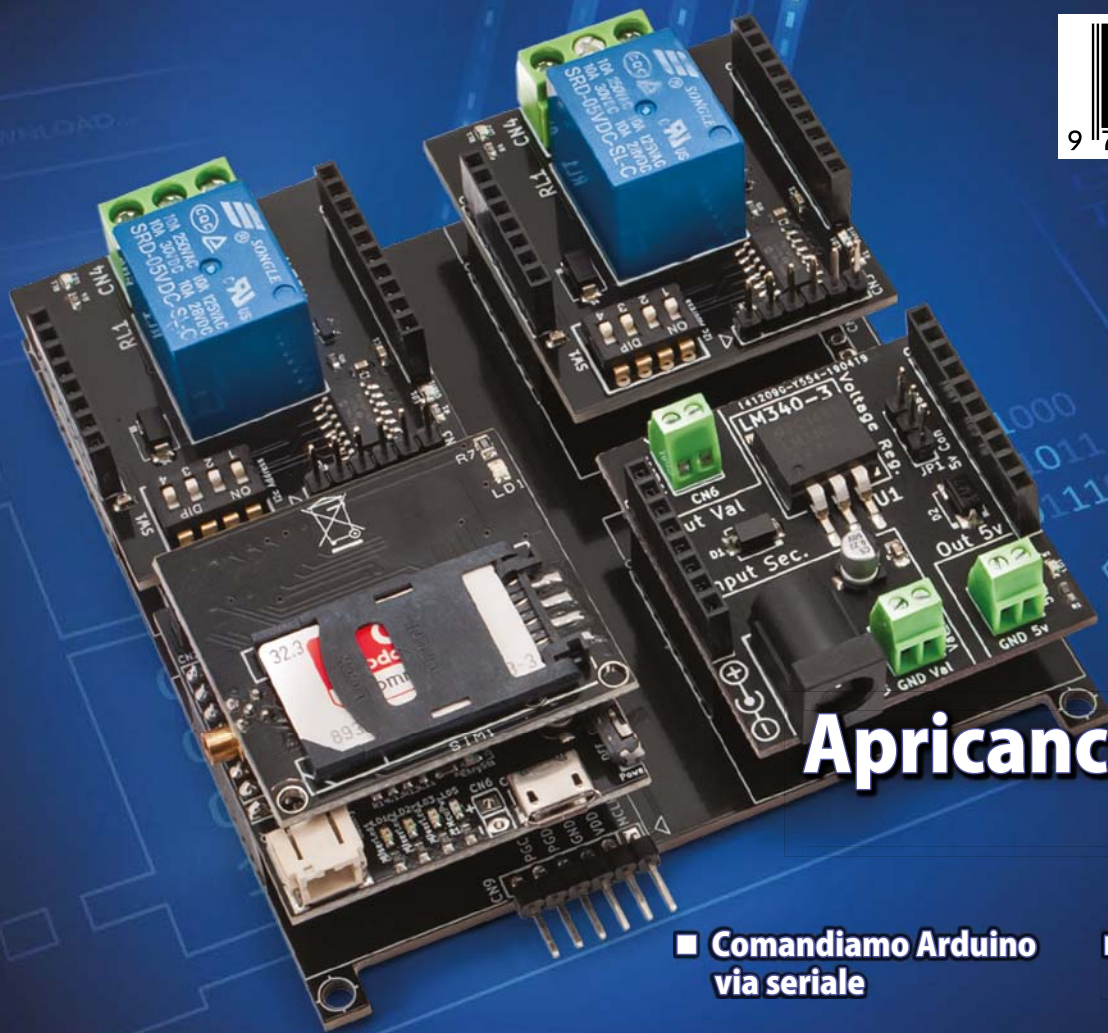


Elettronica In

www.elettronica.in.it

€ 6,90 - Anno XXVIII - n. 266 - LUGLIO / AGOSTO 2022



Apricancello GSM modulare

- Amplificatori audio
- Comandiamo Arduino via seriale
- Blebricks: le idee prendono forma
- Smart Home comfort control
- Gitea: il versioning open-source

Boost converter miniaturizzato



Interruttore battimani



È on-line il NUOVO PORTALE FUTURANET.IT

FUTURA ELETTRONICA SHOP ON LINE

SHOP

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.

FUTURA GROUP PROGETTI

PROGETTI

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

FUTURA ELETTRONICA PROGETTAZIONE

PROGETTAZIONE

Un team di esperti in grado di studiare, progettare e realizzare prototipi utilizzando le ultime tecnologie. Nel team Futura Group fa parte il personale interno e un numero elevato di collaboratori esterni, ciascuno specializzato nel proprio settore.

HOME

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

FUTURA GROUP IDEE IN ELETTRONICA

ACADEMY

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 Academy CORSI DI FORMAZIONE

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 RIVENDITORI



Batterie scariche

Il 7 giugno la Commissione Europea ha votato per l'unificazione dei caricabatterie degli smartphone venduti in UE, quindi avremo un caricabatterie universale. Finalmente l'amico che avrà il telefono scarico potrà usare il caricabatterie del nostro! Ma forse questa l'abbiamo già sentita anni fa, quando si decise di standardizzare il connettore nel microUSB e sappiamo come finì: Apple, ad esempio, disse no ed altri (Samsung, Huawei) pur adottando un connettore standard, ricorsero ad artifici per consentire la ricarica rapida solo con i loro caricabatterie. Sicuramente è bello e utile avere un unico

Raspberry Pi Foundation, causa la carenza di chip, ha scelto di fornire prima le aziende

caricabatterie ed anche svegliarsi nel 2035 scoprendo che dai listini sono scomparse le auto a motore termico. Ma le promesse sono facili; ben più difficile è trovare l'elettricità che ci serve oggi, perché non si possono recuperare in pochi mesi errori decennali nelle

politiche energetiche; neanche con l'aiuto degli USA, che per una volta esporteranno idrocarburi invece di democrazia... Del resto l'UE è solita teorizzare giganti dai piedi d'argilla come il Chips Act o il piano di svincolo dalle fonti fossili Russe e realizzarli a costo di pagarne le conseguenze.

In tema di chip, fa riflettere la notizia che, avendoli anch'essa col contagocce, Raspberry Pi Foundation, ha scelto di fornire prima le aziende, tradendo il principio che l'ha ispirata: dare un computer a tutti. L'intento è preservare le aziende nate intorno alla scheda del lampone, sostenendo l'economia, ma è ovvio che il maker serve a promuovere un prodotto e attirare su di esso parte del mercato, poi passa in secondo piano. Scelte diverse, mondi diversi e quello anglosassone fa sempre le più nette. Speriamo torni presto il periodo in cui non bisognerà decidere chi è più meritevole di avere chip, energia, benessere e che quel caricabatteria davvero universale che è l'ottimismo, ci dia l'energia per sorridere, anziché avvilarci, di fronte a certi avvenimenti.

ON LINE

elettronica.in

SI RINGRAZIA

Futura Elettronica
Mouser



In copertina:
La poliedrica
piattaforma
Mercury System
implementa un
versatile controllo
apricancello.

Boris Landoni
 boris.landoni@elettronica.in

Elettronica In

Rivista mensile, anno XXVIII n. 266
LUGLIO / AGOSTO 2022

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 10 numeri Euro 49,00

Annuo 10 numeri digitale Euro 39,00

Le richieste di abbonamento vanno inviate a:

FUTURA GROUP srl
via Adige 11, 21013 Gallarate (VA)

Tel. 0331-752668

oppure tramite il sito

<https://www.elettronica.in/abbonamenti/>

Distribuzione per l'Italia

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

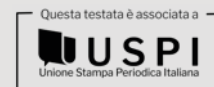
via Bettola 18 - 20092 Cinisello Balsamo (MI)

Tel. 02-660301 Fax 02-66030320

Stampa

ROTO3 Spa - Via Turbigo, 11/b
20022 CASTANO PRIMO (MI)

Tiratura
18.000 copie



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

Prezzo di copertina Euro 6,90.

Gli arretrati nei formati cartaceo e digitale (pdf) sono acquistabili
sul sito della rivista al prezzo di Euro 6,90.

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 (www.elettronica.in/ase.pdf).
Manoscritti, disegni e foto non richiesti non verranno in alcun caso
restituiti. L'utilizzo dei progetti e dei programmi pubblicati non
comporta alcuna responsabilità da parte della Società Editrice.

© 2022 Futura Group srl

Futura
Group
Edizioni

Contenuti

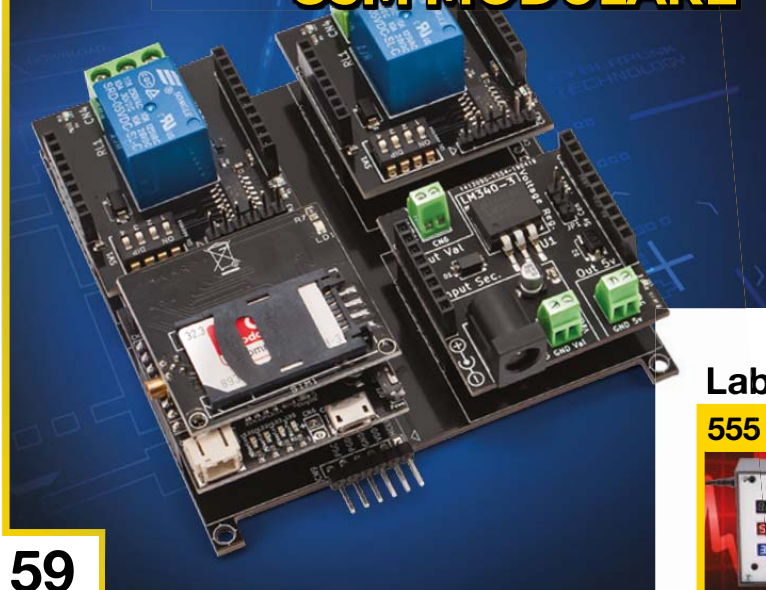
01

RUBRICHE

- 01 Editoriale
- 05 T&T Tips & Tricks
- 09 Embedded IoT
Digital Transformation of Things
- 13 Componenti & Sistemi
- 126 Scienza & Tecnologia

Progetto di Copertina

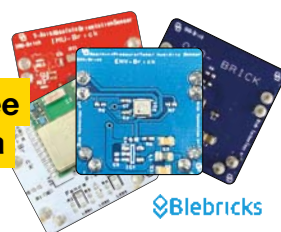
APRICANCELLO GSM MODULARE



59

Networking

Blebricks, le idee prendono forma



Blebricks

27

19

ARTICOLI

HOME PROJECT

19 Clap switch

Comandiamo luci e altri utilizzatori elettrici con un semplice battito di mani: in modalità bistabile, uno attiva, l'altro disattiva, mentre è anche possibile un comando a tempo.

NETWORKING

27 Blebricks, le idee prendono forma

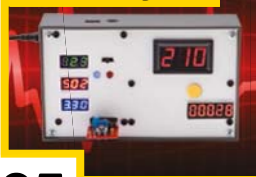
Soluzione modulare con la quale creare in tempo zero il nostro dispositivo hardware IoT programmabile, interattivo e gestibile direttamente da apposita APP.

LABORATORIO

35 555 Analyzer

Realizziamo un analizzatore di circuiti basati sul popolare timer 555 e scopriamo come utilizzarlo. Prima puntata.

Laboratorio 555 Analyzer



35

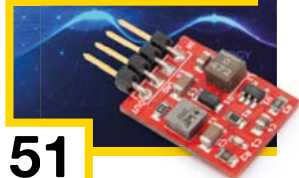
Home Project Clap switch



19

Laboratorio

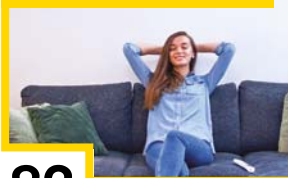
Boost converter in formato TO3-P



51

Domotica

Smart Home Comfort Control

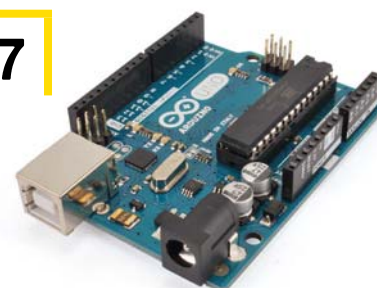


83

Arduino

Serial CMD

97



Didattica

Amplificatori Audio



71

Raspberry Pi

GITEA, il versioning open-source



103

Didattica

Programmiamo con RP2040



115

LABORATORIO

51 Boost converter in formato TO3-P

Minuscolo ed efficiente, dotato di funzione True-Shutdown, può erogare fino a 5 W con un'efficienza compresa tra il 70 e l'80%.

TOP PROJECT

59 Apricancello GSM modulare

Basato su Mercury System, può essere comandato tramite chiamate telefoniche e può essere configurato tramite un numero master impostato alla prima attivazione.

DIDATTICA

71 Amplificatori audio

Analizziamo in laboratorio le caratteristiche tecniche di moduli amplificatori commerciali fino a 2x50W. Terza e ultima puntata.

DOMOTICA

83 Smart Home Comfort Control

Aggiungiamo funzionalità al nostro progetto di domotica sviluppato con micro:bit e in questo appuntamento concentriamoci sulla gestione del comfort domestico.

ARDUINO

97 Serial CMD

Comandiamo le schede Arduino via seriale grazie ad una potente e completa libreria.

RASPBERRY PI

103 GITEA, il versioning open-source

Configuriamo una Raspberry Pi per farle ospitare un'istanza della versione self-hosted e open-source di Git, il popolare tool di versioning per software e firmware.

DIDATTICA

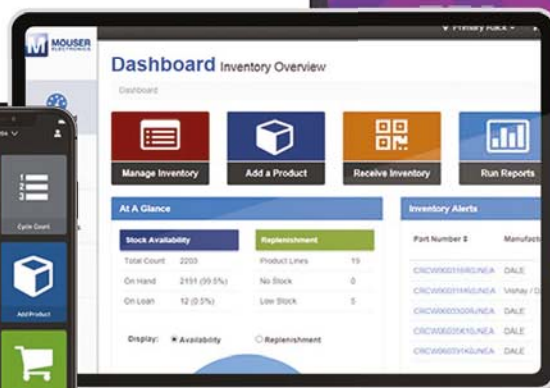
115 Programmiamo con RP2040

Conosciamo e impariamo ad usare il nuovo chip RP2040 grazie alla scheda Raspberry Pi Pico e una board custom perfettamente compatibile con l'ambiente di programmazione previsto da Raspberry Pi Foundation.

Raggiungi nuovi traguardi con il nostro sistema per la gestione dell'inventario



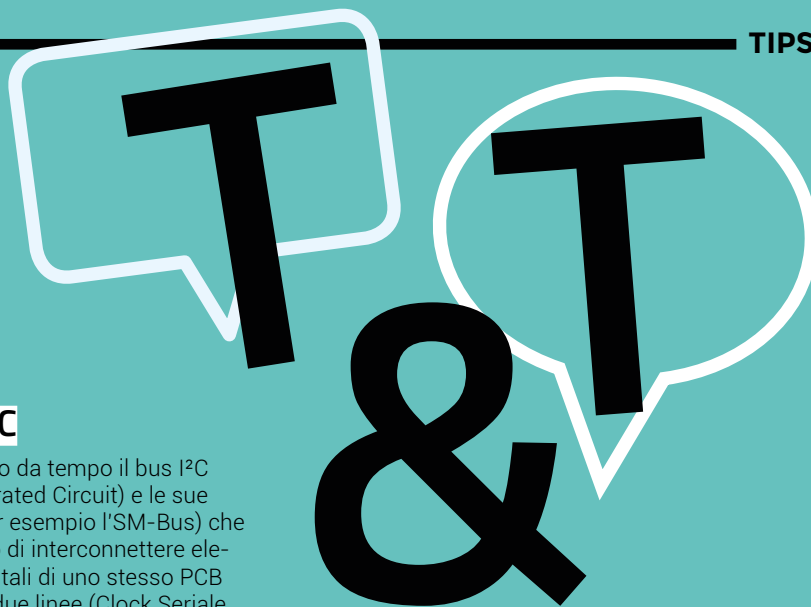
Gestione dell'inventario



mouser.it/inventory-management



**MOUSER
ELECTRONICS**



Il bus I³C

Conosciamo da tempo il bus I²C (Inter Integrated Circuit) e le sue varianti (per esempio l'SM-Bus) che permettono di interconnettere elementi circuitali di uno stesso PCB attraverso due linee (Clock Seriale e Dati) riferite a massa; da poco è apparso il bus MIPI I³C, che è un'interfaccia bus di controllo scalabile per il collegamento di periferiche al processore o ad altri controller di gestione. Offre vantaggi di prestazioni elevate, maggiore affidabilità, potenza molto bassa e bassa interferenza elettromagnetica (EMI). Tutti questi vantaggi consentono nuove funzionalità di sistema come telemetria avanzata, ripristino degli errori, sicurezza della banda laterale, autenticazione dei componenti e tempi di avvio più rapidi.

I progetti di sistema attuali utilizzano spesso il protocollo I²C legacy e semplici interruttori a MOSFET per interconnettere dispositivi di interfaccia e periferiche su una scheda, ovvero traslarne i livelli, ma questo approccio non può raggiungere velocità elevatissime: il data-rate si ferma a 400 kbit/s nella modalità fast mode e a 3,4 Mbit/s nella High Speed mode.

Il bus I³C, anche detto Sensewire, è una variante ad elevata velocità del bus I²C, che raggiunge 12,5 Mbps nella modalità SDR (Simple Data Rate), il doppio nella DDR (Double Data Rate) e 33 Mbps nella modalità Ternary. Questo bus trova impiego anche nella comunicazione con chip di memoria di ultima generazione, di tipo DDR5.

Lo standard definisce la connessione elettrica tra i chip come un bus dati seriale a due fili, condiviso (multidrop), un filo (SCL) utilizzato come clock, l'altro (SDA) utilizzato come linea dati la cui tensione può essere campionata. La specifica I³C utilizza le stesse connessioni elettriche e consente una certa compatibilità con le versioni pre-

cedenti del bus I²C, ed è progettato per mantenere una certa retro-compatibilità con il sistema I²C, in particolare consentendo progetti in cui i dispositivi I²C esistenti possono essere collegati a un bus I³C ma hanno comunque il bus in grado di passare a una velocità dati più elevata per la comunicazione a velocità più elevate. Lo standard I³C combina quindi il vantaggio della semplice architettura I²C a due fili con le velocità di comunicazione più elevate comuni a bus più complicati come l'interfaccia periferica seriale (SPI).

Lo standard I³C è stato rilasciato al pubblico per la prima volta alla fine del 2017 ed è promosso da Google e Intel come standard di interfaccia per sensori per Internet of Things. Uno dei produttori di semiconduttori più attivi nell'I³C-Bus è Renesas, che ha appena sviluppato e proposto una famiglia di switch intelligenti I³C che consente l'espansione di due porte master (a monte) verso quattro, otto o più porte periferiche alla massima velocità con piena aderenza e conformità del protocollo. I componenti forniscono una trasmissione senza interruzioni tra i dispositivi I³C e I²C consentendo la piena compatibilità plug and play dei dispositivi legacy sulla rete del

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.

bus di controllo. Lo switch I³C intelligente è il risultato di una stretta collaborazione tra i team Renesas e quelli di Intel a partire dall'ideazione, dalla definizione delle specifiche e da altre attività pre-produzione. Gli switch I³C intelligenti sono destinati all'interfacciamento su schede madri per i server e per altri apparati per le infrastrutture di comunicazione e di storage. I chip abilitano la gestione delle reti I³C locali come quelle delle CPU e dei Baseboard Management Controllers (BMC), per supportare le periferiche di una ampia rete fisica che lavora a piena velocità migliorando l'integrità del segnale e riducendo il carico capacitivo. Questa famiglia di dispositivi supporta inoltre progetti eterogenei grazie alla traslazione del livello elettrico in modo da interfacciare bus più tradizionali quali I³C e SMBus (utilizzato nel Power Management dei computer) con le nuove reti I³C.

PCB e PCBA: quale è la differenza?

Più o meno tutti sappiamo che per PCB (acronimo di Printed Circuit Board) si intende il classico circuito stampato, lavorato in vario modo, quindi a semplice o doppia ramatura, multilayer (cioè con piste anche negli strati interni della vetronite, con fori metallizzati o meno,



solder-resist, serigrafia eccetera, ma da un po' di tempo, specialmente in ambito tecnico commerciale, si vede utilizzare frequentemente una nuova sigla: PCBA. Ebbene, non si tratta, come verrebbe istintivo pensare, di un nuovo tipo di circuito stampato o di una tecnologia correlata, perché PCBA è l'acronimo di Printer Circuit Board Assembled, vale a dire il circuito stampato montato, quindi completo di tutti i componenti elettronici ed elettromeccanici che deve ospitare da progetto. La sigla PCBA è quindi qualcosa che si utilizza più a livello commerciale per indicare che un'azienda può sia produrre circuiti stampati puri e semplici, sia fornirli già montati.

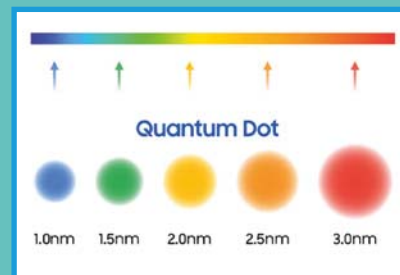
La connessione 10GigE

Quando vogliamo definire una rete dati locale di tipo Ethernet, spesso la distinguiamo per velocità di comunicazione dicendo che è da 10Mbps, 100Mbps, 1Gbps (anche detta rete Gigabit) ma è anche possibile utilizzare altre nomenclature come ad esempio quelle presenti nelle specifiche di telecamere Ethernet industriali: in questo caso, quando si tratta di connessioni LAN Gigabit si utilizza una forma del tipo xGigE, dove al posto della X si indica la velocità in formato numerico. Quindi se una periferica è 10GigE significa che ha un'interfaccia di rete Ethernet a 10 Gbps; se è 1GigE vuol dire che è una Ethernet a 1 Gbps e via di seguito.

