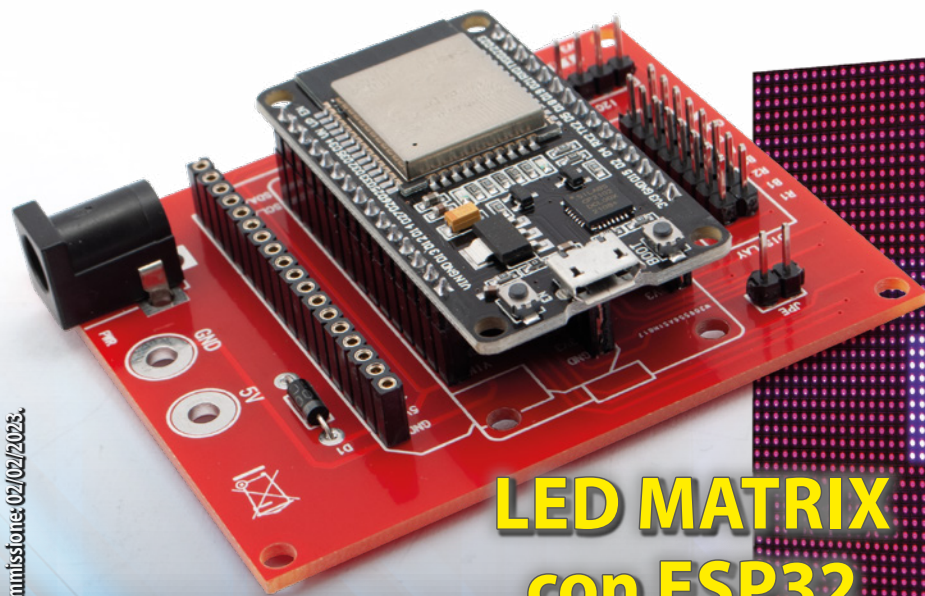


Elettronica In

www.elettronica.in.it

€ 7,50 - Anno XXIX - n. 271 - FEBBRAIO / MARZO 2023



160 PAGINE

12 PROGETTI

LED MATRIX con ESP32



- ▶ Protezione per casse acustiche
- ▶ Misuriamo la risposta in frequenza
- ▶ Breakout I/O expander
- ▶ Generatore di funzioni da laboratorio
- ▶ Breakout con sensore di temperatura
- ▶ Espansione per allarmino
- ▶ Cronometro per atletica
- ▶ Controllo automatico per ventilazione
- ▶ Bleb invaders
- ▶ Contapassi elettronico
- ▶ Soluzioni nell'auto elettrica

Sveglia parlante



Blocca il punto a LED



Indispensabili in ogni laboratorio!

STAZIONE SALDANTE
ANALOGICA
REGOLABILE 48W

cod. VTSS4N
€22,90



STAZIONE SALDANTE/
DISSALDANTE PER
COMPONENTI SMD
CON STILO
AD ARIA CALDA

cod. VTSS230
€165,00



STAZIONE SALDANTE
REGOLABILE 30-130W
CON DISPLAY E ACCESSORI

cod. TP237
€49,00



STAZIONE SALDANTE DA 48W
CON INDICAZIONE
DELLA TEMPERATURA
IMPOSTATA-REALE
E DISPLAY LCD

cod. VTSSC40N
€94,00



STAZIONE SALDANTE CON
ELEMENTO RISCALDANTE IN
CERAMICA DA 48W
E DISPLAY LCD

cod. TIP027
€52,00



STAZIONE SALDANTE
DA 60-130W CON
ASPIRATORE DI FUMO,
DISPLAY LCD E
LUCE A LED

cod. TP240
€139,00



Prezzi IVA inclusa.

**FUTURA
ELETTRONICA®**

Futura Group srl
Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775

Caratteristiche tecniche
e vendita on-line su:

www.futurashop.it



Conversare con l'Intelligenza Artificiale

L'intelligenza artificiale conversazionale si è evoluta fino a riuscire a rispondere agli input del linguaggio naturale dell'uomo. Tra le chat di maggior successo che utilizzano l'AI c'è ChatGPT, acronimo di Generative Pretrained Transformer che con naturalezza risponde alle domande che le vengono poste online. ChatGPT utilizza algoritmi avanzati per apprendere dalle stesse conversazioni e per generare risposte simili a quelle umane. Il livello della conversazione è talmente alto che i leader della Conferenza internazionale sull'apprendimento automatico (ICML)

ChatGPT è il primo esempio di AI accessibile a tutti

hanno vietato agli scienziati di presentare documenti creati con modelli di linguaggio di grandi dimensioni (LLM) come ChatGPT. Quando la tecnologia compie grandi passi in avanti subito si pensa che l'uomo verrà sostituito e che non ci saranno più

programmatori o scrittori, ma tutto verrà gestito artificialmente da questi algoritmi.

Sebbene ChatGPT offra molti vantaggi, ha comunque dei limiti non ancora superati: anche se viene addestrata su una grande quantità di dati, non ha "ancora" la capacità di comprendere il contesto di una conversazione o le sfumature del linguaggio umano.

Inoltre, ChatGPT ha più difficoltà a fornire risposte a domande aperte e su argomenti delicati come la razza, il genere o la politica. Nonostante questo, ChatGPT è il primo esempio di AI accessibile a tutti in grado di comprendere il linguaggio umano e fornire risposte non preconfezionate ma generate da un massiccio apprendimento.

Non rimane che vedere come si evolverà questa tecnologia, sperimentando nel frattempo le attuali potenzialità facendoci scrivere qualche porzione di codice, magari per Arduino o micro:bit.

Boris Landoni

@ boris.landoni@elettronica.in

In copertina:
La potenza e le performance di un ESP32 in un'applicazione d'effetto: un controllo per matrici a LED.



ON LINE

@elettronica.in

SI RINGRAZIA

Futura Elettronica
Microchip
Mouser

Elettronica In

Anno XXIX n. 271
FEBBRAIO / MARZO 2023

EDITORIALE

Direttore Responsabile

Boris Landoni

@ boris.landoni@elettronica.in

Redazione

Stefano Garavaglia, Paolo Gaspari,
Davide Scullino, Gabriele Daghetta

@ redazione@elettronica.in

Grafica

Alessia Sfulcini

@ alessia.sfulcini@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://futuragroup.it/abbonamenti>

Distribuzione per l'Italia

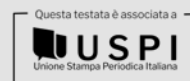
SO.DI.P. Angelo Patuzzi S.p.A.

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.

© 2023 Futura Group srl

Futura
Group
Edizioni



Contenuti

01

RUBRICHE

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

Progetto di Copertina

LED MATRIX CON ESP32

88



AUDIO

Protezione casse acustiche



11

LABORATORIO

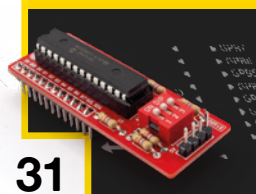
Misuriamo la risposta in frequenza



19

PERIFERICHE

Breakout per MCP23017



31

IN VETRINA

Generatore di funzioni UNI-T



39

11

ARTICOLI

AUDIO

11 Protezione casse acustiche

Unisce la funzione anti-bump, che evita il classico botto in altoparlante all'accensione dell'amplificatore, a quella di distacco in caso di cortocircuito dei finali.

LABORATORIO

19 Misuriamo la risposta in frequenza

Firmware, programma in Matlab e misure eseguibili con il nostro strumento basato su STM32 e DDS AD9833. Seconda e ultima puntata.

PERIFERICHE

31 Breakout per MCP23017

Sperimentiamo con il popolare I/O expander della Microchip, attraverso una scheda applicabile anche su breadboard, interfacciabile con Arduino.

IN VETRINA

39 Generatore di funzioni UNI-T UTG2025A

Abbiamo provato per voi uno strumento professionale che arricchirà il vostro laboratorio: ecco i risultati.

LABORATORIO

55 Breakout ADT7422

Facilita la prototipazione e la sperimentazione con l'integrato sensore di temperatura di Analog Devices, rendendone disponibili i pin su uno strip e provvedendo all'alimentazione.

LABORATORIO
**Breakout
ADT7422**

55
ARDUINO
**Ferma il punto
a LED!**

64
HOME PROJECT
**Speaking
Clock**

75
SICUREZZA
**Espansione per
allarmino ESP32**

99
MISURE
**Cronometro
per atletica**

109
ARDUINO
64 Ferma il punto a LED!

Gioco di abilità dove vince chi riesce a premere il pulsante nel momento in cui il punto luminoso scorrevole formato da un LED rosso passa nella finestrella fra due gialli in un anello NeoPixel.

HOME PROJECT
75 Speaking Clock

L'orologio che non si limita a mostrare l'ora, ma, per renderla più comprensibile a tutti, la pronuncia, associandola agli eventi.

TOP PROJECT
88 LED matrix con ESP32

Pilotiamo una matrice di LED RGB standard tramite bus seriale utilizzando una scheda ESP32, con la possibilità di scegliere tra più versioni.

SICUREZZA
99 Espansione per allarmino ESP32

Nuovo firmware e software di gestione dell'espansione ethernet per Allarmino IoT. Seconda e ultima puntata.

MISURE
109 Cronometro per atletica

Sistema di cronometraggio automatico adatto alla misura di tempi nelle attività di atletica leggera.

AUTOMAZIONE
117 Controllo automatico ventilazione

Basato sull'integrato LM35, accende e spegne l'elettroventola di un sistema di raffreddamento per elettroniche in funzione della temperatura rilevata.

APPLICAZIONI
123 Bleb Invaders

Rievochiamo il fascino dei videogiochi Arcade con la tecnologia di prototipazione rapida BLEB abbinata a uno smartphone o a un tablet.

ARDUINO
129 Contapassi elettronico

Utilizziamo un modulo accelerometro MEMS montato su breakout board, interfacciato tramite I²C-Bus ad una board Arduino per rilevare e contare i passi durante la giornata, un'attività fisica o il lavoro.

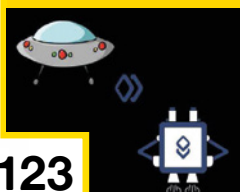
TECNOLOGIA
137 Soluzioni nell'auto elettrica

Scopriamo le tecnologie in uso nella propulsione elettrica e impariamo a distinguere tra le varie tipologie di auto in commercio e del futuro.

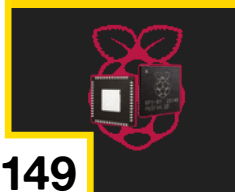
DIDATTICA
149 Programmiamo con RP2040

In questa puntata spieghiamo come usare la nostra scheda FTR2040 come controller per LED NeoPixel.

AUTOMAZIONE
**Controllo automatico
ventilazione**

117
APPLICAZIONI
**Bleb
Invaders**

123
ARDUINO
**Contapassi
elettronico**

129
TECNOLOGIA
**Soluzioni
nell'auto elettrica**

137
DIDATTICA
**Programmiamo
con RP2040**

149

La più ampia selezione di componenti elettronici™

A magazzino e pronti per la spedizione



mouser.it



**MOUSER
ELECTRONICS**

Il comando isolato galvanicamente

Tipicamente il controllo di attivazione e disattivazione delle caldaie e delle pompe di calore si effettua chiudendo, con un contatto pulito, due punti messi a disposizione e chiamati REMOTE CONTROL o in altro modo; tali contatti vengono normalmente chiusi da quello di un termostato ambiente o di un cronotermostato, ma anche da una scheda di telecomando via GSM o Internet o anche WiFi/Bluetooth. Per contatto pulito si intende un interruttore elettrico o altro contatto che non abbia alcun potenziale né connessione galvanica con parti alimentate di qualsivoglia circuito. Tipicamente un termostato ambiente o cronotermostato incorpora un relé o altra forma di interruttore a comando termico che è un contatto pulito, nel senso che è galvanicamente accoppiato solo ai morsetti di uscita; nessun problema, quindi, a fargli controllare una caldaia o una pompa di calore o un condizionatore d'aria.

Esistono però applicazioni in cui per il controllo della climatizzazione e degli apparati che vi provvedono, non è possibile disporre di un contatto pulito o non si ha la certezza che il contatto disponibile sia pulito: per esempio nei sistemi con elettrovalvole di ripartizione dell'impianto; in caso del genere, siccome esistono più termostati e la caldaia o pompa di calore deve accendersi solo quando vi è la certezza che la valvola sia completamente aperta, il contatto pulito del termostato ambiente alimenta il motorino dell'elettrovalvola, la quale è provvista internamente di un contatto che viene chiuso dall'otturatore della valvola stessa quando il motorino (tipicamente tramite una vite senza fine o ingranaggi) lo porta in posizione di "tutto aperto".

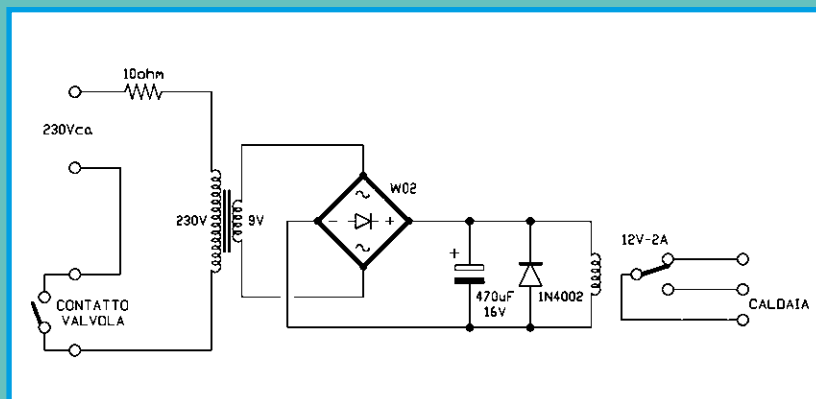
Tips & Tricks sono i nostri suggerimenti in pillole e i piccoli trucchi riguardanti sia l'hardware che il firmware, che sperimentatori, tecnici e progettisti elettronici vorrebbero conoscere per sviluppare meglio la propria passione: quella per l'Elettronica applicata. Ecco quelli che abbiamo preparato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

Talvolta una perdita di isolamento all'interno dell'elettrovalvola può non garantire che il contatto sia pulito, quindi con particolari macchine possono verificarsi problemi all'elettronica di controllo, la quale esige un contatto rigorosamente pulito. In casi del genere e quando non si dispone di un contatto pulito, è possibile interfacciare la caldaia o pompa di calore tramite un circuito di disaccoppiamento costituito come quello proposto nello schema in questa pagina: il contatto chiude il primario di un piccolo trasformatore da 1 VA, alimentato dalla rete elettrica 220Vca, mentre il secondario

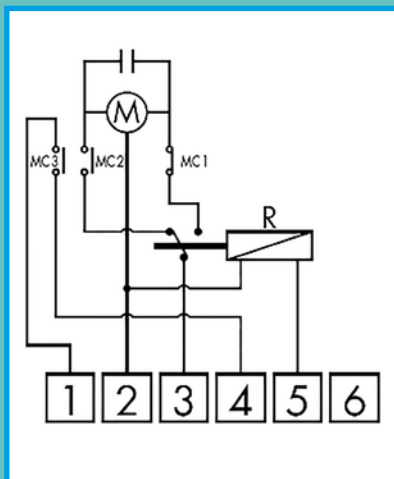
alimenta, tramite un ponte raddrizzatore e un condensatore di livellamento, la bobina di un relé che renderà disponibile un contatto veramente pulito. Al solito, la bobina del relé dispone del suo diodo di spegnimento delle tensioni inverse.

Lo standard RS485

L'interfaccia dati RS485, meglio nota come protocollo TIA/EIA-485, pubblicata dalla Telecommunications Industry Association/Electronic Industries Alliance (TIA/EIA), è una linea di comunicazione bilanciata differenziale che



↑ Il relé disaccoppiatore per il comando galvanicamente isolato della caldaia.



↑ Valvola di zona: il contatto di comando della caldaia è MC2.

consente la trasmissione dei dati su lunghe distanze e in ambienti rumorosi come quelli dell'automazione industriale e di fabbrica. La segnalazione differenziale sopprime il rumore di modo comune e il doppino intrecciato consigliato garantisce che la maggior parte delle interferenze ricevute sia di modo comune. Le lunghe distanze di trasmissione aumentano la possibilità di differenze di potenziale di terra, ma l'ampia gamma di modalità comune (CMR) degli standard garantisce che la rete funzioni correttamente, anche in presenza di tensioni di modalità comune piuttosto elevate.

Il trasmettitore (Tx) e il ricevitore (Rx) hanno entrambi pin non invertenti (Y e A) e invertenti (Z e B). I dispositivi half-duplex vengono utilizzati per la comunicazione bidirezionale su un singolo cavo, quindi i corrispondenti terminali Rx e Tx si collegano allo stesso pin dell'integrato transceiver. Le reti che utilizzano due cavi per la comunicazione bidirezionale impiegano dispositivi full-duplex, in cui i terminali Rx e Tx si connettono a pin separati.

Lo standard RS-485 ha tra i suoi più importanti requisiti:

- segnale differenziale con uno stadio RX molto sensibile ($\pm 200\text{mV}$) e una buona tensione di uscita differenziale (VOD) di $\pm 1,5\text{V}$ Tx; questa combinazione garantisce un robusto margine di rumore che tollera l'attenuazione causata da cavi molto lunghi;
- il TX deve avere un pin di abilitazione perché l'RS-485 consente più driver sul bus per una vera trasmissione bidirezionale su un singolo cavo, quindi

ogni TX deve avere l'uscita three-state;

- elevata corrente di uscita Tx per pilotare cavi a doppia terminazione e cavi lunghi; la trasmissione bidirezionale ad alta velocità su un singolo cavo richiede due resistenze terminatrici;

- elevato CMR da almeno -7V a $+12\text{V}$; siccome l'RS-485 supporta reti lunghe fino a 1200 metri, l'elevato CMR compensa le differenze di potenziale di terra che possono verificarsi su lunghe distanze e permette di tollerare tensioni indotte sul bus, tipiche di ambienti rumorosi e consente anche a dispositivi con tensioni di alimentazione diverse di comunicare sullo stesso bus;
- resistenza di ingresso del ricevitore di circa $12\text{k}\Omega$; lo standard consente fino a 32 dispositivi "one unit load" sul bus e il carico di ciascun dispositivo (Tx o Rx) deve essere $\leq 1\text{mA}$ con una polarizzazione di 12V sul bus.

Lo standard RS-485 è molto simile all'RS-422, tuttavia quest'ultimo consente la comunicazione punto-punto tra un TX e un massimo di 10 RX, mentre l'RS485 è un bus seriale; la configurazione TX-multiple RX è chiamata rete multi-drop o broadcast. La limitazione di un driver elimina la necessità di un pin di abilitazione TX, che riduce il requisito di terminazione a un resistore da 120Ω e richiede l'uso di transceiver full-duplex o circuiti integrati RX e TX separati.

Un ricevitore RS-485 standard riconosce qualsiasi tensione differenziale (Pin A – Pin B) $> +200\text{mV}$ come 1 logico e qualsiasi tensione differenziale inferiore a -200mV come 0 logico; tali soglie di ingresso devono essere soddisfatte su un ampio CMR da -7V a $+12\text{V}$. Qualsiasi tensione di ingresso

all'RX compresa tra -200mV e $+200\text{mV}$ è indeterminata. L'ampio delta tra la tensione di uscita differenziale di $\pm 1,5\text{V}$ del driver e la soglia di $\pm 200\text{mV}$ del ricevitore produce una buona immunità al rumore e la capacità di gestire l'attenuazione da cavi lunghi. Lo standard consente 32 carichi unitari (UL) su un bus, dove 1 UL è specificato come un dispositivo che assorbe non più di 1mA con il bus a $+12\text{V}$ rispetto a GND.

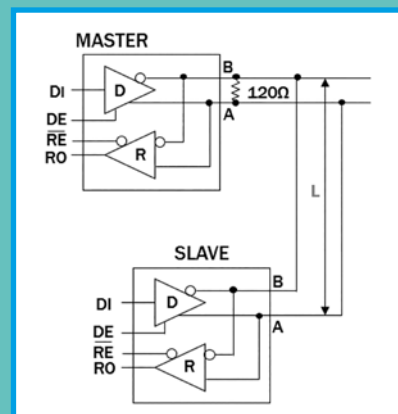
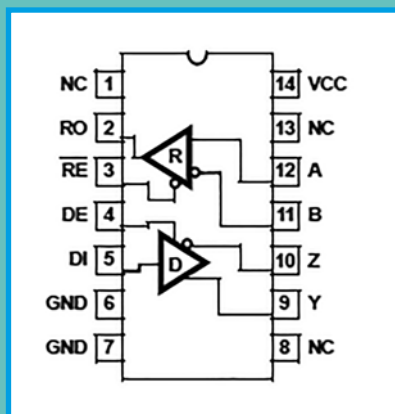
Il trasmettitore RS-485 consta di un driver con uscita differenziale capace di fornire una tensione di almeno $1,5\text{V}$ su un carico differenziale di 54Ω .

Il carico di 54Ω è derivato dal massimo consentito di due resistori di terminazione da 120Ω in parallelo con 32 ricevitori. Le famiglie di integrati RS-485 in genere includono driver con velocità di commutazione selezionabili.

La tipica connessione fra due interfacce RS-485 è mostrata nella figura in fondo a questa pagina, insieme a un tipico integrato transceiver.

Proteggere i componenti dalle ESD

Lavorare con i componenti elettronici, specie quelli a più alta scala di integrazione a tecnologia MOS, implica una serie di accortezze per proteggerli dalle cariche elettrostatiche, perché i danni ai semiconduttori causati dalle scariche elettrostatiche (ESD) sono una delle principali cause di malfunzionamento dei circuiti elettronici. La carica elettrostatica può essere immagazzinata in molte cose, ad esempio indumenti in fibra sintetica, macchinari in movi-



↑ A sinistra, un esempio di transceiver RS485 con enable; a destra, la tipica connessione RS-485.

mento, oggetti con l'aria che vi soffia attraverso, contenitori di plastica, fogli di carta conservati in buste di plastica non antistatica, carta di fotocopiatrici elettrostatiche e persone. Per lavorare con componenti e schede SMD senza rischiare danni da scarica elettrostatica, è importante rispettare le seguenti precauzioni per la protezione ESD; prima di tutto l'area di lavoro deve essere idealmente strutturata come mostra la figura in questa pagina. Il piano di lavoro e la superficie del pavimento devono essere rivestiti con materiale antistatico e le persone al banco di lavoro devono essere collegate a terra tramite un braccialetto antistatico e una catenella metallica con in serie un resistore. Tutte le apparecchiature alimentate dalla rete devono essere provviste di dispersione verso terra, come pure i loro involucri. L'umidità relativa deve essere mantenuta tra il 40% e il 50% e potrebbe essere utile uno ionizzatore negativo per neutralizzare la carica elettrostatica degli oggetti.

Sono da evitare buste e vassoi di plastica sul banco di lavoro e l'imballaggio per i dispositivi elettronici deve essere realizzato con materiali antistatici/conduktiviti.

I componenti elettronici devono essere conservati nel loro imballo originale durante lo stoccaggio. Se un contenitore è parzialmente disimballato, il disimballaggio deve essere eseguito su una postazione di lavoro protetta.

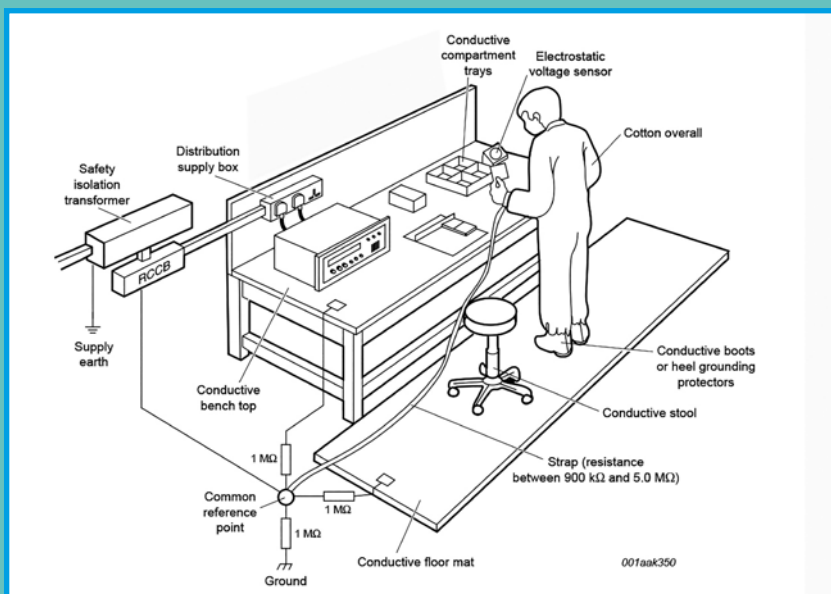


↑ Termometro a energy-harvesting.

Qualsiasi componente elettronico da tenere a disposizione in laboratorio deve essere riposto in buste o scatole antistatiche, ovvero infilato in spugne elettrostatiche (quelle nere). Tutti gli strumenti utilizzati durante il montaggio, compresi gli strumenti di saldatura e i bagni di saldatura, devono essere messi a terra; inoltre gli utensili manuali devono essere di materiale conduttivo o antistatico. Chi volesse approfondire l'argomento troverà utili le raccomandazioni contenute nel manuale EN 100015 (CECC 00015) "Protection of Electrostatic Sensitive Devices", che spiega in modo più dettagliato come organizzare un'area di lavoro con protezione ESD.

Energy harvesting e design

Più o meno tutti regoliamo la temperatura dell'acqua della doccia o del lavabo manovrando il miscelatore secondo abitudine o semplicemente a sensazione, però oggi la tecnologia rende disponibili termometri da rubinetto con display LCD che ci dicono con precisione quanto è calda l'acqua che sta uscendo. Considerato che si tratta di dispositivi elettronici e che vanno alimentati dall'elettricità, nonché la necessità di isolare la rete idrica da quella elettrica, sono nati dispositivi autoalimentati, che ricavano l'energia occorrente al funzionamento tramite raffinate soluzioni di energy-harvesting; nello specifico ci riferiamo al termometro per doccia reperibile sul web all'indirizzo <https://futuranet.it/prodotto/termometro-da-rubinetto-con-display-lcd/> che va installato tra il miscelatore e il flessibile della doccia e consente di rilevare istantaneamente, e con precisione, la temperatura dell'acqua erogata. Funziona grazie a un generatore idroelettrico miniatura composto da una turbinetta fissata internamente al tubo, che aziona un motorino in continua che consente di alimentare il dispositivo tramite il flusso dell'acqua. Si spegne automaticamente alla chiusura del rubinetto, allorché l'elettricità viene a mancare, è sicuro e soprattutto ecologico, perché non necessita di pile o batterie, ma trae energia dalla pressione dell'acqua: un esempio di quando tecnologia e tutela dell'ambiente si incontrano.



↑ Ecco come dev'essere attrezzata l'area ideale per la lavorazione di componenti sensibili alle scariche elettrostatiche.

Web Forum e supporto tecnico

Hai un problema con uno dei circuiti pubblicati? Vorresti effettuare una modifica? Accedi al nostro Web Forum dove i nostri tecnici (ma anche gli altri lettori) ti aiuteranno a chiarire qualsiasi dubbio di natura tecnica. Collegati a:

www.elettronica.in.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 **Futuranet.it**

COME ABBONARSI

→ **on-line**

nella pagina "**Riviste e Corsi Gratuiti**"
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 **Futuranet.it**

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

50% DI SCONTO per l'acquisto di un volume
della libreria "Elettronica In"

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

6
NUMERI
A SOLI

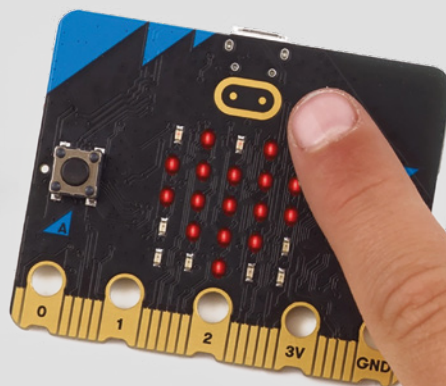
€39

ANZICHÉ € 45,00

Easy to Program

MICRO:BIT 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.

Prezzi IVA inclusa.



cod. MICROBITGOV2

€37,00

UTILIZZALA SUBITO CON IL NUOVO SET!

Set con tutto quello che serve per chi vuole muovere i primi passi nel campo della programmazione con micro:bit. Il set comprende:

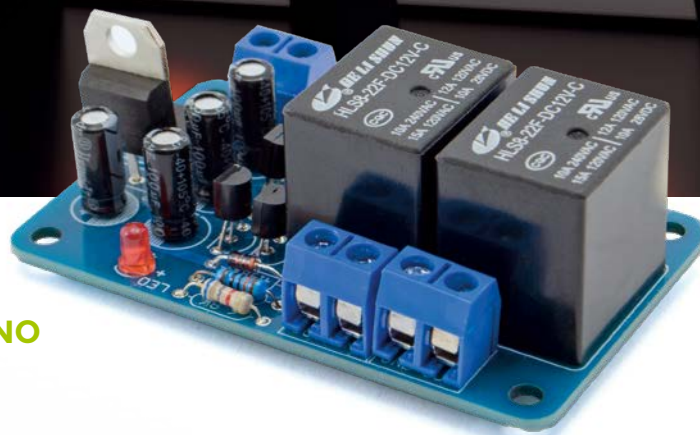
- la scheda micro:bit
- due pile ministilo AAA
- un cavetto USB-micro USB
- un portatile

ROBOT MAQUEEN LITE

Mini robot didattico programmabile basato su micro:bit. Maqueen Lite è compatibile con micro:Bit V1 e V2 e può essere programmato utilizzando la piattaforma di programmazione grafica Mind+ di DFROBOT oppure MakeCode. Il robot viene fornito completamente assemblato! Sarà necessario aggiungere solo le batterie e la scheda micro:bit (acquistabili separatamente) e programmarlo con uno dei molti esempi presenti nella Wiki page. Dispone di connettore per l'inserimento della scheda micro:bit, 2 motori in miniatura con ingranaggi in metallo, 2 ruote in ABS con pneumatico in gomma, 2 sensori ad infrarossi, un buzzer, un ricevitore a infrarossi, 2 luci a LED, 4 LED RGB, un sensore di distanza ad ultrasuoni, un portabatteria e una mappa per l'inseguimento della linea.



PROTEZIONE CASSE ACUSTICHE

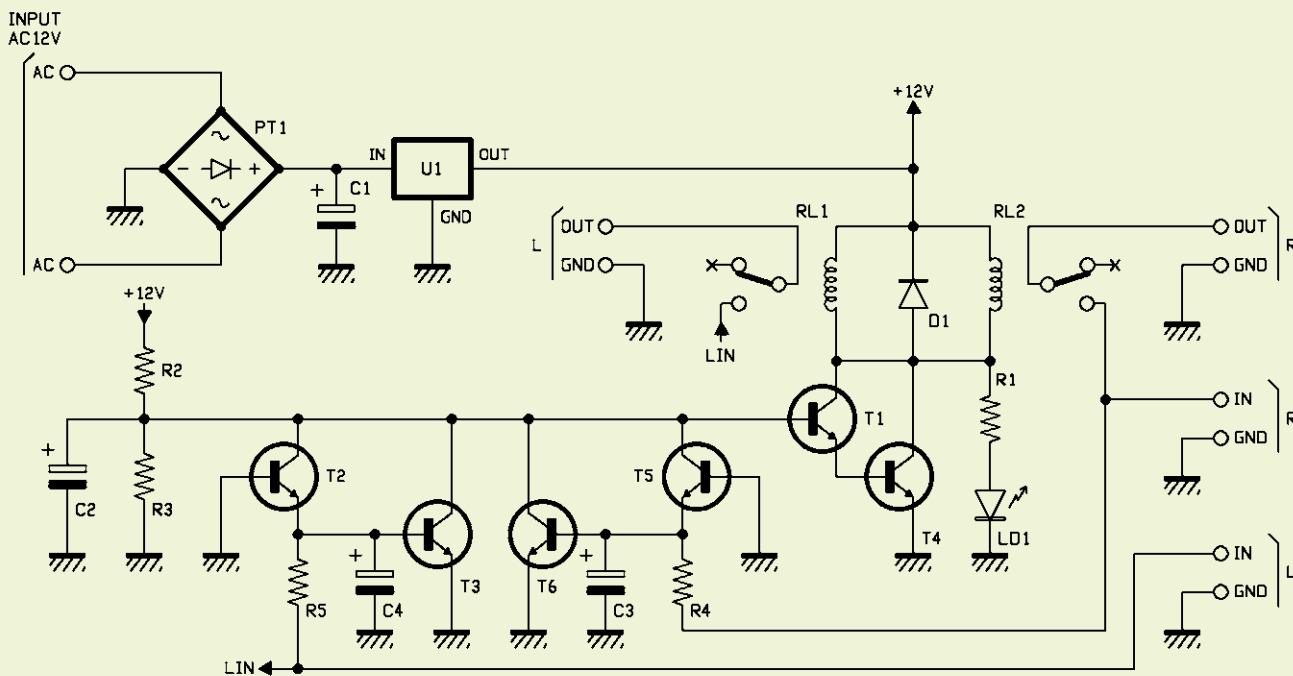


di **DAVIDE SCULLINO**

Unisce la funzione anti-bump, che evita il classico botto in altoparlante all'accensione dell'amplificatore, a quella di distacco in caso di cortocircuito dei finali.

C

hi ama l'autocostruzione dei componenti hi-fi ed in particolare degli amplificatori audio, sa che uno dei problemi da affrontare è impedire il fastidioso "botto" negli altoparlanti che si verifica all'accensione a causa dei transitori, negli stadi di uscita, siano essi ad alimentazione singola (quindi con il corposo condensatore d'uscita) sia ad alimentazione duale; il botto è dovuto alla presenza di tensione continua, impulsiva, a volte anche prossima a quella di alimentazione, derivante dai transitori di carica delle capacità (anche quella di bootstrap)



del circuito amplificatore. Non è soltanto fastidioso da sentire, ma può risultare deleterio per gli altoparlanti, in special modo i woofer delle casse acustiche, quando non siano in grado di sopportare grandi potenze istantanee. Ecco perché sono nati quei circuiti chiamati anti-bump, che in breve mantengono scollegato l'altoparlante mediante un relé fin quando non si ritiene che la tensione all'uscita in assenza di segnale sia nulla, quindi mediamente 1+2 secondi; trascorso questo intervallo di start-up, viene eccitato il relé, il quale collega il carico fino allo spegnimento del circuito, il quale, per avere un funzionamento certo e automatico, deve essere alimentato insieme all'amplificatore.

In queste pagine vi proponiamo un circuito che integra tale funzione, implementando anche un'efficace protezione dalla presenza di componente continua all'uscita che si verifica quando "salta" un finale su uno dei lati di alimentazione: in quel caso la componente continua avrebbe un'ampiezza che difficilmente l'altoparlante potrebbe sopportare e che va a danneggiare il woofer delle casse acustiche, giacché è l'unico altoparlante che viene accoppiato in continua (mid-range e tweeter sono filtrati nel cross-over da almeno un condensatore in serie) e quindi direttamente esposto a tale componente. Il danno è molto più probabile quanto più bassa è la potenza della cassa acustica. Infatti se è vero che di norma il woofer può sopportare la tensione di segnale r.m.s. e quindi quella effica-

ce, è vero altrettanto che in caso di cortocircuito di uno dei rami dello stadio finale si troverebbe sottoposto costantemente a una tensione pari a quella di picco e, cosa che ne aggrava la situazione, a una componente continua, che determina una corrente ben maggiore di quella che scorrerebbe in presenza di segnale, giacché in regime variabile, in serie alla resistenza della bobina mobile si colloca l'impedenza dovuta alla reattanza che nasce dalla natura induttiva della bobina stessa. Ecco quindi che è importante avere un dispositivo in grado di staccare l'altoparlante se viene rilevata all'uscita dell'amplificatore di potenza una componente continua di valore apprezzabile.

Dunque, il nostro circuito svolge due funzioni: appena alimentato, ritarda la connessione del carico che avviene ad opera di un relé per canale, giacché abbiamo voluto realizzare una protezione in versione stereofonica, dopodiché connette l'uscita del finale alle casse; se rileva, all'accensione o durante il funzionamento (per un guasto improvviso che può essere dovuto, ad esempio, a un cortocircuito accidentale dei cavi che portano alle casse...) una componente continua in uscita, manda i relé a riposo e disconnette il carico fin quando tale condizione non viene meno.

SCHEMA ELETTRICO

Vediamo dunque come viene implementato tale funzionamento dal nostro circuito, attraverso

l'analisi dello schema elettrico pubblicato in queste pagine, dalla quale emerge la semplicità e tradizionalità del progetto, costituito da componenti discreti: l'unico integrato è il regolatore di tensione U1, che insieme al ponte raddrizzatore PT1 forma il blocco di alimentazione. Partiamo quindi da questo stadio, che ha come ingresso i morsetti AC-AC (INPUT AC12V) e che idealmente riceverà l'alimentazione da un proprio trasformatore o da uno dei secondari del trasformatore dell'amplificatore di potenza, se è previsto che ne abbia più di uno. Sebbene sia chiamato INPUT AC12V, questo ingresso può essere alimentato a tensione continua, con il vantaggio che l'alimentazione si può dare senza riguardo per la polarità, in quanto la caratteristica del ponte di Graetz (ponte raddrizzatore) è di restituire tra i suoi poli + e - una differenza di potenziale che ha polarità costante a prescindere dal verso della tensione d'ingresso. Analizziamo un caso alla volta, partendo dall'alimentazione in alternata, con tensione da 12Vca a 24Vca: quando la sinusoide fornita dal secondario del trasformatore di alimentazione presenta la semionda positiva sul contatto AC che nello schema elettrico vedete in alto, conduce il diodo che nel ponte è collegato tra ~ e +, la corrente percorre l'elettrolitico C1 caricandolo, quindi dalla massa torna verso il contatto - tramite un diodo che ha l'anodo sul - e il catodo sul ~ dal lato inferiore del PT1, per uscire da quest'ultimo terminale; gli altri due diodi all'interno del PT1 rimangono interdetti. Nella semionda opposta, la corrente entra dal ~ inferiore del PT1 e percorre il

diodo interno che ha l'anodo su di esso e il catodo sul terminale +, percorre nuovamente C1 fornendo ulteriore corrente per la carica ed esce da massa attraverso il - del ponte raddrizzatore, attraverso il suo diodo che internamente ha il catodo sul ~ del lato superiore. Per effetto di ciò, l'elettrolitico C1 si carica ad una tensione pari a quella di picco dell'alternata, diminuita della caduta sulla coppia di diodi che conduce, semionda per semionda: ipotizzando di collegare ai contatti AC il secondario di un trasformatore da 12Vca, C1 si caricherà a:

$$V = 12 \times 1,4142 - 2 \times 0,6V = 15,77 V_{cc}$$

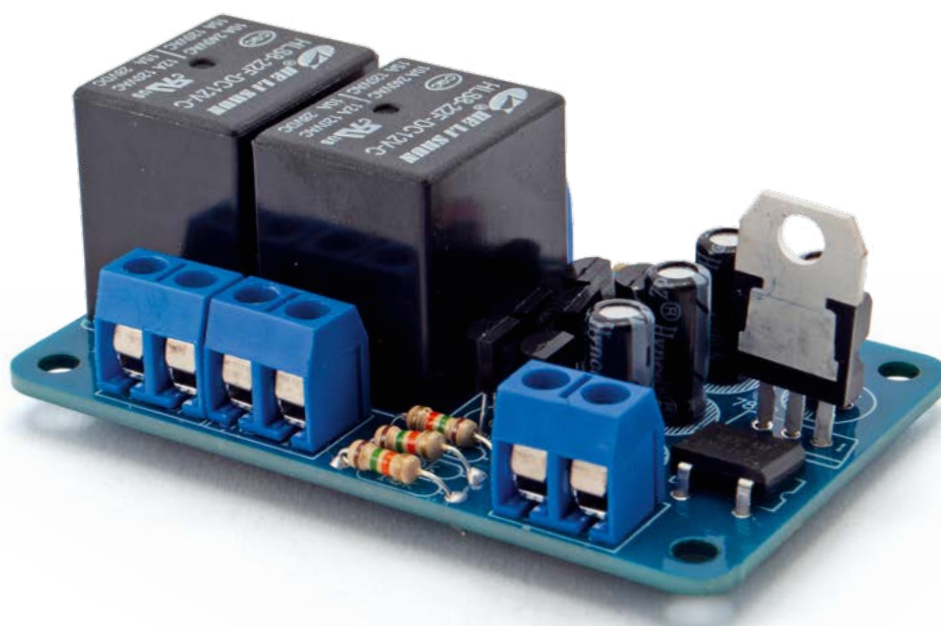
Nel funzionamento in corrente continua, invece, si parte da una tensione che non va raddrizzata: se si applica il polo positivo sul contatto AC in alto nello schema, conducono i diodi del PT1 che condurrebbero nel funzionamento in alternata in semionda positiva, come già spiegato; se il positivo è applicato all'AC in basso, invece, si verifica il funzionamento descritto per la semionda negativa del funzionamento in alternata.

Anche in corrente continua conduce sempre una coppia di diodi alla volta, pertanto il condensatore elettrolitico C1 verrà caricato a una tensione pari a quella presente all'ingresso del ponte, sempre diminuita di 1,2V.

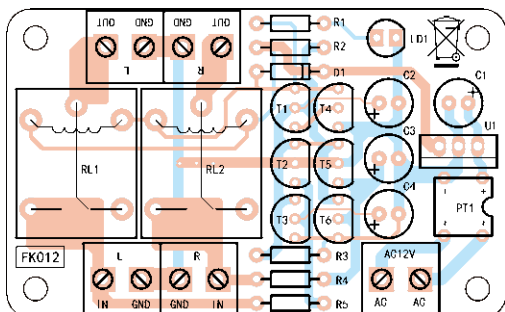
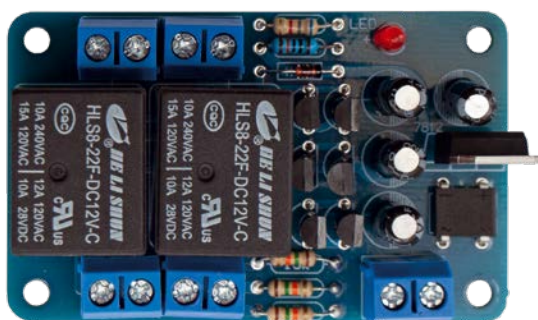
Come vedremo tra breve, per alimentare il nostro circuito è possibile utilizzare la stessa fonte di alimentazione dell'amplificatore ad alcune condizioni. Procediamo con lo stadio di alimentazione e ve-

CARATTERISTICHE TECNICHE

- ◆ **Tensione di alimentazione**
ca 12V
- ◆ **Tensione di alimentazione**
cc 15
- ◆ **Corrente assorbita**
80 mA
- ◆ **Potenza commutabile su 4 ohm:**
400W
- ◆ **Potenza commutabile su 8 ohm:**
800W
- ◆ **Massima tensione d'uscita amplificatore:**
70V
- ◆ **Configurazione amplificatore**
single-ended
- ◆ **Tempo anti-bump:**
2 s
- ◆ **Tensione continua di attivazione:**
±3V



◀ Il prototipo montato con in evidenza le morsettiere di ingresso audio e di alimentazione.



Elenco Componenti:

R1: 6,8 kohm
 R2: 100 kohm
 R3, R4, R5: 15 kohm
 D1: 1N4148
 T1, T2, T3, T4, T5, T6: SS8050 (TO92)
 C1, C2, C3, C4: 100 µF 25 VL elettrolitico
 LD1: LED 3 mm rosso
 U1: 7812

PT1: Ponte diodi DB107
 RL1, RL2: Relé 12V HLS8-22F-DC12V-C

Varie:
 - Morsetto 2 vie passo 5.08 mm (5 pz.)
 - Circuito stampato FK012 (66x41 mm)

diamo che all'uscita del ponte a diodi PT1 troviamo, oltre al condensatore elettrolitico, un regolatore integrato della serie 78xx, che nello specifico è un 7812 e viene impiegato nel circuito per stabilizzare a 12V la tensione con cui funziona il resto dei componenti, in modo da ottenere un funzionamento preciso e costante, nello specifico una soglia certa di tensione continua alla quale la protezione interverrà ed un intervallo di connessione degli altoparlanti certo all'accensione dell'amplificatore. Passiamo ora ad analizzare il resto dello schema, che costituisce la protezione vera e propria, la quale è predisposta per amplificatori stereofonici, seppure possa funzionare anche con un amplificatore in mono e quindi con una sola linea collegata. L'ingresso, corrispondente all'uscita di ciascun canale del finale, è localizzato alle morsettiere

IN/GND R ed L: i contatti IN portano alle resistenze rispettivamente R4 ed R5, le quali a loro volta provvedono alla polarizzazione di base dei transistor T6 e T3. Come vedete dallo schema, le due sezioni a transistor, ossia gli stadi facendo capo a T2, T3, T5, T6 sono identici tra loro, quindi per semplicità descriviamo uno solo di essi, fermo restando che quanto piegato vale pari-pari per l'altro. Concentriamoci sulla porzione di destra, che fa capo alla morsettiera IN R: la resistenza R4 forma, insieme al condensatore elettrolitico C3, un filtro passa-basso la cui frequenza di taglio superiore è più bassa della minima frequenza della banda audio, perché lo scopo di tale filtro è fare in modo che in presenza di segnale il transistor NPN T6 sia interdetto perché la tensione tra i suoi terminali di base ed emettitore non supera quella di soglia; infatti la frequenza di taglio (f_{ts}) del filtro R4-C3 è pari a:

$$f_{ts} = 1/(6,28 \times R4 \times C3) = 1/(6,28 \times 15k\Omega \times 0,1mF) = 0,1 \text{ Hz}$$

La presenza di T5 è irrilevante ai fini della frequenza, almeno in presenza di segnale, perché è posto anch'esso sotto al filtro passa-basso. Vedremo tra breve la funzione di tale transistor.

Trascurando per un istante il funzionamento del ritardo all'accensione, vale a dire dell'anti-bump, immaginiamo che il condensatore C2 sia carico e il partitore R2/R3 polarizzi la base del Darlington NPN formato dai transistor T1 e T4, i cui collettori portano a massa la bobina dei relé RL1 e RL2, i cui scambi commutano gli altoparlanti sulle uscite dell'amplificatore: fin quando non c'è tensione continua di valore apprezzabile tra IN R e GND il T6 è interdetto e T5 pure (la protezione ignora la tensione di offset, che solitamente è dell'ordine di poche decine di millivolt in continua); se la tensione di uscita dell'amplificatore diventa continua per un guasto e supera circa 3 volt, tra base ed emettitore di T6 e tra emettitore e base del T5 si viene a trovare una tensione di valore sufficiente alla conduzione (superiore a 0,6V che rappresenta la tensione V_{be} di soglia) e la corrente di base diventa di entità sufficiente a determinare nel collettore dei transistor una corrente che consente la saturazione dei transistor, i quali si comportano da interruttori statici e portano a pochi decimi di volt il potenziale ai capi di R3.

In questa situazione il Darlington T1-T4, che per restare in conduzione richiede almeno 1,2 volt, viene forzatamente interdetto e rilascia i due relé, interrompendo il collegamento tra le morsettiere

LO SPEAKER BLUETOOTH CHE SUONA E ILLUMINA

I produttori di altoparlanti e casse acustiche wireless non sanno più che cosa inventarsi e lo dimostra Zealot Audio (www.zealot-audio.com), che con il suo S1 multi-function bluetooth speaker lancia l'ultimo grido in fatto di accessori per lo sport: si tratta di un altoparlante Bluetooth multifunzione, che integra una torcia elettrica a LED bianchi. Quindi il ciclista di turno o amante del tracking può ascoltare la propria musica inviata dallo smartphone, e contare, quando occorre, su una luce elettrica che può sostituire quelle portatili a batteria per la bicicletta o le comuni torce a LED. Disponibile in quattro colori (rosso, grigio, azzurro e verde) questo magnifico altoparlante Bluetooth consente un suono omnidirezionale (secondo il costruttore copre 360 gradi) e caratterizzato da bassi potenti e profondi, si collega supportando il più recente protocollo Bluetooth 5.0, dispone di una



presa USB per erogare corrente ad esempio per ricaricare uno smartphone mentre si è in giro e la sua batteria integrata da 4.000 mAh consente il funzionamento ininterrotto per un tempo massimo di 36 ore. Con i suoi 16 watt di potenza e un altoparlante del diametro di 30 mm con membrana in metallo, può coprire con il proprio suono un'area fino a 200 metri quadrati.

di ingresso e quelle di uscita per le casse acustiche. I transistor T5 e T6 svolgono la medesima funzione, risultando sensibili ognuno su una polarità in ingresso, tuttavia, essendo entrambi NPN, sono stati collegati in maniera differente proprio per fare sì che T5 vada in saturazione con la polarità negativa e T6 con quella positiva, restando entrambi interdetti in presenza di segnale entro la banda delle audiofrequenze.

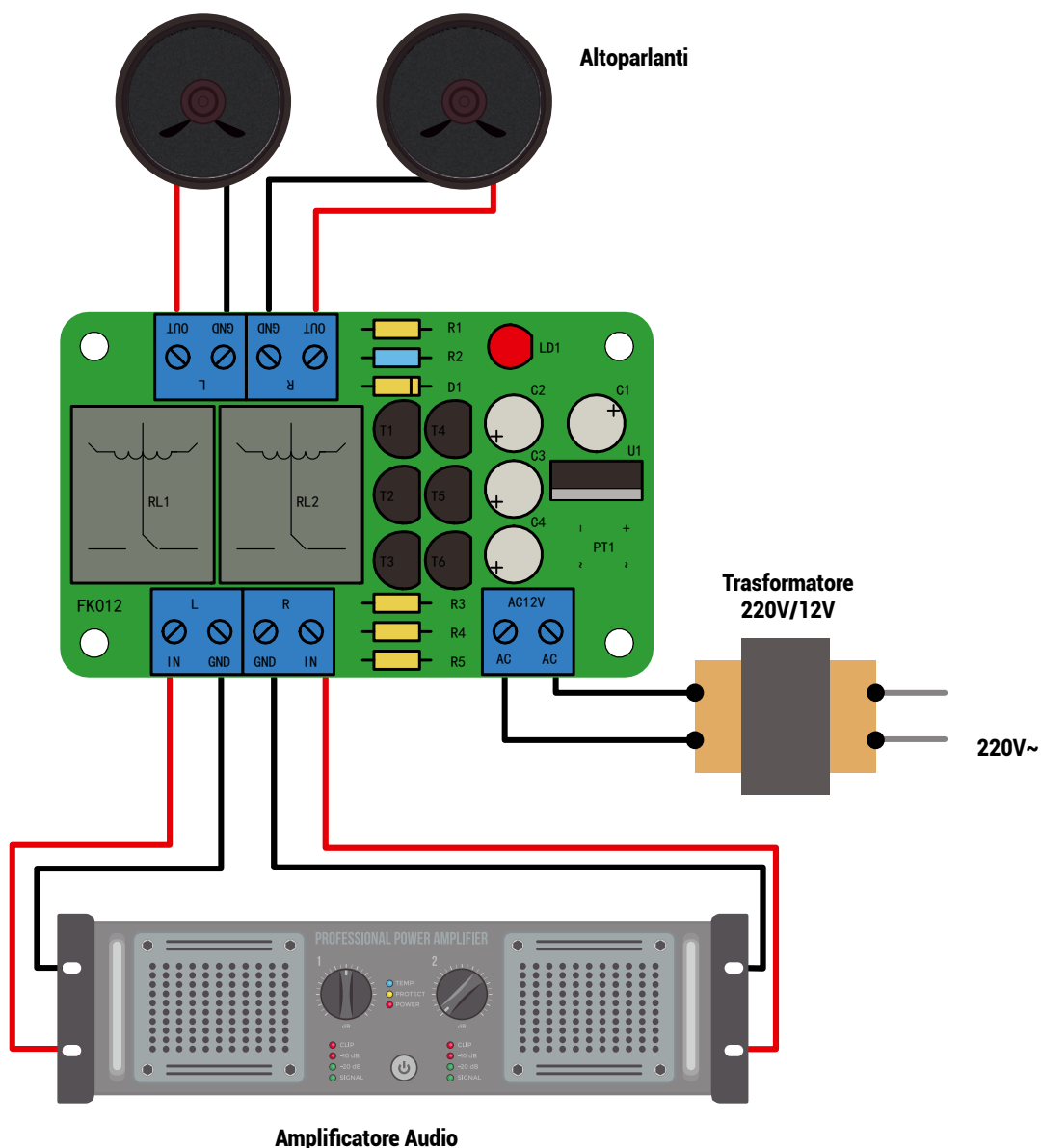
Più nel dettaglio, se tra IN R e GND si presenta una tensione continua di valore superiore a 3V, positiva su IN, T6 va in conduzione e blocca T1 e T4, mentre T5 risulta interdetto perché avendo la base a massa, è polarizzato inversamente tra base ed emettitore; invece nel caso opposto, vale a dire se l'amplificatore porta in uscita una tensione negativa rispetto a GND, la giunzione base-emettitore del T6 viene polarizzata inversamente e T6 non risulta interdetto, però stavolta conduce T5, giacché ha l'emettitore negativo rispetto alla base. Va osservato che i due transistor operano con tensioni di uscita dell'amplificatore che possono raggiungere decine di volt, perché si proteggono a vicenda quando sono interdetti, in quanto la loro connessione, uno opposto all'altro rispetto a massa, fanno sì che tra base ed emettitore vi siano sempre non più di 0,6V positivi o negativi. Questo

perché in presenza di tensione continua positiva all'ingresso R il T6, grazie alla caduta sulla resistenza R4, limita a 0,6+0,7V positivi la componente sul C3, mentre in presenza di tensione negativa ai capi del predetto elettrolitico si trovano 0,6+0,7V negativi che comunque non creano alcun problema. Per come è strutturato il circuito, dato che i due relé vengono comandati da un unico Darlington, operando con un amplificatore stereo è sufficiente l'intervento della protezione su un canale per sconnettere gli altoparlanti di entrambi.

Il funzionamento sinora descritto vale a regime, ossia una volta che l'anti-bump ha esaurito la propria temporizzazione; vediamo quindi di spiegare come opera il ritardo all'accensione, immaginando di alimentare il circuito e di avere tutti i condensatori scarichi: quando tra OUT e GND del regolatore U1 si trovano i 12 volt, l'elettrolitico C2 inizia a caricarsi attraverso la resistenza R2, puntando a un valore di tensione determinato dalla formula del partitore resistivo, che in questo caso è:

$$VC2 = (12V \times R) / (R2 + R3)$$

Con gli attuali valori dei componenti, C2 tenderà a caricarsi a circa 1,56V ma alla fine, si assesterà su una tensione corrispondente alla somma delle



V_{be} di T1 e T4 necessarie a mantenere la corrente di collettore dei due transistor a un valore tale da alimentare le bobine dei due relé RL1 e RL2 (poco meno di 1,4V).

Il tempo necessario affinché la differenza di potenziale ai capi del C2 raggiunga un valore tale da mandare in saturazione il Darlington è pari a circa 1,5÷2 secondi a seconda delle tolleranze dei componenti (gli elettrolitici hanno anche il ±20%, quindi può variare sensibilmente). Trascorso questo intervallo il circuito si porta a regime, i relé vengono eccitati e le morsettiere IN vengono collegate alle corrispondenti OUT, almeno i loro contatti IN/OUT.

Ai capi delle due bobine, che risultano in parallelo, abbiamo previsto il solito diodo di protezione della giunzione di collettore dei transistor T1 e T4, che all'interdizione, a causa della natura induttiva delle bobine stesse, sarebbero esposte a un picco di tensione inversa tale da danneggiarle. Il diodo conduce solo quando la tensione è positiva sul suo anodo e pertanto sopprime questo picco, risultando ininfluente nel normale funzionamento. Osservate, infine, il LED inserito nel circuito tra collettore ed emettitore del T4: questo è acceso per effetto della corrente di fuga nelle bobine dei relé quando il Darlington è interdetto, pertanto

segnala, all'accensione del circuito quando le casse sono scollegate e in caso di guasto che comporti componente continua alle uscite dell'amplificatore, l'intervento della protezione.

Quando il Darlington è in saturazione e il suo collettore alimenta le bobine, non vi è tensione sufficiente a polarizzare direttamente il LED, il quale perciò rimane spento. Dunque, il diodo LD1 si accende in caso di intervento della protezione a seguito della presenza di componente continua all'uscita e nella fase di start-up dell'amplificatore, quando opera la funzione di anti-bump.

REALIZZAZIONE PRATICA

Bene, con questo abbiamo terminato l'analisi circuitale e possiamo concentrarci sulla costruzione della protezione, il cui primo passo consiste nel preparare lo stampato, che è a doppia faccia; consigliamo di ricorrere alla fotoincisione, scaricando dalla sezione Download del nostro sito www.elettronica.in.it le tracce lato rame necessarie ad impressionare, quindi sviluppare e incidere le due facce.

Una volta inciso e forato il PCB dovete montarvi i componenti, a partire dalle resistenze e dal diodo D1 (da orientare come mostrato nel piano di montaggio visibile in queste pagine); procedete con il ponte raddrizzatore, il LED da 3 mm (da orientare con lo smusso verso il lato del circuito stampato) e i sei transistor, da rivolgere tutti con la parte piatta del contenitore rivolta verso il lato più vicino della basetta. Montate quindi le morsettiere, i condensatori elettrolitici (il loro verso è indicato nel piano di montaggio) e il regolatore di tensione 7812, la cui aletta metallica dovrà stare rivolta verso il ponte a diodi PT1. Terminate il montaggio con i due relé 12V a uno scambio, per il cui orientamento non c'è da preoccuparsi perché nel PCB che abbiamo disegnato entrano solo nel verso corretto.

Dopo aver verificato l'esattezza del montaggio confrontandovi con i disegni in queste pagine ed accertato che non vi siano cortocircuiti tra piazzole dovuti a sbavature dello stagno, potete pensare all'installazione, che idealmente andrebbe fatta all'interno dell'amplificatore. In questo caso dovete considerare che è possibile sfruttare l'alimentazione dello stesso (o il ramo positivo, se il finale è ad alimentazione duale) e quindi la continua, collegandovi i morsetti AC senza riguardo per la polarità; ricordate che il circuito tollera all'ingresso una tensione continua massima di 35V.

È esclusa la possibilità, almeno per gli amplificatori alimentati a tensione duale, di collegare l'INPUT

AC 12V a un secondario del trasformatore, perché in tal caso una porzione del ponte raddrizzatore verrebbe a creare un cortocircuito in una delle semionde; a quel punto è meglio prevedere un piccolo trasformatore dedicato da 220/12V, 2VA, che deve comunque essere alimentato insieme a quello di alimentazione dell'amplificatore, perché l'anti-bump deve partire quando l'amplificatore viene messo sotto tensione: né prima, né dopo. Il suddetto trasformatore dedicato è la fonte di alimentazione che dovrete utilizzare se non potete inserire il circuito nello stesso contenitore dell'amplificatore: in questo caso prevedete un box adeguato e ricordate di mettere in serie al primario del trasformatore un interruttore, da portare in posizione ON dopo aver acceso l'amplificatore. Per il cablaggio, connettete le uscite del finale, rispettando la polarità indicata, alle morsettiere IN/GND R ed L, stando attenti che non sfugga qualche filo a creare cortocircuito (le morsettiere sono a passo 5 mm e i morsetti abbastanza vicini) quindi collegate i cavi che andrebbero alle casse acustiche alle morsettiere di uscita OUT/GND R ed L, sempre rispettando la polarità indicata. Un suggerimento è proposto dallo Schema di cablaggio che vedete nella pagina accanto, dove il circuito stampato della protezione viene alimentato con un trasformatore dedicato 220/12Vca.

Rammentiamo che il circuito proposto in queste pagine è adatto solamente ad amplificatori con uscita single-ended, ossia riferita a massa: non è applicabile a finali a ponte, perché in questi ultimi l'altoparlante è sospeso da massa e risulta collegato tra le uscite di due stadi a simmetria complementare o quasi complementare, pertanto se collegaste il "negativo" di uscita a GND dei morsetti mandereste in cortocircuito la sezione relativa al filo negativo (GND). ■



Cosa occorre?

**Il kit di protezione casse acustiche (cod. HR3772) è in vendita a Euro 12,90.
Il prezzo si intende IVA compresa.**

Il materiale va richiesto a:

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



Microchip is...

Aerospace & Defense

Power Management < ASICs <
Frequency and Timing < Discretes <
Synchronization Systems < FPGAs <
Electromechanical Relays < Memory <
Communication Interfaces < Mixed-Signal ICs <
RF, Microwave, Millimeter Wave < MCUs and MPUs <



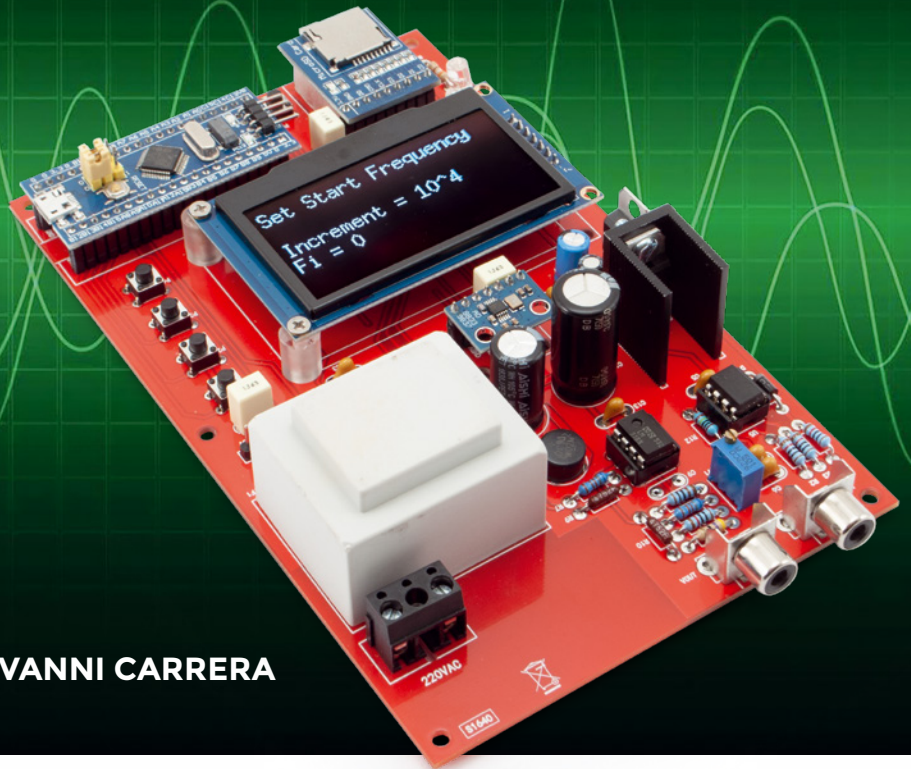
- Cabin Management Systems
- Actuation Systems
- Engine Systems and Controls
- RF and Microwave Systems
- Cockpit Avionics
- Secure Communications
- Radar/Electronic Warfare
- Autonomous Vehicle Systems
- Transparent RF Transponder
- Attitude and Orbit Control
- Digital Communications Payload
- Electronic Power Systems
- Remote Sensing Payload
- Telemetry Tracking and Control



microchip.com/aerospaceanddefense

The Microchip name and logo and the Microchip logo are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. All other trademarks are the property of their registered owners.
© 2022 Microchip Technology Inc. All rights reserved.
MEC2452A-UK-09-22

MISURIAMO LA RISPOSTA IN FREQUENZA

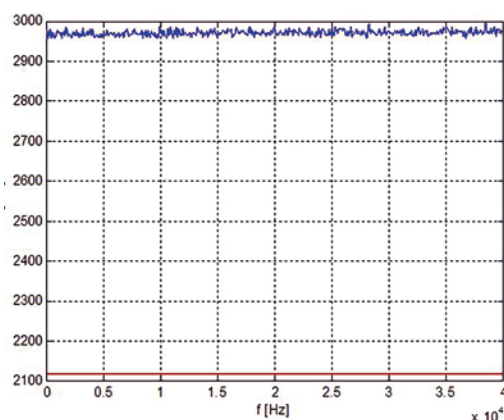


dell'Ing. **GIOVANNI CARRERA**

Firmware, programma in Matlab e misure eseguibili con il nostro strumento basato su STM32 e DDS AD9833. Seconda e ultima puntata.

Misurare la banda passante o risposta in frequenza di un circuito elettronico analogico (amplificatore, filtro attivo o passivo) come anche il comportamento in frequenza di dispositivi e circuiti risonanti, richiede un certo impegno e non è affatto pratico e veloce, utilizzando metodiche tradizionali come l'iniezione di un segnale sinusoidale a varie frequenze e un oscilloscopio per rilevare il risultato in uscita, specie se non è possibile disporre di una memoria dove salvare le singole misure, per i singoli valori di frequenza di test. Abbiamo pensato di offrirvi, per arricchire il vostro laboratorio, una soluzione che consiste in un misuratore della banda passante o risposta in frequenza BF basato su un generatore di segnali a tecnologia DDS, governato da una scheda con

Fig. 1
Circuito LC
parallelo.



microcontrollore STM32 di STMicroelectronics, che rileva le ampiezze all'uscita del circuito sotto test o ai capi del dispositivo accordato, mediante una sezione analogica d'ingresso, che tramite un preciso A/D converter consente di digitalizzare il segnale risultante e memorizzarne in dati corrispondenti su una SD-Card, per esportarli su un Personal Computer dove poterli elaborare grazie ad un programma appositamente scritto in linguaggio Matlab. Questo ci permetterà poi di tracciare dei grafici che descrivono il comportamento del circuito analizzato. Abbiamo concluso la prima puntata dopo aver descritto l'hardware dello strumento, nelle sue parti digitale e analogica, ed ora completiamo la trattazione proponendovi alcuni esempi della risposta in frequenza di circuiti e componenti campione, poi affrontando gli aspetti software, prima con il firmware, sviluppato con Arduino IDE, poi con un programma in Matlab/Octave per l'elaborazione del file generato dal sistema.

ESEMPI DI MISURE DI RIF

Per introdurvi all'utilizzo dello strumento e farvi capire in quale ambito si opera, qui di seguito

saranno presentati e analizzati alcuni esempi di circuiti oscillanti (risonanti, per essere più precisi...) di tipo RLC e le rispettive risposte in frequenza. A riguardo, tenete presente che per brevità, da qui in poi sintetizzeremo il termine "Risposta in Frequenza" con l'acronimo RiF.

Negli esempi descritti in questo articolo, le misure acquisite con il nostro misuratore sono state elaborate in Matlab, ma non è l'unico software utilizzabile: per esempio si può usare anche un normale foglio di calcolo come Microsoft Excel.

Verifica della banda passante del sistema

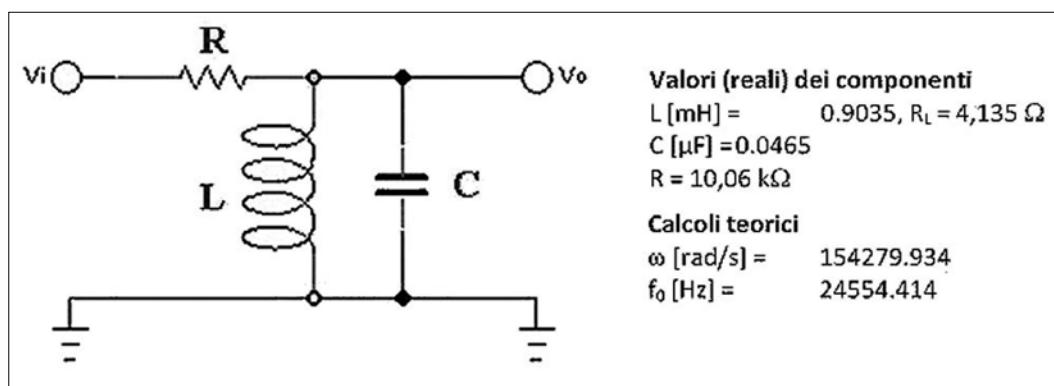
Prima di iniziare a lavorare con il nostro strumento, sarebbe opportuno eseguire una prova preliminare il cui scopo è quantificare qualitativamente e quantitativamente la linearità della banda passante. Allo scopo, bisogna cortocircuitare l'ingresso con l'uscita dello strumento, cosicché il segnale generato verrà letto come se provenisse da un circuito esterno o da un circuito sotto-test. Se tutto va bene, elaborando il file della misura dovremmo avere una banda più o meno piatta, come quella proposta dal grafico che vedete nella **Fig. 1**; la prova corrispondente l'abbiamo fatta sul prototipo, impostando un guadagno dello stadio finale pari a $G = 100\text{k}\Omega / 10\text{k}\Omega = 10$. Il valore medio della RiF è stato di 2.969,8 mV, con una deviazione standard di 6,39 mV, pari allo 0,22 %.

Questa misura diventa necessaria se inseriamo un partitore per adattare i livelli del segnale in ingresso o in uscita.

Circuito RLC parallelo

Partiamo con la prima misura, che è quella su un circuito antirisonante del tipo proposto nella **Fig. 2**, collocando il resistore R in serie ad esso; nella misura è stata trascurata la resistenza elettrica del filo che compone l'induttore.

Fig. 2
Circuito LC
parallelo.




```

Elaborazioni misure di risposta in frequenza
Nome file elaborato   = RIF LC.txt
Start frequency = 10000
Stop frequency = 40000
Delta frequency = 10
Number of data = 3001
Max peak measurements = 1000
Min peak measurements = 1000
Frequenza del picco max = 24880 Hz
Frequenza del picco min = 10090 Hz
Valore a -3dB [mV]    = 418.61
Banda passante       = 7050 [Hz]
Fattore di merito     = 3.53

```

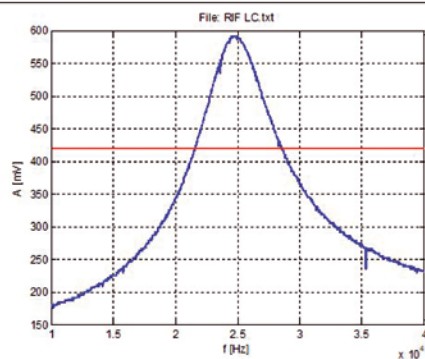


Fig.3
Risposta in
frequenza di
un circuito
risonante LC
parallelo.

I valori dei componenti sono stati misurati con un ponte RLC e la frequenza di risonanza è teorica e non tiene conto della presenza della resistenza R né di quella dell'induttore.

Nella **Fig. 3** sono mostrati i risultati del programma *plotRIF.m* scritto in linguaggio Matlab. La linea rossa è quella corrispondente a -3 dB rispetto al valore di picco. Le sei righe scritte in inglese sono quelle presenti nel file e riportate per conoscenza dal programma di elaborazione. Sono calcolati anche i valori d'intersezione, iniziale e finale, corrispondenti a -3 dB, l'eventuale banda passante e fattore di merito.

Se il fattore di merito è basso, la campana è larga e il picco è tondeggiante, per cui risulta più difficile trovare il massimo. Questo spiega la piccola discordanza (+1.33%) tra il risultato teorico e quello sperimentale.

IL FATTORE DI MERITO

Il fattore di merito o Q (*Quality factor*) è un parametro adimensionale di smorzamento di un'oscillazione, che riassume due parametri fondamentali dell'oscillazione:

- smorzamento;
- frequenza naturale.

Generalmente Q è definito come:

$$Q = 2\pi \times \frac{\text{Energia immagazzinata}}{\text{Energia dissipata per ciclo}} = \omega_0 \times \frac{\text{Energia immagazzinata}}{\text{Dissipazione di potenza}}$$

dove ω_0 è la pulsazione corrispondente alla frequenza naturale del sistema, ed energia immagazzinata, dissipata e dissipazione di potenza sono caratteristiche del sistema analizzato. Un Q alto indica che le oscillazioni si smorzano più lentamente. Per esempio, un pendolo sospeso in aria, avrebbe un alto Q , mentre un pendolo immerso in olio ne avrebbe uno basso. In certi casi, come per le sospensioni di un veicolo, si cerca di

abbassare Q con gli ammortizzatori per evitare dondamenti eccessivi su strade sconnesse. Lo stesso criterio si applica anche alle bussole, immerse in olio per aumentarne lo smorzamento. Risuonatori con un alto fattore di merito, pilotati con una sinusoide, rispondono con grandi ampiezze in corrispondenza della loro frequenza di risonanza e la loro banda è molto stretta. Ad esempio, un circuito di sintonizzazione con alto Q in un ricevitore radio sarebbe più difficile da regolare, ma permetterebbe di distinguere meglio i segnali provenienti da stazioni con portanti vicine tra loro. Un cristallo di quarzo ha un fattore di merito estremamente alto rispetto a quello di un risuonatore ceramico e ha una deriva termica molto contenuta.

Il concetto di fattore di merito nasce in elettronica, come una misura della "qualità" desiderata in un circuito accordato o in un altro tipo di risonatore. Un'altra interpretazione del fattore di merito, è data dalla formula seguente:

$$Q = \frac{f_0}{Bw}$$

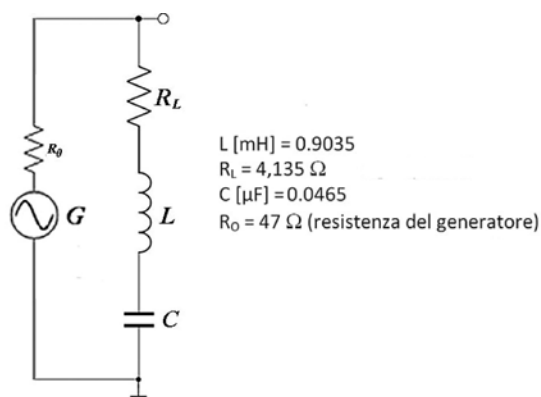
Dove f_0 è la frequenza di picco e Bw è la larghezza di banda, vale a dire l'ambito di frequenze alle l'ampiezza è contenuta in -3dB.

Questa formula è stata utilizzata nei programmi Matlab, sviluppati per elaborare le misure RiF.

MISURA SU CIRCUITO RCL SERIE

Andiamo ora a un'altra misura, che riguarda ancora un circuito RLC, però di tipo serie: in **Fig. 4** si vedono il circuito RLC e i valori dei componenti usati per la nostra misura campione, che sono uguali al circuito parallelo proposto nella **Fig.2**, con la differenza che in questo caso abbiamo voluto ridurre la resistenza serie, che nello specifico è quella parassita in uscita al generatore (R_0). Anche qui si è trascurata la resistenza della bobina R_L . In **Fig. 5** è riportata la RiF del circuito LC serie mi-

Fig. 4
Circuito risonante
serie.



surata con il nostro strumento e plottata. Mettendo un resistore da 1k in serie avevamo un segnale troppo basso, quindi abbiamo deciso di collegare la serie RLC direttamente all'uscita del generatore (con $R_0 = 47 \Omega$). Questo giustifica qualche differenza rispetto al circuito LC parallelo.

Si ricorda che un circuito LC serie, in corrispondenza della frequenza di risonanza, diventa un cortocircuito: la tensione del condensatore è uguale e opposta a quella dell'induttore, per cui la tensione ai capi del bipolo è molto bassa.

