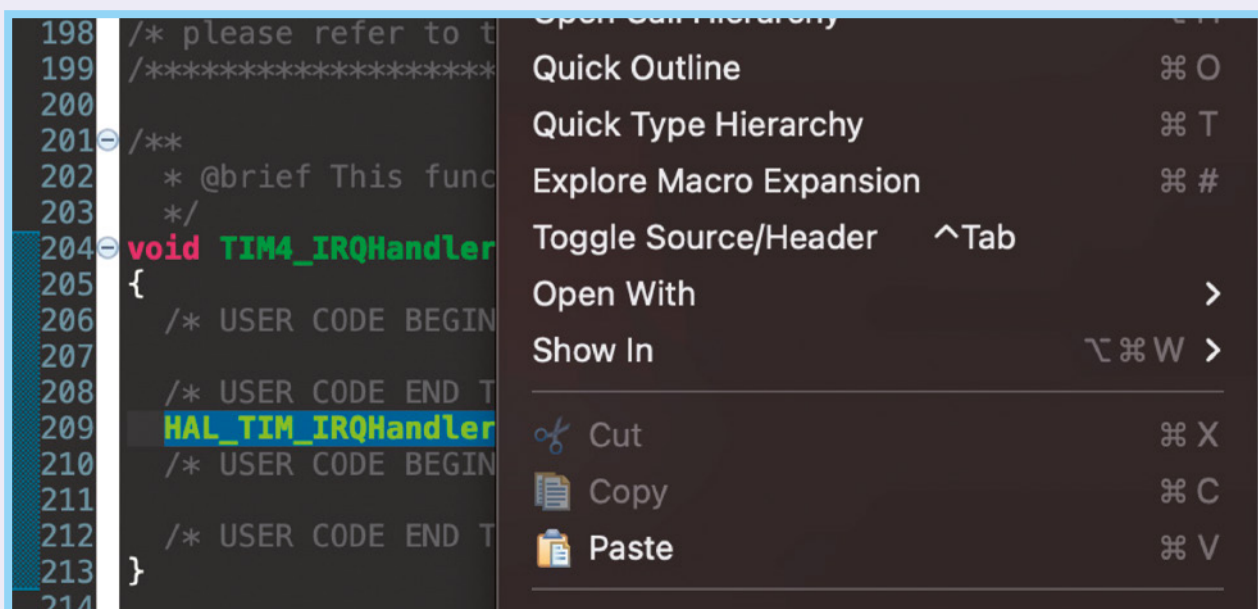


Fig. 33 File stm32f4xx_it.c: ricerca della funzione HAL_TIM_IRQHandler()

memorizzato in Auto Reload Register viene attivato l'interrupt che richiama la funzione 'TIM4_IRQHandler()' nel file 'stm32f4xx_it.c' all'interno della directory Core->Src. Utilizzando il comando 'Open Declaration' mostrato in **Fig. 33** continuiamo a seguire la catena delle chiamate che ci porta nel file 'stm32_hal_tim.c' alla funzione 'HAL_TIM_IRQHandler()' che gestisce le richieste di interrupt. In particolare alla riga 3941 (**Fig. 34**) della stessa funzione viene gestito l'evento del timer che richiama la funzione 'HAL_TIM_PeriodElapsedCallback()'. Alla riga 5645 sempre del file 'stm32_hal_tim.c' (**Fig. 35**) troviamo un'esempio della funzione 'HAL_TIM_PeriodElapsedCallback()' che dovremo copiare e riportare nel file main.c.

La **Fig. 36** riporta la funzione e le righe di processo per la gestione dell'interrupt. Dopo aver verificato che TIM4 ha segnalato l'evento effettueremo una commutazione della porta PA5 che andremo a monitorare con un analizzatore di segnali logici. Per la configurazione della porta PA5 rimandiamo il lettore alla prima puntata (rivista EI277 pag.149). Il documento STM "UM1725" 'Description of STM32F4 HAL and low-layer drivers' (reperibile al link: www.st.com/resource/en/user_manual/um1725-description-of-stm32f4-hal-and-lowlayer-drivers-stmicroelectronics.pdf) elenca e descrive tutti i driver API per l'uso delle periferiche dei microcontrollori serie F4, in particolare il capitolo 68 pag. 1059 è dedicato alla periferica Timer .