Nel caso della 10GigE, o 10 gigabit Ethernet (10 GE, 10GbE) si tratta di un insieme di tecnologie Ethernet definito inizialmente nel 2002 con lo standard IEEE 802.3ae-2002 pubblicato nel 2002. A differenza degli standard Ethernet precedenti, 10GigE definisce solo collegamenti full-duplex punto-punto che vengono realizzati in genere tramite switch e non più hub; inoltre in essi non viene più usato il protocollo CSMA/CD di accesso condiviso al mezzo trasmissivo. Anche in questo caso sono previste varianti che dipendono dal mezzo utilizzato per lo scambio dei dati.

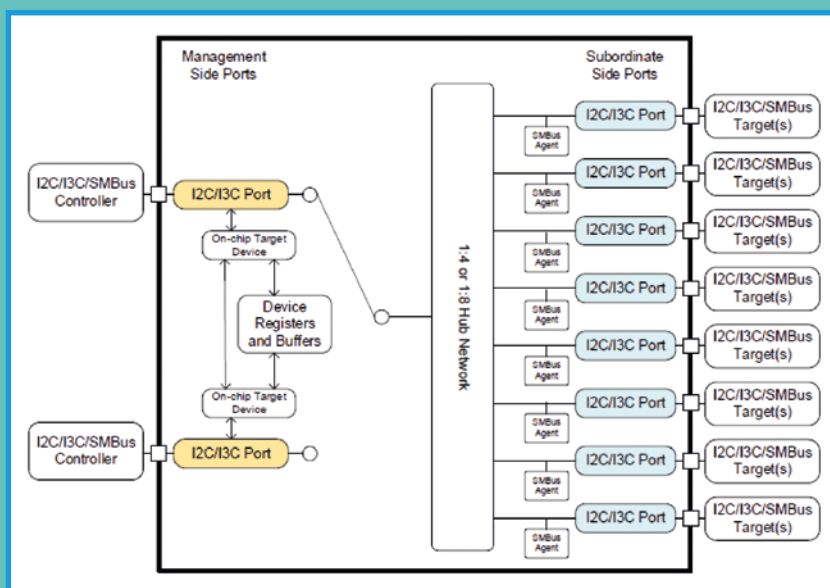
QLED e Quantum Dot

La tecnologia costruttiva degli schermi per monitor e TV evolve costantemente per offrire immagini sempre migliori, definite, contrastate e via di seguito. Una delle più recenti è la QLED (Quantum Dot LED), caratterizzata dall'avere lo schermo a punti quantici; lo schermo Quantum Dot è basato su nanocristalli semiconduttori chiamati punti quantici ed è simile a quello in tecnologia OLED. La tecnologia a punti quantici supporta schermi di grandi dimensioni ed anche flessibili come la OLED, con la differenza che i Quantum Dot non dovrebbero esaurirsi come gli OLED. L'applicazione iniziale di questa tecnologia era finalizzata solo a filtrare la luce proveniente dai LED per la retroilluminazione degli schermi LCD. I primi schermi a punti quantici sono stati commercializzati



↑ Dimensioni e colori dei Qdot.

dalla Sony nel 2013; la stessa Sony, negli anni a venire, ha migliorato la tecnologia e le prestazioni del nero (che comunque rimangono inferiori a quelle degli schermi OLED) affiancando a questi pannelli un ulteriore processore Video chiamato X1 e X1 EXTREME per dare vita agli schermi Triluminos. Le proprietà e le prestazioni degli schermi QLED sono determinate dalla dimensione e dalla composizione dei punti quantici. Una caratteristica dei Quantum Dot è di produrre luce di colore differente a seconda della loro dimensione, come mostra l'immagine qui sopra: maggiore è la dimensione, più rosso è il colore, mentre minore è la dimensione, più il colore tende al blu; più esattamente, la dimensione è proporzionale alla lunghezza d'onda della luce emessa. Si tratta pertanto di punti in grado di emettere luce colorata con grande precisione, perché la dimensione delle particelle cambia con una velocità di livello quantistico, dando vita a emissioni luminose precise ed efficienti. La maggiore efficienza in termini di luminosità garantisce un eccezionale miglioramento della qualità complessiva dell'immagine. I punti quantici possono essere sia foto-attivi (fotoluminescenti) sia elettro-attivi (elettroluminescenti). I QLED sfruttano l'elevata luminosità dei Quantum Dot, la quale indica la luminosità dello schermo ed è importante perché influenza la qualità dell'immagine. Una maggiore luminosità, per esempio, migliora automaticamente il rapporto di contrasto. Quando gli elementi più luminosi e più scuri di un filmato o di un'immagine vengono mostrati in maniera molto dettagliata, si parla di gamma altamente dinamica o HDR (High Dynamic Range). Con la tecnologia Quantum Dot il valore HDR (elemento chiave della qualità dell'immagine) risulta ulteriormente migliorato, dando vita a immagini nitide con elevata profondità di colore. L'effetto della luminosità sulla qualità complessiva del colore dipende dal

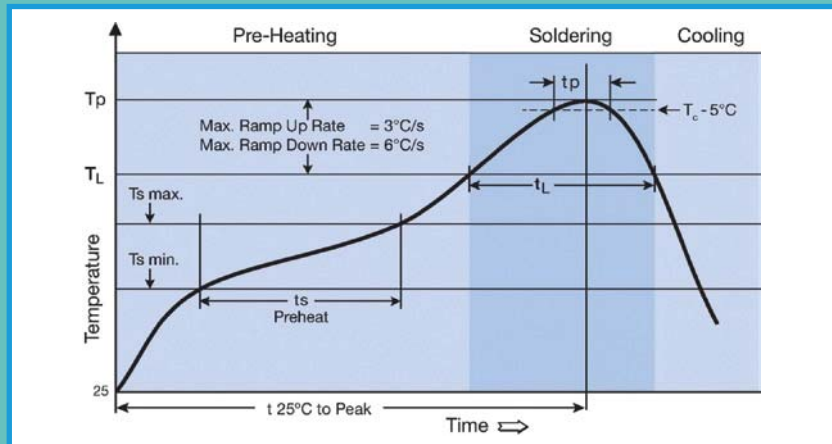


↑ Schema a blocchi degli integrati RG3MxxB12 Hub I²C Renesas.

fatto che lo spettro di ogni colore viene riprodotto con maggiore profondità. La soluzione QLED, unita ad altre tecnologie di formazione del nero, come il Direct Full Array, garantisce una qualità dell'immagine completamente diversa e un'esperienza di visione più realistica rispetto ai televisori LCD e OLED. Peraltro i Quantum Dot, essendo inorganici, durevoli e stabili, superano i limiti degli OLED, i quali sono LED costruiti con un materiale organico che li espone al problema della persistenza dell'immagine e che fa decadere nel tempo la luminosità. Similmente a quanto avveniva con i tradizionali cinescopi, si verifica un effetto di "bruciatura" dei pixel, anche noto come effetto burn-in, al punto che mostrando spesso immagini fisse, a lungo andare un'immagine residua compare sullo schermo: per esempio i loghi dei canali televisivi.

Semiconduttori GaN su Silicio

Nell'intento di mantenere i benefici e minimizzare i difetti, la tecnologia dei semiconduttori ricerca materiali sintetici sempre nuovi e tecnologie costruttive sempre più avanzate; non dimentichiamo la SoS - Silicon on Sapphire, basata su silicio cresciuto sopra uno strato di zaffiro. Ora ST Microelectronics e MACOM Technology Solutions Holdings hanno iniziato la produzione di primi prototipi a radiofrequenza basati su elementi attivi a nitruro di gallio (GaN) cresciuto su substrato silicio; la tecnologia prende il nome di RF GaN-on-Si. Il nitruro di gallio è un semiconduttore di sintesi ottenuto dalla crescita in una camera specifica di materiali nati come



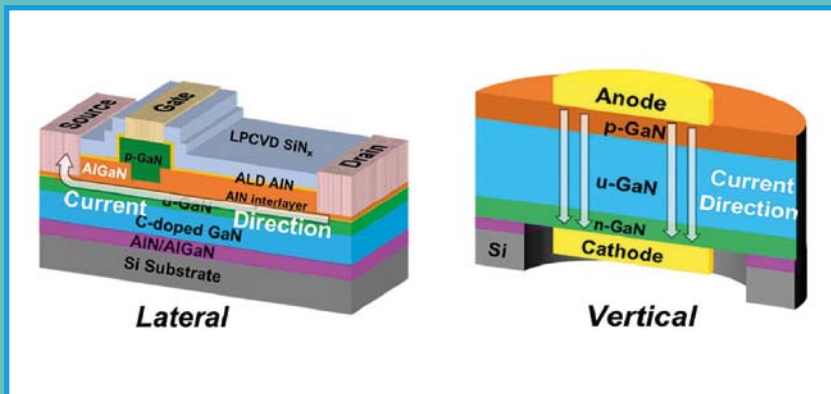
↑ Un tipico Soldering Profile per componenti SMD.

droganti del silicio: pentavalente il nitruro e trivalente il gallio. La crescita sul silicio riduce il costo, mantenendo le qualità del GaN, per competere nella componentistica per altissima frequenza. Questo settore, in particolare quello degli amplificatori di potenza RF, è attualmente dominato dalla tecnologia LDMOS (Laterally Diffused Metal Oxide Semiconductor) che vede MOSFET ottenuti per diffusione laterale anziché planare, prodotti su wafer di silicio o carburo di silicio (SiC). Però la tecnologia RF GaN-on-SiC può essere più costosa a causa del maggior costo dei wafer SiC per alta potenza e per i bassi volumi di questi processi. La tecnologia GaN-on-Si propone come alternativa, in virtù della sua integrazione nei flussi di processo standard, simili a quelli CMOS. I prototipi, per costi e prestazioni, possono competere efficacemente con le tecnologie LDMOS e GaN-on-SiC disponibili sul mercato.

Il Soldering Profile dei componenti

Con l'avvento dei componenti SMD, la saldatura avviene scaldando l'intero corpo e non più i soli terminali, il che impone, soprattutto nelle procedure di montaggio automatizzato ed anche in quelle di rework, il rispetto di determinate temperature e dei tempi per i quali devono essere mantenute, nonché dei gradienti di temperatura ammessi in riscaldamento e raffreddamento. Tutto ciò viene definito dal costruttore nel cosiddetto Soldering Profile, ovvero il profilo di saldatura o rework, un esempio del quale è riportato nell'immagine qui sopra: si compone tipicamente di una curva di pre-riscaldamento (Pre-heating) che definisce il gradiente di riscaldamento, di una fase di picco (Soldering) di temperatura e di una fase di raffreddamento (Cooling) caratterizzata da un gradiente di calo termico.

↓ Struttura di un componente in tecnologia GaN-on-SiC a diffusione laterale (a sinistra) e verticale (a destra).



Web Forum e supporto tecnico

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

www.elettronicaain.it/webforum

OSCILLOSCOPIO DA LABORATORIO 2 CANALI-100 MHz

cod. DS02C10
€ 298,00



CARATTERISTICHE TECNICHE OSCILLOSCOPIO

Display: TFT 7" - Frequenza di campionamento (in tempo reale): 1GSa/s (canale singolo), 500MSa/s (due canali)
Larghezza di banda del canale analogico: 100 MHz - Funzione FFT (Fast Fourier Transform) integrata - Profondità di memoria: 8MB
Software per l'analisi in tempo reale con il PC - Porta USB Host, USB Device

OSCILLOSCOPIO DA LABORATORIO 2 CANALI-100 MHz CON GENERATORE FORME D'ONDA ARBITRARIE FINO A 25 MHz



cod. DS02D10
€ 358,00

CARATTERISTICHE TECNICHE GENERATORE FORME D'ONDA

Numero canali: 1 - Sample Rate: 200 MSa/s - Risoluzione Verticale: 12 bit - Massima Frequenza di uscita: 25 MHz
Forma d'onda di uscita: sine, square, ramp, Exp, noise, DC - Forme d'onda arbitraria: Arb1, Arb2, Arb3, Arb4
Profondità forma d'onda: 4KSa - Ampiezza (range di uscita): 10mV~7Vp-p (alta impedenza) / 5mV~3.5Vp-p (50Ω)
DC offset: range $\pm 3,5$ V, alta impedenza / $\pm 1,75$ V, 50 Ω - Risoluzione: 100 μ V o 3 bits - Impedenza di uscita: 50 Ohm.

Le caratteristiche tecniche della parte relativa all'oscilloscopio sono le stesse del modello DS02C10.

LeddarTech ha rilasciato il sistema LeddarSteer, una soluzione di guida del raggio digitale progettata per gli sviluppatori di sensori intelligenti LIDAR e i fornitori di soluzioni ADAS di livello 1-2. Utilizzando una pila di celle a cristalli liquidi alternati e reticoli di polarizzazione, è possibile orientare la luce su angoli diversi a una lunghezza d'onda specifica, al fine di aumentare un campo visivo lidar massimizzandone le prestazioni. La società afferma che lo sterzo digitale del raggio offre una serie di vantaggi agli sviluppatori di sensori lidar e intelligenti e ai clienti di livello 1 e 2. Ad esempio, la regolazione al volo del raggio può essere ottenuta tramite software, inclusa la regolazione fotogramma per fotogramma. È inoltre compatibile con un'ampia

➤ LIDAR più precisi con il Digital beam



gamma di architetture lidar. LeddarSteer può essere perfettamente integrato in un LiDAR esistente, per espandere il campo visivo, oppure integrato in un nuovo sviluppo LiDAR. La soluzione fornisce anche un rapporto segnale/rumore migliorato concentrando la potenza del laser su una piccola regione di interesse

riducendo le dimensioni, il costo e la complessità dei componenti mantenendo o aumentando il numero di pixel e la risoluzione.

La direzione del raggio digitale si riferisce al cambiamento della direzione degli impulsi laser in un LIDAR. L'utilizzo di una pila di celle a cristalli liquidi alternati e reticoli di polarizzazione consente di orientare la luce a diverse angolazioni a una lunghezza d'onda specifica al fine di aumentare un campo visivo LIDAR massimizzandone le prestazioni, consentendo di ottenere la guida del raggio digitale per LIDAR.

LeddarSteer è l'unico componente intelligente per la guida del raggio LASER progettato per la produzione di massa.

leddartech.com

➤ Può, l'AI, pensare come un essere umano?

Il modo in cui l'intelligenza artificiale raggiunge informazioni e prende decisioni è poco noto, il che solleva preoccupazioni su quanto possa essere affidabile l'apprendimento automatico. Questo perché il Machine Learning trova sempre più applicazioni nel mondo reale. I ricercatori del MIT hanno sviluppato un metodo che aiuta un utente a comprendere il ragionamento di un modello di apprendimento automatico e come tale ragionamento si confronta con quello di un essere umano. Sebbene esistano strumenti per aiutare gli esperti a dare un senso al ragionamento di un modello, spesso questi metodi forniscono informazioni solo su una decisione alla volta e ciascuna deve essere valutata manualmente. I modelli vengono comunemente addestrati utilizzando milioni di input di dati, rendendo quasi impossibile per un essere umano valutare abbastanza decisioni per identificare i modelli. Ora, i ricercatori del MIT e IBM Research hanno creato un metodo che



consente a un utente di aggregare, ordinare e classificare queste singole spiegazioni per analizzare rapidamente il comportamento di un modello di apprendimento automatico. La loro tecnica, chiamata Shared Interest, potrebbe aiutare a scoprire le tendenze relative al processo decisionale di un modello, ad esempio, forse il modello

viene spesso confuso da caratteristiche distraenti e irrilevanti, come gli oggetti sullo sfondo nelle foto. Shared Interest sfrutta tecniche popolari che mostrano come un modello di apprendimento automatico ha preso una decisione specifica, nota come metodi di salienza.

news.mit.edu

» Continental digitalizza il test dei pneumatici

Continental ha presentato un nuovo veicolo di prova senza conducente completamente elettrico per misurare le prestazioni di frenata dei pneumatici delle autovetture, con l'obiettivo di ridurre ulteriormente gli spazi di frenata per migliorare la sicurezza. Situato presso il sito di prova Contidrom dell'azienda vicino ad Hannover, in Germania, l'AVA (Analytic Vehicle AIBA) combina i vantaggi di un metodo di prova indipendente dal modello del veicolo, con l'ambiente di prova controllato e riproducibile dell'Analizzatore automatico di frenata indoor (AIBA). Utilizzando il nuovo sistema, gli esperti di test sui pneumatici di Continental possono ora eseguire test indipendentemente dalle caratteristiche specifiche del veicolo.

Durante i test di frenata dei pneumatici, l'AVA viene accelerato a una velocità di 65 km/h, aiutato da un azionamento lineare elettromagnetico, basato sulla moderna tecnologia delle montagne russe. Una volta raggiunta la velocità, possono essere condotti diversi test di frenata completamente automatizzati. Il sistema di trasmissione AVA ha due assali azionati elettricamente che sono alimentati da una batteria ad alte prestazioni. Producendo un'elevata coppia massima, l'AVA può fornire una velocità costante mentre i pneumatici di prova sul terzo asse vengono sistematicamente frenati. L'impianto frenante integrato è dotato della tecnologia Brake-by-Wire del settore automobilistico di Continental.

» Qualcomm debutta nel WiFi 7 Networking

Qualcomm Technologies sta sviluppando il portafoglio di piattaforme di rete WiFi 7 più scalabili al mondo, con offerte che vanno da 6 a 16 stream, per access point aziendali di prossima generazione, router ad alte prestazioni e gateway carrier. Le piattaforme Qualcomm Networking Pro Series di terza

generazione offrono funzionalità chiave come il supporto del canale a 320 MHz, stabilendo nuovi benchmark delle prestazioni nella rete wireless con capacità dell'interfaccia wireless fino a 33 Gbps e velocità di picco superiori a 10 Gbps. Le piattaforme Qualcomm Networking Pro Series sono supportate dal servizio

chiavi in mano di Qualcomm Technologies per il controllo automatico della frequenza (AFC) per consentire le massime prestazioni nella banda a 6 GHz. Basandosi sull'eredità multigenerazionale delle piattaforme Qualcomm Networking Pro Series, i prodotti combinano le funzionalità Wi-Fi 7 con le tecnologie di gestione multicanale intelligente di Qualcomm Technologies per migliorare le velocità, ridurre la latenza e migliorare l'utilizzo della rete per gli utenti del WiFi. Dispositivi 6/6E offrendo al contempo un throughput rivoluzionario e una latenza incredibilmente bassa per la prossima generazione di dispositivi client Wi-Fi 7. Introducendo innovazioni Wi-Fi 7 come operazioni multi-link simultanee, 4K QAM, configurazioni quad-band in un'architettura altamente scalabile, modulare e ottimizzata, funzionalità avanzate per il rilevamento delle interferenze e il funzionamento multilink, la serie Wi-Fi 7 Network Pro consente una bassa latenza deterministica in ambienti wireless condivisi difficili, consentendo prestazioni delle applicazioni che rivalgano con lo spettro privato. I prodotti possono supportare il backhaul wireless ad alta velocità e bassa latenza per il Wi-Fi mesh domestico e l'infrastruttura aziendale con prestazioni affidabili anche in presenza di interferenze vicine.

www.qualcomm.com





Trasferendo elettronicamente il segnale di frenata, la risposta in frenata del sistema è rapida e precisa, consentendo prove e misurazioni ripetibili e accurate. L'AVA di Continental viene utilizzato all'interno dell'Analizzatore di frenata automatizzata per interni, un sistema entrato in funzione nel 2012. Il banco di prova è in grado di condurre prove di frenata su una vasta gam-

ma di superfici stradali su una pista di prova lunga 75 m. La sala prove è dotata di cinque superfici stradali intercambiabili idraulicamente e fino a 100.000 prove di frenata possono essere eseguite ogni anno su strade asciutte, bagnate o ghiacciate. Il sistema è integrato in una sala lunga 350 m.
www.continental.com

› Sicurezza e privacy nella videosorveglianza con l'AI

Le telecamere di videosorveglianza sono proliferate in tutto il mondo, sollevando preoccupazioni sulla privacy acuite dal fatto che il Machine Learning consente l'analisi video automatizzata su vasta scala. Esistono molte applicazioni per l'analisi video automatizzata dei filmati di sorveglianza, come ad esempio: aiutare i funzionari sanitari a misurare la percentuale di persone che indossano maschere; lasciare che i dipartimenti di trasporto monitorino la densità e il flusso di veicoli, pedoni e biciclette per capire dove aggiungere marciapiedi e piste ciclabili.

I tentativi di difendere la privacy da tale tecnologia spesso comportano l'offuscamento dei volti o la copertura di riquadri neri. Questi metodi possono impedire un'analisi utile, pur non avendo l'effetto previsto di preservare l'anonimato.

Oggi un sistema chiamato Privid (creato dal Computer Science and Artificial Intelligence Laboratory -CSAIL- del MIT), consente agli analisti di esaminare i video alla ricerca di dati statistici senza rivelare informazioni di identificazione personale; funziona accettando prima il codice di un anali-

sta contenente una query che attiva un conteggio automatico, ad esempio, del numero di persone che indossano maschere in un feed video e della densità della folla. Il sistema quindi suddivide il filmato in segmenti ed esegue il codice su ogni blocco. Invece di riportare i risultati di ciascun segmento all'analista, Privid aggrega i dati e aggiunge un po' di rumore prima di restituire i risultati. L'obiettivo è consentire agli analisti con query lecite di ottenere i dettagli che desiderano, limitando l'accesso ai dati di sorveglianza grezzi che consentirebbero ai malintenzionati di ottenere troppe informazioni. Ad esempio, quando si tratta di un feed video che osserva più incroci di città, qualsiasi query potrebbe affermare di voler contare il numero di persone che passano ogni ora, ma mentre la query lecita di un dipartimento di pianificazione urbana potrebbe voler contare i numeri di pedoni per pianificare meglio gli attraversamenti pedonali, quella di un hacker potrebbe mirare a rintracciare alcune persone attraverso i loro volti. Supponendo che Privid esegua tutte le query, l'aggiunta di rumore non disturba la risposta alle query lecite ma confonderebbe chi tentasse di utilizzare in modo improprio i dati.

news.mit.edu



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.