Per plottare la retta rossa a +3dB rispetto al minimo, occorre settare il programma Matlab per picco negativo.

RISONANZA DI UN QUARZO

Come ulteriore esempio di analisi della risposta in frequenza abbiamo voluto proporvi il test effettuato su un componente risonante elettromeccanico molto utilizzato in elettronica, implementando la misura della RiF del circuito formato da un comune quarzo per orologi posto in serie a un resistore da 10 k Ω .

Nella **Fig. 6** potete vedere il particolare della RiF in corrispondenza della zona del picco di segnale. Il cristallo si comporta in modo simile a un circuito LC risonante in serie, per cui, alla risonanza si ha un minimo.

La frequenza di risonanza, in questo caso, è uguale a quella attesa.

Il picco massimo è quello di risonanza parallelo ed ha una frequenza maggiore di 5 Hz rispetto al minimo; bisogna poi considerare l'effetto delle spurie.

RISPOSTA DI UN FILTRO ATTIVO PASSA-BASSO

Abbiamo anche eseguito una prova di risposta in frequenza su un filtro passa-basso di tipo attivo (con operazionale, vista la semplicità di dimensionamento). Nello specifico si è optato per il filtro Sallen-Key a due poli, con allineamento di tipo Butterworth, la cui configurazione circuitale è illustrata nella **Fig. 7**.

Fig. 5
Risposta in
frequenza di un
circuito risonante
LC serie.

Elaborazioni misure di risposta in frequenza
 Nome file elaborato = RIF LCser.txt
 Start frequency = 10000
 Stop frequency = 40000
 Delta frequency = 10
 Number of data = 3001
 Max peak measurements = 1000
 Min peak measurements = 1000
 Frequenza del picco max = 10470 Hz
 Frequenza del picco min = 24130 Hz
 Valore a -3dB [mV] = 518.96
 Banda passante = 5340 [Hz]
 Fattore di merito = 1.96

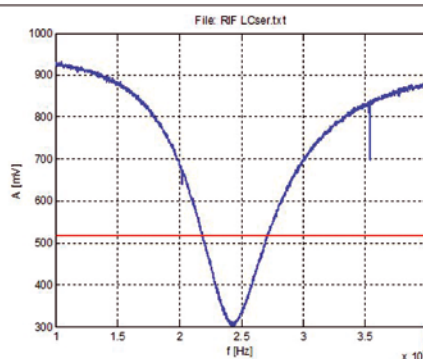
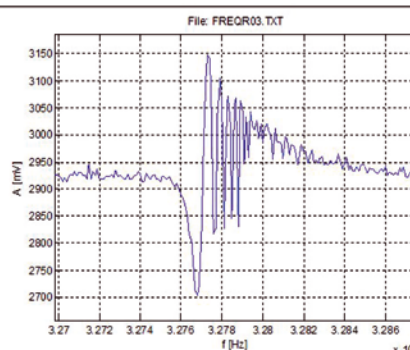


Fig. 6
Misura della RiF
di un quarzo.

Elaborazioni misure di risposta in frequenza
 Nome file elaborato = FREQR03.TXT
 Start frequency = 30000
 Stop frequency = 34095
 Delta frequency = 1
 Step period [us] = 20436.000
 Number of data = 4096
 Peak measurements = 2000 quarzo a 32k, R=10k
 Frequenza del picco max = 32773 Hz
 Frequenza del picco min = 32768 Hz



Nel nostro caso i valori dei componenti sono:

- $R1=R2 = 6,81 \text{ k}\Omega$
- $C2 = 470 \text{ nF}$
- $C1 = 2 \cdot C2 = 940 \text{ nF}$ (2 in parallelo) con una frequenza di taglio teorica di 35,16 Hz.

La frequenza di taglio teorica per questa configurazione, vale:

$$f = \sqrt{2} / (4 \cdot \pi \cdot R1 \cdot C2)$$

I filtri passa-basso sono indispensabili per evitare il fenomeno di *aliasing* del campionamento, per cui occorre che il segnale da campionare sia a banda limitata, ossia che non abbia energia oltre una certa frequenza. Nella realtà, i segnali non sono mai a banda limitata per l'inevitabile presenza di rumore. In **Fig. 8** si vedono i risultati dell'elaborazione eseguita sul segnale rilevato dal nostro strumento. Il grafico della RiF è più facilmente interpretabile se si rappresenta in decibel, con scala bi-logaritmica, ossia come il diagramma di Bode visibile in **Fig. 9**. Per fare questo grafico si è adimensionata la RiF, fermando l'elaborazione ad un certo punto (300) perché erano presenti alcuni zeri e anche valori negativi sul tratto finale, come si vede sempre in **Fig. 9**. Questo perché il segnale era di ampiezza piuttosto piccola, raggiungendo lo zero e anche valori negativi, per effetto dell'offset. Passando ai decibel, ossia $20 \cdot \log(A)$, gli zeri dell'ampiezza portano a $-\infty$ la curva, mentre i numeri negativi invertono la curva e ne risultano valori complessi. La RiF del filtro è stata molto vicina ai valori teorici sia come frequenza di taglio sia come pendenza: 2 poli determinano 40 dB di attenuazione per ogni decade di frequenza.

RISONANZA DI UNA PASTIGLIA PIEZOELETTRICA

Tornando ai componenti risonanti elettromeccanici,

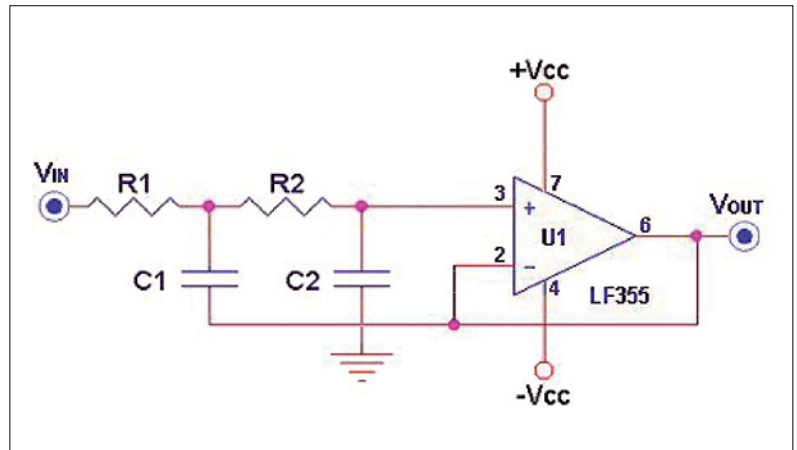


Fig. 7
Filtro passa-basso Sallen-Key a due poli.

abbiamo deciso di analizzare il comportamento nei confronti della frequenza di una pastiglia piezoelettrica; nello specifico abbiamo utilizzato un buzzer a tre terminali, il quale ha un elettrodo in più che serve come feedback. Componenti del genere sono comuni nei cicalini attivi (quelli con elettronica) in quanto basta un transistor e pochi altri componenti per farli oscillare. I risultati della RiF sono visibili in **Fig. 10**, dove si vedono anche i collegamenti. In questo caso il sistema si comporta in modo simile a un circuito LC risonante in parallelo e vi sono due picchi ben pronunciati. Solitamente questi buzzer vengono fatti oscillare (pilotati) intorno ai 3 kHz, frequenza ben percepita dall'orecchio umano. Per questa ragione si deve ripetere la misura centrando il picco d'interesse con risoluzione di 1 Hz. Come si vede in **Fig. 11**, la frequenza del picco più basso rilevata è stata di 3.085 Hz, ma esiste anche un picco più alto a circa 9.880 Hz, con fattore di merito anche migliore.

IL FIRMWARE DELLA SCHEDA STM32

Andiamo adesso ad analizzare e descrivere gli aspetti software del progetto, iniziando con il

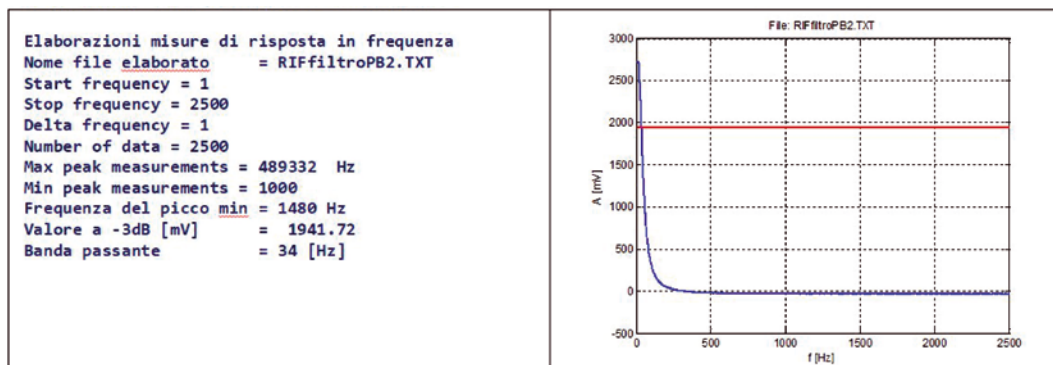


Fig. 8
RiF di filtro passa-basso di ordine due.

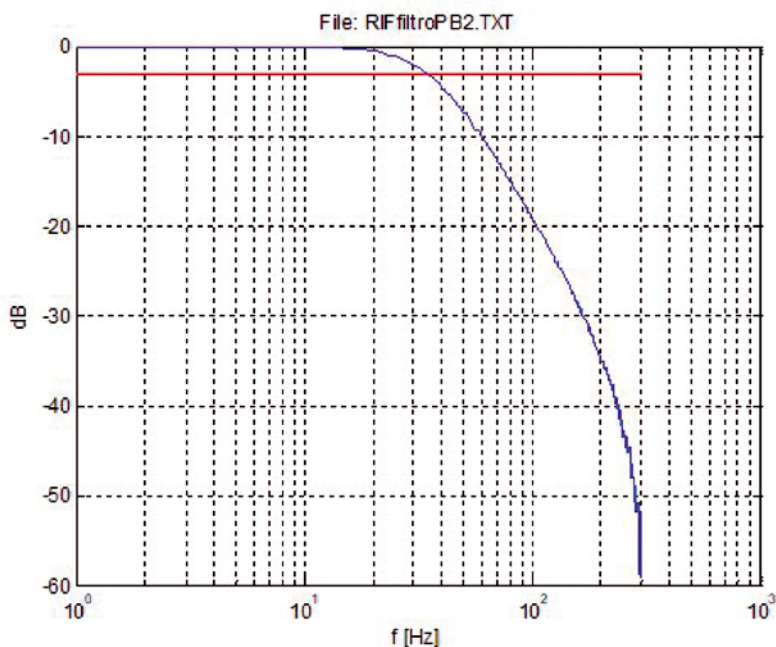


Fig. 9
Diagramma di
Bode del filtro.

firmware o sketch, se preferite, dato che la scheda STM32 si programma come una board Arduino; rendiamo disponibile tale firmware per il download, insieme ai file di questo progetto, sul nostro sito www.elettronica.in.it.

Il firmware si chiama *STM32WaveFRSDV2.ino* e va caricato mediante l'IDE Arduino.

Come accennato nella prima puntata di questo articolo, descrivendo la sezione digitale del progetto, per il controllo del generatore DDS dell'Analog Devices abbiamo deciso di utilizzare la libreria MD_AD9833.h, la quale è in grado di usare sia il bus hardware SPI sia un'interfaccia software con pin di I/O qualsiasi.

Si è usata la versione SPI hardware con la seguen-

te inizializzazione:

```
#define FSYNC PB9 // SPI Load pin number (FSYNC in
AD9833 usage)
MD_AD9833 AD(FSYNC); // Hardware SPI
```

Il programma è tutto nella funzione `setup()` perché non si ripete; ne deriva che per fare una nuova misura va premuto il pulsante di reset. Si inizializza l'I/O e la libreria MD_AD9833 con la funzione:

```
AD.begin();
```

Le varie fasi di esecuzione del programma sono ben illustrate negli *screenshot* di un esempio, mostrati in **Fig. 12**. Il programma, all'avviamento, stampa il messaggio di **Fig. 12-a**, in questo caso non era presente la scheda SD, quindi è stampato il messaggio "No SD present" e la misura sarà solo trasmessa dalla seriale.

All'inizio si devono immettere la frequenza **StartFreq** (**Fig. 12-b**), d'inizio *sweep*, quella finale **StopFreq** (**Fig. 12-c**) e quella di step **Deltaf** (**Fig. 12-d**). Il programma verifica se questi parametri sono compresi nei valori consentiti.

In base ai dati immessi, il programma calcola il numero dei dati della risposta in frequenza, ossia il numero ANdata di elementi del vettore RIF[] d'interi a 16 bit, il cui massimo è 4.096.

A questo proposito, va ricordato che, per le MCU a 32 bit, gli interi sono pure a 32 bit e questo potrebbe saturare la RAM, per cui è necessario dichiarare il tipo di variabile come 'int16_t' e non semplicemente 'int'.

In **Fig. 12-c** si vede la schermata che riassume i parametri impostati e la segnalazione che la misu-

Fig. 10
Misura della RiF di
un buzzer piezo.

Elaborazioni misure di risposta in frequenza

Nome file elaborato = RIFbuzzer3bis.TXT

Start frequency = 500

Stop frequency = 20000

Delta frequency = 10

Number of data = 1951

Max peak measurements = 1000

Min peak measurements = 1000

Frequenza del picco max = 9880 Hz

Frequenza del picco min = 17350 Hz

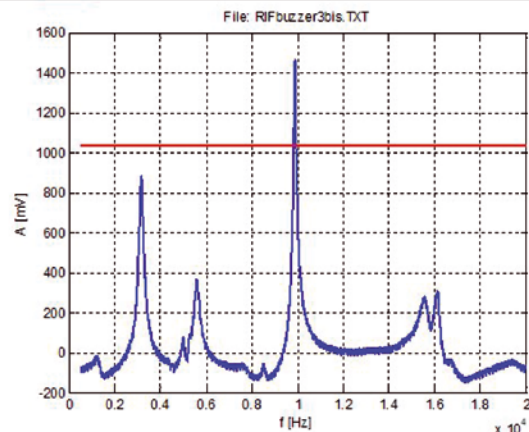
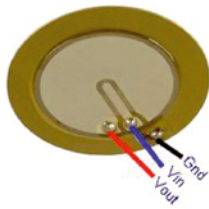
Valore a -3dB [mV] = 1036.62

Valore medio RiF [mV] = 29.81

Dev. standard RiF [mV] = 188.01

Banda passante = 130 [Hz]

Fattore di merito = 76.00



```

Elaborazioni misure di risposta in frequenza
Nome file elaborato      = RIFbuzzer3ter.TXT
Start frequency = 1500
Stop frequency = 5595
Delta frequency = 1
Number of data = 4096
Max peak measurements = 1000
Min peak measurements = 1000
Frequenza del picco max = 3085 Hz
Frequenza del picco min = 1557 Hz
Valore a -3dB [mV]      = 699.33
Valore medio RiF [mV]   = 74.01
Dev. standard RiF [mV]  = 212.37
Banda passante          = 150 [Hz]
Fattore di merito       = 20.57

```

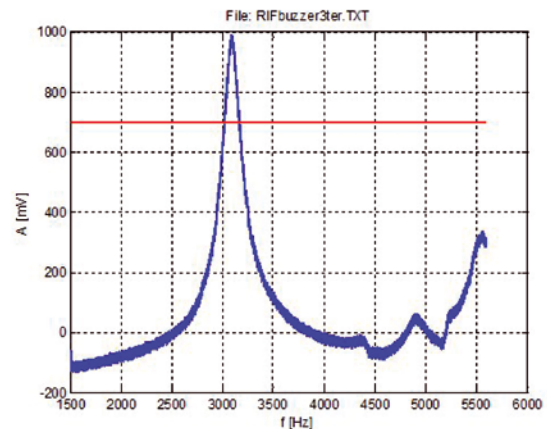


Fig. 11
RiF del primo picco.

ra della risposta in frequenza è iniziata, indicata anche dal LED.

Nel caso in cui sia presente la SD, comparirà una schermata che indica che i dati sono stati salvati. Alla fine della misura (**Fig. 12-f**) sono calcolati il massimo e il minimo, con le relative frequenze. Poi il programma aspetta che si prema il pulsante 'S' per trasmettere i dati sulla seriale a 115.200 baud. Le immissioni dei parametri sono fatte con quattro tasti che servono anche per altre funzioni. I tasti (+) e (-) incrementano o decrementano di un numero il cui esponente è aumentato dal tasto (M) e il tasto (S) salva il dato ed esce dal loop d'immissione.

La funzione **Menu(String prompt,String fvar, byte N)**, serve per immettere le tre frequenze, il **prompt** è stampato sulla prima riga del display, la **fvar** stampa il nome della variabile sulla seconda riga e N è l'esponente di 10 massimo richiesto. Con il tasto 'M' si varia l'esponente dell'incremento, con i tasti '+' e '-' s'incrementa o decrementa il numero e con il tasto 'S' si esce dal loop d'im-

missione. Per esempio: se si vuole immettere il numero 1530, si deve battere il tasto 'M' fino a leggere 10^3 , quindi il tasto '+', comparirà 1000, poi battere nuovamente il tasto 'M' (10^2), cinque volte il tasto '+' fino a leggere 1500, poi battere il tasto 'M' (10^1) e tre volte il tasto '+'. Questo può sembrare un poco macchinoso, ma ci fa risparmiare lo spazio di una tastiera di 12 tasti e dopo un po' ci faremo l'abitudine.

I tre parametri immessi sono controllati dal programma in modo da non superare la lunghezza del buffer e i limiti imposti.

Lo *sweep* in frequenza è creato dalla funzione `SweepFrequency(f1, f2, df)`. Essa passa al modulo AD9833 la frequenza da generare.

Resta da decidere per quanto tempo mantenere le varie frequenze generate. In una prima versione del programma, in base al periodo corrispondente alla frequenza iniziale, si calcolava il numero **Nmeas** di cicli per la ricerca del massimo, conoscendo $t_{meas}=10,218 \mu s$, il tempo di esecuzione di ogni ciclo, con un minimo di 2000.

STM32V2 FR v.271021
No SD present

a

Set Start Frequency

Increment = 10^4
Fi = 20000

b

Set Stop Frequency

Increment = 10^4
Fi = 30000

c

Set Step Frequency

Increment = 10^0
Deltaf = 3

d

20000 -> 30000
Sweep step = 3
N data = 3334

e

Measurement started

Sweep END
Press S to Tx data
Max,Fmax:171,24767
Min,Fmin:33,20054

f

Fig. 12
Screenshot principali del programma.

▼ Listato 1

```
#include <MD_AD9833.h>
#include <U8g2lib.h>
#include <SPI.h>
#include <SD.h>
// Pins for SPI comm with the AD9833 IC
#define DATA PA7 // SPI Data pin number
#define CLK PA5 // SPI Clock pin number
#define FSYNC PB9 // SPI Load pin number (FSYNC in AD9833 usage)
MD_AD9833 AD(FSYNC); // Hardware SPI
#define SDA PB7
#define SCL PB6
#define Mled PB5 // measurements led (active High)
#define Oled_CS PB1
#define Oled_DC PB10
#define Oled_RES PB11
// u8g2 display constructor
U8G2_SSD1309_128X64_NONAME0_F_4W_HW_SPI u8g2(U8G2_R0, Oled_CS, Oled_DC, Oled_RES);

#define AnInPin PA4
#define SD_CS PB0
unsigned int Num, i = 0;
const char keyname[4]= {'+', '-', 'M', 'S'};
byte menu = 0;
int16_t Kbu[] = {PB12, PB13, PB14, PB15}; // keypad buttons
const uint16_t ndata = 4096; // maximum sweep steps
const uint16_t Fmin = 1; // minimum sweep frequency
const uint16_t Fmax = 40000; // maximum sweep frequency
const float tmeas = 10.218; // time of each peak measurement
int16_t RIF [ndata]; // amplitude response buffer
uint16_t StartFreq, StopFreq, Deltaf, ANdata;
uint32_t Nmeas; // number of peak measurements
uint32_t Nmeasma = 0; // maximum number of peak measurements
const uint16_t Nmeasmia = 1000; // minimum number of peak measurements allowed
const uint16_t Ncymi = 5; // minimum number wave cycles allowed
char fname[] = "FREQR00.TXT";
boolean SDok;

|
|

void loop() {

/***** Functions used *****/

byte keypad() {
// Return: 0=no button pressed, 1=+, 2=-, 3=M, 4=S
byte key = 0, pkey = 0;
for (i=0; i < 4; i++){ // identify which button was pressed
if (digitalRead(Kbu[i]) == LOW){
pkey= i+1;
break;
}
}
if (pkey != 0){ // a key was pressed
key = 0;
delay(20); // antibounce
for (i=0; i < 4; i++){ // identify which button was pressed
if (digitalRead(Kbu[i]) == LOW){
key= i+1;
break;
}
}
}
if (pkey == key) return key; else return 0; // the first button must be equal to second
}

void Menu(String prompt, String fvar, byte N){
boolean ok = false;
menu = N;
Num = 0;
while (!ok) {
```



▼ Listato 1 - continua

```
// check if you press the Menu button (3= 'M') and decrease the menu index
if(keypad()== 3) menu= menu-1;
if (menu == 0) menu=N;
// executes the function corresponding to the menu value
if (menu== 1) SetNum(1e0,fvar,prompt);
if (menu== 2) SetNum(1e1,fvar,prompt);
if (menu== 3) SetNum(1e2,fvar,prompt);
if (menu== 4) SetNum(1e3,fvar,prompt);
if (menu== 5) SetNum(1e4,fvar,prompt);
if (menu== 6) SetNum(1e5,fvar,prompt);
if (menu== 7) SetNum(1e6,fvar,prompt);
if(keypad()== 4) ok=true;// exit on 'S' button pressing
}
}
void SetNum(unsigned int incr,String fvar,String prompt){
    u8g2.clearBuffer();
    u8g2.setCursor(0, 12);
    u8g2.print(prompt);
    if(keypad()== 1) Num = Num+incr;
    if (Num > Fmax) Num = Fmax;
    if(keypad()== 2) {
        Num= Num-incr;
        if (Num < 0) Num = 0;
    }
    u8g2.setCursor(0, 36);
    u8g2.print("Increment = 10^");
    u8g2.print(int(log10(incr)));
    u8g2.setCursor(0, 48);
    u8g2.print(fvar);
    u8g2.print(Num);
    u8g2.sendBuffer();
    delay(50);
}

void SweepFrequency(uint16_t f1,uint16_t f2,uint16_t df){
    // Apply a signal to the output
    AD.setMode(MD_AD9833::MODE_SINE);
    // mV conversion (from calibration)
    const float slope = 1.6456;
    const float intercept = -3254.54;
    int m, val, Vmax;
    int BuffInd = 0;
    unsigned int f;
    digitalWrite(Mled, HIGH);// Led ON
    for (f = f1 ; f <= f2; f += df ) {
        AD.setFrequency(MD_AD9833::CHAN_0, f);// generate the frequency
        uint32_t NcyclesT = int(Ncymi*1e6/f);// time interval [us] of 10 cycles
        Nmeas = int(NcyclesT/tmeas);// calculate the number of measurements needed
        if (Nmeas < Nmeasmia) Nmeas = Nmeasmia;// minimum number of measurements
        if (Nmeas > Nmeasma) Nmeasma = Nmeas;// maximum number of measurements
        Vmax = 0;
        for (m = 0 ; m < Nmeas; m++) {
            val = analogRead(AnInPin);
            if (val > Vmax) Vmax = val;
        }
        float mV = slope*Vmax + intercept;// converts to millivolts
        RIF[BuffInd] = int(mV);// save the peak of the response
        BuffInd++;
    }
    digitalWrite(Mled, LOW);// Led OFF
}

void SendData(){
    Serial.print("Start frequency = ");
    Serial.println(StartFreq);
    Serial.print("Stop frequency = ");
    Serial.println(StopFreq);
    Serial.print("Delta frequency = ");
    Serial.println(Deltaf);
    Serial.print("Number of data = ");
    Serial.println(ANdata);
    Serial.print("Max peak measurements = ");
    Serial.println(Nmeasma);
    Serial.print("Min peak measurements = ");
```

▼ Listato 1 - continua

```
Serial.println(Nmeasmia);
Serial.println("f[Hz], A(f)[mV]");
digitalWrite(Mled, HIGH); // Led ON
for (i = 0 ; i < ANdata; i++) {
    Serial.print(StartFreq + i*Deltaf);
    Serial.print(',');
    Serial.println(RIF[i]);
}
digitalWrite(Mled, LOW); // Led OFF
}
void StoreData(){
    digitalWrite(Mled, HIGH); // Led ON
    File dataFile = SD.open(fname, FILE_WRITE);
    if (dataFile) {
        dataFile.print("Start frequency = ");
        dataFile.println(StartFreq);
        dataFile.print("Stop frequency = ");
        dataFile.println(StopFreq);
        dataFile.print("Delta frequency = ");
        dataFile.println(Deltaf);
        dataFile.print("Number of data = ");
        dataFile.println(ANdata);
        dataFile.print("Max peak measurements = ");
        dataFile.println(Nmeas);
        dataFile.print("Min peak measurements = ");
        dataFile.println(Nmeasmin);
        dataFile.println("f [Hz], A(f)[mV]");
        for (i = 0 ; i < ANdata; i++) {
            dataFile.print(StartFreq + i*Deltaf);
            dataFile.print(',');
            dataFile.println(RIF[i]);
        }
        digitalWrite(Mled, LOW); // Led OFF
        u8g2.clearBuffer();
        u8g2.drawStr(0, 12, "Data Saved on SD");
        u8g2.sendBuffer();
        delay(2000);
    }
    dataFile.close();
}
```

Questo creava qualche imprecisione alle frequenze più basse perché i cicli d'onda generati non erano sufficienti per avere una risposta del sistema da misurare.

Nella versione attuale si è imposto che i cicli d'onda generati siano almeno dieci, e si calcola **Nmeas** a ogni frequenza generata, sempre con un minimo di 1000. A differenza della versione attuale, nella prima **Nmeas** era costante durante la misura come anche i tempi di generazione di ogni singola frequenza generata.

Nella versione attuale i tempi di generazione sono variabili e inversamente proporzionale alla frequenza generata: da un massimo per la frequenza iniziale a un minimo per quella finale.

Questo secondo algoritmo ha permesso di ridurre la frequenza iniziale da 10 a 1 Hz. Con una simulazione si sono scelti i valori ottimali: numero cicli minimo = 5, f_1 minima = 1 Hz, f_1 max = 40 kHz e numero minimo di **Nmeas** = 1000.

Se, per esempio, si pone $f_1 = 10$ Hz, $f_2 = 40$ kHz con $df = 100$ si generano 400 frequenze. La prima durerà $5 \cdot T = 5/f_1 = 0,5$ secondi, con **Nmeas** = 48933. La frequenza finale $f_2 = 39910$ avrebbe una durata di $5/T = 0,125$ millisecondi cui corrisponde **Nmeas** = 12, ma il minimo di **Nmeas** è 1000, per cui il tempo di generazione è $1000 \cdot t_{meas} = 10,218$ ms pari a circa 408 cicli d'onda. Per la frequenza più alta si ha un campionamento di solo 2 punti per ciclo, ma ben mille misure del picco. Tutti questi calcoli sono svolti nella funzione **SweepFrequency** che esegue la ricerca del massimo (ampiezza) dell'onda di risposta del sistema, lo converte in millivolt e lo salva nel buffer **RIF[]**. Per facilitare le cose, il programma calcola anche il picco massimo e il minimo della **RIF** con le rispettive frequenze. In questo modo è possibile eseguire una seconda misura più centrata sui picchi. Il led si accende anche per tutto il tempo di trasmissione seriale. Le misure sono immagazzinate nel buffer **RIF[4096]**. In assenza della scheda SD, i dati sono trasmessi via seriale a 115.200 baud ad un Personal Computer, utilizzando un programma di emulazione di terminale.

All'inizio ci sono sette righe di *'header'* contenenti i parametri di sweep e altre informazioni, seguono le righe dei dati con frequenza e ampiezza, separate da virgole. Se nel sistema è presente la SD-Card, alla fine della misura, i dati vengono trasferiti in essa su un file chiamato "FREQRxx.TXT", nello stesso formato, con xx da 00 a 99, incrementato automaticamente secondo il numero di file presenti sulla scheda.

Qui di seguito vi proponiamo la porzione iniziale di un esempio di file del genere, salvato o trasmesso dal nostro strumento:

```
Start frequency = 1
Stop frequency = 2500
Delta frequency = 1
Number of data = 2500
Max peak measurements = 489332
Min peak measurements = 1000
f [Hz], A(f)[mV]
1,2746
2,2746
...
```

Il codice completo, ossia lo sketch utilizzato per la gestione dello strumento, è scaricabile dal nostro sito www.elettronica.it; per ragioni di spazio, nel **Listato 1** lo riportiamo privo della sola parte di **setup()** e quindi proponendo la parte di **include**, **define** ed il **loop()**.



Listato 2

```
% program plotRIF.m legge i dati scritti da STM32devsys
% con le misure di risposta in frequenza e le plotta
clear
clc
files= dir('*.txt');
nfile=size(files,1);% numero file trovati

for indf=1:nfile,
    fprintf('%d) %s\n',indf,files(indf).name);
end
while true
    nf = input('numero di file da leggere = ');
    if (nf>=1) && (nf<=nfile), break, end
end
nomefile = files(nf).name;
fid = fopen(nomefile);
if fid==-1
    error([nomefile ' : il file non esiste']);
end
DELIMITER = ',';
HEADERLINES = 7;
% Import the file
Data = importdata(nomefile, DELIMITER, HEADERLINES);
% Create new variables in the base workspace from those fields.
vars = fieldnames(Data);
for i = 1:length(vars)
    assignin('base', vars{i}, Data.(vars{i}));
end
f = data(:,1);% vettore frequenze in Hz
rif = data(:,2);% vettore risposte in frequenza
clc
n= max(size(f));
[pma,indma] = max(rif);% ricava il picco massimo
[pmi,indmi] = min(rif);% ricava il picco minimo
val3dB = pma/sqrt(2);% per picco massimo
%val3dB = pmi+pmi/sqrt(2);% per picco minimo
disp('Elaborazioni misure di risposta in frequenza')
fprintf('Nome file elaborato      = %s\n',nomefile);
for i = 1:6
    disp(textdata{i})
end
fprintf('Frequenza del picco max = %d Hz\n',f(indma));
fprintf('Frequenza del picco min = %d Hz\n',f(indmi));
fprintf('Valore a -3dB [mV]      = %8.2f\n',val3dB);
fprintf('Valore medio RiF [mV]   = %8.2f\n',mean(rif));
fprintf('Dev. standard RiF [mV]   = %8.2f\n',std(rif));
plot(f,rif,'LineWidth',2),grid,ylabel('A [mV]'),title(['File: ' nomefile]),xlabel('f [Hz]')
hold on
v3dB = ones(n)*val3dB;
plot(f,v3dB,'r','LineWidth',2)
hold off
i1 = find(rif > val3dB,1, 'first');% indice prima intersez.-3dB (picco max)
%i1 = find(rif < val3dB,1, 'first');% indice prima intersez.-3dB (picco min)
f1 = f(i1);
if i1 == indma
    f1 = 0; % correzione necessaria per la banda passante
end
i2 = find(rif > val3dB,1, 'last');% indice ultima intersez.-3dB (picco max)
%i2 = find(rif < val3dB,1, 'last');% indice ultima intersez.-3dB (picco min)
f2 = f(i2);
Bp = f2-f1;
fprintf('Banda passante          = %d [Hz]\n',Bp);
fprintf('Fattore di merito         = %8.2f\n',f(indma)/Bp);
```

IL PROGRAMMA IN MATLAB

Per la lettura delle misure e la conseguente elaborazione grafica al fine di ottenere la visualizzazione sul display, è stato scritto un programma in linguaggio Matlab. Questo programma (lo vedete

dettagliato nel **Listato 2**) va caricato in una cartella di lavoro, insieme ai file delle RiF scaricati dalla SD-Card o trasmessi via seriale.

Il programma elenca e numera i file che trova nella cartella, per cui occorre digitare il numero del

Dai alle tue piante il giusto nutrimento!

Misuratore di luce-umidità-PH



Se ti appresti a coltivare un giardino o un orto, ti sarà utile conoscere la qualità del terreno.

Questo strumento 3 in 1 ti aiuterà a capire rapidamente quando le piante hanno bisogno di acqua, controllare il pH e capire se ricevono abbastanza luce. Basta inserire la sonda nel suolo a una profondità di almeno 10 cm per ottenere dati precisi. Leggero e portatile, lo puoi utilizzare sia per misurazioni in campo sia in vaso, permettendoti di mantenere le piante in condizioni ottimali.

Il misuratore non richiede nessuna fonte di alimentazione.



**FUTURA
ELETTRONICA®**

www.futurashop.it

file da elaborare. A questo punto il programma lo importerà, separando le prime sette righe di intestazione e costruirà la grafica della risposta in frequenza, tracciando in rosso anche la riga corrispondente al punto di attenuazione a -3dB. Il grafico creato dal programma Matlab sarà proposto in scala lineare sui due assi.

CONCLUSIONI

Si chiude qui la presentazione e descrizione del nostro progetto di misuratore della risposta in frequenza in BF, basato su generatore in tecnica DDS e gestito da un microcontrollore STM32 a 32 bit integrato in una board dedicata.

Vi abbiamo esposto la struttura dello strumento esponendovi le sue peculiarità e i vantaggi che il suo utilizzo in laboratorio può offrirvi; avete avuto modo di comprenderne il funzionamento e di valutare alcuni esempi di misura, oltre che di conoscere il firmware che lo gestisce.

Abbiamo completato la trattazione suggerendovi del codice in Matlab che vi permette di elaborare, anche graficamente, le misure che lo strumento effettua e salva in SD-Card o trasmette tramite l'interfaccia seriale della scheda STM32.

Adesso sta a voi far fruttare ciò che avete acquisito, utilizzando al meglio il nostro misuratore di Rf e, se ve la sentite, andando a modificare il codice per trasformare lo strumento in modo che possa implementare nuove misure e funzionalità: ad esempio, fare da semplice generatore o sistema analogico di acquisizione. □



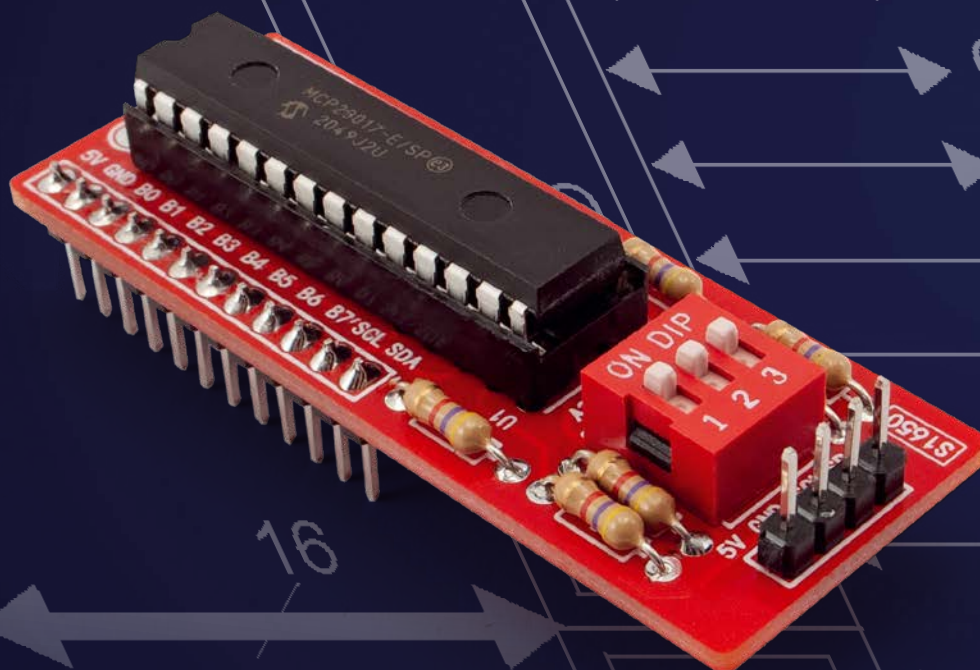
Cosa occorre?

Il componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. La scheda Bluepill (cod. STM32ARMCORE), è disponibile a Euro 14,80, la breakout DDS con AD9833 (cod. AD9833BREAKOUT) a Euro 6,90, il modulo con micro SD card (cod. MICROSDSTORAGE) a Euro 5,50, il display OLED 128x64 (cod. OLED242W) a Euro 28,00.

Il circuito stampato può essere realizzato utilizzando il servizio di prototipazione dei PCB (cod. PCBPRODUCTION).

Il materiale va richiesto a:

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



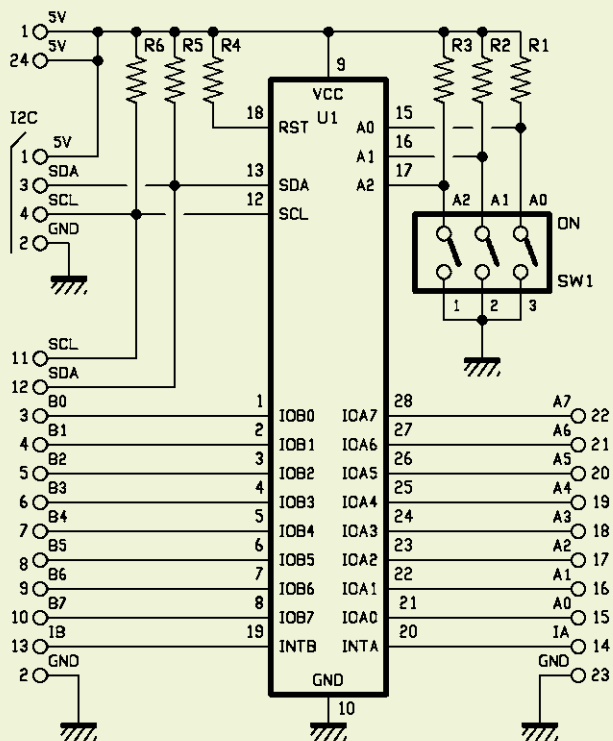
BREAKOUT

PER MCP23017

di ALESSANDRO SOTTOCORNOLA

Sperimentiamo con il popolare I/O expander della Microchip, attraverso una scheda applicabile anche su breadboard, interfacciabile con Arduino.

Eistono varie situazioni pratiche in cui occorre gestire numerosi segnali e attuatori con un microcontrollore, che però non ha sufficienti pin di I/O; in tali situazioni è utile ricorrere a quegli integrati noti come I/O expander, la cui funzione è serializzare un certo numero di linee di input e output così da controllarle tramite un solo I/O, ovvero una linea di comunicazione seriale come può essere la comune e diffusissima I²C-Bus, presente in molti microcontrollori e, non ultime, nelle schede Arduino. Per rispondere all'esigenza di moltiplicare (espandere, appunto, come dice il termine "expander"...) le linee di I/O, abbiamo realizzato la breakout board basata su MCP23017, che pro-



poniamo in questo articolo, sulla quale troviamo tale integrato Microchip che è appunto un I/O expander a 16 bit, oltre all'hardware minimo per farlo funzionare, che nel caso specifico consiste in un dip-switch e qualche resistore di pull-up. Ciò diventerà più chiaro guardando lo schema elettrico della scheda, che trovate in queste pagine.

Infine, per darvi un'idea di cosa si può realizzare con un I/O expander, vi proporremo un esempio applicativo dove la breakout board pilota una serie relé (otto in tutto) montati su una scheda pilotabile tramite livelli TTL o la pressione di pulsanti.

Ma andiamo con ordine ed analizziamo lo schema circuitale.

SCHEMA ELETTRICO

Come potete vedere, la breakout board è qualcosa di estremamente essenziale, perché a bordo vi troviamo l'integrato MCP23017 in versione DIP (a montaggio tradizionale) con riportati sulle piazzole laterali, che sul PCB sono a passo 2,54 mm in previsione del montaggio di pin-strip per l'inserzione in altre schede o su breadboard, tutti i piedini ad eccezione dei tre per l'impostazione dei bit meno significativi dell'indirizzo I²C-Bus di periferica. Pertanto è stato previsto un dip-switch

a 3 vie (siglato SW1 nello schema elettrico) per impostare l'indirizzo di comunicazione sull'I²C-Bus, i cui piedini A0, A1, A2, sono provvisti di resistori di pull-up che a dip-switch aperti li mantengono a livello logico alto.

Riepilogando, tutti i pin relativi agli I/O dei registri A e B sono stati portati in esterno su pin strip maschi. Questi pin sono stati disposti su entrambi i lati per poter connettere la nostra scheda con facilità, ad esempio alla scheda relé cod. RELAY8CH della Futura Elettronica, ma ovviamente sono pin I/O digitali, pertanto si può collegare tutto ciò che si vuole, nei limiti della corrente e tensione supportati dall'integrato MCP23017.

Oltre ai pin strip laterali a passo 2,54 mm, che permettono di ospitare il modulo anche in una breadboard, sono stati previsti altri 4 pin su uno strip (sempre a passo 2,54 mm) chiamato I2C, per poter collegare al bus direttamente nella parte superiore del modulo. Resta inteso che i soli pin laterali sono sufficienti per tutta la comunicazione e interfacciamento con il modulo, pertanto i quattro pin del connettore I2C sono posti in parallelo a quelli già esistenti sulla scheda.

Notate che sul circuito stampato, per praticità i pin del registro A sono stati riportati su un lato e quelli del B si trovano sul lato opposto, sempre nell'ottica di semplificare le connessioni.

I piedini facenti capo al bus I²C sono stati riportati sul pin-strip siglato I2C, insieme al positivo 5V e alla massa; entrambi sono provvisti di pull-up. Il reset dell'I/O-expander nella nostra breakout board non è utilizzato, quindi per disattivarlo abbiamo posto il relativo piedino (il 18, RST...) a livello logico alto mediante il resistore R4.

Tra i pin laterali della scheda troviamo anche ripetute le linee GND e +5V.

L'elemento principale del circuito è chiaramente l'MCP23017, che possiamo considerare un convertitore I²C-Bus/parallelo, prodotto dalla Microchip (siglato U1); l'integrato, del quale vedete lo schema a blocchi interno in **Fig. 1**, funziona come periferica (Slave) del bus I²C e supporta le due modalità di input e output. Nella prima permette di trasferire sul bus, su richiesta del dispositivo Master I²C-Bus, gli stati degli I/O del registro A e del B in formato seriale, un byte per ciascun registro; nella seconda, invece, va ad impostare le linee di I/O convertendo i dati in arrivo sul bus I²C nello stato corrispondente delle linee dei registri A e B.

L'integrato MCP23017 fornisce un'espansione I/O seriale/parallela generica a 16 bit ed è disponibile in due versioni: quella qui utilizzata, dotata di

interfaccia I²C-Bus e la MCP23S17, variante ad interfaccia SPI. Il chip è un I/O-expander a 16 bit, ripartiti in due porte da 8 bit ciascuna, interfacciato mediante I²C-Bus; questo significa che con due soli fili, riferiti a massa, consente di acquisire lo stato di ben 16 linee (modalità input) ovvero impostare lo stato logico di esse (modalità output). Le linee di I/O sono, per impostazione predefinita, funzionanti come input.

L'MCP23017 è costituito da più registri in configurazione a 8 bit per ingresso, uscita e selezione della polarità. Il master di sistema può abilitare gli I/O come ingressi o uscite scrivendo i bit di configurazione I/O (IODIRA/B) corrispondenti.

I dati per ogni ingresso o uscita sono conservati nel corrispondente registro di ingresso o di uscita.

La polarità di registro Input Port può essere invertita mediante il Polarity Inversion Register.

Tutti i registri possono essere letti dal master di sistema.

La porta I/O a 16 bit è funzionalmente composta da due porte a 8 bit, ossia port A e port B, cui fanno capi i piedini, rispettivamente 21÷28 e 1÷8.

L'MCP23X17 può essere configurato per funzionare nelle modalità a 8 bit o 16 bit; dispone inoltre dei due pin di interrupt, INTA e INTB, che possono

essere associati alle rispettive porte (INTA per il Port A e INTB per Port B) o possono essere assoggettati all'operazione logica OR in modo che entrambi i pin si attivino se entrambe le porte determinano un interrupt.

L'uscita di interrupt può essere configurata per l'attivazione in due condizioni (che si escludono a vicenda):

- quando uno stato di input è diverso dal suo corrispondente stato del registro della porta di ingresso; questa condizione è utilizzata per indicare al master di sistema che un stato di ingresso è cambiato;
- quando lo stato di un ingresso è diverso dal valore preconfigurato del registro (registro DEFVAL).

Le linee di interrupt INTA e INTB possono essere configurate come active-high, active-low oppure open-drain. Il registro Interrupt Capture acquisisce i valori delle porte nel momento in cui si attiva l'interrupt, memorizzando così la condizione che ha causato l'interrupt.

Il Power-on Reset (POR) imposta i registri sui loro valori predefiniti e inizializza la macchina a stati del dispositivo.

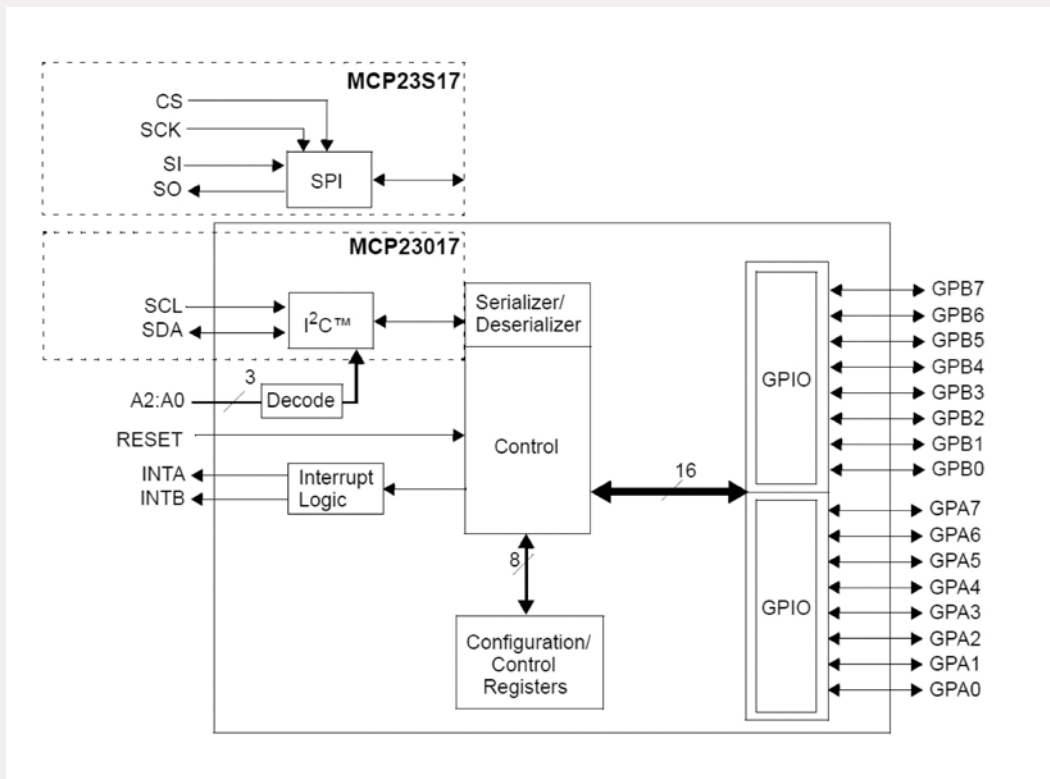


Fig. 1
Schema a blocchi dell'MCP23017: è riportata anche la variante SPI che contraddistingue l'MCP23S17.

Il chip alla base della breakout board qui descritta è un I/O-expander a 16 bit, ripartiti in due porte da 8 bit ciascuna, interfacciato mediante I²C-Bus; questo significa che con due soli fili, riferiti a massa, consente di acquisire lo stato o impostarlo come uscita, di ben 16 linee.

Le sue caratteristiche tecniche sono quelle qui riepilogate.

- Interfaccia dati High-speed I2C, operante a 100 kHz, 400 kHz o

1,7MHz;

- indirizzo I²-Bus impostabile in 8 combinazioni;
- pin di interrupt configurabili per livello e funzione logica;
- sorgente di interrupt configurabile;
- registro inversione di polarità per gli input;
- ingresso di reset esterno;
- corrente di standby di 1 µA max;
- tensione di alimentazione da 1,8V a 5,5V.

La necessità di funzionare in modalità bidirezionale si deve al fatto che ogni periferica I²C-Bus deve sia poter leggere (per esempio i comandi) sia poter inviare i dati acquisiti, ad 8+8 bit, lungo il bus. Come tutte le unità per I²C-Bus, l'MCP23017 permette di impostarne l'indirizzo in un range di 8 indirizzi ed allo scopo dispone dei pin A0, A1, A2, i quali consentono di impostare gli address dell'unità Slave, se indirizzata direttamente dal Bus I²C; queste linee si impostano ciascuna mediante un interruttore del dip-switch SW1, per facilitare l'impostazione del caso: ogni dip chiuso imposta lo zero logico sul rispettivo address, mentre, viceversa, un dip aperto determina lo stato logico 1. La possibilità di definire otto indirizzi consente di affacciare fino a otto I/O-expander sullo stesso bus e controllare quindi un massimo di 128 I/O con tre sole linee.

L'intero circuito prende alimentazione dal contatto 5V (in realtà abbiamo due contatti: 1 e 15, disposti sui lati lunghi della breakout board) riferita a massa

(contatti GND, ossia 2 e 23 delle file laterali). Con questo hardware, il funzionamento è il seguente: ogni volta che riceve una stringa lungo la linea SDA dell'I²C-Bus (scandita dal segnale di clock sulla linea SCL) l'integrato MCP23017 esegue il comando ivi contenuto (in questo caso quello che indica di caricare il byte di dati) e dispone le 8 linee di uscita IOA0+IOA7 e IOB0+IOB7 come i relativi bit, ovvero IOA0 assumerà lo stato del primo bit del byte 1, IOA1 quello del secondo e via di seguito. Analogamente avverrà su IOB0+IOB7, che rispecchieranno pari-pari i bit del secondo byte dati. Naturalmente la conversione e presentazione sulle uscite avviene solo a condizione che la stringa ricevuta contenga l'indirizzo I²C-Bus corrispondente a quello impostato, mediante i dip-switch dell'SW1, per l'U1. Alla ricezione di ogni stringa l'integrato aggiorna lo stato delle proprie uscite e i rispettivi livelli logici determinano l'accensione (o la permanenza nello stato di riposo) dei LED componenti i segmenti del display; se non viene inviata alcuna stringa successivamente, il display mantiene il numero visualizzato, perché le uscite dell'integrato MCP23017 sono dotate di latch.

Quanto esposto vale per la modalità output, ossia in scrittura dello stato dei due byte I²C-Bus sui registri di uscita A e B; se invece il comando proveniente dal bus è di lettura, l'MCP23017 acquisisce lo stato degli I/O di ciascun registro e genera due byte, contenenti il primo lo stato di IOA0+IOA7 e il secondo la condizione logica di IOB0+IOB7, quindi li invia lungo il bus I²C come risposta.

Per ogni applicazione cui vorrete destinare la breakout board, nella **Tabella 1**, vi riportiamo la corrispondenza tra gli address e l'impostazione dei dip-switch.

REALIZZAZIONE PRATICA

Bene, ora che abbiamo descritto lo schema elettrico possiamo passare alle note costruttive, ossia a spiegare come realizzare la breakout board; poi vi proporremo un esempio applicativo basato sull'interfacciamento con una scheda Arduino, corredato da un opportuno sketch. Come di consueto, abbiamo disegnato un circuito stampato del quale vi rendiamo disponibili per il download le due tracce (si tratta di un doppia faccia) dal nostro sito web www.elettronica.in.it; da esse potete procedere a preparare il PCB per fotoincisione, dopodiché una volta che lo avrete inciso e forato potete procedere al montaggio dei pochissimi componenti occorrenti, che, in questo progetto, sono tutti a montaggio tradi-

Tabella 1
Impostazione dell'indirizzo di periferica dell'MCP23017.

ADDR	A2	A1	A0
0x20	ON	ON	ON
0x21	ON	ON	OFF
0x22	ON	OFF	ON
0x23	ON	OFF	OFF
0x24	OFF	ON	ON
0x25	OFF	ON	OFF
0x26	OFF	OFF	ON
0x27	OFF	OFF	OFF

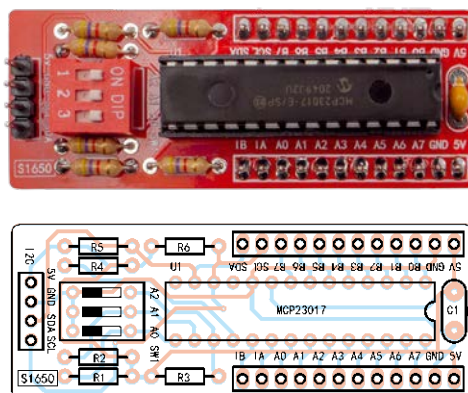
piano di MONTAGGIO

Elenco Componenti:

R1, R2, R3, R4, R5, R6: 4,7 kohm
 C1: 100 nF ceramico
 SW1: Dip-switch 3 vie
 U1: MCP23017-E/SP

Varie:

- Zoccolo 14+14
- Pin strip Maschio 12 vie(2 pz.)
- Pin strip Maschio 4 vie (1 pz.)
- Circuito stampato S1650 (61x23 mm)



zionale, a vantaggio di chi mal digerisce l'SMT. Per prima cosa inserite e saldate le resistenze, poi procedete con lo zoccolo per l'integrato (da posizionare con la tacca rivolta come indicato nel piano di montaggio che vedete in queste pagine) e il dip-switch a tre vie, da montare con lo switch 1 rivolto a sinistra, guardando la scheda con lo zoccolo per U1 in alto. Infine inserite e stagnate nei rispettivi fori il pin strip a 4 poli siglato I2C e poi, dal lato opposto del PCB, inserite e stagnate due file di 12 pin che permetteranno il montaggio su breadboard o l'inserzione su altre schede, ovvero di realizzare eventualmente le connessioni con Arduino tramite i classici jumper a filo maschio/femmina. Al termine della saldatura dei componenti inserite l'MCP23017 nello zoccolo, tenendolo con la tacca rivolta come mostrato nel piano di montaggio in queste pagine e curandovi che non si pieghi alcuno dei terminali sotto al suo corpo. Fatto ciò la vostra breakout board è pronta per la sperimentazione o la prototipazione.

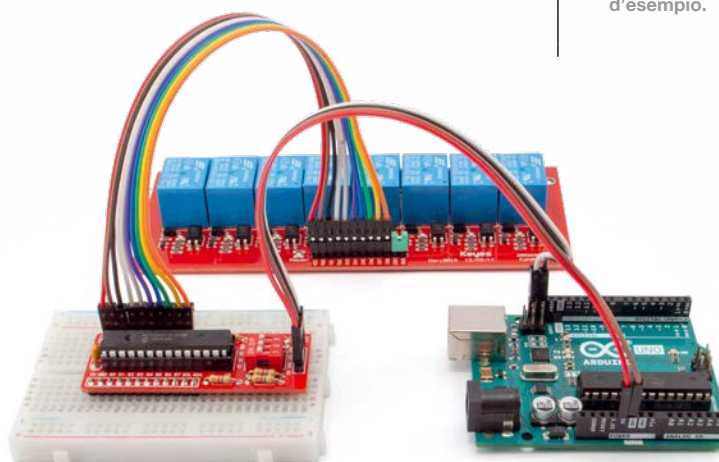
UTILIZZIAMOLA CON ARDUINO

La breakout board nasce per essere interfacciata a un microcontrollore, giacché tipicamente gli I/O-expander vengono utilizzati da dispositivi provvisti di interfaccia seriale a I²C-Bus; siccome Arduino supporta questo bus, abbiamo realizzato un esempio di codice per leggere gestire gli I/O dell'MCP23017 tramite Arduino. Rendiamo disponibile il codice di esempio sul nostro sito www.elettronica.in.it affinché possiate scaricarlo e scriverlo nella vostra scheda tramite l'IDE.

Questo sketch sostanzialmente consente di scrivere lo stato del registro Port A in funzione di un byte inviato da Arduino lungo il bus, i cui bit corrispondono allo stato letto sul Port B, che stavolta funziona da ingresso. Per dare all'esempio un'applicazione concreta, abbiamo deciso di utilizzare gli stati logici degli I/O del Port A, che qui opereranno come uscite digitali, per pilotare una scheda a relé; nello specifico, bisogna collegare le 8 linee di comando delle uscite a relé della scheda RELAY8CH della Futura Elettronica (www.futurashop.it) al banco I/O del Port A, mentre al Port B sono da collegare 8 pulsanti normalmente aperti, aventi in comune il polo collegato a GND. Per realizzare questa applicazione occorre collegare Arduino UNO, la breakout board, la scheda relé e i pulsanti, come mostrato nello schema di cablaggio proposto in **Fig. 2**.



L'hardware per testare lo sketch d'esempio.



↓ Listato 1

```
#include <Adafruit_MCP23X17.h>

Adafruit_MCP23X17 mcp;

int i = 0;

int OUT[] = {7,6,5,4,3,2,1,0}; //Rappresenta il PIN dell'MCP23017 (A7...A0)
int IN[] = {8,9,10,11,12,13,14,15}; //Rappresenta il PIN dell'MCP23017 (B0...B7)
int STATO[] = {0,0,0,0,0,0,0,0}; //Per ogni uscita viene salvato lo stato ad ogni commutazione

void setup()
{
  Serial.begin(9600);
  Serial.println("MCP23017 INPUT/OUTPUT");

  if (!mcp.begin_I2C(0x20)) //0x20 è l'indirizzo dell'MCP23017 con A0=A1=A2 > ON(GND)
  {
    Serial.println("MCP Errore!"); //Se l'MCP non viene trovato si visualizza l'errore
    while (1);
  }
  //Pin del banco A come uscite e del banco B come ingressi
  //La variabile STATO a 0 ad indicare uscite a riposo
  for (i=0; i<8; i=i+1)
  {
    mcp.pinMode(OUT[i], OUTPUT);
    mcp.pinMode(IN[i], INPUT_PULLUP);
    STATO[i] = 0;
  }
}

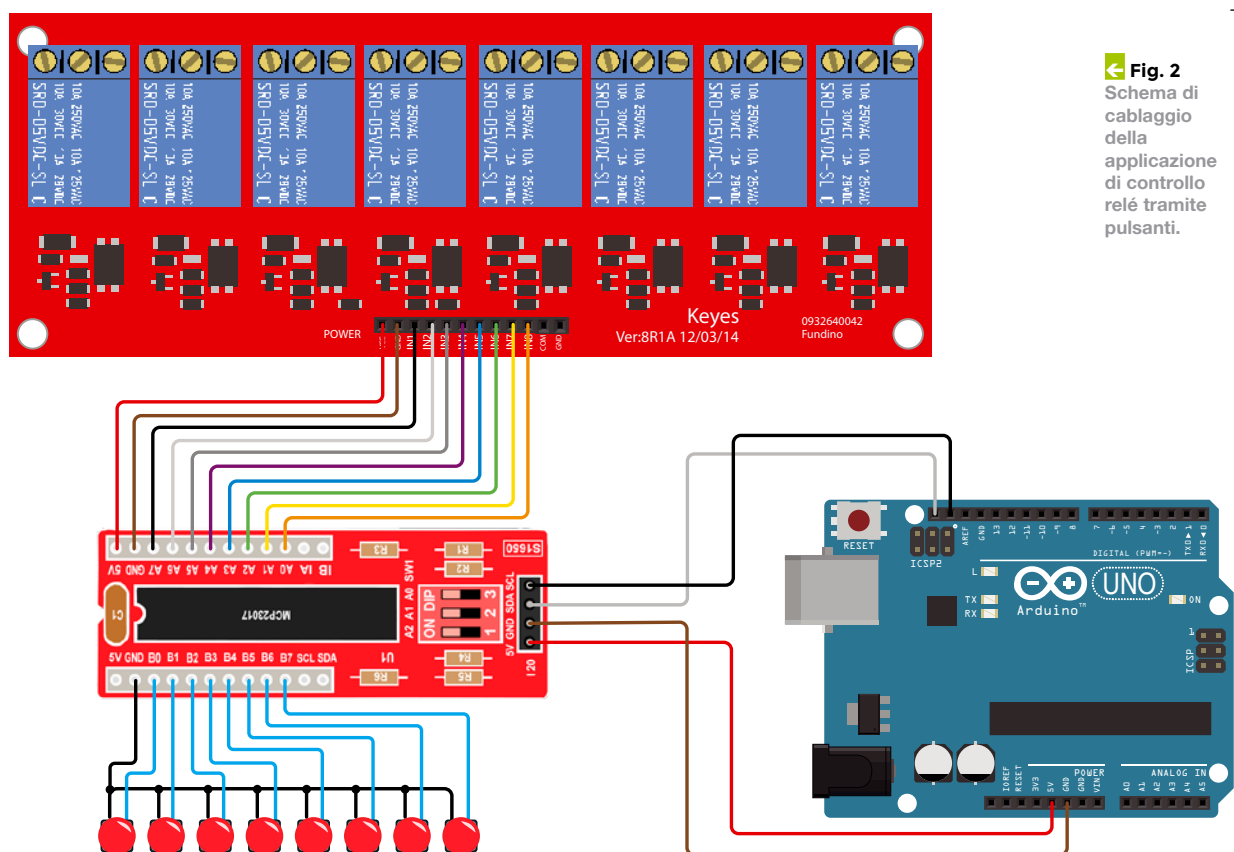
//***** L O O P *****
void loop()
{
  String Testo_Debug = "";
  for (i=0; i<8; i=i+1)
  {
    //Se pulsante premuto o uscita non attivata, la attivo
    if ((mcp.digitalRead(IN[i])==0) && (STATO[i]==0))
    {
      STATO[i] = 1;
      Testo_Debug = "Pulsante " + String(i+1) + " premuto";
      Serial.println(Testo_Debug);
      mcp.digitalWrite(OUT[i], HIGH);
    }
    //Se pulsante rilasciato e uscita attivata, la porto a riposo
    if ((mcp.digitalRead(IN[i])==1) && (STATO[i]==1))
    {
      STATO[i] = 0;
      Testo_Debug = "Pulsante " + String(i+1) + " rilasciato";
      Serial.println(Testo_Debug);
      mcp.digitalWrite(OUT[i], LOW);
    }
  }
  delay(10);
}
```

Trattandosi di normalissimi pulsanti (nello specifico, normalmente aperti) e non avendo elettronica esterna, sono state abilitate le pull-up interne dell'MCP (mediante libreria) in modo da gestirli e riconoscere il cambiamento di stato, in questo modo abbiamo attivato l'uscita rispettiva quando il pulsante viene portato a massa (GND).

Per realizzare tutto ciò, è stato scritto un codice molto semplice basato su Arduino UNO, sfruttando la libreria Adafruit, la quale, come vedete nel **Listato 1**, viene inclusa nella prima riga dello sketch con l'istruzione:

```
#include <Adafruit_MCP23X17.h>
```





Resta inteso che prima di poter caricare il codice nella memoria di programmazione della nostra Arduino è indispensabile scaricare la libreria dal sito www.adafruit.com (o dal nostro sito www.elettronica.in insieme ai file di questo progetto...) e installarla mediante il Gestore delle librerie incluso nell'IDE, o semplicemente estrarre il contenuto del file ZIP e copiare quindi l'intera cartella "Adafruit_MCP23017_Arduino_Library" all'interno della directory "libraries" normalmente presente all'interno del sistema operativo al percorso "Documenti\Arduino\libraries". Caricata la libreria, sarà sufficiente caricare il nostro codice di esempio e caricarlo all'interno della scheda dopo aver scelto la corretta porta COM dal menu "Strumenti" dell'IDE.

Nel codice viene richiesto l'invio di un byte all'MCP23017, contenente lo stato dei pulsanti e i relativi dati vengono elaborati e quindi scritti su un byte diretto all'integrato, che imposterà lo stato delle linee del Port A in maniera stabile, fino al refresh.

Affinché l'interfacciamento funzioni è necessario

impostare correttamente i dip-switch A0, A1 e A2 perché se al sensore non venisse assegnato l'indirizzo 0x20, verrebbe mostrato un messaggio di errore sulla seriale; l'address della breakout board si assegna con la combinazione 000 dei tre bit A0, A1, A2, quindi chiudendo a massa tutti e tre i dip-switch. Se volete modificare l'indirizzo riferitevi alla solita **Tabella 1**, fermo restando che dovrete modificare l'address scritto nello sketch.



Cosa occorre?

La breakout di espansione I/O con MCP23017 (cod. FT1650K) è in vendita in scatola di montaggio a Euro 8,00. Il prezzo si intende IVA compresa.

Il materiale va richiesto a:

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

NUOVE TELECAMERE FULL COLOR technology

Le telecamere Full-color sono dotate di una nuova tecnologia che permette di ottenere immagini a colori anche in condizioni di scarsa luminosità, sia di giorno che di notte.

IP CAMERA 4 MPX – FULL COLOR

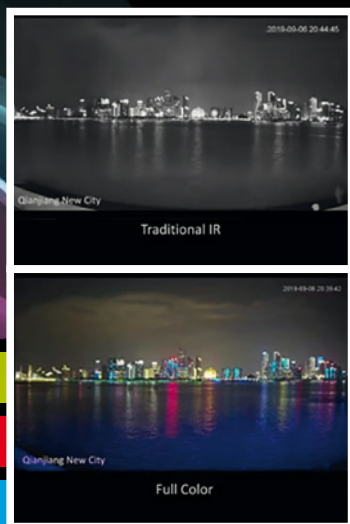
Risoluzione immagine: 4 MPX

Obiettivo: 3,6 mm

Illuminatore luce bianca - numero LED luce bianca: 2 - portata: 30 metri

Grado di protezione: IP66

Alimentazione: 12 VDC



€ 52,00

cod. FR816N

TELECAMERA FULL COLOR
2 MPX

Risoluzione immagine: 2 MPX

Obiettivo: 3,6 mm

Portata Illuminatore: 20-30 metri

Grado di protezione: IP67

Alimentazione: 12 VDC



€ 69,00

cod. FR816VAR

TELECAMERA FULL COLOR
2 MPX - VARIFOCAL

Risoluzione immagine: 2 MPX

Obiettivo: varifocal 2,8-12 mm

Portata Illuminatore: 40 metri

Grado di protezione: IP67

Alimentazione: 12 VDC



€ 44,00

cod. FR817

TELECAMERA DOME
FULL COLOR - 2 MPX

Risoluzione immagine: 2 MPX

Obiettivo: 3,6 mm

Portata Illuminatore: 20-30 metri

Grado di protezione: IP67

Alimentazione: 12 VDC

IN VETRINA



GENERATORE DI FUNZIONI UNI-T UTG2025A

del Prof. MICHELE MENNITI

Abbiamo
provato per voi
uno strumento
professionale che
arricchirà il vostro
laboratorio: ecco i
risultati.

Periodicamente proviamo e condividiamo con voi sulle pagine di Elettronica In dispositivi di vario genere e strumenti da laboratorio: è stato così per la recensione, nel numero 257, di un generatore di funzioni di fascia economica, adatto agli hobbisti, ma comunque con buone caratteristiche. Questa volta invece abbiamo testato uno strumento professionale, il generatore di forme d'onda e funzioni arbitrarie, UTG2025A della UNI-T. Come la maggior parte dei moderni generatori di funzioni, anche questo di basa sulla tecnologia DDS (Direct Digital Synthesis) che, nelle applicazioni più recenti, ha raggiunto livelli di stabilità e precisione di tutto rispetto. È inoltre dotato di due porte USB, una per il collegamento al PC e l'altra per la lettura/scrittura di memorie di tipo U (tipicamente pen-drive)



Fig. 1
Il contenuto
della confezione
del generatore
UTG2025A.

con le quali il generatore si interfaccia per leggere o scrivere dati relativi a forme d'onda arbitrarie. Questo strumento è in grado di generare tutte le tipiche forme d'onda standard, ma anche quelle arbitrarie, ossia quelle realizzabili in base alle proprie esigenze e quindi disegnabili punto per punto, con vantaggi inimmaginabili; alcuni di questi segnali si sono rivelati talmente utili nel tempo da essere integrati in moltissimi strumenti e apparati di nuova generazione, come ad esempio il Sinc, l'ECG (elettrocardiogramma) le onde multitonali, ma il limite è dato dalla fantasia dell'utente. Come è noto ogni forma d'onda ha le proprie specifiche caratteristiche, che il generatore permette di variare in base alle necessità tecniche.

IL GENERATORE UTG2025A

Il generatore di funzioni e forme d'onda arbitrarie UTG2025A si presta quindi a tutte le possibili applicazioni nel campo dell'elettronica, siano esse di tipo hobbistico, didattico o professionale.

Iniziamo dunque a vedere con cosa abbiamo a che fare: una volta aperta la confezione, troviamo lo strumento ed una eccellente dotazione di accessori: un cavo di alimentazione con spina SCHUKO, un cavo BNC-coccodrillo rosso e nero, un cavo BNC maschio-maschio, un cavo USB per il collegamento al PC, ed una pen drive contenente il manuale in lingua inglese (**Fig. 1**). L'autore del presente articolo ha preparato un ottimo manuale, completamente in Italiano, decisamente più chiaro ed esauritivo di quello originale, correggendo anche diversi refusi. Lo abbiamo inserito direttamente nella confezione, per immediata disponibilità di chi acquisterà questo ottimo strumento. L'apparecchio ha le caratteristiche fisiche degli strumenti professionali, con il contenitore metallico e tutti gli angoli anteriore posteriore bordati in gomma, con la doppia funzione di piedini antiscivolo e di protezione dagli urti. Inoltre è dotato di una comoda maniglia che, oltre a servire per il trasporto, può essere bloccata in cinque diverse posizioni, permettendo così anche di orientare ed inclinare lo strumento in base alla collocazione sul banco. Le dimensioni del generatore sono 265 (L) x 110 (H) x 320 (P) mm. In **Fig. 2** abbiamo riportato il pannello frontale dello strumento, con la descrizione delle singole parti. Il pannello anteriore, da sinistra a destra presenta: il tasto di accensione e spegnimento, la porta USB, il display grafico TFT, alla cui base e sul lato destro si trovano due serie di sei pulsanti per la selezione di

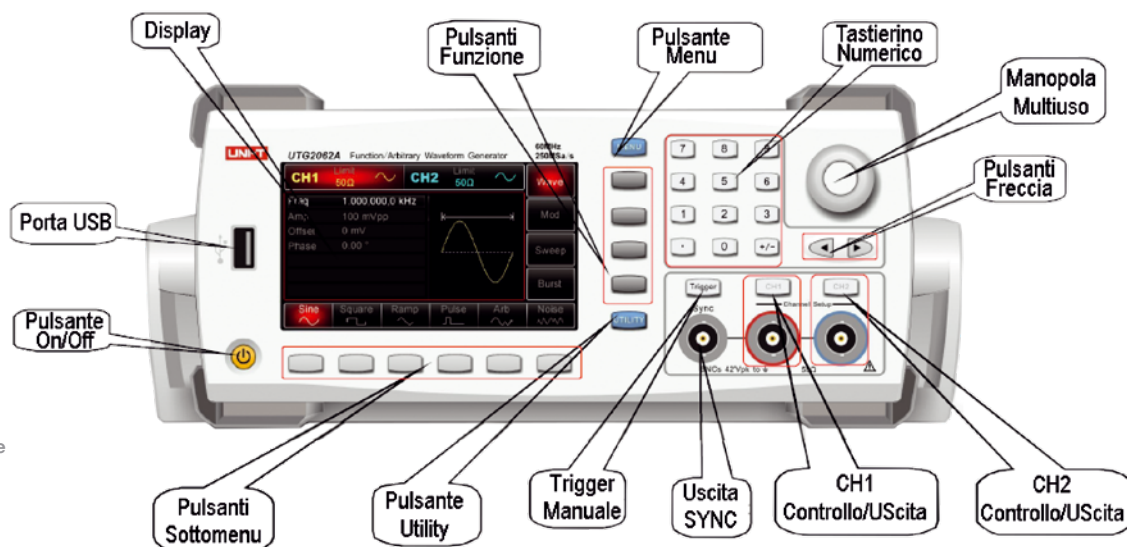


Fig. 2
Il pannello frontale
del generatore
UTG2025A.

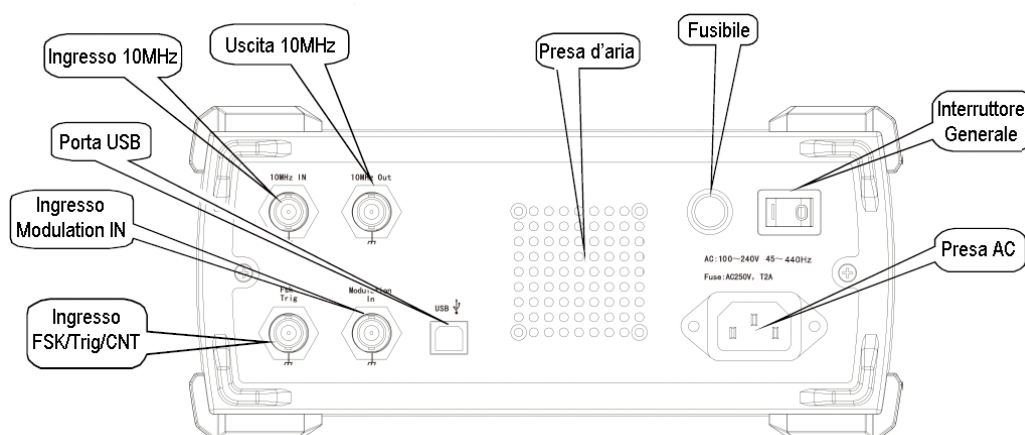


Fig. 3
Il pannello
posteriore del
generatore
UTG2025A

menu, sottomenu e parametri, un tastierino numerico per l'inserimento diretto dei valori, una manopola multiuso con un pulsante centrale, due pulsanti freccia, per la navigazione tra i menu e nei valori dei parametri, tre BNC di uscita (uno per il sincronismo, e due per i canali CH1 e CH2), infine troviamo, subito sopra i BNC, un pulsante di Trigger e i due pulsanti di attivazione/disattivazione dei due canali CH1 e CH2. In **Fig. 3** abbiamo riportato il pannello posteriore dello strumento, con la descrizione delle singole parti che lo compongono.

Il pannello posteriore presenta quattro BNC per:

- ingresso per segnale di clock esterno a 10 MHz;
- uscita segnale di 10 MHz;
- ingresso per la modulazione digitale esterna, che funge anche da input per il frequenzimetro e da input per il Trigger esterno;
- ingresso per la modulazione analogica ester-

na, porta USB, e la sezione AC con presa VDE, portafusibile ed interruttore di accensione generale.

La **Fig. 4** mostra l'organizzazione del display grafico del generatore.

La parte superiore è divisa in due sezioni, una per ogni canale, distinte da una diversa colorazione di base: rossa per il CH1 e azzurra per il CH2, tali colori trovano corrispondenza con la cerchiatura dei rispettivi BNC d'uscita.