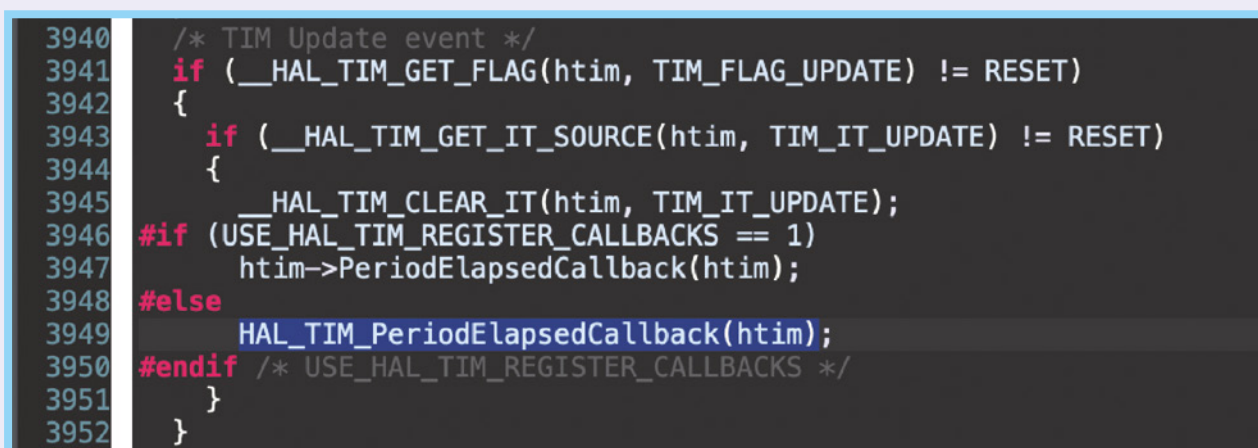


Fig. 34 File stm32_hal_tim.c: gestione dell'evento con chiamata funzione HAL_TIM_PeriodElapsedCallback()


```

5640 /**
5641  * @brief Period elapsed callback in non-blocking mode
5642  * @param htim TIM handle
5643  * @retval None
5644  */
5645 weak void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
5646 {
5647     /* Prevent unused argument(s) compilation warning */
5648     UNUSED(htim);
5649
5650     /* NOTE : This function should not be modified, when the callback is needed,
5651     the HAL_TIM_PeriodElapsedCallback could be implemented in the user file
5652     */
5653 }

```

Fig. 35 File stm32_hal_tim.c: esempio di funzione utile alla copia

```

270
271 /* USER CODE BEGIN 4 */
272 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
273 {
274     if(htim == &htim4) // verifica quale timer ha richiesto l'interrupt
275     {
276         HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin);
277     }
278 }
279 /* USER CODE END 4 */
280
281

```

Fig. 36 File main.c: realizzazione della funzione di gestione dell'interrupt

In questo documento sono elencate le diverse tipologie di chiamate su interrupt. Nel nostro caso è stata utilizzata la 'HAL_TIM_PeriodElapsedCallback', elencata nel manuale a pag. 1069 ma, come si può notare, ne sono state sviluppate una nutrita serie, riuscendo così ad utilizzare appieno le caratteristiche dei timer.

Consultando questo manuale possiamo osservare l'elenco degli HAL driver, sfruttando così le caratteristiche messe a disposizione dalla MCU.

Avendo terminato la realizzazione e l'analisi del software possiamo compilare e avviare il debug e verificare la funzionalità del software.

Il led, ad intervalli di 500 ms, commuta il suo stato e, monitorando con un analizzatore di stati logici, otteniamo una

commutazione con una precisione notevole, come possiamo osservare in **Fig. 37**.

CONCLUSIONI

In questa puntata abbiamo esplorato due periferiche importanti: il convertitore analogico/digitale e il Timer.

Torneremo a parlare del Timer nel prossimo episodio, quando introdurremo la tecnica della modulazione di larghezza di impulso (Pulse Width Modulation, PWM).

Inoltre, esamineremo il bus seriale, analizzando le sue configurazioni sia hardware che software.

Approfondiremo questi argomenti attraverso test pratici ed esempi software, che ci aiuteranno a comprendere meglio come sfruttare vari tipi di interrupt. □

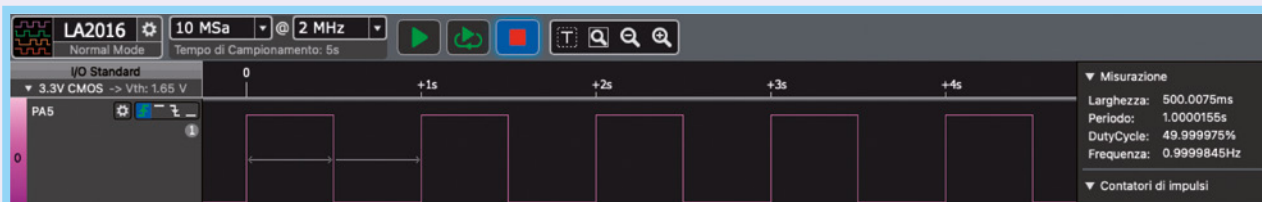


Fig. 37 Analisi stato logico porta PA5

Entra nel mondo



cod. KSR19
€ 44,00

Robot con codifica meccanica 5 in 1 - in kit

Robot didattico in scatola di montaggio con codifica STEM (Science, Technology, Engineering, Mathematics) particolarmente adatto per principianti, pensato per stimolare l'utente ad avvicinarsi all'ingegneria di costruzione delle macchine con codifica manuale.