SSR Automotive da Texas Instruments

TI presenta nuovi relé a stato solido, completi di driver e interruttori isolati, in grado di offrire l'affidabilità del gruppo powertrain e dei sistemi di gestione della batteria da 800 V delle auto elettriche: si tratta dei TPSI3050-Q1 che integra un sistema di alimentazione a 10V per il gate driver di uscita e lo switch isolato TPSI2140-Q1 da 1.400 V e 50 mA. I nuovi relé a stato solido sono in grado di collegare e scollegare i carichi attraverso una singola barriera di isolamento nel giro di pochi microsecondi, rispetto ai millisecondi necessari ai relé elettromeccanici, e consentono quindi una maggiore sicurezza. Il TPSI3050-Q1 offre un isolamento rinforzato fino a 5 kVRMS e una durata operativa 10 volte superiore rispetto ai relé elettromeccanici che possono degradarsi con il passare del tempo. Inoltre, il TPSI2140-Q1 offre un isolamento di base fino a 3,75 kVrms, che gli permette di fornire un'affi-



fidabilità alla perforazione del dielettrico in base al tempo quattro volte superiore rispetto ai fotorelé a stato solido. Progettato per misurazioni in alta tensione e per il monitoraggio dell'isolamento, il TPSI2140Q1 funziona con un monitor per pacco batteria come il

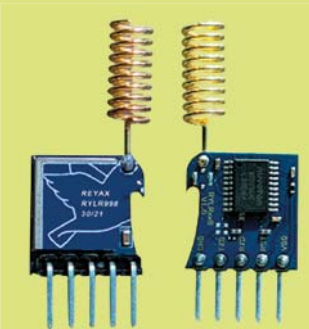
BQ79631-Q1, con una precisione maggiore rispetto ai fotorelé a stato solido. Il TPSI2140-Q1 permette di utilizzare resistori <1 MΩ ed è in grado di resistere ad una corrente a valanga del 300% più elevata rispetto ai fotorelé tradizionali.

www.ti.com

REYAX propone un RTX basato su modem LoRa

REYAX Technology propone il modulo RYLR998, RTX dotato di LoRa di Semtech operante su una gamma di frequenza 868/915 MHz. Come risultato dell'integrazione LoRa, RYLR998 ha un consumo energetico estremamente basso che prolunga efficacemente la durata della batteria, ideale per vari dispositivi di trasmissione che richiedono batterie per una varietà di casi d'uso IoT come Smart Home e monitoraggio delle risorse.

Il modulo RYLR998 di REYAX Technology può controllare una varietà di sensori e driver con solo una piccola quantità di spazio a 32 x 14 x 2,2 mm. Inoltre utilizza un microcontrollore NUVOTON e un motore ad alte prestazioni Semtech LoRa, che insieme supportano l'interfaccia UART e possono essere controllati facilmente tramite AT-Command. Rispetto a FSK, GFSK o



altre tecnologie di trasmissione tradizionali, RYLR998 ha un'elevata sensibilità che può ricevere il più basso -129dBm per ridurre al minimo il consumo di corrente, secondo REYAX Technology. La piattaforma device-to-Cloud LoRa Semtech è una soluzione a lungo raggio e bassa potenza per applicazioni IoT, che consente lo sviluppo e l'implementazione rapidi di reti IoT, gateway, sensori, prodotti a modulo e bassissimo consumo, efficienti in termini di costi e servizi IoT in tutto il mondo.

www.reyax.com

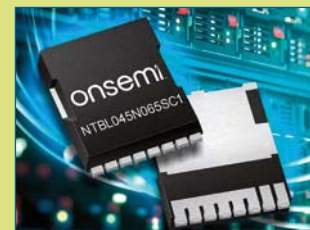
Da ONsemi il primo MOSFET SiC in TOLL

ON ha annunciato il primo MOSFET in carburo di silicio (SiC) in package TOLL (TO-Leadless) che soddisfa la crescente richiesta di dispositivi di commutazione ad alte prestazioni adatti per lo sviluppo di progetti contraddistinti da un'elevata densità di potenza. Fino a non molto tempo fa, i dispositivi SiC erano forniti in package D2PAK a 7 terminali, caratterizzati da ingombri decisamente superiori. Le ridotte dimensioni, pari a soli 9,9x11,68x2,3 mm, del il package TOLL, riducono l'ingombro rispetto al package D2PAK. Oltre alle dimensioni ridotte, il package TOLL assicura anche migliori prestazioni termiche e un'induttanza del package inferiore (pari a soli 2 nH) rispetto al package D2PAK a 7 terminali. La connessione Kelvin-source assicura una riduzione del rumore di gate e delle perdite di potenza in fase di commuta-

zione, e del 60% delle perdite in fase di accensione (EON) rispetto a un dispositivo privo di connessione Kelvin.

Il primo MOSFET SiC disponibile in package TOLL è il modello NTBLO45N065SC1 destinato ad applicazioni complesse quali alimentatori a commutazione (SMPS – Switch Mode Power Supply), alimentatori per server e apparati telecom, inverter fotovoltaici, UPS e sistemi di energy storage. Il dispositivo è adatto a progetti che devono rispettare le specifiche dei più stringenti standard di efficienza, tra cui ErP e 80 PLUS Titanium.

www.onsemi.com



MCU Renesas certificati per algoritmi crittografici



Il motore di sicurezza dei microcontrollori Arm Cortex-M a 32 bit RA di Renesas è ora certificato dal National Institute of Standards e Tecnologia (NIST) Cryptographic Algorithm Verification Program (CAVP). I driver per utilizzare la modalità protetta SCE9 certificata sono inclusi nel pacchetto software

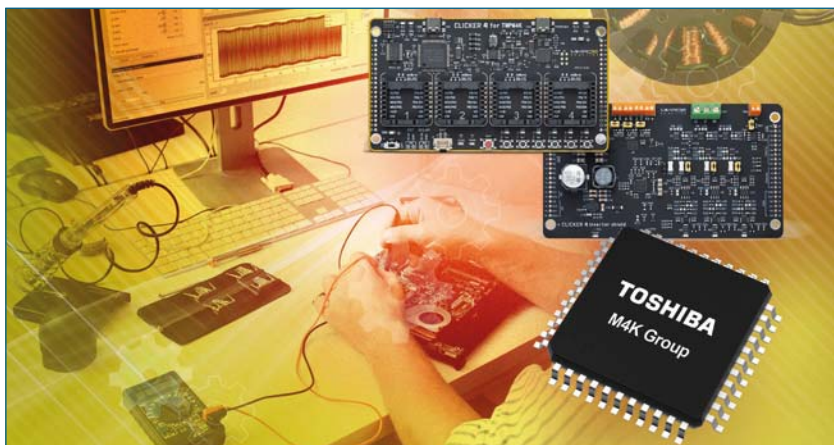
della famiglia RA dalla versione v3.6.0. La certificazione NIST CAVP riguarda una suite completa di algoritmi crittografici, tra cui modalità multiple di Advanced Encryption Standard (AES), hashing, Rivest Shamir Adleman (RSA) ed Elliptic Curve Cryptography (ECC) per la generazione e autenticazione

di chiavi, Key Agreement Schemes e generatore di bit casuali deterministico (DRBG). La certificazione fornisce una verifica indipendente della corretta implementazione degli algoritmi crittografici, necessaria a garantire l'interoperabilità della connettività.

www.renesas.com

Toshiba e MikroE per il controllo motori

Toshiba Electronic supporta MikroElektronika nella piattaforma di valutazione. La scheda di sviluppo Clicker 4 di MIKROE per MCU M4K di Toshiba, combinata con il modulo Clicker 4 Inverter Shield, è una soluzione per il controllo motore BLDC. La scheda Clicker 4 per moduli TMPM4K è dotata di un debugger integrato e include quattro prese mikroBUS per collegare una vasta gamma di schede Click di MIKROE. Sono anche presenti connettori di estensione, porte di debug JTAG/SWD, indicatori LED e pulsanti. Questa scheda è accompagnata dal modulo Clicker 4 Inverter Shield, che dispone di sei MOSFET per il pilotaggio del motore e di un alimentatore switching utilizzato per regolare la tensione di alimentazione del motore fino a 48V, con una fonte di alimentazione regolata a 5V, la quale può essere utilizzata per alimentare la scheda di controllo esterna. La sua interfaccia consente di ottenere una risposta di posizionamento dai sensori ad effetto Hall e dagli encoder incrementali. È incorporata la protezione dalle sovracorrenti. Le MCU



M4K sono basate su core Arm Cortex-M4 con FPU e dispone di unità di protezione della memoria (MPU) e funzionalità A-PMD (Advanced-Programmable Motor Driver), insieme a 256 kByte di Flash codice e 32 kByte di Flash dati. I moduli Clicker 4 per TMPM4K e Clicker 4 Inverter Shield sono supportati dal software MCU Motor Studio disponibile sul sito Web Toshiba. Si tratta di un software semplice,

ben strutturato e versatile con due componenti principali: un PC Tool di controllo motore che consente la configurazione dei parametri, il controllo dell'azionamento, la connessione e la diagnostica in tempo reale tramite interfaccia UART ad alta velocità, e inoltre il firmware di controllo motori scalabile e completamente configurabile per la MCU M4K.

www.toshiba.com

Connettori IP67 per elettromedicali e industria

Binder presenta i connettori serie 770 NCC (Not Connected Closed). Quando sono accoppiati, i connettori soddisfano i requisiti di protezione IP67 e impediscono l'ingresso della polvere e sono protetti contro l'immersione temporanea. Quando sono scollegati, offrono grado di protezione IP54 contro il contatto fisico,

la polvere e gli spruzzi d'acqua da tutti i lati. La serie è stata progettata per l'uso in ambienti sanitari e industriali. I connettori sono dotati di un sistema di bloccaggio a baionetta, affidabile e di facile uso, che assicura l'accoppiamento e il disaccoppiamento rapidi e sicuri. Questo, grazie a un elemento in plastica precaricato

a molla che racchiude i pin proteggendoli. I connettori a 8 pin serie 770 NCC sono dotati di cavi in poliuretano (PUR) o PVC a norma UL, offerti con lunghezze di 2 e 5 m, a sezione di 8x0,25 mm² e diametro della guaina di 6 mm. La resistenza massima del cavo è di 79 Ω/km. La tensione nominale è 175 V a una corrente di



2 A (tolleranza fino a 1,75 kV impulsivi). Il range di temperatura è fra da -25 °C a +85 °C e la durata di 5.000 cicli.

www.binder-connector.com

Da STM un LDO a basso rumore

Il regolatore di tensione a bassa caduta LD56020 200 mA di STM funziona con un'alimentazione da 1,1 V a 5,5 V e presenta un basso rumore di uscita. Ideale per dispositivi mobili, sensori di visione e moduli wireless, l'LD56020 dal prezzo competitivo ha una tensione di caduta estremamente bassa di soli 190 mV e una corrente di riposo massima di 25

µA. Un pin di abilitazione consente alla logica di controllo di mettere il regolatore in modalità standby, che riduce la corrente al di sotto di 0,1 µA.

Con una reiezione della tensione di alimentazione (SVRIN) di 90 dB (1 kHz, 20 mA) e un rumore di uscita di soli 8,8 µVRMS (da 10 Hz a 100 kHz),

l'LD56020 fornisce una linea di alimentazione pulita agli elementi sensibili ai disturbi. Il circuito è stabilizzato con un solo condensatore ceramico di piccola capacità su ciascuno dei terminali di ingresso e di uscita. Il regolatore integra funzioni di protezione tra cui il blocco di

sottotensione (UVLOCK) per prevenire un'eccessiva dissipazione in caso di guasto del sistema.

Nell'integrato è anche prevista una protezione da sovratemperatura e un percorso di scarico interno, nonché il foldback della corrente di cortocircuito. L'LD56020 è ora in produzione nel package chip-scale CSP-4, delle dimensioni di 0,65x0,65mm.

www.st.com/ldo



Ultra-low-dropout voltage regulator boosts battery runtime and stability

Moduli EtherCAT DAQ da AMC

AMC rende disponibili i sistemi di acquisizione dati Gantner Instruments, moduli di misura con un'interfaccia EtherCAT diretta, abbinabili a termocopie, Pt100, LVDT, estensimetri, IEPE/ICP, carica piezoelettrica, misura di alta tensione fino a 1200 Vcc. Validi esempi sono il controller Q.station XT EC e l'accoppiatore bus Q.Series XE BC. Il Q.station XT EC supporta 16 canali (4 canali per UART) a 100 kHz, 128 canali a 10 kHz ed è possibile collegare 64 moduli. Ha interfaccia 1Gig-E, TCP/IP, UDP, fino a 16 MB/s Modbus TCP/IP, ASCII, Server Web con porta ad alta velocità, Client Web ed e-mail, sincronizzazione e timestamp dei valori misurati, memoria dati din. 500 MB, stat. 4GB, 6 ingressi digitali, input encoder per misure veloci con



sincronia angolare, misura di frequenza e PWM, contatore, segnali di stato, funzionalità PAC con ampia libreria, regolatore PID veloce, controllo sequenza, data logger, funzione di trasferimento. A supporto, l'edge controller della serie Q.station X per l'acquisizione, memorizzazione e trasmissione dei dati, la sincronizzazione precisa dei dati, l'acquisizione ridondante ad alta velocità e la comunicazione parallela.

www.amc-systems.de

Qualcomm ottimizza i processori AMD Ryzen

AMD e Qualcomm collaborano per ottimizzare il sistema di connettività Qualcomm FastConnect per le piattaforme di elaborazione basate su processore AMD Ryzen, a partire da AMD Processori Ryzen PRO serie 6000 e sistema Qualcomm FastConnect 6900. Con FastConnect 6900, gli ultimi laptop aziendali con processore AMD Ryzen sono dotati di connettività WiFi 6 e 6E, incluse funzionalità wireless avanzate abilitate con il sistema operativo Windows 11.

In collaborazione con Microsoft, i PC Windows 11 di nuova generazione, come Lenovo ThinkPad Z Series e HP EliteBook 805 Series, possono sfruttare tutto il potenziale di Windows 11 WiFi Dual Station tramite Qualcomm 4-Stream Dual Band simultanea. Sfruttando la banda a 6 GHz, gli utenti di laptop di nuova generazione possono sfruttare appieno la larghezza di banda e i miglioramenti della velocità, senza competere con i dispositivi non 6E per trovare un canale disponibile in bande ormai affollatissime.

Per gli amministratori IT, AMD Manageability Pro-

cessori ora disponibile nei sistemi Ryzen PRO 6000 è una soluzione che consente la gestione remota delle piattaforme commerciali AMD. Se abbinata a FastConnect 6900, questa soluzione consente la gestibilità wireless con il supporto per oltre 32 profili DASH (Open Standard-Based) ampiamente utilizzati. Insieme, il processore AMD Manageability e FastConnect 6900 sbloccano funzionalità avanzate per la gestione dei sistemi da parte dell'IT.

La gestione remota WiFi fuori banda è uno strumento importante per i responsabili IT aziendali, utilissimo per diagnosticare e risolvere i problemi, anche quando il sistema operativo non è in esecuzione.

I processori AMD Ryzen PRO serie 6000 con tecnologia FastConnect 6900 consentono ai laptop aziendali di nuova generazione di disporre degli strumenti di elaborazione e di connettività necessari per poter funzionare in ambienti moderni, offrendo una gestibilità remota di livello professionale per gli utenti nel nuovo ambiente di lavoro ibrido.

www.qualcomm.com



High-performance advanced Wi-Fi 6 & 6E system for compute



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: futuranet.it/elettronica.in/abbonamenti



ABBONAMENTO ANNUALE PER RIVISTA DIGITALE

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

COME ABBONARSI

→ **on-line**

compilando il modulo riportato
nella pagina "**Abbonamenti**"
del nostro sito www.elettronica.in.

→ **e-mail**

scrivendo a
abbonati@elettronica.in

→ **telefono**

telefonando a
+39-0331-752668

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



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

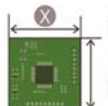
1 Layer

2 Layers

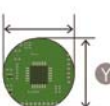
4 Layers

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

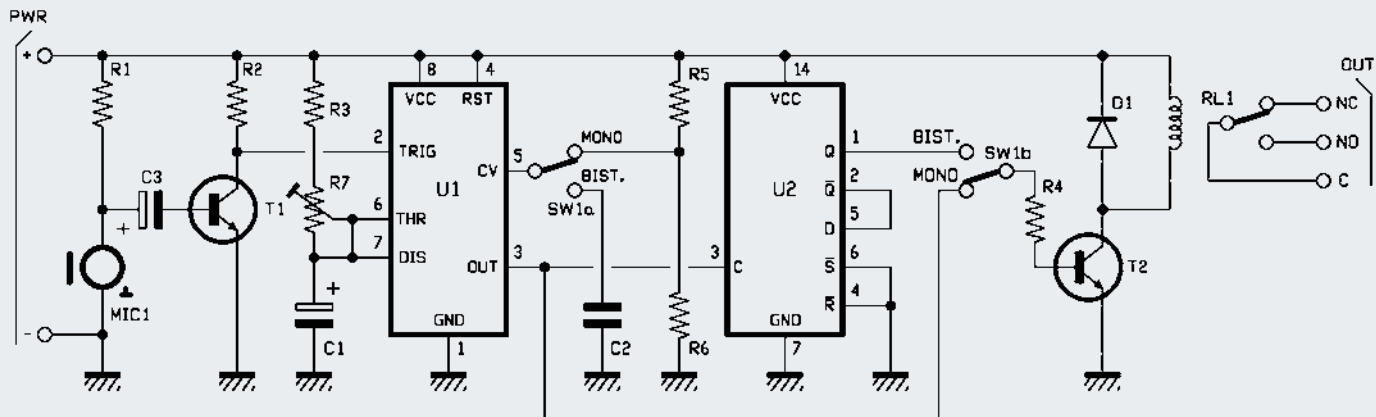
ECCO QUALCHE ESEMPIO:

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.

di **DAVIDE SCULLINO**

In questi anni in cui la Domotica pervade gli ambienti domestici estendendosi in generale a tutti i luoghi chiusi frequentati dall'uomo, pensare di strabiliare il pubblico o i propri ospiti con un automatismo che comandi una lampadina, una tapparella o l'accensione dell'impianto di riproduzione sonora, sarebbe anacronistico e oltretutto illusorio; infatti esistono ormai impianti domotici ultra-sofisticati, interfacciati ad Internet o tramite Bluetooth e WiFi, smartphone ed altra appendice dell'uomo del terzo millennio, che fanno questo senza colpo ferire e con la massima naturalezza. Potrebbe invece stupire, soprattutto coloro che dell'Elettronica percepiscono e conoscono soltanto l'aspetto della programmazione (che poi tanto Elettronica non è, in quanto è più Informatica...) dimostrare a chi entra in casa nostra che è possibile accendere le luci o fare quanto appena accennato, con un semplice circuito a logica cablata e



senza avere nessuno sofisticato, fantasmagorico e costosissimo impianto domotico "connesso" e capace di sopravvivere anche se la rete andasse giù. Un battito di mani entrando in casa e la luce si accende; un altro passando dall'atrio al salone e la luce si spegne. Tutto molto semplice, tant'è che con i componenti tradizionali e senza scrivere una riga di codice, questo già si faceva ai tempi dei primi film dedicati all'Agente 007.

Ecco perché, per non dimenticarci che ci chiamiamo Elettronica In e per allinearci a quella "corrente" di progettisti e produttori di integrati che vogliono rendere accessibili funzioni complesse con un approccio "codeless" vi proponiamo la realizzazione di un Clap Switch, ossia un interruttore capace di riconoscere il rumore del battito delle mani (ed anche quelli simili per spettro di frequenza) ed attivare in modo astabile (temporizzato) o bistabile un

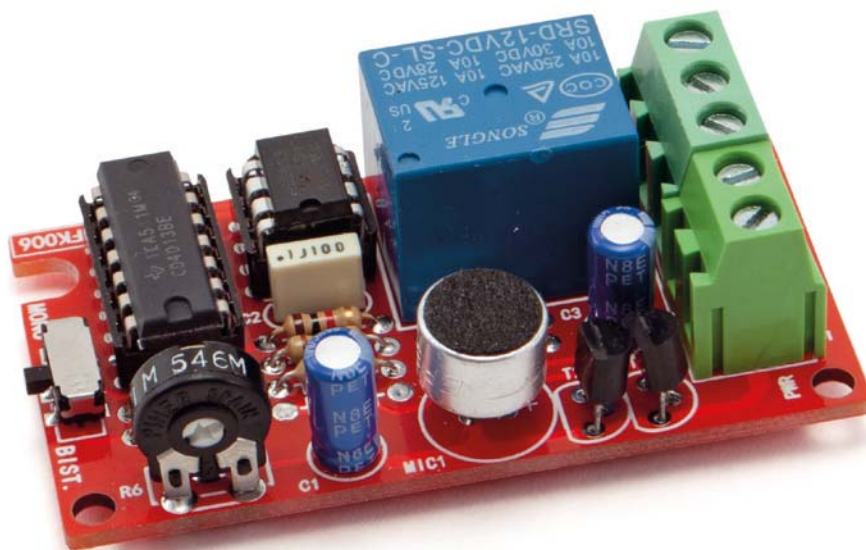
dispositivo alimentato elettricamente; il tutto con pochi, economici e comuni componenti elettronici. Scoprirete di cosa si tratta analizzando con noi lo schema elettrico proposto in queste pagine.