Il riquadro centrale presenta a sinistra i parametri della forma d'onda selezionata e a destra l'anteprima in base ai parametri inseriti; alla base e sulla destra infine sono presenti dieci piccoli riquadri che rappresentano la funzione associata in ogni istante ai pulsanti corrispondenti.

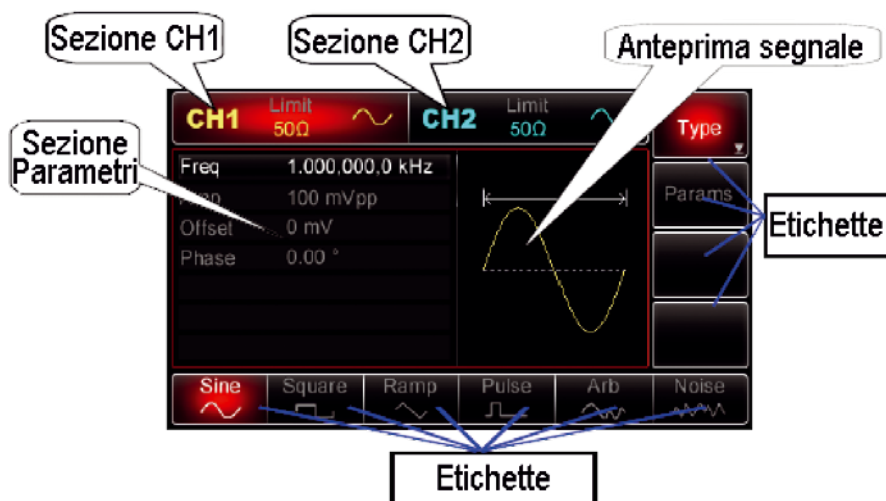


Fig. 4
Il display
grafico.

CARATTERISTICHE DEL GENERATORE UTG2025A

Display: TFT LCD a colori da 4,3 pollici

Canali: 2 (indipendenti)

Frequenza di campionamento: 125 MS/s

Forme d'onda standard: sinusoidale, quadra, rampa, triangolare, impulso, rumore gaussiano, DC, arbitrarie Range di Frequenza:

- Sinusoidale: 1 μ Hz $\div$ 25 MHz
- Quadra: 1 μ Hz $\div$ 5 MHz
- Rampa: 1 μ Hz $\div$ 400 kHz
- Impulso: 500 μ Hz $\div$ 5 MHz
- Arbitraria: 1 μ Hz $\div$ 5 MHz
- Gaussiana: banda passante 25 MHz (-3 dB)
- Risoluzione: 1 μ Hz

Forme d'onda Arbitrarie:

- Risoluzione verticale: 14 bit
- Profondità di memoria: 8 kpts
- Ampiezza del segnale di uscita: fino a 5 Vpp con impedenza di 50 Ω o fino a 10 Vpp in modalità ad alta impedenza
- Impedenza di uscita: regolabile da 1 Ω fino a 1 k Ω , o High Z, impostazione di fabbrica 50 Ω

DC OFFSET

- Range ± 5 V (50 Ω) - ± 10 V (High Z)

COUNTER

- Misurazione della frequenza dei segnali compatibili TTL: Sì, nell'intervallo 100 mHz $\div$ 200 MHz con una precisione di ± 51 ppm

VARIE:

- Tipo di modulazione: AM, FM, PM, ASK, FSK, PSK, PWM
- Modalità Sweep e Burst
- Distorsione armonica totale: 20 kHz, 1Vpp (valore tipico): <0,2%
- Protezione da cortocircuito in uscita: Sì
- Interfaccia: USB

CARATTERISTICHE GENERALI:

- Colore: bianco e grigio
- Alimentazione: 100 $\div$ 240 VAC, 50 Hz / 60 Hz
- Consumo: <40W
- Protezione: fusibile 250 V, T2 A
- Display: 4.3" TFT LCD, WVGA (480x272)
- Accessori: cavo USB, cavo di alimentazione, cavo BNC-BNC, software per PC
- Dimensioni (mm): 265x110x320
- Peso: 3200 grammi

CARATTERISTICHE TECNICHE

Vediamo le principali caratteristiche tecniche del generatore, dettagliate nel riquadro in questa pagina. Le forme d'onda standard sono: sinusoidale (frequenza massima 25 MHz), quadra (5 MHz), rampa/triangolare (400 kHz), impulsiva (5 MHz); le forme d'onda arbitrarie possono raggiungere 5 MHz di frequenza massima.

Inoltre è possibile modulare i segnali con tutte le tipologie note: modulazione d'ampiezza (AM), modulazione in frequenza (FM), modulazione di fase (PM), modulazione a spostamento di ampiezza (ASK), modulazione a spostamento di frequenza (FSK), modulazione a spostamento di fase (PSK), modulazione a larghezza d'impulso (PWM).

È possibile infine emettere le varie forme d'onda in modalità Sweep e Burst. L'impedenza d'uscita standard è di 50 Ω , regolabile nel range 1 Ω $\div$ 1 k Ω , con opzione HighZ. L'ampiezza d'uscita può variare tra 1 mVpp e 10 Vpp (ma il limite massimo dipende dal tipo di forma d'onda). L'offset DC può variare tra 1 mV e 5 V. La fase è configurabile tra 0° e 360°. I due canali sono indipendenti, per cui sono utilizzabili anche settandoli in modo completamente diverso tra loro. Il frequenzimetro può misurare segnali logici TTL tra 100 mHz e 200 MHz.

Il display che troviamo sul pannello frontale dello strumento è un eccellente TFT da 4,3".

USO DELLO STRUMENTO

Per accendere il generatore occorre prima posizionare su "I" l'interruttore generale posteriore e poi premere per qualche istante il pulsante on/off collocato sul pannello frontale; dopo circa 30 secondi di fase di avviamento il display mostrerà l'ultima configurazione utilizzata e lo strumento sarà istantaneamente operativo.

Vediamo subito l'uso dello strumento seguendo l'organizzazione dei pulsanti.

Fondamentalmente l'uso del generatore si basa su due pulsanti principali, MENU e UTILITY, che danno accesso ad una serie di sottomenu e parametri.

La **Tabella 1** mostra l'organizzazione del pulsante MENU. Da notare che tutte le impostazioni che vedremo possono essere applicate indifferente-mente al CH1 e/o al CH2, quindi prima di iniziare a settare i parametri occorre selezionare il canale da utilizzare, premendo il pulsante localizzato sopra il relativo BNC d'uscita; se il pulsante si illumina di verde significa che il segnale è già in uscita sul BNC, volendo disattivarlo fino al completamento delle impostazioni, è sufficiente ripremere, in modo da vederlo spento, il canale resterà comunque selezionato. Per cambiare canale è sufficiente premere l'altro pulsante, per il quale vale quanto appena descritto. Per identificare il canale selezionato occorre osservare la parte superiore del display e le etichette dei pulsanti correntemente selezionati: il CH1 è attivo quando gli sfondi sono illuminati di rosso, mentre il CH2 è attivo quando gli sfondi sono illuminati di azzurro; questo vale

MENU																			
Wave							Mod							Sweep		Burst			
Sine	Square	Ramp	Pulse	Arb	Noise	DC	AM	FM	PM	ASK	FSK	PSK	PWM	Line	Log	NCyc	Infinite	Gated	
Param							Param & Carrier							Param		Param			
Freq	Freq	Freq	Period	ArbSel	Amp	DC	vedi Wave							StartFreq	TrgSrc	TrgSrc	Phase		
Amp	Amp	Amp	Amp	Freq	Offset									StopFreq	TrigOut	TrigOut	Polarity		
Offset	Offset	Offset	Offset	Amp										Time	Period	Phase			
Phase	Phase	Phase	Phase	Offset										TrgSrc	Phase				
	Duty	Simmetry	Duty	Phase										TrigOut	Cycle				
			Rise											Carrier	Carrier				
			Fall											vedi Wave	vedi Wave				

← **Tabella 1**
Organizzazione
del pulsante
MENU.

a prescindere dallo stato delle attivazioni delle relative uscite.

Il pulsante Menu

Andiamo ora ad analizzare le funzionalità cui si può accedere attraverso il pulsante siglato MENU, ben riassunte e organizzate con le relative sotto-funzioni, nella **Tabella 1**.

La pressione di questo pulsante mostra sulla destra i sottomenu disponibili, che sono Wave, Mod, Sweep e Burst; andiamo a dettagliarli.

IL SOTTOMENU WAVE

Permette di accedere, tramite i pulsanti posti nella parte inferiore dello schermo (**Fig. 5**), alle forme d'onda standard, elencate di seguito:

- Sinusoidale;
- Quadra;
- Impulsiva;
- Rampa ascendente o discendente (dente di sega);
- Rampa simmetrica (triangolare);
- Rumore gaussiano bianco;
- Arbitrarie.

LA FORMA D'ONDA SINUSOIDALE

La pressione del pulsante Sine attiva istantaneamente la generazione di una forma d'onda sinusoidale, nel display vengono visualizzate le impostazioni attuali, se vanno bene si può attivare l'uscita

del canale selezionato, per modificarle è sufficiente premere il pulsante Param e i pulsanti inferiori permetteranno di accedere ai suoi quattro parametri. Per spostarsi tra i pulsanti si può usare anche la manopola multiuso, premendo poi il suo pulsante centrale per selezionare quello attivo. Una volta selezionato il parametro si attiva il campo col suo valore, se è di tipo numerico la modifica si può effettuare sempre con la manopola multiuso o più facilmente col tastierino numerico; i due pulsanti freccia consentono lo spostamento all'interno del campo, per modificare le singole cifre.

Svolgiamo questa procedura e impostiamo i relativi parametri (**Fig. 6**):

1. premere il pulsante Freq, inserire il valore 2.5 e premere quindi MHz;
2. premere Amp, inserire 300 e premere mVpp;
3. premere Offset, inserire 0 e premere mV;
4. premere Phase, inserire 0 e premere ° (durante la modifica appare una serie di valori predefiniti che è possibile selezionare direttamente: 90°, 180°, 270°, 360°).

Naturalmente se qualcuno dei parametri è già impostato sul valore richiesto, non è necessario modificarlo. Ora è possibile premere il pulsante del canale attivo per accenderlo e visualizzare la forma d'onda generata su un oscilloscopio digitale (DSO). Il risultato è quello visibile nella **Fig. 7**.

← **Fig. 5**
Selezione delle
forme d'onda
standard.

→ **Fig. 6**
Impostazione dei
parametri per la
forma d'onda
sinusoidale.



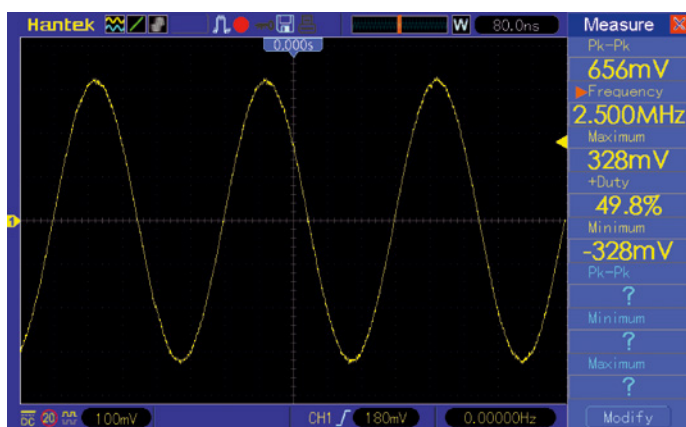


Fig. 7
Forma d'onda
sinusoidale
dell'esempio.

Fig. 8
Forma d'onda
quadra.

Fig. 9
Forma d'onda
impulsiva.

Fig. 10
Forma d'ond
a rampa
ascendente.

Le altre forme d'onda standard

Con lo stesso metodo appena visto per la sinusoide, è possibile impostare le altre forme d'onda standard, proposte in **Fig. 8** (quadra) nella **Fig. 9** (impulsiva) in **Fig. 10** (rampa ascendente) e **Fig. 11** (triangolare) nonché il rumore gaussiano bianco (**Fig. 12**).

Da notare che alcuni pulsanti svolgono una doppia funzione, in quanto il campo associato può essere espresso in due diverse modalità: è il caso del pulsante Freq (Frequenza), i cui valori si esprimono nella scala dell'Hz; se si preme una seconda volta, l'etichetta ed il campo del parametro associato cambiano in Period (Periodo), che viene espresso nella scala dei secondi; la relazione tra i due valori è espressa dalla formula $F=1/T$, cioè frequenza e periodo sono l'uno l'inverso dell'altro. Anche il pulsante Duty (Duty-Cycle) il cui valore si esprime in %, è alternativo al Width (larghezza dell'impulso); il Duty-Cycle indica la percentuale del livello logico High di un segnale rispetto alla durata dell'intera onda (High + Low) e Width calcola la stessa percentuale in relazione al periodo; per esempio se abbiamo un'onda quadra con frequenza di 1 kHz e Duty-Cycle al 70%, il periodo

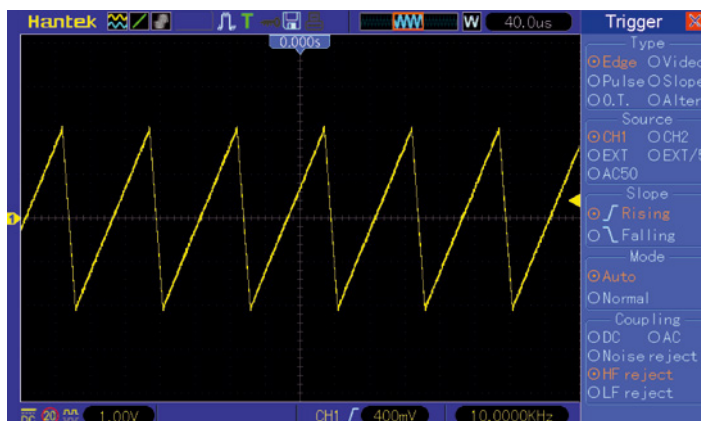
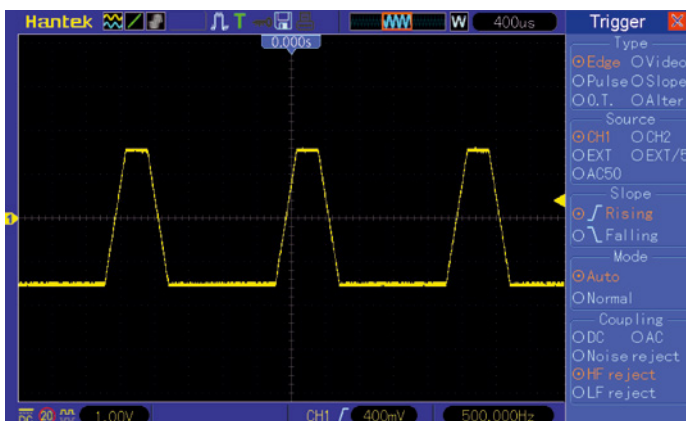
totale è 1/1000 (Hz) quindi 1 ms, mentre il Width è il 70% di tale valore, quindi 0,7 ms (700 μ s).

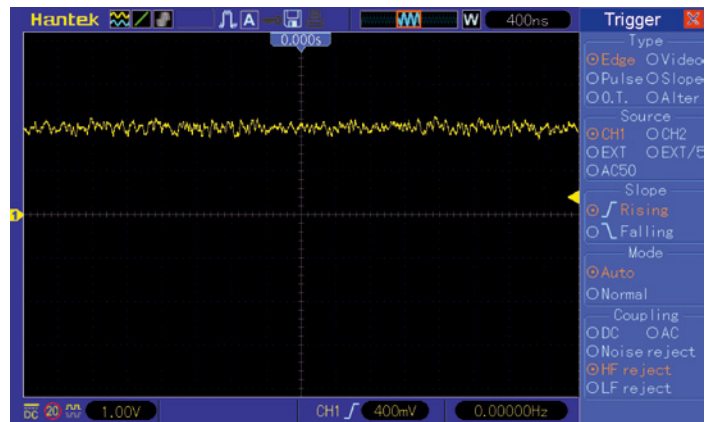
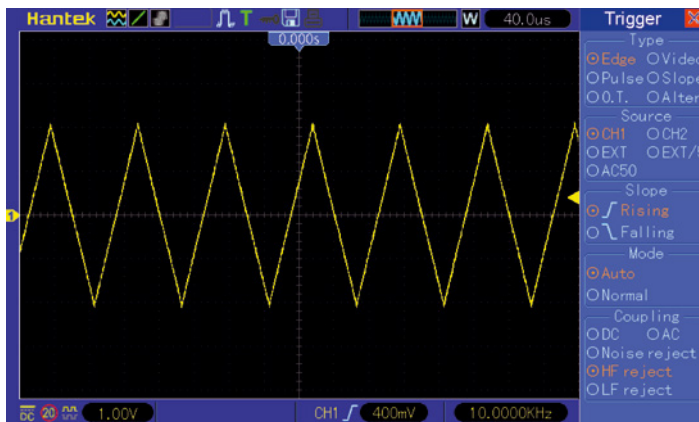
IL SOTTOMENU MOD

Permette di accedere, tramite i pulsanti inferiori, ai numerosi tipi di modulazione di segnale disponibili, ed applicabili alle forme d'onda standard:

- AM (Modulazione d'Ampiezza);
- FM (Modulazione di Frequenza);
- PM (Modulazione d'Impulso);
- ASK (Modulazione a spostamento di ampiezza);
- FSK (Modulazione a spostamento di frequenza);
- PSK (Modulazione a spostamento di fase);
- PWM (Modulazione a larghezza d'impulso).

Dopo aver attivato il sottomenu Mod, la selezione di una delle modalità genera istantaneamente una forma d'onda modulata, in base ai parametri attuali, che vengono visualizzati sul display. La modulazione di una forma d'onda generalmente prevede un segnale portante ed un segnale modulante, quindi è necessaria una doppia impostazione per settarla correttamente. Vi sono tre etichette associate ai





pulsanti di destra, descritte qui di seguito.

- **Type:** mostra alla base del display l'elenco dei tipi di modulazioni disponibili, da selezionare mediante il pulsante corrispondente (**Fig. 13**). La parte inferiore del display ha sei etichette, mentre i tipi di modulazione sono sette, per visualizzare la settima (PWM) occorre premere nuovamente il pulsante **Type**, una ulteriore pressione torna alle prime sei.
- **Param:** mostra alla base del display i parametri del segnale modulante.
- **Carrier:** apre la sezione relativa al segnale portante, allorché sulla destra vengono visualizzate tre etichette:
 - **Type:** (da non confondere con quello della schermata principale), permette di selezionare la forma d'onda standard da usare come portante;
 - **Param** (da non confondere con quello della schermata principale) mostra alla base del display i parametri del segnale portante;
 - **Return** serve per tornare alla schermata del segnale modulante.

Nella Tabella 2 abbiamo riportato i valori impostati sul generatore per ottenere le forme d'onda modulate, dettagliate di seguito, con le relative schermate visualizzate sul DSO. Proponiamo esempi di impostazione della AM, della FSK e della PWM. Quanto alle FM, PM, ASK e PSK trovate i relativi oscillogrammi rispettivamente in **Fig. 14**, **Fig. 15**, **Fig. 16** e **Fig. 17**.

Modulazione d'ampiezza (AM)

Vediamo come settare tutti i parametri di un se-

Fig. 11
Forma d'onda a rampa simmetrica (triangolare).

Fig. 12
Forma d'onda con rumore gaussiano bianco.



Mod													
AM		FM		PM		ASK		FSK		PSK		PWM	
Param (Modulante)													
Source	Internal	Source	Internal	Source	Internal	Source	Internal	Source	Internal	Source	Internal	Source	Internal
ModWave	Sine	ModWave	Square	ModWave	Sine	Rate	300 Hz	CarrFreq	2 kHz	Rate	1 kHz	ModWave	Sine
ModFreq	200 Hz	ModFreq	2 kHz	ModFreq	200 Hz			HopFreq	800 Hz	Phase	180°	Rate	1 kHz
Depth	80%	FreqDev	5 kHz	PhaseDev	200°			Rate	200 Hz			Duty	40%
Carrier (Portante)													
Type	Square	Type	Sine	Type	Sine	Type	Sine	Type	Sine	Type	Sine	Type	Pulse
Freq	10 kHz	Freq	10 kHz	Freq	900 Hz	Freq	15 kHz	Freq	2 kHz	Freq	2 kHz	Freq	10 kHz
Amp	200 mVpp	Amp	100 mVpp	Amp	100 mVpp	Amp	100 mVpp	Amp	1 Vpp	Amp	2 Vpp	Amp	2 Vpp
Offset	0 mV	Offset	0 mV	Offset	0 mV	Offset	0 mV	Offset	0 mV	Offset	0 mV	Offset	0 mV
Phase	0°	Phase	0°	Phase	0°			Phase	0°			Phase	0°
Duty	45%											Duty	50%
												Rise	100 ns
												Fall	100 ns

Fig. 13
Selezione del tipo di modulazione.

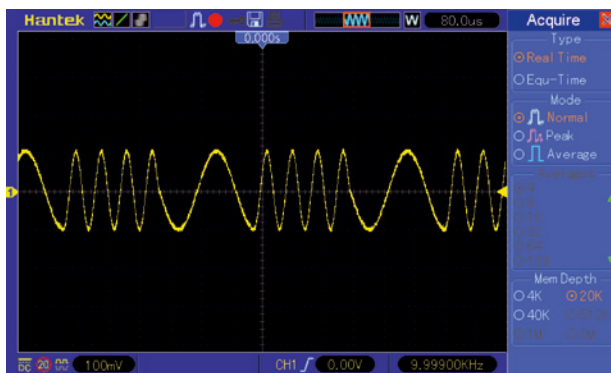
Tabella 2
Parametri usati per il test delle forme d'onda.

gnale modulato in ampiezza.

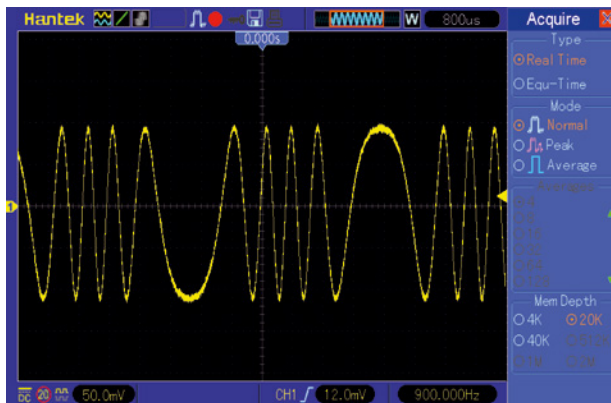
Dopo la pressione del pulsante Mod occorre premere il pulsante AM, quindi premere Param (sezione del segnale modulante) e selezionare o inserire i valori come di seguito descritto.

- Source: è la sorgente del segnale modulante, può essere interna, sfruttando quindi l'elenco delle forme d'onda standard disponibili nel generatore, oppure esterna, per applicare un segnale esterno al generatore, tramite il BNC localizzato sul pannello posteriore; selezionare Internal.

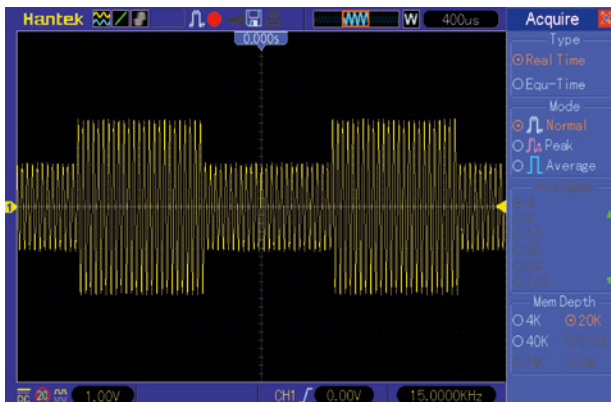
→ Fig. 14
Segnale
modulato
in FM.



→ Fig. 15
Segnale
modulato
in PM.



→ Fig. 16
Segnale
modulato
in ASK.



Da notare che i BNC d'ingresso per fonti esterne sono due, uno prevede l'applicazione di segnali di tipo analogico (Modulation In), l'altro invece accetta solo segnali digitali (FSK/Trig/CNT).

Nel caso della modulazione AM si usa il Modulation In.

- ModWave: è la forma d'onda standard da usare come modulante; selezionare Sine.
- ModFreq: è la frequenza della forma d'onda selezionata; inserire il valore 200 e premere Hz.
- Depth: è la profondità di modulazione (inserire il valore 80 e premere %).

Naturalmente se qualcuno dei parametri è già impostato sul valore richiesto, non è necessario modificarlo. Una volta completato l'inserimento dei parametri del segnale modulante, premendo Carrier si passa a quelli del segnale portante:

- Type: permette di selezionare una forma d'onda standard per la portante; selezionare Square e premere Param (da notare che l'elenco dei parametri visualizzati dipende dalla forma d'onda selezionata);
- Freq: è la frequenza della forma d'onda selezionata; inserire il valore 10 e premere kHz;
- Amp: è l'ampiezza della forma d'onda selezionata; inserire il valore 200 e premere mVpp;
- Offset: è il livello di Offset DC da applicare alla portante; inserire 0 e premere mV;
- Phase: è la fase del segnale portante, inserire 0 e premere °;
- Duty: è il Duty Cycle della forma d'onda selezionata; inserire il valore 45 e premere %.

Come sempre, se qualcuno dei parametri è già impostato sul valore richiesto, non è necessario modificarlo. Terminato l'inserimento di tutti i parametri della portante premendo Return si torna alla schermata della modulante ed il display mostrerà in anteprima il segnale impostato (Fig. 18).

Ora è possibile premere il pulsante del canale attivo per accenderlo e visualizzare la forma d'onda generata su un oscilloscopio digitale (DSO). Il risultato è quello visibile nella Fig. 19.

La modulazione di frequenza (FSK)

Vediamo adesso come settare tutti i parametri di un segnale modulato in FSK. Dopo la pressione del pulsante Mod occorre premere il pulsante FSK, quindi premere Carrier per impostare i parametri del segnale portante, i quali sono:

- Type: permette di selezionare una forma d'onda standard per la portante; selezionare Sine e

premere Param (l'elenco dei parametri visualizzati dipende dalla forma d'onda selezionata);

- Freq: è la frequenza della forma d'onda selezionata; inserire 2 e premere kHz;
- Amp: è l'ampiezza della forma d'onda selezionata, inserire 1 e premere Vpp;
- Offset: è il livello di Offset DC da applicare alla portante, inserire 0 e premere mV;
- Phase: è la fase del segnale portante, inserire il valore 0 e premere il pulsante °.

Come sempre, se qualcuno dei parametri è già impostato sul valore richiesto, non è necessario modificarlo. Terminato l'inserimento di tutti i parametri della portante premendo Return si passa alla schermata delle frequenze di salto e FSK, premere Param e selezionare o inserire i seguenti parametri e rispettivi valori.

- Source: è la sorgente del segnale modulante, può essere interna, sfruttando quindi l'elenco delle forme d'onda standard disponibili nel generatore, oppure esterna, per applicare un segnale esterno al generatore, tramite il BNC localizzato sul pannello posteriore. Selezionare Internal. Da notare che i BNC d'ingresso per fonti esterne sono due: uno prevede l'applicazione di segnali di tipo analogico (Modulation In), mentre l'altro accetta solo segnali digitali (FSK/Trig/CNT). Nel caso della modulazione FSK si usa l'ingresso FSK/Trig/CNT.
- CarrFreq: è la frequenza della forma d'onda portante, già inserita nella sezione Carrier, quindi non va modificata.
- HopFreq: è la frequenza di salto (inserire 800 e premere Hz).
- Rate: è la frequenza FSK; inserire 200 e premere Hz.

Naturalmente se qualcuno dei parametri è già impostato sul valore richiesto, non è necessario modificarlo. Una volta completato l'inserimento dei parametri relativi alle frequenze, il display mostrerà in anteprima il segnale impostato (**Fig. 20**). Ora è possibile premere il pulsante del canale attivo per accenderlo e visualizzare la forma d'onda generata su un oscilloscopio digitale (DSO). Il risultato è quello visibile nella **Fig. 21**.

Modulazione PWM

Vediamo come settare tutti i parametri di un segnale modulato in PWM. Dopo la pressione del pulsante Mod occorre premere il pulsante Type, se l'etichetta PWM non è visibile occorre ripremere Type e selezionare PWM, quindi premere Carrier

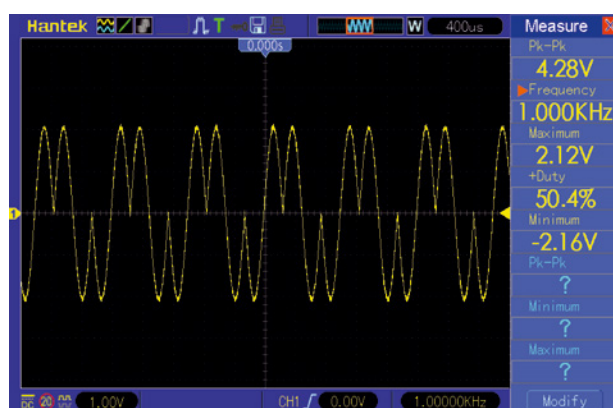


Fig. 17
Segnale
modulato
in PSK.



Fig. 18
Impostazione dei
parametri per la
modulazione AM.

per impostare i parametri del segnale portante:

- Type: servirebbe per selezionare la forma d'onda standard per la portante, ma in questo caso l'unica disponibile è quella impulsiva (Pulse), quindi non bisogna usarlo; l'elenco dei parametri visualizzati dipende dalla forma d'onda selezionata;
- Freq: è la frequenza della forma d'onda selezionata, inserire 10 e premere kHz;

Fig. 19
Segnale
dell'esempio
modulato
in AM.

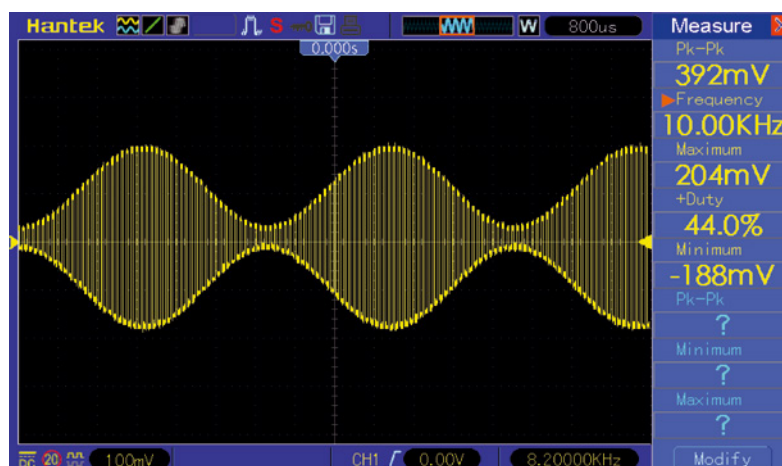




Fig. 20
Impostazione dei
parametri per la
modulazione FSK.

- Amp: è l'ampiezza della forma d'onda selezionata, inserire 2 e premere Vpp;
- Offset: è il livello di Offset DC da applicare alla portante, inserire 0 e premere mV;
- Phase: è la fase del segnale portante, inserire 0 e premere °.
- Duty: è il Duty-Cycle della portante, inserire 50 e premere %;
- Rise: è il tempo del fronte di salita, inserire 100 e premere ns, quindi premere Param per visualizzare il successivo parametro, localizzato in un'altra maschera;
- Fall: è il tempo del fronte di discesa, inserire 100 e premere ns.

Come sempre, se qualcuno dei parametri è già impostato sul valore richiesto, non è necessario modificarlo. Terminato l'inserimento di tutti i parametri della portante, premendo Return si passa alla schermata della forma d'onda modulante, premere Param e selezionare o inserire i valori dettagliati qui di seguito.

- Source: è la sorgente del segnale modulante, può essere interna, sfruttando quindi l'elenco



delle forme d'onda standard disponibili nel generatore, oppure esterna, per applicare un segnale esterno al generatore, tramite il BNC localizzato sul pannello posteriore. Selezionare Internal; da notare che i BNC d'ingresso per fonti esterne sono due, uno prevede l'applicazione di segnali di tipo analogico (Modulation In), l'altro invece accetta solo segnali digitali (FSK/Trig/CNT). Nel caso della modulazione PWM si usa l'ingresso Modulation In.

- ModWave: è la forma d'onda standard da usare come modulante, selezionare Sine.
- Rate: è la frequenza modulante, inserire 1 e premere kHz.
- Duty: è il Duty-Cycle della modulante, inserire 40 e premere %.

Naturalmente se qualcuno dei parametri è già impostato sul valore richiesto, non è necessario modificarlo. Una volta completato l'inserimento dei parametri relativi alla modulazione PWM, il display mostrerà in anteprima il segnale impostato (**Fig. 22**). Ora è possibile premere il pulsante del canale attivo per accenderlo e visualizzare la forma d'onda generata su un oscilloscopio digitale (DSO). Il risultato è quello visibile nella **Fig. 23**.

IL SOTTOMENU SWEEP

Questo sottomenu è dedicato esclusivamente alla modalità sweep, la quale consiste nel generare un range di frequenze, racchiuse tra una frequenza di start ed una di stop, in un lasso di tempo preimpostabile; il processo può essere di tipo lineare o logaritmico. Dopo la pressione del pulsante Sweep, sulla parte destra appaiono le consuete tre etichette per le impostazioni; a questo punto bisogna premere il pulsante Type e quindi Log. Per operare nella modalità Sweep premere il pulsante Carrier per impostare la forma d'onda da usare:

- Type: selezionare Sine, quindi premere Param;

Fig. 21
Segnale
dell'esempio
modulato in FSK.

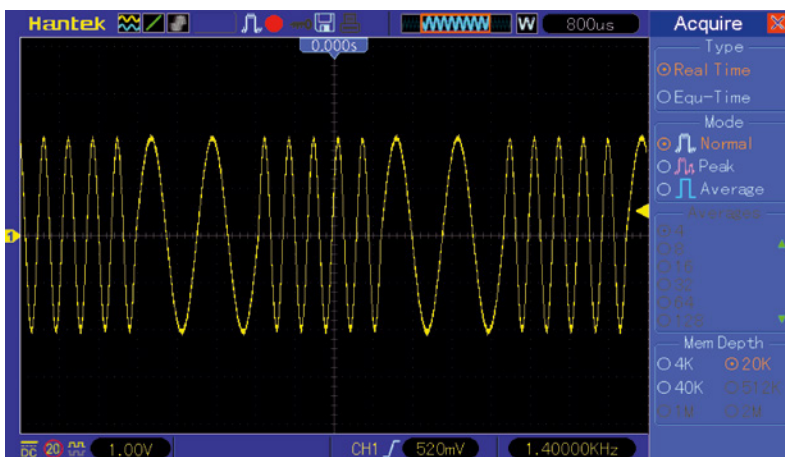




Fig. 22
Impostazione
dei parametri
per la
modulazione
PWM.

- Freq: è la frequenza di START, inserire 100 e premere Hz;
- Amp: è l'ampiezza del segnale, inserire 1 e premere Vpp;
- Offset: è il livello di Offset DC da applicare al segnale, inserire 0 e premere mV;
- Phase: è la fase del segnale, inserire 0 e premere °.

A questo punto si può premere Return per tornare alla maschera precedente, e premere Param:

- StartFreq: è la frequenza iniziale (START) del range, già inserita nella sezione Carrier, quindi non va modificata;
- StopFreq: è la frequenza finale (STOP) del range, inserire 20 e premere kHz;
- Time: è la durata dello Sweep, inserire 5 e premere ms;
- TrigSrc: è la sorgente del Trigger, selezionare Internal;
- TrigOut: è il comando di attivazione o disattivazione del segnale di Trigger destinato ad una apparecchiatura esterna, selezionare Off. Se questo parametro viene impostato su On, il segnale di Trigger viene emesso sul BNC FSK/Trig/CNT, localizzato sul pannello posteriore del generatore.

alla modalità *Burst*, che consiste nel generare treni di impulsi a frequenza fissa, intervallati da un periodo di assenza di segnale. Il processo può essere di tipo lineare o logaritmico. Dopo la pressione del pulsante Burst, sulla parte destra appaiono le consuete tre etichette per le impostazioni. Premere il pulsante Type e quindi NCyc. Premere il pulsante Carrier per impostare la forma d'onda da usare:

- Type: selezionare Sine, quindi premere Param;
- Period: è il periodo del segnale, inserire 5 e premere ms;
- Nota bene: se sul display è visibile l'opzione frequenza, bisogna premere il pulsante Freq per passare all'opzione periodo, e viceversa.
- Amp: è l'ampiezza del segnale, inserire 500 e premere mVpp;
- Offset: è il livello di Offset DC da applicare al segnale, inserire 0 e premere mV;
- Phase: è la fase del segnale, inserire 0 e premere °.
- A questo punto si può premere Return per tornare alla maschera precedente, e premere Param:

Fig. 23
Segnale
modulato
in PWM
relativo
all'esempio.

Naturalmente, anche in questo caso se qualcuno dei parametri è già impostato sul valore richiesto, non è necessario modificarlo.

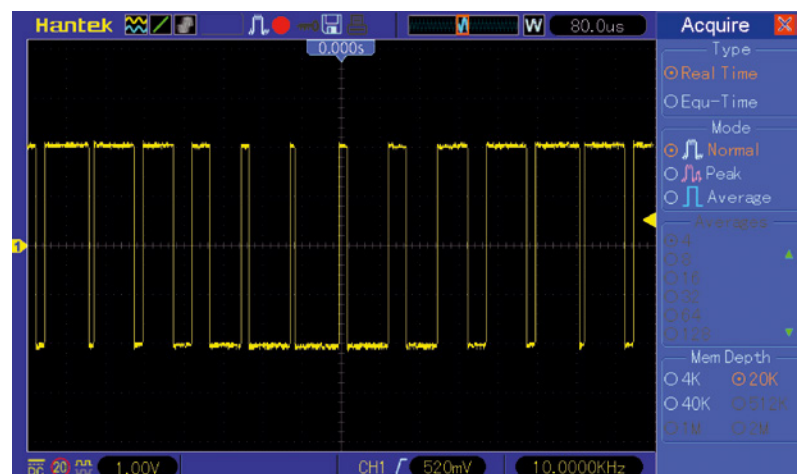
Una volta completato l'inserimento di tutti i parametri relativi allo Sweep, il display mostrerà in anteprima il segnale impostato (**Fig. 24**).

Ora è possibile premere il pulsante del canale attivo per accenderlo e visualizzare la forma d'onda generata su un oscilloscopio digitale (DSO).

Il risultato è quello visibile nella **Fig. 25**.

Il sottomenu BURST

Il predetto sottomenu è dedicato esclusivamente



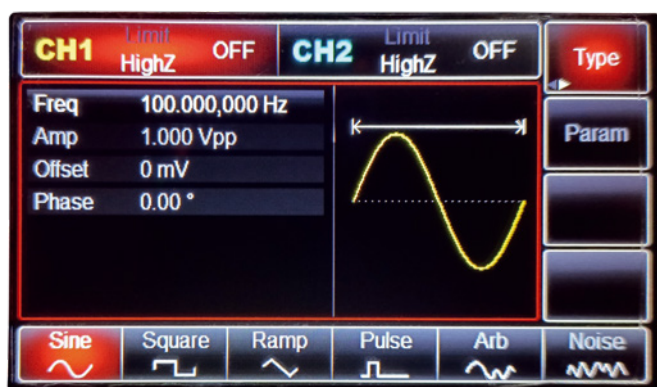


Fig. 24
Impostazione dei
parametri per la
modalità Sweep.



da generata su un oscilloscopio digitale (DSO). Il risultato è quello visibile nella **Fig. 27**.

Le forme d'onda arbitrarie

Per forme d'onda arbitrarie si intendono segnali non standard, quindi anche non periodici, che in genere vengono programmati per usi specifici.

L'etichetta corrispondente all'attivazione dei parametri delle forme d'onda arbitrarie si trova con le forme d'onda standard, quindi occorre seguire il percorso *Menu>Wave>Type>Arb*.

L'unico pulsante disponibile è Param, che permette di attivare tutti i parametri.

Nella memoria interna del generatore sono presenti ben 200 forme d'onda arbitrarie, organizzate in 40 pagine.

Per il nostro esempio attiveremo la forma d'onda che emula il segnale tipico dell'elettrocardiogramma (ECG), che trova importanti applicazioni nel campo della ricerca biomedica.

- ArbSel: permette di cercare la forma d'onda, si può usare la manopola multiuso o i quattro pulsanti con le frecce direzionali per selezionare quella richiesta.

Selezionare la forma n. 46 (con il nome *cardiac.bsv*) e premere Ok o il pulsante centrale della manopola multiuso;

- Period: è la durata del singolo ciclo della forma d'onda selezionata, inserire 820 e premere ms;
- Amp: è l'ampiezza del segnale, inserire 100 e premere mVpp;
- Offset: è il livello di Offset DC da applicare al segnale, inserire 0 e premere mV;
- Phase: è la fase del segnale, inserire 0 e premere °.

Una volta completato l'inserimento di tutti i parametri relativi alla forma d'onda arbitraria, il display mostrerà in anteprima il segnale impostato (**Fig. 28**). Ora è possibile premere il pulsante del canale attivo per accenderlo e visualizzare la forma d'onda gene-

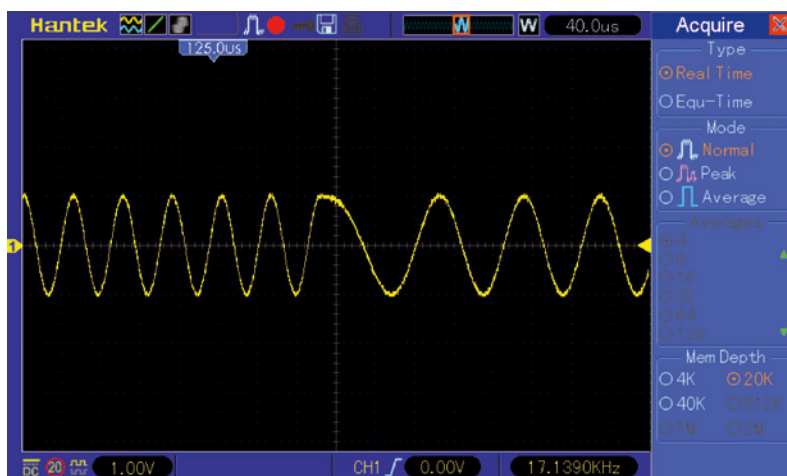
- TrigSrc: è la sorgente del Trigger, selezionare Internal;
- TrigOut: è il comando di attivazione o disattivazione del segnale di Trigger destinato ad una apparecchiatura esterna, selezionare Off. Se questo parametro viene impostato su On, il segnale di Trigger viene emesso sul BNC FSK/ Trig/CNT, localizzato sul pannello posteriore del generatore.
- Period: definisce l'intervallo di tempo tra l'inizio di un ciclo Burst e quello del ciclo Burst successivo, inserire 15 e premere ms;
- Phase: è la fase del Burst, inserire 0 e premere °.
- Cycles: è il numero di cicli che compongono il singolo Burst, inserire 2 e premere Ok.

Naturalmente se qualcuno dei parametri è già impostato sul valore richiesto, non è necessario modificarlo.

Una volta completato l'inserimento di tutti i parametri relativi al Burst, il display mostrerà in anteprima il segnale impostato (**Fig. 26**).

Ora è possibile premere il pulsante del canale attivo per accenderlo e visualizzare la forma d'on-

Fig. 25
Segnale di
esempio della
modalità
Sweep





rata su un oscilloscopio digitale (DSO).
Il risultato è quello visibile nella **Fig. 29**.

IL PULSANTE UTILITY

La pressione di questo pulsante mostra sulla destra i sottomenu: CH1Setting, CH2Setting, Counter e System. Vediamo le funzioni associate, dettagliate nella struttura proposta dalla **Tabella 3** e descritte qui di seguito.

CH1Setting e CH2Setting

Servono per le impostazioni di base dei due canali CH1 e CH2. Sono disponibili sette parametri, l'ultimo dei quali è visibile premendo il pulsante CH1/2Setting una seconda volta.

- Output: (Off/On) serve per l'attivazione dell'uscita del relativo canale; in alternativa si può usare il pulsante posto sopra il BNC;
- Inversion: (Off/On) inverte la forma d'onda in senso verticale;
- SYNC Output: (CH1/CH2/Off) emette un segnale di sincronismo, collegato al canale selezionato, sul BNC Sync, posto sul pannello anteriore;
- Load: è l'impedenza d'uscita del canale, può essere impostata tra 1 Ω e 1 k Ω o su HighZ (alta impedenza).

Counter

Attiva la funzione frequenzimetro, per la lettura di frequenza, periodo e Duty Cycle di un segnale a livello logico TTL, applicato al BNC FSK/Trig/CNT, localizzato sul pannello posteriore. Sono presenti tre etichette: Freq, Period e Duty, ma la pressione dei relativi pul-

Fig. 28
Impostazione dei parametri per la forma d'onda arbitraria.



Fig. 26
Impostazione dei parametri per la modalità Burst.

System

Questo sottomenu serve per utilizzare alcune funzioni generali del generatore. Sono disponibili diversi parametri, di seguito elencati e descritti.

- Language: è possibile selezionare la lingua dello strumento tra Inglese, Cinese e Tedesco.
- Clock Source: (Internal/External) selezione la sor-

Fig. 27
Segnale prodotto in modalità Burst.

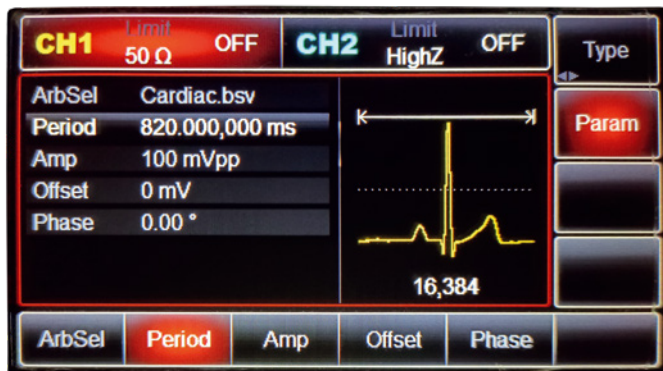
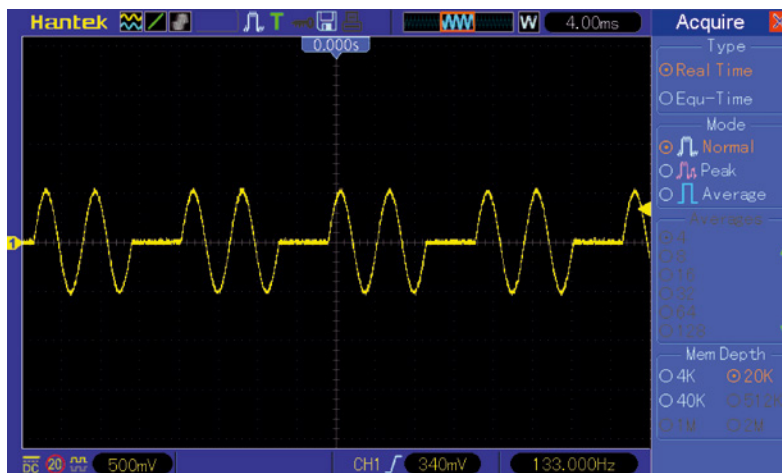


Tabella 3
Organizzazione
del pulsante
UTILITY

UTILITY				
CH1Setting	CH2Setting	Counter	System	
Output	Output	Freq	Language	Load Setting
Inversion	Inversion	Period	Clock Source	Save Setting
SYNC Output	SYNC Output	Duty	Clk Out	SW Upgrading
Load	Load		Beep	Help
AMP Limit	AMP Limit		Num Format	About
Limit Upper	Limit Upper		Backlight	
Limit Lower	Limit Lower			

gente di clock interna dello strumento oppure una esterna da 10 MHz, da applicare al BNC "10MHz In", localizzato sul pannello posteriore.

Questa opzione è utile per il collegamento sincronizzato di più generatori, un master che mette a disposizione il proprio clock interno sul BNC "10MHz Out", e uno o più slave che useranno questo clock come sorgente esterna da applicare al loro BNC "10MHz In".

- Clock Out: (Off/On) attiva l'uscita del BNC "10MHz Out".
- Beep: (Off/On) attiva o disattiva l'emissione di un beep alla pressione dei pulsanti.
- Num Format: (Comma/Space/None) attiva il separatore con la virgola o con lo spazio, oppure disattiva entrambi.
- Backlight: serve per regolare la retroilluminazione del display, premendo uno dei pulsanti inferiori associati ai seguenti valori: 10%, 30%, 50%, 70%, 90% (default) e 100%.
- Load Setting: serve per caricare i parametri di impostazione dello strumento da un file con estensione **.ucf** che deve essere inserito in una pen drive.
- Save Setting: salva i parametri di impostazione attuali dello strumento su una pen drive, sotto

Fig. 29
Segnale
arbitrario tipo
ECG.



forma di file con denominazione "ConfigXXX.ucf", dove XXX è un valore progressivo.

- Software Upgrading: aggiorna il firmware dello strumento tramite uno specifico file da caricare sulla pen drive.
- Help: appare una dicitura che spiega che premendo e mantenendo premuto un pulsante apparirà un riquadro che ne spiega la funzione;
- About: fornisce le informazioni dello strumento: modello, numero di serie e versione hardware e software.

PROGRAMMAZIONE DEL GENERATORE

Un'ultima nota riguarda la possibilità di interagire con il generatore, per la creazione e modifica di forme d'onda arbitrarie, collegandolo ad un Personal Computer tramite la porta USB localizzata sul pannello posteriore. Per fare ciò, nel PC bisogna installare un software specifico reperibile sul sito del produttore al link <https://instruments.uni-trend.com/static/upload/file/20210531/UTG2000A%20Software.zip>.

Naturalmente esiste un corposo manuale, scaricabile tramite il link web <http://unitrend.oss-cn-hongkong.aliyuncs.com/upload/file/20210910/UTG2000A%20and%20UTG2000B%20Programming%20Manual.pdf>. Ulteriori informazioni sono reperibili direttamente sul sito della UNI-T (<https://www.uni-trend.com>), alla pagina dedicata a questo eccellente strumento.

CONCLUSIONI

Bene, possiamo terminare qui la recensione del generatore di funzioni professionale UNI-T UTG2025A, avendo descritto nei particolari le qualità, potenzialità e funzionalità salienti.

Se ritenete di aver riconosciuto, in questa recensione, lo strumento che fa per voi, vi diamo un'indicazione per acquistarlo: il prodotto è disponibile presso la Futura Elettronica, anche on-line su www.futuranet.it, con il codice TP388.



Il generatore di funzioni professionale è disponibile presso Futura Elettronica con il codice TP388 a Euro 498,00. Il prezzo si intende IVA compresa.

Il materiale va richiesto a:

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

Dai la giusta tensione al tuo progetto.

SCEGLI TRA LA VASTA GAMMA DI CONVERTITORI DC-DC QUELLO CHE FA PER TE.

STEP UP

CONVERTITORE DC/DC
CON INGRESSO 3,5-30V
E USCITA 4-30V

cod. MODSTEPUP

€ 5,00



CONVERTITORE DC/DC
CON INGRESSO 10-60V
CON USCITA 12-80V

cod. BOOSTDCDC

€ 25,00



CONVERTITORE DC/DC
CON INGRESSO 3-30V
E USCITA 5-35V

cod. YXA347

€ 6,00



STEP DOWN

CONVERTITORE DC/DC
CON INGRESSO 4-40V
E USCITA 1,25-37V

cod. STEPDOWNDISP

€ 6,90



CONVERTITORE DC/DC
CON INGRESSO 9-35V
E USCITA 5V

cod. DCDCSTEPDOWN

€ 6,50



CONVERTITORE DC/DC
CON INGRESSO 7-36V
E USCITA 1,25-32V

cod. YB728

€ 14,00



CONVERTITORE DC/DC
CON INGRESSO 5-32V
E USCITA 1,2-30V

cod. STEP5V32

€ 5,90



CONVERTITORE DC/DC
CON INGRESSO 3-40V
CON USCITA 1,5-35V

cod. MODULODCDC

€ 4,30



CONVERTITORE DC/DC
CON INGRESSO 7,5-20V
E USCITA 5V-1,2A

cod. YM355

€ 5,00



STEP UP/DOWN

CONVERTITORE DC/DC
CON INGRESSO 4-35V
E USCITA 1,5-25V

cod. STEPUPDOWN

€ 16,00



Entra nel mondo dell'elettronica!

Le scatole di montaggio KIT di Futura Elettronica sono semplici da montare e di facile utilizzo, ideali per avvicinarsi al fantastico mondo dell'elettronica. Non utilizzano microcontrollori e sono fornite con schemi elettrici e istruzioni così da capire e personalizzare il funzionamento.

Interruttore Crepuscolare

cod. FK001 | € 9,00



Modulo Intermittenza

cod. FK002 | € 12,00

Alimentatore stabilizzato da 1,5 a 35 V

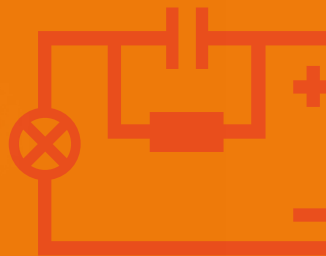
cod. FK003 | € 10,50

Alimentatore stabilizzato duale e regolabile

cod. FK004 | € 14,00

Timer Start e Stop da 1 a 240 secondi

cod. FK005 | € 14,00



Interruttore Battimani

cod. FK006 | € 14,00

Controllo automatico ventilazione

cod. FK007 | € 14,00

Regolatore di velocità per ventole

cod. FK008 | € 12,00

Luce psichedelica a LED

cod. FK009 | € 12,00



Protezione bassa tensione

cod. FK010 | € 9,90

Regolatore PWM per pannelli fotovoltaici 12/24V

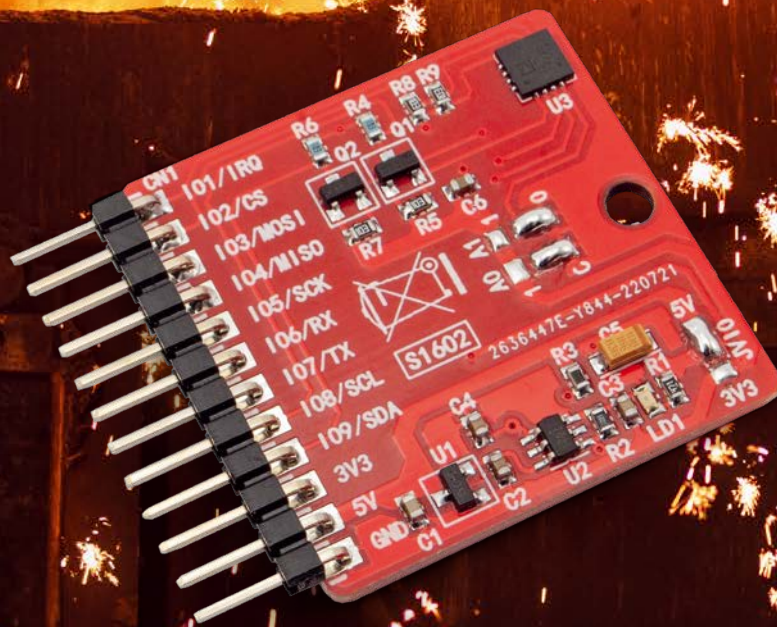
cod. FK011 | € 18,00

LABORATORIO

BREAKOUT

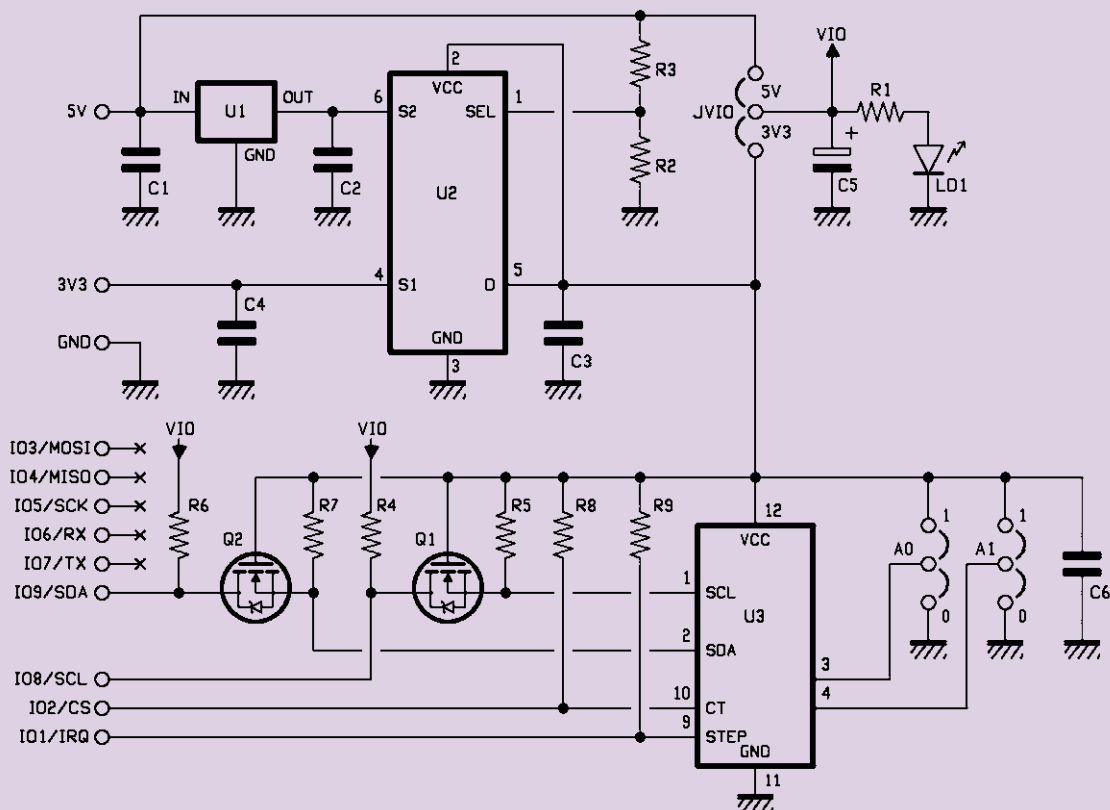
ADT7422

di DAVIDE SCULLINO



Facilita la prototipazione e la sperimentazione con l'integrato sensore di temperatura di Analog Devices, rendendone disponibili i pin su uno strip e provvedendo all'alimentazione.

Uno dei parametri che viene spesso misurato in una grandissima varietà di applicazioni è la temperatura, sia essa dell'ambiente o di macchinari, liquidi, schede elettroniche, sistemi e sottosistemi vari, fino ad arrivare al rilevamento di quella corporea; per questo l'industria elettronica è prodiga di sensori e dispositivi per il rilevamento delle condizioni termiche sia a contatto che contactless, declinati in varie esecuzioni e ciascuno con un proprio ambito applicativo. Tra questi



citiamo l'ADT7422 della Analog Devices, incapsulato in package SMD e idoneo al rilevamento della temperatura ambiente (comprendendo anche quella di camere climatiche e contenitori di apparati elettronici) con un elevato grado di precisione. Come per tutti i componenti di ultima generazione, anche questo integrato è molto piccolo e difficilmente manipolabile da chi non ha attrezzatura e pratica sufficienti a trattare dispositivi per montaggio superficiale, quindi con esso abbiamo realizzato una breakout board allo scopo di semplificarne il collegamento e l'integrazione in vari sistemi, anche a scopo di sperimentazione e prototipazione. In queste pagine ve la descriveremo, partendo da una panoramica sul componente, ossia col vedere di cosa si tratta e quali caratteristiche ha.

L'INTEGRATO ADT7422

Il cuore del progetto è l'integrato ADT7422, il quale è un sensore di temperatura ad elevata precisione, dotato di interfaccia digitale su I²C-Bus e progettato per soddisfare le specifiche di termometria clinica secondo lo standard ASTM E1112 utilizzato a bordo di un circuito stampato (PCB). Internamente all'ADT7422 si trovano un riferimen-

to di tipo band gap, un sensore di temperatura e un convertitore analogico-digitale di precisione che permette di fornire il valore di temperatura rilevato con risoluzione ben a 16 bit, corrispondenti a 0,0078°C, con una precisione fino a $\pm 0,1^\circ\text{C}$ nell'intervallo di temperatura da 25°C a 50°C senza necessità di calibrazione dopo il processo di saldatura sul PCB. La frequenza di campionamento può essere impostata via I²C-Bus. Il riferimento band gap significa avere due tensioni prese a riferimento, una che ha un coefficiente di temperatura positivo e una con coefficiente negativo, i quali si compensano teoricamente, fornendo la massima stabilità possibile della misura della temperatura al variare della temperatura stessa (si possono ottenere tolleranze ridottissime, tra 5 e 100 ppm/°C).

Siccome l'integrato opera a una tensione di alimentazione di 3,3V (il costruttore dichiara come tensione nominale 3V), nel nostro circuito è stato provvisto di un traslatore dei livelli logici che poi descriveremo; la corrente assorbita mediamente è molto contenuta, ossia è tipicamente di 210 μA . L'ADT7422 implementa una modalità di spegnimento che, quando il dispositivo non è utilizzato,

permette di mantenerlo alimentato con una corrente di “sleep” tipicamente di 2 μA a 3 V di alimentazione.

Il range di temperatura di funzionamento dell'integrato spazia da -40°C a $+125^{\circ}\text{C}$, quindi è possibile l'utilizzo in svariate applicazioni, anche industriali. Per la selezione dell'indirizzo di periferica I²C-Bus, sono previsti due pin, ossia A0 e A1.

L'integrato dispone di un pin siglato CT, che è un'uscita open-drain che si attiva al raggiungimento della temperatura considerata critica, il cui valore è programmabile; questo significa che possiamo non solo misurare la temperatura, ma anche impostare una soglia di allarme, con segnalazione da parte dell'ADT7422, senza bisogno di scrivere codice in un microcontrollore per gestire l'allarme.

Abbiamo anche un pin siglato INT, che è un'uscita open-drain che si attiva quando la temperatura supera un limite programmabile. Il pin INT ed il pin CT possono funzionare in modalità sia comparatore che interrupt su evento.

Le principali caratteristiche dell'ADT7422 sono:

- precisione di $\pm 0,1^{\circ}\text{C}$ da 25°C a 50°C a 3V di alimentazione;
- $\pm 0,25^{\circ}\text{C}$ da -20°C a $+105^{\circ}\text{C}$, con alimentazione da 2,7 V a 3,3 V;
- deriva termica di $0,0073^{\circ}\text{C}$;
- tempo di risposta all'accensione di 6 ms;
- assorbimento tipico di 140 μW a 3 V, in modalità 1 SPS (un campionamento della temperatura al secondo);
- interfaccia I²C-Bus.

L'integrato ADT7422 è reso disponibile nel package LFCSP a 16 pad, delle dimensioni di 4×4 mm, che è poi quello da noi utilizzato nel progetto della breakout board.

CARATTERISTICHE TECNICHE

❖ Tensione di alimentazione: 3-5 V_{cc} ❖ Corrente assorbita: 15 mA ❖ Temperatura misurabile: $-40^{\circ}\text{C} + 125^{\circ}\text{C}$	❖ Uscita: I²C-Bus a 3,3/5V ❖ Risoluzione uscita: 16 bit ❖ Deriva termica: $0,0073^{\circ}\text{C}$
--	--

La Fig. 1 propone lo schema a blocchi che descrive la struttura interna del nostro integrato meglio di ogni parola.

SCHEMA ELETTRICO

Bene, una volta che sappiamo come è fatto e in che modo funziona, vediamo come l'ADT7422 viene impiegato nella nostra scheda, la quale, come tutte le breakout board, nasce per semplificare l'utilizzo del componente e per sperimentarne le sue caratteristiche prima di impiegarlo in sistemi e sottosistemi vari. Guardiamo lo schema elettrico, dove il componente è siglato U3 ed è alimentato tramite una sezione di alimentazione che riceve sia 5V rispetto a massa (contatti 5V e GND) che 3,3V (contatto 3V3); questa doppia tensione serve per adattare il bus I²C della breakout board a funzionare a 3,3V o a 5V.

L'alimentazione a 5 volt viene inviata all'ingresso di un regolatore LDO (U1) di tipo

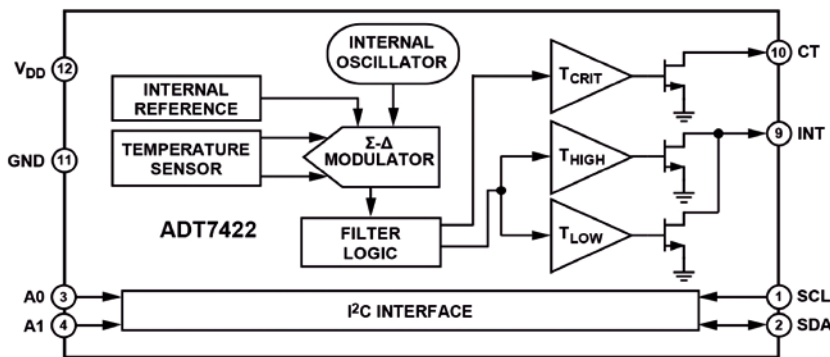


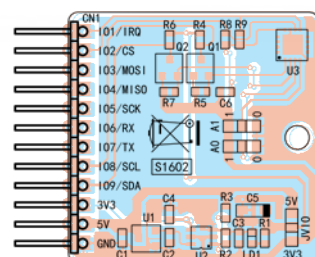
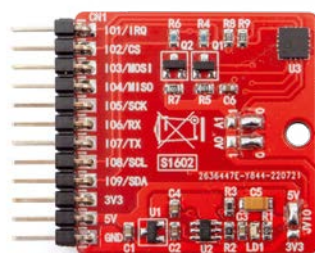
Fig. 1
Schema
a blocchi
dell'ADT7422

Elenco Componenti:

R1: 470 ohm (0603)
 R4, R5, R6, R7, R8, R9: 10 kohm (0603)
 R2: 6,8 kohm (0603)
 R3: 3,3 kohm (0603)
 C1, C2, C3, C4, C6: 1 μ F 6,3V ceramico (0603)
 C5: 10 μ F 6,3V tantalio (1206)
 U1: MCP1700T-3302E/TT
 U2: TMUX1219DBVR
 U3: ADT7422CCPZ-RL7
 Q1, Q2: BSS138W-7-F
 LD1: LED rosso (0603)

Varie:

- Circuito stampato S1602 (33x33 mm)



MCP1700T-3302E/TT la cui funzione è ricavare 3,3V stabilizzati da inviare al selettore di alimentazione; quest'ultimo è sostanzialmente uno switch a MOSFET che fa passare al proprio piedino D una delle tensioni applicate ad S1 (linea 3V3) ed S2 (uscita dell'LDO) in base alla tensione applicata al pin SEL. Per definire la tensione su SEL utilizziamo un partitore resistivo alimentato dalla linea di ingresso 5V, il quale serve ad abbassare la tensione, perché il pin in questione ha una logica 0÷1,8V. Quando è posto a 0 logico, la tensione applicata ad S1 (linea 3V3) viene applicata a D, mentre se è a 1,8V il piedino D viene alimentato dalla linea S2 (uscita del regolatore LDO); ne deriva che quando è presente la tensione di 5 volt, l'integrato switch alimenta il resto del circuito con la tensione fornita dal regolatore a basso drop-out, mentre se applichiamo tensione al contatto 3V3 della breakout board, il circuito riceverà tensione da essa. Nello schema elettrico vedete un jumper a tre vie per la selezione dell'alimentazione, siglato JVIO, il quale permette di scegliere con che potenziale, fra i 5V di ingresso e i 3,3V ricavati dal regolatore LDO, alimentare il traslatore di livelli del bus I²C; l'integrato sensore di temperatura, invece, è alimentato fisso a 3,3V.

In ogni caso, nel circuito è presente il LED LD1, che, alimentato tramite la resistenza di limitazione della corrente R1, si illumina quando il circuito è alimentato, a prescindere dall'impostazione del jumper e dalla fonte scelta. I condensatori disposti a monte e a valle del regolatore LDO (C1 e C2)

nonché tra la linea D dello switch U2 e massa (C3) e sull'ingresso 3V3 (C4) filtrano le tensioni dai disturbi impulsivi. La linea VIO è invece filtrata dall'elettrolitico C5, mentre la Vcc dell'ADT7422 è filtrata dal condensatore ceramico C6.

U2 è un TMUX1219DBVR prodotto dalla Texas Instruments e può essere alimentato da 1,08 V a 5,5 V e, sebbene lo abbiamo utilizzato come commutatore della linea di alimentazione del modulo STP201M, è uno switch bidirezionale, quindi può commutare dai pin S1/S2 verso il D, o viceversa; i piedini S corrispondono ai source dei MOSFET componenti lo switch, mentre il D è il drain in comune.

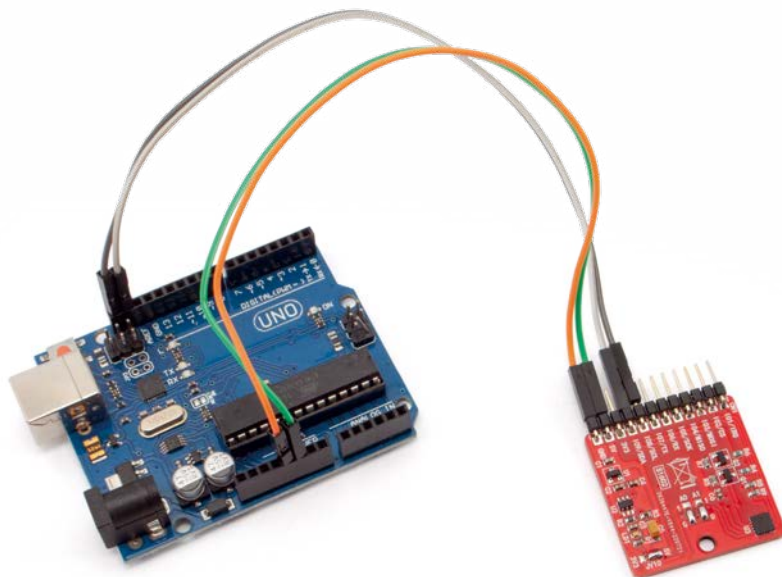
Il doppio jumper permette di impostare la tensione di lavoro del bus I²C (VIO) in base alla tensione cui alimenterete la breakout board; questo perché per rendere più versatile la scheda, abbiamo inserito nel circuito un adattatore a MOSFET per l'interfaccia I²C-Bus utile a traslare i livelli logici supportati dal sensore per adattarli ai TTL standard di schede a logica tradizionale e ad Arduino. Ricordiamo comunque che l'U3 funziona sempre a 3,3V e quindi il jumper JVIO permette di scegliere una tensione lato connessioni esterne della scheda fra 3,3V (quindi la stessa del bus I²C del modulo) e 5V (tipica delle schede Arduino UNO, Duemilanove ecc.). L'integrato ADT7422 dispone di due jumper con i quali possiamo impostare gli ultimi due bit (ossia quelli di minore peso) dell'address che il device esporrà sul bus I²C; in tal modo è possibile optare per uno dei quattro indirizzi possibili, giacché due bit forniscono 4 combinazioni. Il jumper A0, se portato verso la linea 3V3 imposta il rispettivo bit di indirizzo ad 1 logico, mentre se posto a massa porta a livello basso il bit A0. Per A1 si applica un ragionamento analogo. Chiaramente i jumper vanno chiusi solo in una direzione alla volta, ossia su 3V3 o verso massa.

Bene, completiamo ora l'analisi circuitale spiegando come funziona il traslatore di livello del bus I²C, al quale, lo notate dallo schema elettrico, fanno capo solo le linee SCL e SDA; i pin di uscita CT e STEP escono tali e quali ed entrambi forniscono livelli logici 0/3,3V in virtù del pull-up operato dalle resistenze R8 ed R9.

Iniziamo l'analisi del traslatore dalla linea di clock, che sul lato dell'U3 fa capo al piedino 5 (SCL), mentre verso il pin-strip di connessione è collegato a SCL (entrambe le linee sono provviste di un resistore di pull-up, ossia R4 sul lato SCL ed R5 lato sensore): quando SCL del connettore è a livello logico alto, il diodo di protezione interno al

MOSFET Q1 (posto tra drain e source) è interdetto perché il source si trova negativo (alimentazione VIO a 5 volt e bus lato strip, a 5V) o equipotenziale (VIO a 3,3V) rispetto al drain (che è mantenuto a 3,3 volt dalla R4) e pur se l'impulso non giunge al contatto SCL dell'U3, quest'ultimo è ad 1 logico grazie al pull-up operato da R5. Invece quando il contatto SCL del pin-strip è a zero logico (0 V), il MOSFET è ancora interdetto (perché il suo gate è polarizzato con 3,3V) ma conduce il suo diodo di protezione, dato che il source è a 3,3V mentre il catodo è a zero, quindi il livello basso su SCL trascina a zero logico (0,6 per l'esattezza) il piedino 5 dell'U3. Il canale SCL è unidirezionale, quindi la situazione opposta non si presenta, cioè U3 riceve il clock dall'unità Master, che nella nostra applicazione, come esporremo tra breve, sarà Arduino. Passiamo adesso all'SDA, analizzando il caso in cui è l'unità Master a inviare i dati e ipotizziamo che il dispositivo con cui la breakout board è interfacciata abbia il bus operante a 5V: quando tale contatto è a zero logico, il diodo di protezione del Q2 conduce e trascina a 0,6 V il source e con esso il piedino 4 dell'U3 cui fa capo; invece quando è a 5 volt, MOSFET e diodo sono interdetti, quindi SDA dell'U3 viene portata a 3,3V (livello alto) dal resistore di pull-up Rt. Se invece è il sensore ADT7422 a trasmettere dati, quando pone il proprio piedino 4 a zero logico il MOSFET Q2 conduce tra drain e source, perché il suo elettrodo di source è a zero volt mentre il gate è a 3,3 V, quindi la sua corrente di drain trascina a circa 0 V l'SDA del connettore, a causa della caduta sul resistore di pull-up R6; invece quando il pin 6 dell'U3 è a livello alto (3,3 V) il source e il gate sono equipotenziali e il MOSFET rimane interdetto, lasciando che sia il resistore di pull-up R3 a mantenere a 1 logico (ma a 5 volt) la linea SDA che porta al pin-strip.

Se invece il bus viene impostato per interfacciarsi con dispositivi a 3,3V, le considerazioni sono comunque analoghe perché il traslatore comunque funziona similmente, con la sola differenza che quando il contatto SDA sul lato strip si trova a livello alto, drain e source sono equipotenziali e i MOSFET non intervengono. Notate che per ragioni di uniformità con le altre della serie, non da meno quelle proposte negli ultimi tempi su Eletttronica In (per esempio quella con ADXL372 proposta nel fascicolo di settembre 2022 e quella con il sensore UVC proposta nel fascicolo di ottobre scorso) la piedinatura di questa breakout board è standard, pertanto nello schema elettrico e nel PCB trovate alcuni contatti che nella realtà non sono utilizzati



in questa breakout. I contatti rispondono a una standardizzazione che abbiamo voluto per tutta la serie di breakout board proposte e da proporre, allo scopo di unificare le connessioni e semplificare l'implementazione di applicazioni.