cod. SOLARDINO
€ 15,00



Robot Solare 3 in 1 - in kit

Particolarmente indicato come gioco didattico per bambini a partire dagli 8 anni e non solo, questo kit permette di assemblare ed ottenere 3 diversi robot alimentati ad energia solare, tra cui un dinosauro, un insetto e una trivella. Quando il sole non è disponibile è possibile utilizzare una luce alogena come alimentazione.



cod. KSR10N
€ 58,00

Interfaccia USB
per braccio robotico

cod. KSR10/USB | € 53,00

Braccio robotico

Braccio meccanico, dotato di cinque motori, cinque articolazioni con unità di controllo via cavo a cinque comandi.

cod. KSR18
€ 45,00



Robot TOBBIE II - in kit

Può funzionare in due differenti modalità: "Seguimi" e "Esplora". Nella modalità "Seguimi" è in grado di seguire qualsiasi cosa gli si avvicini. Quando funziona in modalità "Esplora" è in grado di evitare gli ostacoli che trova davanti a sé trovando sempre un nuovo percorso alternativo.

della robotica

cod. KSR20
€ 92,00



Robot TOBBIE II con MICRO:BIT - in kit

Robot a 6 gambe con corpo libero di ruotare, dotato di tutti i componenti elettronici necessari per renderlo attivo e interattivo. Ideale in combinazione con Micro:bit. È possibile programmarlo in tutta semplicità con Javascript Blocks Editor o Python Editor sul PC portatile o smartphone.

cod. KSR14
€ 24,50



Robot SOLAR ROVER - in kit

Particolarmente indicato come gioco didattico per bambini (da 8 anni in su). Questo kit permette di assemblare un mini rover solare con 6 ruote. Dispone di un sistema di trasmissione a ingranaggi planetari, piccoli bracci robotici, un pannello solare e un piccolo motorino.

cod. KSR17
€ 38,00



Robot solare e idraulico 12 in 1 - in kit

Permette di realizzare fino a 12 differenti modelli di robot con movimento idraulico e alimentati tramite energia solare. Il kit viene fornito completo di un manuale ricchissimo di illustrazioni.

cod. SOLARDOG
€ 23,00



Cane Robotico in kit

Particolarmente indicato come gioco didattico (da 8 anni in su), il kit permette di assemblare un cane robotico alimentato da 2 batterie o da una cella solare. Un motore DC permette al robot di muoversi.

cod. KSR16
€ 32,00



Kit per auto con motore ad aria

Kit di montaggio per auto con motore alimentato ad aria compressa. È composto da 102 parti di montaggio estremamente facili da assemblare tra di loro. La macchinina è in grado di percorrere fino a 50 metri di distanza.



Batterie efficienti e affidabili con la diagnostica predittiva

Un software innovativo per il monitoraggio avanzato e la diagnostica delle batterie: è l'oggetto della collaborazione avviata da Enel Green Power con Volytica diagnostics GmbH, pioniera tedesca della diagnostica delle batterie, per migliorare l'efficienza e la sicurezza dei sistemi di accumulo energetico. Le batterie di accumulo sono fondamentali per la continuità dell'energia ottenibile dalle fonti rinnovabili come eolico e fotovoltaico, che necessitano, per essere pienamente valorizzate e integrate nel sistema elettrico, di sistemi in grado di immagazzinare l'energia prodotta in eccesso nei momenti di disponibilità di sole e vento, per restituirla nei momenti di picco dei consumi.

Lo storage è indispensabile per lo sviluppo delle rinnovabili: il punto è come assicurarne la necessaria affidabilità e flessibilità per le esigenze dell'utente finale.

Le batterie integrate con gli impianti di generazione, come tutti i componenti industriali, vanno gestite al meglio per garantirne una vita più lunga ed efficiente, ottimizzando i continui cicli di carica e scarica di energia ai quali sono sottoposte. Il primo passo per fare ciò è seguirne il comportamento e le prestazioni con attenzione e cura in tempo reale.