SCHEMA ELETTRICO

Il circuito dell'interruttore a battito di mani è sostanzialmente composto da tre blocchi: un rilevatore acustico basato su un microfono e un transistor, un multivibratore monostabile che serve, quando triggerato dal rilevatore acustico, a generare impulsi di durata ben definita ed un flip-flop, che consente di implementare la modalità bistabile, quando desiderata.

All'uscita del flip-flop si trova l'elemento attuatore elettromeccanico che provvederà a commutare la corrente nel carico e che, nel caso specifico, è un relé tradizionale alimentato da un transistor.

Il prototipo realizzato per le prove: a sinistra, notate il doppio deviatore.



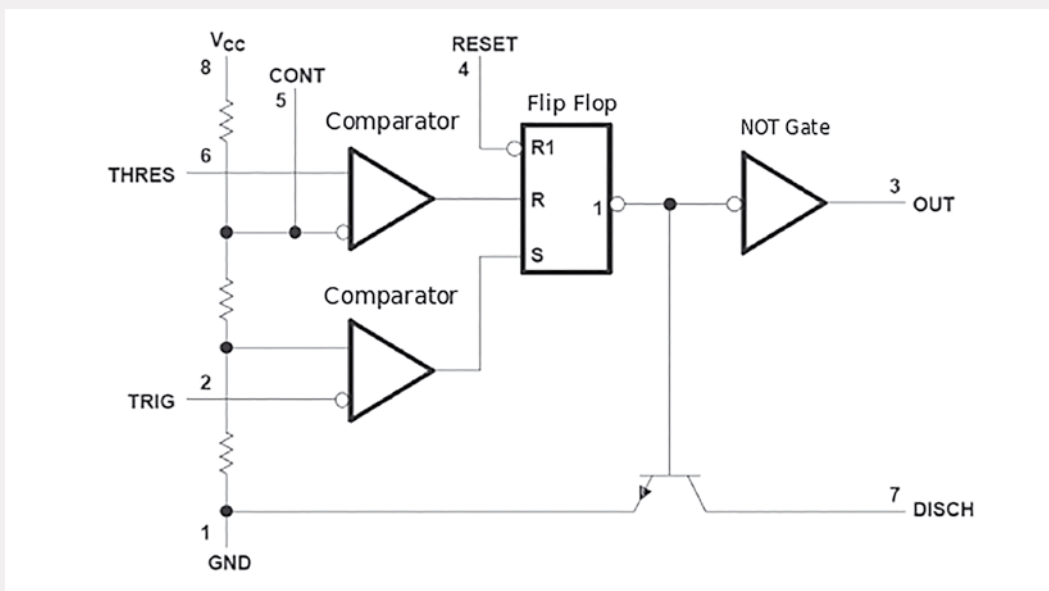


Fig. 1
Schema a
blocchi del
555.

Ma andiamo con ordine e analizziamo individualmente i singoli stadi, correlandone il funzionamento con quelli via-via seguenti.

Il rilevatore acustico è composto da una capsula microfonica electret condenser di tipo attivo (con jFET all'uscita) che ci serve a captare suoni e rumori presenti nell'ambiente e che, nello specifico, capterà il rumore del battito di mani o di due tavole o di oggetti rigidi e semi-rigidi che si urtano; il segnale acustico captato dalla membrana del microfono MIC1 viene convertito in una tensione variabile che, amplificata dal FET interno alla capsula, si presenta sul polo positivo del trasduttore, ma sovrapposta alla componente continua dovuta alla resistenza di polarizzazione R1, necessaria a mantenere il punto di lavoro del FET in una zona che consenta di ottenere in uscita oscillazioni simmetriche quanto più possibile.

A sopprimere la componente continua e lasciar transitare il solo segnale variabile (quello audio, in definitiva...) provvede il condensatore elettrolitico C3, posto tra il polo positivo della capsula microfonica e la base del transistor T1; questo NPN viene utilizzato come amplificatore di tensione in classe C, cioè amplifica solamente quando il livello della tensione tra base ed emettitore supera gli 0,6V. Per effetto di ciò, l'amplificazione non sarà lineare, ma del resto questo non ci interessa, in quanto T1 serve per elevare ulteriormente il livello del segnale trasdotto da MIC1 affinché raggiunga il nostro scopo, che è quello di ottenere impulsi abbastanza ampi da essere sfruttati per triggerare lo stadio temporizzatore facente capo all'integrato 555 (U1)

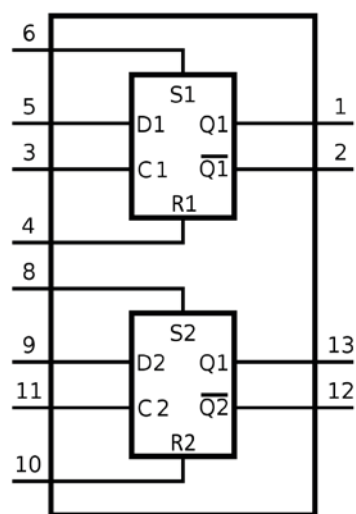
che trovate nella classica configurazione da multivibratore monostabile suggerita dal costruttore. Quindi abbiamo il RESET disattivato e posto fisso sulla linea di alimentazione ed abbiamo il trigger (piedino 2) dotato di resistore di pull-up affinché rimanga normalmente a livello alto (lasciando il timer a riposo) e venga eccitato dal collettore del transistor T1 quando il suo potenziale scende al di sotto di 1/3 di quello di alimentazione dell'U1 (questo lo esporremo tra breve).

Vediamo dunque come funziona il 555 partendo dall'accensione e supponendo C1 inizialmente scarico, ossia avente una tensione di 0 volt; per comodità riportiamo in **Fig. 1** lo schema a blocchi interno del timer 555, in modo che vi sia più chiara la descrizione. La condizione iniziale del circuito vede il piedino 6 (ed il 7 con esso...) a circa zero

CARATTERISTICHE TECNICHE

 Tensione di alimentazione: 12Vcc	 Tensione commutabile: 250Vca
 Corrente assorbita 60mA	 Corrente commutabile: 10A
 Uscita a relé con scambio completo	 Modalità d'uscita monostabile o bistabile

Fig. 2
Schema a blocchi
interno del 4013.



volt e il 2 a livello alto (vi provvede il resistore di pull-up R4, che termina sul collettore dell'NPN) quindi l'uscita del comparatore inferiore è a livello basso e quella del comparatore superiore si trova anch'essa a livello basso; ciò determina il livello logico basso sull'ingresso SET del flip-flop interno al 555 e lo stesso stato sul RESET. L'uscita Q è a livello basso, mentre la /Q si porta ad 1 logico mandando in saturazione il transistor collegato al DISCH e mantenendo quindi il condensatore scarico, il che conferma la condizione supposta nello stato iniziale.

La situazione rimane questa fin quando non viene portato il piedino 2 a un potenziale inferiore ad 1/3 di quello di alimentazione (questo perché l'ingresso invertente del comparatore inferiore è polarizzato con 1/3 del potenziale applicato al piedino 8), il che si traduce nel forzare a livello alto l'uscita del comparatore inferiore e con essa il SET del flip-flop. Per effetto di ciò, il SET del flip-flop riceve un impulso a livello alto e l'uscita va a zero logico (così fa anche la /Q) mentre il piedino 3 del 555 passa anch'esso a livello alto.

A questo punto C1 inizia a caricarsi attraverso la corrente che fluisce nella serie R1-R8 (il pin 7 non assorbe corrente perché il relativo transistor è interdetto), fin quando supera la tensione di soglia del comparatore superiore, che ha un valore di poco superiore ai 2/3 della tensione di alimentazione del 555, ovvero alla soglia di commutazione del comparatore stesso.

Quando ciò accade, l'uscita del comparatore superiore si porta a livello logico alto e resetta il flip-flop, la cui uscita diretta passa a zero logico mentre la /Q si porta a livello alto, mandando in saturazione il transistor interno collegato al piedino 7, il quale

scarica il condensatore elettrolitico C1 e pone fine alla temporizzazione.

Se nel frattempo nessun suono viene più rilevato, il circuito rimane come nelle condizioni iniziali, in attesa che il microfono capti un nuovo rumore abbastanza intenso da far scendere il collettore del transistor T1 ad un potenziale inferiore a quello della soglia di trigger, altrimenti ricomincia il ciclo descritto. Ciò detto, andiamo a vedere quale è la durata teorica dell'impulso d'uscita a livello alto e quindi dell'eccitazione del relé si determina come:

$$T = 1,1 \times (R3+R7) \times C1$$

Essendo R7 un trimmer, l'intervallo e quindi la durata dell'impulso uscente dal piedino 3 può essere variato a piacimento entro certi limiti, che sono:

$$T_{min} = 1,1 \times 10k\Omega \times 100 \mu F = 1,1 \text{ sec.}$$

se R7 è tutto cortocircuitato, ovvero:

$$T_{max} = 1,1 \times 1,01M\Omega \times 100 \mu F = 110 \text{ sec.}$$

con il trimmer completamente inserito. In realtà, questa è la formula suggerita dal costruttore senza alcuna correzione delle tensioni fornite dal partitore multiplo interno al 555; invece nel nostro caso la tensione dell'ingresso invertente del comparatore viene alterata in un caso, ossia nel funzionamento monostabile (per l'esattezza, i tempi sono allungati). E qui entra in gioco il doppio switch SW1, del quale una sezione agisce sul pin 5 (il Control Voltage) del timer 555 e l'altra preleva il segnale d'uscita dall'U1 nel caso desideriamo il funzionamento impulsivo (monostabile, posizione MONO) o dal flip-flop se la modalità desiderata è quella bistabile (BIST.); rimanendo un istante sul 555, in quest'ultima modalità il piedino 5 termina su un condensatore, come nello schema canonico, mentre in posizione MONO riceve il potenziale di un partitore resistivo esterno formato da R5 ed R6. Se questo è connesso all'U1, ne altera i potenziali di riferimento dei comparatori interni, al punto che le soglie di commutazione sono leggermente innalzate, tanto che nella formula del monostabile si può sostituire il fattore 1,1 con 1,4. Ciò vuol dire che gli impulsi forniti dal 555 saranno in realtà più lunghi di quanto calcolato teoricamente. Comunque sia, se con l'SW1 scegliamo il funzionamento monostabile, il relé sarà comandato dall'impulso prodotto dal timer 555 ad ogni rilevamento di un battito di mani.

Se invece spostiamo il doppio deviatore su BIST entrerà in gioco il flip-flop U2, che, triggerato dagli impulsi forniti dal monostabile, alternerà lo stato logico alle proprie uscite Q e /Q attivando stabilmente il relé e mantenendolo a riposo alternativamente. Peraltro va notato che nella modalità bistabile avremo all'uscita dell'U1 impulsi più brevi, dato che la loro durata sarà determinata solo dal partitore interno al 555.

Vediamo dunque cosa accade inserendo U2 (uno dei due flip-flop di tipo D contenuti nell'integrato 4013 montato nel circuito, secondo lo schema a blocchi proposto nella **Fig. 2**) fra l'uscita dell'U1 e la base del transistor T2: ogni impulso, ovvero ciascun livello logico alto al piedino 3 dell'U1, triggera l'ingresso di clock (C) del flip-flop, che nel nostro circuito abbiamo connesso in modo latch; questo tipo di configurazione implica che l'uscita complementata (/Q) comandi l'ingresso D (Data, piedino 5) e questo permette di ottenere dal flip-flop proprio il funzionamento descritto. In pratica, partendo dalla condizione iniziale in cui Q è a zero logico e /Q ad 1, al primo impulso ricevuto dal clock l'uscita Q passa a livello alto. Questo perché la caratteristica del flip-flop D-type è che ad ogni impulso positivo sulla linea di clock il Q presenti lo stesso stato logico dell'ingresso D e, di conseguenza, /Q si ponga al livello logico opposto. Al successivo impulso, giacché ora Q è ad 1 e /Q a zero, l'uscita Q torna a livello basso e la Q ad 1. Andando avanti a fornire impulsi, il ciclo si ripete fino a quando il piedino di clock viene eccitato, vedendo l'avvicinarsi degli stati logici 1 e 0 alle uscite (piedini 1 e 2) del flip-flop.

Chiaramente nella modalità bistabile bisognerà scegliere durate dell'impulso inferiori, tipicamente, a quelle della bistabile, altrimenti se in quest'ultima modalità desiderassimo attivazioni e disattivazioni molto ravvicinate, non riusciremmo nell'intento, in quanto dovremmo attendere il ritorno a livello basso del piedino 3 del 555.

Giunti a questo punto, abbiamo visto le vicende del 555 e non ci resta che analizzare la parte di uscita, ovvero l'attuatore di cui il circuito si serve per commutare l'alimentazione sul carico; come accennato, questo blocco viene collegato all'uscita del 555 o a quella del flip-flop a seconda che si voglia, rispettivamente, sfruttare la modalità monostabile (uscita temporizzata) o quella bistabile (uscita on/off alternata) e tale selezione viene operata tramite il doppio deviatore SW1.

Quale che sia la fonte del segnale di comando, esso pilota la base del transistor T1 attraverso

la resistenza di polarizzazione R4, mandando in saturazione il BJT quando si trova a livello alto e pertanto per tutta la durata della tensione di comando presente sul cursore del doppio deviatore. Pertanto, quando sulla resistenza R4 è presente il livello logico alto, il collettore del transistor alimenta la bobina del relé RL1 facendone scattare l'equipaggio mobile e chiudendo il contatto centrale (C) sul normalmente aperto (NO); al ritorno a zero volt (o comunque al disotto della tensione V_{be} di soglia) il T1 torna interdetto, priva la bobina del relé dell'alimentazione e lo scambio ritorna a riposo, chiudendo C su NA.

Siccome la bobina del relé, quando viene privata della tensione tende a reagire con un'extratensione inversa (questo fenomeno si deve alla sua natura induttiva) che danneggerebbe la giunzione base-collettore del transistor T1, tra i suoi capi (ossia tra collettore del transistor e positivo di alimentazione) abbiamo applicato un diodo con funzione di soppressione di tale picco, collegato in antiparallelo, vale a dire con il catodo sul positivo; in tal modo in condizioni normali il D1 è interdetto e conduce solamente quando si presenta una componente di polarità inversa.

Il circuito si alimenta in corrente continua attraverso la morsettiera facente capo a +12V e GND.

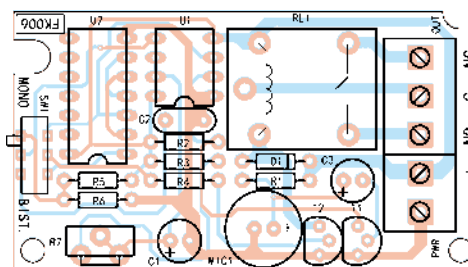
REALIZZAZIONE PRATICA

Bene, spiegato come funziona il nostro timer start/stop passiamo adesso alle note costruttive, partendo dalla realizzazione del circuito stampato, che è un singola faccia (quindi realizzabile da tutti con facilità) e soprattutto basato esclusivamente su componenti a montaggio tradizionale, a tutto vantaggio di chi, per manualità e attrezzatura disponibile, non riuscirebbe a montare un circuito in SMD.

Il PCB occorrente al progetto va ottenuto per fotoincisione seguendo la traccia lato rame scaricabile dal nostro sito www.elettronica.in.it; inciso e forato il c.s. dovete iniziare il montaggio dei componenti con le resistenze, il diodo al silicio D1 e i condensatori non polarizzati, quindi procedendo con gli integrati 555 e 4013, che potete montare su uno zoccolo dip 4+4 contatti (in tal caso inserirete gli integrati una volta che avrete saldato tutti gli altri componenti). Passate quindi al transistor NPN, quindi procedete con le morsettiere a passo 5 mm, i condensatori elettrolitici (C1 e C3) ed il relé, che entrerà solo nel verso giusto.

La capsula microfonica electret può essere montata sul circuito stampato oppure collegata alle





Elenco Componenti:

R2: 1 kohm
R1, R3, R6: 10 kohm
R4: 15 kohm
R5: 4,7 kohm
R7: Trimmer 1 Mohm MV
C1, C3: 100 μ F 16 VL
elettrolitico

C2: 100 nF 63 VL poliestere
D1: 1N4148
T1, T2: BC547
U1: NE555P
U2: CD4013BE
RL1: Relé 12V/10A 1 scambio
MIC1: Capsula micorofonica
preamplificata $\varnothing$ 10 mm

Varie:

- Deviatore slitta 2 vie
- Morsetto 2 poli passo 5 mm
- Morsetto 3 poli passo 5 mm
- Zoccolo 4+4
- Zoccolo 7+7
- Porta batteria 9V
- Circuito stampato FK006 (62x40 mm)

rispettive piazzole mediante uno spezzone di cavo schermato coassiale, avendo cura di utilizzare il filo centrale per il segnale e la calza di schermo per la massa; riguardo al microfono, rammentate che di norma il polo negativo si riconosce perché è quello elettricamente connesso al contenitore. La scelta della collocazione della capsula electret si fa in base al tipo di installazione desiderata, ovvero se volete che il battito di mani o altro rumore di eccitazione sia rilevato vicino o lontano da dove si trova il cablaggio del carico.

Per l'orientamento dei componenti polarizzati fate riferimento al disegno di montaggio visibile in alto questa pagina.

UTILIZZO DEL CLAP SWITCH

Bene, ora che avete montato il circuito potete pensare all'installazione e all'utilizzo pratico, giacché il circuito è subito utilizzabile e non richiede regolazioni preliminari, salvo quella del trimmer che imposta il tempo dell'impulso monostabile (o comunque di trigger del flip-flop) e la selezione, tramite il doppio deviatore, della modalità di funzionamento.

Per l'installazione rammentate che lo scambio del relé può essere utilizzato per interrompere uno dei fili (se trattasi di corrente alternata, raccomandiamo quello della fase, da identificare preventivamente mediante un cacciaviti cercafase) di alimentazione dell'utilizzatore. Ricordate i limiti di tensione e corrente commutabile indicati nelle caratteristiche tecniche esposte a inizio articolo. Per l'utilizzo ricordiamo che in modalità monostabile bisogna spostare il deviatore SW1 in posizione "MONO" e si potrà quindi agire sul trimmer per im-

postare il tempo trascorso il quale il relé ritornerà a riposo. Il battito di mani commuterà l'uscita, dopo di che allo scadere del tempo impostato l'uscita tornerà a riposo.

Il tempo di attivazione varia da 1s a 110s.

In modalità bistabile bisogna spostare il deviatore SW1 in posizione "BIST" e il trimmer R5 dovrà essere girato completamente in senso antiorario, in modo da portare la resistenza al minimo possibile: ciò, allo scopo di avere un periodo molto corto e poter quindi impartire più comandi ravvicinati, giacché va considerato che la distanza temporale minima tra un battito di mani e il successivo non può essere inferiore alla durata dell'impulso prodotto dal monostabile implementato dal 555. Ricordate che in modalità bistabile un battito di mani commuterà l'uscita e un successivo battito di mani la porterà a riposo. □



Cosa occorre?

Il progetto presentato in questo articolo è disponibile presso Futura Elettronica con il codice FK006 a Euro 14,00. Chi volesse realizzare in autonomia il circuito stampato del progetto può utilizzare il servizio di produzione PCB (cod. PCBPRODUCTION) presente sul sito di Futura Elettronica. 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>

Costruisci un vero robot!

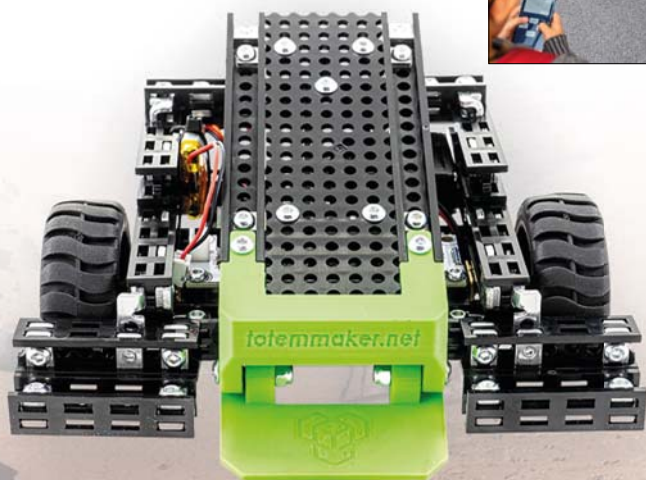


Mini robot da combattimento controllato da App

Minitrooper è un kit "Totem STEAM Toys" che permette di realizzare un piccolo robot da combattimento con catapulta e controllabile attraverso App (Totemmaker) per smartphone. Minitrooper può essere impiegato in vari giochi come: sumo, football, corse, battaglie tra robot, ecc. La catapulta può essere utilizzata per ribaltare gli avversari in piccoli combattimenti tra due o più robot, oppure per spingere una pallina nel gioco del football, ecc.

€ 79,⁵⁰

Cod. MINITROOPER



**CONTROLLABILI
TRAMITE
APP!**

Rover con braccio robotico e pinza controllato da App

Gripper Bot è un kit "Totem STEAM Toys" che permette di realizzare un piccolo rover con braccio robotico e pinza, controllabile attraverso App (Totemmaker) per smartphone. La pinza, montata sul braccio robotico, consente di afferrare e sollevare piccoli oggetti (max. 60 grammi).



€ 69,⁰⁰

Cod. GRIPPERBOT

Assemblando i robot imparerai:

- > Nozioni base di meccanica, elettricità e programmazione.
- > Soluzioni ingegneristiche semplici.

- > Utilizzo di diverse parti meccaniche e strumenti.
- > Utilizzo di parti meccaniche in diversi modelli/situazioni.

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

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.