La breakout board connessa ad Arduino Uno.

REALIZZAZIONE PRATICA

Bene, ora che abbiamo descritto lo schema elettrico nei suoi particolari, possiamo passare alle note costruttive, ossia a spiegare come realizzare la breakout board; successivamente vi proporremo un esempio applicativo basato su Arduino, che consente di valutare bene le possibilità di utilizzo della scheda.

Come di consueto, abbiamo disegnato un circuito stampato del quale vi rendiamo disponibili per il download le due tracce (si tratta di un doppia faccia) dal nostro sito web www.elettronica.in; da esse potete procedere a preparare il PCB per fotoincisione, dopodiché una volta che lo avrete inciso e forato potete procedere al montaggio dei pochi componenti occorrenti. Trattandosi di componenti in SMT, è richiesto un minimo di attrezzatura ed occorre prestare anche un po' di attenzione ed avere manualità; l'attrezzatura del caso consta di un saldatore a punta molto fine, di filo di lega saldante (o pasta salda) del diametro massimo di 0,5 mm, della pasta fluxante a media densità, una lente d'ingrandimento e una pinzetta per il posizionamento dei componenti. Una volta procurato il tutto, prendete il circuito

↓ Listato 1

```
#include <Wire.h>

int MSB = 0;          //byte alto del dato temperatura
int LSB = 0;          //byte basso del dato temperatura
uint16_t TEMP = 0;    //dato temperatura composto da MSB + LSB
float TEMPC = 0.0;    //temperatura convertita in gradi Celsius

void setup()
{
  Wire.begin(0x48);    // connessione I2C all'indirizzo 0x48 (A0=0 - A1=0)
  Serial.begin(9600);  // velocità per debug seriale a 9600bps
}

void loop() {
  Wire.requestFrom(0x48, 2);    //richiesta di 2 byte al sensore
  while (Wire.available())
  {
    MSB = Wire.read(); //ricezione primo byte (parte alta temperatura)
    LSB = Wire.read(); //ricezione secondo byte (parte bassa temperatura)
  }
  TEMP = MSB;
  TEMP = TEMP << 8;          //shift di 8 bit del byte per ottenere un dato a 16bit
  TEMP = TEMP + LSB;         //sommo il dato LSB a quello da 16bit così da ottenere MSB + LSB
  if ((TEMP >> 15) == 1) //se la temperatura è negativa la converto per ottenere il dato negativo
  {
    //Temperatura Negativa
    TEMPC = TEMP - 65536;
    TEMPC = TEMPC /128.0;
  }
  else
    //se la temperatura è positiva ne ottengo il dato Celsius
  {
    //Temperatura Positiva
    TEMPC = TEMP / 128.0;
  }
  //Visualizzo su seriale la temperatura risultante ogni secondo
  Serial.print("Temp: ");
  Serial.print(TEMPC);
  Serial.println("°C");
  delay(1000);
}
```

stampato, spalmate con la pasta flussante le piazzole destinate ai componenti SMD e saldate per prime le resistenze e i condensatori, poi il LED e i MOSFET BSS138W-7-F; questi ultimi possono essere saldati solo nel verso giusto, per l'univoca disposizione dei terminali, mentre per il condensatore elettrolitico e il LED dovete fare riferimento all'orientamento indicato nel piano di montaggio che trovate in queste pagine. Per quanto riguarda il LED, cercate di scaldarlo il meno possibile, perché altrimenti è facile che si deformi la finestrella in resina trasparente da cui esce la luce.

Infine inserite nei rispettivi fori il pin strip ad angolo retto a 12 poli e saldatelo, tenendolo ben allineato alla superficie del PCB. Il pin-strip è utile per realizzare le connessioni con Arduino tramite i classici jumper a filo maschio/femmina, ma potete anche farne a meno, se volete saldare direttamente i fili

sulle piazzole laterali che riportano le connessioni del circuito.

Concludiamo la descrizione pratica del circuito con una considerazione sui jumper di selezione della tensione sul bus, JVI0 e per l'address I²C-Bus: per mantenere la breakout board con il più basso profilo possibile, abbiamo previsto che il ponticello non sia il classico formato da un pin-strip 3 vie a passo 2,54 ma che sia ottenuto stagnando piazzole attigue, quindi l'impostazione va eseguita durante il montaggio, al termine della saldatura dei componenti.

La disposizione dei jumper suddetti è ben visibile nella **Fig. 2**, nella quale si vede che sono realizzati con un cortocircuito ottenuto colando una goccia di stagno; nello specifico, quelli degli indirizzi sono chiusi fra il polo centrale e il contatto 0, ad impostare.



BIT DI SERVIZIO							INDIRIZZO ESADECIMALE
A6	A5	A4	A3	A2	A1	A0	
1	0	0	1	0	0	0	0x48
1	0	0	1	0	0	1	0x49
1	0	0	1	0	1	0	0x4A
1	0	0	1	0	1	1	0x4B

MISURIAMO LA TEMPERATURA CON ARDUINO

La breakout board descritta in queste pagine nasce per essere interfacciata a un microcontrollore, giacché l'informazione sulla temperatura rilevata dall'ADT7422 viene fornita sotto forma di stringhe di dati su I²C-Bus; siccome Arduino supporta questo tipo di bus, abbiamo realizzato un esempio di codice per leggere la temperatura tramite una scheda Arduino ed in particolare la popolare e collaudatissima Arduino UNO rev.3.

Rendiamo disponibile il codice di esempio sul nostro sito www.elettronica.in.it affinché possiate scaricarlo e scriverlo nella vostra scheda tramite l'IDE. Questo sketch non sfrutta alcuna libreria ad hoc, bensì si basa unicamente sulla libreria Wire (wire.h) per dialogare su I²C con il sensore; tale libreria viene inclusa nella parte iniziale del codice. Nel **Listato 1** trovate l'intero codice per leggere la temperatura dal sensore.

Nel firmware di esempio viene richiesto l'invio di due byte (MSB e LSB) al sensore all'indirizzo esadecimale 0x48 (impostato attraverso i jumper A0 e A1), quindi da questi due singoli byte viene composto dall'ADT7422 un dato rappresentante la temperatura in gradi Celsius, dato che poi viene mostrato sulla seriale di Arduino.

L'interrogazione del sensore mediante i due byte e la lettura della risposta che ne deriva, avvengono ciclicamente ogni secondo, pertanto l'ADT7422 viene impostato per implementare un campionamento al secondo.

Per il corretto funzionamento dello sketch, ricordate quindi di impostare correttamente i jumper A0 e A1, perché se al sensore non venisse assegnato l'indirizzo 0x48, verrebbe mostrato un messaggio di errore sulla seriale; l'address 0x48 si assegna con la combinazione 00 di A0 e A1, quindi chiudendo a massa entrambi tali jumper (riferitevi, allo scopo, alla già citata **Fig. 2**).

Per la corretta impostazione dell'indirizzo I²C-Bus della breakout board fate riferimento alla **Tabella 1**,

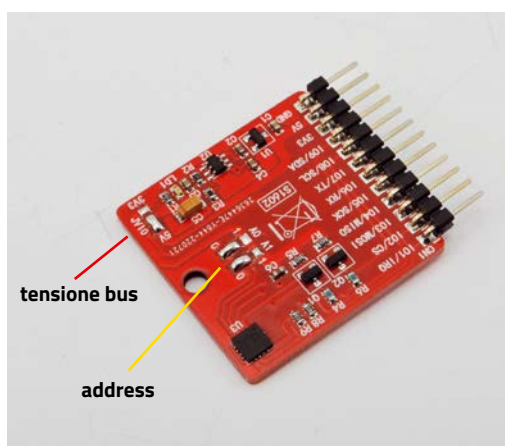


Tabella 1
Indirizzi I²C-Bus
per il chip ADT7422.

Fig. 2
I ponticelli
sono chiusi
con una
goccia di
stagno.

che riporta i quattro possibili indirizzi attribuibili all'integrato ADT7422 riferiti ai due jumper A0 e A1, e i corrispondenti valori esadecimali.

CONCLUSIONI

Bene, termina qui la trattazione della breakout board per la misura della temperatura tramite il precisissimo integrato sensore ADT7422 dell'Analog Devices; sulla base dell'esempio firmware appena proposto e dettagliato, potete sviluppare un vostro codice, anche non necessariamente per Arduino, con il quale ad esempio visualizzare la temperatura su un piccolo display. Buon lavoro! ▣



Cosa occorre?

La breakout board con ADT7422 (cod. FT1602M) è disponibile presso Futura Elettronica con tutti i componenti SMD preassemblati a Euro 25,00. Il prezzo si intende IVA compresa.

Il materiale va richiesto a:

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

FUTURA GROUP
IDEE IN ELETTRONICA

I prodotti più recenti, i più venduti, le ultime novità dal mondo dell'elettronica, i progetti più interessanti, i corsi e i webinar.

FUTURA ELETTRONICA
SHOP ON LINE

Oltre 6.000 prodotti, prezzi sempre aggiornati, disponibilità in tempo reale, aggiornamenti quotidiani su nuovi prodotti, schede tecniche e immagini dettagliate, centinaia di video dimostrativi.

Elettronica In
PROFESSIONAL
Elettronica In
ELETTRONICA PROFESSIONAL

Elettronica In ed Elettronica In Professional, le più importanti riviste del settore con centinaia di progetti e news dal mondo dell'elettronica.

FUTURA GROUP
PROGETTI

Progetti pratici, articoli didattici per chi si avvicina al mondo dell'elettronica, corsi hardware/software e molto altro ancora!

Futura Academy
CORSI DI FORMAZIONE

Corsi in presenza ed online organizzati con livelli base per i principianti fino a livelli avanzati per i più esperti. Vasta scelta di argomenti, dall'elettronica all'informatica, tecnologie innovative e molti altri ancora.

FUTURA ELETTRONICA
PROGETTAZIONE

Un team di esperti in grado di studiare, progettare e realizzare prototipi utilizzando le ultime tecnologie. 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.

FUTURA ELETTRONICA
RIVENDITORI

Oltre 150 punti vendita in tutta Italia dove acquistare i prodotti Futura Elettronica. Personale preparato e disponibile in grado di trovare la soluzione più adatta alle diverse esigenze.

FUTURA GROUP
IDEE IN ELETTRONICA

Categorie ▾

HOME

SHOP

RIVISTE

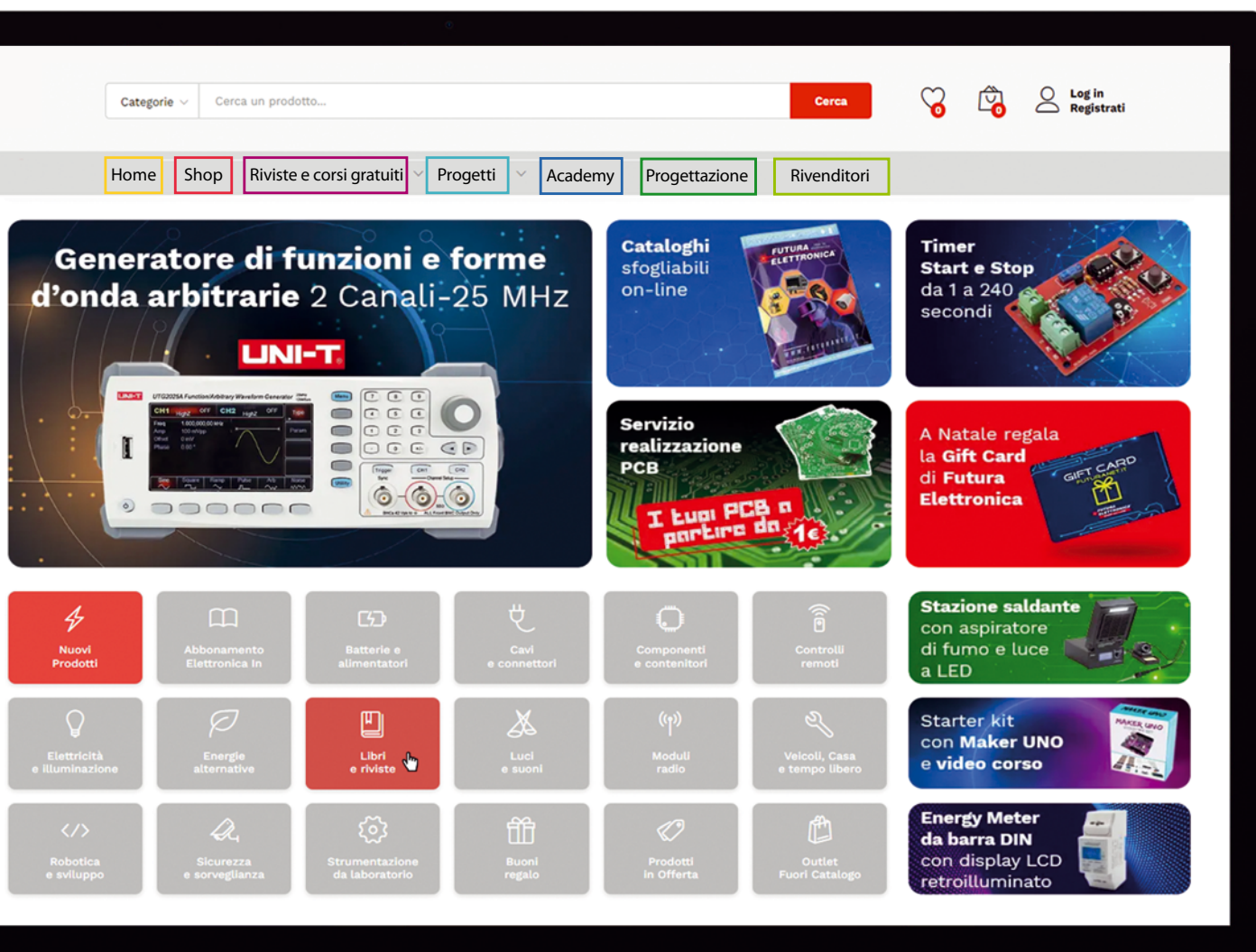
PROGETTI

ACADEMY

PROGETTAZIONE

RIVENDITORI

È on-line il NUOVO PORTALE **FUTURANET.IT**





**Gioco di abilità
dove vince chi
riesce a premere
il pulsante nel
momento in cui il
punto luminoso
scorrevole
formato da un
LED rosso passa
nella finestrella
fra due gialli
in un anello
NeoPixel.**



FERMA IL PUNTO A LED!

di ROBERTO TESTI

Giochi elettronici ce ne sono tantissimi, per tutti i gusti e complessità: da quelli su PC alle console che ne supportano moltissimi, fino a dispositivi come quello proposto in queste pagine, che è tutto basato sull'elettronica (in verità, anche su un piccolo firmware) e che consiste in una prova individuale di abilità e di riflessi: un punto luminoso percorre un anello a LED RGB e quando entra in una finestrella formata da due LED accesi a luce gialla fissa tra i quali c'è posto per il punto, bisogna premere un pulsante per fermarlo in quel punto esatto. Chi ci riesce vince la manche e vedrà accendersi l'intero cerchio a luce verde. Per rendere la cosa meno difficile (vedrete in pratica che alla velocità di scorrimento non è semplicissimo colpire il centro...) abbiamo previsto che ci si possa aggiudicare la manche anche

se il pulsante viene premuto quando il punto luminoso rosso raggiunge uno dei LED accesi di giallo che delimitano la zona bersaglio.

IL CIRCUITO

Andiamo dunque a spiegare come è costituito questo gioco, che nelle foto in queste pagine vedete concretizzato in una struttura in plastica stampata in 3D, con un anello che richiama quello del popolare film Stargate del 1994, ed un pulsante posto su un supporto solidale con esso; l'involucro che vedete, composto da una parte superiore e un coperchio inferiore, è stato stampato con una stampante 3D e di esso vi rendiamo disponibili per il download, insieme ai file di questo progetto, i file .STL per la stampa. L'hardware del nostro gioco di abilità è incentrato su una scheda Arduino e, per l'esattezza, la diffusissima Arduino UNO, che qui utilizziamo nella rev.3, che è la più reperibile; tramite una linea di

Fig. 1
Il pulsante scelto
per il progetto.



I/O digitale, la D11, essa controlla tramite stringhe dati seriali secondo l'opportuno protocollo, un anello di 60 LED NeoPixel (reperibile presso Futura Elettronica, on-line su www.futurashop.it, codice prodotto RING60NEOPIXEL) sul quale scorrerà il punto luminoso.

Tramite un secondo pin digitale, D6, Arduino UNO controlla una strip NeoPixel da 8 LED (Stick Neo-pixel con 8 LED RGB WS2812 e driver integrato, reperibile anch'essa presso Futura Elettronica, on-line su www.futurashop.it, codice prodotto STICK8NEOPIXEL), la cui funzione è segnalare l'inizio della manche e la vittoria, nonché il livello di difficoltà, che incrementa da 1 a 8 ogni volta che si riesce a centrare il bersaglio.

La striscia di 8 LED RGB funziona similmente all'anello, quindi i suoi 8 LED RGB possono essere gestiti individualmente e ciascuno può produrre 256 tonalità di colore.

Il canale dati per la comunicazione è seriale di tipo One-Wire e può essere controllato con microcontrollori come Arduino, PIC, mbed, ecc.

Ogni stick può essere collegato in cascata per realizzare vari effetti.

Un terzo pin di I/O digitale legge lo stato del pulsante, con il quale il giocatore interagisce con il sistema; si tratta di un pulsante di tipo per videogiochi, giacché serve qualcosa che abbia una discreta robustezza e una superficie adatta ad essere premuto senza incertezza.

Nel nostro prototipo abbiamo adottato un pulsante per giochi Arcade, commercializzato dalla Futura Elettronica con il codice prodotto COM09336 (**Fig. 1**) composto da un corpo in plastica a vite per fissaggio con l'apposita ghiera, molto simile ai pulsanti utilizzati nei giochi arcade, dove il pulsante scorrevole agisce su un microswitch a 3 terminali (comune, normalmente aperto e normalmente chiuso) in grado di commutare 3 A a 125 Vca, requisiti, questi, più che abbondanti perché nel

nostro circuito deve commutare 5 volt e la debole corrente di un pin di Arduino con pull-up.

L'anello NeoPixel conta 60 LED RGB NeoPixel (di tipo 5050 con chip WS2812B) appunto, che possono essere gestiti individualmente.

È possibile impostare a piacimento la frequenza di refresh, in modo da rendere impercettibili determinati giochi di luce.

L'anello viene venduto in una confezione diviso in 4 archi da 15 LED ciascuno, che devono essere saldati tra loro per creare il cerchio da 60 LED (**Fig. 2**). Arduino controlla molto semplicemente le strip o gli anelli NeoPixel semplicemente utilizzando una specifica libreria per Arduino resa disponibile da Adafruit, che viene inclusa all'inizio dello sketch. La stessa libreria viene richiamata nel controllo della strip a 8 LED.

La tecnologia Neopixel permette di realizzare LED RGB "intelligenti" con controller a bordo, facilmente integrabili nell'ambiente Arduino grazie a librerie proprietarie che Adafruit (www.adafruit.com) rende disponibili liberamente.

Una particolarità dei LED Neopixel è che possono essere collegati in cascata in modo che la linea dati da uno passi al successivo, però il prezzo da pagare è che oltre un certo numero di LED, la velocità di gestione deve ridursi sensibilmente; a causa di ciò, se si devono realizzare matrici per mostrare della grafica veloce occorre impiegare molte linee con pochi LED ciascuna.

L'alimentazione prevista per i LED Neopixel è a 5 volt; la comunicazione avviene a un massimo di 800 kbps.

Il protocollo di comando del sistema NeoPixel prevede l'invio di gruppi di tre byte in una stringa di 24 bit, ognuna delle quali contiene lo stato di illuminazione di ciascun colore base.

Siccome un gioco elettronico è bello se oltre che effetti visivi produce effetti sonori, abbiamo pensato di aggiungere un trasduttore sonoro, anche questo gestito da Arduino UNO per fargli generare delle musicchette nei passaggi più importanti e all'inizio del gioco; si tratta di un cicalino piezoelettrico senza elettronica, che è quindi una pastiglia piezoelettrica pura a 5 volt, pilotata da Arduino tramite il pin di I/O D9, associato al PWM e comandato di volta in volta per costruire delle note con la tecnica PWM, ricorrendo (come vedremo dallo sketch) all'istruzione Tone.

L'intera elettronica del nostro gioco è alimentata tramite il jack DC di Arduino UNO, scheda dal cui pin 5V si preleva l'alimentazione (riferita ai pin GND) per le due unità NeoPixel.

SVOLGIMENTO DEL GIOCO

Vediamo dunque come funziona in pratica questo nostro gioco di abilità, analizzando le varie fasi a partire dall'accensione: una volta alimentata la scheda Arduino UNO, questa inizializza gli I/O e dopo aver caricato le librerie comanda i LED NeoPixel distribuiti sull'anello e sulla barra per implementare due effetti, uno sull'anello e l'altro sulla barra; in entrambi i casi si tratta di comporre un arcobaleno, ossia l'iride, partendo dal rosso e arrivando al violetto. Tutti i LED permangono in questo stato finché non viene premuto il pulsante, allorché si avvia il gioco, con una fase introduttiva che vede due eventi:

- la riproduzione di una musicchetta all'avvio, che inizia quando iniziano a spegnersi, in sequenza, i LED dell'anello e si arresta allo spegnimento dell'ultimo;
- lo spegnimento uno alla volta dei LED dell'anello.

Al termine della musicchetta e allo spegnimento dell'ultimo LED NeoPixel dell'anello, si spengono uno alla volta anche i LED della piccola barra di 8 LED RGB e resta acceso solo il primo, a luce verde, indicante che partiamo dal livello di difficoltà più basso (indicato nel codice con *gameState* == 0). La porzione di codice che provvede a ciò è la seguente:

```
void loop() {  
    FastLED.setBrightness(BRIGHTNESS );  
  
    if (gameState == 0) {  
        fill_rainbow(leds, NUM_LEDS, 0, 20);  
        //2 = longer gradient strip  
        fill_rainbow(sleds, SCORE_LEDS, 0, 40);  
        //2 = longer gradient strip
```

Adesso siamo pronti a giocare, con la premessa che il gioco prevede 8 livelli di difficoltà. Nel primo, quello da cui si parte, un punto luminoso rosso scorre lungo l'anello (ciò viene ottenuto comandando l'accensione di rosso, quindi con RGB = 255, 0, 0) di uno dei 60 LED dell'anello a NeoPixel e tre LED rimangono accesi a luce fissa: i due estremi sono gialli e rappresentano la finestra (target) entro cui possiamo fermare il punto premendo il pulsante, mentre il centrale è acceso di verde. Ci si aggiudica la manche se si riesce a premere il pulsante quando il punto scorrevole rosso è in corrispondenza di uno di questi tre LED, che al suo passaggio diviene rosso anch'esso. Se si riesce nell'intento, il LED dove fermiamo



Fig. 2
L'anello a LED NeoPixel è formato da quattro archi da saldare insieme.

la corsa diviene rosso, sia che si tratti di quello centrale (nei primi due livelli di difficoltà) sia che la pressione del pulsante avvenga in corrispondenza del passaggio su uno dei LED laterali della finestra. Il cicalino emette quindi 3 beep (lo farà ad ogni centro ottenuto) e contestualmente avverrà il lampeggio dell'intero anello di 60 LED, a luce verde, cui seguirà la presentazione del livello successivo e quindi dei soli LED target e del punto scorrevole. Queste due segnalazioni avvengono dopo la pressione del pulsante a manche in corso e se la pressione del pulsante è avvenuta quando il punto passa all'interno della finestra di tre LED RGB, si passa al livello di difficoltà superiore, cosa, questa, che viene evidenziata dall'accensione a luce fissa del secondo LED (oltre che del primo) della barra di 8 NeoPixel.

La relativa porzione di sketch è la seguente, dove l'aumento del livello di difficoltà corrisponde a *gameState* == 1:

```
if (digitalRead(BUTTON_PIN) == LOW) {  
    Position = 0;  
    findRandom = true;  
    delay(500);  
    for (byte i = 0; i < NUM_LEDS; i++) {  
        leds[i].setRGB(0, 0, 0);  
        delay(40);  
        FastLED.show();  
        int thisPitch = map (i, 60, 0, 100, 1500);  
        tone(BUZZER, thisPitch, 120);  
    }  
  
    for (byte i = 0; i < SCORE_LEDS; i++) {  
        sleds[i].setRGB(0, 0, 0);  
        delay(100);  
        FastLED.show();  
    }  
}
```

```

    gameState = 1;
  }
  FastLED.show();
}

```

Ora inizia una nuova manche dove il punto luminoso scorrerà più velocemente: se riusciremo a premere il pulsante otterremo una serie di segnalazioni identica in tutto e per tutto al caso precedente, ossia la serie di tre beep emessi dal cicalino e il lampeggio dei LED, ma stavolta la barra di 8 LED RGB accenderà anche il terzo LED per avvisarci che passiamo al terzo livello di difficoltà (indicato nel codice con *gamestate* = 2).

A qualsiasi livello sia il gioco, in caso di pressione del pulsante quando il punto rosso è fuori dalla finestra target (o dal LED target) e perciò in caso di errore, la partita in corso ritorna al livello iniziale e bisognerà ricominciare da capo, quindi se non si centra il bersaglio, ai tre beep si accompagna il lampeggio dell'intero anello di 60 LED a luce rossa (e non verde, come quando centriamo il target...) e poi l'anello di LED NeoPixel ripropone i colori della fase iniziale a luce fissa, in attesa che premiamo il pulsante per ricominciare a giocare.

Dal terzo livello di difficoltà (*gameState* = 2) non c'è più la finestra di tre LED ma è acceso un solo LED verde fisso indicante il punto in cui dovrà trovarsi il punto scorrevole rosso quando dovremo premere il pulsante: il gioco diventa quindi un po' più difficile perché ci sarà un solo punto su 60 in cui la pressione sarà valida.

Come al solito, se centreremo il target otterremo i tre beep e lampeggi dell'anello a luce verde, con

conseguente passaggio al livello superiore, mentre in caso di errore ci saranno i tre beep e contemporanei lampeggi a luce rossa dei LED dell'anello, con retrocessione al primo livello di gioco.

Via-via aumenterà la difficoltà, non soltanto perché ci sarà sempre un solo punto verde da centrare, ma anche perché il punto luminoso rosso scorrerà più rapidamente.

A proposito di velocità, la riga d'istruzione che imposterà la pausa nel passaggio dallo spegnimento di un LED dell'anello all'accensione del successivo è la seguente:

```

const byte ledSpeed[8] =
{100, 90, 80, 70, 55, 40, 20, 10};
//const byte ledSpeed[8] =
{100, 100, 150, 150, 150, 150, 150, 150};
//low speed for test

```

In realtà sono due, da scegliere in alternativa a una all'altra, semplicemente decommentando o commentando l'altra; i numeri tra parentesi indicano la durata della pausa, quindi tanto sono più alti, quanto più lento è lo scorrimento e viceversa. Ogni parentesi graffa riporta otto valori, tanti quanti sono i livelli di difficoltà. Come notate, dalla prima all'ultima posizione il valore decrementa, quindi il gioco accelera.

La riga di codice attualmente eseguita è quella decommentata; se volete un gioco più lento, potete commentarla e decommentare quella subito sotto, ovvero intervenire manualmente sui tempi descritti nella prima riga.

Una volta raggiunta l'ottava manche vinta, ossia



Ogni volta che si vince una manche i LED diventano verdi; se si perde si colorano di rosso.



Listato 1

```
#include "FastLED.h"
#define NUM_LEDS 60
#define DATA_PIN 11
#define SCORE_PIN 6
#define SCORE_LEDS 8
#define BUTTON_PIN 2
#define BUZZER 9
#define BRIGHTNESS 55
CRGB leds[NUM_LEDS];
CRGB slds[NUM_LEDS];

bool reachedEnd = false;
byte gameState = 0;
//byte ledSpeed = 0;
int period = 1000;
unsigned long time_now = 0;
byte Position = 0;
byte level = 0;
const byte ledSpeed[8] = {100, 90, 80, 70, 55, 40, 20, 10};
//const byte ledSpeed[8] = {100, 100, 150, 150, 150, 150, 150, 150};
bool findRandom = false;
byte spot = 0;

void setup() {
  FastLED.addLeds<WS2812B, DATA_PIN, GRB>(leds, NUM_LEDS);
  FastLED.addLeds<WS2812B, SCORE_PIN, GRB>(slds, SCORE_LEDS);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
  Serial.begin(9600);
  Serial.println("Reset");
}

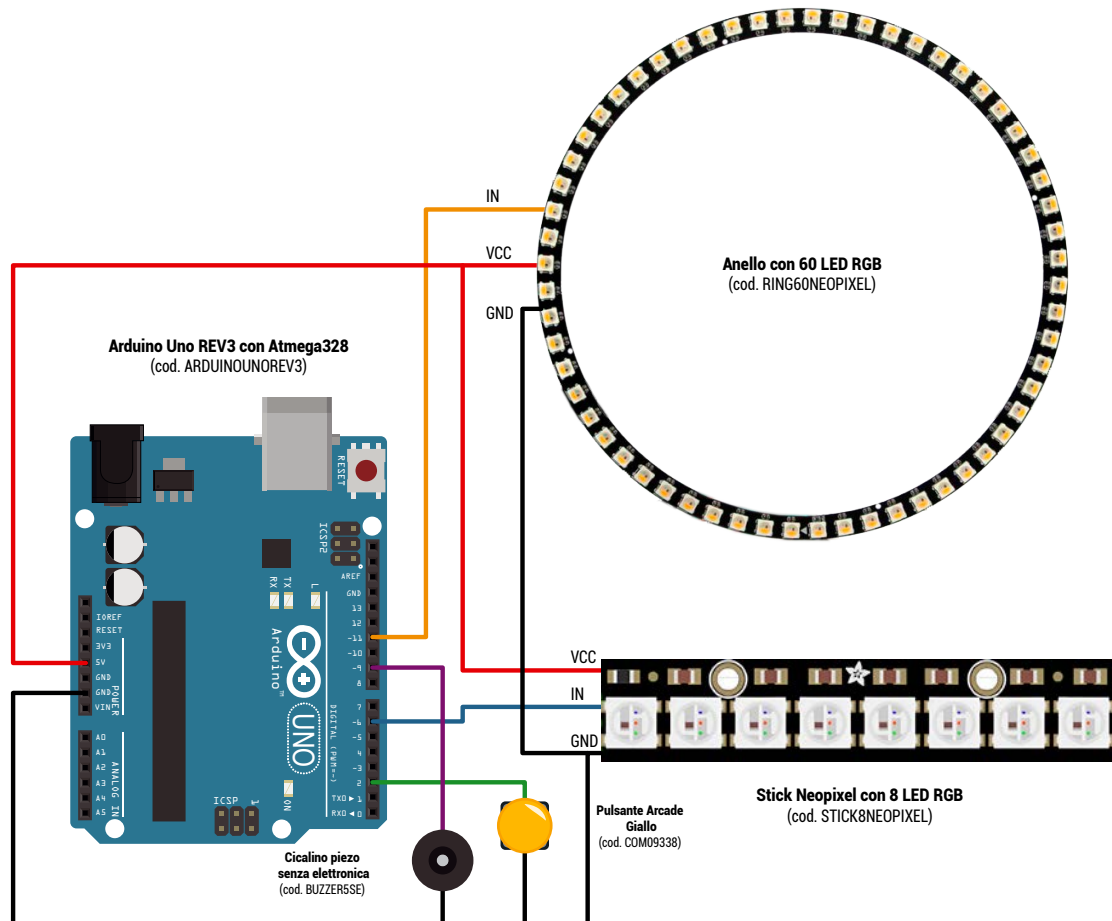
void loop() {
  FastLED.setBrightness(BRIGHTNESS );

  if (gameState == 0) {

    fill_rainbow(leds, NUM_LEDS, 0, 20); //2 = longer gradient strip
    fill_rainbow(slds, SCORE_LEDS, 0, 40); //2 = longer gradient strip

    if (digitalRead(BUTTON_PIN) == LOW) {
      Position = 0;
      findRandom = true;
      delay(500);
      for (byte i = 0; i < NUM_LEDS; i++) {
        leds[i].setRGB(0, 0, 0);
        delay(40);
        FastLED.show();
        int thisPitch = map(i, 60, 0, 100, 1500);
        tone(BUZZER, thisPitch, 120);
      }
      for (byte i = 0; i < SCORE_LEDS; i++) {
        slds[i].setRGB(0, 0, 0);
        delay(100);
        FastLED.show();
      }
      gameState = 1;
    }
    FastLED.show();
  }

  if (gameState == 1) {
    period = ledSpeed[0];
    if (millis() > time_now + period) {
      time_now = millis();
      if (findRandom) {
        spot = random(56) + 3;
        findRandom = false;
      }
      leds[spot - 1].setRGB(255, 140, 0);
      leds[spot].setRGB(0, 255, 0);
      leds[spot + 1].setRGB(255, 110, 0);
      slds[0].setRGB(0, 255, 0);
      PlayGame(spot - 1, spot + 1);
    }
    if (digitalRead(BUTTON_PIN) == LOW) {
      delay(300);
      findRandom = false;
      if (Position > spot - 1 && Position < spot + 3) {
        level = gameState;
        gameState = 98;
      } else {
        gameState = 99;
      }
    }
  }
}
```



superato il più alto livello, il gioco riprenderà dal principio; per l'esattezza, centrando il target all'ottavo livello l'anello pulsa di verde e il cicalino emette una musicchetta diversa da quelle riprodotte prima, per festeggiare il vincitore, dopodiché torna l'arcobaleno sull'anello NeoPixel e la barra da 8 LED ripropone la segnalazione del primo livello. La porzione di sketch che gestisce l'ultimo livello di difficoltà è la seguente:

```
if (gameState == 8) {
  period = ledSpeed[7];

  if (millis() > time_now + period) {
    time_now = millis();
    if (findRandom) {
      spot = random(56) + 3;
      findRandom = false;
    }

    leds[spot].setRGB(0, 255, 0);
    sleds[7].setRGB(255, 250, 255);
    PlayGame(spot, spot);
  }
}
```

```
if (digitalRead(BUTTON_PIN) == LOW) {
  delay(300);
  if (Position == spot+1) {
    level = gameState;
    gameState = 98;
  } else {
    gameState = 99;
  }
}

if (gameState == 98) {
  winner();
}

if (gameState == 99) {
  loser();
}
}
```

Come vedete, *gameState == 98* corrisponde alla vittoria dell'intera partita, mentre *gameState == 99* all'errore e al ritorno forzato al primo livello di difficoltà (insomma, si riprende la partita dall'inizio). Le operazioni eseguite in caso di vittoria (supe-

ramento di tutti i livelli di difficoltà) sono quelle definite da questa porzione di codice:

```
void winner() {  
  for (byte i = 0; i < 3; i++) {  
    for (byte j = 0; j < NUM_LEDS; j++) {  
      leds[j].setRGB(0, 255, 0);  
      tone(BUZZER, 1000, 250);  
    }  
    FastLED.show();  
    delay(500);  
    clearLEDS();  
    FastLED.show();  
    delay(500);  
  }  
}
```

Bene, con questo abbiamo esaurito la descrizione del funzionamento del nostro gioco elettronico e delle parti più significative dello sketch. Nel **Listato 1** trovate la parte iniziale dello sketch. Passiamo adesso alle note costruttive, poche e semplici perché si tratta di fare "due" saldature per assemblare l'anello NeoPixel e collegare sia esso che i pochi elementi circuitali ad Arduino UNO mediante cavetti jumper da saldare su un lato e innestare dall'altro.

REALIZZAZIONE PRATICA

La prima cosa da fare è dunque preparare l'elettronica e collaudarla, dopodiché occorrerà scegliere e realizzare un contenitore e supporto idoneo ad ospitarla; per l'elettronica dovete procurarvi una scheda Arduino UNO e collegarla alla barra Stick NeoPixel a 8 LED, all'anello da 60 LED e al pulsante, nonché al cicalino piezo, come indicato nello schema di cablaggio visibile in queste pagine. Per il cicalino non è necessario rispettare una determinata polarità, dal momento che non ha un'elettronica interna, ma è invece indispensabile collegare come indicato le unità a NeoPixel, per le quali sono previsti tre contatti: +5V, GND e DI (linea dati). Il pulsante collegatelo come preferite, mediante sottili spezzoni di filo di rame in guaina, da 0,2÷0,5 mmq di sezione e lo stesso dicasi per il cicalino, che conviene connettere ai pin D9 e GND di Arduino UNO mediante dei jumper femmina-femmina. Prima di alimentare il tutto verificate che i collegamenti siano come indicato, quindi dovete passare a programmare il tutto, perché altrimenti una volta alimentato rimarrà inerte; per la programmazione di Arduino UNO (ossia per caricare lo sketch che implementerà il funzionamento del gioco) accendete un Personal Computer nel quale è installato l'IDE di Arduino collegate la vostra Arduino UNO

 **Fig. 3**
Il prototipo inserito nel suo contenitore in tre parti.

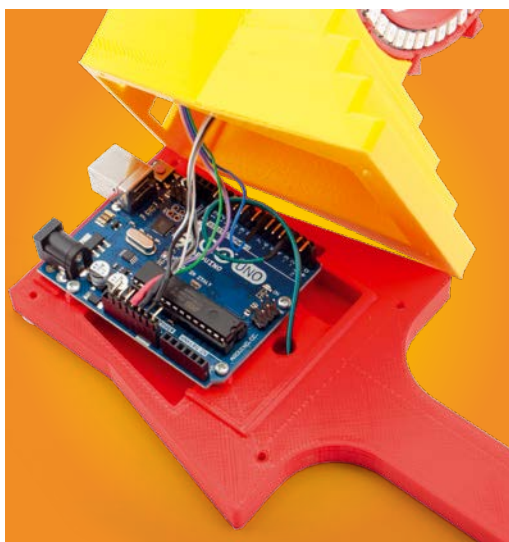


con un cavo USB A/B al computer stesso (ad una porta USB libera...) e dopo aver scaricato nel disco rigido del PC lo sketch (file ad estensione .ino) dalla sezione Download della pagina di presentazione di questo numero della rivista, cliccate su *File>Apri* e cercate il file .ino nel percorso in cui l'avete salvato, poi una volta che lo avete aperto andate nel menù *Strumenti* e cliccate su Scheda: e, dal sotto-menù che si apre, cliccate su Arduino UNO. A questo punto dovete caricare il firmware nella



 **Fig. 4**
Dal retro della base si accede ai connettori USB e jack di Arduino UNO.

➔ **Fig. 5**
Sul fondo del
coperchio è
applicata la scheda
Arduino UNO.



➔ **Fig. 6**
L'effetto visivo
presentato
all'accensione
del circuito.

memoria di programma di Arduino e per farlo dovete semplicemente cliccare sul pulsante con la freccia orizzontale (il secondo da sinistra nella toolbar, sotto la barra dei menù) e attendere il messaggio di caricamento completato fornito a video dall'IDE Arduino.

Sconnettete quindi il cavo USB da Arduino UNO e spegnete il PC, che ormai non vi servirà più, almeno per questo scopo.

Fatto ciò prendete un alimentatore da rete di quelli con spina incorporata (i comunissimi wall-cube...) capace di erogare una tensione stabilizzata di 7÷9 volt e una corrente continua di almeno 800 mA, dotato di cavo terminante con uno spinotto coassiale con foro interno di diametro adatto al jack DC di Arduino UNO, quindi alimentate il tutto (accertandovi di avere sconnesso il cavo USB per evitare un conflitto tra le alimentazioni) e poi verificate che l'anello di LED esegua il gioco di luci iniziale.

Ora siete pronti a giocare, ovvero sareste, perché prima dovete pensare a un ancoraggio meccanico per il pulsante e il resto dell'elettronica, che faccia anche da contenitore.

LA PARTE MECCANICA

Un suggerimento può derivare dalle foto del prototipo che vedete in queste pagine, la cui struttura è stata stampata in 3D con del filamento di PLA di due colori: consta di una base a forma di gradinata in PLA giallo che ospita la barra di 8 LED RGB e sorregge l'anello in PLA rosso che supporta l'anello di 60 LED NeoPixel, ed è a sua volta appoggiata a una base allungata (anche questa in PLA rosso) che funziona da coperchio di fondo e incorpora il supporto del pulsante e che idealmente fa da raccordo fra il pulsante e il visualizzatore.

Il tutto risulta come nelle foto che vedete in queste pagine ed in particolare nella **Fig. 3**.

La base a scalinata dell'anello NeoPixel in PLA giallo che vedete in fotografia, è cava e contiene la scheda Arduino UNO, oltre al cicalino piezoelettrico e ai fili di collegamento che portano all'anello e alla barra NeoPixel, quest'ultima fissata in una cava ricavata frontalmente.

Sul retro della base sono state previste due aperture per poter inserire lo spinotto jack di alimentazione e il connettore USB del cavo da utilizzare per la programmazione (**Fig. 4**); a tale riguardo facciamo notare che sebbene sia possibile alimentare tutto dal connettore USB di Arduino UNO, perché i 5V



necessari ai LED NeoPixel arriverebbero appunto dal pin 5V della scheda, nella pratica non è l'ideale perché quando sono tutti accesi, i LED assorbono una discreta corrente.

Comunque nella buona parte delle condizioni, l'alimentazione da USB è possibile.

Se optate per l'alimentazione da rete, dovete utilizzare un alimentatore in continua che non superi i 9V, per non surriscaldare il regolatore interno alla scheda Arduino UNO che fornisce i 5 volt ai pin 5V, giacché la potenza dissipata dipende strettamente dalla caduta di tensione ingresso-uscita.

Nella nostra soluzione Arduino UNO è fissata al fondo o coperchio della base, mediante viti autofilettanti da 3 mm, come si vede a sinistra nella **Fig. 5**, mentre a destra nella stessa vedete come è stato fissato e cablato il pulsante nel suo supporto: in pratica lo abbiamo introdotto da sotto e avvitato con l'apposita flangia a vite.

Naturalmente siete liberi di realizzare la parte meccanica in altro modo, per esempio utilizzando un disco di compensato sottile (spesso 5 mm) per incollarvi, mediante della colla a caldo, l'anello NeoPixel, fissando poi il disco con dei listelli in legno incollati ad angolo retto per applicarvi sotto la barra di 8 LED in verticale (allineata all'anello) e più distante, una scatoletta tonda in plastica per elettronica o per impianti elettrici (ma va bene anche un piccolo contenitore cilindrico per alimentari) che ospiterà il pulsante.

Le possibilità sono molteplici e il progetto non è affatto critico sotto l'aspetto pratico, né tantomeno per quel che riguarda la disposizione e il collegamento dei componenti elettrici ed elettronici. Comunque abbiate realizzato l'insieme, prima di metterlo sotto tensione verificate bene i collegamenti dell'elettronica e a quel punto potete alimentarlo: se tutto funzionerà correttamente, vedrete illuminarsi sia la barra di 8 LED RGB, sia tutti i 60 LED dell'anello NeoPixel e tutti dovranno rimanere accesi, proponendo un effetto come quello che vedete nella **Fig. 6**, fino a quando verrà premuto il pulsante di gioco.

CONCLUSIONI

Bene, con queste ultime note costruttive possiamo ritenere di aver concluso: non vi resta che perfezionare e personalizzare il vostro gioco, fissarlo ad un tavolo e dare inizio alle gare con amici e conoscenti, per vedere chi ha più riflessi.

La personalizzazione può riguardare anche il firmware, ad esempio per modificare la velocità di scorrimento dei punti luminosi lungo l'anello

Guarda il Gioco Neopixel in azione su YouTube!



Vuoi vedere in azione il nostro Gioco Neopixel? Nulla di più semplice: collegati alla pagina <https://youtu.be/mptm52qLV-U> Potrai anche vedere i video di moltissimi altri nostri progetti proposti in passato.

NeoPixel, aumentandola per rendere il gioco più impegnativo, ovvero riducendola per farlo diventare più semplice e avvicinabile.

Ricordiamo che la riga di codice dove si imposta la velocità è:

```
const byte ledSpeed[8] = {100, 90, 80, 70, 55, 40, 20, 10};
```

dove potete modificare i valori 100, 90 ecc. scrivendo quelli che desiderate, come spiegato qualche pagina indietro. □



Cosa occorre?

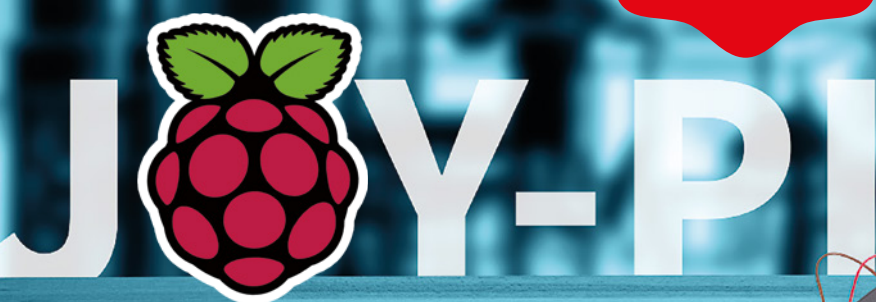
I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. La board Arduino UNO (cod. ARDUINOUNOREV3) è in vendita a Euro 27,50, l'anello NeoPixel da 60 led (cod. RING60NEOPIXEL) a Euro 38,00, la barra con 8 led (cod. STICK8NEOPIXEL) a Euro 6,50, il buzzer senza elettronica (cod. BUZZER5SE) a Euro 0,50, mentre l'alimentatore a 9Vdc (cod. AL9V1A5) è disponibile a Euro 9,50. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

Notebook Joy-Pi Note con Raspberry Pi 4 da 4GB

RASPBERRY PI 4
TIPO B CON 4GB
DI MEMORIA
INCLUSA NELLA
CONFEZIONE!



COD. JT-PINOTE
€ 518,00

Prezzo IVA inclusa.

Joy-Pi Note è un computer portatile basato su Raspberry Pi 4 Tipo B con all'interno un vero e proprio laboratorio per sperimentare, programmare ed apprendere. Viene fornito con 46 corsi, 18 progetti per Python e Scratch, corredato con oltre 22 sensori e moduli.



Altoparlante
stereo



Fotocamera da 2 MPX
con microfono



Tastiera
Wireless



Alimentatore
incluso



Con Raspberry
Pi 4B



Studiato per
l'apprendimento



Display Full HD
da 11,6"



46 Corsi



18 Progetti



Oltre 22 sensori
e moduli

SPEAKING CLOCK

di GUGLIELMO
BRAGUGLIA

*SONO
LE ORE 20:52,
L'APPUNTAMENTO
È ALLE ORE 21:00!*



L'orologio che non si limita a mostrare l'ora, ma, per renderla più comprensibile a tutti, la pronuncia, associandola agli eventi.

È

innegabile che con il passare degli anni e magari con l'insorgere di alcuni disturbi della vista ci siano cose della vita quotidiana che divengono un po' più difficili: un esempio è leggere l'ora mostrata dall'orologio sul mobile della sala o appeso a muro. L'avanzare dell'età, inoltre, ci rende talvolta difficile memorizzare e ricordare cose come il prendere delle medicine ad un certo orario, oppure l'ora del pranzo o della cena e così via. Ecco perché abbiamo progettato qualcosa che potesse aiutare in entrambe le situazioni e da questo lavoro è nato "l'orologio parlante" con messaggistica programmabile descritto in queste pagine. La fase di progetto ha vissuto due momenti, durante i quali siamo partiti

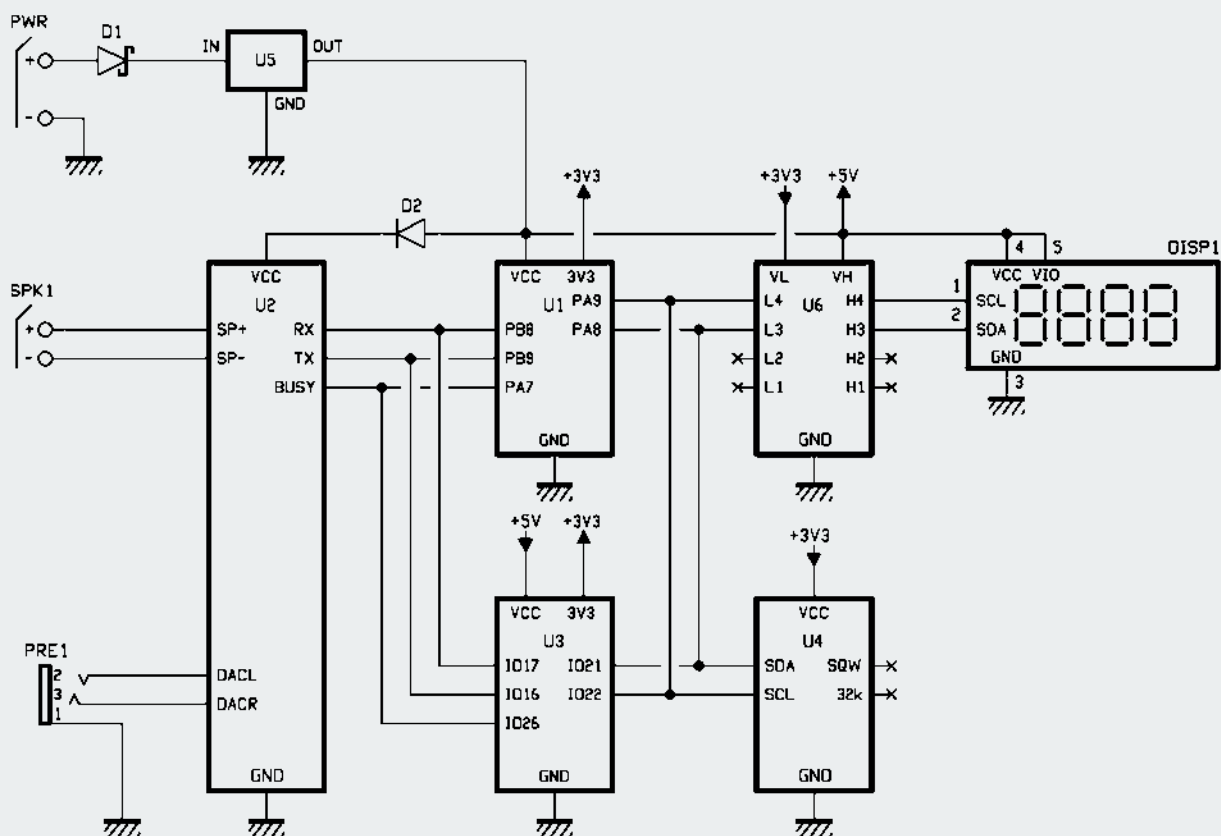


Fig. 1
Modulo RTC
DS3231 con
EEPROM
AT24C32.

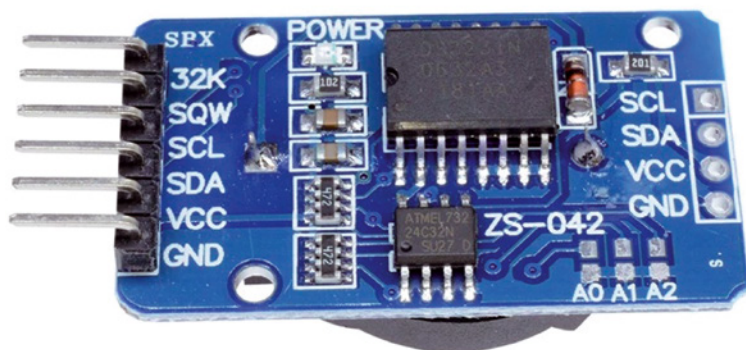
da una versione basata su un ATSAM21, per poi arrivare a una sua evoluzione dotata di interfaccia WiFi. Difatti, la possibilità di programmare messaggi di vari tipo (file audio MP3) ed associarli a giorni diversi, ad orari diversi ed a condizioni varie, ci ha portato a pensarne un utilizzo molto più diffuso come, ad esempio, in ambienti di ufficio (messaggi sull'orario di entrata, di uscita e delle pause), nei negozi (orari di apertura e chiusura diffusi al pubblico ed annuncio di imminente chiusura), lavoratori in smart-working (promemoria degli orari e delle pause), e così via negli ambienti

più svariati con le più variegate esigenze. Abbiamo quindi condensato in un unico circuito le due versioni, consentendo di montare i componenti che servono per quella base o quella avanzata. Infatti la seconda versione è compatibile con la prima e consente sia l'uso del modulo MCU con ATSAMDT1 e sia l'adozione di un ESP32 con possibilità di connessione WiFi per semplificare la configurazione e l'invio di comandi (oltre alla possibilità originale via porta seriale).

SCHEMA ELETTRICO

Guardando lo schema, vediamo che a prescindere dalla versione, il sistema si basa su tre moduli fondamentali:

- un modulo RTC basato su DS3231 (di quelli che hanno a bordo anche la EEPROM AT24C32) che offre la garanzia di una notevole precisione e stabilità dell'ora (**Fig. 1**);
- un modulo riproduttore di MP3 basato su JQ8400 (**Fig. 2**) con memorizzazione dei file audio su flash esterna (e non su SD-Card);
- un modulo MCU, dalle dimensioni estremamente contenute.



Tutti questi moduli hanno il vantaggio di lavorare a 3,3V rendendo semplice la loro interconnessione. Nella versione base, il modulo MCU è prodotto da SeeedStudio, basato su ATSAMD21 e appartenente alla famiglia dei moduli XIAO (**Fig. 3**); nello schema elettrico è siglato U1. Nella versione avanzata, ha invece un core ESP32 (U3).

Particolare attenzione è stata poi posta nella scelta del display da utilizzare per la visualizzazione dell'ora, scelta che è caduta su un modulo prodotto da Adafruit, in cui i quattro digit, di colore giallo (colore molto visibile), sono alti ben 3 cm (1.2") permettendone una chiara visione anche ha chi ha la vista più debole (**Fig. 4**).

Il suddetto modulo ha il vantaggio di colloquiare tramite bus I²C, occupando così solo due linee di I/O del microcontrollore, pin che sono utilizzati anche per il colloquio con il modulo RTC, giacché si tratta di un bus e su un bus possono affacciarsi più dispositivi.

Anche se Adafruit indica che questo modulo va alimentato a 5V ma è controllabile da una MCU a 3,3V (difatti ha il pin dell'alimentazione separato dal pin che porta il positivo sulle resistenze di pull-up incorporate sul bus I²C) la cosa non è sempre vera, perché il chip di controllo montato su tale display richiede almeno 3.5V e a volte, a causa delle tolleranze, può capitare di trovare quello che accetta sul bus I²C a 3,3V, ma è un caso. Per garantire in ogni caso il funzionamento, si è deciso di utilizzare un modulo traslatore di livelli adatto al bus I²C come il Bi-Directional Level Shifter 757, prodotto sempre da Adafruit.

Tali traslatori di livello si trovano, a costi decisamente inferiori, anche di produzione cinese, ma occorre prestare particolare attenzione alla disposizione dei pin, che può essere leggermente

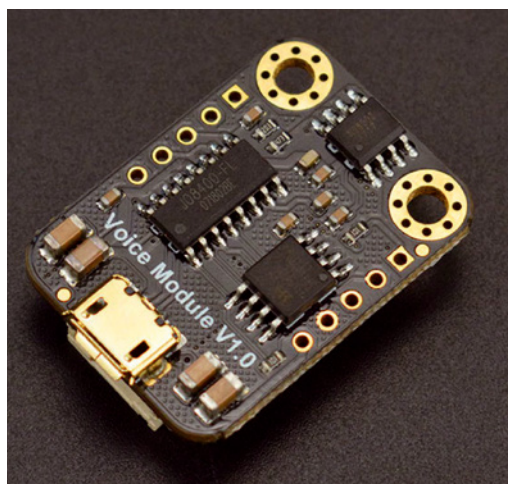


Fig. 2
modulo
vocale
DFRobot
DFR0534.



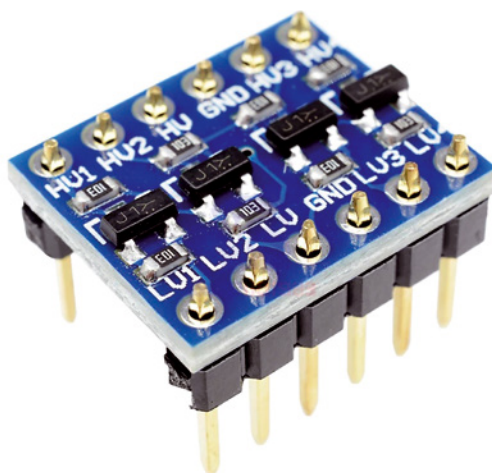
Fig. 3
Modulo
SeeedStudio
Seeeduino
con
ATSAMD21.

diversa (i pin di alimentazione, normalmente sui lati, sono posti al centro invece che agli estremi); un esempio è il convertitore di livelli logici bidirezionale LLCONBI commercializzato da Futura Elettronica, anche on-line su www.futurashop.it.



Fig. 4
Il display
Adafruit 1269.

→ **Fig. 5**
Breakout traslatore
di livelli
LLCONBI.



Come sarà illustrato più avanti, nel disegno del circuito stampato, si è tenuto conto di questa cosa e sono state predisposte più serie di fori per poter così montare sia i moduli originali di Adafruit, sia i loro cloni cinesi con i pin di alimentazione in posizione diversa.

Dato che sia il display, che specialmente il modulo vocale, assorbono parecchia corrente (il modulo vocale, durante la riproduzione, con lo speaker collegato, arriva ad assorbire anche 1A) abbiamo previsto un ingresso in corrente continua da 7V a 15V (in realtà il regolatore utilizzato accetta da 6,5V a 36V, ma, per sicurezza, è meglio restringere l'intervallo) con il classico jack da 2,1 mm e di montare a valle un alimentatore switching della "Traco Power", modello TSR2-2450, in grado di fornire in uscita 5V regolati con una corrente sino a 2 ampere.

I 5V forniti in uscita da detto modulo sono utilizzati per alimentare il display, la parte HV del traslatore di livelli ed il modulo MCU che, a sua volta, fornisce in uscita i 3.3V utilizzati per la parte LV del traslatore di

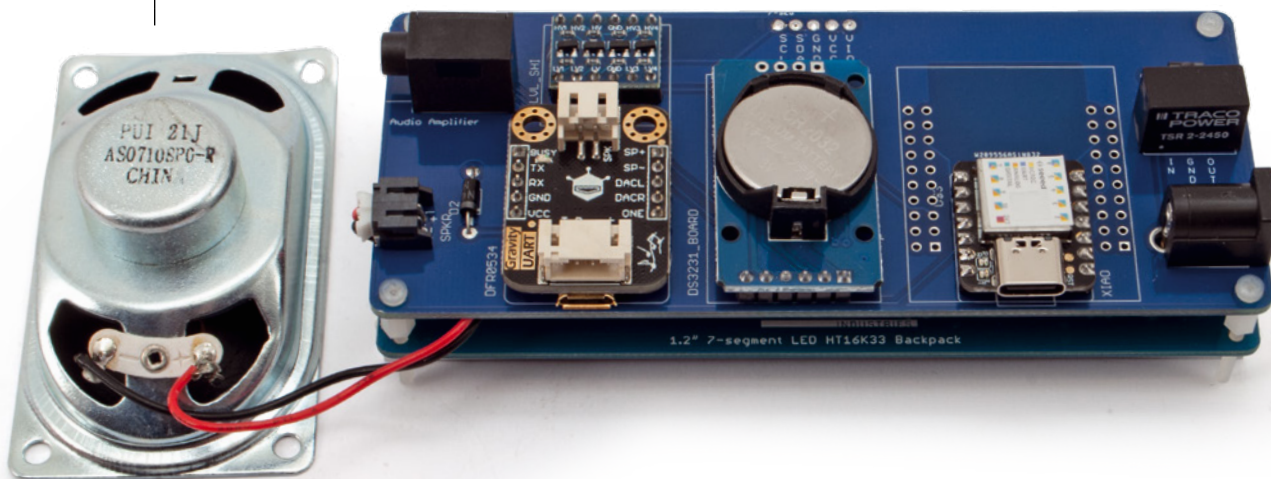
livello e per l'alimentazione del modulo RTC.

Discorso a parte va fatto per il modulo riproduttore di MP3 basato su JQ8400, il quale, come visibile dal suo datasheet, accetta un'alimentazione da 3,3V a 5,5V, ma il miglior funzionamento è garantito attorno ai 4,2V, per cui, per avvicinarsi il più possibile a tale valore ottimale, si è messo in serie al pin di alimentazione, un diodo al silicio SB240 che provoca una caduta di tensione di circa 0,7V e che permette di abbassare i 5V forniti dal regolatore switching ai circa 4,2V ottimali. Altra particolarità di detto integrato è che ha i pin di I/O seriali (TX ed RX) che lavorano invece a 3,3V e che quindi sono direttamente collegabili alla UART della MCU.

Oltre alle parti già descritte in precedenza, in più troviamo un secondo diodo, a protezione delle inversioni di polarità, in serie all'alimentazione generale (subito dopo il jack da 2,1mm), un connettore JST da 2 mm utilizzato per collegare direttamente un altoparlante, da 8Ω 3W, al JQ8400 ed un jack femmina stereo per il collegamento dell'uscita preamplificata del JQ8400 ad un amplificatore esterno (o ad una coppia di casse amplificate come quelle che si collegano all'uscita audio dei PC tramite jack stereo da 3,5mm). Per il disegno del circuito e per la successiva realizzazione dello stampato, si è utilizzato il software "Eagle" nella versione 7.7, ultima realizzata ancora da Cadsoft prima dell'acquisizione da parte di Autodesk. Dato che alcune parti non erano disponibili (ad esempio il modulo audio, il modulo RTC, il modulo XIAO, ecc. ecc.) sono state disegnate e sono fornite in forma di libreria Eagle (.lbr) tra i file di questo progetto.

Il circuito stampato è stato disegnato in modo da poter essere direttamente attaccato dietro al display così da formare un sandwich unico. Le sue misure sono 50,8 mm di altezza per 119,5 mm di lunghezza

↓ L'elettronica
dell'orologio
con il suo
altoparlante.



e, sia il modulo audio, che il modulo di MCU, sono posizionati in modo da rendere facilmente disponibile l'accesso alle relative porte USB necessarie sia per il caricamento/aggiornamento del firmware sulla MCU che per il caricamento/aggiornamento dei file audio sul modulo con il JQ8400. La porta USB sulla MCU è anche utilizzata, come porta seriale, nella fase di inizializzazione dell'orologio o durante la modifica, in funzione delle proprie necessità, di alcuni suoi parametri di funzionamento.

LA VERSIONE WIFI

Per l'utilizzo negli ambienti dove avere un WiFi attivo non crea problema (sia sotto il profilo ambientale di emissione di onde elettromagnetiche, sia sotto l'aspetto delle interferenze ad apparecchiature, come alcuni elettromedicali e strumenti diagnostici) abbiamo realizzato la versione "connessa" la quale si distingue per il modulo MCU: basta montare, in luogo del modulo con ATSAM21 quello con ESP32. Questo permetterà, ad esempio di fornire via wireless i comandi all'orologio. Lo stampato prevede entrambi, ma, come vedete dal piano di montaggio, praticamente, è possibile montare solo uno dei due nel relativo alloggiamento.

Per contenere al massimo l'ingombro e mantenere la compatibilità con il modulo basato su ATSAM21, si è scelto il modulo ESP32 della WeMos D1 mini dalle dimensioni estremamente contenute (**Fig. 6**). Come accennato, lo stampato ha dimensioni tali da poterlo montare a sandwich sul retro del display, ottenendo un modulo compatto e con i connettori USB sempre accessibili per aggiornamenti e personalizzazioni (vedremo nella parte dedicata al software che, salvo una serie di messaggi MP3 riservati per il funzionamento del sistema, è comunque possibile personalizzare il tutto con l'aggiunta di propri file audio).

Se guardate il piano di montaggio del circuito stampato, notate in alto a sinistra, la doppia fila di fori per l'inserimento di due tipi di traslatori di livello, sia quello originale prodotto da Adafruit, con le alimentazioni agli estremi, sia i compatibili cinesi, con le alimentazioni al centro. In basso, verso destra, più file di fori per poter inserire o il modulo MCU XIAO con ATSAM21 di SeeedStudio o il modulo MCU WeMOS ESP32 D1 mini.

Verso il centro dello stampato trova posto il classico modulo RTC, basato su DS3231, con l'alloggiamento della batteria rivolto verso l'alto (così da permetterne sempre la sostituzione), mentre in basso a sinistra trova spazio il modulo vocale MP3 basato su chip JQ8400. Sempre a sinistra, a fianco di tale modulo,

si trova sia il connettore JST da 2mm per il collegamento diretto di un altoparlante da 8Ω 3W, sia il connettore audio stereo per jack da 3.5 mm dove collegare delle casse amplificate per avere un audio più potente. Sulla destra si trova infine il classico jack nero con contatto centrale da 2,1 mm, che permette l'alimentazione del tutto con un alimentatore esterno da 7 a 12V.

Si conclude qui la descrizione dell'hardware del nostro progetto; ora analizziamo gli aspetti firmware, descrivendo il software che permette il suo funzionamento e la sua programmazione.

IL FIRMWARE

Siccome il progetto permette l'utilizzo sia di MCU ATSAM21 che di SoC ESP-WROOM-32, a livello di codice, per compilare, selezionata nell'IDE di Arduino la giusta MCU, non occorre fare alcun intervento manuale, perché una serie di #define inserite in vari punti del programma, provvedono a riconoscere se si sta compilando per ATSAM21 (*viene verificata la presenza della definizione ARDUINO_SEED_XIAO_MO*) o per ESP32 (*viene verificata la presenza della definizione ESP_PLATFORM*) ed ad abilitare o disabilitare intere parti di codice.

La sola linea di programma che ha senso modificare a mano è quella per abilitare tutta la messaggistica di "DEBUG" che, se attiva, viene inviata sulla porta seriale (USB) della MCU. Tale riga è ad inizio programma ed è normalmente commentata:

```
// #define DEBUG // uncomment this line ONLY
// for debug and then keep
// the serial monitor opened
```

Nel corso del programma viene verificata la



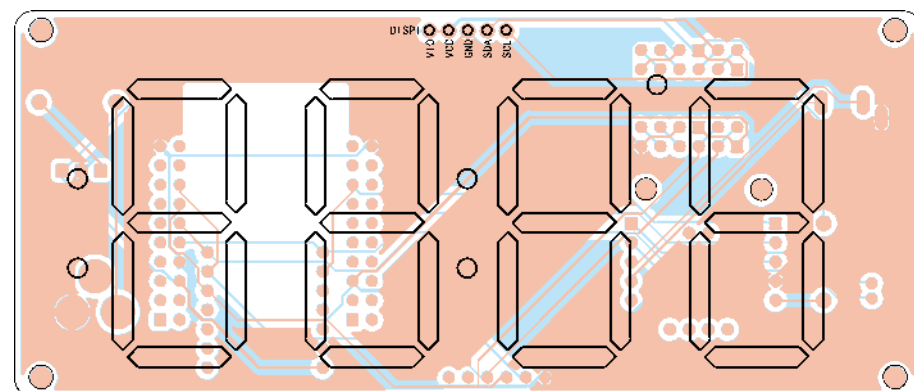
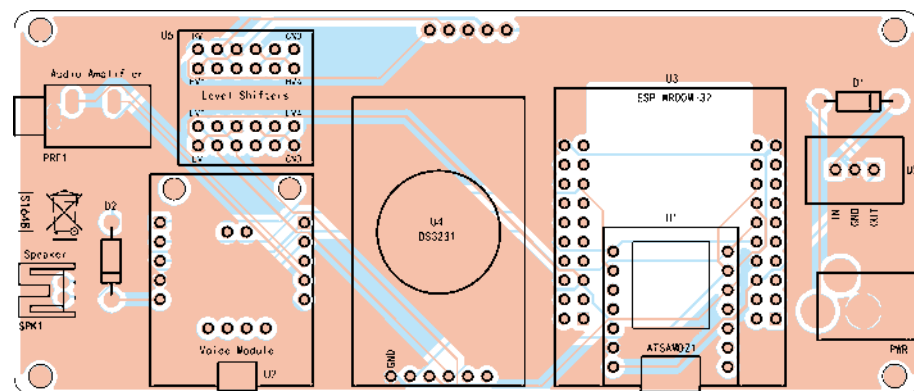
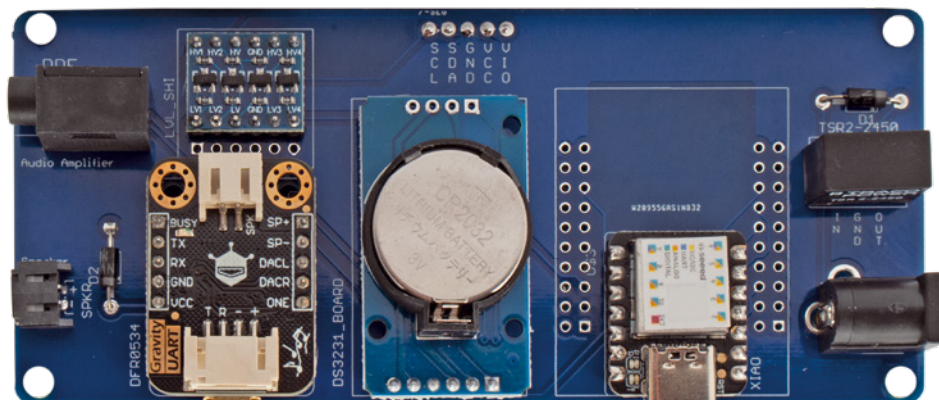
Fig. 6
Il modulo
WeMos
ESP32 D1
Mini.

Elenco Componenti:

DISP1: Display 7 segmenti
A.C. I2C
D1: SB240
D2: SBYV27
U1: Modulo ATSAM21
U2: Modulo MP3 DFR0534
U3: Modulo ESP-WROOM-32
U4: Modulo RTC DS3231
U5: Regolatore TSR2-2450
U6: Adattatore di livelli
PRE1: Presa Jack 3.5 mm
SJ1-3533NG
SPK1: Connettore JST 2 vie
passo 2 mm

Varie:

- Plug DC
- Strip femmina 5 vie (2 pz.)
- Strip femmina 6 vie (2 pz.)
- Strip femmina 7 vie (2 pz.)
- Strip femmina 2x10 vie (2 pz.)
- Strip maschio 5 vie (2 pz.)
- Strip maschio 6 vie (2 pz.)
- Strip maschio 7 vie (2 pz.)
- Strip maschio 2x10 vie (2 pz.)
- Circuito stampato S1648
(51x120 mm)



presenza o meno della definizione "DEBUG" e, di conseguenza, attivata o meno la messaggistica di diagnostica. Normalmente, salvo non si facciano modifiche e non si voglia verificare il corretto funzionamento del programma, tale costante NON deve essere definita ovvero la riga va lasciata con il commento davanti.

Sempre in testa al programma troviamo una serie di definizioni relative ad alcune costanti utilizzate all'interno del codice tra cui il livello logico con cui il LED incorporato sulla scheda della MCU si accende, il numero di alcuni pin ed l'indirizzo I²C dei moduli utiliz-

zati. A seguire c'è una serie di #include che includono le varie librerie utilizzate, alcune standard del 'C', altre proprie di Arduino, altre specifiche per la gestione dei moduli hardware utilizzati ed altre generiche. Alcune di esse sono state parzialmente modificate o corrette e devono essere necessariamente scaricate o modificate e devono essere necessariamente scaricate dalla sezione di *download* della rivista (assieme al sorgente completo del programma) per avere la giusta versione. L'utilizzo di quelle normalmente reperibili su GitHub, fornite dagli autori, compromette il funzionamento dell'applicazione e quindi è da evitare (ad esempio, la libreria "JQ8400_Serial" originale

contiene un metodo, ampiamente usato all'interno della nostra applicazione, che, nella sua versione originale contiene un baco e non funziona). terminate le varie definizioni costanti, prima di permettere la compilazione vera e propria, il programma esegue una serie di verifiche sui valori definiti all'interno di librerie/include esterni e, se qualche valore non è conforme alle aspettative, la compilazione viene bloccata.

Se le verifiche sono soddisfatte, si passa alla parte di dichiarazione delle variabili globali, degli "enums", delle strutture ed ad istanziare le varie classi. Dopo tale parte, nel codice troviamo prima di tutto una serie di funzioni di utilità, come ad esempio le funzioni per il salvataggio ed il recupero dei parametri iniziali dalla EEPROM, le funzioni per la lettura e scrittura del RTC DS3231, le funzioni per il calcolo dell'ora legale (se utilizzata), ecc. ecc., funzioni che sono poi richiamate a livello degli immancabili (essendo un programma sviluppato in ambiente IDE Arduino) `setup()` e `loop()`.

Finita la serie di funzioni di utilità inizia tutta una serie di callback ad uso della libreria **SerialCmd**, libreria descritta nel numero 266 (luglio/agosto 2022) di Elettronica In e ampiamente utilizzata nel nostro programma per interpretare tutti i possibili comandi che si possono fornire sia via porta seriale (collegamento USB) che, nel caso di utilizzo di ESP32, anche via WiFi. L'elenco di tutti i comandi disponibili con la versione 1.2.1 del firmware è riassunto nella **Tabella 1**, dove sono anche dettagliati i parametri che ciascun comando si attende e cosa viene fatto quando viene ricevuto.

Nel sorgente, ogni callback relativa alla "SerialCmd" ha un commento che indica a quale comando essa è associata (l'effettiva associazione tra comando e callback viene fatta nel `setup()`).

Finita tutta la serie di callback si raggiunge finalmente la funzione `setup()` e quindi la funzione `loop()`. Rammentiamo che il "core" Arduino, per semplicità, nasconde all'utilizzatore la funzione principale di ogni programma C/C++, ovvero la funzione `main()`, ma che in realtà tale funzione esiste ed è lei che richiama una sola volta la funzione `setup()` e poi, in un ciclo infinito (quindi in continuazione), chiama la funzione `loop()`.

Quanto viene svolto nel **setup()** è illustrato nel diagramma di flusso proposto nella **Fig. 7**.