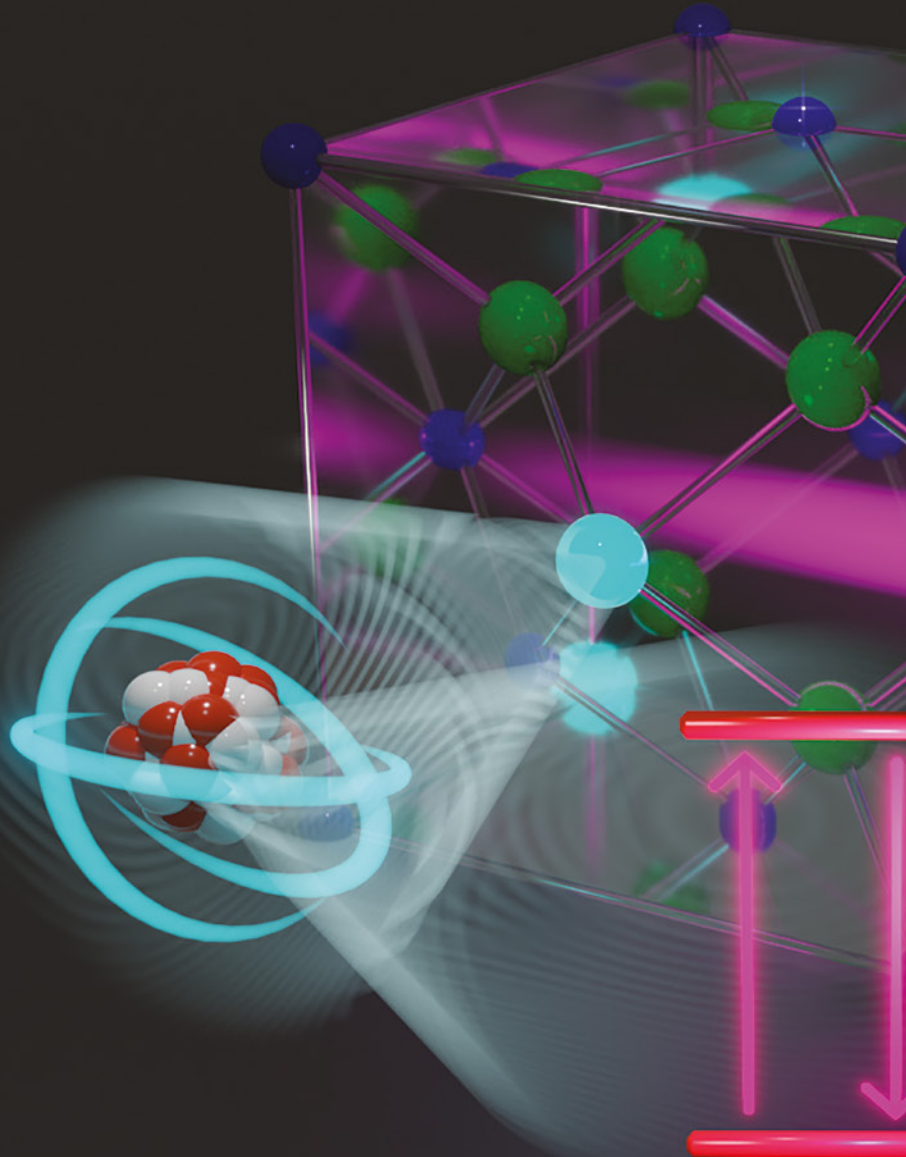
Il software di Volytica identifica e analizza gli indicatori critici di prestazione delle batterie, anticipando l'individuazione di eventuali anomalie. Ciò consente di intervenire nella gestione di eventuali problemi e differenze di funzionamento tra i diversi elementi che compongono ogni batteria, e garantire la continuità del servizio in modo efficiente. Tale soluzione verrà validata per un anno nell'ambito di un progetto coinvolgente cinque impianti situati sia in Italia che all'estero.

www.enelgreenpower.com

Il laser eccita il nucleo dell'atomo

Da molti anni gli scienziati di tutto il mondo sono alla ricerca di uno stato specifico dei nuclei atomici di torio che promette applicazioni rivoluzionarie, come la creazione di un orologio nucleare in grado di misurare il tempo in modo più preciso rispetto ai migliori orologi atomici oggi disponibili. Tale stato potrebbe anche aiutare a rispondere a domande fondamentali completamente nuove in Fisica, come ad esempio capire se le costanti della natura siano effettivamente costanti o se cambino nello spazio e nel tempo.

Recentemente, è stato usato un laser per portare un nucleo atomico in uno stato di energia più elevato. Successivamente, è stato possibile osservare e monitorare con precisione come il nucleo tornasse al suo stato energetico originale. Ciò rende possibile unire due aree della Fisica che prima avevano poco a che fare l'una con l'altra: la Fisica quantistica classica e la Fisica nucleare. Un prerequisito cruciale per questo successo è stato lo sviluppo di speciali cristalli contenenti torio. Un gruppo di ricerca guidato dal Prof. Thorsten



Schumm della TU Wien (Vienna) è alla base della scoperta.

Oggi la manipolazione di atomi o molecole con i laser è una pratica comune: se la lunghezza d'onda del laser viene scelta esattamente, gli atomi o le molecole possono passare da uno stato all'altro. In questo modo è possibile misurare in modo molto preciso l'energia degli atomi o delle molecole. Su questo si basano molte tecniche di misurazione di precisione, come quella degli attuali orologi atomici, ma anche metodi di analisi chimica. I laser vengono spesso utilizzati anche nei computer quantistici per archiviare informazioni in atomi o molecole.