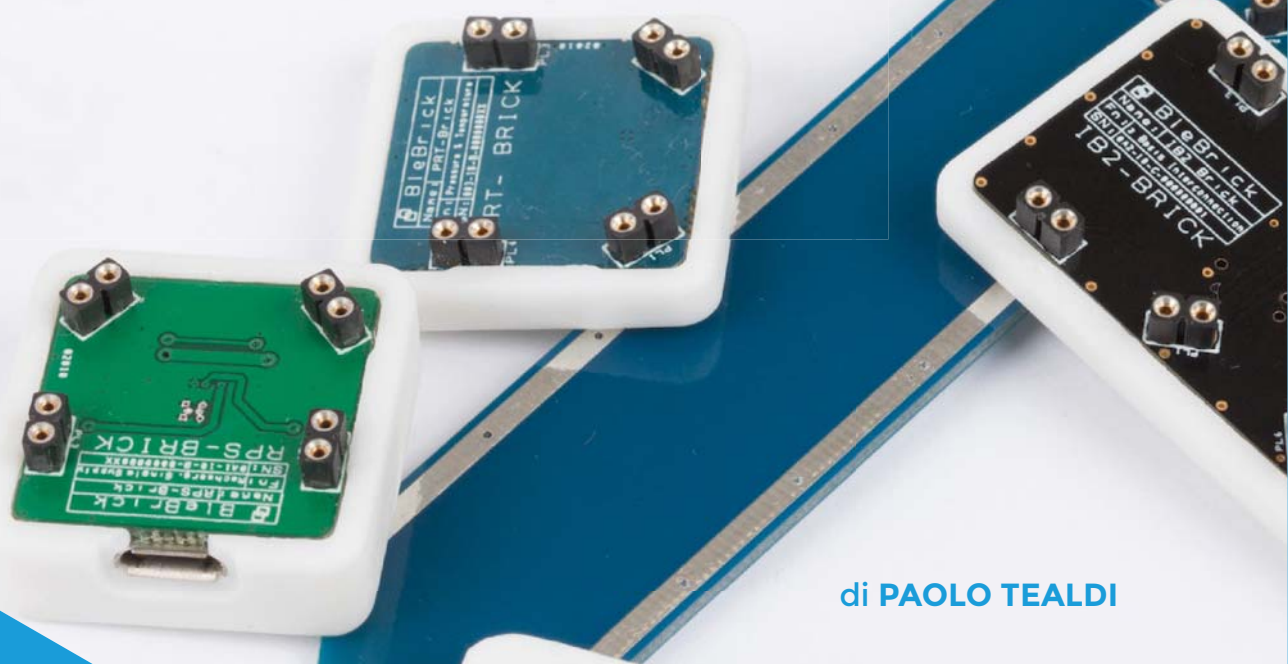
HDMI
↓
AV

€ 11,00

Cod. **BOXHDMITOAV**

Blebricks

LE IDEE PRENDONO FORMA



di **PAOLO TEALDI**

Soluzione modulare con la quale creare in tempo zero il nostro dispositivo hardware IoT programmabile, interattivo e gestibile direttamente da apposita APP.

Termini come Smart Tag, RFID, Beacon non sono certo nuovi ai nostri lettori; si tratta di ottime soluzioni tecnologiche, che però implicano una curva di apprendimento molto ripida nelle fasi iniziali, andando a pesare sui tempi necessari allo sviluppo delle applicazioni, tanto che il divario temporale tra progetto e prototipo funzionante non è certamente trascurabile, senza contare il bagaglio di conoscenze richieste come prerequisiti per iniziare; e questo vale anche quando metà del lavoro è già fatto, come è il caso delle più note piattaforme di prototipazione. In questo scenario si colloca una soluzione rapida, efficace, economica, di facile implementazione (in termini hardware e software), impiego e utilizzo, professionale, per progetti in cui sia necessaria l'interazione fra una serie di oggetti fisici e lo smartphone degli utenti: si tratta dei Blebricks di Bleb

Technology, una piattaforma made in Italy per la prototipazione di applicazioni Internet of Things, nata in ambito B2B per applicazioni industriali ma che “strizza l’occhio” anche ai maker e agli studenti. I Blebricks sono stati brevettati come soluzione di prototipazione istantanea, in quanto subito funzionanti nel momento in cui si incastrano i mattoncini. Questa modularità risolve molti dei problemi insiti nello sviluppo di applicazioni altamente tecnologiche, permettendo al progettista di concentrarsi sulla propria idea, senza doversi cimentare nello sviluppo del codice per disporre delle singole tecnologie occorrenti. Detto in breve, i Blebricks (mattoncini Bluetooth Low Energy, ma non solo...) rappresentano un modo di maneggiare con facilità l’elettronica moderna senza fare troppa fatica, il che da un lato permette di accelerare lo sviluppo ma dall’altro indurrebbe a pensare, erroneamente, che non serva conoscere l’elettronica; vediamo così: sono un modo per prototipare in fretta applicazioni “connesse” senza la fretta di imparare subito tutta la teoria di elettronica e programmazione che occorrerebbe a implementarle in maniera tradizionale.

I Blebricks racchiudono concetti quali prototipazione istantanea, RFID, beacon, Bluetooth, MIT App Inventor, modularità, sensoristica, IoT, coprendo le esigenze di molti utenti. La Bleb Technology (www.bleb.it) creatrice dei Blebricks, è un’azienda affermata nel campo delle soluzioni IoT e non solo, che, oltre a lavorare a stretto contatto con importanti aziende per lo sviluppo di nuove soluzioni in ambito IoT, collabora anche con istituti scolastici superiori nei test, ricevendo riconoscimenti a livello nazionale e internazionale. Progetti realizzati con i Blebricks da studenti sono stati presentati e premiati ad esempio al MIT App Inventor Summit 2019 in USA, all’IPRED VI a Tel Aviv, al Focus Live a Milano, al Fiera Didacta a Firenze e all’EGU General Assembly 2021 e 2022 a Vienna.

COSA SONO E A COSA SERVONO I BLEBRICKS

I Blebricks sono dispositivi hardware che permettono la prototipazione istantanea di sistemi basati su sensori/attuatori, in ambito IoT (Internet of Things). I Blebricks sono mattoncini intelligenti assemblabili a incastro, quindi senza necessità di cavi o saldature, per creare svariati sistemi elettronici, dotati di sensori ed attuatori, capaci di comunicare con lo smartphone/tablet e con

Internet; le connessioni vengono garantite da specifici connettori opportunamente posizionati sui PCB dei moduli. Con i Blebricks è possibile comporre l’hardware dell’applicazione desiderata, senza avere competenze specifiche o dover utilizzare attrezzature/strumenti di laboratorio, nell’ambito dell’IoT, degli Smart Sensors o RFID e delle soluzioni connesse.

I Blebricks presentano consumi ridotti (sono definiti dispositivi Ultra low power) e costituiscono una tecnologia adatta per creare dispositivi di ridotte dimensioni, indossabili, portatili e con lunga autonomia se alimentati a batterie. Inoltre, permettono una soluzione di continuità dal prototipo al prodotto finito industrializzato secondo le proprie esigenze per produzione in serie.

Consentono altresì un abbattimento delle barriere tecnologiche, nonché una riduzione di tempi e costi di sviluppo del 90% dall’idea al prodotto finito.

TIPOLOGIE DI BLEBRICKS

Per creare il nostro prototipo dobbiamo incastrare fra di loro diversi Blebricks, appartenenti almeno a due distinte tipologie, dal momento che ognuno di essi implementa una specifica funzione.

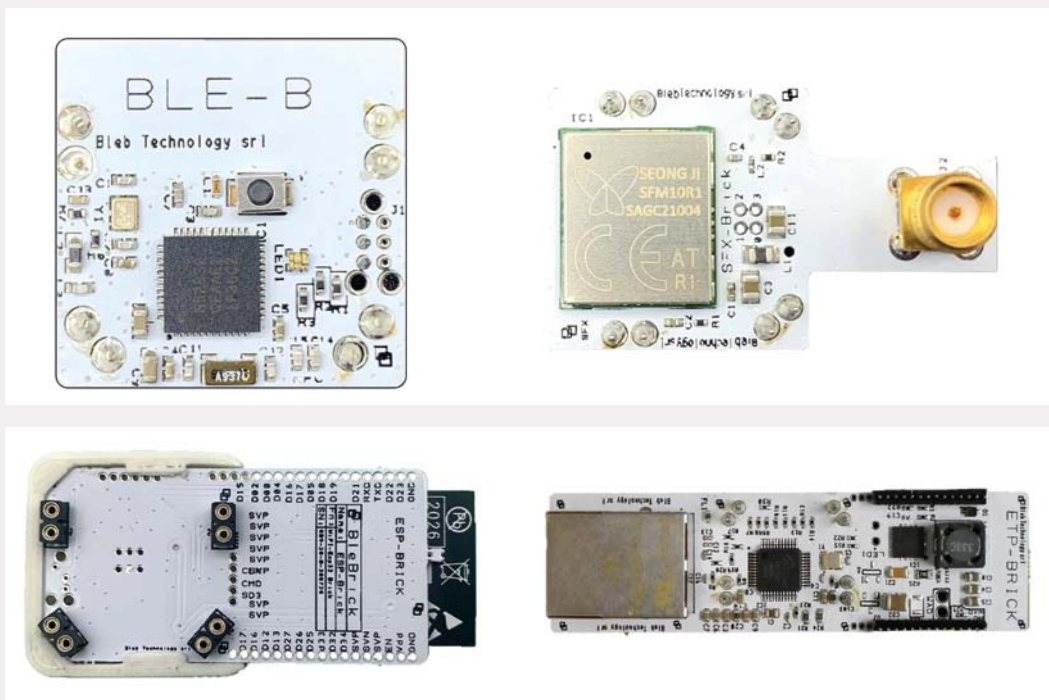
Di mese in mese vengono inoltre messi a catalogo nuovi modelli. Il mattoncino alla base di tutto il sistema è il BLE-B che comunica nativamente via BLE (Bluetooth Low Energy) con l’ambiente circostante e costituisce il cuore di questa soluzione tecnologica.

Ad esso si deve poi aggiungere un mattoncino di alimentazione ed eventualmente altri, ognuno dei quali implementa funzionalità diverse.

La configurazione minimale prevede un BLE-B e un modulo di alimentazione (ad esempio Blebrick RPS o Blebrick SPS).

Di seguito elenchiamo le varie categorie di mattoncini Blebrick.

- **Blebricks di comunicazione**, che permettono di aggiungere funzioni di comunicazione quali WiFi, Ethernet, Infrarosso e LP-WAN (Sigfox, NB-IoT e LoRa); per esempio BLE-B, Blebrick SFX, Blebrick ESP, Blebrick ETP che vedete nelle **Fig. 1** e **Fig. 2**.
- **Blebricks sensori**, per aggiungere sensori inerziali (es. accelerazione, orientamento, campo magnetico, ecc.), ambientali: ad esempio temperatura, pressione, umidità, CO₂, particolato, qualità dell’aria, ozono, vento, pioggia, umidità

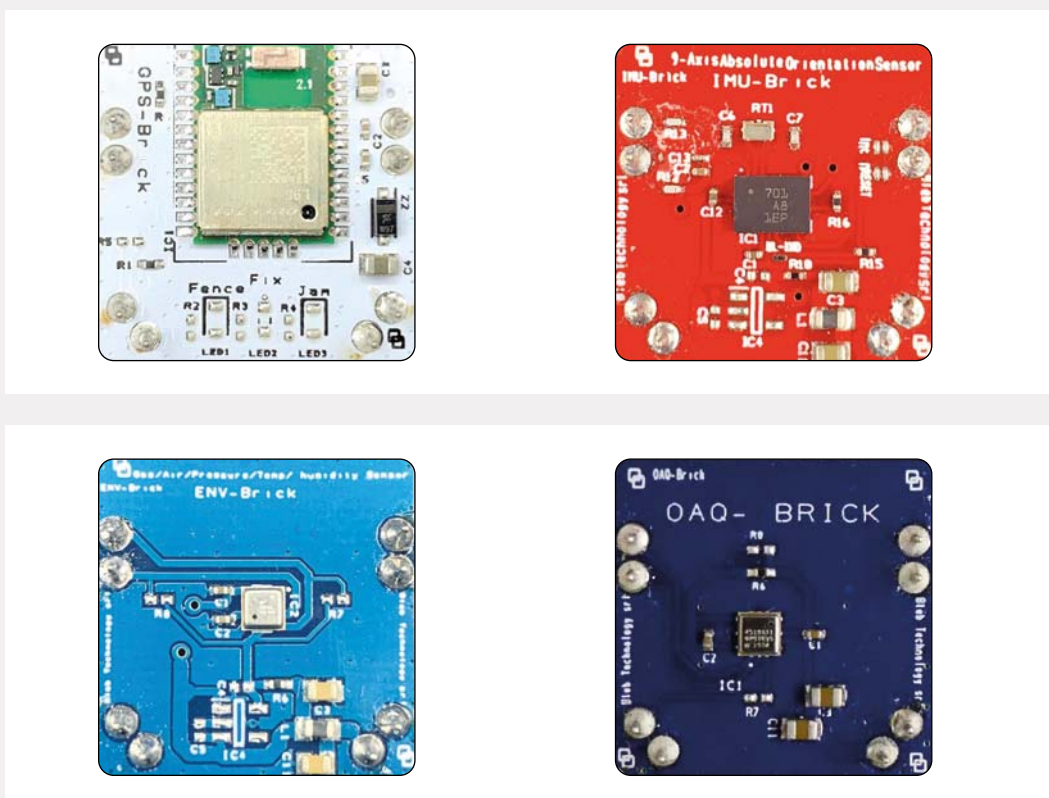


← **Fig. 1**
I moduli
Blebricks
BLE-B (a
sinistra) e
Blebrick SFX
(a destra).

← **Fig. 2**
I moduli
Blebricks
ESP (sinistra)
e Blebrick
ETP (destra).

del terreno, posizionamento globale (GPS), piattaforma inerziale (IMU) ed altro che si renderà disponibile. Sono esempi quelli mostrati nella **Fig. 3** e nella **Fig. 4**.

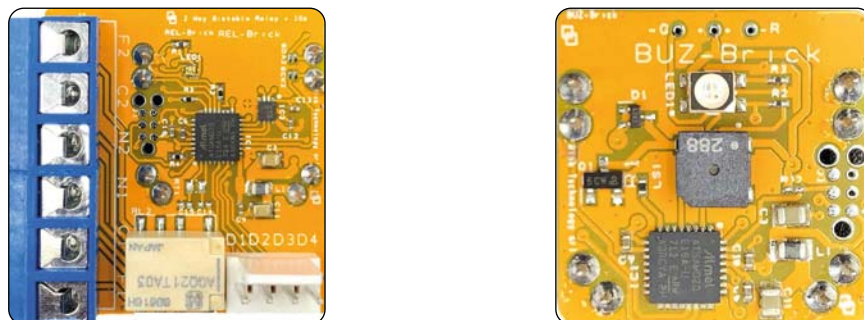
- **Blebricks attuatori**, per aggiungere ad esempio funzioni di allarme acustico, vibrazione, LED ad alta intensità, relé ecc.
Ne vedete due nelle foto in **Fig. 5**.



← **Fig. 3**
I moduli
Blebrick GPS
(sinistra) e
Blebrick IMU
(destra).

← **Fig. 4**
I moduli
Blebrick ENV
(a sinistra) e
Blebrick OAQ
(a destra).

→ **Fig. 5**
I moduli attuatori
Blebrick REL (a
sinistra) e Blebrick
BUZ (a destra).



→ **Fig. 6**
I moduli Blebrick
RPS (per batteria
ricaricabile, a
sinistra) e Blebrick
BPS (a destra).



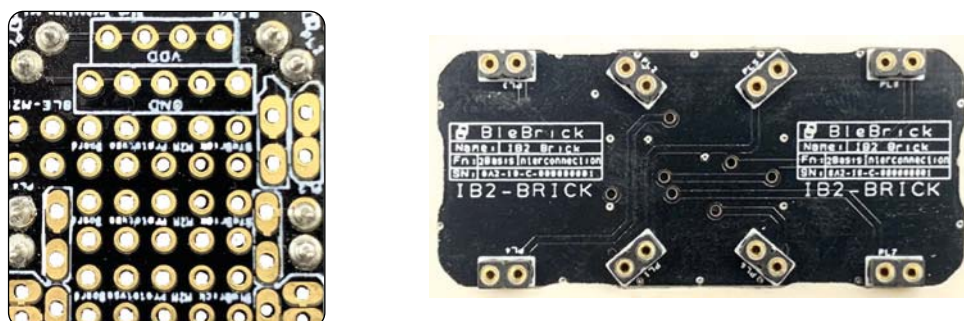
- **Blebricks di alimentazione**, per scegliere l'alimentazione preferita tra batteria ricaricabile, non ricaricabile, a pila, a bottone, pannello solare, tensione continua 5-24V cc, uscita USB. La **Fig. 6** propone i moduli Blebrick RPS e Blebrick BPS.
- **Blebricks di interconnessione e montaggio**, fatti per interconnettere orizzontalmente più mattoncini (così come si fa con i famosi mattoncini giocattolo) allo scopo di realizzare collegamenti cablati, soluzioni di impatto visivo ecc. Vedete questi elementi, ossia il modulo Blebrick PCB e il Blebrick IB2, nella **Fig. 7**.

Pertanto, per aggiungere le funzionalità desiderate al nostro prototipo, è sufficiente incastrare i Blebricks tra loro. Avvenuto ciò, essi si configurano automaticamente e sanno già come funzionare insieme: il nostro dispositivo prototipale è subito pronto all'uso!

PERCHÉ BLEBRICKS

La natura modulare dei Blebricks permette di creare istantaneamente un prototipo funzionante. E questo in definitiva è più che sufficiente in ambito Educational piuttosto che Maker. Dobbiamo ricordare però che partendo proprio

→ **Fig. 7**
I moduli Blebrick
PCB (a sinistra)
e Blebrick IB2
(a destra).



da questo prototipo è possibile volendo creare un proprio prodotto finito personalizzato ed industrializzato ad hoc, in modo semplice, rapido ed economico. Ovviamente il dispositivo industrializzato, oltre a essere personalizzato nella forma e marchiato con un proprio logo, potrà essere più performante, più compatto e risulterà apprezzabilmente più economico rispetto al prototipo.

E così finalmente, dopo pochissimi istanti, abbiamo in mano il nostro prototipo, il nostro dispositivo elettronico. Non ci resta che provarlo, per testare sul campo le sue potenzialità.

Ci sono diverse modalità, a seconda delle esigenze specifiche: vediamole.

Siamo impazienti e non abbiamo particolari esigenze di personalizzazione del software?

La **app MakeApp**, disponibile gratuitamente sul Play Store per smartphone e tablet con sistema operativo Android, è sicuramente quello che fa per noi. Una volta scaricata, installata e lanciata dal nostro dispositivo mobile (ricordatevi prima di attivare su di esso il Bluetooth e il Sensore di Posizione), potete visualizzare subito i dati dei sensori, interagire con i Blebricks e configurare il loro funzionamento con semplice logica IFTTT (IFTTT è un acronimo per If This Then That, ovvero se questo accade allora fai accadere quello).

Vogliamo invece creare un'app tutta nostra personalizzata? **MIT App Inventor e l'estensione gratuita per i Blebricks** fanno al caso nostro: in modo semplice e intuitivo con il coding potremo in pochi minuti ottenere il risultato desiderato. Ricordiamo, a questo proposito, che abbiamo pubblicato in passato un corso dedicato a MIT App Inventor in 6 puntate, sui numeri 224-225-226-227-228-229 di Elettronica In. Invece per i programmatori esperti di software, con le **librerie Java, le API e librerie PHP** è anche possibile realizzare la propria applicazione.

QUALI APPLICAZIONI CREARE CON BLEBRICKS?

Fornire un elenco esaustivo non è affatto facile, in generale però osservando quelle create da maker, professionisti, studenti, insegnanti o semplici hobbisti, è possibile dividere le applicazioni in tre grandi tipologie ovvero Professionali (applicazioni IoT per prevenzione e controllo ambientale, sicurezza sul lavoro, per il turismo, per l'agricoltura...), Personali (ritrovare gli oggetti, SOS anziani, campanello wireless, qualità dell'aria, annaffiare una pianta...), Educational (caccia al tesoro, Bleb Invaders, CO₂ in scatola...).

FUNZIONALITÀ DATA LOGGER

Il data logger è la funzionalità di salvataggio dei dati dei sensori Blebricks nella memoria del BLE-B (con periodo impostabile dall'utente). Questa funzionalità è utile, ad esempio, quando non c'è garanzia di copertura di rete; ad esempio se siamo in treno oppure in montagna.

I dati registrati dal BLE-B possono essere riletti dalla app MakeApp (per dispositivi Android, disponibile gratuitamente sul Play Store di Google), semplicemente accedendo alla funzionalità data logger come indicato nelle **Fig. 8, Fig. 9 e Fig. 9**.

In pratica avviamo la app MakeApp, clicchiamo sull'icona del BLE-B per accedere alle impostazioni e mettiamo il flag su data logger, quindi, nella schermata principale della MakeApp ci comparirà la sezione data logger: premiamo il pulsante "Start logging" (**Fig. 9**) per far partire la registrazione dei dati. Muovendo l'apposito cursore a video "Select the sampling interval" possiamo a questo punto selezionare l'intervallo (la frequenza, in pratica) del campionamento dei dati da porre nel log e premiamo quindi OK. Fatto ciò, il data logging inizia e a questo punto, mediante i pulsanti posti in basso nello scher-



Fig. 8
Avviamo la app MakeApp, clicchiamo sull'icona del BLE-B per accedere alle impostazioni e mettiamo il flag su data logger.

Per prima cosa andiamo sul Play Store di Google, cerchiamo la app denominata "MakeApp for Blebricks", installiamola e attiviamo Bluetooth e Info posizione (GPS) sul nostro smartphone. Verifichiamo inoltre che la MakeApp abbia i permessi di localizzazione GPS per permettere il funzionamento della tecnologia Bluetooth Low Energy. A questo punto possiamo avviare la app MakeApp.