Vengono prima di tutto inizializzati i vari pin di I/O, le porte seriali (sia fisiche che eventuali porte virtuali), i vari bus (in particolare il bus I²C) e, solo se si sta compilando per ESP32, il modulo WiFi di detto SoC. Di seguito vengono verificati i vari moduli hardware

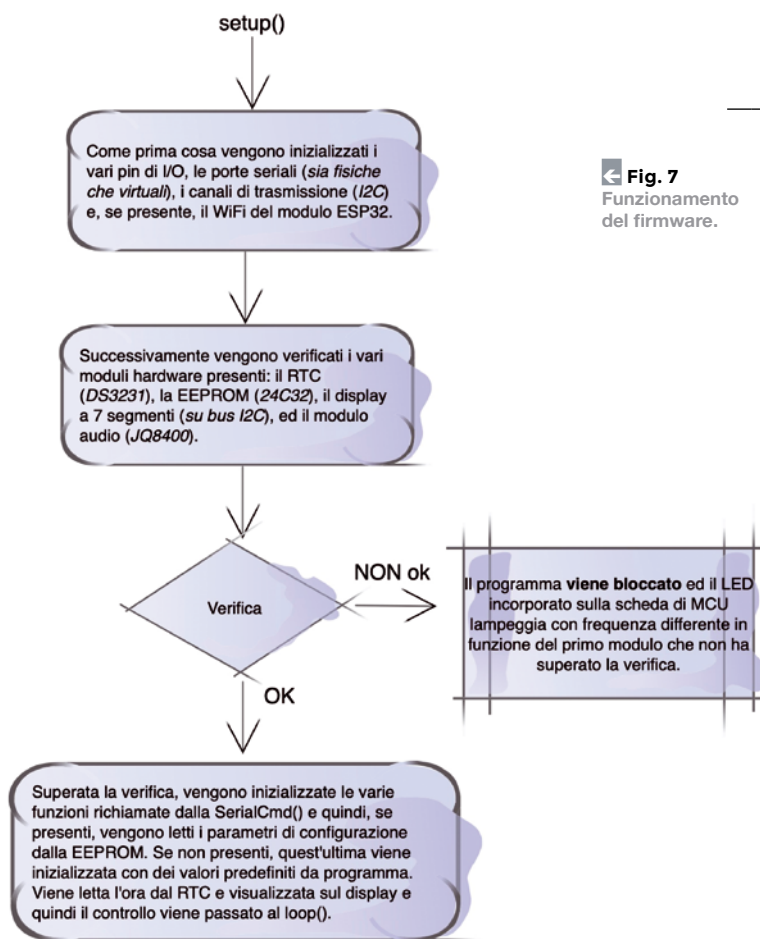


Fig. 7
Funzionamento del firmware.

presenti nel circuito, controllando la loro risposta o sul bus I²C (per la EEPROM, il modulo RTC ed il display a 7 segmenti), o sulla porta seriale (per il modulo audio basato su JQ8400). Se la verifica ha successo, il programma prosegue, altrimenti, se uno dei moduli hardware fallisce la verifica, il programma viene arrestato ed il LED presente a bordo del modulo MCU viene fatto lampeggiare con frequenza differente in funzione del primo modulo non funzionante trovato (**Tabella 2**).

Se la verifica viene superata, si procede con l'assegnazione dei vari comandi alle varie callback della "SerialCmd":

```

...
mySerCmd.AddCmd ( "SETDT" , SERIALCMD_FROMALL,
set_DTTM );
mySerCmd.AddCmd ( "GETDT" , SERIALCMD_FROMSERIAL,
get_DTTM );
mySerCmd.AddCmd ( "SETUD" , SERIALCMD_FROMALL,
set_UDST );
mySerCmd.AddCmd ( "SETDB" , SERIALCMD_FROMALL,
set_DIBT );

```


 **Tabella 1**
Comandi e
parametri attesi.

SETDT,YY,MM,GG,HH,MN,SS,DW

Set date/time

YY : Year (0 .. 99)
MM : Month (1 .. 12)
GG : Day (1 .. 31)
HH : Hours (0 .. 23)
MN : Minutes (0 .. 59)
SS : Seconds (0 .. 59)
DW : Day of Week (1 .. 7, 1 = Domenica)

N.B.: si **DEVE** sempre impostare l'"ora solare" e mai l'"ora legale", anche quando essa è attiva. Ciò significa che durante l'"ora legale" è necessario fornire la data e l'ora come se fosse in vigore l'"ora solare". L'orologio si corregge da solo e visualizza l'ora corretta.

GETDT

Get (print) date/time in the same format of SETDT (valid only from serial port)

SETUD,DS

Set use DST

DS : 1 = usa DST, 0 = Non usa DST (DST: utilizzo dell'ora legale)

SETDB,DS,DN

Set display brightness

DS : luminosità del display durante il giorno (0 .. 15)
DN : luminosità del display durante la notte (0 .. 15)

SETPA,MN

Set period of announcements

MN : gli annunci possono avere solo periodicità di: **10, 15, 30 o 60** minuti

SETON,HH,MN

Set audio ON time for announcements

HH : Hours (0 .. 23)
MN : Minutes (0 .. 59)

SETOF,HH,MN

Set audio OFF time for announcements

HH : Hours (0 .. 23)
MN : Minutes (0 .. 59)

SETAM,MN,VD,HO,MO,HF,MF,M1,M2,... up to max of (MAX_USR_LNG - 1) messages

Set User defined announcement (played after the standard announcement)

MN : Message Number (0 .. MAX_USR_MSG - 1)
VD : Valid days (0 = all, 1 .. 254 days selection, see note 1)
HO : Hours Start (0 .. 23)
MO : Minutes Start (0 .. 59)
HF : Hours Stop (0 .. 23)
MF : Minutes Stop (0 .. 59)
Mn : Audio message number (ZH/nn.MP3), can be repeated max (MAX_USR_LNG - 1) times

note 1: VD è un campo binario, i bit validi vanno dal bit 1 al bit 7, il bit zero è riservato. Per indicare i giorni validi sommare il valore decimale di ciascun bit.
Quindi, Dom = 2, Lun = 4, Mar 8, Mer = 16, Gio = 32, Ven = 64, Sab = 128.
Ad esempio, per essere valido solo per Lun, Mer e Ven si deve sommare 4 + 16 + 64 = 84, che è il valore da inserire nel campo VD.

Esempio: SETAM, 0, 0, 10, 30, 12, 7, 70, 12, 0 - Programma il messaggio personalizzato numero 0, valido tutti i giorni (0), dalle ore 10, minuti 30, alle ore 12, minuti 5, file 70.mp3, file 12.mp3, 0 finale (*non obbligatorio, rende solo più chiaro*).

CLRAM,MN

Clear User defined announcement (played after the standard announcement)

MN : Message Number (0 .. MAX_USR_MSG)

GETPM

Get (print) all parameters (valid only from serial port)

SETOD,WD

Set audio off week day

wd : da 0 a 7 (1 = Domenica, 2 = Lunedì, .. 7 = Sabato, 0 = tutti i giorni)

SETAV,VL

Set audio volume

VL : da 0 a 30

SETDY,UD

Set adds the date in the time announcement

UD : 1 = adds, 0 = Don't adds

Nota: Se il parametro UD è messo ad 1, dopo l'annuncio dell'ora (*solo ogni ora esatta*), viene anche indicata la data

RESET

Do a software reset of the MCU and restart the program

```

mySerCmd.AddCmd ( "SETPA" , SERIALCMD_FROMALL,
set_PERA );
mySerCmd.AddCmd ( "SETON" , SERIALCMD_FROMALL,
set_TMON );
mySerCmd.AddCmd ( "SETOF" , SERIALCMD_FROMALL,
set_TMOF );
mySerCmd.AddCmd ( "SETOD" , SERIALCMD_FROMALL,
set_ODAY );
mySerCmd.AddCmd ( "SETAV" , SERIALCMD_FROMALL,
set_AVOL );
mySerCmd.AddCmd ( "SETDY" , SERIALCMD_FROMALL,
set_UDAY );
mySerCmd.AddCmd ( "SETAM" , SERIALCMD_FROMALL,
set_AMSG );
mySerCmd.AddCmd ( "CLRAM" , SERIALCMD_FROMALL,
clr_AMSG );
mySerCmd.AddCmd ( "GETPM" , SERIALCMD_FROMSERIAL,
get_PARM );
mySerCmd.AddCmd ( "RESET" , SERIALCMD_FROMALL, do_RE-
SET );

```

...

proseguendo poi con la lettura dei dati presenti in EEPROM (se presenti, altrimenti vengono utilizzati dei valori predefiniti), con quelli presenti nell'RTC e con l'attivazione del display per la visualizzazione dell'ora.

Le informazioni salvate in EEPROM (ed eventualmente memorizzate con predefiniti la prima volta) sono descritte nella struttura 'C' "params". I vari elementi hanno nomi descrittivi ed il commento, associato ad ogni singola riga, ne permette facilmente la comprensione. Unico elemento degno di nota è il primo, "char blockID" che serve a indentificare la corretta inizializzazione della EEPROM e la versione della sua struttura (il contenuto è uguale a quello dalla costante BLOCK_ID). Per maggiore sicurezza, su tutto il blocco dei parametri è calcolato un **CRC32**, ed il valore è salvato, sempre in EEPROM, subito dopo il blocco dei parametri. Ogni volta che il suddetto blocco viene letto, viene ricalcolato il CRC32 e confrontato con il valore memorizzato. Se i due valori non corrispondono si considera il contenuto non

valido e la EEPROM viene reinizializzata con i valori predefiniti. Per completezza riportiamo il contenuto del blocco "params" (relativo alla versione **1.2.1** del programma):

```

struct params {
    char    blockID [7];
    // block_id to identify valid params
    bool    use_DST;
    // automatically adjust the clock following DST
    uint8_t display_bright;
    // display brightness during day (0 .. 15)
    uint8_t display_bright_night;
    // display brightness in the night (0 .. 15)
    uint8_t audio_period;
    // period of audio announcements (10/15/30/60)
    uint32_t audio_on;
    // audio-on time and day brightness display
    uint32_t audio_off;
    // audio-off time and night brightness display
    uint8_t audio_off_day;
    // audio off day (0 = none, 1 Sun .. 7 Sat)
    uint8_t audio_volume;
    // audio volume (0 .. 30)
    bool    audio_add_day;
    // adds the date in the time announcements
    message user_msgs[MAX_USR_MSG];
    // Array of custom user message list
} clock_params;

```

mentre la funzione che calcola il CRC32 si trova nei due file (che vengono inclusi ad inizio programma) di nome "crc32.h" e "crc32.c" e che si trovano nella stessa cartella del programma principale (*phxSpeakingClock*).

Terminata la funzione di setup(), ricordiamo che il controllo torna al main() (che, come detto è nascosto nel "core" Arduino) il quale, all'interno di un ciclo FOR infinito, chiama la funzione loop() dalla quale, quindi, si entra e si esce in continuazione; questa cosa è da tenere sempre a mente perché eventuali variabili locali dichiarate dentro la funzione loop(), se non dichiarate di tipo "static", vengono ogni volta create ad

COMPONENTE GUASTO	SEMIPERIOLO DI LAMPEGGIO DEL LED
DS3231 RTC	250 ms
24C32 EEPROM	125 ms
Display 7 segmenti I ² C-Bus	62 ms
Modulo Audio con JQ8400	30 ms

 **Tabella 2**
Segnalazioni
del LED.



Fig. 8
Funzionamento
della funzione
loop().



inizio funzione e distrutte al momento della sua fine, con conseguente perdita del valore tra una chiamata al `loop()` e la successiva).

Quanto svolto ad ogni chiamata della funzione **loop()** è illustrato nel diagramma di **Fig. 8**.

In detta funzione tutte le varie operazioni vengono programmate facendo uso della funzione `millis()` nella classica forma:

```

if ( millis() - ultimoMillis >= tempoPrevisto ) {
    // ... esegue quello che deve fare
    // ...
    ultimoMillis += tempoPrevisto;
}

```

ad eccezione del controllo relativo alla scrittura su EEPROM che segue una logica leggermente diversa. Tale controllo è il primo che viene effettuato; per evitare una continua scrittura sulla EEPROM ad ogni cambiamento di un singolo parametro (ricordiamo che la vita di una EEPROM prevede, per quanto alto, comunque un numero limitato di scritture), si è scelta la strada di memorizzare il valore di `millis()` al momento della modifica di uno qualsiasi dei parametri e di effettuare la scrittura fisica solo dopo il passare di `EEPROM_W_TIME` (valore fissato con una *#define*) millisecondi dall'ultima modifica. Unico accorgimento, viene riservato il valore di `millis()` pari a zero (che indica nulla da salvare), per cui, se una qualsiasi modifica avviene proprio nell'istante in cui `millis()` vale zero, viene memorizzato il valore subito successivo, ovvero 1.

In pratica la parte di codice che effettua il salvataggio dei parametri in EEPROM ha, all'incirca, questa forma:

```

if ( EEP_millis != 0 ) {
    if ( millis() - EEP_millis > EEPROM_W_TIME ) {
        EEP_millis = 0;
        save_params();
    }
}

```

Dopo il controllo relativo alla scrittura su EEPROM, c'è una serie di altri compiti che vengono svolti ad intervalli prefissati.

1. se è tempo di leggere l'ora esatta dall'RTC e visualizzarla sul display (questo viene fatto ogni 15 secondi anche perché il display visualizza solo ore e minuti quindi un aggiornamento al secondo sarebbe del tutto superfluo).
2. se è il minuto in cui è necessario dare qualche annuncio (rammentiamo che si può programmare la

NOME	CONTENUTO	NOME	CONTENUTO	NOME	CONTENUTO
00.mp3	Sono le ore	31.mp3	e quaranta	62.mp3	sabato
01.mp3	una	32.mp3	e cinquanta	63.mp3	<i>vd.sotto</i>
02.mp3	due	33.mp3	e quindici	64.mp3	primo
03.mp3	tre	34.mp3	e quarantacinque	65.mp3	<i>riservato</i>
04.mp3	quattro	35.mp3	ventiquattro	66.mp3	<i>riservato</i>
05.mp3	cinque	36.mp3	venticinque	67.mp3	<i>riservato</i>
06.mp3	sei	37.mp3	ventisei	68.mp3	<i>riservato</i>
07.mp3	sette	38.mp3	ventisette	69.mp3	1 sec. silenzio
08.mp3	otto	39.mp3	ventotto	70.mp3	<i>vd.sotto</i>
09.mp3	nove	40.mp4	ventinove	71.mp3	<i>vd.sotto</i>
10.mp3	dieci	41.mp3	trenta	72.mp3	<i>vd.sotto</i>
11.mp3	undici	42.mp3	trentuno	73.mp3	<i>vd.sotto</i>
12.mp3	dodici	43.mp3	gennaio	74.mp3	<i>vd.sotto</i>
13.mp3	treddici	44.mp3	febbraio	75.mp3	<i>vd.sotto</i>
14.mp3	quattordici	45.mp3	marzo	76.mp3	<i>vd.sotto</i>
15.mp3	quindici	46.mp3	aprile	77.mp3	<i>vd.sotto</i>
16.mp3	sedici	47.mp3	maggio		
17.mp3	diciassette	48.mp3	giugno		
18.mp3	diciotto	49.mp3	luglio		
19.mp3	diciannove	50.mp3	agosto		
20.mp3	venti	51.mp3	settembre		
21.mp3	ventuno	52.mp3	ottobre		
22.mp3	ventidue	53.mp3	novembre		
23.mp3	ventitré	54.mp3	dicembre		
24.mp3	mezzanotte	55.mp3	di		
25.mp3	e un quarto	56.mp3	domenica		
26.mp3	e mezza	57.mp3	lunedì		
27.mp3	e tre quarti	58.mp3	martedì		
28.mp3	e dieci	59.mp3	mercoledì		
29.mp3	e venti	60.mp3	giovedì		
30.mp3	e trenta	61.mp3	venerdì		
63.mp3	controllo iniziale completato con successo, l'orologio viene avviato				
70.mp3	il pranzo è alle ore				
71.mp3	la cena è alle ore				
72.mp3	ricordarsi di prendere le medicine alle ore				
73.mp3	il negozio apre alle ore				
74.mp3	il negozio chiude alle ore				
75.mp3	l'ufficio apre alle ore				
76.mp3	l'ufficio chiude alle ore				
77.mp3	e riapre alle ore				

← **Tabella 3**
contenuto
audio dei file
.mp3

periodicità degli annunci con intervallo di 10, 15 30 e 60 minuti).

3. dopo un annuncio, calcolare il nuovo "minuto" a cui dovrà essere dato il prossimo annuncio.
4. accendere o spegnere (*far lampeggiare*) il LED che si trova sulla scheda della MCU per indicare che comunque il tutto sta lavorando e che il programma non si è bloccato.
5. verificare se ci sono dei comandi in arrivo da porta seriale (USB) e/o, nel caso di moduli ESP32, in arrivo da WiFi ed eseguire i comandi ricevuti.
6. solo se ci si trova su piattaforma ATSAM21, segnalare al watchdog che il programma sta regolarmente girando e che non è necessario il suo intervento. Su piattaforma ESP32 il watchdog è direttamente gestito dal firmware del modulo.

Svolti tutti i vari punti il loop() giunge al termine e, richiamato immediatamente dopo dal main(), ricomincia da capo.

Esaminata la logica del programma, dedichiamoci a capire il funzionamento della "messaggistica vocale". Come detto, il compito di riprodurre i messaggi vocali è demandato ad un modulo basato su **JQ8400** con i vari messaggi vocali memorizzati in una memoria non volatile (*NON su SD, ma su un chip di memoria*). Il JQ8400 ha la particolarità di poter riprodurre sia singoli file audio, sia, cosa da noi sfruttata, sequenze di file audio concatenati l'un l'altro, quindi, ad esempio, per indicare che sono le 10:30, verranno concatenati una serie di tre file audio (file .mp3) contenenti:

- sono le ore;
- dieci;
- e trenta.

Ovvero tre file separati che risulteranno in un unico audio riprodotto la frase "sono le ore 10 e trenta". Notate che per come è stato realizzato dal produttore il firmware del JQ8400, i file audio che si vogliono poter riprodurre, oltre che singolarmente, in forma **concatenata**, devono risiedere in una cartella, subito sotto la radice principale, con nome fisso "ZH" ed i file .mp3 è bene che abbiano come prefisso due caratteri numerici (*da 00.mp3 fino a 99.mp3*).

L'attuale versione del programma (che, rammentiamo, al momento della scrittura di questo articolo, è la versione 1.2.1) si riserva l'uso dei file mp3 da 00.mp3 a 69.mp3, mentre da 70.mp3 a 99.mp3 sono a disposizione dell'utente per eventuali messaggi personalizzati. Quale esempio, i messaggi da 70.mp3 a 77.mp3 vengono forniti già registrati per gli utilizzi più comuni. La **Tabella 3** riassume il contenuto audio dei file .mp3 che vengono forniti già codificati.

Visti tutti i vari messaggi che sono a disposizione, è ora più facile comprendere il senso del messaggio personalizzato che abbiamo riportato precedentemente quale esempio, nel comando SETAM. Se ben si ricorda avevamo fatto l'esempio di un messaggio del tipo:

SETAM, 0, 0, 10, 30, 12, 7, 70, 12, 0

ovvero la programmazione del messaggio personalizzato numero 0, valido tutti i giorni (0), dalle ore 10, minuti 30, alle ore 12, minuti 5, file 70(.mp3), file 12(.mp3), 0 finale (non obbligatorio, rende solo più chiaro). Per quanto sopra riportato, ad ogni periodo prefissato (ricordiamo 10, 15, 30 o 60 minuti) verrà inviato il messaggio vocale relativo all'ora seguito dal messaggio "il pranzo è alle ore dodici" (file 70.mp3 e file 12.mp3).

Per completezza (e per chi fosse interessato a creare nuovi messaggi) segnaliamo che tutti i vari file .mp3 sono stati generati scrivendo il testo del messaggio ed utilizzando il servizio "Add to iTunes as a spoken text" che è disponibile su macOS utilizzando la "voce" di lingua Italiana "Luca" che è tra quelle scaricabili dalle preferenze di sistema. I file generati da iTunes sono stati poi convertiti in .mp3 utilizzando il programma, sempre per macOS (gratuito e scaricabile dalla rete), Audacity.app che permette anche delle eventuali modifiche (tonalità, velocità di riproduzione, taglio di parti audio, e via di seguito).

CONCLUSIONI

Si conclude così quest'articolo dedicato all'orologio parlante; l'hardware è semplice e versatile in modo da poter essere personalizzato ed il programma è piuttosto lineare e dovrebbe essere abbastanza semplice, per chi abbia le conoscenze di base della programmazione C/C++, poterne seguire il flusso con l'aiuto delle nostre indicazioni. □



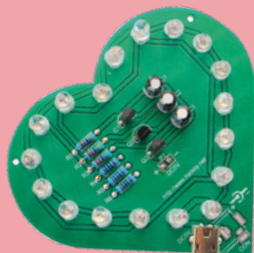
Cosa occorre?

Il circuito stampato dell'orologio parlante può essere realizzato con il servizio di realizzazione PCB (cod. PCBPRODUCTION) scaricando il Gerber dalla pagina download della rivista.

Il materiale va richiesto a:

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

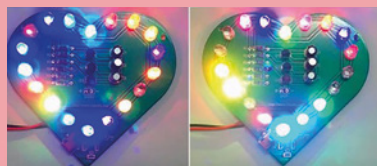
Sorprendila con i kit di San Valentino



cod. CUORELED
€ 5,00

CUORE A LED IN KIT

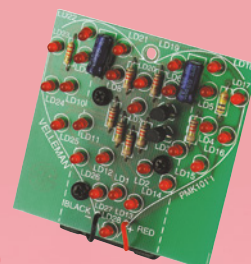
Gadget elettronico a forma di cuore. Composto da 18 LED cambiacolore, suddivisi in tre gruppi, che si accendono in modo casuale. Ideale come semplice kit per hobbisti alle prime armi, come gadget per un regalo, per San Valentino, ecc. Il circuito può essere alimentato con tre pile da 1,5 volt tipo AA oppure a 5 VDC tramite il connettore micro USB, dimensioni 84x77x13 mm.



CUORE LAMPEGGIANTE IN KIT

28 LED formano due cuori lampeggianti. Alimentazione con batteria da 9V (acquistabile separatamente), assorbimento $\pm$ 8mA, dimensioni 65 x 70 x 30 mm

cod. MK101
€ 10,50



cod. MK144
€ 9,00

GHIRLANDA LUMINOSA A LED CON CUORI A LUCE COLORATE

Ghirlanda costituita da 10 cuori con all'interno un LED bianco. La distanza tra i cuori è di 9 cm e la lunghezza del cavo 1,15 m. Alimentazione con 2 batterie da 1,5 V (tipo AA - acquistabili separatamente). Solo per ambienti interni. Disponibili a luce rossa o bianca calda.



cod. XML24R
€ 6,20



cod. XML24W
€ 6,20

CUORE LAMPEGGIANTE SMD

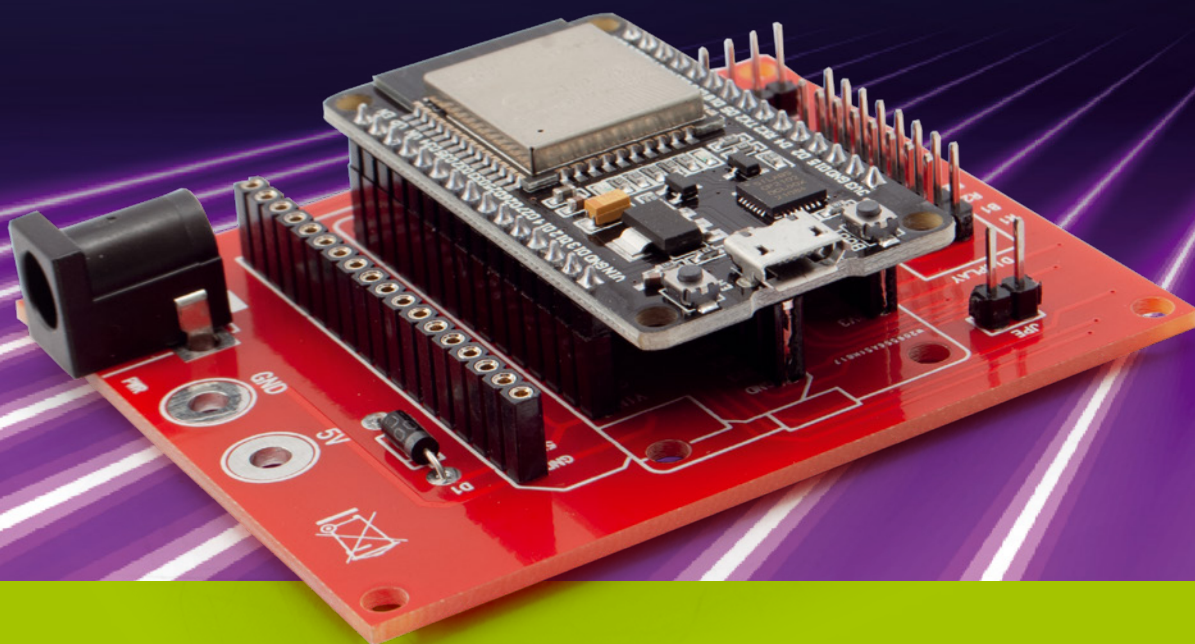
Gadget elettronico munito di 6 LED che si illuminano alternativamente, realizzato con componenti SMD. Alimentazione mediante batteria a bottone da 3V (CR2025 o CR2032 acquistabili separatamente). Dimensioni 49 x 44 x 8 mm



Prezzi IVA inclusa.

LED MATRIX CON ESP32

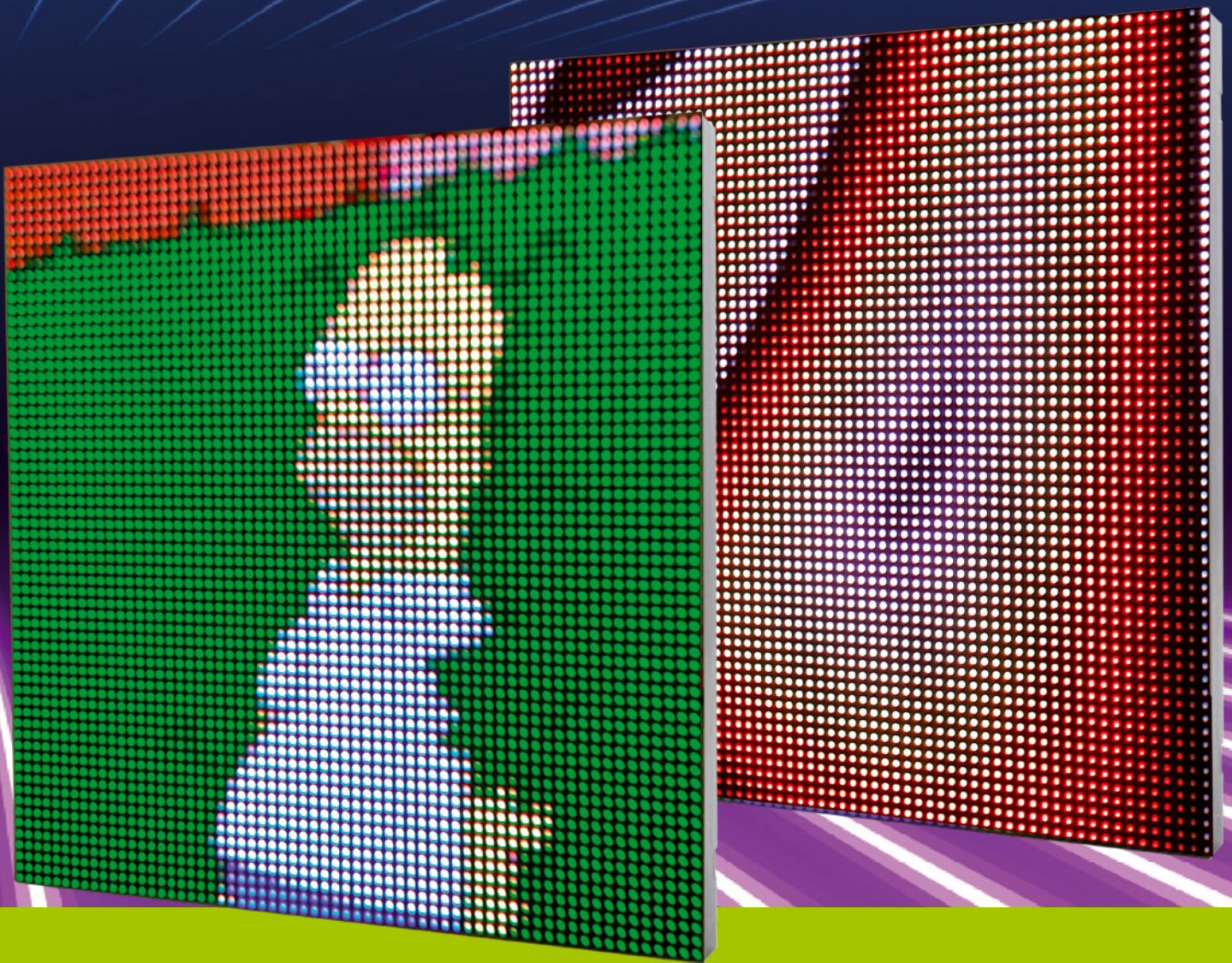
di DARIO CICERI



Pilotiamo una matrice di LED RGB standard tramite bus seriale utilizzando una scheda ESP32, con la possibilità di scegliere tra più versioni.

Le matrici a LED sono sempre state qualcosa di molto affascinante, perché ci permettono di sprigionare tutta la nostra creatività attraverso centinaia, se non migliaia di pixel colorati. Questo vale ancor di più per le matrici di LED RGB che negli ultimi anni hanno pervaso il mercato consumer e sono state rese accessibili, per via del costo e della facilità di integrazione, anche ai Maker e al pubblico amatoriale. Possiamo utilizzarle veramente per qualsiasi cosa, come nei pannelli di controllo, in insegne pubblicitarie o cartelli informativi, tabelloni segnapunti, ma anche display per animazioni, videogiochi e -perché no- anche un bellissimo orologio con uno sfondo composto dai nostri personaggi preferiti, eroi dei fumetti e personaggi dei videogiochi del passato (**Fig. 1**). Dallo stile un po' retrò, queste matrici non smettono mai di





stupire, arrivando ad avere risoluzioni come 64x64 LED RGB da 3mm, per un totale di 4.096 LED (e altrettanti pixel), con la possibilità di concatenare più matrici, superando ogni limite e creando quindi display con risoluzioni ancora maggiori, pur accettando dei compromessi quando si parla di immagini in movimento. Questo permette di realizzare grandi schermi e di mostrare anche dei video con buona risoluzione, perché se è vero che i pixel sono grandi (ognuno è un LED RGB del diametro di alcuni millimetri) osservandoli da lontano la trama degli schermi diviene quasi impercettibile. In passato abbiamo già utilizzato le matrici a LED RGB, pilotandole per esempio con circuiti a FPGA o computer, ma in questo articolo vogliamo proporvi qualcosa di più originale e abbordabile, perché

esporremo un metodo a basso costo, compatto, dotato di interfacciamento WiFi e Bluetooth, per la gestione di una o più matrici collegate in cascata, utilizzando l'ormai noto modulo ESP32 e, più esattamente, la serie di moduli basata su questo SoC.

I VANTAGGI DI USARE UN ESP32

Ed è proprio grazie a questo piccolo modulo a microcontrollore che possiamo evitare l'utilizzo di componenti più impegnativi e costosi come le FPGA, da noi utilizzate in passato e che pure offrono prestazioni superiori nel controllo di matrici. Oggi vogliamo presentarvi questo progetto per gestire le LED matrix in maniera efficace, potendo creare animazioni fluide e avendo la possibilità di controllare tutto quanto in modo facile, ma

soprattutto potendo utilizzare le capacità di un microcontrollore come ESP32 che ci permette di collegarci ad internet, avendo così un modo non solo per gestire le matrici LED ma anche per integrarle in sistemi più complessi come l'IoT (Internet of Things).

Possiamo inoltre gestire la rotazione di una matrice tramite sensori specifici come, ad esempio, un giroscopio/accelerometro, gestire la luminosità in modo automatico tramite sensori LDR, fotoresistenze e simili, creare orologi personalizzati, mostrare immagini personalizzate, immagini GIF, animazioni 3D e con la possibilità di programmare tutto quanto tramite Arduino IDE, per i più appassionati, e tramite ESP-IDF/PlatformIO per i più esperti.

Insomma, il motivo per cui abbiamo scelto di utilizzare le schede ESP32 ormai vi sarà chiaro e oltre a quanto detto prima, tra gli altri vantaggi abbiamo un basso consumo di energia, tanta memoria a disposizione e infine il protocollo wireless Esp-Now, che permette di gestire più schede ESP32 e di conseguenza più matrici.

Quindi è possibile creare molti progetti smart rispetto ad utilizzare altri sistemi per la gestione dei display, ed essendo un sistema più compatto rispetto ad altri, è anche possibile creare facilmente un contenitore che vada a racchiudere tutta l'elettronica.

ESP32 COME CONTROLLER

Per quanto la ESP32 sia un'ottima scheda a microcontrollore, c'è comunque bisogno di convivere con i numerosi cavetti da dover utilizzare per gestire una matrice LED; per evitare di tirare troppi fili e limitare il cablaggio a un flat-cable standard per le matrici, dal momento che la ESP32 è una scheda molto compatta, abbiamo creato un circuito stampato che possa ospitarla e che sia dotato di

un connettore a vaschetta a passo 2,54 mm da 2x8 poli.

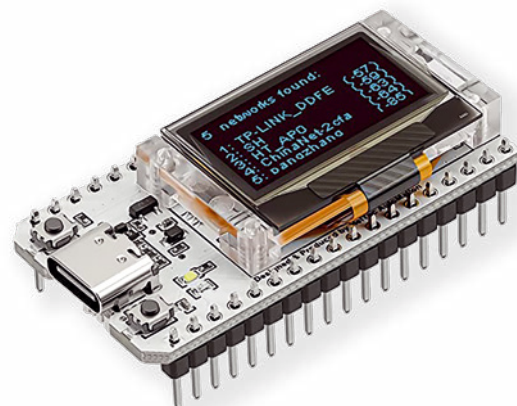
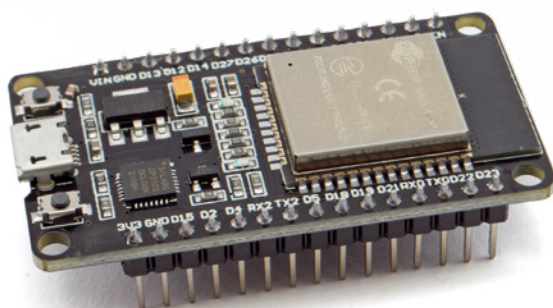
Con questa soluzione plug-and-play non sarà più necessario utilizzare cavi e cavetti impazzendo per capire cosa è collegato dove: basterà semplicemente innestare la ESP32 nel circuito stampato della scheda controller e quest'ultima alla matrice, o matrici se ne mettete più d'una in cascata, per realizzare il vostro visualizzatore LED Matrix.

Per rendere il controller più versatile, abbiamo previsto la possibilità di montare più versioni di schede ESP32, ossia la YB555 commercializzata dalla Futura Elettronica (anche on-line su www.futurashop.it) oppure la ESP32 dotata di un piccolo display OLED da 0,96" (anche questa commercializzata dalla Futura Elettronica con il codice prodotto ESP32OLED).

La prima è una compatta board (open source) di sviluppo e prototipazione basata sull'ESP32 (il SoC è, per l'esattezza, l'ESP-WROOM-32) che integra un transceiver WiFi 802.11 b/g/n, un ricetrasmittitore Bluetooth dual-mode (classico e BLE) e dispone di 25 linee GPIO. Supporta una velocità di trasmissione dati fino a 150 Mbps, ha una potenza di uscita sull'antenna di 20,5 dBm, per garantire la massima portata RF; il consumo in modalità sleep è di appena 5 µA. Dispone inoltre di interfacce per sensori di temperatura, touch sensor, SD card, UART, SPI, SDIO, I²C, LED PWM, Motor PWM, I²S, IR, connettore micro USB, pulsante di boot e pulsante di reset.

La versione ESP32OLED è ancora una piattaforma di sviluppo basata sul System on Chip ESP32 della Espressif, ma dotata di processore Dual-Core Tensilica LX6 a 32 bit con una frequenza di clock di 240 MHz, una memoria SRAM integrata da 520 kB (una capacità notevole e superiore alla media dei moduli ESP32) interfaccia wireless WiFi 802.11 b/g/n, Bluetooth dual-mode (classico e BLE) e

→ I due moduli utilizzabili con la nostra scheda.



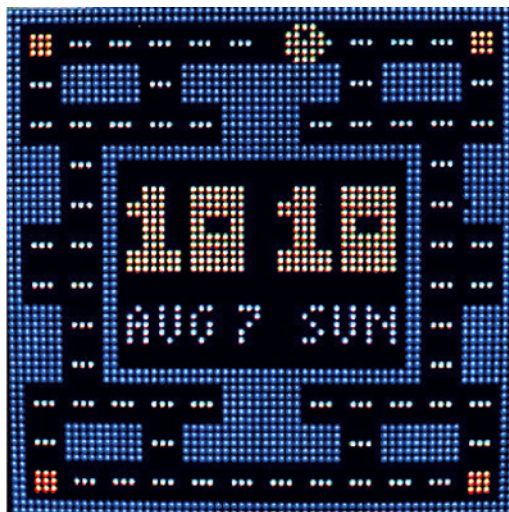
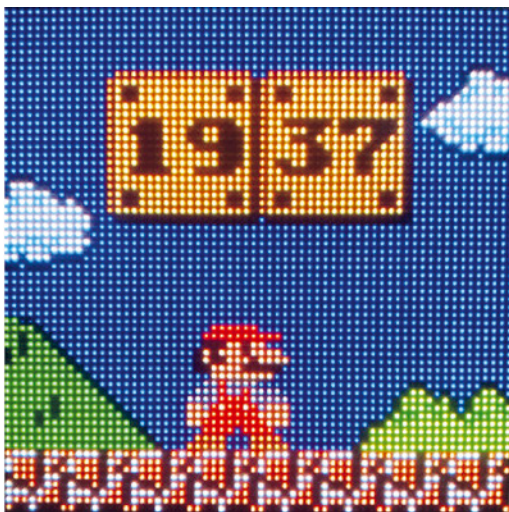


Fig. 1
Con le matrici
è possibile
creare sfondi
animati tipo
videogame.

28 linee di I/O generico (GPIO).

Sulla scheda è presente anche una memoria flash da 32 MB, un'antenna WiFi su circuito stampato, un display OLED da 0,96", una sezione caricabatteria per batterie ai polimeri di litio, un convertitore USB-seriale CP2102, due pulsanti (uno per il reset e l'altro per GPIO 0) e 2 LED di stato. Ideale per applicazioni IoT (Internet of Things), può essere programmata tramite l'IDE di Arduino, come del resto la ESP32 "normale".

SCHEMA ELETTRICO

Per quanto riguarda lo schema elettrico della nostra scheda controller, lo trovate in queste pagine e, come vedete, è molto semplice: tutto quanto è presente viene alimentato a 5 volt e nel circuito sono presenti una morsettiere e uno strip per l'alimentazione, un connettore a vaschetta per la connessione tramite flat-cable alla matrice, nonché un pin-strip per portare all'esterno il bus I²C con la possibilità di selezionare tramite l'apposito jumper (siglato JV_I2C) la tensione di alimentazione fra 3,3V e 5V. Notate che i 3,3V vengono generati da un regolatore a bordo dell'ESP32 (a prescindere dal modello) e resi disponibili tramite l'omonimo pin. Lo strip da 4 pin per la comunicazione I²C è stato previsto per gestire eventuali sensori e moduli; il jumper serve per selezionare l'alimentazione positiva, così da poter utilizzare ad esempio sensori di temperatura, moduli RTC (ad esempio per gestire da firmware e visualizzare l'ora e la data sulla matrice a LED) e molto altro, funzionanti a 3,3 o a 5V: se vi connettete dispositivi e breakout board a 5V spostate JV_I2C su 5V, mentre se ciò che affaccerete sul bus richiederà 3,3 volt, portate il jumper su 3V3. Al pin-strip giungerà, in base all'impostazione

del ponticello, la tensione di ingresso 5V o quella a 3,3V fornita dal modulo ESP32.

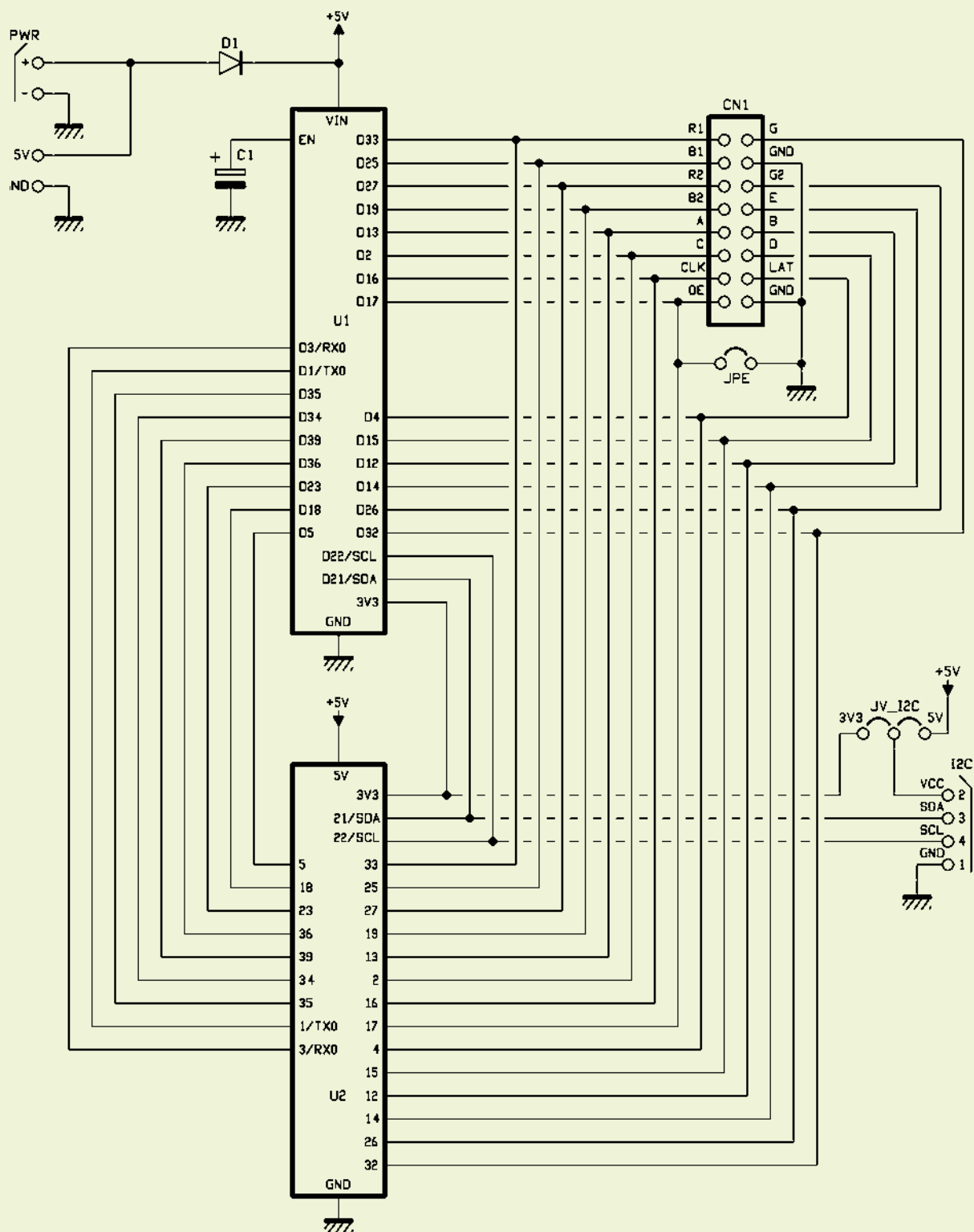
Nello schema elettrico vedete due integrati siglati U1 e U2: il primo è l'ESP normale e il secondo quello con OLED; chiaramente ne va montato uno solo, ossia quello che sceglierete di volta in volta.

Notate il condensatore elettrolitico posto sulla linea /EN dell'U1, che permette di caricare il codice su ESP32 all'accensione del circuito (ossia quando fornite alimentazione) senza dover tener premuto il tasto boot: il condensatore, inizialmente scarico, nell'istante di applicazione dei 5V è un cortocircuito e trascina /EN a massa, caricandosi poi gradualmente per effetto del resistore di pull-up internamente connesso a tale linea.

Sempre in tema di alimentazione, abbiamo posto un diodo di protezione in serie al pin VIN della scheda ESP32, quindi sulla linea positiva di alimentazione in arrivo dal contatto +PWR, per evitare di alimentare le matrici tramite lo stesso modulo ESP32, rischiando di danneggiare il microcontrollore. Infatti, le matrici vanno rigorosamente alimentate a 5 volt tramite un alimentatore esterno, possibilmente abbastanza potente in base a quante unità volete usare e a quanto consumano, perché non possono essere assolutamente alimentate dall'ESP32.

I terminali per l'alimentazione dei pannelli vanno avvitati sul circuito negli appositi slot tenendo conto della corretta polarità, infine, sul circuito è presente un jumper per collegare o meno il pin "E" a massa: questo perché alcune matrici lo utilizzano direttamente, mentre altre no.

Per concludere con la spiegazione del circuito, specifichiamo che ci sono due modi per collegare la matrice alla nostra scheda, il primo dei quali



consiste nel saldare sulla scheda controller stessa un connettore maschio e utilizzare un cavo flat femmina/femmina per l'interconnessione; in alternativa potete saldare un connettore femmina

e collegare la scheda controller direttamente alla matrice, ma in quest'ultimo caso il connettore femmina andrà saldato sotto la scheda e non sopra come avviene con il connettore maschio.

TIPOLOGIA MICROCONTROLLORE	NOTE
ESP32 originale	modulo ESP-WROOM-32 con chip ESP32-D0WDQ6 del 2017. L'MCU ha 520 kB di SRAM, che è superiore a quella di tutti i modelli recenti.
ESP32-S2	-
ESP32-S3	-

Tabella 1
Moduli Esp32 supportati.

COMPATIBILITÀ DI CHIP DRIVER, PANNELLI E SCHEDE ESP32

Le tipologie di matrici, chip driver e schede ESP32 supportate ufficialmente dalla libreria di gestione che utilizzeremo e di conseguenza dalla nostra scheda controller, sono dettagliati nella **Tabella 1** e nella **Tabella 2**. I modelli di scheda ESP32 V-RISC (come il C3) non sono supportati e non lo saranno mai poiché non hanno un output DMA parallelo, che è richiesto dalla libreria.

Esistono alcuni chip driver per matrici LED che sono compatibili con la libreria in uso, ossia ICND2012, RUC7258, SM5266P ed FM6126A, quest'ultimo noto come ICN20385 o FM6124 (PatternPlasma per il corretto utilizzo, fate riferimento all'esempio).

I pannelli noti per non funzionare con la libreria in uso sono invece quelli a scansione 1/8, i pannelli basati su RUL5358/SHIFTREG_ABC_BIN_DE e i pannelli basati su CN2053/FM6353: per quest'ultimo bisogna utilizzare in alternativa la libreria reperibile all'indirizzo Internet <https://github.com/Lautour/ESP32-HUB75-MatrixPanel-DMA-ICN2053>. Per utilizzare i pannelli non supportati si consiglia di utilizzare un'altra libreria, scaricabile dal web all'indirizzo <https://github.com/2dom/PxMatrix>.

REALIZZAZIONE PRATICA

Vediamo ora le note costruttive e di utilizzo del progetto, partendo dalla preparazione della scheda controller, per la quale è previsto un PCB da preparare per fotoincisione dopo aver ricavato le tracce delle due facce da stampe di buona qualità

(su carta da lucido o acetato) delle rispettive tracce lato rame disponibili per il download sul nostro sito Internet www.elettronica.in.it insieme ai file di questo progetto.

Inciso e forato lo stampato, montatevi i pochi elementi occorrenti iniziando con il diodo D1 (da disporre orientato come si vede nel piano di montaggio) e gli strip femmina con cui costruire gli "zoccoli" per i moduli ESP, quindi passate allo strip maschio da 4 pin per l'estensione I²C-Bus e infine montate il connettore CN1 da 16 poli su 2 file a passo 2,54x2,54 mm e la presa jack DC da circuito stampato per l'alimentazione.

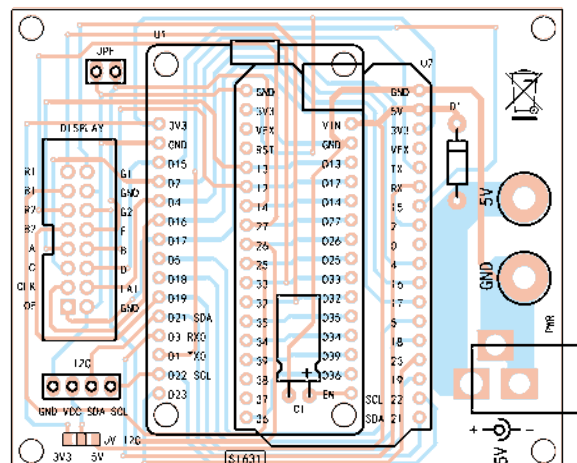
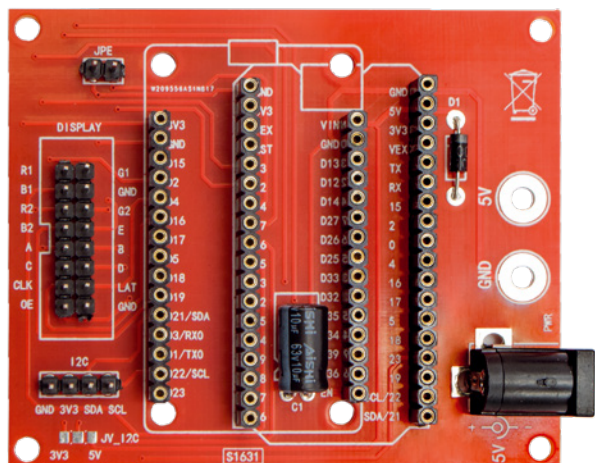
A questo punto, dopo una rapida verifica del montaggio e di eventuali saldature fredde o "baffi" di stagno che possono creare cortocircuiti tra piste e piazzole attigue, passate a innestare la board ESP32 da voi scelta nell'apposito slot (riferitevi alla **Fig. 2**); bisognerà poi collegare il connettore di alimentazione, che viene fornito con la matrice, andando prima ad avvitare i terminali capocorda del cavo alle rispettive piazzole GND e 5V del PCB (**Fig. 3**) e infine collegando il connettore allo slot presente sulla matrice verificando bene la polarità, infine potete collegare la scheda controller alla matrice.

Rammentate che alla scheda è consigliato di collegare una sola matrice 64x64, perché il jack di alimentazione è dimensionato per una corrente che corrisponde a quella della stessa e della scheda controller; eventuali ulteriori matrici andranno alimentate con connessioni esterne al PCB. In questa fase non è necessario posizionare il ponticello del pin-strip per il bus I²C, giacché non

TIPOLOGIA PANNELLO	SCANSIONE	NOTE
Pannello 64x32 pixel	scansione 1/16	-
Pannello 64x64 pixel	scansione 1/32	-
Pannello 32x16 pixel	scansione 1/4	vedere esempio 32x16_1_4_ScanPanel
Pannello 126x64 SM5266P	scansione 1/32	-

Tabella 2
Le matrici RGB supportate.





Elenco Componenti:

C1: 10 μ F 63 VL elettrolitico
 D1: 1N4007
 U1: Modulo ESP32
 U2: Modulo ESP32 con OLED

Varie:
 - Strip femmina 15 vie (2 pz.)
 - Strip femmina 18 vie (2 pz.)
 - Strip maschio 2 vie

- Strip maschio 4 vie
 - Jumper
 - Plug alimentazione
 - Circuito stampato S1631 (73x61 mm)

stiamo utilizzando alcuna periferica del genere. A questo punto che tutto è collegato procediamo con il codice e la libreria in uso, che trovate alla pagina GitHub <https://github.com/Dario-Ciceri/Futura-Elettronica-Esp32-Led-Matrix>, e vediamo come funzionano.

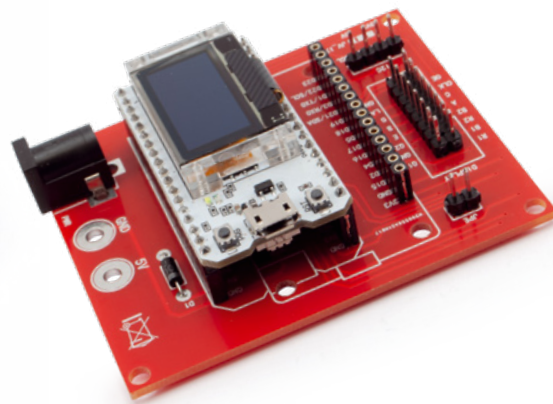
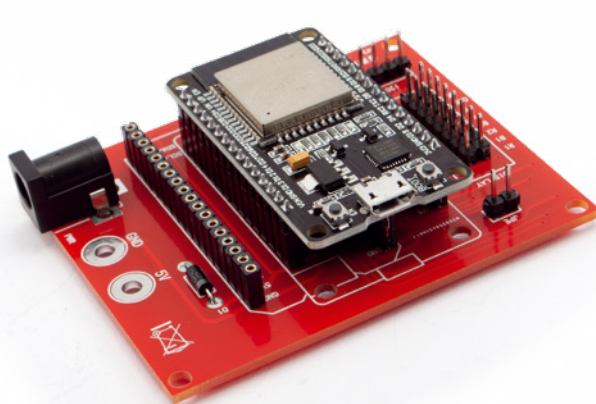
LA LIBRERIA DI GESTIONE

Per lavorare con le matrici a LED RGB andremo ad utilizzare una nota libreria implementata per l'impiego dell'ESP32 nella gestione di matrici LED; questa libreria la potete trovare all'indirizzo Internet <https://github.com/mrfaptastic/ESP32-HUB75-MatrixPanel-DMA>.

I vantaggi che offre questa libreria sono notevoli:

1. basso carico della CPU; una volta che i dati dei pixel vengono inizializzati vengono trasmessi agli input della matrice tramite il motore DMA direttamente dalla memoria;
2. l'aggiornamento dei dati dei pixel comporta solo la logica bit a bit sul buffer DMA, nessuna manipolazione dei pin o blocco dell'I/O;
3. BCM a schermo intero; utilizza la modulazione del codice binario per la profondità/luminosità del colore dei pixel sull'intera matrice;
4. profondità di colore variabile; è possibile un output fino a TrueColor a 24 bit a seconda

Fig. 2
 Posizionamento dei moduli ESP: a sinistra quello semplice e a destra la versione OLED.



- delle dimensioni della matrice e della frequenza di refresh richiesta;
- 5. correzione della luminanza CIE 1931, anche nota come attenuazione naturale dei LED;
- 6. la libreria può essere utilizzata con AdafruitGFX, simplified GFX o senza l'API GFX.

L'unico "svantaggio" di questa libreria è di essere vincolata all'utilizzo di determinate risorse fisiche, il che la rende compatibile solo con un hardware selezionato.

CALCOLARE LE RISORSE IMPEGNATE

Una cosa molto interessante di questa libreria e più in generale delle matrici LED è che ci permettono di mettere in serie più matrici, andando a creare una catena. Tuttavia, se una matrice possiamo gestirla senza troppi problemi due o più matrici potrebbero occupare troppe risorse e il nostro ESP32 potrebbe non avere abbastanza memoria per poterle gestire. Per dirimere il dubbio esiste un'ottima risorsa, che potete trovare al link Internet <https://github.com/mrfaptastic/ESP32-HUB75-MatrixPanel-DMA/blob/master/doc/memcalc.xlsm>, la quale permetterà di calcolare se l'ESP32 sarà in grado o meno di soddisfare i requisiti del nostro progetto (Fig. 4).

In linea generale ci sono due risorse principali che vengono utilizzate per pilotare le matrici LED e sono la memoria e la bus clock speed; quest'ultima, sostanzialmente corrispondente alla velocità di scambio dati raggiungibile sul bus, possiamo considerarla come la larghezza di banda disponibile per trasferire i dati relativi al colore di ogni pixel. I fattori principali che vanno a determinare la quantità occupata delle due risorse dette prima sono:

1. la risoluzione della matrice (numero di pixel);



Fig. 3
Cablaggio con occhielli capocorda per alimentare la matrice RGB.

2. il numero di pannelli concatenati / in serie;
3. la profondità del colore dei pixel;
4. la transizione da BCM (modulazione del codice binario) LSB a MSB;
5. doppio buffer di memoria.

Questi fattori determinano la frequenza di aggiornamento, ciò può influire sulla fluidità delle animazioni e sui tempi di risposta dell'intero sistema.

UTILIZZARE LA LIBRERIA PER IL CONTROLLO DELLE MATRICI

Ora che abbiamo visto vantaggi e svantaggi della libreria in uso, vediamo come utilizzarla. Per prima cosa avremo bisogno della libreria

Fig. 4
Il tool per calcolare le risorse occorrenti ai progetti.

ESP32 HUB75 I2S Calculator			Pls, enable macro's to calculate RefreshRate and Optimal color depth		
Input Change values as required			Output		
Module Width	64 px		Total pixels	24,576 pixels	
Module Height	64 px		RGB24 Buffer size	73,728 bytes	
Chain length	6 pcs		DMA Buffer size	196,608 bytes	
Colordepth	8 bits		LSB2MSB bit	3 bit	
Min refresh rate	90 Hz		RefreshRate	143.61213 Hz	
Double buffer	0 true/false		Optimal colordepth	6	
I2S Clock speed	10 MHz		HUB75 Color data transfer rate	60 Mbit/s	
			Likely outcome (enable debugging to confirm!)		
			Warning: Not enough RAM for requested input parameters!		

Adafruit_GFX, che è possibile scaricare direttamente dal gestore delle librerie dell'IDE Arduino; in alternativa è possibile cercarla su Internet aiutandovi con la ricerca di Google o altro motore di ricerca. Per quanto riguarda la configurazione dei pin, per la driver board abbiamo una configurazione non standard e ciò significa che quella utilizzata nei vari esempi contenuti nella libreria ESP32-HUB75-MatrixPanel-DMA non sarà compatibile. Dovremo quindi utilizzare la configurazione proposta qui di seguito:

```
#define R1_PIN 33
#define G1_PIN 32
#define B1_PIN 25
#define R2_PIN 27
#define G2_PIN 26
#define B2_PIN 21
#define A_PIN 13
#define B_PIN 12
#define C_PIN 2
#define D_PIN 15
#define E_PIN 14
#define LAT_PIN 4
#define OE_PIN 17
#define CLK_PIN 16
```

Per inizializzare la matrice con questa configurazione possiamo procedere come mostrato nel codice contenuto nel **Listato 1**. Volendo spiegare sinteticamente cosa fanno le righe di codice riportate in questo listato, possiamo dire che vanno ad inizializzare una nuova matrice con la configurazione dei pin ivi definita, che abbiamo creato appositamente per la driver board, riassegnando quelli predefiniti della libreria. Il codice contenuto nel **Listato 1** verifica, quindi, se i collegamenti funzionano tramite la funzione `begin(R1_PIN,`

Guarda il LED MATRIX in azione su YouTube!



Vuoi vedere in azione il nostro LED MATRIX? Nulla di più semplice: collegati alla pagina <https://youtu.be/5Bim6QX78DQ> Potrai anche vedere i video di moltissimi altri nostri progetti proposti in passato.

`G1_PIN, [...]]`, in caso di problemi viene stampata sul monitor seriale una stringa di errore. Per configurare una o più matrici possiamo utilizzare la struttura presente nel predetto listato, composta dai parametri `mxconfig.mx_width` e `mxconfig.mx_height` che definiscono rispettivamente larghezza e altezza della matrice o delle matrici, mentre `mxconfig.chain_length` definisce il numero di pannelli in serie. Il parametro `mxconfig.gpio.e` serve per indicare il pin "E", utilizzato da particolari pannelli con scansione 1/32 come i 64x64 pixel.

↓ Listato 1

```
HUB75_I2S_CFG mxconfig;
mxconfig.latch_blanking = 4;
mxconfig.i2sspeed = HUB75_I2S_CFG::HZ_10M;
mxconfig.clkphase = false;
mxconfig.mx_width = 64;
mxconfig.mx_height = 64;
mxconfig.chain_length = 1;
mxconfig.gpio.e = E_PIN;
dma_display = new MatrixPanel_I2S_DMA(mxconfig);
if ( not dma_display->begin(R1_PIN, G1_PIN, B1_PIN, R2_PIN, G2_PIN, B2_PIN, A_PIN, B_PIN, C_PIN, D_PIN, E_PIN, LAT_PIN, OE_PIN, CLK_PIN ))
Serial.println("***** !KABOOM! I2S memory allocation failed *****");
```



Matrice 16x32
connessa alla
scheda controller.

La riga di codice `dma_display = new MatrixPanel_I2S_DMA(mxconfig);` crea la matrice e sovrascrive la nostra configurazione sostituendola a quella predefinita (default).

I comandi per la grafica, come ad esempio le primitive grafiche (rettangoli, cerchi, triangoli ecc), per il colore del testo e le font per i testi, vengono gestiti principalmente dalla libreria *Adafruit_GFX*, della quale potete trovare facilmente la documentazione su Internet.

Per farvi un esempio, se volessimo mostrare a pannello un cerchio di raggio 10, di colore blu, con il centro in coordinate $x = 10$ e $y = 10$ ci basterebbe la seguente riga di codice:

```
dma_display->drawCircle(10, 10, 10, dma_display-
>color444(0, 0, 15));
```

che altro non è se non la funzione `drawCircle` della libreria *Adafruit_GFX*:

```
void drawCircle(uint16_t x0, uint16_t y0, uint16_t r,
uint16_t color);
```

Notate bene che l'istruzione `dma_display->color444(0, 0, 15)` costituisce solo uno dei modi che abbiamo a disposizione per definire di che colore vogliamo che siano i pixel che la funzione `drawCircle` andrà ad utilizzare; per raggiungere lo scopo, possiamo infatti utilizzare uno qualsiasi dei metodi supportati dalla libreria *Adafruit_GFX*.

CONCLUSIONI

Possiamo concludere qui questa panoramica sulle matrici LED e sul loro utilizzo in abbinamento ai

popolari moduli ESP32, aggiungendo che esistono molteplici progetti da poter realizzare con questo fantastico hardware e che le matrici sono facilmente integrabili in progetti di vario tipo, che vanno dalla semplice gestione di interfacce ai videogiochi (anche interattivi) per finire -poi- nel mondo dell'IoT.

Ci sono davvero pochi limiti a quello che possiamo realizzare e una volta presa confidenza con le librerie del caso, saremo in grado di realizzare progetti sempre più complessi e particolari.

La scheda che abbiamo proposto, inoltre ci permette di semplificare la gestione dei cavi e di avere un'unica piattaforma di sviluppo facilmente integrabile in progetti compatti.

Infine, un consiglio che vogliamo darvi è di sbirciare sul web la mole di progetti disponibili, dedicati a videogiochi degli anni '80 del secolo scorso, come pac-man, tetris, space invaders, pong e altri dedicati a smart home e ambienti, come orologi di Super Mario, il nostro orologio Tetris o pannelli smart per bitcoin, news e molto altro ancora.

Buon lavoro!



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica.

La board controller per matrici a LED (cod. FT1631K) è in vendita in scatola di montaggio a Euro 14,00, il modulo ESP32 (cod. YB555) a Euro 13,50 mentre il modulo ESP con display OLED (cod. ESP32OLED) a Euro 29,50.

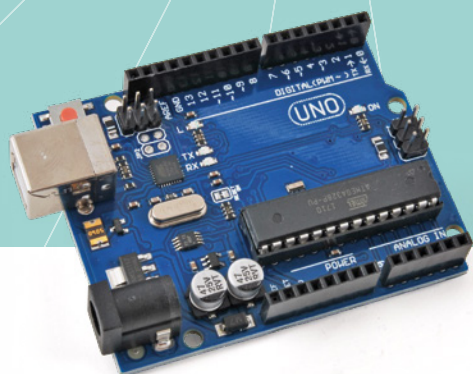
Le matrici a LED sono disponibili in diversi formati alla pagina futura.net.it/tag-prodotto/matrici-led. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

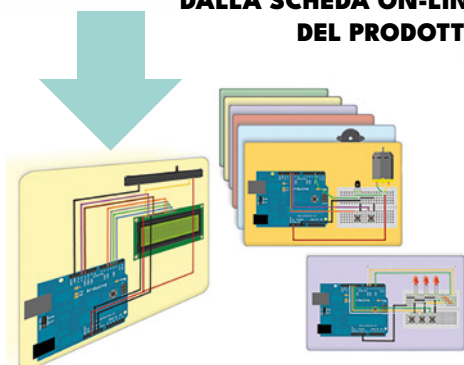
Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 - www.futurashop.it

Programmazione alla portata di tutti!

Sei un maker, uno studente
oppure un hobbista e vuoi
programmare subito
e facilmente?
Questo è il kit che
fa per te!



**NUMEROSI PROGETTI DA REALIZZARE
SONO SCARICABILI **GRATUITAMENTE**
DALLA SCHEDA ON-LINE
DEL PRODOTTO**



cod. ELEARDUKIT1

€39,00
IVA INCLUSA

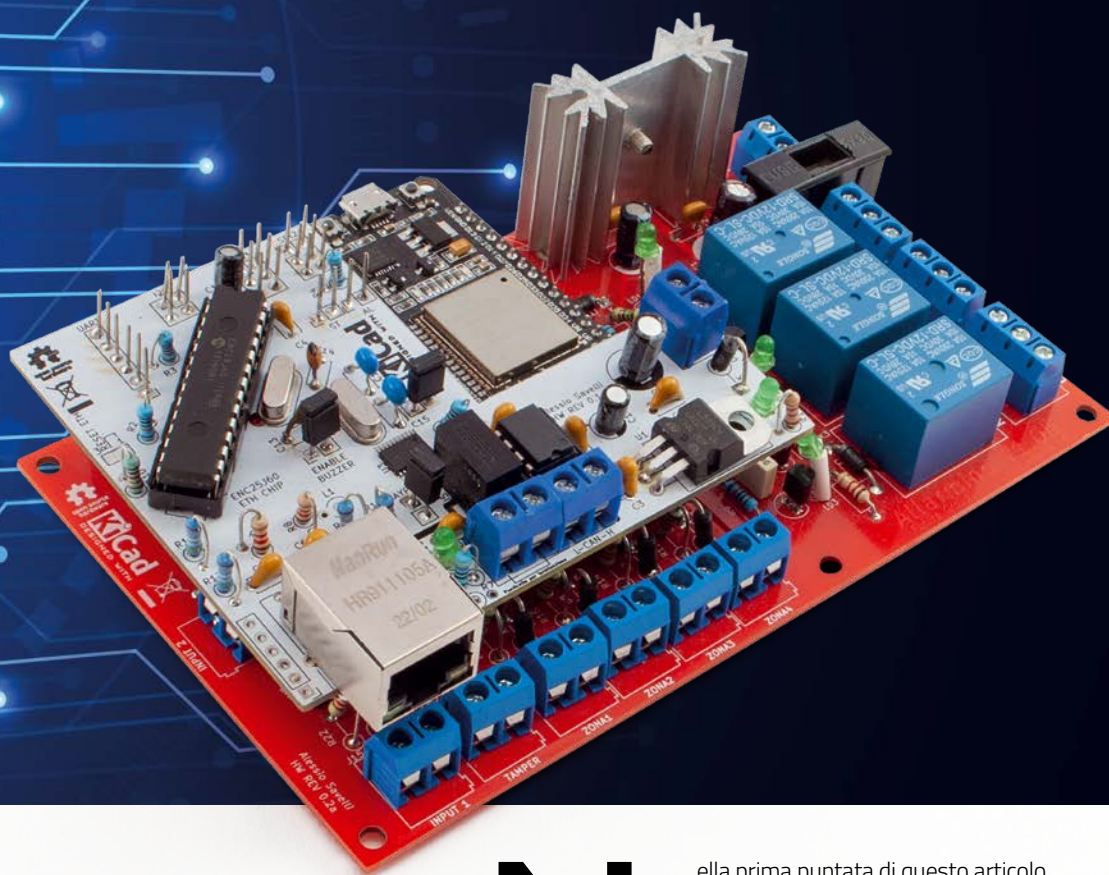
STARTER KIT CON UNO R3

Starter kit destinato a maker, studenti e hobbisti
che vogliono iniziare a programmare con il linguaggio
di programmazione di Arduino.

Il kit comprende la board UNO R3 (una scheda
di sviluppo compatibile con Arduino UNO basata
sull'ATmega328P) e tutti i componenti necessari
per sperimentare e programmare.

ESPANSIONE PER ALLARMINO ESP32

di ALESSIO SAVELLI

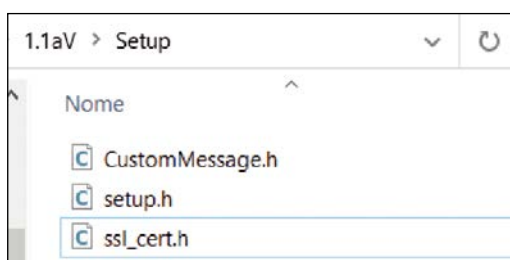


Nuovo firmware e software di gestione dell'espansione ethernet per Allarmino IoT. Seconda e ultima puntata.

Nella prima puntata di questo articolo abbiamo introdotto un modulo di espansione per il nostro progetto Allarmino IoT (proposto nel fascicolo n° 264 di Elettronica In) finalizzato ad aggiungere la connettività ethernet e, all'occorrenza, un'interfaccia CAN-Bus. Queste due interconnessioni vanno a completare quelle native di Allarmino IoT: WiFi e Bluetooth Low Energy; questo, per sopperire alla necessità di avere connessioni stabili in situazioni in cui il link wireless risulta impraticabile o inaffidabile. L'espansione consiste in un circuito stampato che, mediante connettori montati sulla faccia inferiore, si inserisce sulla board Allarmino, tramite connettori di quest'ultima.

Nella prima parte abbiamo spiegato in cosa consiste questo hardware, accennando al fatto che per renderlo più versatile

Fig. 1
File cartella Setup
del Firmware.



il software per configurare il sistema, per gli utenti Windows, i quali devono comunque avere installato l'IDE di Arduino e i driver di terze parti delle board ESP32 (le librerie aggiuntive usate dal firmware sono integrate nel software e quindi non è necessario scaricarle). In alternativa forniamo un container gestibile con Docker, che configura ed compila il firmware a partire da un file json.

è stata prevista la possibilità di scegliere tra due moduli ethernet il W5500 e l'ENC25J60. L'espansione è associata a una nuova versione firmware, la quale integra l'utilizzo dei moduli compatibili con lo stack LwIP TCP/IP dell'ESP32 e introduce una nuova funzionalità basata su un BOT Telegram. Il firmware è stato già descritto, quindi ora vediamo

IL SOFTWARE DI GESTIONE

Ci siamo resi conto che il firmware è diventato molto complesso perché presenta innumerevoli parametri da configurare, quindi vi proponiamo un software appositamente realizzato per configurare i parametri della centralina e compilare correttamente il codice sorgente di Allarmino in modo che non

COMANDO

AZIONE

[autopatch_arduino-builder]	Ottiene il percorso del file 'arduino-builder.exe'
[autopatch_arduino-hardware]	Ottiene il percorso della cartella 'hardware' di Arduino
[autopatch_tools-builder]	Ottiene il percorso della cartella 'tools-builder' di Arduino
[autopatch_tools\avr]	Ottiene il percorso della cartella '\hardware\tools\avr' di Arduino
[autopatch_arduino-packages]	Ottiene il percorso della cartella '\Arduino15\packages'
[autopatch_built-extra-libraries]	Ottiene il percorso delle librerie presente nei documenti ('\Arduino\libraries')
[speed_esp32]	Ottiene il baud rate configurato nelle impostazioni di compilazione
[autopatch_build]	Imposta il percorso di output del file compilato (di default li mette nella cartella del progetto)
[autopatch_cache]	Imposta il percorso di cache del file compilato (di default li mette nella cartella del progetto)
[autopatch_extensa-esp32]	Imposta la cartella dei tool estensa dell'ESP
[autovers_extensa-esp32]	Imposta la versione dei tool estensa disponibile per l'ESP
[autopatch_mkspiiffs]	Imposta la cartella dei tool dell'ESP per il makespiiffs
[autovers_mkspiiffs]	Imposta la versione dei tool disponibile sull'ESP per il makespiiffs
[autopatch_esptool]	Imposta la cartella dei tool esptool disponibile per l'ESP
[autovers_esptool]	Imposta la versione dei tool esptool disponibile sull'ESP
[autopatch_soruce.ino]	Imposta il percorso dove si trova il codice sorgente da compilare
[serialport_esp32]	Imposta la porta seriale configurata nelle impostazioni del compilatore
[autopatch_boot_app]	Imposta i relativi percorsi per i file bin da caricare sull'ESP
[autopatch_bootloader_qio]	Imposta i relativi percorsi per i file bin da caricare sull'ESP
[autopatch_main.ino.bin]	Imposta i relativi percorsi per i file bin da caricare sull'ESP
[autopatch_main.ino.partitions.bin]	Imposta i relativi percorsi per i file bin da caricare sull'ESP

Tabella 1
Descrizione dei
comandi.



dobbiate intervenire manualmente sul firmware stesso. Tale software diventerà importante per future versioni di Allarmino, poiché sarà l'unico modo per creare un file **.bin** valido da poter inviare su Telegram e quindi gestire l'aggiornamento da remoto. Una volta scaricato e installato il software dal nostro sito, andate nella cartella d'installazione dove troverete il file "AllarminoIDE.exe" che è l'eseguibile da avviare o da GitHub al link <https://github.com/Alessio Savelli/Allarmino/tree/main/Software/Relase> (dal link possono seguire gli aggiornamenti futuri del software). Nel caso in cui scriverete un vostro Firmware custom potrete inserire il codice nella cartella "Resource/Firmware", dargli un nome e seguire il template della cartella 1.1av. Nella cartella 'Source' dovrà essere copiata la cartella del Firmware che avete customizzato tenendo bene in mente che dovrà chiamarsi 'main' e contenere il file 'main.ino'. Nella cartella "Setup" dovranno essere presenti esclusivamente i file che il software dovrà andare a modificare per la personalizzazione delle impostazioni (**Fig. 1**); il software conosce come modificare i parametri standard dei firmware proposti da noi, ulteriori file o #define aggiunti dovranno essere riconfigurati manualmente e salvati direttamente nella cartella "Source/main". Attualmente il software non è in grado di aggiornare le librerie aggiuntive e quindi dovranno essere aggiornate a mano, facendo il download da GitHub.

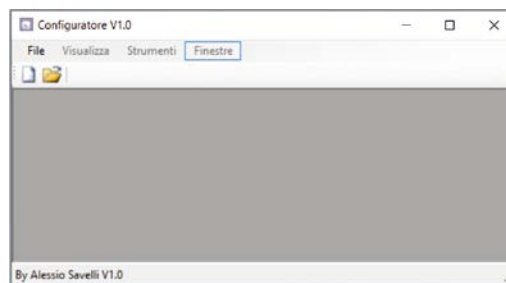


Fig. 2
Schermata iniziale del software.

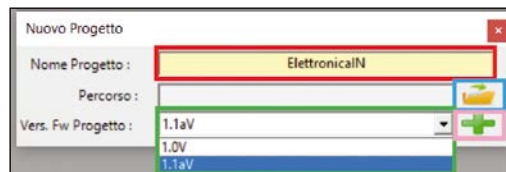


Fig. 3
Schermata nuovo progetto.