Per molto tempo, però, sembrò impossibile applicare queste tec-

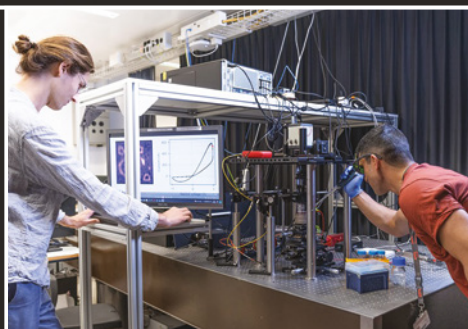
niche ai nuclei atomici, perché per farli passare da uno stato quantico all'altro occorre almeno mille volte l'energia degli elettroni in un atomo o in una molecola, che i fotoni dei laser non riescono a cedere. Dagli anni '70 del secolo scorso si ipotizza che un determinato nucleo atomico possa essere manipolato con un laser: quello del torio-229. Questo nucleo ha due stati energetici molto vicini, tanto che un laser dovrebbe essere sufficiente per cambiare lo stato del nucleo, ma il problema è che per ottenere tale transizione bisogna conoscere con estrema precisione l'energia corrispondente.

Alcuni gruppi di ricerca hanno provato a studiare i nuclei di torio tenendoli individualmente in posizione in trappole elettromagnetiche. Tuttavia, Thorsten Schumm e il suo team hanno adottato un altro approccio: hanno sviluppato cristalli in cui sono incorporati numerosi atomi di torio, in modo da poterne colpire contemporaneamente circa 10^{17} , il che riduce il tempo di misurazione richiesto e aumenta la probabilità di trovare la transizione energetica.

Ciò segna l'inizio di una nuova ed entusiasmante era di ricerca: ora che il team sa come eccitare lo stato del torio, questa tecnologia può essere utilizzata per misurazioni di precisione come quella del tempo, attraverso la costruzione di un orologio atomico al torio. Similmente a come un orologio a pendolo utilizza l'oscillazione del pendolo come base tempi, l'oscillazione della luce che eccita la transizione del torio potrebbe essere utilizzata come timer per un nuovo tipo di orologio che sarebbe significativamente più preciso dei migliori orologi atomici disponibili oggi.

Ma ciò permette non solo di misurare il tempo in modo più preciso rispetto a prima; ad esempio consentirebbe un'analisi del campo gravitazionale della Terra tanto precisa da fornire indicazioni sull'esistenza di risorse minerarie o terremoti. Il metodo di misurazione potrebbe anche essere utilizzato per andare a fondo dei misteri fondamentali della fisica e capire se le costanti della natura sono davvero costanti. Inoltre è possibile misurare piccoli cambiamenti nel tempo.

www.tuwien.at/en/tu-wien/news



Circuiti liquidi per imitare il cervello

I circuiti liquidi che imitano le sinapsi nel cervello possono ora eseguire il tipo di operazioni logiche alla base dei computer. Le applicazioni di questi dispositivi potrebbero includere attività come il riconoscimento delle immagini, nonché calcoli alla base della maggior parte dei sistemi di intelligenza artificiale. Proprio come i neuroni biologici elaborano e memorizzano dati, la tecnologia neuromorfica che imita il cervello spesso combina entrambe le operazioni. Tali dispositivi possono ridurre notevolmente l'energia e il tempo persi nei microchip convenzionali che trasportano i dati avanti e indietro tra i processori e la memoria. Potrebbero anche rivelarsi ideali per l'implementazione di reti neurali.

Il cervello umano funziona a potenze inferiori a quelli dei computer richiesti per svolgere compiti equivalenti. Una strategia per sviluppare computer neuromorfici coinvolge i memristori, che sono essenzialmente interruttori in grado di ricordare in quale stato elettrico sono stati commutati dopo aver tolto l'alimentazione; perciò somigliano alle sinapsi che collegano i neuroni, la cui conduttività elettrica si rafforza o si indebolisce a seconda di una serie di fattori biofisici, inclusa la quantità di carica elettrica che li ha attraversati in passato. I memristori fluidici sperimentati all'EPFL di Losanna si basano invece sullo spostamento degli ioni in un chip di silicio con al centro una membrana di nitrato di silicio avente un poro circolare largo circa 100 nm; sopra si trovano strati di grafite e palladio (con all'interno una rete di canali larghi pochi nanometri). Immettendo una soluzione acquosa di ioni potassio e applicando tensione, gli ioni fluiscono dal poro creando una bolla sotto la grafite, spingendo la quale il dispositivo diventa più conduttivo, andando in stato ON.

actu.epfl.ch/news

Dal sudore si conosce lo stato di salute

Il sudore può trasportare sali e altre molecole utilizzabili come indicatori dello stato di salute. Dei ricercatori in Cina hanno sviluppato un sistema di sensori indossabili in grado di raccogliere ed

elaborare il sudore; il sistema si chiama "patch nanoelettronico di grafene intelligente interfacciato con la pelle", o SIGN. Affinché questo cerotto funzioni, il sudore deve essere trasportato dalla pelle nelle camere microfluidiche, dove una membrana speciale rimuove le impurità. Il liquido così ottenuto viene consegnato a un biorecettore che può essere sintonizzato per rilevare diversi metaboliti. Il sistema di trasporto si basa su una combinazione di materiali idrofili e idrofobi e può spostare soluzioni acquose



lungo microcanali, anche contro la gravità. Ciò rende possibile trasportare piccoli campioni con precisione, indipendentemente dall'orientamento del dispositivo. Il fluido viene

portato in una membrana Janus, dove le impurità vengono bloccate, in modo da fornire risultati accurati. Il sudore purificato arriva al biosensore flessibile in grafene, attivato da enzimi adatti a rilevare il biomarcatore desiderato. Il risultato è un transistor in grado di misurare la quantità di biomarcatore, che nello specifico è il lattato, un metabolita che può essere utilizzato per valutare i livelli di esercizio e affaticamento di una persona.

www.sciencedirect.com



Il transistor è riconfigurabile

Un gruppo di ricerca della Vienna University of Technology ha creato una nuova generazione di transistor riprogrammabili che potrebbe rivoluzionare l'elaborazione dei dati.