Vicino a noi teniamo alcuni Blebricks opportunamente incastrati e alimentati (la configurazione minimale prevede un BLE-B e un modulo di alimentazione come Blebrick RPS o Blebrick SPS). Sul nostro smartphone apparirà la videata principale che mostra le risorse del nostro dispositivo.

Proviamo ad aggiungere qualche Blebrick al nostro BLEB-B per vedere aumentare la lista. Le Direct Interactions consentono ai Blebricks di interagire tra di loro o con il nostro smartphone, autonomamente.

È sufficiente istruirli sulle azioni che devono compiere in base a certi eventi, in una logica tipo "se questo succede, allora fai questo" (es. se Temperatura > 30° C, allora Accendi led spia).

Il BLE-B è il modulo di comunicazione di base Bluetooth Low Energy e nella schermata principale della app MakeApp, in alto accanto all'icona, sono indicate le sue informazioni generali: nome, distanza stimata, RSSI, livello della batteria e indirizzo Bluetooth.

Per accedere alla pagina delle impostazioni è sufficiente cliccare sull'icona.

Oltre a gestire tutti i Blebricks connessi, il BLE-B implementa anche funzioni aggiuntive quali temperatura misurata, feedback del suo pulsante fisico, messaggio di testo memorizzato, led e GPIO e relative impostazioni.

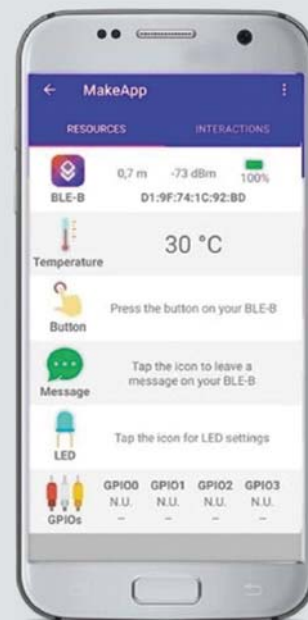


Fig. 9

Nella schermata principale della MakeApp ci comparirà la sezione data logger: premiamo "Start logging" per far partire la registrazione dei dati.

Fig. 10

Data logging avviato: notate in basso i pulsanti per fermare, annullare o scaricare i dati registrati.



mo, possiamo fermare (STOP, quadratino nero) annullare (X in rosso) o scaricare (freccia verde in giù) i dati registrati che saranno salvati in formato **.csv** nella cartella MakeApp del nostro smartphone (**Fig. 10**).

CONCLUSIONI

Con questo articolo abbiamo voluto rendervi partecipi della nostra personale esperienza nel mondo della prototipazione istantanea grazie alla tecnologia italiana Blebricks, mattoncini intelligenti IoT prodotti dalla Bleb Technology. Vi sarete accorti di come sia semplice ed intuitivo far prendere forma in pochi attimi alle vostre idee. Questo ovviamente è solo un piccolo assaggio, tanto da stuzzicare la vostra curiosità e fantasia creativa. Nelle prossime puntate vi racconteremo nel dettaglio alcuni progetti nati per venire incontro a reali esigenze quotidiane, realizzati nel corso di questi anni da maker e studenti. □

Cosa occorre?

I moduli Blebricks sono disponibili in comodi starter-kit o anche singolarmente. In particolare lo starter-kit base (cod. BLEBSTARTERKIT) è in vendita a Euro 41,50. 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.futuranet.it

SCEGLI LA TUA STAMPANTE 3D

Affidati a chi la progetta e la sviluppa

Stampante 3DFDM (Fused deposition modeling) in KIT
capace di produrre stampe di 40x40x40 cm (64.000 cm³)



- Struttura interamente in alluminio
- Area di stampa: X 40 cm, Y 40 cm, Z 40 cm
- Diametro filamento: 1,75 mm
- Tipo di filamento: ABS, PLA, NYLON ed altri ancora
- Diametro ugello fornito: 0,4 mm
- Diametri ugelli opzionali: da 0,3 mm a 0,8 mm
- Velocità di stampa massima: 300 mm/s (in funzione dell'oggetto da stampare)
- Piatto di stampa fisso: vetro temperato da 6 mm

- Riscaldatore per piatto di stampa: 40 x 40 cm – 12V/240 W con adesivo 3M (opzionale)
- Controllabile da PC o da modulo LCD (opzionale)
- Alimentazione tramite modulo switching 220 VAC/12 VDC 350 W
- Istruzioni di montaggio in italiano con illustrazioni

VERSIONE MONTATA

€ 1.299,00

IVA inclusa



€999,00

Cod. 3D4040



Macchina 3 in 1 in KIT:

- Stampante 3D
- Incisore laser
- Fresatrice CNC

€899,00

Cod. CRCP3IN1

- Telaio in profilati di alluminio V-slot da 40 x 40 mm
- Scocca in alluminio a struttura aperta
- 5 motori passo-passo Nema17 (200 step)
- Spostamento asse Z: doppia vite trapezia Ø8 mm a 3 principi in acciaio INOX
- Struttura con profilati in alluminio 40 x 40 mm

- Schermo grafico touch da 4,3"
- Firmware Marlin open source
- Connettività SD – USB
- Alimentazione 200-240 V / 3.4 A
- Output 24VDC - 270 W
- Dimensioni telaio (xyz): 400 x 450 x 650 mm
- Peso 10 Kg

Stampante 3D per stampe fino a 22 x 22 x 25 cm in KIT

- Tecnologia FFF FDM
- Diametro filamento 1,75 mm
- Volume di stampa (xyz) 220 x 220 x 250 mm
- Dimensione Nozzle 0,4 mm
- Estrusore MK10
- 4 motori passo-passo Nema17 (200 step)
- Spostamento asse Z con vite trapezia Ø8 mm a 3 principi in acciaio INOX
- Struttura con profilati in alluminio
- Scocca in alluminio a struttura aperta
- Livellamento del piatto manuale di precisione
- Schermo LCD
- Firmware Marlin open source
- Connettività SD – USB
- Sistema di recupero della stampa
- Slicing consigliato Cura
- Sistemi operativi Windows, Mac OSX, Linux
- Dimensioni telaio (xyz) 440 x 410 x 650 mm
- Peso 5,6 Kg
- Alimentazione 200-240 V / 3.4 A
- Output 24VDC - 270W



€379,00

Cod. CR3DPRO

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.

TELECAMERA FULL COLOR 2 MPX



€ 52,00

cod. FR816N

- ✓ Uscita Video (BNC): 4 in 1 - AHD / CVI / TVI / CVBS
- ✓ Sensore di immagine: 1/2.8" CMOS
- ✓ Risoluzione immagine: 2 MPX
- ✓ Pixel effettivi: 1920(H)*1080(V)
- ✓ Sistema TV: PAL
- ✓ Obiettivo: 3,6 mm
- ✓ Portata Illuminatore: 20-30 metri
- ✓ Grado di protezione: IP67
- ✓ Alimentazione: 12 VDC
- ✓ Consumo: IR OFF <1W; LED ON <3W
- ✓ Temperatura di Lavoro: da -20 C a +50 C
- ✓ Dimensioni: 180x70x59 mm
- ✓ Peso: circa 335 grammi

TELECAMERA FULL COLOR 2 MPX - VARIFOCAL



€ 69,00

cod. FR816VAR

- ✓ Uscita Video (BNC): 4 in 1 - AHD / CVI / TVI / CVBS
- ✓ Sensore di immagine: 1/2.7" CMOS
- ✓ Risoluzione immagine: 2 MPX
- ✓ Pixel effettivi: 1920(H)*1080(V)
- ✓ Sistema TV: PAL
- ✓ Obiettivo: varifocal 2,8-12 mm
- ✓ Otturatore elettronico: Automatico
- ✓ Portata Illuminatore: 40 metri
- ✓ Grado di protezione: IP67
- ✓ Alimentazione: 12 VDC
- ✓ Consumo: IR OFF: <1W; LED ON: 3,6 W
- ✓ Temperatura di Lavoro: da -20 C a +50 C (RH95% Max)
- ✓ Dimensioni corpo telecamera: 138x75x67 mm
- ✓ Peso: circa 408 grammi

TELECAMERA DOME FULL COLOR - 2 MPX

- ✓ Uscita Video (BNC): 4 in 1 - AHD / CVI / TVI / CVBS
- ✓ Sensore di immagine: 1/2.8" CMOS
- ✓ Risoluzione immagine: 2 MPX
- ✓ Pixel effettivi: 1920(H)*1080(V)
- ✓ Sistema TV: PAL
- ✓ Obiettivo: 3,6 mm
- ✓ Otturatore elettronico: Automatico
- ✓ Portata Illuminatore: 20-30 metri
- ✓ Grado di protezione: IP67
- ✓ Alimentazione: 12 VDC
- ✓ Consumo: IR OFF <1W; LED ON <3W
- ✓ Temperatura di Lavoro: da -20 C a +50 C
- ✓ Materiale involucro: plastica
- ✓ Dimensioni: 108 (diametro) x 86 (altezza) mm
- ✓ Peso: circa 210 grammi

€ 44,00

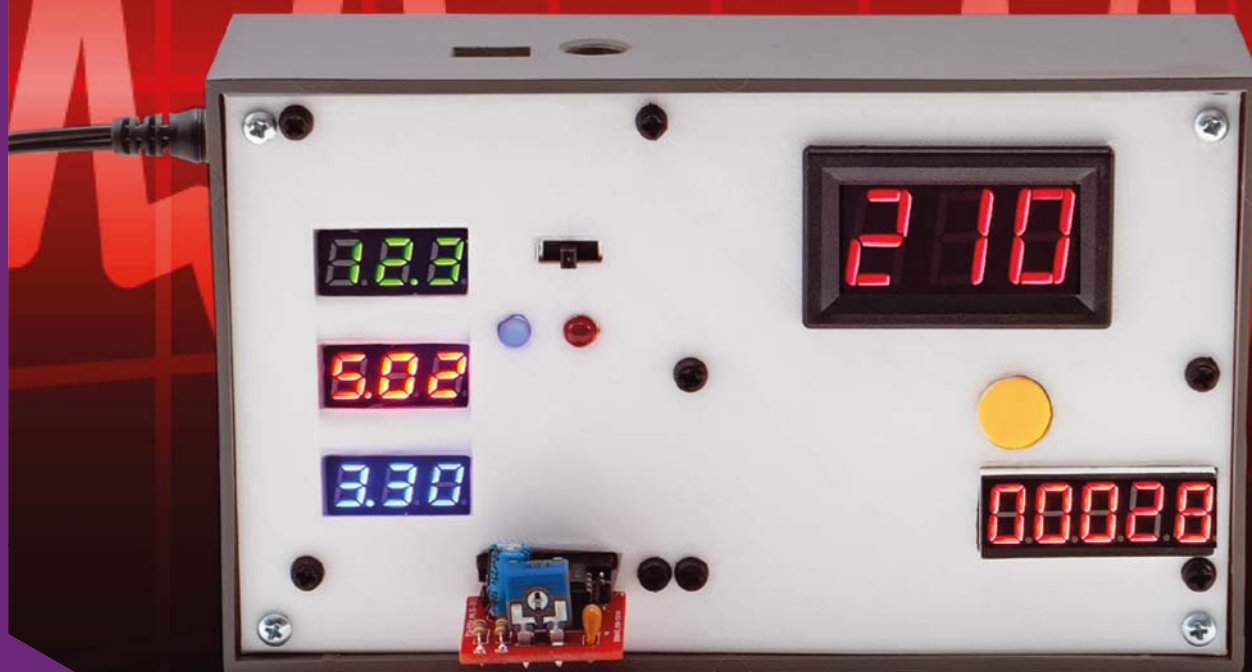
cod. FR817



Trasformano
la notte 
in  giorno!

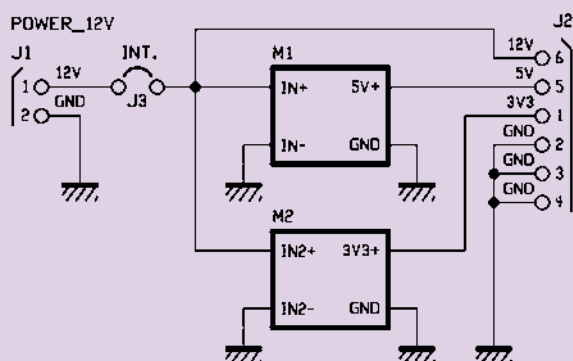
555 ANALYZER

del Prof. **MICHELE MENNITI**



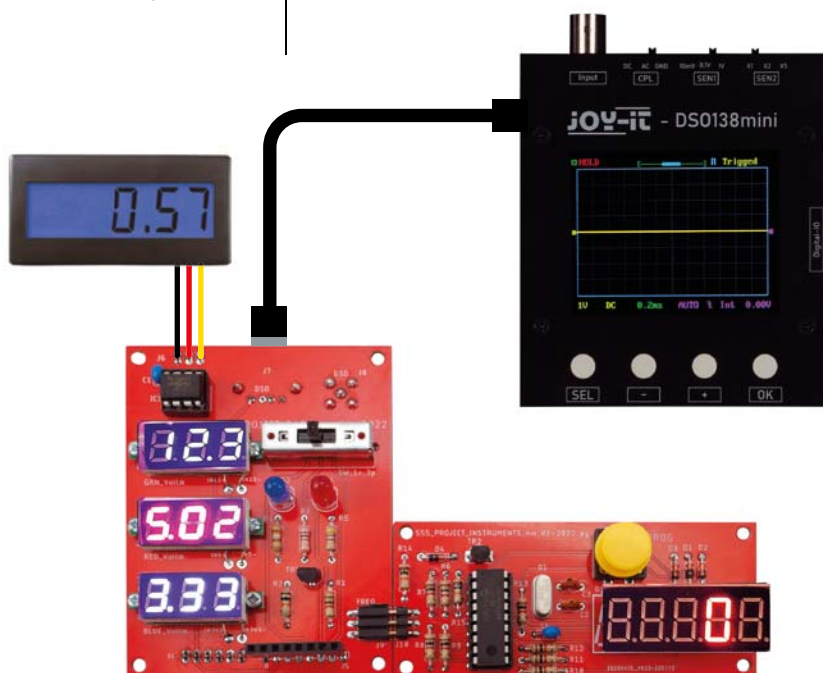
Realizziamo un analizzatore di circuiti basati sul popolare timer 555 e scopriamo come utilizzarlo.
Prima puntata.

Nel numero 255 (giugno 2021) della nostra rivista abbiamo studiato il timer 555, nelle sue tre configurazioni standard, dimostrando quanto esso sia ancora molto utile, pur essendo estremamente datato, tanto che non a caso continua ad essere regolarmente prodotto dai principali costruttori di componenti elettronici. Come abbiamo visto è abbastanza semplice riprodurre le varie configurazioni, adattandole alle nostre esigenze, ma a volte è necessario osservare bene il comportamento di un circuito, per realizzare un'applicazione specifica. Abbiamo quindi pensato di realizzare uno strumento Analizzatore, in grado di leggere e fornire informazioni sullo stato dei quattro pin fondamentali dell'integrato, cioè: SOGLIA, TRIGGER, SCARICA e USCITA. Dei primi tre pin possiamo seguire costantemente l'andamen-



to delle variazioni in tensione, mediante un voltmetro digitale a tre cifre, l'USCITA invece ha diverse interfacce: due LED per la verifica degli stati HIGH e LOW per segnali a bassissima frequenza, un frequenzimetro per la misura del segnale, un mini DSO per il controllo visivo dell'onda quadra, utilissimo ai fini della verifica di parametri come l'ampiezza ed il Duty Cycle. Il circuito da testare è alimentabile a 3,3 V, 5 V o 12 V, tutte tensioni disponibili sul connettore header di test, al quale è possibile collegare sia piccoli PCB, mediante un connettore strip, sia circuiti sperimentali, realizzati su breadboard o piastre millefori, mediante cavetti jumper.

Fig. 1
Prototipo
dell'Analiz-
zatore per
timer 555
completo.



Al fine di poter utilizzare subito il nostro Analizzatore, abbiamo realizzato 12 mini circuiti di test, distribuiti tra le tre configurazioni monostabile, astabile e bistabile, ognuno dei quali sarà illustrato approfonditamente in questo articolo e nelle puntate che seguiranno.

Consigliamo vivamente la lettura dell'articolo sul numero 255, in quanto qui ci limiteremo a riportare solo alcune informazioni essenziali sul 555.

Il nostro Analizzatore è organizzato su tre PCB: alimentazione, base e frequenzimetro, prevede inoltre l'utilizzo di due strumenti esterni: un voltmetro digitale ed un DSO, che può essere di qualsiasi tipo. La **Fig. 1** mostra il nostro prototipo completo.

STADIO ALIMENTATORE

Lo stadio di alimentazione è molto semplice: prevede un ingresso PLUG DC (J1) per un alimentatore switching da 12 V 0,5 A (reperibile sul sito www.futurashop.it con il codice APR-P0012) o di potenza superiore, il cui polo negativo va alla massa comune del circuito, mentre il positivo va ad un pin del connettore J3, che serve per il collegamento esterno di un qualsiasi tipo di interruttore. Nel momento in cui esso sarà posto su ON, i 12 V saranno applicati contemporaneamente agli ingressi di due moduli switching step-down (reperibili sul sito www.futurashop.it col codice MODULODCDC), che saranno stati preventivamente regolati mediante i trimmer di precisione, per erogare rispettivamente le tensioni di 5 V e 3,3 V. Il circuito è completato da un connettore header a sei pin (J2), sul quale vengono riportate la tensione di ingresso di 12 V, le due tensioni aggiuntive di 5 V e 3,3 V e la massa comune.

STADIO BASE

Questo circuito riceve le tre alimentazioni dal connettore header J2 del relativo PCB, mediante il connettore strip a sei pin J1. Tre mini-voltmetri digitali basati su display a LED a tre cifre (reperibili sul sito www.futurashop.it col codice MINIVOLT VERDE-BLU-ROSSO) sono costantemente collegati alle tre alimentazioni, per il controllo dei valori, utile per segnalare un malfunzionamento dei moduli switching o un cortocircuito sull'alimentazione usata. Questi mini voltmetri sono molto interessanti, per la loro caratteristica di non richiedere un'alimentazione esterna, in quanto sfruttano la stessa tensione che misurano.

Un connettore header a otto pin (J2) rappresenta l'interfaccia con i circuiti esterni di test; ad esso sono collegate le tre alimentazioni e la massa

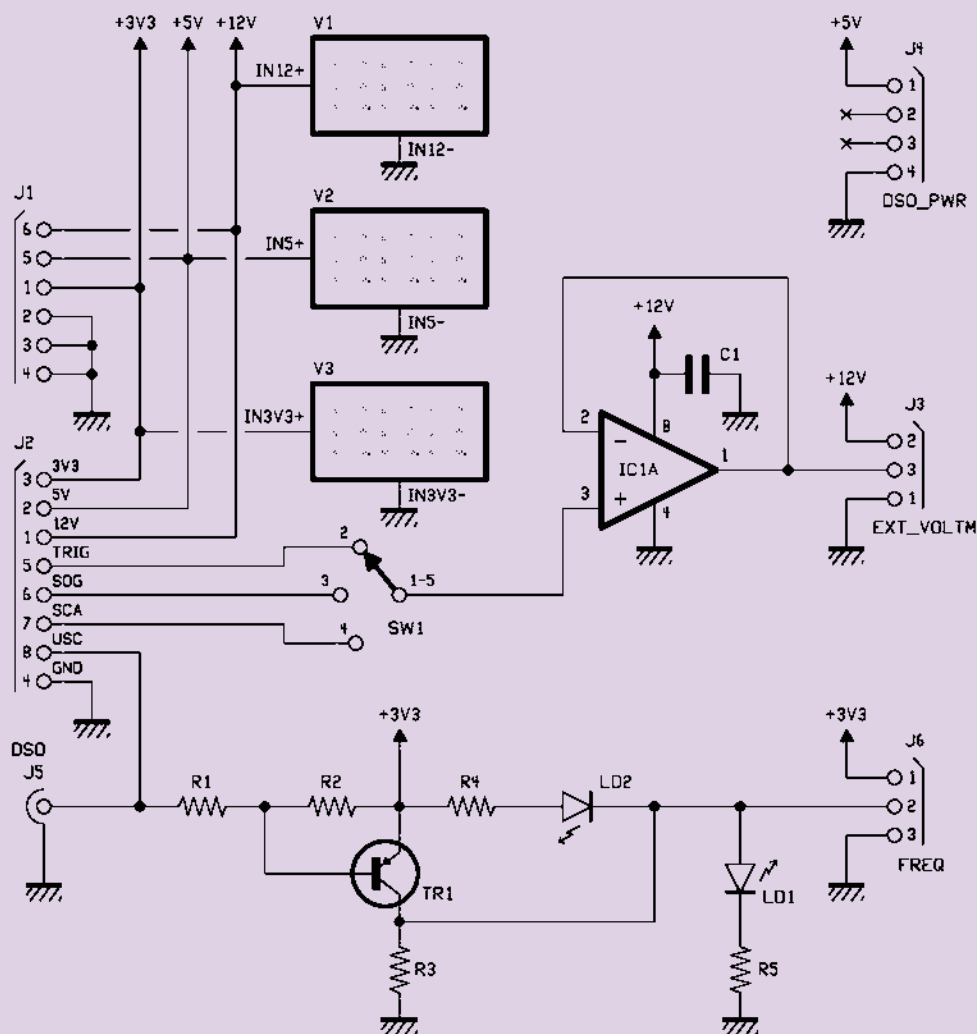
comune, mentre gli altri pin servono per portare al circuito i quattro segnali del 555 sotto test: SOGLIA, TRIGGER, SCARICA e USCITA.