Si può gestire interamente la configurazione dei parametri di compilazione tramite il file "confcomp.ini" presente nella cartella d'installazione e riportato nel **Listato 1**. Per inserire un commento basta digitare il simbolo '#' a inizio riga, il file di configurazione mette a disposizione la possibilità di usare l'auto compilazione dei parametri, in caso di problemi utilizzarne una personalizzata. I comandi messi a disposizione per l'auto-compilazione sono rappresentati in **Tabella 1**. Detto ciò, avviamo il file "AllarminoIDE.exe" il quale si presenta come in **Fig. 2**; cliccando su "File->Nuovo" vedremo comparire la schermata

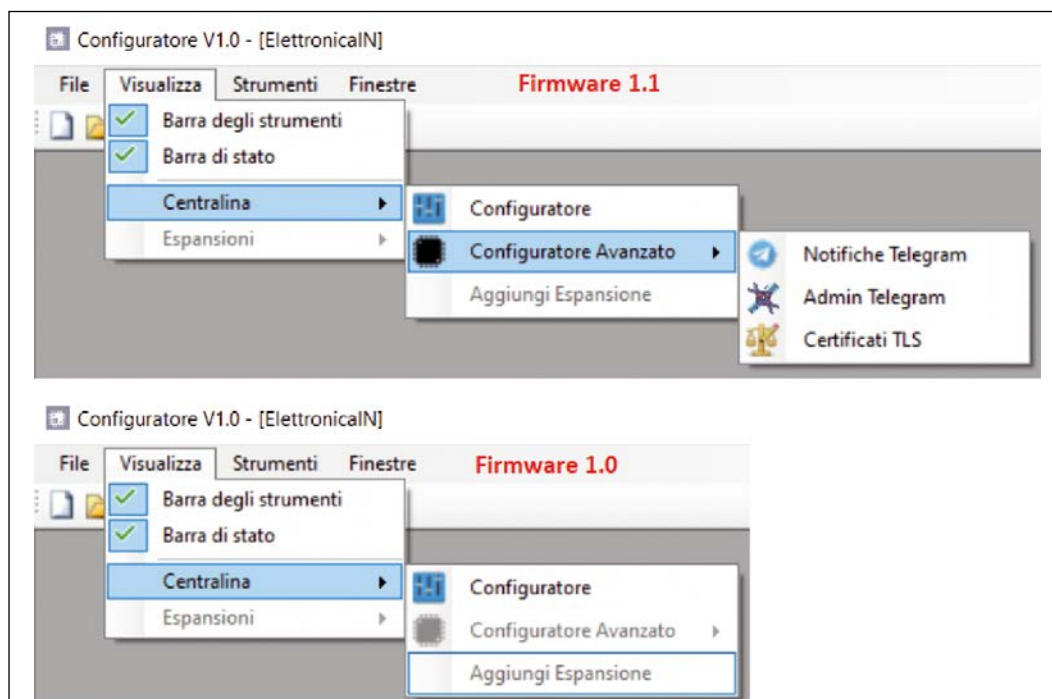


Fig. 4
Differenze tra tool messi a disposizione a seconda del Firmware scelto.

▼ Listato 1

```
#@usecustombuild #abilitando @usecustombuild si utilizzeranno i parametri custom
arduino-builder="[autopatch_arduino-builder]"
precompileparms="--dump-prefs -logger=machine
compileparms="--compile -logger=machine
-hardware "[autopatch_arduino-hardware]"
-hardware "[autopatch_arduino-packages]"
-tools "[autopatch_tools-builder]"
-tools "[autopatch_tools\avr]"
-tools "[autopatch_arduino-packages]"
-built-in-libraries "[autopatch_built-in-libraries]"
-libraries "[autopatch_built-extra-libraries]"
-fqbn=esp32:esp32:esp32:PSRAM=disabled,PartitionScheme=default,CPUFreq=240,FlashMode=qio,FlashFreq=80,FlashSize=4M,UploadSpeed=[speed_esp32],DebugLevel=none
-vid-pid=10C4_EA60
-ide-version=10815
-build-path "[autopatch_build]"
-warnings=none
-build-cache "[autopatch_cache]"
-prefs=build.warn_data_percentage=75
-prefs=runtime.tools.xtensa-esp32-elf-gcc.path="[autopatch_extensa-esp32]"
-prefs=runtime.tools.xtensa-esp32-elf-gcc-[autovers_extensa-esp32].path="[autopatch_extensa-esp32]"
-prefs=runtime.tools.mkspiffs.path="[autopatch_mkspiffs]"
-prefs=runtime.tools.mkspiffs-[autovers_mkspiffs].path="[autopatch_mkspiffs]"
-prefs=runtime.tools.esptool_py.path="[autopatch_esptool]"
-prefs=runtime.tools.esptool_py-[autovers_esptool].path="[autopatch_esptool]"
-verbose "[autopatch_soruca.ino]"
[ESPTOOL_PY]
esptoolspath="[autopatch_esptool]"
esptoolparms="--chip esp32 --port [serialport_esp32] --baud [speed_esp32] --before default_reset --after hard_reset write_flash -z --flash_mode dio --flash_freq 80m --flash_size detect
esptoolboot=0xe000 "[autopatch_boot_app]"
esptoolbootloader_qio=0x1000 "[autopatch_bootloader_qio]"
main.ino.bin=0x10000 "[autopatch_main.ino.bin]"
main.ino.partitions.bin=0x8000 "[autopatch_main.ino.partitions.bin]"
#Builder con parametri custom - precompilato con esempio su windows
[CUSTOM PARMS]
arduino-builder="C:\Program Files (x86)\Arduino\arduino-builder.exe"
precompileparms="--dump-prefs -logger=machine
compileparms="--compile -logger=machine
-hardware "C:\Program Files (x86)\Arduino\hardware"
-hardware "C:\Users\USER\AppData\Local\Arduino15\packages"
-tools "C:\Program Files (x86)\Arduino\tools-builder"
-tools "C:\Program Files (x86)\Arduino\hardware\tools\avr"
-tools "C:\Users\USER\AppData\Local\Arduino15\packages"
-built-in-libraries "C:\Program Files (x86)\Arduino\libraries"
-libraries "C:\Users\USER\Documents\Arduino\libraries"
-fqbn=esp32:esp32:esp32:PSRAM=disabled,PartitionScheme=default,CPUFreq=240,FlashMode=qio,FlashFreq=80,FlashSize=4M,UploadSpeed=921600,DebugLevel=none
-vid-pid=10C4_EA60
-ide-version=10815
-build-path "C:\Users\USER\AppData\Local\Temp\arduino_build_702037"
-warnings=none
-build-cache "C:\Users\USER\AppData\Local\Temp\arduino_cache_76075"
-prefs=build.warn_data_percentage=75
-prefs=runtime.tools.xtensa-esp32-elf-gcc.path="C:\Users\USER\AppData\Local\Arduino15\packages\esp32\tools\xtensa-esp32-elf-gcc\1.22.0-97-gc752ad5-5.2.0"
-prefs=runtime.tools.xtensa-esp32-elf-gcc-1.22.0-97-gc752ad5-5.2.0.path="C:\Users\USER\AppData\Local\Arduino15\packages\esp32\tools\xtensa-esp32-elf-gcc\1.22.0-97-gc752ad5-5.2.0"
-prefs=runtime.tools.mkspiffs.path="C:\Users\USER\AppData\Local\Arduino15\packages\esp32\tools\mkspiffs\0.2.3"
-prefs=runtime.tools.mkspiffs-0.2.3.path="C:\Users\USER\AppData\Local\Arduino15\packages\esp32\tools\mkspiffs\0.2.3"
-prefs=runtime.tools.esptool_py.path="C:\Users\USER\AppData\Local\Arduino15\packages\esp32\tools\esptool_py\3.0.0"
-prefs=runtime.tools.esptool_py-3.0.0.path="C:\Users\USER\AppData\Local\Arduino15\packages\esp32\tools\esptool_py\3.0.0"
-verbose "[autopatch_soruca.ino]"
[ESPTOOL_PY]
#se si vuole usare una porta fissa basta sostituire [serialport_esp32] con il nome della porta, es. COM3
esptoolspath="C:\Users\USER\AppData\Local\Arduino15\packages\esp32\tools\esptool_py\3.0.0/esptool.exe"
esptoolparms="--chip esp32 --port [serialport_esp32] --baud 921600 --before default_reset --after hard_reset write_flash -z --flash_mode dio --flash_freq 80m --flash_size detect
esptoolboot=0xe000 "C:\Users\USER\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.6/tools/partitions/boot_app0.bin"
esptoolbootloader_qio=0x1000 "C:\Users\USER\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.6/tools/sdk/bin/bootloader_qio_80m.bin"
main.ino.bin=0x10000 "[autopatch_main.ino.bin]"
main.ino.partitions.bin=0x8000 "[autopatch_main.ino.partitions.bin]"
```

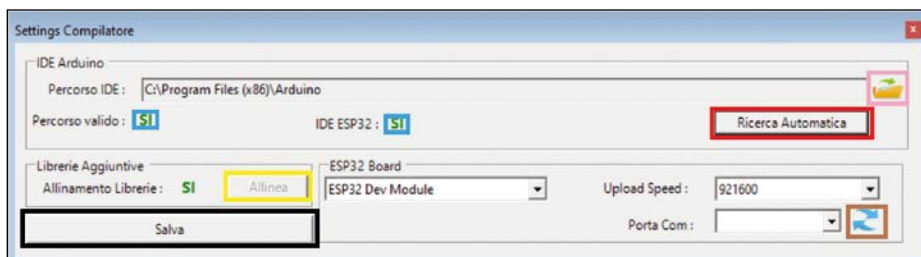


Fig. 5
Schermata di
configurazione
compilatore.

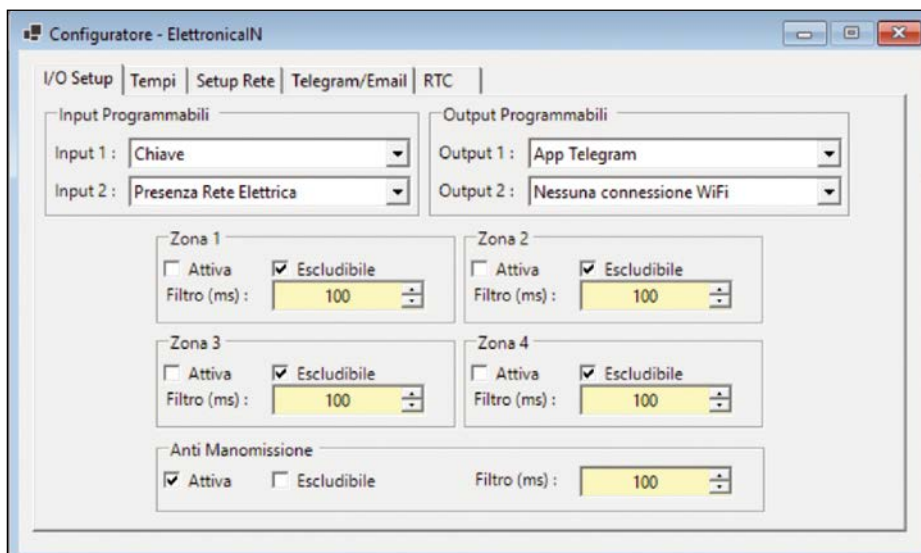


Fig. 6
Setup I/O di
Allarmino.

proposta nella **Fig. 3**, nella quale scriviamo il nome del nostro progetto nel riquadro evidenziato in rosso, dopo di che clicchiamo sull'immagine del riquadro azzurro e impostiamo una cartella nella quale organizzare i file del progetto.

Ora selezioniamo la versione Firmware dal menu a tendina mostrato nel riquadro verde e confermiamo con il pulsante "+" che vediamo nel riquadro rosa; ora, cliccando su "Visualizza", comparirà un menù a tendina con tutte le impostazioni modificabili per il Firmware scelto (**Fig. 4**). Prima di iniziare la configurazione della centrale, colleghiamo l'ESP32 al computer tramite il cavo USB e andiamo a cliccare su "Strumenti->Opzioni Compilatore" e configuriamo i parametri base.

In **Fig. 5** è mostrata la schermata di configurazione in cui abbiamo evidenziato il pulsante di ricerca automatica in rosso; cliccando questo pulsante il software inizierà a ricercare tutti i percorsi utili per interfacciarsi con l'IDE di Arduino (il software dispone di una versione portatile, che viene imposta di default insieme alle board installate). Se questa procedura va a buon fine, apparirà nei riquadri blu la scritta 'SI' in verde che indica la presenza dell'IDE di Arduino e la corretta installazione del board ESP32, in caso contrario la cartella di Arduino si può

impostare manualmente cliccando sul pulsante nel box rosa. È possibile installare nell'IDE di Arduino le librerie presenti nella cartella "Librerie"; per fare ciò, cliccate il pulsante "Allinea" evidenziato in giallo. Impostate il parametro "Upload speed" a 921600 e inserite la Porta COM dell'ESP32, se non la trovate cliccate sul pulsante evidenziato in marrone, il quale aggiornerà le porte COM disponibili; adesso fate clic su "Salva" (evidenziato in nero nella **Fig. 5**).

Tornate sul menu "Strumenti->Configuratore" (**Fig. 6**) dove configurerete in maniera grafica e semplificata tutti i parametri presenti nel file 'setup.h'.

In questo articolo ci soffermiamo sulle pagine 'Setup Rete' e 'Telegram/Email' in quanto le altre sono molto intuitive e facilmente comprensibili, ancor più se avete già letto l'articolo pubblicato sulla rivista n. 264 dedicato al progetto Allarmino.

Nella scheda 'Setup Rete' (**Fig. 7**) è possibile scegliere se utilizzare il WiFi o la connessione cablata ethernet; nel nostro caso, installando il modulo di espansione selezioniamo 'Ethernet' e inseriamo il MAC Address o generiamolo cliccando sul pulsante evidenziato in rosso, assicuriamoci che il chipset e la porta SPI selezionati siano corretti, come nei riquadri neri della foto in esempio.

Fig. 7
Setup Rete
Allarmino

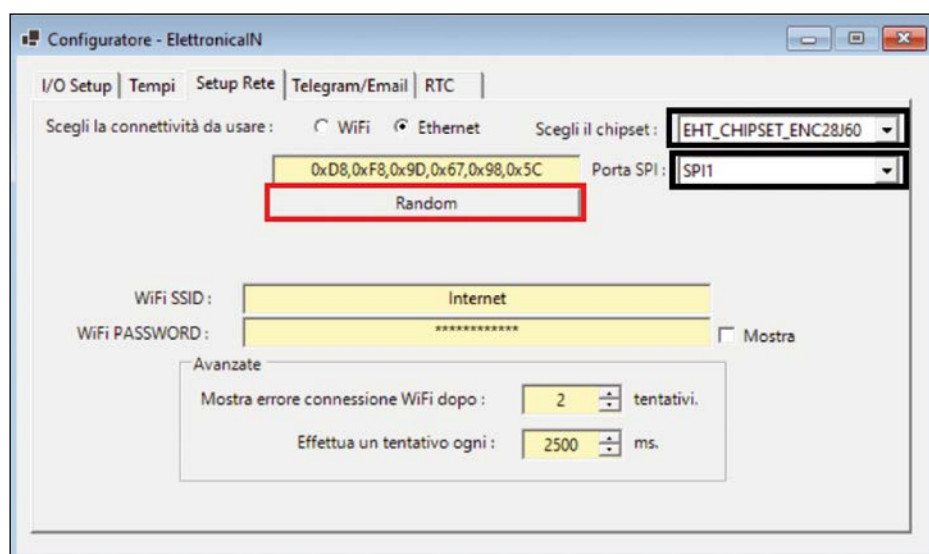


Fig. 8
Setup Telegram/
Email Allarmino.

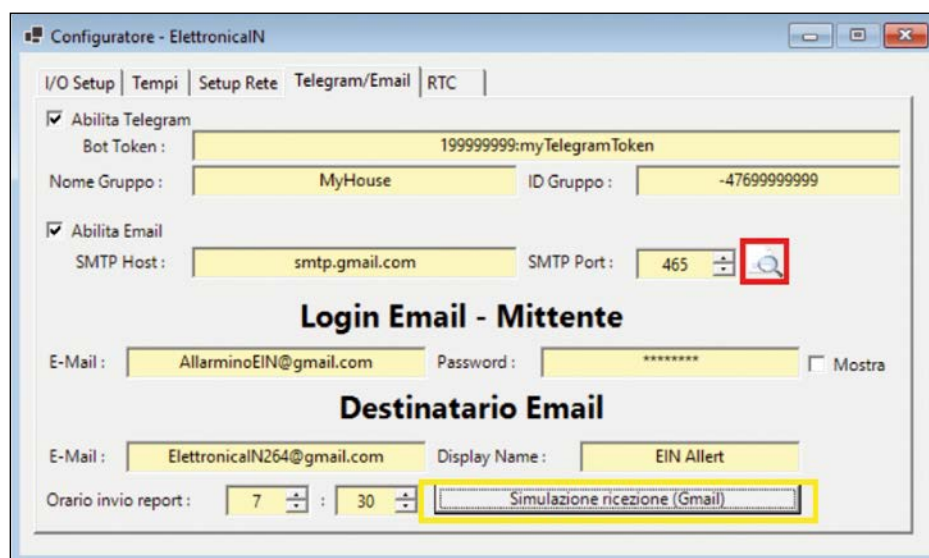
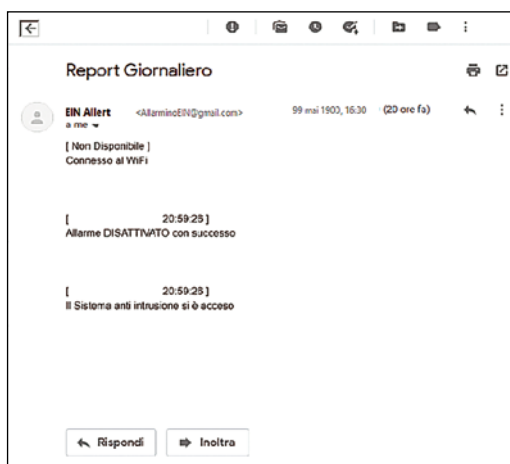


Fig. 9
Schermata
simulazione report
Email.



La scheda 'Telegram/Email' (**Fig. 8**) implementa un pulsante evidenziato in rosso che vi propone una lista di SMTP preconfigurati e che si possono scegliere facendo doppio clic su di essi, mentre in giallo è evidenziato un pulsante che vi permette di simulare la ricezione di un'e-mail: cliccandoci sopra vi apparirà la schermata di **Fig. 9**. Uscite dalla precedente schermata e andate su 'Visualizza->Centralina->Configuratore Avanzato->Admin Telegram' ora sarà possibile modificare la password di accesso alla sessione Admin ed impostare la durata massima della sessione. Mentre Cliccando su 'Certificati TLS' (**Fig. 10**) sarà possibile Generare (pulsante evidenziato in rosso) o Caricare (pulsante evidenziato in giallo) un certificato da

usare per la connessione SSL/TLS, spostandovi nella sezione 'Avanzata' evidenziata in nero apparirà il codice sorgente del file "ssl_cert.h"; potete modificarlo e/o rimpiazzarlo con quello creato dai tool ufficiali di BearSSL.

In "Visualizza->Centralina->Configuratore Avanzato->Notifiche Telegram" si aprirà la schermata di **Fig. 11**, la quale vi permetterà in maniera intuitiva di modificare tutti i messaggi riportati nei messaggi telegram visti nella prima parte dell'articolo. Una volta ultimata la configurazione del nostro progetto dobbiamo andare su "Strumenti->Programma centralina" e a quel punto cliccare su Inizia; fatto questo, attendiamo la fine della programmazione e a quel punto il nostro sistema Allarmino sarà completamente funzionante e pronto all'installazione e all'uso.

AGGIORNARE IL FIRMWARE DA TELEGRAM

Una volta installato il nuovo Firmware su Allarmino potremo finalmente accedere alla modalità amministratore tramite Telegram; questa funzionalità vi permetterà di aggiornare la configurazione e/o fare un upgrade/downgrade della centralina. Immaginiamo una situazione in cui vogliamo riprogrammare la funzione di un output o semplicemente cambiare un messaggio di Telegram: basterà ricompilare il progetto tramite il software togliendo il flag "Carica USB" dalla finestra "Programmazione" come mostrato in **Fig. 12** ed eseguire delle semplici istruzioni.

Inserite il jumper 'PRG JMP' sulla centralina e riavviate la tramite il pulsante 'EN' presente sull'ESP32, quindi andate sul gruppo Telegram e inviate il comando di help (di base è '/help'): si aprirà key-board custom come in **Fig. 13**, quindi cliccate il pulsante per avviare la sessione admin, che nell'esempio di **Fig. 13** è "/advanced_login", Attendete di ricevere un messaggio in privato dal BOT Telegram il quale vi guiderà nella procedura di aggiornamento Firmware. In alto nella **Fig. 14** è mostrato il messaggio standard che il BOT manda in privato al richiedente: ad esso rispondete con la password di amministratore (di base è 'Admin@123' come vedete in basso nella stessa **Fig. 14**), se

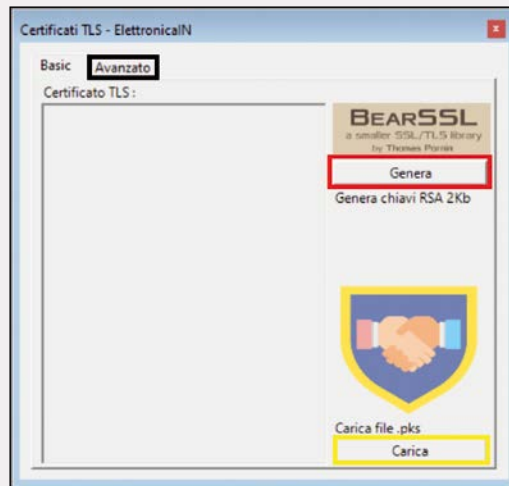


Fig. 10
Setup certificati TLS per Firmware 1.1V Allarmino.

corretta il BOT vi chiederà di inviare il file di aggiornamento firmware che troverete nella cartella 'build' del progetto. Una volta inviato il file su Telegram (se quest'ultimo fosse corretto) il BOT vi inviterà a confermare l'aggiornamento digitando il comando 'yes' come in **Fig. 15**; in essa, il box rosso evidenzia che il comando 'yes' deve essere una risposta al messaggio ricevuto dal BOT (in caso contrario viene

Fig. 11
Sezione avanzata per il setup certificati TLS-x.

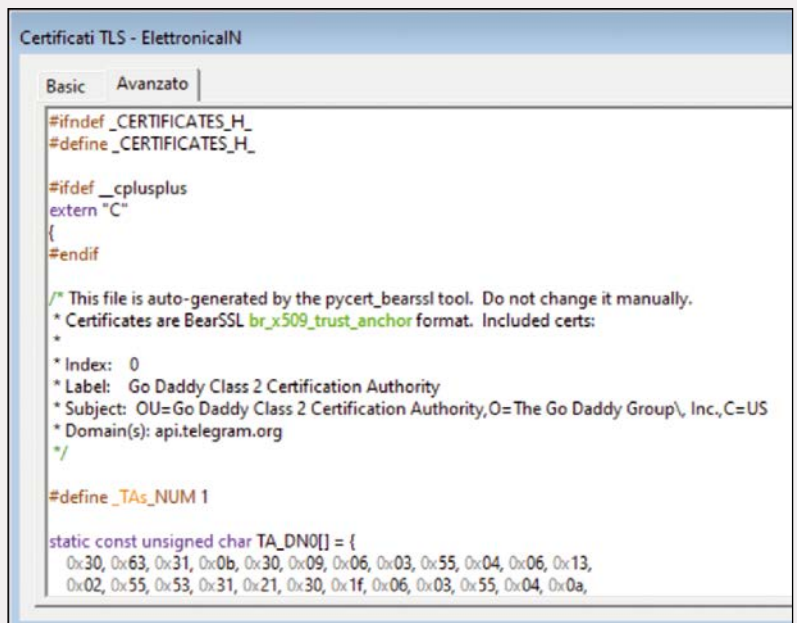


Fig. 12
Schermata del compilatore.



Fig. 13
Custom keyboard
generata su
Telegram.

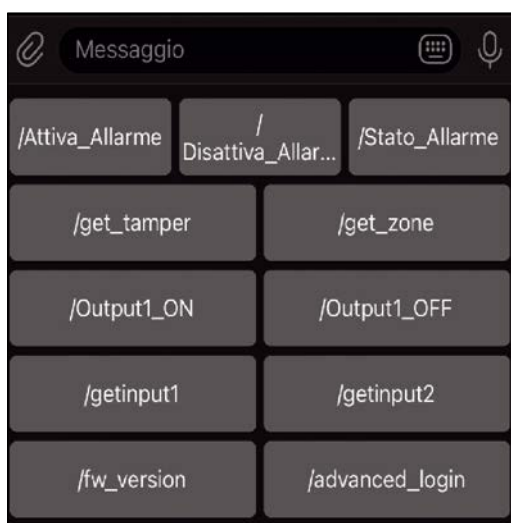


Fig. 14
Sessione di login
Admin e messaggio
di corretta login.

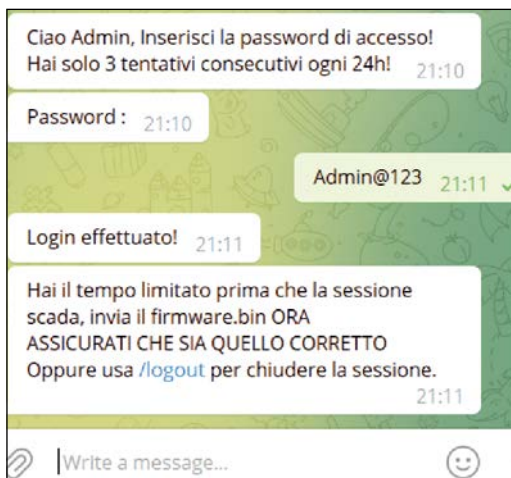
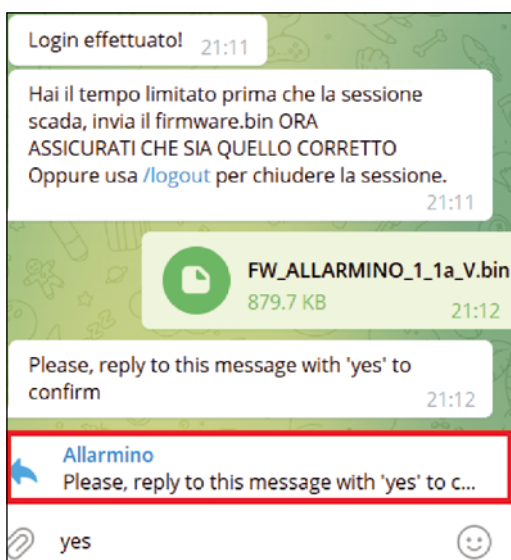


Fig. 15
Invio file .bin per
l'aggiornamento da
remoto.



ignorato). Una volta caricato e confermato l'aggiornamento, Allarmino provvederà a scaricarlo dai server di Telegram ed installarlo sulla Flash. Durante questa operazione l'utente non vedrà alcun tipo di messaggio o notifica, ma noterà che i LED AL e ST su Allarmino si accenderanno contemporaneamente per indicare che è in fase di flashing. Una volta finito, si riavvierà per aggiornare il nuovo firmware. Non appena Allarmino invierà il messaggio di Wake-up sarà possibile controllare la versione del firmware richiedendola tramite apposito comando. In ultimo facciamo notare che software di configurazione è stato testato utilizzando una versione customizzata (presente nella cartella d'installazione di AllarminoIDE) dell'ide di Arduino il quale integra anche una versione della board ESP32 portabile su un Personal Computer il software risulta funzionante su Windows 10 64bit e Windows 11 64bit, altre versioni di windows non sono state testate.

CONCLUSIONI

Nelle due puntate in cui è stato sviluppato questo articolo abbiamo voluto mostrarvi come realizzare un'espansione per il nostro sistema Allarmino IoT e darvi un primo assaggio delle reali potenzialità di questo sistema completo e dotato di più opzioni di connettività che contemplano interfacce wireless e cablate. Vi abbiamo anche proposto un nuovo firmware, aggiornabile via Telegram, nonché un software per gestire gli aggiornamenti. Ora vi invitiamo a sperimentare con questa espansione e con il software di gestione firmware dedicato. Speriamo di avervi dato un input per sviluppare ed implementare i vostri moduli usando anche il CAN-bus, che, lo ricordiamo, attualmente non è ancora gestito dal firmware di Allarmino; potete comunque cimentarvi nell'implementare da voi la gestione.



Cosa occorre?

I componenti principali utilizzati in questo progetto sono disponibili presso Futura Elettronica. Il modulo con ESP32 (cod. YB555) è in vendita a Euro 13,50, il controller ethernet ENC28J60 (cod. ENC28J60) è disponibile a Euro 9,80 mentre il W5500 (cod. MINIETHW5500) costa Euro 15,50. Il PCB può essere realizzato utilizzando il servizio PCBPRODUCTION. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

Fai il pieno di energia con le batterie ai polimeri di litio!

€ 27,90



COD. PLIB6000

3,7V-6000 mAh.
Connettore 2 mm

€ 6,90



COD. PLIB1000-2MM

3,7V-1000 mAh.
Connettore 2 mm

€ 7,90



COD. PLIB1100-2MM

3,7V-1100 mAh
connettore 2 mm

€ 21,00



COD.
25C2000MAH7V4

7,4V-2000 mAh.
Connettori JST-XH
Deans T

€ 34,00



COD.
25C2200MAH11V1

11,1V-2200 mAh.
Connettori JST-XH

€ 4,90



COD. PLIB500NEW

3,7V-500 mAh.
Connettore 2,54 mm

€ 6,90



COD. PLIB1000-1.25

3,7V-1000 mAh.
Connettore 1,25mm

€ 21,90



COD. PLIB5000MAH

3,7V-5000 mAh.
Connettore 2 mm

€ 21,90



COD. PLIB5000-2.54

3,7V-5000 mAh.
Connettore 2,54 mm

€ 12,50



COD. PLIB430

3,7V-500 mAh con
controllo temperatura.
Connettore JST 3 fili

€ 12,90



COD. PLIB2000

3,7V-2000 mAh.
Connettore 2,54 mm

€ 7,90



COD. PLIB1100

3,7V-1100 mAh.
Connettore 1,25mm

Prezzi IVA inclusa.

Non farti sfuggire nulla.

Riprendi ogni attimo con le mini telecamere.



cod. FR806

€ 79,00

**TELECAMERA WI-FI
IN LAMPADA A LED**

Telecamera Wi-Fi da 2 Megapixel nascosta in una lampada con 21 LED a luce bianca, Motion Detection, sensore crepuscolare, illuminatore con 8 LED IR con 15 m di portata.

Full HD
1080

SUPPORTA
MICRO SD CARD
MAX 128 GB



FORNITO DI
CONTENITORE
ERMETICO
PER RIPRESE
SUBACQUEE



cod. FR834

€ 33,00

**MINI TELECAMERA
ACTION SPORT
FULL HD CON AUDIO**

Può essere utilizzata per riprese sott'acqua. Integra un magnete che consente di fissarla facilmente a qualsiasi superficie metallica e dispone di staffa di fissaggio ruotabile di 360°.



SUPPORTA
MICRO SD CARD
MAX 128 GB

Full HD
1080



cod. FR825

€ 80,00

**TELECAMERA
A BOTTONE CON
TELECOMANDO**

Indossabile, di piccole dimensioni, facilmente occultabile con obiettivo a forma di bottone. Alimentazione tramite batteria interna Li-Ion ricaricabile.



SUPPORTA
MICRO SD
CARD MAX
32 GB

Full HD
1080



cod. CP756

€ 19,90

**MINI TELECAMERA
23 X 23 X 23 mm**

Registra video, scatta foto con una risoluzione fino a 12 Megapixel. Può essere utilizzata anche come Webcam. Alimentazione tramite batteria interna Li-Ion ricaricabile.



SUPPORTA
MICRO SD CARD
MAX 128 GB



Full HD
1080



cod. FR830

€ 38,00

**PENNA CON
TELECAMERA AD
ALTA RISOLUZIONE**

Elegante penna con telecamera nascosta in grado di effettuare riprese audio-video e di scattare fotografie in alta qualità. Alimentazione tramite batteria interna Li-Ion ricaricabile.

CRONOMETRO PER ATLETICA

dell'Ing. **MIRCO SEGATELLO**

Sistema di cronometraggio automatico adatto alla misura di tempi nelle attività di atletica leggera.

U

ltimamente è cresciuto, nel grande pubblico appassionato di sport (e non) l'interesse verso l'atletica, probabilmente per effetto degli importanti risultati conseguiti dai nostri atleti, che si sono distinti in competizioni sia europee che mondiali. Le prestazioni degli atleti, soprattutto nell'atletica leggera, sono oramai arrivate ai limiti fisici dell'essere umano, così che il "record del mondo" è sempre più difficile da ottenere e le competizioni di velocità si giocano sui centesimi di secondo e quasi sempre al photofinish.

Anche se meno noto, perché sta dietro le quinte dell'atletica, il cronometraggio si è evoluto nel tempo, diventando sempre più preciso e affidabile, tanto che nelle competizioni ufficiali delle federazioni nazionali e internazionali, sono ammessi

➔ **Fig. 1**
Starter con pistola
elettronica.



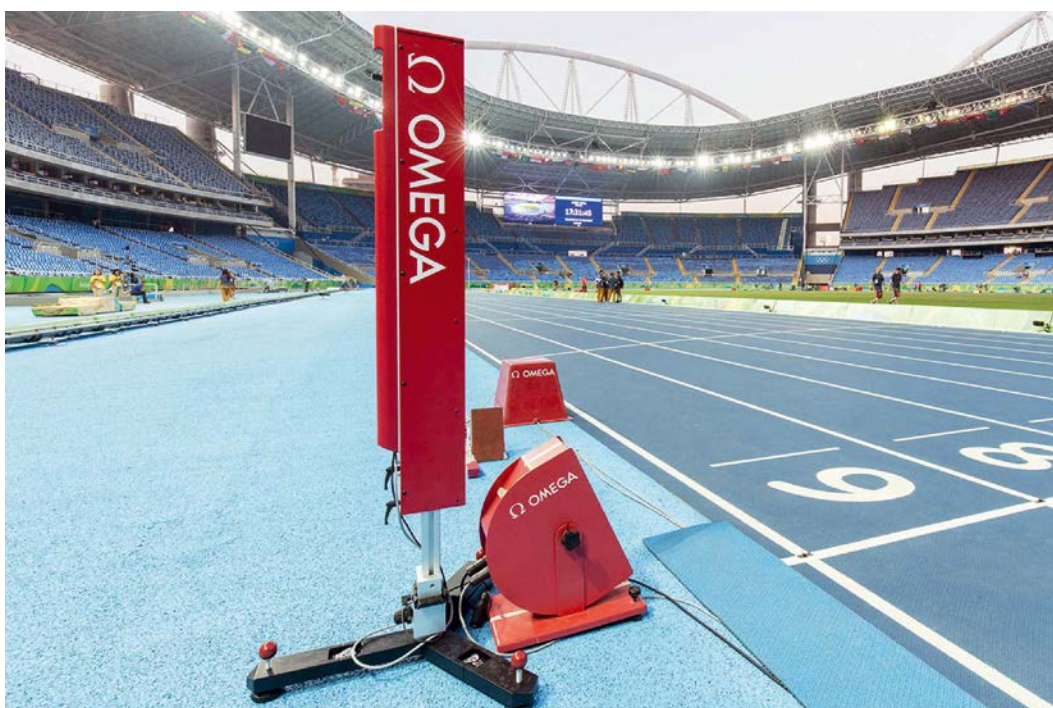
solo specifici sistemi omologati; questo perché è necessario considerare alcuni aspetti. Vediamone alcuni riferiti alle gare di corsa: lo start avveniva con uno scenico colpo di pistola a salve,

il cui suono però arrivava agli atleti in momenti diversi perché essi erano disposti a distanza diversa rispetto allo starter. Nella staffetta 4x100, in cui gli atleti non partono allineati ma sfalsati di diversi metri l'uno dall'altro, lo scarto di tempo è di circa 150 millisecondi. Oggigiorno ciascun blocco di partenza ha un proprio altoparlante che emette il suono di "via" triggerato da una pistola elettronica, che consente lo start esattamente nello stesso istante per tutti (**Fig. 1**).

Sulla linea di arrivo una barriera fotoelettrica rileva i tempi, per la precisione si tratta di tre fotocellule che forniscono l'impulso di stop solo se coperte contemporaneamente garantendo di non avere falsi contatti, infatti un atleta è considerato arrivato solo quando il torace attraversa l'arrivo e nel rilevamento del tempo non vanno considerati la testa, il collo, le braccia, le mani, le gambe e i piedi (**Fig. 2**).

Nel caso di gare di velocità, visti gli spazi ridotti, accade spesso che più atleti si sovrappongano sulla linea di arrivo, pertanto la classica fotocellula fallisce nell'individuare i tempi di arrivo, pertanto occorre ricorrere al photofinish. Il primo photofinish vero e proprio è stato usato a Los Angeles 1932 ed era già in grado di rilevare i tempi al centesimo di secondo. Alle olimpiadi Londra 1948, per la prima volta il dispositivo venne collegato alla pistola dello starter, misurando quindi il tempo dall'i-

➔ **Fig. 2**
Fotocellula
impiegata nelle
olimpiadi di Tokyo.





← **Fig. 3**
Immagine di
un photofinish.

stante esatto dello sparo. Il sistema consisteva in una videocamera che riprendeva, attraverso una piccola fessura direzionata sulla linea di arrivo, le scene che andavano ad impressionare la pellicola. In tempi moderni si utilizza una videocamera ad alto frame rate (almeno cinquemila) che registra solo una fila di pixel (in corrispondenza della linea di arrivo) che successivamente saranno usati per ricreare un'immagine, dalla quale ricavare i tempi di ciascun atleta con la precisione del millesimo di secondo. In **Fig. 3** potete vedere l'immagine di un photofinish che per sua natura (non essendo una foto) risulta alquanto deformata, pur mantenendo

le esatte proporzioni temporali e quindi le posizioni degli atleti.

Aggiungiamo un'ultima curiosità: alle Olimpiadi di Londra del 2012, a supporto del cronometraggio è stato introdotto un particolare orologio al quarzo in grado di rilevare il milionesimo di secondo!

Tralasciamo ora i sofisticatissimi sistemi usati nei campionati mondiali e olimpici e pensiamo a semplici atleti dilettantistici o anche comuni studenti nell'ora di scienze motorie; possiamo immaginare per loro un sistema di cronometraggio automatico di facilissimo utilizzo?

È proprio quello che andiamo a presentarvi in questo articolo, vale a dire un semplice ma efficace sistema di cronometraggio utile nelle competizioni amatoriali di velocità su pista di atletica.

L'HARDWARE

Come obiettivi, ci siamo posti la semplicità di utilizzo e l'immediatezza nell'installazione, immaginando che non sia disponibile uno staff adibito al cronometraggio, per cui il tutto il sistema si basa su un'unica unità di misurazione "all in one" di facile installazione. Essenzialmente abbiamo bisogno di una barriera fotoelettrica per il rilevamento del passaggio degli atleti, un cronometro e un'interfaccia utente con display, decisamente più immediato dell'utilizzo di un PC con software dedicato.

La barriera fotoelettrica utilizzata è siglata PEM10D ed è reperibile on-line sul sito Internet www.futurashop.it (di Futura Elettronica) con il

↓
Prototipo
assemblato.



Fig. 4
Il sensore
fotoelettrico
impiegato nel
progetto.



codice PEM10D (**Fig. 4**). Si tratta di una fotocellula a riflessione con catarifrangente che ha le seguenti caratteristiche:

- Portata massima: fino a 10 m
- Tempo di risposta: 5 – 100 ms
- Alimentazione: AC / DC 12 ~ 250 V
- Elemento trasmettente: LED IR
- LED giallo: quando è acceso indica l'interruzione del fascio della barriera
- Lunghezza cavo: 1,8 m
- Dimensioni: 20 x 39 x 63 mm
- Consumo: ~ 10 mA
- Uscita: Relè (corrente contatti 1 A / 250 Vac)
- Temperatura di lavoro: da -10° a +55°C
- Tenuta stagna: IP66

I parametri per noi fondamentali sono la portata di 10m, sufficienti per coprire la larghezza delle corsie, ed il tempo di reazione, contenuto in 5ms; le versioni con l'uscita a stato solido (NPN o PNP) possono scendere anche sotto il millisecondo. La configurazione a riflessione con catarifrangente assicura un'elevata portata e dai datasheets si evince anche l'elevata immunità alla luce solare, possiamo quindi usarla tranquillamente anche in esterni. Questo sensore dispone di due trimmer di regolazione: il trimmer più esterno serve ad impostare il tempo di risposta e nel nostro caso va impostato con il tempo minore ovvero 5msec, il secondo trimmer determina la distanza di funzionamento e va impostato in modo da garantire una portata di poco superiore alla larghezza della pista. L'uscita della fotocellula dispone di un contatto meccanico che comporta spesso rimbalzi e contatti elettrici multipli e instabili, di cui abbiamo tenuto conto a livello software, implementando la funzione anti-debouncing. Nel caso, tuttavia, si verificassero problemi a riguardo, consigliamo di inserire in parallelo ai contatti un semplice condensatore da 100nF, implementando così un efficace antirimbato hardware.

Come scheda a microcontrollore optiamo per la Maker UNO (codice MAKERUNO) che preferiamo rispetto una scheda Arduino ufficiale perché la presa dell'alimentazione è di tipo micro USB. Inoltre la board ha già integrato un cicalino, che utilizzeremo come segnale di conferma ad ogni attraversamento.

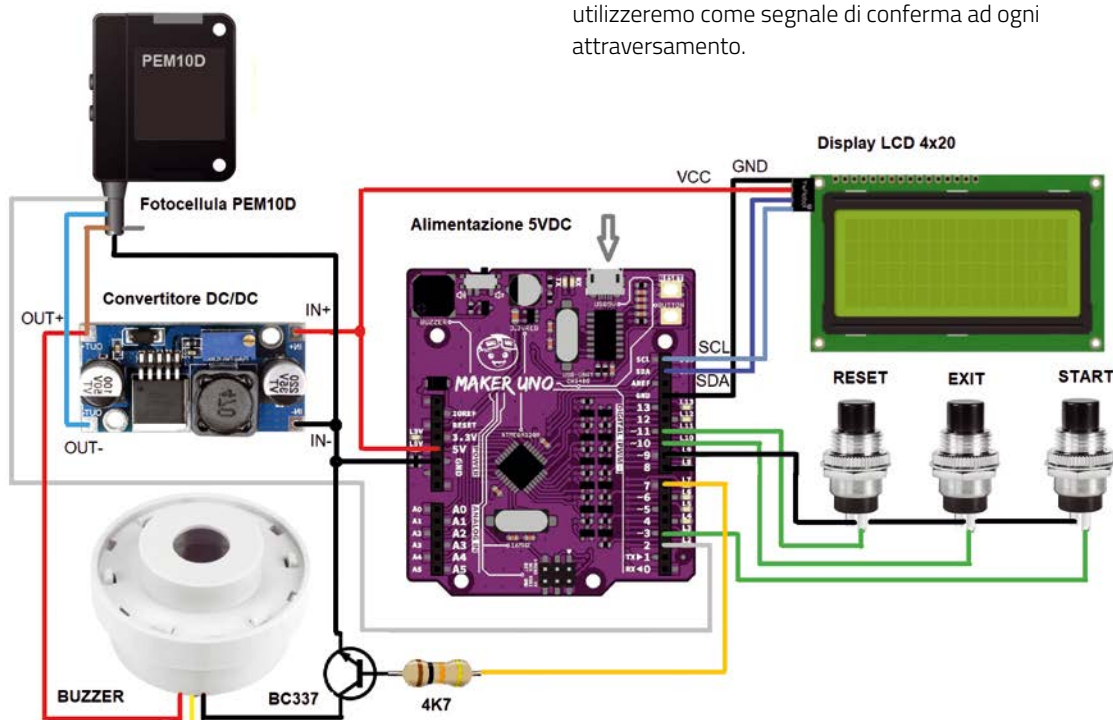


Fig. 5
Schema di cablaggio
del cronometro.

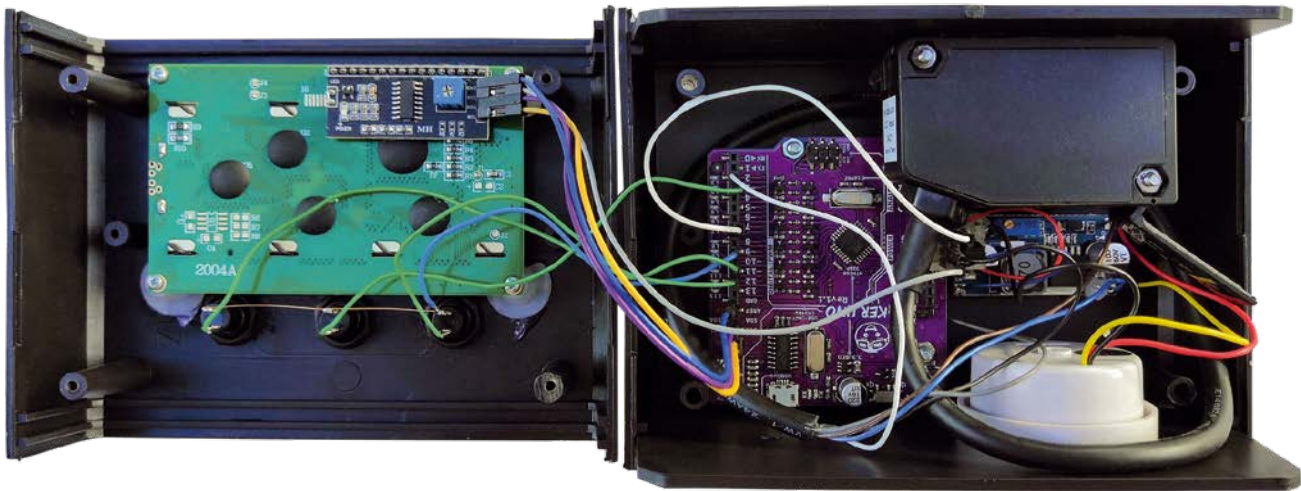
Ci servirà anche un display LCD da 20 caratteri su 4 righe con interfaccia I²C (codice LCD20X4AI2C) per la visualizzazione di tutte le informazioni ed un buzzer (codice SV1N) sufficientemente potente da generare il segnale di start al posto del colpo di pistola. Si tratta infatti del buzzer usato negli impianti di allarme in grado di emettere un suono netto e penetrante, facilmente udibile anche a distanza. Inizialmente avevamo pensato di utilizzare un microfono per rilevare il colpo di una pistola carica a salve, ma questo non rientrava negli obiettivi di semplicità e sicurezza che ci eravamo posti. Ipotizzando di alimentare tutto il circuito con un semplice power bank, possiamo contare su una tensione di 5volt, insufficiente per far funzionare la fotocellula ed il buzzer che necessitano di almeno 10volt, pertanto utilizziamo un convertitore DC/DC elevatore di tensione come quello di codice YXA347. In **Fig. 5** potete vedere lo schema di cablaggio che espone con il collegamento di tutti gli elementi, mentre in **Tabella 1** sono riassunti i collegamenti della scheda Maker UNO. Tutte le connessioni elettriche posso avvenire con semplici contatti strip, solo il bjt e la resistenza di base richiedono qualche attenzione; nel caso è possibile usare una piccola basetta millefori per la loro saldatura. A completare l'hardware il contenitore di codice KZ2 che raccoglie e protegge tutta la componentistica elettronica; la sistemazione interna di tutti i componenti è visibile in **Fig. 6**. Sia l'unità centrale che il catarifrangente dovranno essere posizionati su due treppiedi distinti, per-

MAKER UNO	COMPONENTE
GND	PEM10D filo nero, Display GND, Convertitore IN-
5V	Display Vcc, Convertitore IN+
2	PEM10D filo bianco
3	Pulsante Start
7	Base BJT con resistenza da 4.7Kohm
9	Comune pulsanti
10	Pulsante Reset
11	Pulsante Exit
SCL	Display SCL
SDA	Display SDA

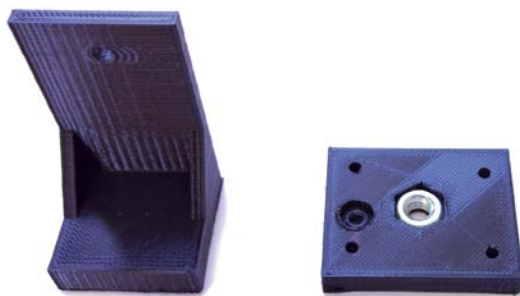
Tabella 1
Collegamenti
elettrici della
scheda
Maker UNO.

tanto abbiamo predisposto due semplici adattatori che permettono un facile fissaggio sui supporti standard dei treppiedi delle macchine fotografiche, provvisti di attacco a vite. I file per la stampa 3D dei due adattatori sono disponibili assieme ai file di questo numero, è però necessario vi dotiate di due dadi con misura anglosassone da 1/4" adatti agli attacchi dei cavalletti per fotografia (in **Fig. 7** trovate i due adattatori).

Fig. 6
I componenti
del cronometri
cablati nel
contenitore.



➔ **Fig. 7**
Adattatori per
treppiedi.



➔ **Fig. 8**
Schermata del
cronometro
manuale.

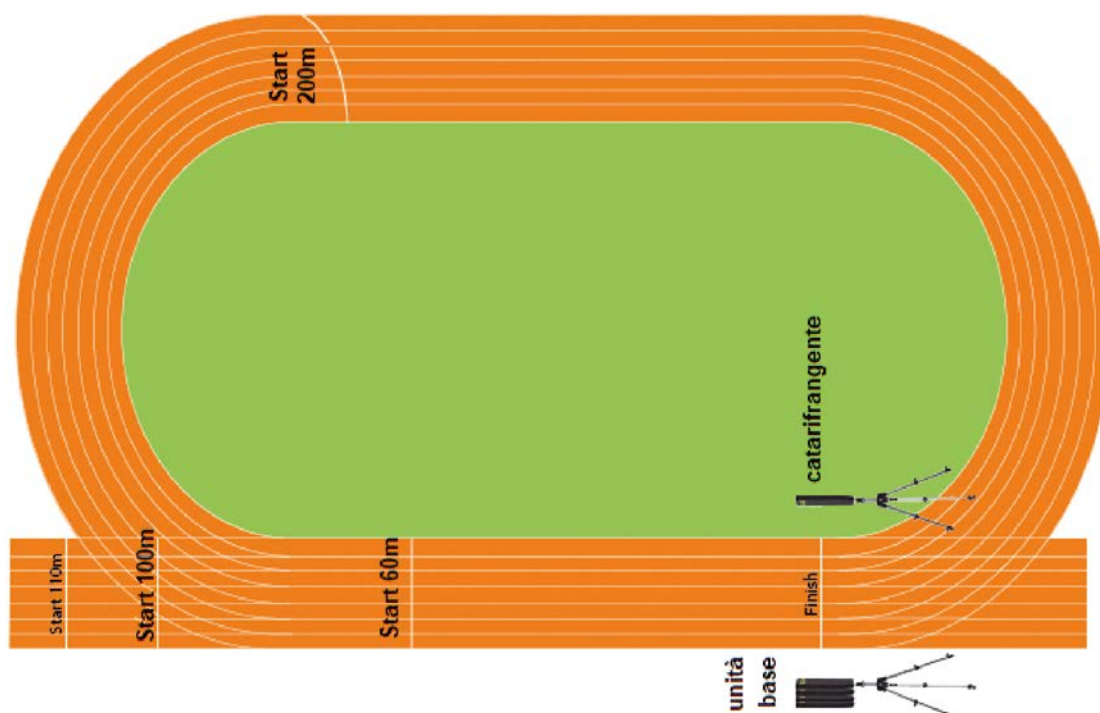


IL FIRMWARE

Terminata la descrizione della parte hardware, occupiamoci ora dello sviluppo del software, nel farlo, andiamo a descrivere le funzionalità del progetto. Per semplificare al massimo l'utilizzo abbiamo previsto un menu attraverso il quale poter selezionare la funzione desiderata, la quarta riga del display sarà utilizzata per avvisare il cronometrista della funzione svolta dai pulsanti. La prima funzione che abbiamo implementato è il classico cronometro

manuale, utile per rilevare i tempi in competizioni singole o di gruppo su lunghi percorsi. Un pulsante svolge la funzione di start ed un pulsante la funzione di reset; dopo la partenza il pulsante di start svolge la funzione di rilevare i tempi sul giro (lap time). È il caso di un corridore che deve fare un certo numero di giri di pista e si vuole monitorare la progressione dei tempi ad ogni giro, ma può essere utile nel caso vi siano più atleti a fare un giro di pista e si vuole rilevare il tempo di ciascuno di essi; in questo caso un secondo operatore dovrà trascrivere il tempo ed il nome dell'atleta (**Fig. 8**). Questa funzione non utilizza la barriera fotoelettrica e non è richiesta alcuna installazione particolare, le funzioni successive invece prevedono l'impiego della barriera fotoelettrica, pertanto è necessario predisporre il corretto posizionamento dell'unità centrale e del catarifrangente nell'area di gara. Supponiamo di voler cronometrare gli atleti in un giro di pista in modo totalmente automatico, la fotocellula svolgerà la funzione di start e di stop. Fissiamo l'unità centrale ed il catarifrangente sui rispettivi treppiedi e li posizioniamo sulla linea di partenza/arrivo, come indicato in **Fig. 9**. Dal menu principale selezioniamo allineamento e provvediamo ad allineare la fotocellula con il catarifrangente; conviene prima orientare il catarifrangente verso l'unità base e successivamente

➔ **Fig. 9**
Posizionamento
del sistema per la
misura del tempo
sul giro.



orientare quest'ultima, le indicazioni a display forniranno supporto dell'operazione come visibile in **Fig. 10**.

Completato l'allineamento, premiamo EXIT per tornare al menu principale e selezioniamo la funzione giropista. I concorrenti sono posizionati poco prima della linea di partenza ed attendono lo start da parte del cronometrista, mentre il sistema si predispone per l'avvio del conteggio al passaggio del primo atleta che interromperà la barriera ottica (**Fig. 11**), il sistema commuterà in automatico nella visualizzazione del cronometro.

La fotocellula rimarrà inibita per 5 sec per permettere al gruppo di atleti di passare dopodiché si predispone per rilevare il tempo degli atleti a giro concluso, sul display saranno visibili i tempi dei primi sei corridori arrivati (**Fig. 12**).

Le successive funzioni implementate riguardano la corsa dei 60, 100 e 200 metri piani, in questo caso gli atleti si posizionano nei rispettivi blocchi di partenza (riferitevi alla già citata **Fig. 9**) mentre il nostro sistema è posizionato sulla linea di arrivo. Lo start viene generato dal buzzer ad alta intensità sonora incorporato nel sistema, il suono raggiunge gli atleti praticamente nello stesso istante, ma con un certo ritardo da quando è stato generato, occorre infatti tenere in considerazione il tempo impiegato dal suono a raggiungere gli atleti: esso può essere valutato con la formula:

$$\text{Ritardo} = \text{distanza} / (331,45 + 0,62 \times \text{Temperatura})$$

Questo sistema è decisamente più semplice dell'impiego della pistola elettronica, però di contro soffre di alcuni limiti di precisione dovuti essenzialmente al tipo di suono generato e al tempo di ritardo nella percezione, che è influenzato dai parametri atmosferici come temperatura, vento e umidità. La validità del sistema rientra negli impieghi dilettantistici e la precisione è migliore del classico cronometro manuale, affetto dall'imprecisione umana, inoltre come nel caso del giropista, vengono mostrati a video i tempi di ciascun atleta in modo automatico.

I tempi sono mostrati nella modalità riportata nella **Tabella 2**, scalando in automatico la visualizzazione dei centesimi, decimi e secondi in base al tempo totale da visualizzare; il massimo tempo visualizzabile è 9 ore 59 minuti e 59 secondi, più che sufficiente per qualsiasi competizione.

CONCLUSIONI

Assemblando pochi componenti, avrete a disposi-



Fig. 10
Allineamento fotocellula.



Fig. 11
Schermata per il giropista.

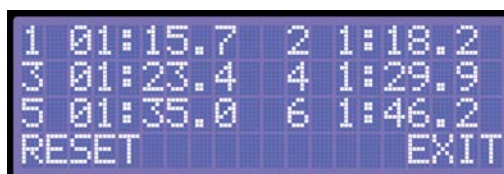


Fig. 12
Schermata con i tempi sul giro.

TEMPI	FORMATO
Da 0 a 10 minuti	minuti:secondi.centesimali
Da 10 minuti ad un'ora	minuti:secondi.decimali
Oltre un'ora	ore:minuti:secondi

Tabella 2
Formato di visualizzazione.

zione un buon cronometro automatico per atletica che vi permetterà di rendere più professionali le vostre competizioni su pista. La semplicità del montaggio (non è prevista la realizzazione di un PCB) e la facile installazione sul campo ne consentono l'utilizzo anche ai meno esperti di elettronica, che magari sono più "vicini" all'attività fisica all'aperto che alla tecnologia. Un po' di lavoro, il giusto posizionamento e vedrete subito chi corre più veloce! □



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. La board maker UNO (cod. MAKERUNO) è in vendita a Euro 13,50, la fotocellula (cod. PEM10D) a Euro 82,00, il buzzer (cod. SV1N) a Euro 4,90, il contenitore (cod. KZ2) a Euro 5,30, il convertitore step up (cod. YXA347) a Euro 6,00, il display I2C (cod. LCD20X4AI2C) a euro 15,00. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

Fai il pieno di energia!

BATTERIA RICARICABILE ULTRACELL AL PIOMBO SENZA MANUTENZIONE

Batterie al piombo senza manutenzione, ad alta densità d'energia e a bassissima autoscarica, prodotte dalla ULTRACELL. Per uso generico, la loro robustezza e affidabilità le consentono un ampio campo di applicazioni: solare, antifurto, backup, UPS, illuminazione d'emergenza ecc. La modalità di ricarica consigliata è quella a corrente costante.



BATTERIE RICARICABILI AL NI-MH VARTA "READY TO USE"



€ 12,00
Cod. HR03AAA800



€ 13,00
Cod. HR6AA2100



€ 13,80
Cod. HR14C3000

Prezzi IVA inclusa.

**FUTURA
ELETTRONICA®**

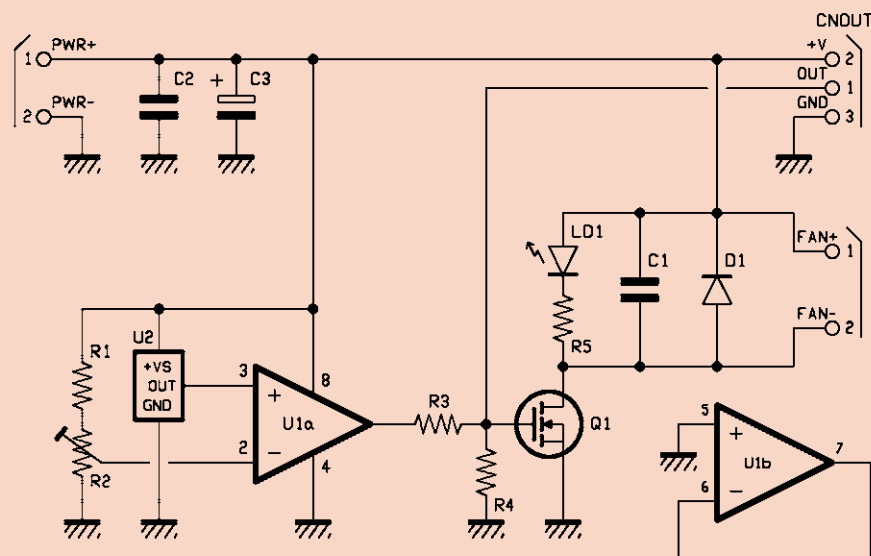
www.futurashop.it

Futura Group srl
Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775

Caratteristiche tecniche
di questi prodotti e acquisti on-line
su www.futurashop.it

Basato sull'integrato LM35, accende e spegne l'elettroventola di un sistema di raffreddamento per elettroniche in funzione della temperatura rilevata.





aria nuova li lambisca e li raffreddi, dato che il contatto cede all'aria una certa quantità di calore e che aumentando la quantità d'aria "fresca" è possibile rimuovere una maggior quantità di calore.

Siccome le elettroventole fanno un certo rumore, in funzione dell'aerodinamica loro e del contenitore e del percorso d'aria in cui vengono collocate, è utile che funzionino solo quando la temperatura rilevata indica che il solo dissipatore non basta più e magari che la loro velocità di rotazione, da cui dipende il flusso d'aria nel tempo, sia modulata in funzione delle esigenze di ventilazione e quindi della temperatura rilevata. Questa forma di controllo proporzionale, alternativo all'on/off (acceso quando serve ventilare e spento al di sotto di una certa temperatura) è preferibile non solo per il motivo su menzionato, ma anche perché consente di ottimizzare il consumo energetico, giacché permette di fornire all'elettroventola o gruppo di elettroventole la sola potenza necessaria in base al quadro termico del momento. Non a caso l'abbiamo proposta nell'articolo dedicato, nel fascicolo di novembre scorso, al controller PWM per ventole.

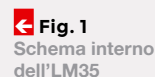
In questo articolo vogliamo invece presentare il progetto di un circuito controllore automatico e gestore di elettroventole per il raffreddamento funzionante in modo acceso/spento, utilizzabile in vari ambiti su elettroventole in corrente continua, a bassa tensione; un circuito reso ancor più versatile dalla presenza di un'uscita per il

controllo di dispositivi esterni, la quale fornisce una tensione continua da utilizzare ad esempio per leggere la condizione rispetto alla soglia di temperatura o per pilotare dispositivi controller di maggior potenza.

La differenza con il precedente è che qui la ventola, una volta accesa, gira alla velocità nominale, quindi non viene variata in base alla temperatura.

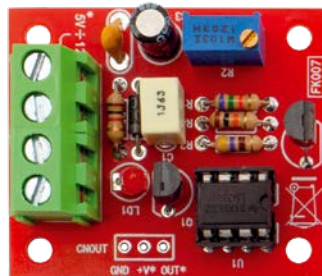
SCHEMA ELETTRICO

Vediamo dunque più nel dettaglio come è composto e in che modo funziona il circuito, appoggiandoci allo schema elettrico proposto in queste pagine; vedete come il tutto sia semplicissimo e utilizzi, a garanzia di precisione, un sensore di temperatura integrato, che poi non è altro che il popolare e sempreverde LM35. Questo componente a tre terminali è un sensore termico integrato basato sulla deriva termica in un diodo al suo interno, misurando i parametri della quale è possibile desumere a che temperatura sta funzionando. Contrariamente a molti dispositivi di pari funzione, come ad esempio il noto DS18B20, questo componente è completamente analogico e fornisce all'uscita una tensione continua il cui valore rispecchia con estrema linearità (malgrado la giunzione PN del diodo sensore e la circuizione interna, non siano proprio lineari) quello della temperatura alla quale il suo corpo si trova. Per l'esattezza, l'uscita può fornire una tensione che, in funzione della temperatura e in base alla



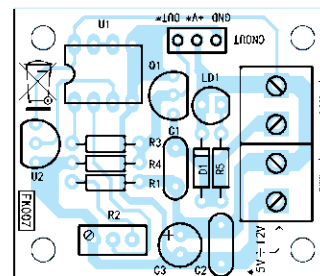
Elenco Componenti:

R1: 75 kohm
 R2: Trimmer multigiri 10 kohm
 R3: 470 ohm
 R4: 10 kohm
 R5: 1 kohm
 LD1: LED rosso 3 mm
 C1: 1 μ F 63 VL poliestere
 C2: 100 nF ceramico
 C3: 10 μ F 35 VL elettrolitico
 D1: 1N4007
 U1: LM358
 U2: LM35
 Q1: BS170



Varie:

- Strip maschio 3 vie
- Zoccolo 4+4



- Morsetto 2 vie passo 5 mm (2 pz.)
- Circuito stampato FK007 (36x42 mm)

3 dal sensore LM35 supera quella di riferimento applicata dal trimmer al piedino 2 e rimane invece a circa zero volt nella situazione contraria, ovvero se la tensione fornita dall'U2 è inferiore a quella fornita dall'R2; quando il piedino 1 è a livello alto, il MOSFET Q1 viene mandato in stato di ON perché il suo gate risulta polarizzato positivamente (è un canale N e si attiva così) attraverso il partitore resistivo R3-R4, mentre se lo stesso è a zero volt, Q1 rimane interdetto. Dunque, quando la temperatura sale e con essa la tensione fornita dall'uscita dell'LM35, superata quella presente tra il piedino 2 e massa, il comparatore U1a commuta la propria uscita attivando il MOSFET e quest'ultimo chiude a massa il circuito di alimentazione dell'elettroventola, ovvero idealmente cortocircuita il polo FAN- della morsettiera riservata al collegamento dell'elettroventola; viceversa, appena la temperatura scende abbastanza da portare il piedino 3 ad un potenziale inferiore a quello di riferimento impostato con il trimmer R2, l'uscita dell'U1b scende a circa zero volt lasciando interdire il MOSFET e spegnere la ventola.

Notate che per aumentare la precisione del controllo, ossia per selezionare finemente la temperatura alla quale accendere o spegnere l'elettroventola, abbiamo optato per un trimmer multigiri.

Dovendo operare su un carico fortemente induttivo quale è quello del motore dell'elettroventola, il nostro circuito è stato dotato del diodo D1 per spegnere i picchi di extratensione inversa che si verificano quando il MOSFET si interdice e sospende repentinamente la corrente negli avvolgimenti della ventola, allorché questi ultimi reagiscono sviluppando una tensione di valore

molto più elevato di quello nominale e di polarità opposta. Il condensatore C1 sopprime, invece, i disturbi impulsivi che la ventola produce. Ogni volta che il MOSFET Q1 va in stato di ON, la condizione viene evidenziata dal LED LD1, il quale, alimentato attraverso la resistenza di limitazione della corrente R5, viene alimentato e si illumina. Il transistor Q1 è un MOSFET a canale N enhancement mode realizzato in tecnologia DMOS, capace di commutare una corrente massima di 500 mA, perché tale è la sua corrente di drain; il circuito può quindi controllare ventole che assorbano non oltre 500 mA, pertanto da 6W se alimentate a 12V e 2,5W se funzionanti a 5V. Per eventuali sviluppi, abbiamo implementato nel circuito il connettore CNOOUT, il quale riporta sia la linea di alimentazione, sia l'uscita del comparatore U1a; tale uscita può essere utile ad esempio per trasmettere a un microcontrollore lo stato di sovratemperatura o per attivare un modulo di potenza che comandi altri utilizzatori, sconnetta il sistema sotto controllo perché magari l'elettroventola non riesce a raffreddarlo ecc.

Resta inteso che il livello dell'uscita OUT sul connettore CNOOUT dipende dalla tensione d'uscita dell'operazionale U1a e siccome questo è legato alla tensione di alimentazione del circuito, non è da intendersi come un vero e proprio livello logico standard, a meno di non alimentare il controller con una tensione standard tipo 5V: in questo caso OUT lavorerà a livello TTL.

L'intero circuito controller si alimenta a tensione continua del valore compreso fra 5 e 12V, applicata alla morsettiera +/- PWR; i condensatori C1 e C2, uno elettrolitico e l'altro non polarizzato, ripuliscono la linea di alimentazione da disturbi impulsivi,

interferenze della rete elettrica a 230Vca ed eventuale residuo di alternata presente all'uscita dell'alimentatore utilizzato per far funzionare il circuito.

Notate, infine, la particolare connessione del secondo operativo contenuto nell'LM358, il quale, non essendo utilizzato, è configurato come buffer (uscita collegata all'ingresso invertente) per renderlo stabile ed evitare che autoscilli o amplifici disturbi che potrebbero propagarsi all'interno del chip alterando il funzionamento di U1a; sempre in quest'ottica, l'input non-invertente (piedino 5) è stato collegato a massa, in modo da avere tensione di ingresso teoricamente nulla.

REALIZZAZIONE PRATICA

Bene, esaurita la teoria possiamo passare alla parte costruttiva di questo progetto: tutti i componenti necessari prendono posto su un circuito stampato a singola lamina e pertanto facilmente realizzabile anche da chi è alle prime armi, ad esempio per fotoincisione; allo scopo dovete scaricare dal nostro sito web www.elettronica.in.it e stampare su un foglio di acetato o carta da lucido la traccia lato rame. Potete anche ottenere il c.s. per tracciatura diretta con l'apposito pennarello. In ogni caso, una volta inciso e forato il PCB, potete montarvi i pochi componenti che servono, tutti a montaggio tradizionale: questa caratteristica rende adatto il montaggio anche ai principianti e a chi non ha l'attrezzatura per saldare gli SMD. I primi elementi da saldare sono quelli a più basso profilo, quindi resistenze e zoccolo DIP a 4+4 pin per l'integrato LM358, quindi proseguite con il trimmer multigiri, il diodo D1 e i condensatori, prestando attenzione alla polarità dell'elettrolitico C3; montate quindi il LED, ricordando che il catodo è l'elettrodo che si trova dal lato dello smusso sul contenitore, e l'integrato LM35, il cui lato piatto deve trovarsi rivolto al LED. Proseguire il montaggio con le due morsettiere a passo 5 mm per l'alimentazione (+ e - PWR) e l'uscita di controllo della ventola (+ e - FAN, rispettivamente contatto 1 e 2 della morsettiera) da collocare una accanto all'altra, incastrandole se il contenitore lo prevede. Completate il circuito inserendo nello zoccolo l'integrato U1, badando di orientarlo come mostrato nel piano di montaggio e verificando che nell'inserzione non si pieghi alcuno dei terminali sotto al corpo. Il connettore CNOUT è previsto a passo 2,54 mm e può essere costituito semplicemente da una fila di tre pin a passo 2,54 mm. Fatto ciò il controller è pronto per l'uso, perché non

ha alcun elemento da tarare: alimentate la morsettiera + e - PWR con una tensione continua idonea a quella della ventola (tipicamente 12Vcc) mediante un alimentatore in grado di erogare la corrente richiesta dalla ventola stessa, più una ventina di milliampere occorrenti al controller per funzionare. La ventola va connessa alla morsettiera FAN, rispettando la polarità indicata, altrimenti girerà al contrario o non girerà affatto.

Regolate il trimmer R2 in modo che la soglia di accensione/spegnimento corrisponda alla temperatura desiderata, che potrete determinare in due modi: appoggiando un termometro elettronico ad una parete o al dissipatore del sistema da ventilare, oppure convertendo, sulla base della caratteristica di uscita dell'LM35, la temperatura in tensione e poi registrando R2 con un tester commutato sulle misure di tensione continua i cui puntali avrete collegato tra massa e piedino 3 dell'integrato U1. Esempifichiamo quest'ultima modalità: se volete una soglia di commutazione a 60 °C, considerando che il gradiente di temperatura dell'uscita dell'integrato LM35, con la configurazione adottata nel nostro circuito, determina 600 mV (60 °C x 10 mV/°C) con il trimmer dovete impostare la tensione sul piedino 3 dell'LM358 a 0,6V.

CONCLUSIONI

Bene, con questo abbiamo terminato la descrizione del circuito, il quale, lo ricordiamo, va applicato a elettroventole funzionanti a 5 o a 12Vcc, di potenza fino a 6 watt. Sappiate che per particolari esigenze è possibile portare l'LM35 fuori dalla basetta, collegandolo con del cavetto schermato a due conduttori più maglia schermo (quest'ultima va a collegata a massa) di lunghezza massima di un paio di metri; non è invece consentito immergerlo nei liquidi, a meno di non impermeabilizzarlo con una guaina. ▣



Cosa occorre?

La board per controllo ventola (cod. FK007) è in vendita in scatola di montaggio a Euro 14,00. Il prezzo si intende IVA compresa.

Il materiale va richiesto a:

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



Consumi energetici sotto controllo!

Tieni sotto controllo i consumi della tua abitazione e risparmi sulla bolletta!



Misuratore di consumo di energia elettrica

€ 22,⁵⁰

cod. NETBSEM5

Permette di conoscere il consumo ed il costo dell'energia consumata dall'apparecchiatura collegata ad esso. Alimentazione 230 Vac; massimo carico applicabile 3,6 kW / 16 A.

Termostato Wi-Fi touch screen da incasso



€ 59,⁰⁰

cod. FR695

Ideale per gestire facilmente il clima domestico (riscaldamento o climatizzazione) anche lontano da casa grazie all'App gratuita per Android e iOS, risparmiando così sui consumi energetici.



Termostato Wireless



€ 62,⁰⁰

cod. TP384



Sistema per il controllo dell'impianto di riscaldamento o condizionamento, composto da un termostato wireless (868,35 MHz) con ampio display retroilluminato e da un'unità ricevente in grado di controllare un carico resistivo max. 16 A / 230 Vac o induttivo max. 4 A / 230 Vac.

DISPONIBILE ANCHE NELLA VERSIONE CON IL COLLEGAMENTO VIA CAVO

€ 32,⁰⁰

cod. TP385



Timer digitale programmabile su guida DIN

Timer digitale con display LCD per montaggio su guida DIN. Ideale per fornire o togliere l'alimentazione a vari generi di apparecchiature elettriche secondo il tempo desiderato e ad orari prestabiliti siano essi giornalieri o settimanali. Carico max. 3680 W, 16 A, alimentazione 230 Vac.

€ 17,⁹⁰

cod. TP238



Timer digitale settimanale da esterno (IP44)

Consente di accendere e spegnere apparecchiature elettriche nei giorni e nelle ore della settimana prestabilite. Dispone di 10 programmi di funzionamento per ciascun giorno della settimana e di 16 diverse combinazioni giornaliere. Potenza massima applicabile 3600 W / 16 A.

€ 18,⁹⁰

cod. WT51301



Prezzi IVA inclusa.

BLEB INVADERS

di PAOLO TEALDI E MARCO DARSENIO

Rievochiamo il fascino dei videogiochi Arcade con la tecnologia di prototipazione rapida BLEB abbinata a uno smartphone o a un tablet.

Da qualche tempo è arrivata sulla scena delle piattaforme di prototipazione didattiche e professionali Blebricks, una serie di moduli coordinati da un "cervello" con capacità di calcolo e interfaccia Bluetooth Low Energy (da cui il prefisso BLE...) che consente di implementare molte architetture hardware e di gestirle con dei firmware modulari e di rapidissima implementazione già pronti e configurabili all'occorrenza grazie ad un ambiente di sviluppo basato, ad esempio, su App Inventor, che si utilizza anche per la programmazione della popolare micro:bit. Nei mesi passati abbiamo dedicato una serie di articoli a Blebricks e ad applicazioni dimostrative in grado di evidenziare come con tale piattaforma si possa prototipare all'istante una vasta gamma di circuiti contenenti sensori

e attuatori, già pronti per l'IoT e interfacciabili a dispositivi mobile come gli smartphone. Adesso vogliamo tornare sull'argomento per farvi conoscere i Blebricks dal lato ludico, perché possono anche essere una maniera per imparare la programmazione realizzando un gioco: un videogioco, per l'esattezza, che ricalca lo stile Arcade in voga negli anni '80 del secolo scorso, ma che utilizza come elettronica un gruppo di mattoncini Blebricks e come schermo quello di un device mobile Android che può essere un tablet o uno smartphone.