Nei transistor convenzionali si arricchiscono le regioni N di elettroni e le P di lacune, per consentire lo scorrimento di corrente; questo drogaggio è immutabile nel corso della vita del componente.

I transistor riconfigurabili, invece, sfruttano un drogaggio elettrostatico. Si tratta di un metodo che sostituisce il complesso e costoso processo di doping chimico con campi elettrici. Controllando il trasporto di elettroni e lacune nel materiale attraverso campi elettrici, questi transistor possono essere riconfigurati dinamicamente per svolgere compiti diversi.

Il gruppo di ricerca ha presentato per la prima volta questi transistor configurabili nel 2021.

La svolta è che ora ha dimostrato come tutti i circuiti logici di base possano essere assemblati utilizzando questi transistor riconfigurabili. Inoltre, i relativi circuiti logici possono essere convertiti in altri circuiti riconfigurando i componenti. Le celle logiche elementari realizzate indicano che è possibile fabbricare circuiti generici e che la loro effettiva funzione può essere definita dopo la fabbricazione. Il drogaggio elettrostatico richiede però elettrodi aggiuntivi che aumentano l'ingombro dei transistor rispetto a quelli tradizionali, che non verranno soppiantati ma affiancati per ottenere logiche riconfigurabili.

www.tuwien.at

Intel sfida AMD e Nvidia sull'AI

Intel ha presentato l'acceleratore Gaudi 3 AI, che offre elaborazione AI 4x per BF16 ed elevate prestazioni per il training AI e l'inferenza sui modelli linguistici di grandi dimensioni (LLM) e su quelli multimodali. Gaudi 3 AI, l'unica alternativa sul mercato con benchmark MLPerf per LLM, permette di scegliere tra software open e reti Ethernet standard per scalare i sistemi. Gaudi 3 è basato su un processo a 5 nm ed è progettato per consentire l'attivazione di tutti i

motori in parallelo: Matrix Multiplication Engine (MME), Tensor Processor Core (TPC) e Networking Interface Card (NIC). Ogni acceleratore presenta un motore di calcolo eterogeneo composto da 64 TPC programmabili e personalizzati tramite AI ed otto MME in grado di eseguire 64.000 operazioni parallele, supportando più tipi di dati, inclusi FP8 e BF16. Gaudi 3 ha 128 GB di memoria HBM2e, 3,7 TB di larghezza di banda della memoria e 96 MB di SRAM, per l'elaborazione di set di dati GenAI di grandi dimensioni con un numero ridotto di dispositivi. In ogni Gaudi 3 vi sono 24 porte ethernet a 200 Gb. Il software integra il framework PyTorch e fornisce modelli Hugging Face, il framework AI più comune oggi per gli sviluppatori GenAI. Gaudi 3 offrirà rispetto al Nvidia H100 un tempo di addestramento minore del 50% sui parametri Llama2 7B e 13B e sui modelli di parametri GPT-3, oltre a un throughput di inferenza più veloce del 50%.

www.intel.com



ANCHE I DETTAGLI PIÙ PICCOLI DIVENTANO VISIBILI!

**MICROSCOPI DIGITALI IN GRADO DI CATTURARE IMMAGINI
CON UN'INGRANDIMENTO FINO A 1200X.**

MICROSCOPIO DIGITALE 1-1200X CON MONITOR LCD A COLORI DA 7"

Microscopio digitale con ampio monitor LCD a colori da 7 pollici e ingrandimento fino a 1200X. Dispone di una telecamera a colori che riprende l'oggetto da ingrandire e lo visualizza in alta risoluzione sullo schermo LCD.

Le foto e i video degli oggetti ingranditi possono essere salvati su una SD card inserita nell'apposito slot (max 128GB, acquistabile separatamente). Il microscopio dispone di una base stabile in alluminio e di doppia luce a LED con bracci flessibili. La distanza del punto di osservazione del campione può essere facilmente regolata tramite l'asta a cremagliera che consente di ottenere una distanza che va da 10 a 130 mm. Lo schermo è ruotabile di 360° e inclinabile. Può funzionare tramite batteria al litio integrata oppure tramite l'alimentatore USB in dotazione.



cod. LCDMICROS7

€154,00



Prezzi IVA inclusa.