I primi tre vanno direttamente al deviatore a slitta SW1, ad 1 via e 3 posizioni (Alps Alpine SSSU-014800); il comune del commutatore è collegato all'ingresso non invertente di un op amp MCP6002P, in configurazione Voltage Follower (inseguitore di tensione), la cui funzione è quella di adattatore di impedenza tra il segnale da misurare ed un voltmetro digitale esterno a tre cifre di dimensioni maggiori rispetto agli altri tre (reperibile

schema elettrico BASE

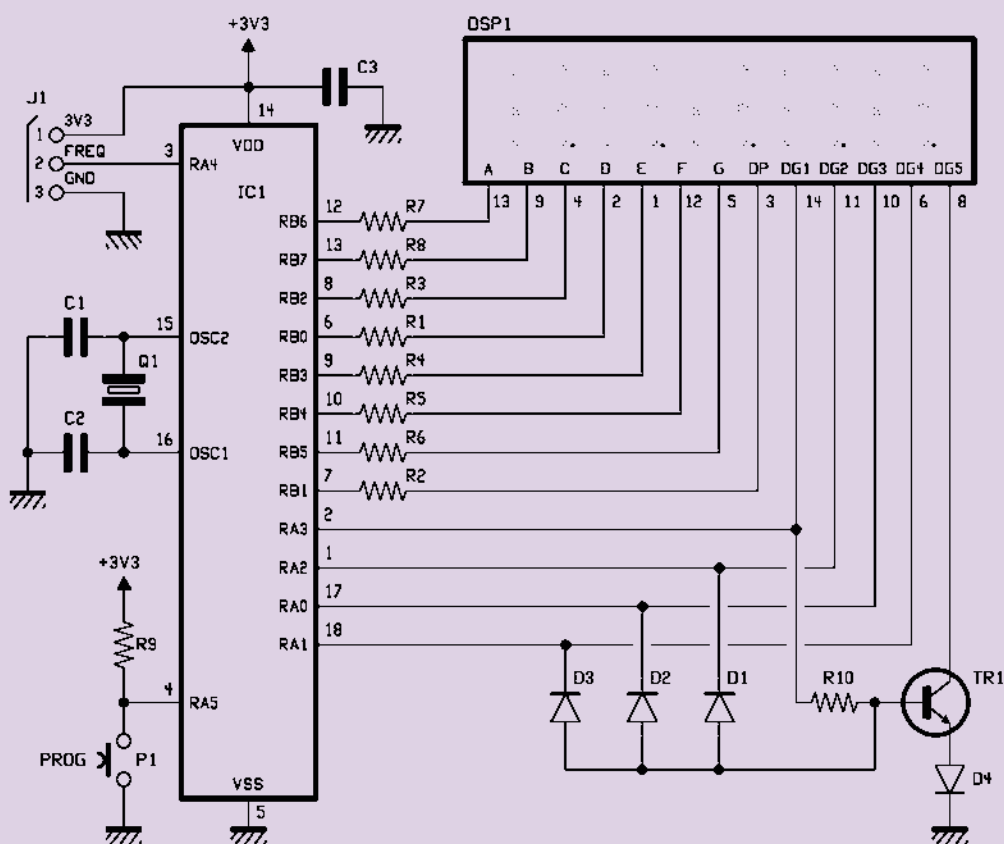
CARATTERISTICHE TECNICHE

- Alimentazione generale esterna: 12 V
- Alimentazioni disponibili: 3,3 V, 5 V, 12 V
- Voltmetri di check per le tre alimentazioni di test
- Connettore header 8 pin per i circuiti di test
- Voltmetro con commutatore a tre vie per l'analisi dei pin SOGLIA, TRIGGER e SCARICA
- Due LED per il controllo degli stati High e Low dell'USCITA
- Connettore a tre pin per frequenzimetro, collegato all'USCITA
- Connettori SMA e USB per mini DSO esterno, collegato all'USCITA



schema elettrico

FREQUENZIMETRO



sul sito www.futurashop.it col codice MINIPAN100). Infatti, collegando direttamente il segnale da misurare all'ingresso del voltmetro, la sua impedenza creerebbe problemi di funzionamento al 555 sotto test; con questo accorgimento invece il segnale viene applicato all'op amp, che ha un'impedenza elevatissima, viene riportato sulla sua uscita, ed infine collegato al voltmetro. Contrariamente ai mini voltmetri, questo strumento richiede un'alimentazione separata, tra 4,5V e 30V, abbiamo sfruttato i 12V del circuito.

Il voltmetro è collegato esternamente al PCB tramite uno spinotto JST XH a tre pin (J3).

In pratica, in base alla posizione della levetta del commutatore, il voltmetro rileva la tensione presente sul pin collegato, in tal modo è possibile controllare l'andamento della tensione su ognuno dei tre pin, durante in funzionamento del circuito sotto test.

Il pin più importante, ai fini dell'analisi del circuito, è sicuramente quello dell'USCITA del 555; come si vede dallo schema elettrico, questo segnale è collegato direttamente al pin centrale dello spinotto SMA (J5) destinato al mini DSO esterno; vicino ad

esso è presente una presa USB (J4) che serve per fornire l'alimentazione da 5 V al mini DSO esterno. Inoltre il segnale di USCITA è collegato anche ad un circuito costituito dal transistor TR1, alcuni resistori e due LED; descriviamo questa particolare configurazione, in quanto potrebbe essere un utile spunto per altre applicazioni.

Per comprendere la necessità di far ricorso a questo circuito, iniziamo col dire che, avendo la possibilità di testare i circuiti con tensioni da 3,3V a 12V, abbiamo dovuto alimentare lo stadio del frequenzimetro con la tensione più piccola, ossia 3,3V, in quanto il suo microcontrollore avrebbe difficoltà a riconoscere e misurare segnali di ampiezza inferiore a quella di alimentazione. Questa scelta però ha un rovescio della medaglia, in quanto applicando segnali con ampiezze maggiori di 3,3 V, è il caso dei circuiti test alimentati a 5V o 12V, si rischierebbe di bruciare l'ingresso del microcontrollore, anche se esso è dotato di una protezione interna.

In pratica TR1, R1, R2 e R3 costituiscono un adattatore di livello, la cui funzione è quella di portare qualsiasi segnale in ingresso, di ampiezza variabile tra 3,3V e 12V ad un livello costante di 3,3V, pro-

teggendo in tal modo il microcontrollore. Il segnale proveniente dal pin di USCITA del 555 sotto test viene applicato alla base del transistor TR1, un comune BJT PNP, tramite R1. Quando l'USCITA è a livello LOW, TR1 va in conduzione e sul suo collettore si ha la tensione applicata all'emettitore (circa 3,3V), quindi il microcontrollore riceve un livello HIGH. Viceversa, quando l'USCITA del 555 è a livello HIGH, TR1 si trova nello stato di interdizione, per cui sul suo collettore abbiamo circa 40 mV, a causa del resistore pull-down R3, che il microcontrollore interpreta come livello LOW. In pratica il segnale in ingresso viene sfasato di 180°, ma questo non è un problema per la misurazione della frequenza. Il resistore R2 serve per la polarizzazione di TR1, mantenendo la sua base a livello HIGH, in assenza di segnale. Con questo semplice metodo, qualunque sia l'ampiezza del segnale applicato alla base di TR1, al microcontrollore del frequenzimetro arriverà sempre un segnale di ampiezza pari a 3,3V. Per poter controllare visivamente il comportamento del circuito sotto test abbiamo aggiunto due LED, che si comportano in questo modo: in presenza di un segnale LOW sul collettore di TR1, si creerà una differenza di potenziale rispetto ai 3,3V presenti sull'emettitore, e la corrente scorrerà attraverso R4 facendo accendere il LED blu LD2, che corrisponderà quindi allo stato HIGH dell'uscita del 555 (ricordiamo che sul collettore il segnale è sfasato di 180°). In presenza di un segnale HIGH sul collettore di TR1, si creerà una differenza di potenziale rispetto alla massa comune, e la corrente in questo caso scorrerà attraverso R5 facendo accendere il LED rosso LD1, che corrisponderà quindi allo stato LOW dell'uscita del 555. Questi due LED ci permetteranno di verificare costantemente lo stato dell'USCITA del 555, cosa molto utile nei circuiti monostabili o bistabili a comando manuale. Nel caso dei circuiti astabili o bistabili con segnale alternato in ingresso, i due LED si accenderanno alternativamente, ma oltre una certa frequenza essi sembreranno accesi contemporaneamente, a causa del fenomeno della persistenza visiva dell'occhio umano, quindi si potrà avere solo conferma del funzionamento del circuito. A questo punto il controllo del segnale si potrà effettuare tramite il frequenzimetro, che viene collegato al circuito base tramite il connettore header a tre pin J6, ed anche tramite il DSO collegato al connettore J5.

STADIO FREQUENZIMETRO

Il cuore del circuito è il microcontrollore MICROCHIP PIC16F628A, che lavora a 20 MHz, in virtù dell'o-



Fig. 2
Visualizzazione di 5 digit in multiplexing.

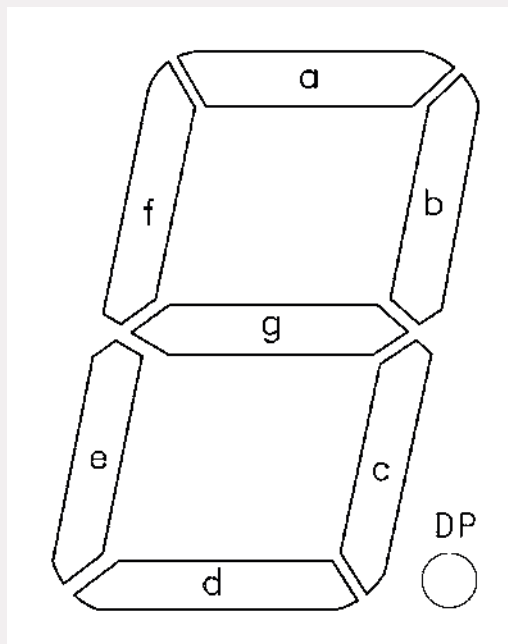


Fig. 3
Display a 7 segmenti con punto decimale.

scillatore esterno costituito dal quarzo Q1 applicato ai pin 15 e 16, e dai condensatori C1 e C2.

Il connettore strip a tre pin J1, innestato nel corrispondente header J6 del circuito base, riceve l'alimentazione a 3,3 V ed il segnale USCITA del 555 del circuito test, entrambi riferiti alla massa comune. Il segnale da misurare viene applicato al pin 3 (RA4). Il micro, oltre a leggere il segnale, si occupa di pilotare i cinque digit del display a LED a catodo comune, con la tecnica del multiplexing; gli otto resistori da R1 a R8 servono per limitare la corrente che dovrà scorrere nei display.

Dal momento che il numero di pin disponibili permette di pilotare solo quattro delle cinque cifre, sfruttando le potenzialità del firmware (scritto da Wolfgang Buescher, DL4YHF), abbiamo aggiunto un circuito esterno per aggirare tale limitazione. In pratica, mentre i primi 4 digit sono pilotati rispettivamente dai pin 2, 1, 17 e 18, l'accensione del quinto digit (l'ultimo a destra, cifra meno significativa) dipende, solo per frequenze uguali o maggiori di 10 kHz (condizione dettata dal firmware), dalla combinazione degli altri quattro. I diodi D2, D3 e D4, ed il transistor NPN TR1 costituiscono una porta NAND a 4 ingressi, quando uno dei pin indicati in

precedenza si trova a livello LOW (digit acceso), tale livello si troverà sulla base del transistor, mantenendolo in stato di interdizione. Solo nell'istante in cui il multiplexer spegne i quattro digit (pin a livello HIGH), il resistore R10, collegato al pin 2, piloterà la base di TR1 mandandolo in saturazione e collegando quindi il pin 5 del display a massa, provocandone l'accensione.

Per tale ragione all'accensione del frequenzimetro vedrete che lo zero, o la cifra meno significativa, per frequenze fino a 9,999 Hz, saranno visualizzati sul quarto digit. Per meglio comprendere questo particolare sistema di pilotaggio, bisogna tener conto che i cinque digit del display sono costituiti da una serie di sette LED (segmenti) orientati in modo costruire un "8"; trattandosi di un display a catodo comune, ogni segmento si accende quando viene pilotato da un segnale HIGH, mentre il pin comune è su LOW.

Facciamo un esempio: immaginiamo di dover visualizzare la lettura "32.768" (**Fig. 2**), la visualizzazione avviene mediante cinque operazioni eseguite in rapidissima sequenza (per le lettere dei segmenti fate riferimento alla **Fig. 3**):

1. il microcontrollore mette in HIGH i pin necessari per formare il numero **3** (segmenti "a", "b", "c", "d" e "g"), poi mette in LOW il solo pin 2 (gli altri tutti in HIGH), collegato al catodo pin 14, in tal modo il numero 3 viene mostrato solo sul primo digit (partendo da sinistra, cifra più significativa);
2. il microcontrollore mette in HIGH i pin necessari per formare il numero **2** con il punto decimale (segmenti "a", "b", "d", "e", "g" e "DP"), poi mette in LOW il solo pin 1 (collegato al catodo pin 11) in tal modo il numero 2 viene mostrato solo sul secondo digit;

3. il microcontrollore mette in HIGH i pin necessari per formare il numero **7** ("a", "b" e "c"), poi mette in LOW il solo pin 17 (collegato al catodo pin 10) in tal modo il numero 7 viene mostrato solo sul terzo digit;
4. il microcontrollore mette in HIGH i pin necessari per formare il numero **6** (tutti ad eccezione di "b" e "DP"), poi mette in LOW il solo pin 18 (collegato al catodo pin 6) in tal modo il numero 6 viene mostrato solo sul quarto digit;
5. il microcontrollore mette in HIGH i pin necessari per formare il numero **8** (tutti ad eccezione di "DP"), ed anche i quattro pin dei catodi (2, 1, 17, 18); questa condizione, come abbiamo visto, manda in conduzione TR2, il catodo pin 8 viene collegato a massa tramite il diodo D4 e il numero 8 viene mostrato sul quinto digit.
6. il microcontrollore ricomincia la sequenza dal punto **1**, e continuerà finché non cambierà il valore misurato.

Abbiamo usato il termine "sequenza" perché infatti queste cinque operazioni avvengono separatamente, e per ogni digit che si accende gli altri quattro restano spenti, però noi li vedremo accesi contemporaneamente e ciò è dovuto alla frequenza elevata con la quale il microcontrollore ripete la sequenza ed al fenomeno della persistenza visiva dell'occhio umano. Seguendo questo ragionamento si potrebbe obiettare che il quinto digit allora dovrebbe accendersi sempre, quando gli altri sono spenti, durante il multiplexing; in effetti sarebbe così ma, come abbiamo detto, ci pensa il firmware a tenere spento il digit quando si misurano frequenze inferiori a 10.000 Hz, semplicemente mantenendo LOW tutti i pin che pilotano i 7 segmenti ed il DP. Il pulsante P1, assieme al resistore di pull-up R9, serve per entrare nella modalità di programmazione del microcontrollore, ne approfondiremo l'uso nella parte relativa al funzionamento.

MINI DSO

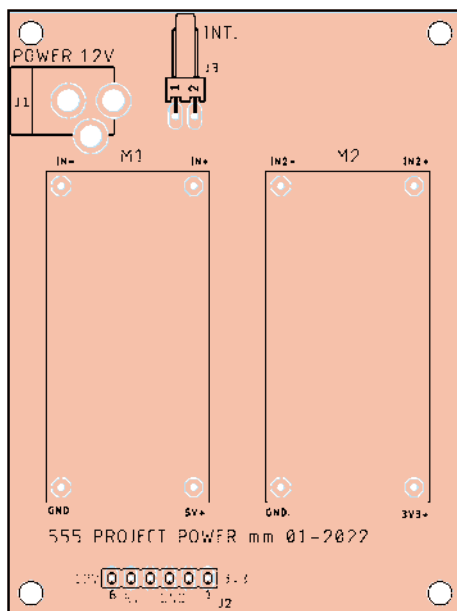
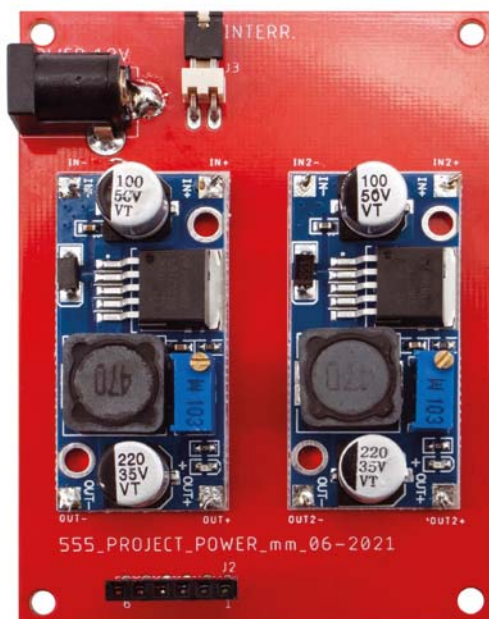
In **Fig. 4** potete vedere l'immagine del mini DSO che abbiamo usato per il nostro Analizzatore. Si tratta di un piccolo strumento (reperibile sul sito www.futurashop.it col codice DSO138MINI) con un display LCD TFT a colori da 2,4", il connettore d'ingresso è un BNC standard, quindi per il collegamento alla base serve un cavetto BNC maschio - SMA maschio; in alternativa è possibile usare un comune cavetto BNC M-M ma occorre dotare la presa SMA del circuito base di uno spinotto adattatore da SMA M a BNC F (reperibile sul sito www.futurashop.it

Fig. 4
Il mini
DSO138.



piano di montaggio

ALIMENTAZIONE



Elenco Componenti:

J1: PLUG DC 2,1mm
 J2: strip femmina 6 vie H= 8,5+15 mm
 J3: connettore 2 poli 90° c.s. BTWMH2
 M1, M2: modulo step-down, out 1,5 V - 35 V, 3 A

Varie:

- alimentatore switching 12 V - 1,2 A
- strip maschio 1 via
- circuito stampato S1635_Power (80x60 mm)

col codice SMABNCMF). Esistono svariati modelli di mini DSO, per questo Analizzatore vanno bene nella maggior parte dei casi, così come, rinunciando alla trasportabilità del nostro strumento, si può collegare tranquillamente un DSO professionale. Anche l'uso del mini DSO sarà approfondito nella sezione dedicata al funzionamento dell'Analizzatore.

MONTAGGIO DEI CIRCUITI

Vediamo ora nel dettaglio il montaggio dei tre PCB. Nella sezione download del sito www.elettronica.in.it troverete i file gerber per la realizzazione dei PCB occorrenti; se non disponete dell'attrezzatura per prepararli, potete affidarvi al servizio PCBPRODUCTION, accessibile da www.futurashop.it/produzionePCB.

Stadio di alimentazione

Il montaggio di questo primo circuito è molto semplice, bisogna saldare un pin in ognuno degli

8 fori serigrafati: IN+, IN-, IN2+, IN2-, 5V+, 3V3+, GND. e GND, quindi si possono inserire i due moduli switching step-down, facendo ben attenzione al verso, è sufficiente far combaciare i pin IN+ e IN- con i corrispondenti fori dei moduli; ora basta saldare i pin anche sulla superficie dei moduli e l'operazione è terminata.

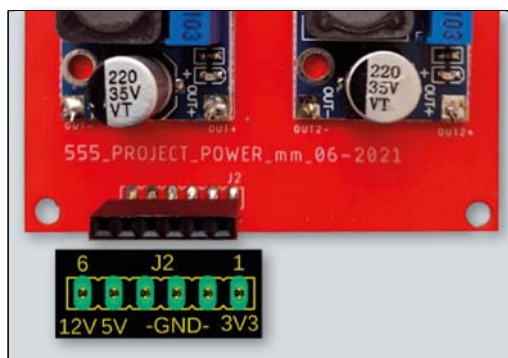
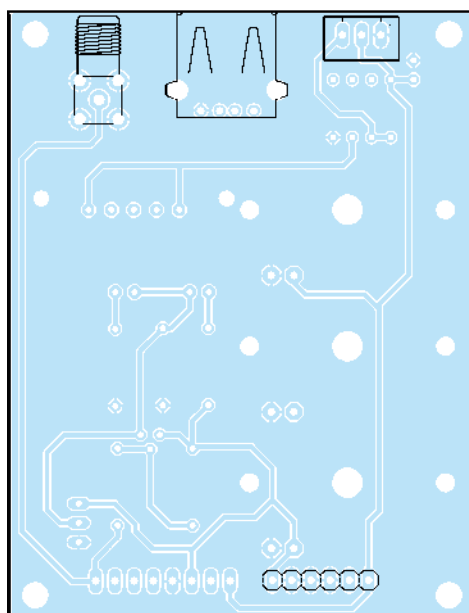


Fig. 5
 La connessione del circuito di alimentazione montato.



Elenco Componenti:

- R1: 1 kohm
- R2: 10 kohm
- R3: 100 ohm
- R4: 390 ohm
- R5: 470 ohm
- C1: 100 nF 50 VL poliestere
o ceramico
- LD1: LED 5 mm rosso
- LD2: LED 5 mm blu
- TR1: BC557
- IC1: MCP6002P
- J1: strip maschio 6 vie L= 13 mm
- J2: strip femmina 8 vie
H= 8,5+15 mm
- J3: JST XH 3 poli maschio c.s.
- J4: USB-A femmina c.s.
- J5: SMA femmina 90° c.s.
- J6: strip femmina 3 vie 90°
- SW1: deviatore 1 via 3 pos.