BLEB INVADERS: GIOCHIAMO CON I BLEBRICKS

Il videogioco che vi proponiamo di realizzare lo abbiamo chiamato BLEB Invaders e può essere considerato la versione 4.0 di Space Invaders, ovvero il remake del 21° secolo di uno dei giochi Arcade più amati di tutti i tempi. Il gioco è pensato per accompagnare i ragazzi alla scoperta del mondo dei videogiochi, dell'inclusione sociale e della coordinazione del movimento del proprio corpo. Ma siamo andati oltre: grazie ai sensori Blebricks, abbiamo portato l'esperienza di gioco ad un livello superiore

con un'interfaccia utente molto speciale.

Dunque, cosa aspettate a realizzare questo videogame e a mettervi alla prova?! Se volete, qui di seguito vi spieghiamo come farlo.

Iniziamo con il descrivere da cosa è composto il nostro progetto: innanzitutto dal modulo immancabile nei progetti basati su Blebricks, ovvero il BLE-B: questo, caricandovi l'apposito programma, coordinerà l'attività degli altri moduli e, attraverso la propria connessione Bluetooth nativa, consentirà l'interfacciamento con il tablet o smartphone, che grazie a un'apposita app farà da schermo del videogioco, visualizzando sia il gioco in corso, sia informazioni per il giocatore, come meglio descritto nella **Fig. 1**. Il modulo BLE-B svolge inoltre un'altra funzione: tramite il suo pulsante, il giocatore può far sparare la propria navicella spaziale, quindi ad ogni pressione dello stesso, sullo schermo del device mobile vedremo partire i colpi diretti verso l'invasore spaziale, che costituisce il nemico da affrontare e combattere.

Al mattoncino BLE-B si collegano, tramite il pratico sistema a innesto, un modulo BLEB-ACC, la cui funzione è fare da dispositivo di input per rilevare il movimento che intendiamo far compiere (nel videogioco, è solo quello laterale) all'astronave; in pratica muovendo l'accelerometro con le mani (o applicando il tutto a una cintura e scuotendo il bacino...) indichiamo al BLE-B dove deve spostare la navicella spaziale sullo schermo.

Il Blebrick ACC è un sensore di accelerazione a 3 assi che permette di rilevare urti, attività e cambi di orientamento; di esso possiamo impostare l'intervallo di campionamento, configurare la sensibilità, la portata dei sensori e le condizioni di attività, shock e cambio orientamento. Il range di accelerazione rilevabile è compreso in $\pm 4g$ ed è configurabile da firmware, mentre la frequenza di refresh delle letture è 25 Hz, anche questa configurabile dall'esterno. L'intervallo di campionamento è 500 ms (ma può essere impostato su valori differenti) e l'assorbimento di corrente in fase di misura è di appena $< 14,5 \mu A$ al massimo delle prestazioni, quindi più che adatto ad un'applicazione che verrà alimentata a batteria, attraverso un terzo modulo: il BLEB-RPS. Il Blebrick RPS è un modulo di alimentazione che include un caricatore per batterie Li-Po/Li-Ion micro-USB (Type B) e una batteria da 3,7 volt - 80 mAh; nasce per alimentare altri Blebricks utilizzando una batteria Li-Po/Li-Ion oppure attingendo, come fonte di alimentazione, ad un alimentatore a 5 Vcc connesso alla porta micro-USB (Type B). Questo modulo eroga 3Vcc,

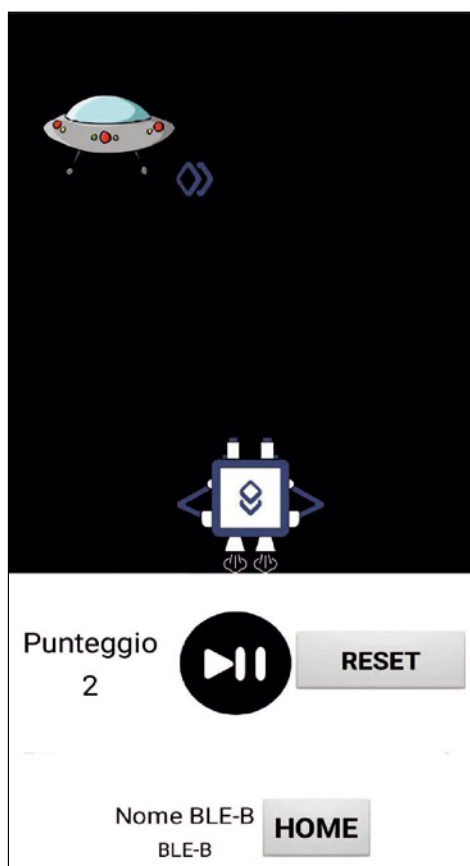


Fig. 1
Interfaccia grafica
del videogioco.

una corrente massima di scarica continua di 80 mA ed è in grado di caricare batterie al litio collegate con l'apposito connettore; nel nostro progetto consentirà l'alimentazione della parte indossabile (l'insieme dei Blebricks) in autonomia, quando la batteria sarà stata caricata.

Per facilitare l'utilizzo dell'hardware e applicarlo al corpo, è stato previsto l'utilizzo di un supporto specifico H3DR (H-3D-FDM/R) che è possibile ordinare o realizzare direttamente in stampa 3D utilizzando i files .stl scaricabili dal sito della rivista in modo da potervi montare e fissare i tre mattoncini richiesti dall'applicazione. Nell'eseguire l'assemblaggio, il mattoncino RPS va posto alla base e orientato in modo da far coincidere il connettore microUSB con l'apposita apertura posta sul davanti del supporto FIX; sopra di esso si applica l'accelerometro e in cima il BLE-B, in modo da renderne accessibile dall'esterno il pulsante e da far vedere la luce del suo LED. L'hardware qui descritto, per funzionare richiede la programmazione con uno specifico firmware che implementa essenzialmente tre operazioni: acquisisce ciclicamente i dati sul movimento trasmessi dal modulo BLEB-ACC, testa lo stato del pulsante del BLE-B per capire se è stato premuto, quindi prepara una stringa di dati contenenti le azioni conseguenti a tali informazioni e la trasmette via Bluetooth all'indirizzo con cui è identificato lo smartphone o tablet accoppiato durante il pairing Bluetooth.

Il modulo BLE-B, inoltre, verifica lo stato di carica della batteria e se ritiene che sia insufficiente disconnette l'elettronica e si pone nello stato di minimo consumo.

Una volta generati i dati e trasmessi, il resto lo deve fare il dispositivo mobile, nel quale un'apposita app provvederà a fare il resto; per questa applicazione è stata implementata una app per ambiente Android, della quale nel nostro sito www.elettronica.in.it troverete, insieme ai file di questo numero della rivista, il file .apk e il file .aia (quest'ultimo è un sorgente da aprire con MIT App Inventor). Il file .apk è un file di installazione da copiare nella cartella Android (o dove altro volete nello smartphone o tablet) ed eseguire tappando sulla relativa icona nella cartella in cui il file è stato salvato e scegliendo l'opzione *Installa* tra quelle proposte. L'applicazione prevede tre livelli di difficoltà del gioco: FACILE, MEDIO, DIFFICILE-SALTA; la differenza sta sostanzialmente nel modo di interagire con il gioco. Nella modalità FACILE, si utilizza semplicemente il pulsante del BLE-B per sparare contro l'astronave nemica (quella in alto nello schermo, nel livello MEDIO si sposta

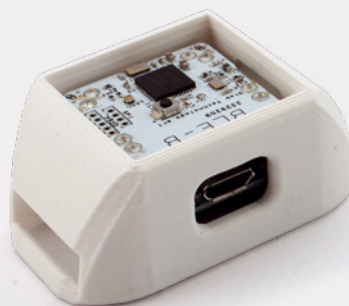


Fig. 2
Il dispositivo
FIX: ospita i
mattoncini e
consente il
fissaggio, ad
esempio, a
cintura.

l'astronave muovendo l'elettronica (o il corpo, se applichiamo il tutto alla cintura) e si spara non con il solito pulsante del BLE-B, ma con una singolare modalità di interazione, dove l'astronave si muove ancora da sinistra a destra muovendo l'elettronica, ma per sparare bisogna ...saltare! Sì, esattamente: in questo caso il modulo accelerometro rileverà l'accelerazione nell'asse Y (quello verticale) dovuta al salto e la interpreterà come comando di sparo. La modalità DIFFICILE è come MEDIO, però aumenta la velocità dell'astronave nemica.

LA APP PER ANDROID

La nostra app, sostanzialmente si connette via Bluetooth al dispositivo accoppiato, nello specifico il BLE-B, si predispone ad acquisire i dati da esso trasmessi e a visualizzarli. Per iniziare a giocare la prima volta occorre selezionare il dispositivo. A tal fine basta cliccare sull'iconcina di setup (rotella) che compare nella zona centrale in basso a destra della schermata iniziale e quindi premere seleziona Blebbo e cliccare sul dispositivo di interesse. Così facendo si salva il dispositivo in memoria e si imposta l'advertising a 20mS così da rendere fluida l'acquisizione dei dati di accelerazione. Questa funzione è utile specialmente nel caso ci siano più giocatori così che ognuno seleziona il suo con la propria App. Per distinguere meglio il proprio dispositivo possiamo anche assegnargli un nome breve di 6 caratteri.

L'app del Bleb Invaders come prima cosa, una volta avviata (l'icona è quella mostrata nella **Fig. 3**)



Fig. 3
Icona
dell'app.

Fig. 4
Schermata iniziale
dell'App.



Fig. 5
Il costrutto if/then
per gestire la fine
della partita.

propone la schermata di inizio del gioco nella quale viene offerta al giocatore la possibilità di scegliere uno dei livelli di gioco suaccennati (**Fig. 4**). Una volta scelta (tappando sulla relativa icona)

```

when post_esplosione_bomba_aliena .Timer
do
  set ESITO_GIOCO . Visible to true
  call PlayerEsplosione .Stop
  set Clock_esplosione . TimerEnabled to true
  set clock_bomba_aliena . TimerEnabled to true
  set AMBIENTE_DI_GIOCO . Visible to false
  set VerticalArrangement6 . Visible to true
  set Label2 . BackgroundColor to red
  set Label2 . Text to "HAI PERSO, RITENTA!"
  set navicella_nemica . Y to 50
  set punteggio_input . Text to 0
  set Clock_esplosione . TimerEnabled to false
  set global modalita_di_gioco to 0
  
```

la modalità di gioco preferita, inizia la partita. Ricordiamo che nel livello FACILE muovete orizzontalmente la navicella spostando l'hardware e sparate con il pulsante del modulo BLE-B, mentre nel MEDIO si spara saltando; nel livello DIFFICILE il controllo della nostra navicella avviene come nel MEDIO, però la velocità del gioco è elevata, quindi occorre più destrezza per colpire l'astronave nemica. L'icona con l'ingranaggio che vedete in basso a destra consente di impostare ad esempio l'orientamento dello schermo, in modo da adeguarlo alle vostre necessità e adattare la visualizzazione a quella dello schermo di un televisore o di un vecchio videogame da bar, rigorosamente a 4/3. La nostra applicazione è stata scritta in MIT App Inventor e del relativo codice andiamo ad analizzare il costrutto if / then che chiude una partita. Questo costrutto permette di far finire la partita in corso, far comparire la scritta «HAI PERSO, RITENTA» e altre istruzioni solo se l'astronave nemica raggiunge un certo valore di Y e collide con la nostra navicella, che qui è stata chiamata Blebbo (**Fig. 5**).

Passiamo ad esaminare i dati inviati all'APP dal modulo BLE-B, ossia dal "cervello" del nostro progetto, la quale riguarda la gestione del mattoncino accelerometro. Per rilevare il movimento abbiamo pensato di sfruttare l'accelerometro triassiale del Brick ACC. Disponendo il dispositivo con la presa microUSB rivolta verso il basso l'asse X corrisponde con il piano trasverso rispetto al corpo e muovendosi verso destra si rileverà un'accelerazione positiva, viceversa se ci spostiamo a sinistra. Utilizzeremo questi valori per far spostare di conseguenza l'astronave quando ci spostiamo con il corpo. L'asse Z corrisponde invece al piano frontale e rileverà le accelerazioni quando saltiamo. Utilizzeremo questo valore di accelerazione per sparare, stabilirne i parametri di inclinazione (**Fig. 6**). L'accelerometro è stato utile nella costruzione dell'algoritmo del salto perché vengono rilevati i valori di accelerazione sui tre assi (X,Y,Z) forniti da esso e i valori vengono messi in una formula costruita in modo da dare in output il modulo del vettore risultante dell'accelerazione. Infine è stato impostato un parametro (il cui valore di soglia è più o meno alto a seconda della modalità di gioco) il quale, messo in un costrutto IF, fornisce in output lo sparo se e solo se la condizione è verificata (accelerazione risultante maggiore del valore inserito). La porzione di codice che riguarda la gestione dell'accelerometro è proposta nella **Fig. 7**.

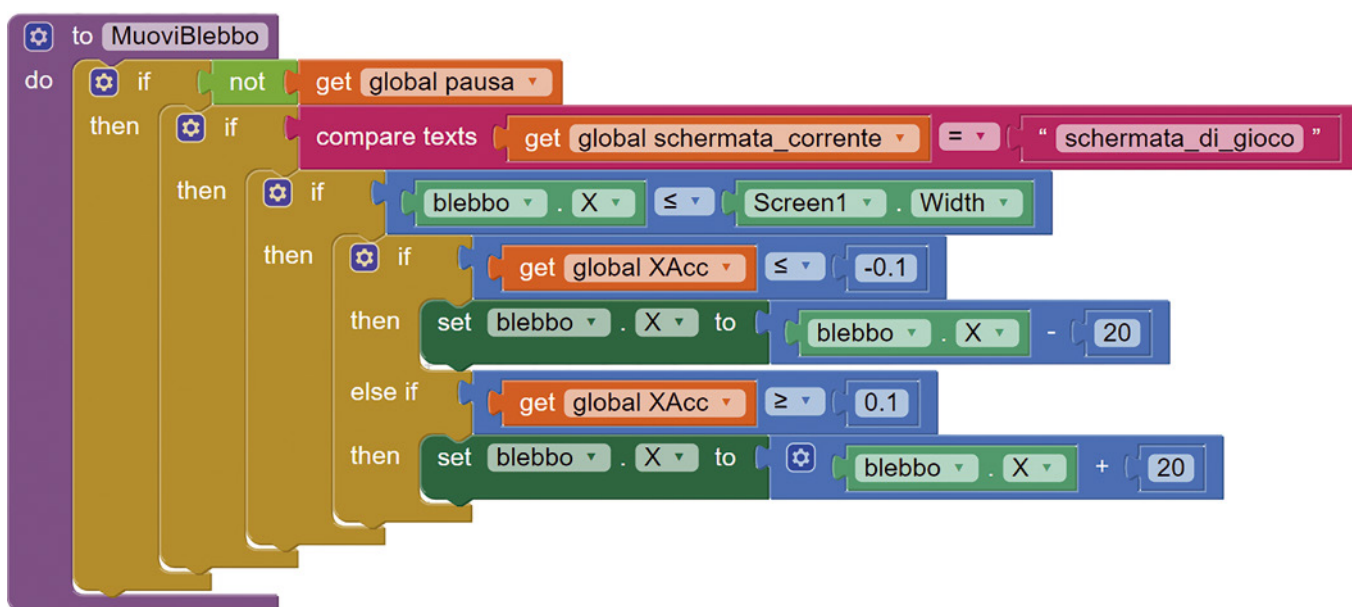


Fig. 6
Come vengono stabiliti i parametri di inclinazione.

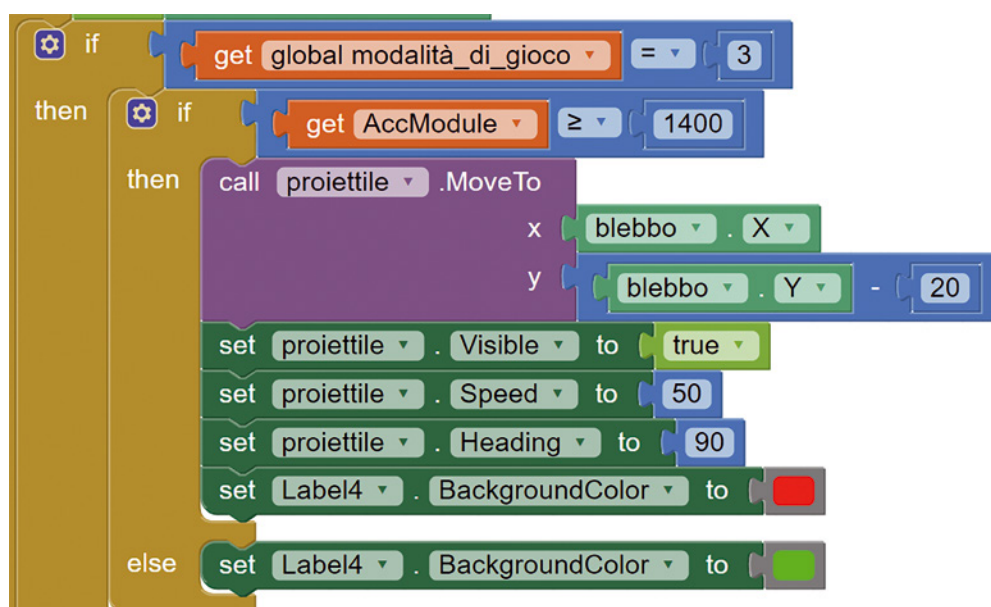


Fig. 7
L'algoritmo per il calcolo dell'accelerazione.

CONCLUSIONI

Termina qui la descrizione di un videogame che è insieme un'occasione per imparare il coding divertendosi e per riscoprire il fascino di quei videogiochi Arcade (da bar) che i più giovani hanno conosciuto solo nei ricordi dei loro genitori, ma che hanno affascinato una generazione e funzionato da propulsore per arrivare ai giochi da PC e a quelli in Rete. Ancora una volta avete visto come con la piattaforma Blebricks e un affermato ambiente di programmazione quale MIT App Inventor sia possibile sviluppare applicazioni software e interfacciarle al mondo dei dispositivi mobile. □



Cosa occorre?

I moduli Blebricks utilizzati in questo progetto sono disponibili presso Futura Elettronica. La main board (cod. BLE-B) è in vendita a Euro 29,00, il modulo di alimentazione (cod. BLEBRPS) a Euro 38,00, il modulo accelerometro (cod. BLEBACC) a Euro 19,00. Tutti i prodotti Blebricks sono disponibili alla pagina <https://bit.ly/blebricks> I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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



CATTURA L'ENERGIA DEL FUTURO

L'energia pulita che ti fa risparmiare!

**Pannello solare monocristallino
semirigido 12V**



**Pannello solare monocristallino
rigido 12V**



PANNELLI SOLARI MONOCRISTALLINI

Sono composti da singoli e sottilissimi strati di cristallo di silicio che opportunamente trattato 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.

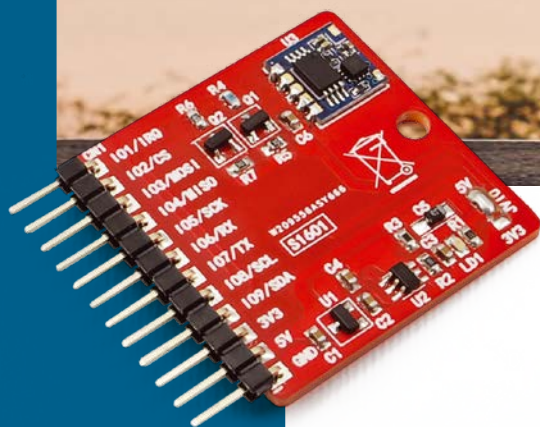
E per dare la giusta corrente alle tue batterie... REGOLATORI DI CARICA PER PANNELLI FOTOVOLTAICI A 12 E 24V



FORNISCONO ALLE BATTERIE LA QUANTITÀ DI CORRENTE NECESSARIA, PROTEGGENDOLE DA UN ECCESSO DI CARICA O SCARICA.

di ALESSANDRO SOTTOCORNOLA

CONTAPASSI ELETTRONICO

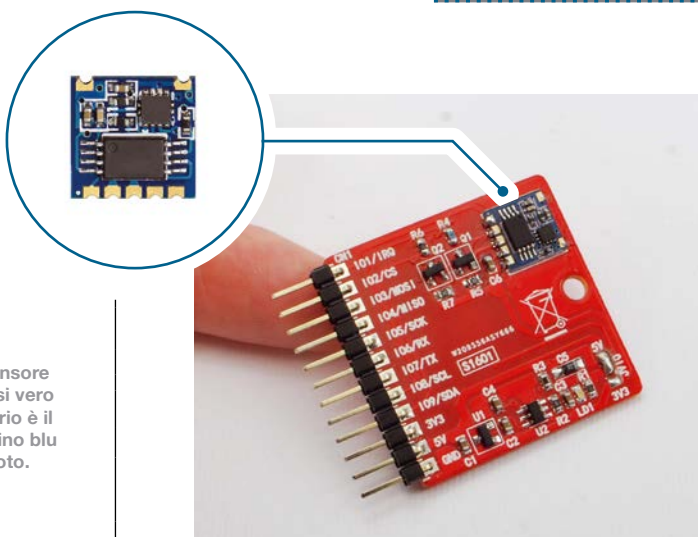


Utilizziamo un modulo accelerometro MEMS montato su breakout board, interfacciato tramite I²C-Bus ad una board Arduino per rilevare e contare i passi durante la giornata, un'attività fisica o il lavoro.

Q

uello del wellness o benessere che dir si voglia, è un concetto ormai assodato nella vita quotidiana, non fosse altro che per la martellante pubblicità e la crescente offerta di prodotti e servizi che lo sostentano: non solo abitudini di vita, diete e palestra, ma anche "oggetti"

familiari e di uso quotidiano come il telefono cellulare o gli orologi per sportivi, con misuratore di pulsazioni, di passi e capacità di rilevamento del tipo di attività. Questo rilevamento viene effettuato tipicamente da sensori di movimento e, più nello specifico, dagli accelerometri; ecco perché negli smartphone, dove l'accelerometro già esiste per rilevare l'orientamento dell'apparecchio e adeguare la visualizzazione sul display, troviamo app e funzionalità contapassi, rivolte sia allo sportivo che a chi fa la vita di tutti i giorni. Considerato l'interesse che sempre più persone dimostrano nei confronti del proprio benessere e della salute e il desiderio, sostenuto dalle raccomandazioni dei medici, di camminare di più e fare scale, invece di muoversi sempre in auto e prendere l'ascensore, abbiamo pensato di suggerirvi un progetto in linea con



Il sensore di passi vero e proprio è il modulino blu nella foto.

il tema, ovvero la realizzazione di un pedometro elettronico che vi consentirà di contare i passi quando camminate e capire se avete fatto la vostra "razione quotidiana" per bruciare calorie in eccesso e mantenervi in forma.

IL PROGETTO

Il nostro contapassi si basa su un piccolissimo modulo (misura circa 10x10 mm) montato su una breakout board appositamente relizzata; il modulo è siglato STM201M ed è un pedometro 3D con chipset IC funzionale che include un sensore di accelerazione (G-sensor) e una MCU per interfacciarlo e controllarlo. È prodotto dalla cinese NiceRF (www.nicerf.cn) e adotta l'interfaccia I²C standard; con il sensore 3D MEMS (sensore G) e l'alta precisione dell'algoritmo del pedometro 3D, può fornire un contapassi preciso in qualsiasi direzione. Questo chipset ha ottime caratteristiche, dimensioni ridotte, basso consumo energetico (assorbimento di 5 microampere in modalità sleep e di

60 microampere nel funzionamento ordinario...) e così via.

L'interfaccia digitale I²C standard (nella quale il nostro modulo è uno Slave) garantisce che possa essere facilmente integrato in vari tipi di sistemi funzionali contapassi e per l'idea che abbiamo di abbinare la breakout board ad Arduino, è la soluzione ottimale.

La **Fig. 1** vi mostra il modulo da entrambe le facce: come vedete, è piccolissimo, tanto da essere integrato ovunque.

Nella **Tabella 1** trovate invece la piedinatura dell'STM201M, con relativa indicazione della destinazione dei suoi 7 pad di saldatura.

Il grande vantaggio del modulo sta nel fatto che a bordo dispone di un microcontrollore (la suddetta MCU, appunto...) programmata in modo da interfacciarsi direttamente al sensore MEMS e leggere, interrogandolo quando serve, i dati sui passi rilevati; a riguardo precisiamo che la scheda può fornire le informazioni sui passi in due modi:

- attraverso un pin di interrupt connesso direttamente all'integrato accelerometro MEMS e riportato all'esterno su un pad (pin STEP_OUT); in questo caso abbiamo a disposizione una transizione di livello logico da 0 ad 1 della durata di 50 ms per ogni passo;
- interrogando al bisogno, tramite I²C-Bus, il microcontrollore, che provvede all'acquisizione del segnale sui passi fornito dall'accelerometro MEMS e risponde con una stringa di dati contenente le informazioni richieste.

In quest'ultima modalità, il modulo viene risvegliato dallo sleep e consuma di più, ma comunque si mantiene su un valore di 60 microampere. Invece con la modalità STEP_OUT il chipset può emettere l'impulso per ogni azione di passaggio.

L'algoritmo implementato nel microcontrollore per la funzione pedometro, e quindi per decifrare il segnale fornito dal sensore MEMS, è regolabile in base alle esigenze del cliente, nel senso che il produttore può fornire personalizzazioni in base a dove il sensore MEMS viene collocato: contapassi per scarpe, contapassi da indossare al polso, contapassi da indossare in vita e contapassi da tenere in tasca. Nella nostra versione è implementato il firmware generico, non personalizzato, quindi potete utilizzare la breakout board applicata a una cintura o al braccio.

Torniamo ora un momento all'interrogazione, ipotizzando di lavorare, come poi faremo, con una scheda a microcontrollore esterna che dialogherà

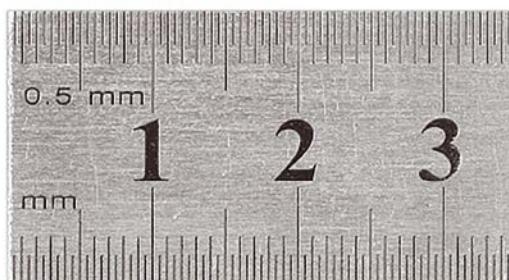
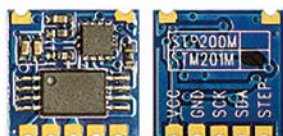


Fig. 1
Il modulo visto da entrambe le facce.

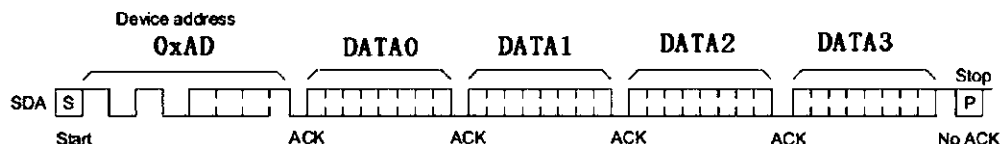


Fig. 2
Diagramma della comunicazione I²C-Bus del modulo.

con la breakout board attraverso il bus I²C; il protocollo di comunicazione del modulo STM201M è quello descritto nella **Fig. 2**.

Nella relativa stringa possiamo distinguere questi byte esadecimali:

- DATA0 = step lower byte
- DATA1 = step high byte;
- DATA2 = DATA0 + DATA1
- DATA3 = DATA2

Quando si verifica l'overflow del conteggio, ossia i passi superano il valore esadecimale FFFF (65.535), il conteggio riparte da 0. I passi vengono azzerati quando il blocco DATA3 sarà stato letto dal dispositivo Master.

Il chipset commuterà dalla modalità sleep a quella di normale operatività quando verrà rilevato un passo ed entrerà in modalità di sospensione se non viene rilevato alcuno step per 20 secondi. In modalità di sospensione il sensore G non funziona, l'MCU entrerà in modalità di sleep e l'intero modulo minimizzerà col consumo energetico. Il modulo può essere riattivato attraverso l'interfaccia I²C, ma tenendo conto che in questo caso esistono alcune limitazioni: per il primo comando di "avvio" del protocollo I²C, l'SCK dovrà permanere a livello alto per almeno 500us dopo il ritorno a zero della linea SDA, altrimenti non potrà essere riattivato il microcontrollore on-board.

La massima frequenza di clock (ovvero della linea SCK dell'STM201M) supportata sul bus I²C, in questo modulo, è 100 kHz, quindi ridotta rispetto al clock usato più frequentemente nelle soluzioni che comunicano su I²C-Bus (dove tipicamente si va a 400 kHz).

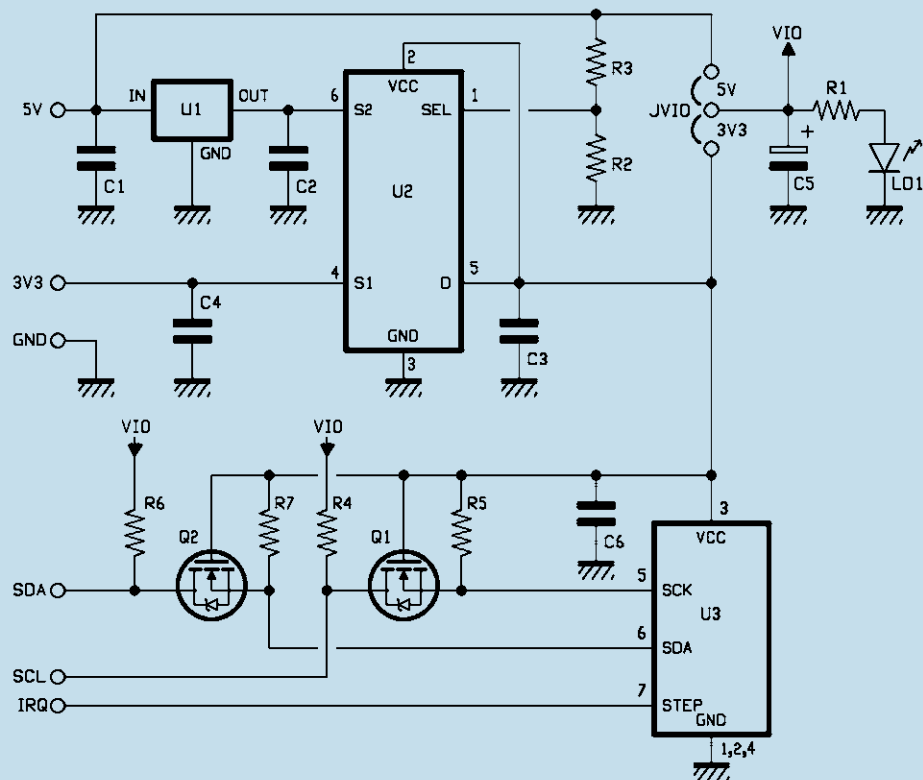
SCHEMA ELETTRICO

Ora che conosciamo il modulo vediamo in che modo è stato utilizzato nella nostra breakout board: lo facciamo riferendoci allo schema elettrico che lo descrive. Come vedete, il piccolo modulo è siglato U3 ed è alimentato tramite una sezione di alimentazione che riceve sia 5V rispetto a massa (contatti 5V e GND) che 3,3V (contatto 3V3);

questa doppia tensione serve per adattare il bus I²C della breakout board a funzionare a 3,3V o a 5V. L'alimentazione a 5 volt viene inviata all'ingresso di un regolatore LDO (U1) di tipo MCP1700T-3302E/TT la cui funzione è ricavare 3,3V stabilizzati da inviare al selettore di alimentazione; quest'ultimo è sostanzialmente uno switch a MOSFET che fa passare al proprio piedino D una delle tensioni applicate ad S1 (linea 3V3) ed S2 (uscita dell'LDO) in base alla tensione applicata al pin SEL. Per polarizzare SEL utilizziamo un partitore resistivo alimentato dalla linea di ingresso 5V; tale partitore serve ad abbassare la tensione, perché il pin in questione ha una logica 0÷1,8V. Quando è posto a 0 logico, la tensione applicata ad S1 (linea 3V3) viene applicata a D, mentre se è a 1,8V il piedino D viene alimentato dalla linea S2 (uscita del regolatore LDO); ne deriva che quando è presente la tensione di 5 volt, l'integrato switch alimenta il modulo U3 con la tensione fornita dal regolatore a basso drop-out, mentre se applichiamo tensione al contatto 3V3 della breakout board, il modulo pedomero riceverà tensione da essa. U2 è un TMUX1219DBVR prodotto dalla Texas Instruments e può essere alimentato da 1,08 V a 5,5 V e, sebbene lo abbiamo utilizzato come commutatore della linea di alimentazione del modulo STM201M, è uno switch bidirezionale, quindi può commutare dai pin S1/S2 verso il D, o viceversa;

Tabella 1
Connessioni del modulo STP201M.

PAD N.	NOME	DESCRIZIONE
1	GND	Connettere a massa
2	GND	Connettere a massa
3	VCC	Positivo di alimentazione 2.3-3.6V
4	GND	Connettere a massa
5	SCK	Clock del bus I²C
6	SDA	Linea dati del bus I²C
7	STEP	Impulsi di conteggio dei passi



i piedini 5 corrispondono ai source dei MOSFET componenti lo switch, mentre il D è il drain in comune.

Sulla scheda trovate un doppio jumper che permette di impostare anche la tensione di lavoro del bus I²C (VIO) da impostare in base alla tensione cui alimenterete la breakout board; questo perché per rendere più versatile la breakout board abbiamo inserito nel circuito un adattatore a MOSFET per l'interfaccia I²C-Bus utile a traslare i livelli logici supportati dal sensore per adattarli ai TTL standard di schede a logica tradizionale e ad Arduino. Ricordiamo comunque che l'U3 funziona sempre a 3,3V e quindi il jumper JVIO permette di scegliere una tensione lato connessioni esterne della scheda fra 3,3V (quindi la stessa del bus I²C del modulo) e 5V (tipica delle schede Arduino UNO).

Per spiegare come funziona il traslatore partiamo dal clock, che sul lato dell'U3 fa capo al piedino 5 (SCK), mentre verso il pin-strip di connessione è collegato a SCL (entrambe le linee sono provviste di un resistore di pull-up, ossia R4 sul lato SCL ed R5 lato sensore): quando SCL del connettore è a livello logico alto, il diodo di protezione interno al MOSFET Q1 (posto tra drain e source) è interdetto perché il source si trova negativo (alimentazione

VIO a 5 volt e bus lato strip, a 5V) o equipotenziale (VIO a 3,3V) rispetto al drain (che è mantenuto a 3,3 volt dalla R4) e pur se l'impulso non giunge al contatto SCK dell'U3, quest'ultimo è ad 1 logico grazie al pull-up operato da R5. Invece quando il contatto SCL del pin-strip è a zero logico (0 V), il MOSFET è ancora interdetto (perché il suo gate è polarizzato con 3,3V) ma conduce il suo diodo di protezione, dato che il source è a 3,3V mentre il catodo è a zero, quindi il livello basso su SCL trascina a zero logico (0,6 per l'esattezza) il piedino 5 dell'U3. Il canale SCL è unidirezionale, quindi la situazione opposta non si presenta, cioè U3 riceve il clock dall'unità Master, che nella nostra applicazione sarà Arduino.

Passiamo adesso all'SDA, analizzando il caso in cui è l'unità Master a inviare i dati e ipotizziamo che il dispositivo con cui la breakout board è interfacciata abbia il bus operante a 5V: quando tale contatto è a zero logico, il diodo di protezione del Q2 conduce e trascina a 0,6 V il source e con esso il piedino 6 dell'U3; invece quando è a 5 volt, MOSFET e diodo sono interdetti, quindi SDA dell'U3 viene portata a 3,3V (livello alto) dal resistore di pull-up Rt. Se invece è il sensore U3 a trasmettere, quando pone il proprio piedino 6 a zero logico il MOSFET Q2 con-

duce tra drain e source, perché il suo elettrodo di source è a zero volt mentre il gate è a 3,3 V, quindi la sua corrente di drain trascina a circa 0 V l'SDA del connettore, a causa della caduta sul resistore di pull-up R6; invece quando il pin 6 dell'U3 è a livello alto (3,3 V) il source e il gate sono equipotenziali e il MOSFET rimane interdetto, lasciando che sia il resistore di pull-up R3 a mantenere a 1 logico (ma a 5 volt) la linea SDA che porta al pin-strip. Se invece il bus viene impostato per interfacciarsi con dispositivi a 3,3V, le considerazioni sono comunque analoghe perché il traslatore comunque funziona similmente, con la sola differenza che quando il contatto SDA sul lato strip si trova a livello alto, drain e source sono equipotenziali e i MOSFET non intervengono.

L'unica linea del modulo pedometro U3 che viene portata direttamente al pin-strip della breakout board, senza passare da alcun traslatore di tensione (perché non serve, in quanto non c'è il problema del riferimento al positivo di alimentazione, come nell'I²C-Bus) è la STEP (pad 7 del modulo) che raggiunge il contatto IRQ del suddetto strip; questa fornisce impulsi a livello logico alto, uno per ciascun passo rilevato attraverso il sensore MEMS

e decifrato dal microcontrollore di bordo, secondo l'algoritmo del costruttore.

Tale linea può essere utilizzata se si desidera avere un conteggio "nudo e crudo" dei passi espletato da un contatore indipendente con display, invece che l'elaborazione dei dati da parte di un software. Concludiamo l'analisi dello schema elettrico con una considerazione sul jumper di selezione della tensione sul bus, JVIO: per mantenere la breakout board con il più basso profilo possibile, abbiamo previsto che il ponticello non sia il classico formato da un pinstrip 3 vie a passo 2,54 ma che sia ottenuto stagnando piazzole attigue.

REALIZZAZIONE PRATICA

Bene, ora che abbiamo descritto lo schema elettrico possiamo passare alle note costruttive, ossia a spiegare come realizzare la breakout board; come di consueto, abbiamo disegnato un circuito stampato del quale vi rendiamo disponibili per il download le due tracce (si tratta di un doppia faccia) dal nostro sito web www.elettronica.in.it. Da esse potete procedere a preparare il PCB per fotoincisione, dopodiché una volta che lo avrete inciso e forato potete procedere al montaggio

piano di MONTAGGIO

Elenco Componenti:

- R1: 470 ohm (0603)
- R4, R5, R6, R7: 10 kohm (0603)
- R2: 6,8 kohm (0603)
- R3: 3,3 kohm (0603)
- C1, C2, C3, C4, C6: 1 µF 6,3V ceramico (0603)
- C5: 10 µF 6,3V tantalio (1206)
- U1: MCP1700T-3302E/TT
- U2: TMUX1219DBVR
- U3: STM201M
- Q1, Q2: BSS138W-7-F
- LD1: LED rosso (0603)

Varie:

- Circuito stampato S1601 (33x33 mm)

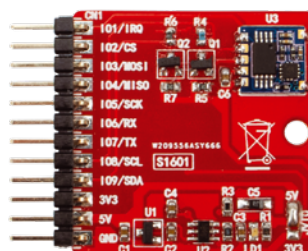
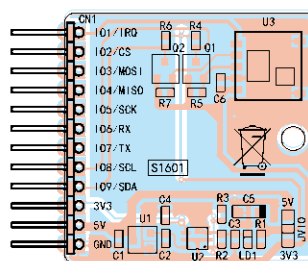
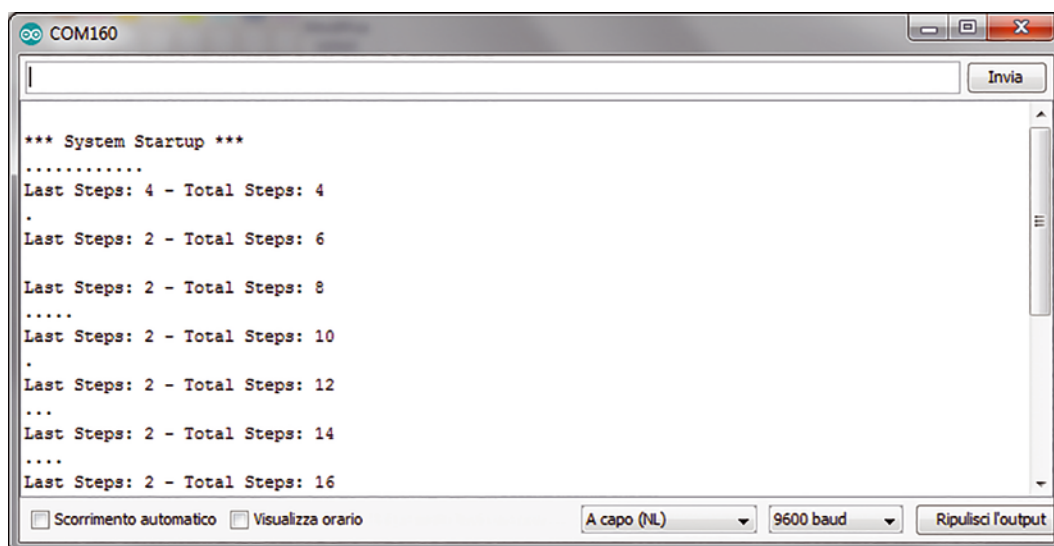


Fig. 3
Informazioni
fornite da Arduino
collegata alla
breakout board.



dei pochi componenti occorrenti. Trattandosi di componentistica SMD, è richiesto un minimo di attrezzatura ed occorre prestare anche un po' di attenzione ed avere manualità; l'attrezzatura del caso consta di un saldatore a punta molto fine, di filo di lega saldante (o pasta salda) del diametro massimo di 0,5 mm, della pasta flussante a media densità, una lente d'ingrandimento e una pinzetta per il posizionamento dei componenti.

Una volta procurato il tutto, prendete il circuito stampato, spalmate con la pasta flussante le piazzole destinate al modulo STM201M, quindi appoggiate questo componente ben centrato in esse e saldatelo appoggiando la punta del saldatore lateralmente insieme al filo di lega saldante; operate stagnando prima un contatto e poi, una volta che il modulo sarà stabile, stagnate i pad uno per volta operando lateralmente, alternando quelli di un lato a quelli del lato opposto. Notate che è impossibile montare l'STM201M nel verso sbagliato, perché ha due pad da un lato e cinque dall'altro, quindi l'orientamento è facilmente identificabile perché sul circuito stampato ci sono due piazzole da una parte e tre dall'altra.

Sistemato il modulino, spalmate di volta in volta le piazzole dei componenti con il flussante e saldate per prime le resistenze e i condensatori, poi il LED e i transistor MOS; questi ultimi possono essere saldati solo nel verso giusto, per l'univoca disposizione dei terminali, mentre per il condensatore elettrolitico e il LED dovete fare riferimento all'orientamento indicato nel piano di montaggio che trovate in queste pagine. Nel saldare i componenti badate di non creare cortocircuiti con "baffi" ed eccesso di lega saldante. Infine inserite nei

rispettivi fori il pin strip ad angolo retto a 12 poli e saldatelo, tenendolo ben allineato alla superficie del PCB e poi, con una goccia di stagno, chiudete il jumper JVIO verso la linea di alimentazione che desiderate. Il pin-strip è utile per realizzare le connessioni con Arduino tramite i classici jumper a filo maschio/femmina, ma potete anche farne a meno, se volete saldare direttamente i fili sulle piazzole laterali che riportano le connessioni del circuito. A riguardo, nella **Tabella 2** proponiamo i collegamenti da effettuare tra i pad laterali e la scheda Arduino, per implementare l'applicazione demo proposta in queste pagine.

LA NOSTRA APPLICAZIONE

Per testare le funzionalità della breakout board ed in particolare per prendere dimestichezza con il sensore contapassi abbiamo sviluppato un'applicazione dimostrativa che punta su uno sketch per Arduino scritto appositamente, che potete scaricare dal nostro sito www.elettronica.in insieme ai file di questo progetto, tra cui le librerie per la gestione della breakout board.

Scaricate questi file sotto la cartella "Arduino" e a questo punto potete aprire l'IDE di Arduino, impostarlo per la scheda Arduino UNO (cui il progetto si rivolge...) mediante il comando di Scheda: del menù Strumenti, quindi aprire il file dello sketch (**pedometro.ino**) e caricarlo nella board.

Una volta avviato il codice di esempio, che permette di leggere il numero di passi e stamparlo sulla porta seriale (quindi mostrarlo a video nella finestra del monitor seriale dell'IDE Arduino) indicando l'ultimo numero di step rilevato e il totale contabilizzato dall'accensione, con il Monitor Seriale



assegnato alla COM cui fa capo l'USB della scheda Arduino, potete visualizzare una situazione come quella proposta nella **Fig. 3**, naturalmente a patto di indossare il sensore quando si è in movimento (anche solo su un tapis roulant).

Il codice di gestione è molto semplice, in quanto non appena il sensore comunicherà la registrazione di un passo mediante il pin IRQ che andrà basso, Arduino rileverà mediante interrupt il cambiamento di stato e mostrerà su seriale il numero di passi. Nel codice reso a disposizione vi è il file

"STP201M.h" che include il codice che permette di comunicare direttamente con il sensore via I2C software sui pin 11 (SDA) e 12 (SCL), restituendo attraverso la funzione "MCU_Step_Request" il valore intero relativo al numero dei passi appena letto. A tal proposito, va precisato che se si effettua la lettura mediante interrupt al cambiamento di stato del pin, il valore dei passi letto sarà sempre il medesimo, ma se la lettura la si fa cadenzata ogni 10 secondi, o comunque un tempo prestabilito, leggerete un valore ogni volta diverso perché rappresenterà il valore reale dei passi che ci sarà stato in quel frangente di secondi.

Il file "**Pedometro.ino**" invece integra tutte le chiamate di base previste per Arduino, tra cui "**loop()**", "**setup()**" e tutte le dichiarazioni del caso.

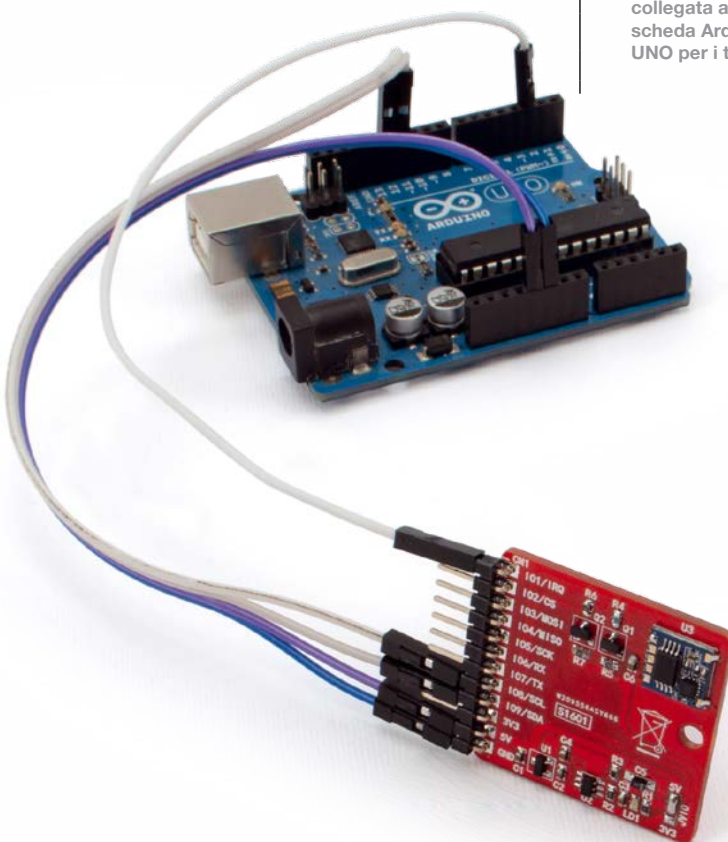
Vi ricordiamo che, come indicato poc'anzi, nella nostra applicazione dimostrativa stiamo usando la connessione I²C-Bus Software, perché date le caratteristiche del sensore, questo non può essere gestito via I²C-Bus Hardware; ciò pertanto vi rende liberi di collegare il sensore a qualsiasi pin digitale di Arduino cui andare ad applicare la libreria. La libreria legata alla I²C software può essere scaricata direttamente dal gestore delle librerie, ma in ogni caso è stata resa disponibile per il download dal sito della rivista, pertanto troverete una cartella "libraries" che conterrà la libreria in questione.

BREAKOUT	ARDUINO
GND	GND
5V	5V
SDA	pin 11
SCL	pin 12
IRQ	pin 2

Tabella 1
Conessioni
fra la
breakout
board e
Arduino.

Questa cartella potrà essere copiata direttamente all'interno della cartella documenti dell'utente di sistema, ovvero in "**..\Documents\Arduino**"

Il codice fornito è solo un esempio pratico relativo alla lettura dei passi, che sfrutta il byte meno significativo di quelli della stringa di risposta restituita dal sensore quando viene interrogato. In questo modo il conteggio si effettua a livello software evitando eventuali overflow, ma approfondendo il datasheet si potrà anche effettuare un reset dei passi direttamente via I²C-Bus.



La breakout board
collegata alla
scheda Arduino
UNO per i test.



Cosa occorre?

La breakout board contapassi (cod. FT1601M) è in vendita a Euro 22,50 con tutti i componenti SMD preassemblati, il modulo STM201M (cod. STM201M) è disponibile anche separatamente a Euro 15,50. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 - <http://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.



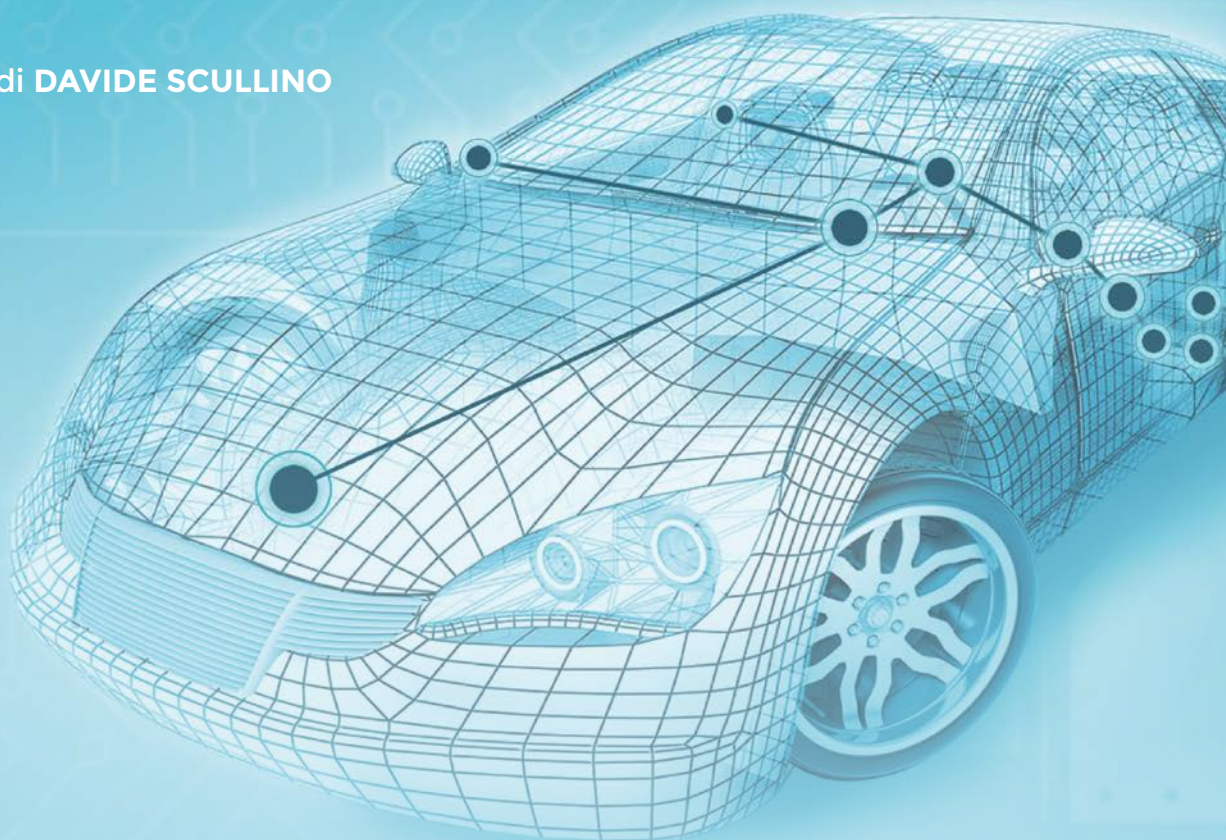
**FUTURA
ELETTRONICA®**

www.futurashop.it

Futura Group srl
Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775

SOLUZIONI NELL' AUTO ELETTRICA

di DAVIDE SCULLINO



Scopriamo le tecnologie in uso nella propulsione elettrica e impariamo a distinguere tra le varie tipologie di auto in commercio e del futuro.

Full Electric, Hybrid, Full Hybrid, Mild Hybrid e chi più ne ha più ne metta: sembra proprio che il vasto assortimento di nomi, più o meno rispondenti alle soluzioni tecnologiche, sia finalizzato quasi più a confondere le idee e far scegliere un'auto invece di un'altra, piuttosto che a spiegare cosa sia esattamente una vettura, in modo che un acquirente possa fare una scelta consapevole. In questo articolo proveremo a spiegare in maniera semplice quali sono le tecnologie oggi in campo (o meglio, in concessionaria...) impiegate nella trazione delle auto elettriche e ibride, specialmente per chiarire le idee a chi vorrebbe comperare un'auto "alternativa" e "green" ma non ha particolari competenze nel settore automobilistico. Dopo aver letto queste pagine, capirete che esistono auto più

o meno ibride, più o meno ecologiche, più o meno rispondenti alle aspettative di chi si alleggerirà il portafogli per acquistarle.

ELETTRICHE A CONFRONTO

Nel mondo dell'auto elettrica coesistono varie tecnologie, distinte per famiglie e per tipo e modalità di trazione: l'automobile che ha solo la propulsione elettrica è detta Full Electric, mentre quella in cui si trovano, cooperanti o in maniera esclusiva, una propulsione elettrica ed una a motore endotermico, è detta Ibrida. Ma attenzione, perché da qualche anno si utilizza quest'ultimo termine in maniera impropria, improvvida e soprattutto finalizzata ad attirare il cliente con la prospettiva che avrà un beneficio fiscale nell'acquisto; infatti ibrido in sé vuol dire poco, perché nella categoria si trova un ampio spettro di soluzioni tecniche. C'è l'auto dove il motore elettrico può provvedere da solo alla trazione, spegnendo il motore endotermico: questa è la vera auto ibrida, per distinguere la quale da altre soluzioni, si utilizza tecnicamente (o forse più commercialmente...) l'espressione Full Hybrid. E poi c'è l'auto dove il motore elettrico è di piccola potenza e spesso si tratta di un alternatore reversibile BSG (Belt-driven Starter Generator, ossia alternatore/starter azionato a cinghia) che da solo non può muovere l'automobile, ma che può cooperare con il motore endotermico, il quale deve quindi essere in funzione durante la marcia; in questo caso non c'è una vera trazione elettrica ma un booster elettrico, nel senso che il generatore reversibile aiuta la trazione fornendo un massimo di 4,8 kW (tale soluzione viene offerta anche su

motori da 120 kW, quindi l'apporto è scarso). Ecco svelato il segreto del Mild Hybrid, che in definitiva non assicura una trazione ibrida ma pare essere più una soluzione per convincere il cliente che sta comperando un'auto ibrida e che quindi gli darà benefici fiscali. In verità esiste anche la soluzione Toyota HSD, composta da due motori elettrici che possono essere combinati: uno fornisce la trazione esclusiva in alternativa al motore endotermico ma può fare da booster, insieme al secondo, che implementa anche le funzioni di starter e start & stop. A prescindere dall'esclusività o meno della trazione elettrica, tutte le ibride e le full-electric sfruttano una batteria e la differenza tra le varie tecnologie è nel come questa viene ricaricata. In tutte le architetture si fa uso della frenata rigenerativa, vale a dire che si sfrutta la reversibilità del motore elettrico per produrre elettricità, da immagazzinare nell'accumulatore. In aggiunta a questa, è possibile disporre di una soluzione di ricarica da rete elettrica, che semplificando avviene tramite una "presa di corrente" monofase o trifase, ovvero in continua: le auto full-electric o ibride ricaricabili attraverso la rete elettrica vengono chiamate "plug-in"; esistono quindi le plug-in hybrid, quando si tratta di ibride con la presa, mentre le full-electric sono evidentemente plug-in perché altrimenti non sarebbe possibile ricaricarle o quantomeno dare la carica iniziale.

FRENATA RIGENERATIVA

Anche nota come KERS, ossia Kinetic Energy Recovery System, è una tecnologia che permette di recuperare, in rallentamento e in frenata, parte

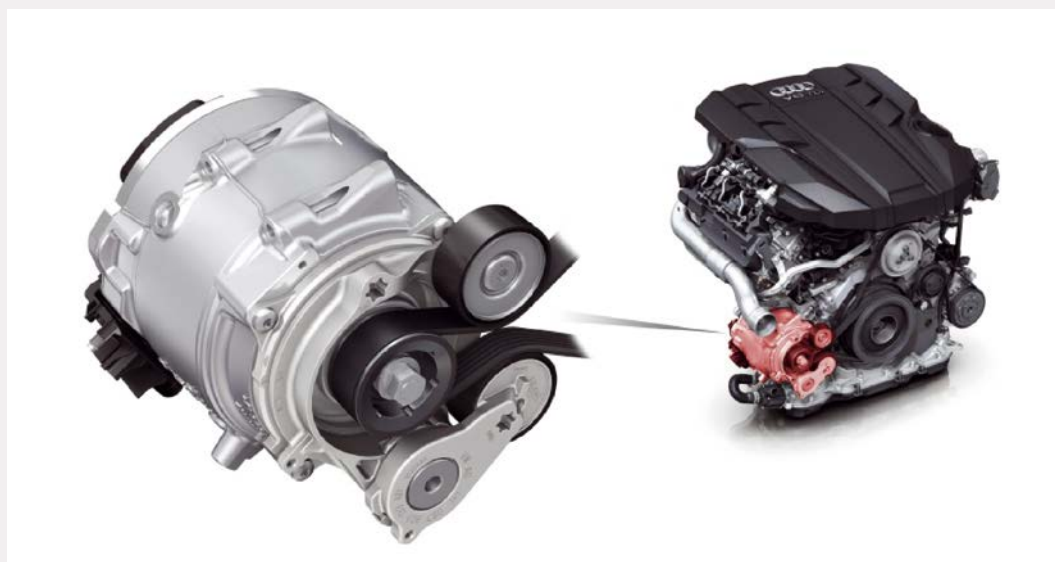


Fig. 1
BSG per auto
Mild-Hybrid (Audi)
secondo
lo schema P0.



Fig.2
Connettori
Combo2 (a
sinistra) e
ChaDeMo
(destra).

dell'energia cinetica posseduta dall'automobile, la quale è direttamente proporzionale alla massa e al quadrato della velocità istantanea; rimanendo costante la massa del veicolo e diminuendo, la velocità, man mano che gli attriti e la reazione d'indotto del motore elettrico esercitano la loro azione frenante, tale energia cinetica diminuisce progressivamente fino ad annullarsi a ruote ferme. L'energia cinetica, con un certo rendimento che dipende dalla bontà delle soluzioni adottate, soprattutto dall'elettronica di controllo, si trasforma in elettricità immagazzinata nell'accumulatore e pronta per essere riutilizzata.

Il KERS è integrato con il sistema di ricarica della batteria e per l'esattezza, con il BMS (Battery Management System) che è l'elettronica di gestione dell'erogazione e dell'accumulo di energia elettrica nella batteria di trazione, la quale è composta da combinazioni serie-parallelo di celle al litio, ciascuna avente una tensione nominale di $3,6 \div 3,7V$. Elemento chiave del BMS è il bilanciatore di carica delle celle, indispensabile perché quando si pongono varie celle al litio in serie (nelle batterie delle ibride possono esserci serie di 150 celle!) è necessario compensare la corrente assorbita da ciascuna in fase di carica in funzione della tensione assunta da ciascuna e letta da un A/D converter veloce, che insieme a quelli delle altre celle invia i dati alla CPU affinché provveda alle correzioni del caso. La compensazione si rende necessaria per via della tensione nominale degli elementi al litio e della fluttuazione della corrente in essi per piccole variazioni della tensione ai capi (ricordiamo che le celle al litio hanno una carica che prevede fasi a corrente costante e a tensione costante) e che celle con tensione maggiore di altre inevitabilmente

assorbirebbero meno corrente; in una connessione serie questo implicherebbe che la celle meno cariche non riuscirebbero a caricarsi correttamente, essendo la corrente di una serie unica. Il bilanciatore di carica invece permette di assumere la tensione di ogni singola cella e intervenire shuntando la cella stessa in maniera più o meno rilevante in base alla differenza di potenziale misurata, secondo questa logica: se la tensione è abbastanza alta rispetto alle altre deriva una parte della corrente destinata a quella cella, mediante un transistor che viene portato in conduzione, ovvero limita la corrente sottratta dal transistor se la tensione della cella è bassa, perché questa deve caricarsi più rapidamente.

Il BMS può essere alimentato esclusivamente dal circuito del KERS, quindi da un convertitore DC/DC o AC/DC collegato al motore di trazione quando l'acceleratore è rilasciato, ma anche, nelle vetture plug-in, dal caricabatteria (anche noto come EVSE, acronimo per Electric Vehicle Supply Equipment). Quest'ultimo può essere di vario tipo a seconda della sorgente di alimentazione, ovvero alimentato a corrente continua (ricarica DC) o in alternata, monofase o trifase; il tipo di tensione dipende dalle scelte fatte dal produttore dell'automobile e da quanto vuole risparmiare, perché ad esempio accettare solo la ricarica in continua significa non dover implementare a bordo dell'auto lo stadio alimentatore c.a./c.c. con cui alimentare poi il caricabatteria interno che interagisce con il BMS. Sulle opzioni di ricarica andrebbero fatte molte considerazioni, sebbene in questa sede ci interessa solo evidenziare come la soluzione DC sia quella che consente la ricarica più rapida e meno critica, in quanto lascia all'EVSE o colonnina di ricarica

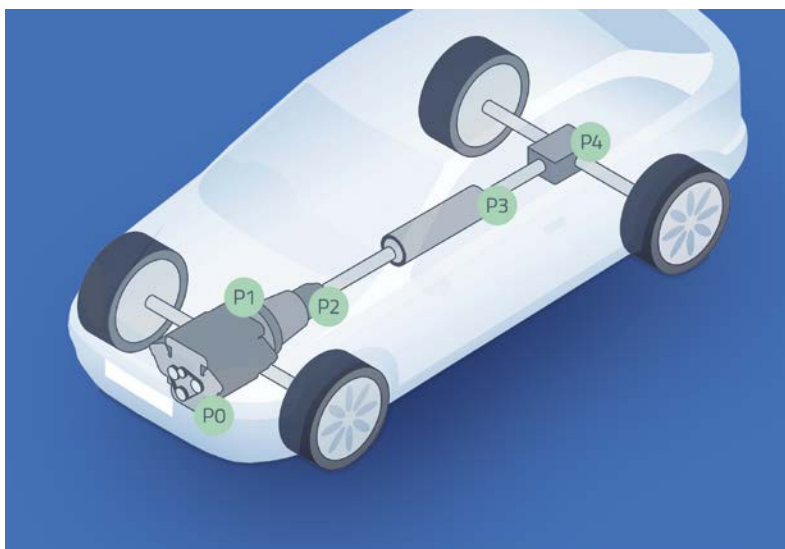


Fig. 3
Schemi di
ibridazione.

pubblica la parte di conversione dall'alternata alla continua ed evita il riscaldamento a bordo; tra le opzioni in alternata, chiaramente quella trifase permette maggior potenza erogabile e quindi ricariche più veloci, compatibilmente con le caratteristiche elettriche del sistema caricabatteria di bordo. Con i sistemi in DC si arriva tranquillamente a 150 kW, che corrispondono a ricaricare una batteria da 80 kWh in poco più di mezz'ora. A livello di colonnine stradali, esistono anche stazioni da 300 kW, orientate prevalentemente ai mezzi pesanti. Ad ogni tipo di tensione corrisponde uno standard del connettore di ricarica e a riguardo va precisato che le due prese più usate sono la ChaDeMo e la CCS Combo2 (**Fig. 2**).

Tornando alla carica che viene eseguita in autonomia a bordo del veicolo, l'energia elettrica ricavata durante la frenata rigenerativa è tanto maggiore quanto più il veicolo rallenta, quindi nei percorsi misti e urbani: qui si va a recuperare energia altrimenti dispersa in attrito e nei freni; l'energia viene recuperata anche in discesa, dove un'auto a motore endotermico avrebbe un consumo ridottissimo dovuto al "minimo" oppure, in presenza di cut-off del combustibile (motori a benzina e diesel common-rail), pari a zero. Nell'elettrica, invece, in rallentamento e in discesa, il freno motore comporta un recupero di quell'energia altrimenti dissipata in altra maniera. Ecco il perché una vettura elettrica, sia essa ibrida o full-electric, consente un risparmio reale sui consumi: quella full-electric, quindi, in realtà percorre più chilometri di quanti ne consentirebbe la carica della batteria eseguita dalla rete elettrica o colonnina di distribuzione, mentre la ibrida fornisce autonomia aggiuntiva.

Per fare un esempio, una vettura con motore da 80 kW che viaggi a metà potenza per 2 ore richiederebbe 80 kWh di ricarica, ma sfruttando la frenata rigenerativa potrebbe viaggiare anche per tre ore alla stessa potenza, proprio recuperando i percorsi a motore inattivo.

Chiaramente se l'auto viaggia a velocità costante senza mai fermarsi, se non a batteria scarica, la frenata rigenerativa non dà alcun vantaggio pratico, ed ecco perché full-electric e ibride danno il meglio nei percorsi misti e cittadini, dove invece le fermate e i rilasci coprono una parte più o meno cospicua del tragitto. Il limite del risparmio è dettato dalla capacità della batteria, nel senso che quando è carica non può essere caricata ulteriormente, pertanto se si prende il caso di una full-electric che poco dopo essere partita a piena carica percorre un lungo tratto in discesa, difficilmente sarà possibile accumulare sotto forma di elettricità l'energia cinetica neutralizzata dal KERS; diciamo che parte di quell'energia potrà essere sfruttata convenientemente dagli utilizzatori di bordo, come ad esempio la pompa di calore per la climatizzazione e la stabilizzazione della temperatura del pacco batteria. In questo contesto rientra anche la temperatura esterna, perché quando fa molto freddo l'efficienza della ricarica è nettamente minore, in quanto le batterie al litio, che sono alla base della moderna trazione elettrica automobilistica, si caricano e scaricano ottimamente quando gli elettrodi sono intorno ai 45-50 °C; non è un caso che l'impianto di raffreddamento a liquido del motore (o dei motori) sia derivato verso la batteria, allo scopo di mantenerne la temperatura nel campo ottimale. Questo problema è più sentito dalle ibride, in special modo full-hybrid, perché se fa molto freddo non possono funzionare a trazione elettrica, almeno nei primi minuti: serve che il motore endotermico si scaldi e che il liquido del circuito di raffreddamento scaldi un po' la batteria, così da poterla ricaricare efficacemente e utilizzare in modo ottimale.

SCHEMI DI IBRIDAZIONE

Mentre nelle auto puramente elettriche c'è solo un motore o un gruppo di motori elettrici, in quelle ibride esistono combinazioni con il motore endotermico, che rientrano nel concetto di schemi di ibridazione; nello specifico, esistono cinque diverse posizioni di installazione possibili per il motore elettrico e varie soluzioni di accoppiamento. Le descriviamo qui di seguito e schematizziamo nella **Fig. 3**:

- P0, dove il motore elettrico è accoppiato a

cinghia al motore endotermico, quindi la sua forza viene trasmessa all'albero del motore endotermico attraverso la cinghia dei servizi e la medesima puleggia che lo aziona quando fa da generatore. Il motore, che in questo caso è un BSG (è un alternatore reversibile) è allineato con la trasmissione. Lo schema P0 viene adottato nelle ibride entry level con ibridazione a 48 volt e il generatore reversibile funziona a tale tensione, funzionando anche in modalità start & stop. Da motore, si aggiunge a quello endotermico quando serve maggiore potenza (boost) e se la batteria di trazione (distinta da quella di avviamento) è sufficientemente carica. Il recupero di energia è scarso perché in rallentamento, parte dell'energia cinetica in arrivo dalla trasmissione viene neutralizzata dall'azione frenante del motore.

- P1, dove c'è ancora un motore elettrico reversibile che funziona da booster (e da starter), però si adotta una disposizione che riduce l'ingombro di installazione e agisce mediante catena o ingranaggi, sull'albero motore. È una soluzione semplice ed economica, ma come nel caso precedente il motore a combustione interna e la trasmissione non possono essere disaccoppiati, quindi ne deriva che il recupero di energia cinetica è poco efficiente (vale quanto spiegato per lo schema P0) e non è permesso il funzionamento a trazione puramente elettrica. Un esempio è in **Fig. 4**.
- P2, in questo schema il motore elettrico è posizionato tra il motore a combustione (può

essere disaccoppiato) e la trasmissione; l'accoppiamento avviene ad opera di una frizione o sistema elettromagnetico. La soluzione è economica, compatta e facilmente integrabile, anche e soprattutto adottando cambi di velocità robotizzati. Questo schema (**Fig. 5**) permette la modalità "boost" e "coasting" (trazione in parcheggio) nonché la trazione elettrica pura. Il recupero di energia è ottimizzato dal fatto che il motore elettrico può essere collegato in esclusiva al resto della trasmissione e pertanto il motore endotermico non frena, giacché può essere sconsesso. Consumo ed emissioni sono significativamente ridotti.

- P3, in questo schema il motore elettrico può funzionare come trazione esclusiva ed essere accoppiato esclusivamente alla trasmissione per ottenere il massimo recupero di energia. Può essere accoppiato e disaccoppiato dal motore endotermico con una frizione elettrica e spesso supporta due velocità. Il motore elettrico fa presa sull'uscita del cambio o della trasmissione principale (albero di trasmissione). Il risparmio sui consumi con le soluzioni P2 e P3 si colloca intorno al 20÷22%.
- P4, questo schema di ibridazione prevede un assale separato a trazione elettrica non collegato alla motore a combustione interna. Consente la trazione integrale temporanea (booster) combinando la trazione elettrica con quella a motore tradizionale; in questo caso la sovrapposizione dei due motori è gestita

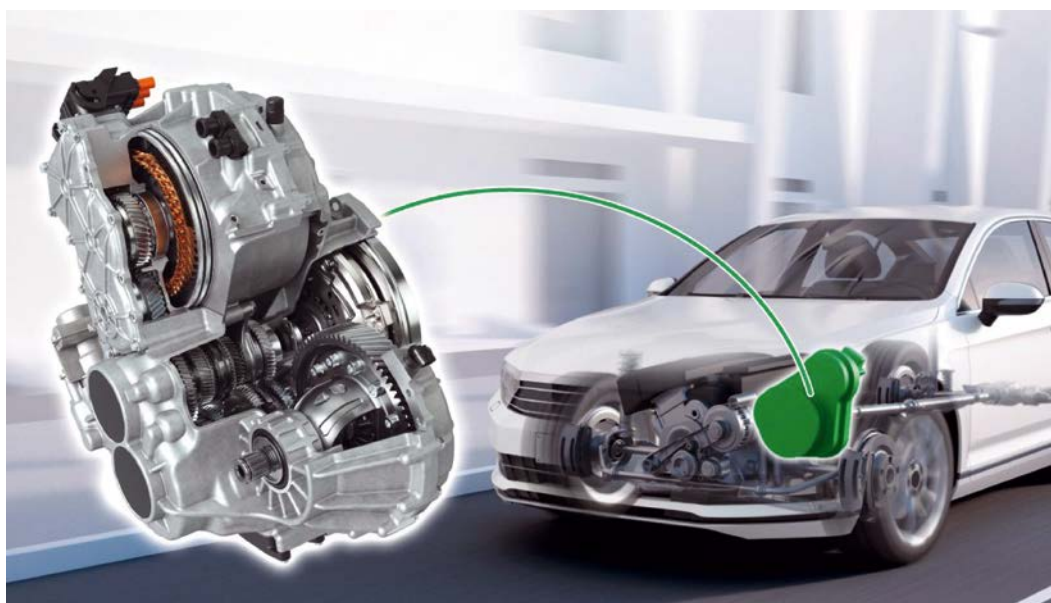


Fig. 4
Motore elettrico applicato all'uscita del cambio (Schaeffler) con schema P1.

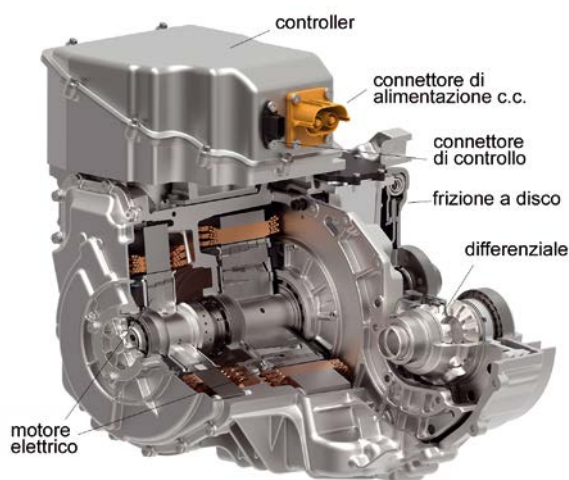


Fig. 5
Gruppo di
ibridazione
a valle della
frizione (P2).

dall'elettronica rilevando i giri delle ruote e lo sforzo di trazione. È anche possibile spegnere il motore endotermico, sconnetterlo con la frizione ad azionamento elettrico, quindi usare per la trazione l'asse elettrificato. Il risparmio sui consumi in questo caso arriva al 22%.

Da questa descrizione appare evidente come le vere auto ibride siano quelle in grado di provvedere alla trazione elettrica alternativa a quella a motore endotermico e pertanto quelle rispondenti agli schemi P2, P3 e P4, che sono da considerare dei full-hybrid; invece P0 e P1 sono considerati Mild Hybrid. Nel primo caso il risparmio sui consumi è decisamente importante e sale tanto più quanto maggiore è la capacità della batteria, nonché quanto più sono presenti momenti di rilascio o frenata

(quindi nel percorso urbano o misto); nel secondo è marginale, mediamente nell'ordine del 10%.

Resta valido il ragionamento esposto pochi paragrafi indietro, sulla temperatura dell'ambiente esterno e della batteria.

Gli schemi di ibridazione P2 e P3 supportano spesso l'alimentazione del motore elettrico a bassa tensione, di 48Vcc; il P4 è normalmente ad alta tensione (400÷600V in base alla potenza) mentre P0 e P1 hanno motori/generatori a bassa tensione, talvolta a 12V ma quasi sempre a 48V.

Il gruppo motopropulsore adottato nello schema P4 è identico a quello di un assale di trazione per auto puramente elettrica, perché infatti ha la stessa tecnologia; anzi, se si rimuovesse il motore a combustione interna da un'auto ibrida con schema P4, si otterrebbe una Full Electric a trazione anteriore o posteriore.