MICROSCOPIO DIGITALE CON INGRANDIMENTO FINO A 600X CON MONITOR LCD DA 4,3"

Microscopio digitale portatile, dotato di telecamera a colori con sensore da 3,6 Megapixel, monitor LCD a colori da 4,3", illuminatore composto da 8 potenti LED a luce bianca, batteria al litio ricaricabile integrata, slot per micro SD card (max 64GB) e staffa a ventosa.



cod. LCDMICROS

€102,00

MICROSCOPIO USB DIGITALE CON INGRANDIMENTO FINO A 500X

Microscopio USB digitale a colori da 2 Megapixel dotato di stand con supporto regolabile in altezza. Si collega alla porta USB e consente di catturare e visualizzare immagini e video direttamente sullo schermo del computer. Dispone di 8 LED bianchi, con intensità luminosa regolabile.



cod. CP745

€28,00

IoTicontrollo

DISPOSITIVI INNOVATIVI STUDIATI PER METTERE IN COMUNICAZIONE APPARATI E OGGETTI CON LA RETE INTERNET PER SEMPLIFICARE LA VITA DI TUTTI I GIORNI.

POWER CONTROL IOT



cod. PWC3NV000

€129,00

Collegato alla rete elettrica, permette di essere avvisati da remoto, tramite apposita APP, di una eventuale mancanza di corrente e di intervenire tempestivamente. Il dispositivo dispone anche di una uscita a contatto pulito molto utile per attivare da remoto dispositivi elettrici. Dispone di una SIM on chip con durata di connettività illimitata.

SECURITY IOT BOX



cod. SEC3NV000

€129,00

Ideale per proteggere gli oggetti dai furti. Basato su un accelerometro, Security IoT BOX è in grado di rilevare e segnalare tempestivamente, tramite l'apposita APP, eventuali spostamenti degli oggetti sui quali viene posizionato. Per funzionare non necessita di Bluetooth o Wi-Fi; dispone infatti di una SIM on chip con durata di connettività illimitata. Funziona con due batterie a stilo tipo AA (incluse). Autonomia circa 18 mesi.

SMOKE DETECTOR IOT



cod. SMK3NV000

€159,00

È in grado di rilevare la presenza di fumi potenzialmente pericolosi segnalandolo, tempestivamente tramite l'apposita APP. Per funzionare non necessita di Bluetooth o Wi-Fi; dispone infatti di una SIM on chip con durata di connettività illimitata. Funziona con una batteria al litio da 3V (incluse). Autonomia circa 18 mesi.

CONNETTIVITÀ ILLIMITATA

nessun abbonamento



APP GRATUITA



NUOVI ALIMENTATORI SWITCHING DA RETE

COD. ER24W24V

€ 10,50



ALIMENTATORE SWITCHING
24V / 1A - 24W

Tensione di ingresso: da 90 a 240 VAC - 50/60 Hz
Potenza: 24 W
Tensione in uscita: 24 Vdc
Massima corrente di uscita: 1 A
Tensione di ripple: 200 mVpp
Consumo di potenza a vuoto (max): 0.075 W
Lunghezza del cavo: 1,5 metri
Plug: 5,5 x 2,1 mm, positivo centrale
Dimensioni (mm): 82,3 x 32,6 x 74,1 (L x W x H)
Peso: 112 grammi

ALIMENTATORE SWITCHING
6V / 1A - 6W

Tensione di ingresso: da 90 a 240 VAC - 50/60 Hz
Efficienza: 75%
Potenza: 6 W
Tensione in uscita: 6 Vdc
Massima corrente di uscita: 1 A
Tensione di ripple: 200 mVpp
Consumo di potenza a vuoto (max): 0.075 W
Lunghezza del cavo: 1,5 metri
Plug: 5,5 x 2,1 mm, positivo centrale
Temperatura di lavoro: 0°C ÷ 40°C
Dimensioni (mm): 38 x 25 x 69,6 (L x W x H)
Peso: 80 grammi

COD. ER06W06V

€ 7,00



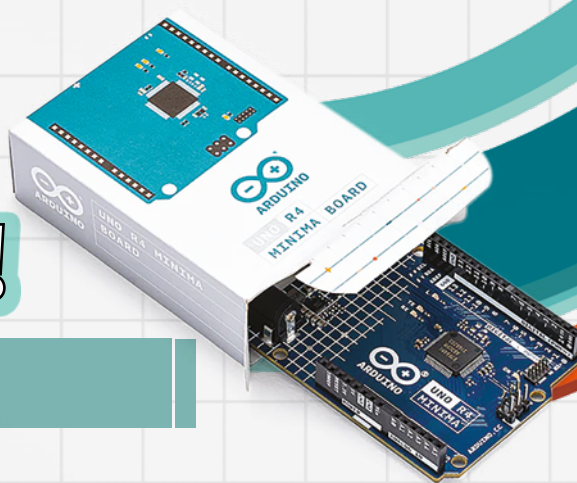
Prezzi IVA inclusa.