La **Fig. 6** propone un esempio di gruppo per ibridazione P4 su ruote posteriori, ovvero di trazione Full Electric, adatto sia alla trazione posteriore che a quella anteriore, dotato di differenziale e semiassi con giunti omocineticici, mosso da un motore trifase/BLDC.

SOLUZIONI DI TRAZIONE

La trazione delle auto elettriche si basa evidentemente su motori elettrici, che possono essere di vario genere: alternatori reversibili (BSG) nelle Mild Hybrid, tipicamente funzionanti a 48Vcc, motori coassiali posti sull'albero motore (funzionano anche come generatore e starter, ma a volte il

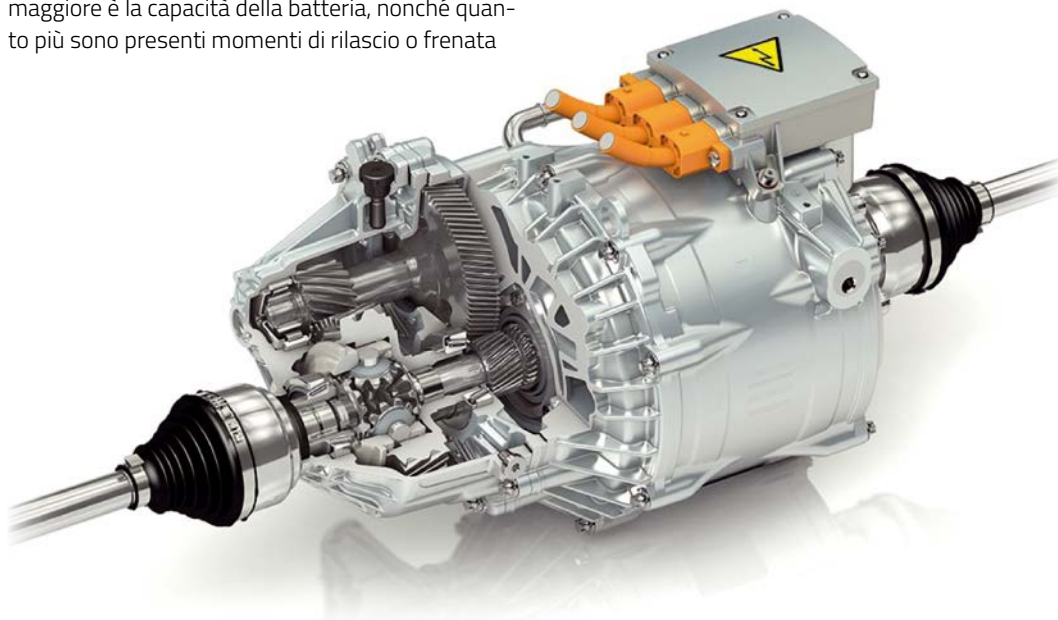


Fig. 6
Gruppo di trazione
elettrica con
differenziale
e giunti.

generatore per l'impianto elettrico di servizio è tradizionale) sempre a 48Vcc, motori ingranati lateralmente alla trasmissione sull'uscita del cambio o su un albero di esso, quando si abbinano a cambi a doppia frizione come il noto DSG (Double Shift Gearbox) del gruppo VAG e la sua controparte ESG (Electrical Shift Gearbox) della LuK; ma esistono anche altre soluzioni.

I motori possono essere in continua o in alternata, di tipo sincrono: nel primo caso si prediligono i brushless trifase, mentre nel secondo si adottano spesso motori a magneti permanenti sul rotore. Il controllo è rigorosamente elettronico, operato sul motore da un driver di potenza pilotato, anzi interfacciato, alla centralina di gestione della trazione, la quale:

- nell'auto ibrida coopera con la ECU del motore endotermico, ad un livello gerarchico superiore perché la centralina principale dell'auto deve governare anche il funzionamento del motore a combustione interna e, nelle Full Hybrid, poter anche spegnere quest'ultimo; la trazione viene controllata in accordo con i sistemi di sicurezza attiva (ABS/ESP, servosterzo a gestione elettronica) e ADAS facendo all'occorrenza cooperare i motori per implementare la trazione ibrida durante la marcia;
- nella Full Electric si occupa della gestione della trazione, sempre in accordo con i sistemi di sicurezza attiva (ABS/ESP, servosterzo a gestione elettronica) e ADAS, controllando il motore o i motori, eventualmente disposti sui due assi (trazione AWD o 4WD).

Per il recupero dell'energia cinetica sotto forma di corrente elettrica, nei momenti in cui l'acceleratore è rilasciato i cavi del motore elettrico vengono "sconnessi" dai dispositivi di potenza (MOSFET SiC o TRIAC) e "girati" su un circuito raddrizzatore e regolatore di carica, che va a fornire l'elettricità al BMS di bordo per cedere l'energia alla batteria. La fase di riutilizzo dell'energia è particolarmente complessa perché implica una fonte di energia incostante, dato che l'energia cinetica dipende, a parità di massa del veicolo, dal quadrato della velocità e che quindi in rallentamento decresce; il controller deve quindi essere in grado di sfruttare tale energia inviando picchi di corrente elevati in breve tempo, mantenendo però la tensione a valori accettabili. La carica ha quindi una fase dinamica da gestire istante per istante: una condizione ben diversa dalla normale ricarica effettuata tramite la rete elettrica.

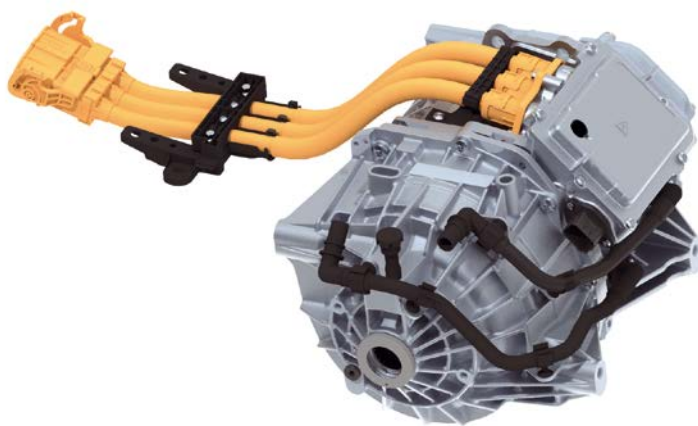


Fig. 7
Motore con ingresso trifase alimentato da Inverter.

Per il comando dei motori brushless viene utilizzato il classico controller trifase (a MOSFET o IGBT) che genera impulsi periodici a terne (ogni impulso è distanziato nel tempo di 1/3 di periodo dal successivo) la cui frequenza e larghezza sono modulati in base alle condizioni di marcia: la frequenza determina la velocità di rotazione del motore e quindi la velocità di marcia dell'automobile, mentre la larghezza (gestita con la tecnica PWM) condiziona la potenza sviluppata dal motore, ovvero più aumenta, più forza il motore trasmette alle ruote e viceversa.

La frequenza viene impostata in base al comando dell'acceleratore o del cruise-control (sistema ad anello chiuso che mantiene la velocità di marcia al valore impostato dal conducente) e il PWM è gestito da un controllo ad anello chiuso che verifica se a una data richiesta tramite l'acceleratore corrisponde effettivamente la velocità desiderata: se questa è inferiore viene aumentata la larghezza degli impulsi e quindi la potenza alla ruota e viceversa.

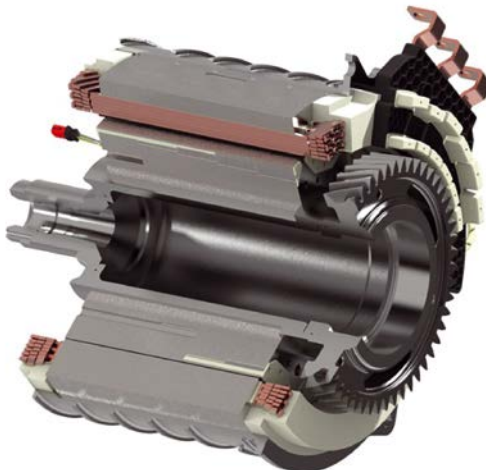
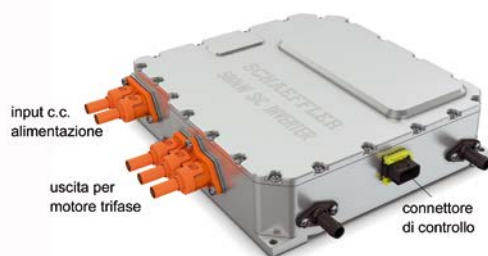


Fig. 8
Anima di motore con lamelle di connessione al gruppo inverter integrato.

➔ **Fig. 9**
Inverter per controllo
di motori trifase da
180 kW.

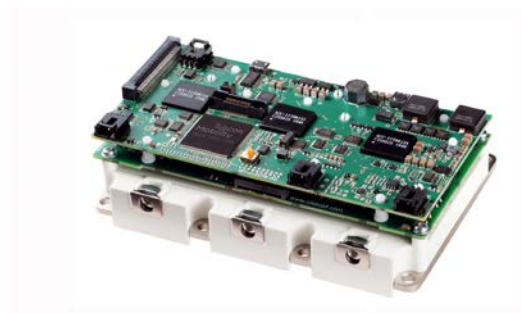


Il controller può sia essere distinto, sia essere applicato alla carcassa del motore; il motore ha, nel primo caso, tre cavi (come nel caso della **Fig. 7**) mentre nel secondo ne riceve 2 e internamente è poi connesso ad esempio tramite contatti a lamina, alle uscite del controller (**Fig. 8**).

Per il controllo dei motori in corrente alternata, il sistema è analogo ma si ricorre a TRIAC (uno per fase) o a ponti variamente costituiti da MOSFET SiC, operando in parzializzazione d'onda, di preferenza mediante la tecnica trailing-edge, vale a dire parzializzando sul fronte di discesa dell'impulso di semionda della sinusoide.

Questo tipo di propulsione fa uso di un inverter che ricava l'alternata dalla componente continua resa disponibile dalla batteria (un esempio è in **Fig. 9**, dove vediamo anche l'elettronica di un caratteristico inverter SiC) e nello stesso gruppo genera le fasi per il controllo a parzializzazione; è in pratica l'unione di un inverter e di un variatore di potenza, come una sorta di chopper.

L'acceleratore controlla la parzializzazione d'onda, intervenendo come segnale di ingresso di un sistema retroazionato dove il segnale di errore deriva dalla velocità reale, che viene confrontata con quella attesa.



I generatori elettrici reversibili usati negli schemi di ibridazione P0 (BSG) e P1 hanno normalmente due poli e sono gestiti da un proprio controller che funziona sia da raddrizzatore come nei tradizionali alternatori, sia da driver trifase per gli avvolgimenti statorici; nel funzionamento da generatore, la corrente di eccitazione percorre gli avvolgimenti statorici e variandone l'intensità si determina la corrente e la tensione erogata, mentre in quello da motore modula la potenza erogata al motore endotermico. Due esempi di booster sono proposti nella **Fig. 10**.

Per quanto riguarda la trazione Full Hybrid trifase, si adottano motori PMSM (Permanent Magnet Synchronous Motor) ossia motori sincroni ma con il rotore a magneti permanenti invece che avvolto con gli avvolgimenti rotorici alimentati in continua. Questi motori vengono realizzati in due versioni: "surface-mounted" dove i magneti sono montati sulla superficie esterna del rotore e "interior-mounted", dove i magneti risultano montati nell'interno della struttura stessa del rotore.

Il vantaggio del tipo surface-mounted è nell'elevatissima coppia specifica, ma siccome la permeabilità magnetica dei magneti permanenti è simile a quella dell'aria, tale struttura del rotore comporta

➔ **Fig. 10**
Due generatori
reversibili: a sinistra
un BSG (schema
P0) e a destra il tipo
da ingranare lato
motore (P1).



un maggior traferro effettivo. Per quanto riguarda il tipo interior-mounted, offre una protezione fisica ai magneti (tenuta alla forza centrifuga). L'uso dei rotori surface-mounted, che hanno un'alta riluttanza magnetica, mantiene il flusso del rotore relativamente costante, tuttavia per estendere il campo di velocità, è possibile ottenere l'equivalente del deflussaggio dei motori DC a eccitazione separata, attraverso la generazione sullo statore di un opportuno controcampo magnetico di compensazione.

Seppure siano analoghi ai motori brushless in continua a magneti permanenti (BDCM), i PMSM si distinguono per la diversa forma d'onda delle correnti statoriche, la quale è sinusoidale invece che ad impulsi rettangolari. Ne deriva che mentre il brushless in continua è caratterizzato da una maggiore pulsazione della coppia causata dalla brusca variazione della corrente durante la commutazione da parte dei dispositivi di potenza, il PMSM ha una coppia più dolce dovuta all'interazione di corrente sinusoidale e flusso sinusoidale, oltre ad avere perdite inferiori.

I motori con rotore a magneti permanenti hanno coppia molto elevata e sono attualmente quelli a più alta densità di potenza (possono superare i 10 kW/kg) oltre ad avere efficienze che possono superare il 90%; l'unica fonte di perdite è tendenzialmente (se si trascurano le perdite nei magneti permanenti indotte dallo switching dell'inverter, per le quali spesso i magneti vengono laminati come i lamierini in ferro) lo statore, che viene raffreddato solitamente a liquido per aumentare le prestazioni specifiche del motore.

Come per i motori in continua a magneti permanenti e quelli in alternata asincroni, per i PMSM è stata sperimentata la configurazione a disco (usualmente a flusso assiale), che ne permette l'uso direttamente nella trazione a motoruota (motore applicato al mozzo della ruota).

IMPIANTO ELETTRICO DI SERVIZIO

Come l'automobile tradizionale, anche quella elettrica ha equipaggiamenti elettrici estranei alla propulsione che devono funzionare a bassa tensione continua; si pone quindi il problema di come generare e immagazzinare l'elettricità del caso. Il tipico impianto elettrico di un'automobile funziona a 12V, ma per ragioni pratiche nelle ibride e ancor più nelle full-electric si tende a standardizzare l'impianto di servizio a 48Vcc. Ma vi sono eccezioni. Bisogna analizzare i vari casi, riassunti qui di seguito.

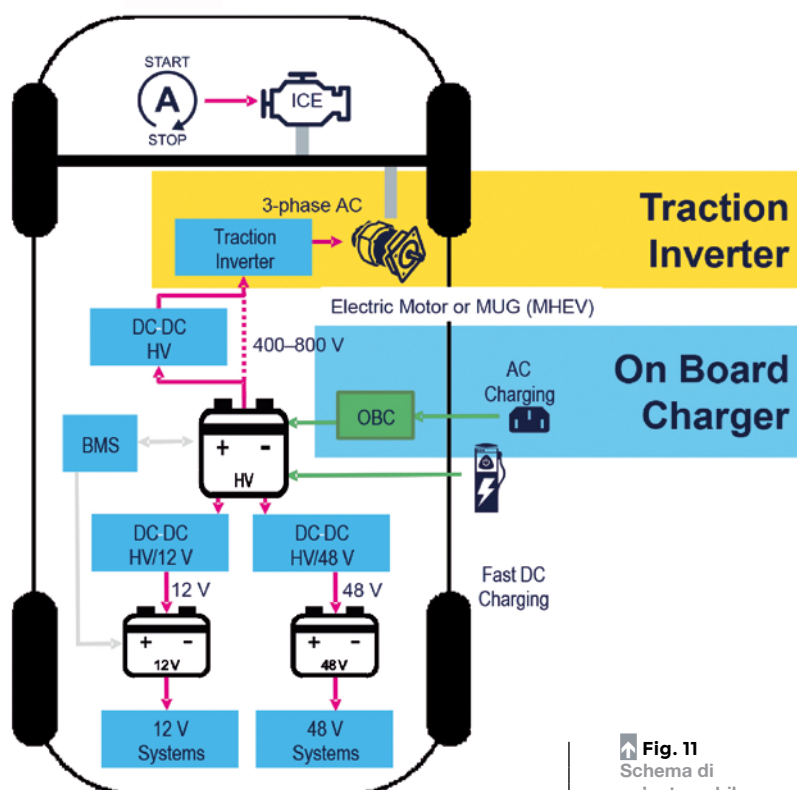
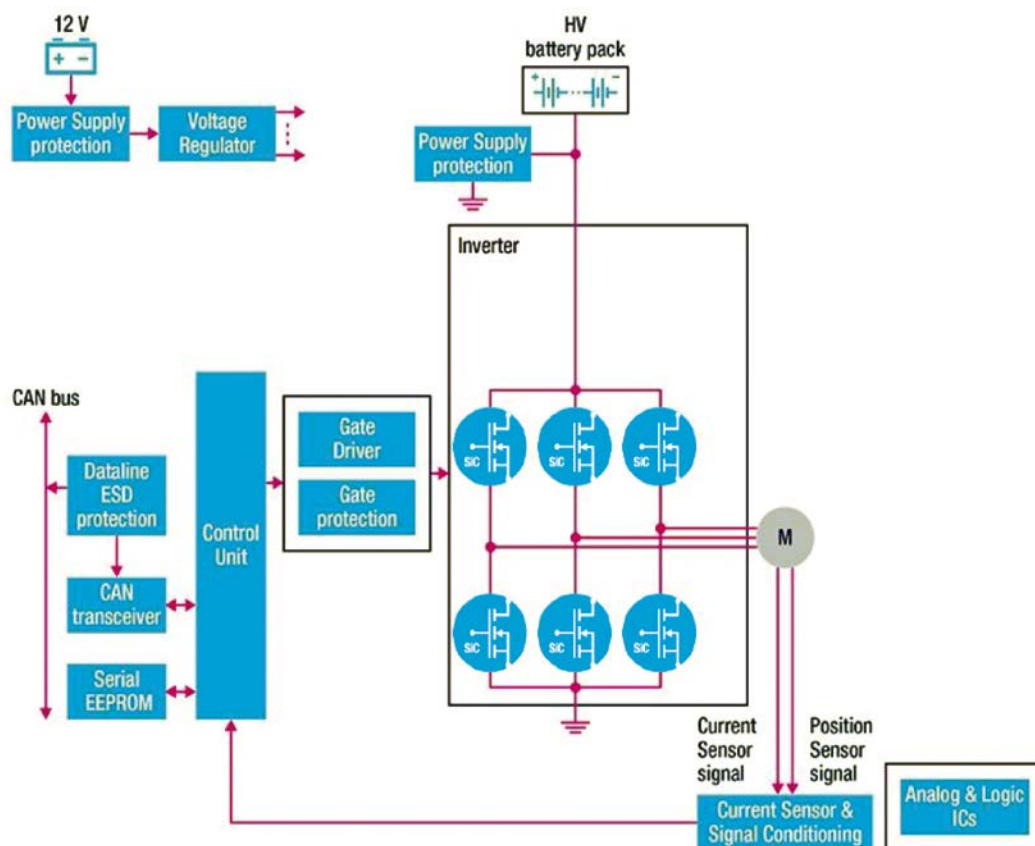


Fig. 11
Schema di
un'automobile
ibrida plug-in (ST).

1. Mild Hybrid con motore booster a 12V: qui l'alternatore fa da motore supplementare, eroga pochissima potenza e l'unico vantaggio è di mantenere un unico impianto elettrico, con batteria al piombo che serve sia alla funzione booster, sia durante l'avviamento del motore endotermico.
2. Mild Hybrid con motore di trazione a 48V: qui esistono una batteria al litio per la trazione (booster o esclusiva) ed una di servizio a 12V al piombo per l'avviamento del motore endotermico; sebbene l'alternatore reversibile possa anche fare da motorino d'avviamento, non è conveniente che sia esso ad avviare il motore a combustione interna perché quando fa molto freddo, la batteria al litio non fornisce un grande spunto (è la ragione per cui le ibride in inverno partono quasi sempre con il motore endotermico).
3. Full Hybrid con motore di trazione ad alta tensione; qui esistono una batteria al litio per la trazione esclusiva ed una di servizio, che può essere a 12V al piombo per l'avviamento del motore endotermico; sebbene l'alternatore reversibile possa anche fare da motorino d'avviamento, non è conveniente che sia esso ad

Fig. 12
Schema di massima
di un inverter/
controller per
motore trifase.



avviare il motore a combustione interna perché quando fa molto freddo, la batteria al litio non fornisce un grande spunto.

4. Full Electric; qui si adotta un DC/DC converter step-down per alimentare l'impianto elettrico di servizio, da fermo o in movimento.

Nel caso 2, quando si adotta lo schema P0 o quello P1, l'automobile è dotata di un DC/DC converter step-down che ricava 12V per la carica della batteria al piombo dai 48V forniti dal motore elettrico; se si desidera che il motore/generatore provveda anche all'avviamento del motore endotermico (in questo caso si risparmierebbe il motorino d'avviamento) per rendere possibile ciò partendo dalla batteria al piombo occorre anche uno step-up 12/48V, da mantenere acceso solo nella fase di avviamento. L'architettura del tipico impianto elettrico di un'automobile ibrida plug-in, integrante quello di servizio e quello di trazione, è rappresentata esaurientemente nella **Fig. 11**, tratta da documentazione della STMicroelectronics, azienda che da tempo opera nell'elettronica automotive e che spicca anche in quel "piatto ricco" che l'elettronica trova nell'auto del futuro. Nella suddetta figura

si vedono ben distinti i blocchi DC/DC per l'impianto elettrico di servizio a 12V, step-down per caricare la batteria e step-up per l'avviamento del motore endotermico, nonché l'inverter trifase per la trazione. Proponiamo anche, nella **Fig. 12**, lo schema a blocchi di un tipico inverter/controller per motore trifase di trazione, composto da tre mezzi ponti a MOSFET SiC disposti in simmetria quasi complementare, uno per ciascuna fase, nonché da una logica di controllo che genera gli impulsi di comando dei gate. L'adozione negli stadi di potenza, dei MOSFET SiC (in carburo di silicio) invece di quelli in silicio tradizionali o IGBT, consente di ottenere vantaggi tra cui consumi ridotti, ingombri fino a 5 volte inferiori, maggiore efficienza e possibilità di controllare motori con potenze più elevate.

CONCLUSIONI

Bene, termina qui questa rassegna sull'auto elettrica del presente; aspettando di conoscere e spiegarvi le soluzioni del futuro, speriamo di avervi dato un'infarinatura sulle tecnologie che stanno dietro alle auto elettriche o elettrificate che già guidate o che guiderete quando deciderete di cambiare quella a motore endotermico che avete. □

Contatore Geiger



€169,00
Cod. FR822

Misura l'intensità di raggi X, raggi gamma (γ) e raggi beta (β). Utilizza un microprocessore a 32 bit e dispone di ampio display LCD con retroilluminazione, tastiera a membrana, allarme acustico con soglia regolabile.



Air Quality

CONTROL

Purificatore d'aria da tavolo a raggi ultravioletti

MicroUVAIR è un sistema di sanificazione aria con ventilazione forzata a ciclo chiuso basato su led UVC. L'aria in ingresso viene purificata attraverso un filtro ad alta densità (30 PPI), quindi viene sterilizzata tramite luce ultravioletta emessa da LED UVC in grado di effettuare la massima azione germicida. Una volta purificata l'aria viene reimpressa nell'ambiente.

€199,00
Cod. MICROUVAIR-T



Prezzi IVA inclusa.

Misuratore di formaldeide e monitor della qualità dell'aria



Fornisce in tempo reale la quantità di formaldeide (HCHO) presente nell'ambiente circostante e monitora il livello di TVOC (Composti Organici Volatili Totali). È dotato di display per la lettura dei valori rilevati.

€21,90
Cod. FR832

HDMI
↓
VGA

CONVERTITORE DA HDMI A VGA 1080P PLUG AND PLAY

Permette di convertire un segnale digitale HDMI (compatibile con HDMI1.3) in un segnale VGA (SUB-D) ad alta definizione con audio in uscita. Con questo convertitore, il computer, il lettore video portatile, le console di gioco e alcuni altri dispositivi video con uscita HDMI possono connettersi a schermi HD di grandi dimensioni oppure a proiettori HD con ingresso VGA.

€ 11,50

Cod. **BOXHDMITOVGA**

Convertitori di segnale

Permettono di convertire un segnale digitale HDMI in VGA, AV in HDMI, VGA in HDMI, HDMI in AV.

CONVERTITORE DA AV A HDMI PLUG AND PLAY

AV
↓
HDMI

€ 14,00

Cod. **BOXAVTOHDMI**

Permette di convertire un segnale audio/video CVBS (PAL e NTSC) in ingresso in un segnale HDMI in uscita. Con questo convertitore, il computer, il lettore video portatile, le console di gioco e alcuni altri dispositivi video con uscita CVBS possono connettersi a schermi con ingresso HDMI. L'uscita HDMI supporta una risoluzione fino a 1080P 60Hz.

VGA
↓
HDMI

€ 11,50

Cod. **BOXVGATOHDMI**

CONVERTITORE DA VGA A HDMI PLUG AND PLAY

Permette di convertire un segnale audio e video VGA (SUB-D) in un segnale HDMI (tipo A) ad alta definizione con audio. Con questo convertitore, il computer, il lettore video portatile, le console di gioco e alcuni altri dispositivi video con uscita VGA possono connettersi a schermi HD di grandi dimensioni oppure a proiettori HD con ingresso HDMI. L'ingresso VGA supporta video fino a 1920x1080 a 60 Hz, l'uscita HDMI può supportare video fino a 1080P.

CONVERTITORE DA HDMI AD AV PLUG AND PLAY

Permette di convertire un segnale HDMI in ingresso in un segnale video composito CVBS (PAL e NTSC) in uscita. Con questo convertitore, il computer, il lettore video portatile, le console di gioco e alcuni altri dispositivi video con uscita HDMI possono connettersi a schermi con ingresso CVBS. L'ingresso HDMI supporta una risoluzione fino a 1920x1080 a 60 Hz.

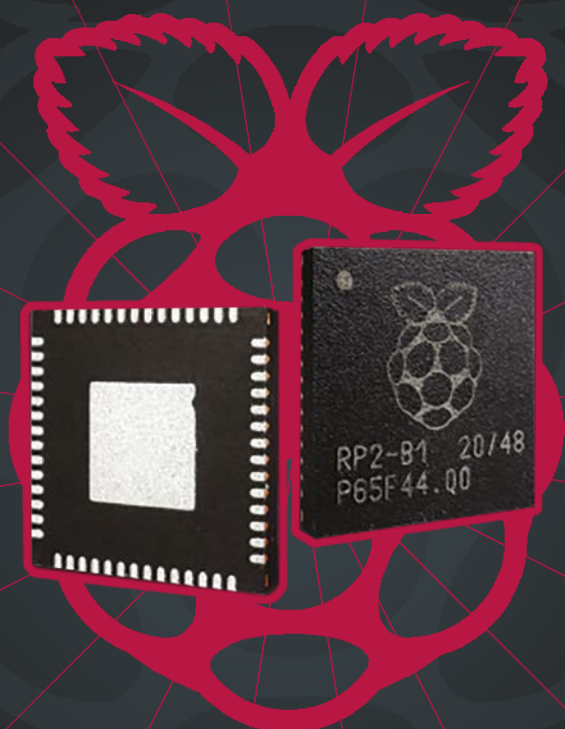
HDMI
↓
AV

€ 11,00

Cod. **BOXHDMITOAV**

Programmare con RP2040

di PIER CALDERAN



PUNTATA N. 6

In questa puntata spieghiamo come usare la nostra scheda FTR2040 come controller per LED NeoPixel.

In questo corso abbiamo illustrato le molte potenzialità della scheda FTR2040, ovvero del processore Raspberry Pi RP2040 e di tutti i modelli compatibili. Potremmo proseguire all'infinito, ma come per ogni cosa, anche questo corso deve terminare e vogliamo lasciarvi con un progetto facile e divertente.

LED NEOPIXEL

Nel kit di Futura Elettronica per Raspberry Pi Pico, fra le molte cose disponibili, c'è anche un anello con 12 LED RGB con driver integrato WS2812B. Per semplicità di spiegazione, d'ora in poi chiameremo questo dispositivo con il termine NeoPixel, amichevolmente chiamato così da Adafruit (Fig. 1). Si fa notare che questo progetto può essere applicato anche a strisce di LED NeoPixel indirizzabili con lo stesso driver. Fate attenzione! Per alimentare diret-

tamente dalla scheda un anello a 12 o 24 LED non ci sono problemi, ma per strisce di LED molto lunghe dovete procurarvi un'alimentazione esterna. Per fare funzionare l'anello o la striscia NeoPixel, dovete saldare sul retro dell'anello tre dei quattro terminali:

- GND: massa
- 5 o 3.3 volt: alimentazione positiva
- DI: ingresso digitale

Se volete collegare in cascata più anelli (o più strisce) di LED, dovete saldare il terminale DO (digital output) al corrispondente terminale DI del successivo anello (o striscia).

CABLAGGIO DELL'ANELLO NEOPIXEL

Il primo passo è quello di collegare l'anello NeoPixel alla scheda FTR2040, come mostrato in Fig. 2.



Fig. 1 - L'anello NeoPixel a 12 LED.

Si ricorda che la scheda può alimentare un numero limitato di LED NeoPixel a causa della corrente ridotta che può essere assorbita dal pin 5V sulla scheda.
Gli esempi seguenti presuppongono l'uso di un anello o striscia NeoPixel con al massimo 24 LED.

Dopo aver saldato i tre terminali all'anello, inserite nella breadboard i cavetti in questo modo:

SCHEDA FTR2040	NEOPIXEL
GND	GND
VBUS	3.3V
GP28	DI

PROGRAMMAZIONE DEI LED NEOPIXEL

Prima di poter programmare l'anello, dovete installare la libreria NeoPixel nella scheda FTR2040. Fra le tante librerie disponibili abbiamo scelto quella che ci sembra più facile da usare rispetto a quella fornita da Adafruit.
Il repository GitHub della libreria NeoPixel è il seguente:
https://github.com/blaz-r/pi_pico_neopixel
Una volta clonato il repository sul vostro Raspberry Pi, troverete all'interno della cartella la libreria "neopixel.py".
Aprirete il file con Thonny e, con il comando salva con nome, salvatelo all'interno della scheda (Fig. 3).
All'interno della cartella examples, troverete vari esempi.

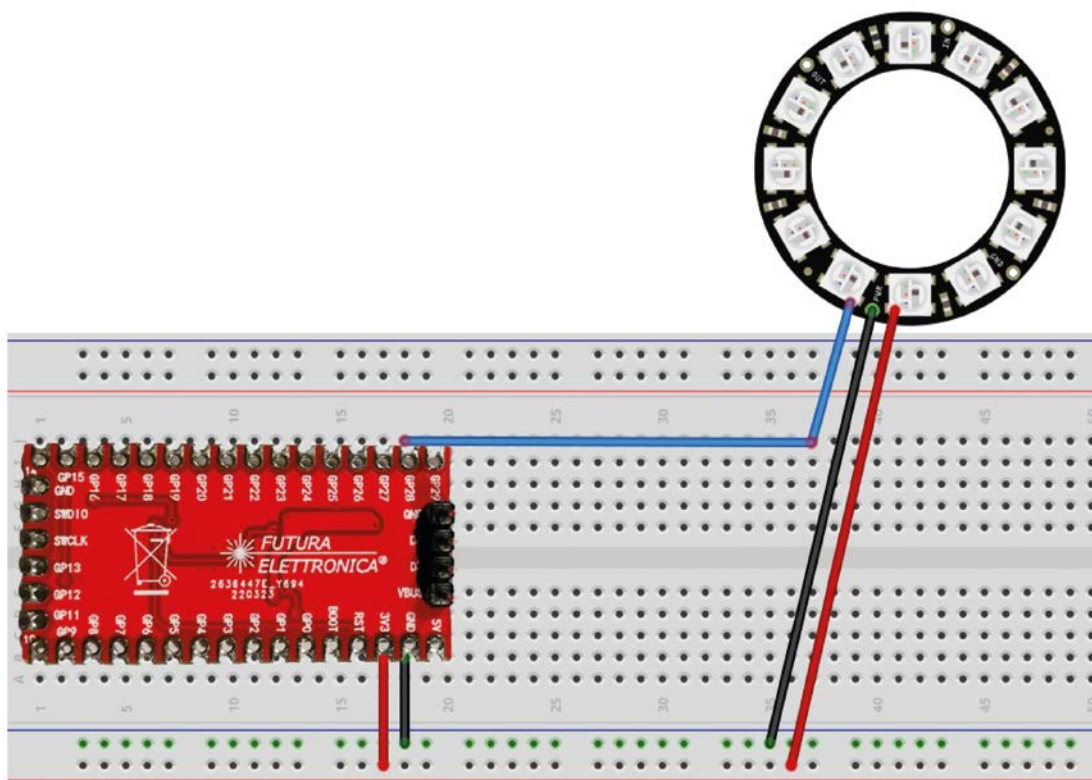


Fig. 2 - Il cablaggio dell'anello NeoPixel e la scheda FTR2040.

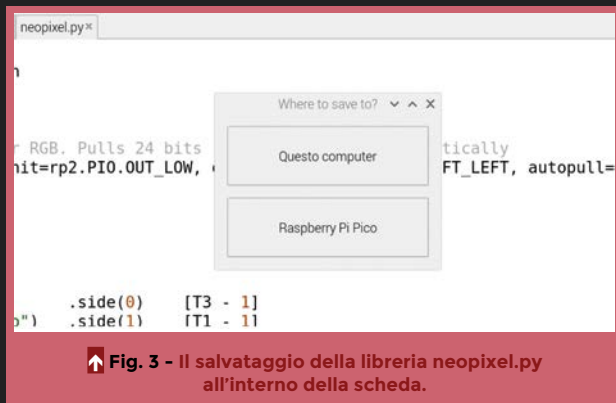


Fig. 3 - Il salvataggio della libreria neopixel.py all'interno della scheda.

Per vedere se tutto funziona, aprite il file di esempio set_range.py e modificate le righe seguenti, come evidenziato nel **Listato 1**:

```
strip = Neopixel(numpix, 0, 0, "GRB")
numpix = 60 # numero di LED
time.sleep (5.0) # tempo di attesa con tutti i LED accesi
time.sleep (0.5) # tempo della velocità di rotazione dei LED
```

... in questo modo:

```
strip = Neopixel(numpix, 0, 24, "RGB")
numpix = 12
time.sleep (1.0)
time.sleep (0.1)
```

In pratica, abbiamo impostato a 12 il numero di LED dell'anello e ridotto il tempo di attesa per i LED tutti accesi e il tempo di rotazione dei LED. Lasciate tutto il resto inalterato. Analizzando le parti salienti dell'esempio, potete notare che all'inizio viene importata la libreria neopixel, ovvero quella che avete salvato nella scheda con l'operazione descritta prima.

```
from neopixel import Neopixel
```

Quindi, la variabile numpix = 12 imposta il numero di LED dell'anello e va commisurata al numero di LED dell'anello o della striscia NeoPixel. Subito dopo l'oggetto strip viene impostato dalla classe Neopixel con i seguenti parametri:

```
strip = Neopixel (numpix, 0, 24, "RGB")
```

In pratica, l'oggetto strip viene impostato a 12 LED con il collegamento al pin GP24 della scheda. Il tipo di colorazione è RGB, ma potrebbe anche essere RGBW, se esiste anche il canale White. Notare che si possono ordinare i colori con varie combinazioni, per esempio, GRB, BRG ecc. a seconda di come si vogliono accendere nell'ordine i LED in rosso, in verde o blu. La variabile K = 3 imposta la posizione dei tre colori RGB che

vengono stabiliti dalle variabili seguenti:

```
red = (255, 0, 0)
green = (0, 255, 0)
blue = (0, 0, 255)
```

L'istruzione seguente imposta il primo valore di K sul rosso, il successivo valore di K sul verde, il successivo valore di K sul blu e il resto a R, G, B, R, G, B ecc...

$K = \min(K, (\text{numpix}-1) // 3)$ # notare il doppio simbolo di divisione per ottenere un risultato intero e non float

Per impostare la luminosità si usa l'istruzione seguente con valori da 1 a 255:

```
strip.brightness (255)
```

Quindi si impostano i colori di partenza:

```
strip[:] = blue # tutto blu
```

poi vengono alternati i colori red e green con delle operazioni sulla costante K.

Listato 1

```
import time
from neopixel import Neopixel

numpix = 12
K = 3

strip = Neopixel(numpix, 0, 28, "GRB")
red = (255, 0, 0)
green = (0, 255, 0)
blue = (0, 0, 255)

# imposta la prima K sul rosso, la successiva K sul verde,
# la successiva K sul blu;# e il resto in R,G,B,R,B ..
# e poi inizia la rotazione .

# riduce K, se numpix è < K*3+1K = min(K,(numpix-1)//3)

strip.brightness(80)

strip[:] = blue # prima tutto in blu...
# ora riempie il rosso e il verde...
strip[K:2*K] = red
strip[K*2*K] = green
strip[3*K::3] = red
strip[3*K+1::3] = green

strip.show()

# appaiono i LED per 2 secondi...
time.sleep(2.0)

# rotazione...

while(True):
    strip.rotate_right()
    strip.show()
    time.sleep(0.1)
```


↓ Listato 2

```
from machine import ADC, Pin
from time import sleep

photoPIN = 27

def readLight(photoGP):
    photoRes = ADC(Pin(27))
    light = photoRes.read_u16()
    light = round(light/65535*100,2)
    return light

while True:
    print ("luminosità: " + str(readLight(photoPIN)) + "%")
    sleep (1)
```

```
strip[:K] = red
strip[K:2*K] = green
strip[3*K::3] = red
strip[3*K+1::3] = green
```

Il comando seguente fa apparire il tutto:

```
strip.show ()
```

... viene introdotta un'attesa di 2 secondi:

```
time.sleep (2.0)
```

... e poi inizia la rotazione a destra di tutti i LED con un tempo di 0.1 secondi fra le accensioni. In questo modo si ottiene una rotazione abbastanza gradevole. Per cambiare la velocità di rotazione, basta variare il tempo di `time.sleep`.

```
While (True):
    strip.rotate_right()
    strip.show()
    time.sleep(0.1)
```

I METODI DELLA LIBRERIA NEOPIXEL

Per vostra comodità riportiamo alcuni metodi della libreria neopixel da usare per svariati effetti. Tutti gli esempi seguenti sono riportati nel nostro codice "esempi_neopixel.py".

brightness(brightness=None)

Parametro luminosità: [predefinito: Nessuno]

Valore di luminosità nell'intervallo da 1 a 255.

Esempio:

```
strip.brightness (42) #imposta la luminosità a 42
```

set_pixel_line_gradient(pixel1, pixel2, left_rgb_w, right_rgb_w, how_bright=None)

Parametro pixel1: indice del pixel iniziale (incluso).

Parametro pixel2: indice del pixel finale (incluso).

Parametro left_rgb_w: tupla di forma (r, g, b) o (r, g, b, w) che

rappresenta il colore iniziale.

Parametro right_rgb_w: tupla di forma (r, g, b) o (r, g, b, w)

che rappresenta il colore finale.

Parametro how_bright: [predefinito: Nessuno] Imposta la luminosità dell'intervallo corrente. Se how_bright è None, viene usato il valore di luminosità globale.

Il metodo crea un gradiente di colore, ovvero una transizione di colore da left_rgb_w a right_rgb_w tra i led con indice pixel1 e indice pixel2 (inclusi).

Esempio per un anello di 12 LED RGB:

```
rgbw1 = (0, 0, 50)
```

```
rgbw2 = (50, 0, 0)
```

```
strip.set_pixel_line_gradient(0, 11, rgbw1, rgbw2) # crea un
gradiente da blu a verde per tutti i LED da 0 a 11
```

set_pixel_line(pixel1, pixel2, rgb_w, how_bright = None)

Parametro pixel1: indice del pixel iniziale (incluso).

Parametro pixel2: indice del pixel finale (incluso).

Parametro rgb_w: tupla di forma (r, g, b) o (r, g, b, w) che rappresenta il colore da utilizzare.

Parametro how_bright: [predefinito: Nessuno] La luminosità dell'intervallo corrente. Se how_bright è None, viene usato il valore di luminosità globale.

Il metodo riempie i led dal led con l'indice pixel1 all'indice pixel2 (inclusi) con il colore rgb_w.

Esempio:

```
rgb = (0, 50, 0)
```

```
strip.set_pixel_line(0, 5, rgb) #accende 6 LED in rosso da 0 a 5.
```

```
rgb = (0, 0, 50)
```

```
strip.set_pixel_line(6, 8, rgb) #accende 3 LED in blu da 7 a 9.
```

set_pixel(pixel_num, rgb_w, how_bright=None)

Parametro pixel_num: indice del pixel da impostare o oggetto slice che rappresenta più led.

Parametro rgb_w: tupla di forma (r, g, b) o (r, g, b, w) che rappresenta il colore da utilizzare.

Parametro how_bright: [predefinito: Nessuno] La luminosità dell'intervallo corrente. Se how_bright è None, viene usato il valore di luminosità globale.

Il metodo imposta il pixel sull'indice pixel_num sul colore rgb_w. Se pixel_num è un oggetto slice, l'operazione viene applicata a tutti i pixel inclusi nella slice.

Esempio:

```
strip.set_pixel(5, (0, 0, 92)) # accende il LED numero 6 di blu
con intensità 92
```

setitem(idx, rgb_w)

Parametro idx: l'indice può essere un numero di indicizzazione o uno slice (vedere sotto).

Parametro rgb_w: tupla di forma (r, g, b) o (r, g, b, w) che rappresenta il colore da utilizzare.



Imposta il pixel o i pixel rappresentati da `idx` al colore `rgb_w`.
Esempi con anello da 12 LED:

```
strip[2] = (0,255,0)      # imposta il LED 2 in verde
strip[3:6] = (255,0,0)    # imposta i LED da 3 fino a 6 in rosso
strip[7:9:11] = (0,0,255) # imposta i LED 7, 9, 11 in blu
strip[1::2] = (0,0,0)    # imposta tutti i LED dispari su 'off'
```

rotate_left(num_of_pixels=None)

Parametro: [predefinito: Nessuno] Imposta il numero di pixel da spostare a sinistra. Se `num_of_pixels` è `None`, si sposta di 1. Attenzione! Dovete mettere il metodo all'interno di un ciclo `While`.

Il metodo sposta tutti i pixel in modo circolare a sinistra con un passo definito da `num_of_pixels`.

Esempio:

```
while True:
    strip.rotate_left(1) # [1, 2, 3] diventa [2, 3, 1]
    time.sleep (.1)
```

rotate_right(num_of_pixels=None)

Parametro: [predefinito: Nessuno] Imposta il numero di pixel da spostare a destra. Se `num_of_pixels` è `None`, si sposta di 1. Attenzione! Dovete mettere il metodo all'interno di un ciclo `while`.

Il metodo sposta tutti i pixel in modo circolare a destra con un passo definito da `num_of_pixels`.

Esempio:

```
while True:
    strip.rotate_right(1) # [1, 2, 3] diventa [2, 3, 1]
    time.sleep (.1)
```

show()

Il metodo invia i dati in uscita, facendo in modo che tutte le modifiche sui LED abbiano effetto. Questo metodo dovrebbe essere utilizzato dopo ogni metodo che modifica lo stato dei LED o dopo una serie di modifiche.

Esempio:

```
strip.rotate_right(1)
strip.set_pixel(0, (0, 50, 0))
strip.show()
```

clear()

Cancella l'intera striscia di LED, cioè imposta ogni colore su 0.

Esempio :

```
strip.clear()
```

Potete trovare ulteriori informazioni su tutti i metodi della libreria `neopixel` a questo link:

https://github.com/blaz-r/pi_pico_neopixel/wiki/Library-methods-documentation

CONTROLLARE LA LUCE CON... LA LUCE

Abbiamo pensato di arricchire il progetto NeoPixel con qualcosa di utile e divertente, ovvero controllare il numero di LED

da accendere attraverso l'uso di un sensore di luce. Per prima cosa procuratevi un fotoresistore, comunemente chiamato LDR (Light Dependent Resistor), e collegatelo alla scheda come illustrato in Fig. 4. Usate un resistore da 10 kΩ come pulldown per l'ingresso GP27 della scheda.

Aprirete il codice di esempio `ldr_test.py` e vedrete quanto è semplice impostare la lettura del fotoresistore su un pin con caratteristiche ADC. Ricordiamo che i pin ADC dell'RP2040 sono GP26 (ADC0), GP27 (ADC1) e GP28 (ADC2).

Tenendo d'occhio il Listato 2, potete vedere come funziona la lettura del pin ADC della scheda. Tutto è demandato alla seguente istruzione:

```
photoRes = ADC(Pin(27))
```

Quindi, basta impostare la lettura a 16 bit:

```
light = photoRes.read_u16()
```

... e applicare l'operazione seguente per ottenere una percentuale di luminosità, con due decimali di precisione:

```
light = round (light/65535 * 100 , 2)
```

Nel ciclo `while` si stamperà il valore in stringa della percentuale della luminosità a intervalli di un secondo:

```
while True:
    print ("luminosità: " + str (readLight(photoPIN)) + "%")
    sleep(1)
```

Il risultato è visibile in Fig. 5.

A questo punto non ci resta che collegare l'anello NeoPixel e il fotoresistore alla scheda. In questo modo creerete una specie di "monitor ottico" della percentuale di luminosità oppure un semplice gadget di divertimento (Fig. 6).

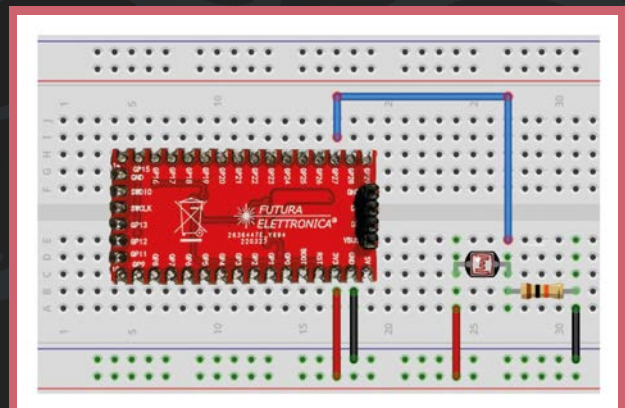


Fig. 4 - L'LTR collegato alla scheda.

```

File Modifica Mostra Esegui Strumenti Aiuto
+ [Icons]
ldr_test.py* set_range.py* ldr_neopixel.py* neopixel.py*
1 from machine import ADC, Pin
2 from time import sleep
3
4 photoPIN = 28
5
6 def readLight(photoGP):
7     photoRes = ADC(Pin(28))
8     light = photoRes.read_u16()
9     light = round(light/65535*100,2)
10    return light
11
12 while True:
13     print("light: " + str(readLight(photoPIN)) + "%")
14     sleep(1)

```

```

Shell x
light: 78.24%
light: 77.51%
light: 76.63%
light: 76.48%
light: 77.49%
light: 78.88%
light: 79.02%
light: 78.93%
light: 79.02%

```

Fig. 5 - I dati di luminosità in percentuale.

Per fare questo, dovete lasciare intatti i due circuiti sulla breadboard e caricare lo script `ldr_neopixel.py`, che riportiamo nel [Listato 3](#).

In pratica, abbiamo unito i due script precedenti con l'unica differenza sostanziale nel ciclo while, in cui abbiamo aggiunto alcune istruzioni. Il valore della variabile K, invece di essere un valore fisso, viene determinato dalla lettura del pin ADC che, come detto poc'anzi è il GP27.

```
K = readLight (photoPIN)
```

Quindi, avendo un anello di 12 Led abbiamo imposto un valore minimo e un valore massimo della variabile K, che non deve essere maggiore di 12 o inferiore a 1.

```
if K > 11: K = 12
```

```
if K < 1: K = 1
```

Nel primo ciclo for abbiamo usato il metodo `strip.set_pixel` per accendere tutti i LED in blu.

```
for x in range (12):
```

```
    strip.set_pixel (x, blue)
```

Nel secondo ciclo for abbiamo usato il metodo `strip.set_pixel` per accendere i LED in rosso in base al valore determinato dal-

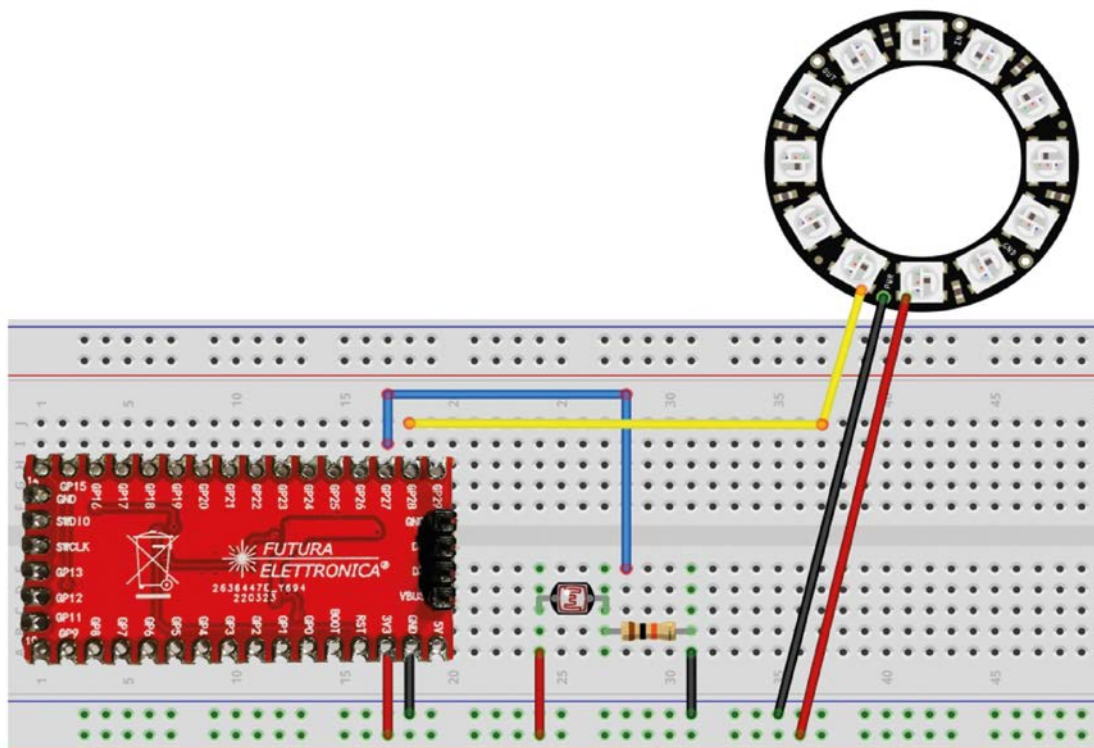


Fig. 6 - L'anello NeoPixel e il fotoresistore collegati alla scheda.



Listato 3

```
from neopixel import Neopixel
from machine import ADC, Pin
import time

photoPIN = 27
numpix = 12
K = 1

strip = Neopixel(numpix, 0, 28, "GRB")
red = (255, 0, 0)
blue = (0, 0, 255)

strip.brightness(255)

strip[:] = blue
strip[:K] = red

strip.show()

time.sleep(2.0)

def readLight(photoGP):
    photoRes = ADC(Pin(27))
    light = photoRes.read_u16()
    light = int(round(light/65535*10,0))
    return light

while True:
    print("light: " + str(readLight(photoPIN)))
    K=readLight(photoPIN)
    if K > 11: K = 12
    if K < 1: K = 1

    for x in range(12):
        strip.set_pixel(x, blue)
    for x in range(K):
        strip.set_pixel(x, red)
        time.sleep(0.1)
    strip.show()
    time.sleep(0.3)
```



Fig. 7 - Il valore arrotondato da 1 a 12 della percentuale.



Fig. 8 - I led accesi proporzionalmente alla luce.

la variabile K, ovvero del valore di luminosità letto dal pin ADC. for x in range(K):

```
strip.set_pixel(x, red)
time.sleep(0.1)
```

Abbiamo effettuato anche una piccola modifica alla percentuale che deve essere arrotondata da 1 a 12, cioè come il numero di LED, dividendo il valore per 10, anziché 100 (Fig. 7).

```
light = int(round(light/65535*10,0))
```

Se non avete commesso errori, vedrete accendersi in rosso i LED in modo proporzionale alla luce raccolta dal fotoreistore, come illustrato in Fig. 8.

CONCLUSIONI

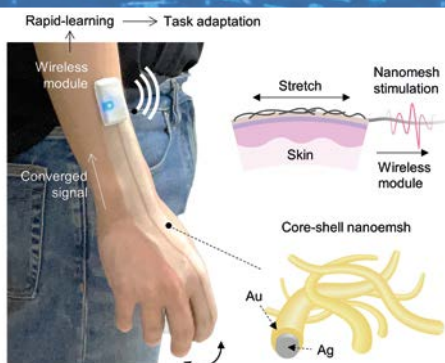
Con questa puntata si conclude questo corso e, nella speranza che vi sia stato utile, vi diamo appuntamento su queste pagine per future sperimentazioni. □

Cosa occorre?

La board di Futura Elettronica con RP2040 (cod. FTR2040) è disponibile a Euro 13,90 e include i pin strip da saldare, lo starter kit per Raspberry Pi Pico (cod. RPICOKIT) è in vendita Euro 56,00. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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



La pelle intelligente

In un articolo pubblicato sulla rivista *Nature Electronics*, i ricercatori della Stanford University descrivono un nuovo tipo di materiale biocompatibile elastico che viene spruzzato sul dorso della mano come uno spray abbronzante.

Questa pellicola rileva come la pelle si allunga e si piega e, utilizzando l'intelligenza artificiale, i ricercatori promettono di poter interpretare una miriade di attività quotidiane dai movimenti e dai gesti delle mani. La rete è composta da milioni di nanofili d'argento rivestiti d'oro che sono in contatto tra loro per formare percorsi elettrici dinamici. Questa rete è elettricamente attiva, biocompatibile, traspirante e rimane intatta a meno che non venga strofinata con acqua e sapone.

È sufficiente poi utilizzare un modulo Bluetooth per trasferire in modalità wireless le modifiche del segnale.

Finora la misurazione dell'attività elettrica muscolare è stata realizzata utilizzando cinturini da polso o guanti indossabili.

Tuttavia, questi dispositivi sono ingombranti e sono necessari più componenti sensoriali per individuare i movimenti di ogni singola articolazione. Inoltre, è necessario raccogliere una grande quantità di dati per ciascun utente e attività al fine di addestrare l'algoritmo.

La pelle intelligente è il primo approccio pratico sufficientemente snello nella forma e abbastanza adattabile nella funzione, da funzionare praticamente per qualsiasi utente, anche con dati limitati.

I ricercatori hanno costruito un prototipo che riconosce oggetti semplici al tatto e può persino eseguire la digitazione predittiva a due mani su una tastiera invisibile.

Tale precisione potrebbe essere fondamentale nelle applicazioni di realtà virtuale per trasmettere movimenti finemente dettagliati per un'esperienza più realistica.

www.nature.com

Vetroresina con nanotubi

Un sensore con nanostruttura di carbonio, il cui sviluppo e i cui test hanno coinvolto i ricercatori di Skoltech, si è mostrato promettente per migliorare le parti costruite con materiali compositi polimerici rinforzati con fibra di vetro come ad esempio ali di aerei, pale di turbine eoliche, campate di ponti, carrozzerie di automobili, scafi di barche, ecc. Realizzato in nanotubi di carbonio, genera calore se sottoposto ad un flusso di corrente elettrica. Può fungere sia da sensore per il monitoraggio della struttura o da antighiaccio. Quest'ultima peculiarità lo rende importante per la cantieristica navale e la piccola aviazione, dove vengono utilizzate spesso le strutture in vetroresina. "I risultati della nostra ricerca sono notevoli per diversi mo-

tivi", ha commentato Sergey Abaimov di Skoltech. "Per prima cosa, il composito è dotato di funzionalità extra: applicando una corrente elettrica allo strato di nanotubi di carbonio e monitorando il segnale, è possibile tenere traccia del deterioramento della struttura ed evitare il disastro anticipando la rottura. L'elettricità può essere utilizzata anche per il riscaldamento, ad esempio per sbrinare un'ala di un aereo o una pala di una turbina eolica. Inoltre il riscaldamento può essere utilizzato anche per produrre questo materiale "autoindurendo" il composito polimerico senza utilizzare un'autoclave, che è costosa e richiede molta energia".

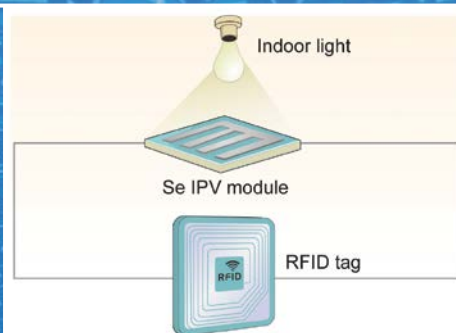
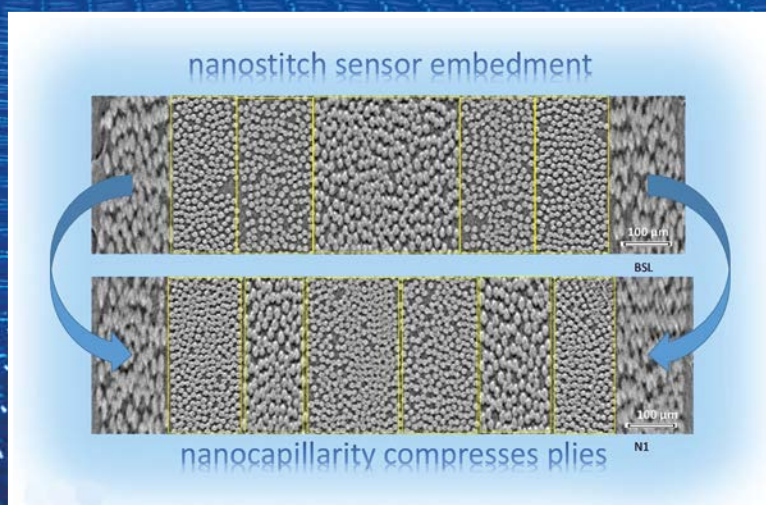
Per produrre un composito di alta qualità, l'intera parte composita - che



potrebbe essere enorme, ad esempio, nell'ingegneria aeronautica - viene polimerizzata in una camera ad alta temperatura e ad alta pressione chiamata autoclave. Solo le grandi aziende possono permettersi un'autoclave abbastanza capiente per produrre pezzi di grandi dimensioni con questa tecnologia. Inoltre viene consumata molta energia per riscaldare ogni volta l'intero volume della camera, indipendentemente dall'effettiva dimensione e forma della parte composta da indurire. "Se si incorporano nanotubi di carbonio tra gli strati adiacenti del composito, è possibile applicare una corrente elettrica e fornire calore in modo molto efficiente nel materiale, senza sprecare energia come si farebbe per riscaldare l'intero volume

dell'autoclave", ha detto Abaimov. Lo studio ha anche dimostrato come l'inserimento di nanotubi non provoca alterazione del materiale, ci si aspetterebbe infatti uno spessore maggiore e ulteriori difetti sotto forma di pori o orientamento delle fibre distorto. Ma questa integrazione non modifica lo spessore e non aggiunge alcuna porosità, infine le proprietà meccaniche sono migliorate, non compromesse. Per confrontare le caratteristiche dei materiali, il team ha realizzato il composito polimerico sia senza la struttura di nanotubi di carbonio che con essa. I ricercatori hanno spiegato questo in termini di nanocapillarità estremamente elevata dei nanotubi.

www.mdpi.com



Il fotovoltaico da interno

I primi moduli fotovoltaici allo stato solido al mondo furono creati nel 1883 ed erano composti da selenio. Le prime celle solari al selenio (Se) avevano una larghezza di banda limitata (~1,9 eV) e non ottimale per l'utilizzo con la luce solare.

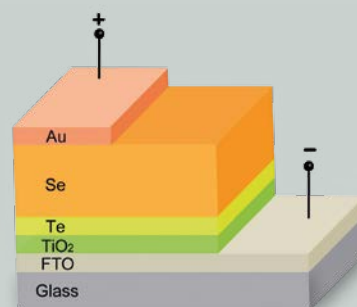
In un recente lavoro, pubblicato su Science Advances, Bin Yan e un team di ricercatori in chimica, nanotecnologia e scienza dei materiali in Cina, hanno rivisto l'utilizzo delle vecchie celle al selenio per utilizzarle nelle applicazioni fotovoltaiche indoor.

Lo spettro di assorbimento del materiale corrispondeva infatti perfettamente alle emissioni delle sorgenti luminose per interni comunemente utilizzate (400 ÷ 700 nm).

Dagli esperimenti la nuova composizione di celle fotovoltaiche per interno mostrano un'efficienza del 15,1% con un'illuminazione di 1000 lux e non mostrano alcuna perdita di efficienza dopo 1000 ore di illuminazione continua, superando l'attuale standard industriale IPV del silicio amorfo in termini di efficienza e stabilità.

I ricercatori hanno utilizzato moduli di selenio per produrre una potenza di uscita di 232,6 µW sotto illuminazione interna per alimentare un tag di localizzazione basato sull'identificazione a radiofrequenza.

www.science.org



La filosofia di ChatGP

ChatGPT (Generative Pre-trained Transformer) è un chatbot lanciato da OpenAI (tra i fondatori Elon Musk) nel novembre 2022. È basato sulla famiglia GPT-3.5 di OpenAI per modelli linguistici di grandi dimensioni ed è ottimizzato con tecniche di apprendimento con supervisione e rinforzo. In un documento, il Dr. Alex Zhavoronkov di Insilico Medicine ha utilizzato ChatGPT per discutere l'uso della rapamicina come farmaco per l'anti-invecchiamento attraverso il quadro filosofico della scommessa di Pascal. La scommessa di Pascal è un'argomentazione proposta da Blaise Pascal (1623-1662) sull'utilità di credere o meno in Dio. Il ragionamento di Pascal conduce alla conclusione che conviene credere in Dio. Infatti se Dio esiste, si ottiene la salvezza; se Dio non esiste, si è vissuto un'esistenza lieta pur nella consapevolezza di finire in polvere. "In questo articolo, collaboriamo con ChatGPT, un modello di intelligenza artificiale sviluppato da OpenAI per speculare sulle applicazioni

della rapamicina, nel contesto dell'argomentazione filosofica della scommessa di Pascal comunemente utilizzata per giustificare la fede in Dio", ha affermato il dott. Alex Zhavoronkov.

In risposta alla domanda "Scrivi una prospettiva di ricerca esaustiva sul motivo per cui l'assunzione di Rapamicina potrebbe essere più vantaggiosa rispetto a non assumerla dal punto di vista della scommessa di Pascal", ChatGPT ha fornito i pro e i contro dell'uso di Rapamicina considerando l'evidenza preclinica di una potenziale estensione di vita negli animali. Questo articolo dimostra il potenziale di ChatGPT per produrre argomen-

tazioni filosofiche complesse.

"In conclusione, la decisione di assumere rapamicina o meno è in ultima analisi personale e dipenderà dai valori e dalle priorità di un individuo. Dal punto di vista della scommessa di Pascal, l'assunzione di rapamicina potrebbe potenzialmente fornire significativi benefici per la salute e prolungare la durata della vita, ma potrebbe anche comportare effetti collaterali e problemi a lungo termine. In definitiva, la decisione dovrebbe essere presa in consultazione con un operatore sanitario, che può fornire indicazioni e consigli in base alla situazione specifica di un individuo", scrive ChatGPT.

"La capacità dei grandi modelli linguistici e di altri sistemi di intelligenza artificiale di dare contributi significativi al lavoro accademico può giustificare la futura copaternità su prospettive accademiche, revisioni e documenti di ricerca", osserva Zhavoronkov.

www.oncoscience.us



Vedere attraverso i muri con il WiFi

Un team di ricercatori dell'Università di Waterloo ha creato un dispositivo in grado di vedere attraverso i muri utilizzando le reti WiFi. Il dispositivo, denominato Wi-Peep, viene posizionato su un piccolo drone che può volare vicino a un edificio e quindi monitorare la rete WiFi per identificare e localizzare rapidamente qualsiasi dispositivo radio all'interno. Il Wi-Peep sfrutta una scappatoia che i ricercatori chiamano Polite WiFi: anche se una rete è protetta da password, i dispositivi risponderanno automaticamente ai tentativi di contatto da qualsiasi dispositivo nel raggio d'azione. Il Wi-Peep invia diversi messaggi a un dispositivo mentre vola e quindi misura il tempo di risposta di ciascuno, consentendogli di identificare la posizione del target entro un metro. Il dottor Ali Abedi, professore di informatica a Waterloo, spiega il significato di questa scoperta: "I dispositivi Wi-Peep sono come luci nello spettro visibile e le pareti sono come il vetro", ha detto Abedi. "Utilizzando una tecnologia simile, si potrebbero tracciare i movimenti delle guardie di sicurezza all'interno di una banca seguendo la po-

sizione dei loro telefoni o smartwatch. Allo stesso modo, un ladro potrebbe identificare la posizione e il tipo di dispositivi intelligenti in una casa, tra cui telecamere di sicurezza, laptop e smart TV, per trovare un sistema per entrare. Inoltre, il funzionamento del dispositivo tramite drone significa che può essere utilizzato rapidamente e da remoto senza molte possibilità che l'utente venga rilevato."

Il Wi-Peep si distingue per la sua accessibilità e facilità di trasporto. Il team di Abedi lo ha costruito utilizzando un drone acquistato in negozio e \$20 di hardware facilmente reperibile.

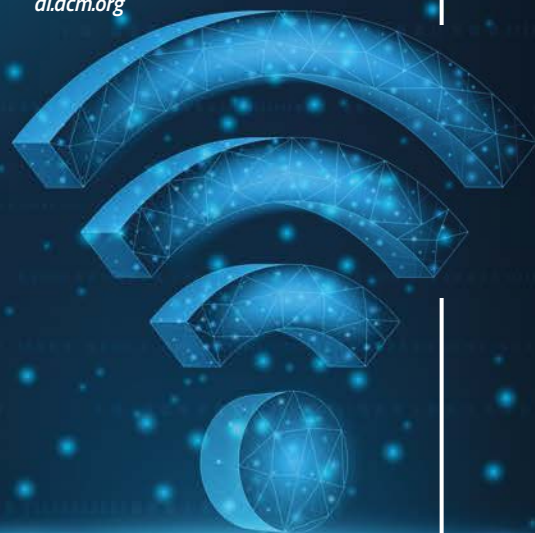
"Non appena è stata scoperta la scappatoia di Polite WiFi, ci siamo resi conto che questo tipo di attacco era possibile", ha affermato Abedi. Il team ha costruito il Wi-Peep per testare la loro teoria e si è presto reso conto che chiunque avesse la giusta esperienza poteva facilmente creare un dispositivo simile.

"E' necessario correggere il Bug del Polite WiFi in modo che i dispositivi non rispondano agli estranei", ha affermato Abedi. "Speriamo che il nostro lavoro

sia utile per la progettazione del protocollo di prossima generazione".

Nel frattempo, invita i produttori di chip WiFi a introdurre una modifica artificiale e casuale del tempo di risposta del dispositivo, che renderà i calcoli come quelli utilizzati da Wi-Peep estremamente imprecisi.

dl.acm.org



RICARICA I TUOI DISPOSITIVI OVUNQUE TI TROVI!

PANNELLO SOLARE PORTATILE DA 60W



PANNELLO SOLARE MONOCRISTALLINO DA 60W

ad alta efficienza (16,85%), pieghevole e portatile. Dispone di uscita DC max 20,9 volt 2,87A, uscita USB-A da 5V 3A, uscita USB-QC: 5V 3A / 9V 2A / 12V 1,5A. Esternamente è realizzato in nylon e dispone di due robuste maniglie con potenti magneti che assicurano una salda chiusura. Ideale per il campeggio, l'escursionismo e altre attività all'aperto. Dimensioni: 530 x 440 x 30 mm (chiuso), 880 x 530 x 10 mm (aperto), peso: 2,9 g.



cod. FOLDINGPAN60

€ 135,00



USCITA PRESA DC

(5,5x2,1 mm): max 20,9 volt – 2,87 A

USCITA USB-QC
5V 3A / 9V 2A / 12V 1,5A

USCITA USB-A
5V 3A

Prezzo IVA inclusa.

Sicurezza e affidabilità senza compromessi!

Rendi più sicura la casa, il negozio e l'ufficio con questi set professionali per videosorveglianza e registrazione.
Si installano facilmente e permettono la visione remota tramite PC e smartphone connessi a Internet.

SET COMPLETO VIA CAVO

Il set comprende un DVR ibrido a 4 ingressi che può funzionare sia in modalità HD (AHD / CVI / TVI) che in modalità analogica, completo di mouse, quattro telecamere Bullet da esterno da 2 Megapixel con illuminatori IR, due alimentatori da 12 VDC (uno per le telecamere e uno per il DVR), quattro cavi video da 14 metri e cavi di alimentazione per telecamere. L'hard disk è acquistabile separatamente.

VIA CAVO



€175,00

Cod. VIDEOCAMSET1



App
gratuite
per Android
e iOS



€269,00

Cod. VIDEOCAMSET2

SET COMPLETO WIRELESS

Set composto da un NVR Wi-Fi a 4 canali, quattro telecamere IP IR wireless, un mouse USB, cinque alimentatori 12 Vdc (quattro per le telecamere e uno per l'NVR), quattro antenne per le telecamere, due antenne per l'NVR, un cavo di rete da 1 metro. L'hard disk è acquistabile separatamente.

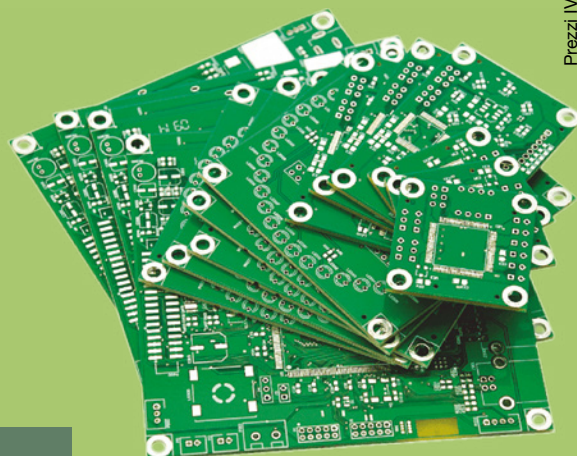
I tuoi PCB a partire da

1€

Con il servizio PCB di Futura Elettronica, potrai realizzare facilmente il circuito stampato del tuo progetto!

Vai su www.futurashop.it, inserisci nella ricerca il codice "PCBPRODUCTION" e carica il tuo Gerber scegliendo le dimensioni e le caratteristiche preferite.

Prezzi IVA inclusa.



cod. PCBPRODUCTION

COLORE SERIGRAFIA



BIANCA



NERA

La serigrafia nera è disponibile solamente per PCB bianchi e gialli.

COLORE SOLDER



VERDE



ROSSO



GIALLO



BLU



BIANCO



NERO

SPESSORE (mm)

mm



0.2

0.4

0.6

0.8

1.0

1.2

1.6

2.0

2.4

2.8

3.2

NUMERO DI LAYERS

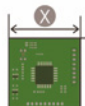
1 Layer

2 Layers

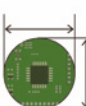
4 Layers

6 Layers

DIMENSIONI (cm)



Single Size



Pannel Size

- 5x5 - 5x10 - 5x15 - 5x20 - 5x25 - 5x30
- 10x10 - 10x15 - 10x20 - 10x25 - 10x30
- 15x15 - 15x20 - 15x25 - 15x30
- 20x20 - 20x25 - 20x30
- 25x25 - 25x30
- 30x30

ECCO QUALCHE ESEMPIO:

PCB con 1 layer
da 5x5 cm
spessore 1,6 mm
solder verde
serigrafia bianca
€ 1 cad.

PCB da 2 layer
da 10x10 cm
spessore 1,6 mm
solder verde
serigrafia bianca
€ 6 cad.

www.futurashop.it

Futura Group srl
Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775

**FUTURA
ELETTRONICA®**

Affidabilità e precisione!

Generatore di forme d'onda arbitrarie 2 Canali-25 MHz

- Canali: 2
- Lunghezza record: 2 Mpts
- Range frequenza: 25 MHz
- Frequenza di campionamento: 300 MSa/s
- Risoluzione di tensione: 16 Bits
- Uscita forma d'onda standard: sinusoidale, quadra, triangolare, pulsante, noise, armoniche
- Uscita forma d'onda arbitraria: 160 tipi di forme d'onda arbitrarie, tra cui: exponential rise, exponential fall, ECG signal, Gauss, half vector, Lorentz, dual tone multi-frequency, DC voltage, etc.
- Noise (3 db): Larghezza di banda 100 MHz
- Onda Arbitraria: 1 μ Hz ~ 15 MHz
- Risoluzione: 1 μ Hz
- Precisione: ± 1 ppm, 18~28 °C



CON
TECNOLOGIA
DDS

cod. HDG3022B

€ 388,00

Hantek



cod. JDS6600

€ 189,00

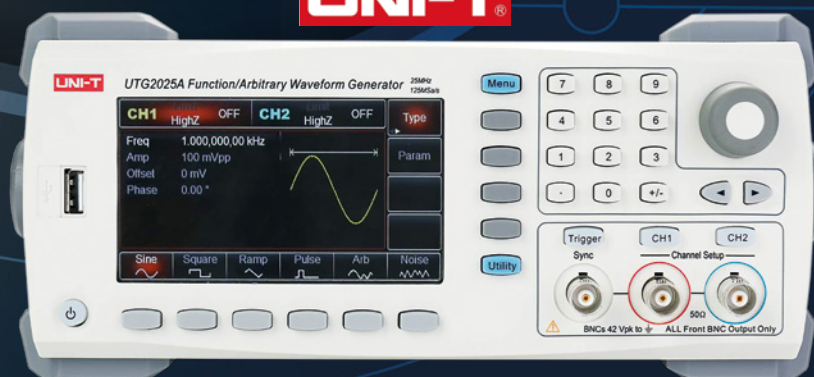
JOY-IT

Generatore di segnali con frequenzimetro 2 CH-60 MHz

- Canali: 2
- Frequenza max. di uscita: 60 MHz
- Frequenza di campionamento: 100 MSa/s
- Forme d'onda standard: sinusoidale, quadra, rampa, triangolare, impulso, noise, arbitraria, impulso di Lorenz, aumento/diminuzione esponenziale, CMOS
- Interfaccia: USB
- Display: 2.4" TFT LCD colori

Generatore di funzioni e forme d'onda arbitrarie 2 Canali-25 MHz

- Display: TFT LCD a colori da 4,3 pollici
- Frequenza di campionamento: 125 MS/s
- Forme d'onda standard: sinusoidale, quadra, rampa, triangolare, impulso, rumore gaussiano, DC, arbitrarie
- Ampiezza del segnale di uscita: fino a 5 Vpp con impedenza di 50 Ω o fino a 10 Vpp in modalità ad alta impedenza
- Impedenza di uscita regolabile
- Tipo di modulazione: AM, FM, PM, ASK, FSK, PSK, PWM
- Modalità Sweep e Burst
- Distorsione armonica totale: 20 kHz, 1Vpp (valore tipico): <0,2%
- Interfaccia: USB



cod. TP388

€ 498,00

UNI-T

Prezzi IVA inclusa.

**FUTURA
ELETTRONICA®**

www.futurashop.it

Futura Group srl
Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775

Caratteristiche tecniche di questi prodotti
e acquisti on-line su www.futurashop.it