**VASTA GAMMA DI MODELLI
PER TUTTE LE ESIGENZE**

DISPONIBILI ANCHE CON LE SEGUENTI CARATTERISTICHE:

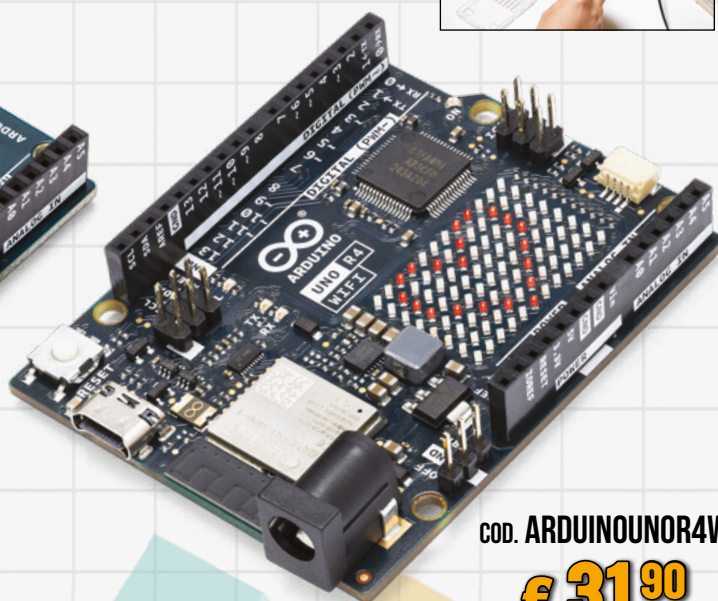
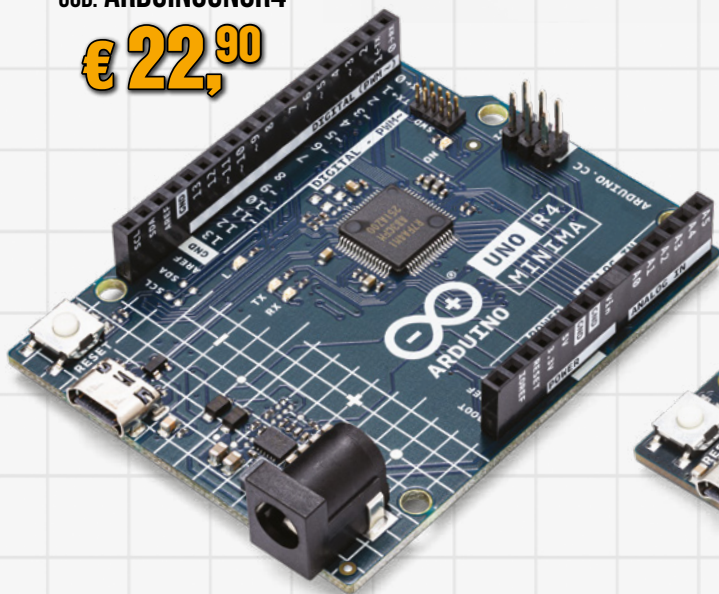
CODICE		TENSIONE D'INGRESSO	TENSIONE D'USCITA	CORRENTE D'USCITA	POTENZA	RIPPLE	DIMENSIONI	PREZZO
ER09W09V		90 a 240 VAC	9 VDC	1 A	9 W	150 mVpp	76,2 x 27,6 x 61,6 mm	7,00
ER12W12V		90 a 240 VAC	12 VDC	1 A	12 W	150 mVpp	76,2 x 27,6 x 61,6 mm	7,50
ER15W15V		90 a 240 VAC	15 VDC	1 A	15 W	200 mVpp	79,2 x 27,6 x 69,1 mm	8,50
ER24W12V		90 a 240 VAC	12 VDC	2 A	24 W	120 mVpp	82,3 x 32,6 x 74,1 mm	11,50

Scopri Arduino UNO R4!!!



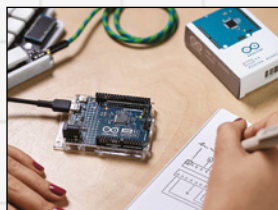
COD. ARDUINOUNOR4

€ 22,90



COD. ARDUINOUNOR4W

€ 31,90



Prezzi IVA inclusa.

CARATTERISTICHE TECNICHE		UNO R4 MINIMA	UNO R4 WiFi
MCU	R7FA4M1AB3CFM#AA0 (Arm cortex M4)	✓	✓
Memoria	256KB Flash / 32KB SRAM	✓	✓
Tensione operativa	5V	✓	✓
Tensione d'ingresso	6-24V	✓	✓
Velocità Clock	48 MHz	✓	✓
Porta di programmazione	USB-C	✓	✓
WiFi/BLE	ESP32-S3-MINI		✓
LED Matrix	12x8 (96 LED rossi)		✓
Connessioni supplementari	-connettore Qwiic -pin OFF -pin VRTC		✓
Pin digitali	14	✓	✓
PWM	6	✓	✓
ADC	6	✓	✓
DAC	1 (12 bit)	✓	✓
SPI	1	✓	✓
I2C	1	✓	✓
CAN	1	✓	✓