A seguire bisogna saldare il PLUG DC ed il connettore MK a 2 poli a 90°; specifichiamo che questo connettore serve per dotare lo strumento di un interruttore generale di alimentazione, se non prevedete di usarlo potete sostituirlo con un ponticello per mettere in cortocircuito i due pad, oppure potete comunque montarlo, come predisposizione, e poi cortocircuitare i due pin con un jumper. Infine si deve inserire il connettore header femmina a 6 poli, deve essere del tipo a passo lungo, quello che viene usato negli shield per Arduino; questo connettore deve essere perfettamente perpendicolare alla superficie del PCB, in quanto servirà per innestare la base. Il circuito è pronto, consigliamo di procedere subito con la taratura dei due moduli. Basta collegare al PLUG DC un alimentatore da 12 V 0,5 A e si accenderanno i LED dei due moduli; bisogna applicare un multimetro digitale tra OUT+ e OUT- e ruotare il trimmer di precisione fino a leggere 5,0-5,1 V sul modulo M1 e 3,3-3,4 V sul modulo M2, le serigrafie aiuteranno ad indentificare correttamente i due moduli. Dopo la taratura prendete nota dei valori esatti, anche quello dell'alimentatore esterno, in quanto serviranno più avanti. In **Fig. 5** potete vedere il circuito di alimentazione montato, ed il particolare delle alimentazioni sul connettore J2.

Stadio base

Questo circuito è leggermente più complesso, in quanto la maggior parte dei connettori va saldata

sul lato inferiore del PCB, inoltre ci sono alcuni montaggi di tipo meccanico, ma andiamo con ordine, iniziando dalla faccia superiore.

Come sempre conviene iniziare a saldare i componenti più bassi, quindi i cinque resistori, il connettore J6, poi lo zoccolo a 8 pin ed il condensatore, il transistor TR1 ed il connettore J2; riguardo quest'ultimo, è importante che il corpo sia sufficientemente alto da poter permettere l'innesto dei pin strip dei circuiti di test. Noi abbiamo usato quello per gli shield di Arduino, fermo restando che in questo caso i pin sporgeranno sul lato inferiore del PCB, potete decidere di tagliarli ma, poiché non toccano da nessuna parte, magari conviene lasciarli per poterli usare all'occorrenza come test point delle alimentazioni e dei segnali.

Per quanto riguarda i tre voltmetri V1, V2 e V3, bisogna accorciare e saldare i due fili nei rispettivi pad, facendo attenzione alla polarità, il filo rosso va saldato al pad IN+, quello nero al pad IN-. Abbiamo usato display con colori diversi più per coreografia che per necessità, ma potrete benissimo adottare un solo colore; se seguirete la nostra modalità però è importante che rispettiate le serigrafie: verde per 12 V, rosso per 5 V, blu per 3,3 V, in quanto il verde è il meno luminoso e conviene alimentarlo con la tensione maggiore, mentre il blu è il più luminoso e va alimentato con la tensione minore.

Il fissaggio dei tre moduli si effettua con sei viti M2 con i relativi dadi, ma bisogna fare attenzione

c.s. SSSU-014800

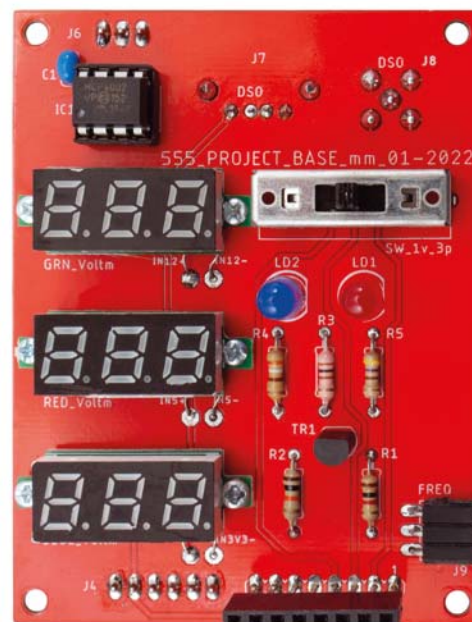
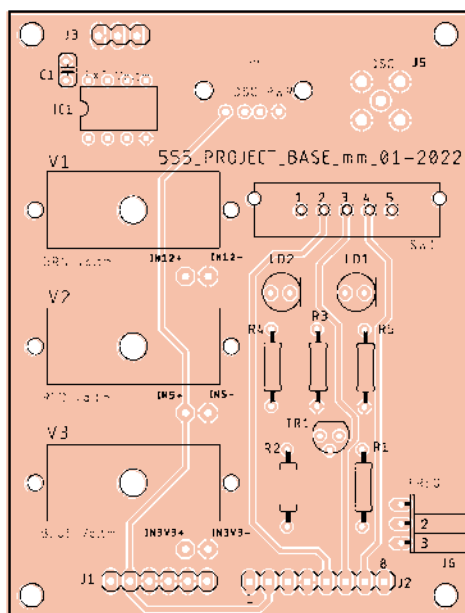
V1: Mini Voltmetro 3 cifre Verde

V2: Mini Voltmetro 3 cifre Rosso

V3: Mini Voltmetro 3 cifre Blu

Varie:

- voltmetro 3 cifre Rosso
- cavetto board-to-wire con JST XH 3 poli femmina 150 mm
- zoccolo 4+4
- distanziale esagonale M/F 20 mm
- distanziale esagonale F/F 8 mm
- vite M3x5
- circuito stampato S1635_base (80x61,5 mm)



a stringere il minimo indispensabile, altrimenti si rischia di spaccarli, visto che le alette forate non toccano direttamente la superficie del PCB; oppure si può inserire uno spessore tra ogni aletta ed il PCB.

Ora potete inserire e saldare il deviatore, consigliamo di usare il modello indicato nell'elenco componenti, perché altri tipi probabilmente non combaceranno con i fori ma, soprattutto, potrebbero avere un sistema di connessione diverso delle vie, rischiando di mandare in corto i segnali del circuito test. I due LED andranno distanziati dal PCB di circa 4+5 mm. Infine occorre girare il PCB dal lato saldature ed inserire i connettori J1, J3, J5 e J6; per J1 occorre avere la stessa accortezza usata per l'header del PCB di alimentazione, cioè i pin devono essere perfettamente perpendicolari alla superficie del PCB. Una volta completate tutte le operazioni di saldatura si potrà inserire l'integrato MCP6002 nell'apposito zoccolo, facendo attenzione al verso. In **Fig. 6** abbiamo riportato il particolare delle alimentazioni e dei segnali sul connettore J2.

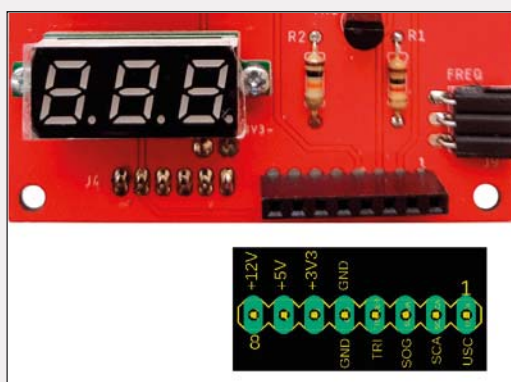
Come potete notare sul lato inferiore ci sono tre fori, all'interno dei quali si vedono i piccoli trimmer di taratura dei minivoltmetri; per questa operazione bisogna collegare il connettore header J2 del PCB di alimentazione con il connettore strip J1 della base, ma non direttamente, altrimenti sarà impossibile raggiungere i trimmer. Un ottimo sistema è quello di ricorrere a 4 cavetti jumper M-F, sufficiente-

mente lunghi, per collegare i fori 1, 2, 5 e 6 di J2 alimentazione ai corrispondenti pin di J1 base. Una volta alimentato il circuito i display si accenderanno mostrando indicativamente le tre tensioni di alimentazione; agendo con un piccolo cacciavite, possibilmente in plastica, occorrerà regolare i trimmer dei minivoltmetri per avvicinarsi il più possibile ai valori che avete precedentemente rilevato con il multimetro. La precisione assoluta non è richiesta, anche perché probabilmente l'ultima cifra degli strumentini apparirà "ballerina", e lo scopo fondamentale di queste misure è solo quello di verificare che siano presenti le tre tensioni e non si verifichino cortocircuiti, cosa che sarebbe indicata dal quasi totale azzeramento di uno o più valori.

Stadio Frequenzimetro

Anche per l'ultimo stadio procediamo come per i precedenti, quindi iniziamo il montaggio con i resistori, i diodi ed i condensatori, il piccolo strip a 3 pin, poi quarzo e zoccolo. Per quanto riguarda il display, avendo i pin molto lunghi, bisognerà fare in modo che la sua altezza sia identica a quella dei tre minivoltmetri della base, ai fini del montaggio in un contenitore, altrimenti ci saranno problemi con le sporgenze. Una volta stabilito quanta porzione dei pin deve restare fuori dal PCB, per saldare il display in modo perfettamente orizzontale, consigliamo di preparare quattro spessori identici, per esempio con una guaina tolta da un filo di rame, o qualsi-

Fig. 6
Le connessioni del
circuit base.

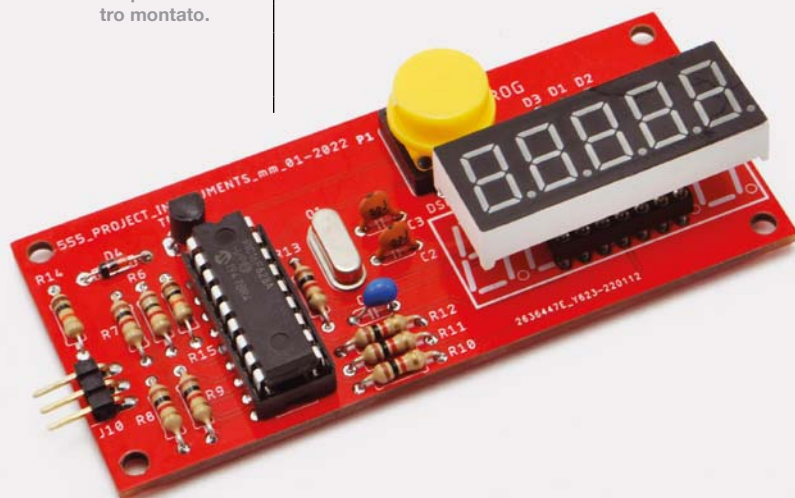


asi altro sistema, ed infilarli nei due pin esterni di ognuna delle due file, in questo modo sarà molto semplice fare le saldature senza commettere errori di posizionamento. Una volta completate tutte le operazioni di saldatura si potrà inserire l'integrato PIC16F628A nell'apposito zoccolo, facendo attenzione al verso; naturalmente, il microcontrollore dovrà essere preventivamente programmato con il firmware che troverete nella sezione download del sito www.elettronica.it. Il file da scaricare si chiama "freq_counter.zip", una volta estratto il contenuto troverete tre versioni del firmware, quella che serve nel nostro caso è la versione 2. In **Fig. 7** potete vedere il circuito stampato del frequenzimetro montato.

VOLTMETRO ESTERNO

Il voltmetro MINIPAN100 viene fornito con i tre terminali liberi quindi, per il collegamento al connettore J3 della base, occorre dotarlo di uno spinotto JST XH femmina a tre pin; se necessario, nel caso di montaggio dell'Analizzatore in un contenitore, allungare prima i tre fili; comunque si trovano in

Fig. 7
Il circuito del
frequenzimetro
montato.



commercio cavetti già pronti, potete giuntarli ai tre fili esistenti oppure sostituirli completamente. Per il collegamento al connettore fare riferimento alla **Fig. 8**, attenzione a non invertire i colori dei fili, altrimenti rischiate di bruciarlo.

IL VOLTMETRO? ANCHE ANALOGICO!

Il voltmetro digitale da noi previsto non effettua una lettura istantanea del segnale, quindi, nel caso dei circuiti astabili, che oscillano ciclicamente, solo a frequenze molto basse è possibile seguire l'andamento delle tensioni rilevate sui pin TRIGGER, SOGLIA e SCARICA; già con frequenze maggiori di 1 Hz le variazioni sono troppo rapide per essere rilevate correttamente da questo strumento. Il problema non si pone invece per la maggior parte dei circuiti monostabili e bistabili in quanto le tempistiche sono molto lunghe e quindi l'osservazione si può fare ed anche con molta precisione. È però possibile sostituire lo strumento digitale con il corrispondente analogico, cioè uno strumento meccanico a lancetta, che invece reagisce istantaneamente alle variazioni, e permette di valutare i momenti di carica e scarica del condensatore CX, oppure le variazioni manuali, o ancora quelle dovute a sensori, mediante l'oscillazione della lancetta nel range dei valori misurati. Bisogna sceglierne uno con fondo scala idoneo alle tensioni da misurare, nel nostro caso va benissimo un voltmetro da 10 V, che rileva tutte le tensioni dei nostri circuiti. Non trovando un voltmetro con fondo scala idoneo, si può anche adattare, come abbiamo fatto per le nostre prove, un microamperometro, aggiungendo un resistore di precisione in serie al segnale.

Nella **Fig. 9** vedete i due strumentini analogici fotografati dal quadrante.

Specifichiamo che deve trattarsi proprio di un microamperometro, con un fondo scala massimo di 50 o 100 μA , in quanto le correnti erogate dall'amplificatore operativo della scheda base sono limitate.

Il collegamento avviene sempre sulla presa JST HX J3 della base, solo che bastano il pin del segnale e quello della massa, in quanto non serve l'alimentazione. Nel caso del voltmetro è sufficiente collegare l'occhietto "+" al segnale e quello "-" alla massa mentre, nel caso dell'amperometro, bisogna aggiungere un resistore in serie tra il segnale e l'occhietto "+", mentre il "-" va collegato direttamente alla massa (**Fig. 10**).

Il valore del resistore va calcolato in base al fondo scala dell'amperometro ed al valore di fondo scala che si vuole ottenere per farlo funzionare come

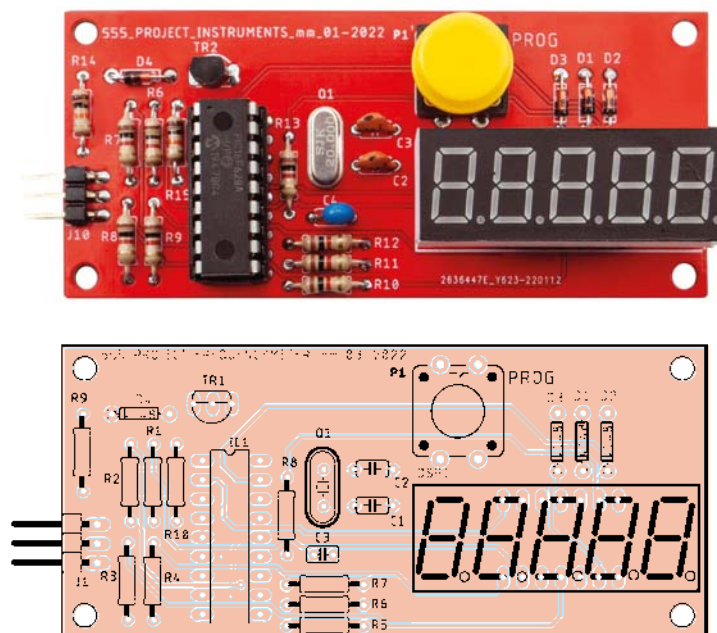
piano di montaggio FREQUENZIMETRO

Elenco Componenti:

R1, R2, R3, R4, R5, R6, R7, R8: 1 kohm
 R9, R10: 10 kohm
 C1, C2: 22 pF ceramico
 C3: 100 nF poliestere o ceramico
 D1, D2, D3, D4: 1N4148
 TR1: BC547
 DSP1: Display 5 cifre 3651AS-1
 Q1: quarzo 20 MHz
 IC1: PIC16F628A
 P1: pulsante 12x12 mm c.s.
 J1: strip maschio 3 vie 90°

Varie:

- zoccolo 9+9
- distanziale esagonale F/F 8 mm
- vite M3x5
- circuito stampato S1635_Freq (85,5x39 mm)



voltmetro. È sufficiente ricorrere alla notissima Legge di Ohm, nella quale R è il valore da calcolare espresso in ohm, V la tensione di fondo scala da ottenere (espressa in volt) ed I il fondo scala del microamperometro in ampere.

Siccome lo strumentino noi lo abbiamo utilizzato per i circuiti funzionanti a 5 V, dal calcolo otteniamo quindi una resistenza di 100 kohm. Se invece avessimo voluto avere un fondo scala da 10 volt, il resistore serie da inserire sarebbe risultato da 200 kohm.

IL COLLAUDO

Prima di fornire l'alimentazione principale al nostro Analizzatore, bisogna collegare tra loro i tre circuiti che lo compongono. L'alimentazione e la base vanno innestate tra loro mediante i connettori J2 (alimentazione) e J1 (base), e occorre usare 4 distanziali plastici da 20 mm, fissandoli al PCB dell'alimentazione mediante altrettante viti M3, mentre il PCB base andrà inserito ad incastro sulla parte filettata dei distanziali, e fissato mediante 4 dadi M3 (Fig. 11).

Il frequenzimetro deve essere collegato alla scheda base mediante i connettori J6 (base) e J1 (frequenzi-

metro), come si vede nella Fig. 12.

Per il momento non bisogna collegare il voltmetro esterno ed il mini DSO, si deve alimentare il circuito mediante l'alimentatore da 12V esterno, se è tutto

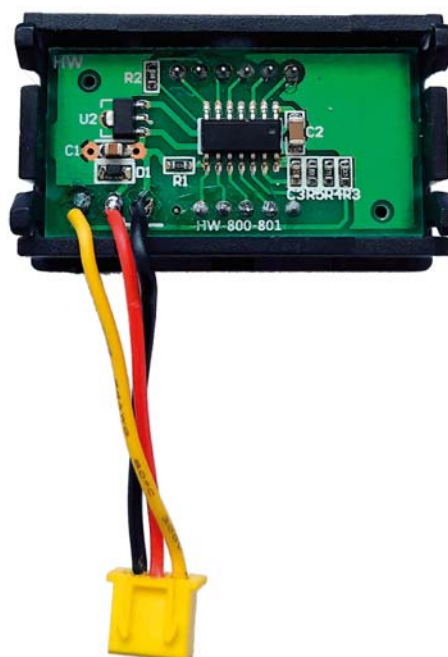


Fig. 8
 Il voltmetro
 esterno
 dotato di
 spinotto JST
 XH.

Fig. 9
Strumenti analogici:
Voltmetro da 10 V
e Amperometro da
50 μ A.

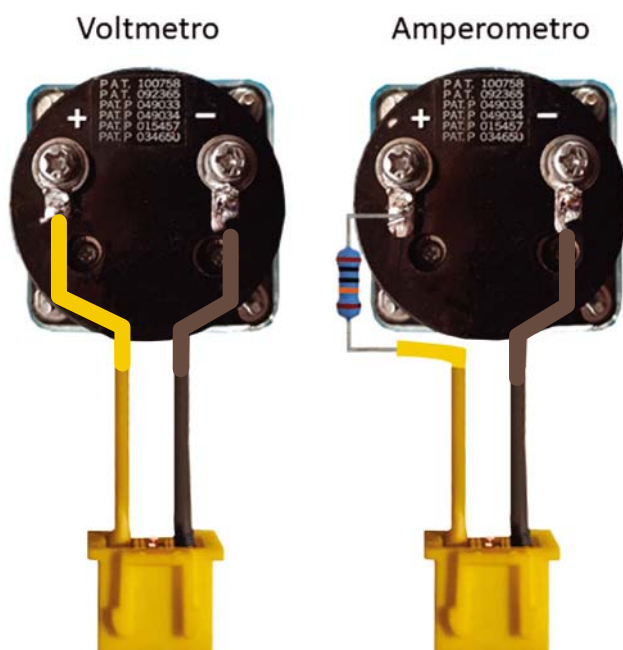


a posto, si avrà questa situazione: i tre minivoltmetri mostreranno le relative tensioni, come da precedente taratura, il LED blu sarà acceso, quello rosso spento, infine il display del frequenzimetro visualizzerà uno zero sulla quarta cifra, mentre tutti gli altri digit saranno spenti. Nel caso in cui il frequenzimetro dovesse mostrare una misura qualsiasi, allora sarà necessario mettere mano al menu di programmazione, per il set dello ZERO, mediante il pulsante P1.

Fig. 10
Collegamento degli
strumenti
analogici alla
base.

PROGRAMMAZIONE DEL FREQUENZIMETRO

Questo menu serve solo se si verifica qualche incongruenza nel funzionamento del frequenzimetro, in caso contrario si può saltare la lettura di questo paragrafo. Prima di iniziare occorre ricordare che



per entrare nel menu occorre una pressione lunga del pulsante P1, mentre una pressione breve serve per passare alla voce di menu successiva o per navigare all'interno della voce di menu (nel caso di "Table"); infine, per attivare la voce di menu selezionata, occorre una pressione lunga (circa 2 secondi). Entrando nel menu appare subito la voce "Prog", rilasciando il pulsante appare la voce "Quit", per uscire senza modificare nulla basta una pressione lunga in questa fase.

Nella **Fig. 13** abbiamo inserito le voci di menu in sequenza e la funzione svolta con la pressione lunga di P1; inoltre nell'ultima colonna abbiamo riportato le foto del display in quanto, essendo a 7 segmenti e non alfanumerico, non sempre le parole sono di immediata lettura.

Una prima pressione breve di P1 fa passare alla voce "PSave" o "NoPSave": nel primo caso è attivo il risparmio energetico, per cui dopo 15 secondi dall'ultima lettura, in mancanza di variazioni, il display si spegne, nel secondo caso invece il display resterà sempre acceso, anche in assenza di segnale in ingresso. Una pressione lunga di P1 provocherà il passaggio da un'opzione all'altra e poi, premendo ulteriormente, l'uscita dal menu. Una seconda pressione breve di P1 farà scorrere il menu sulla voce successiva "Add"; questa opzione serve per aggiungere alla lettura reale un valore prefissato (offset), rappresentato dal valore visualizzato sul display nel momento in cui si entra nel menu oppure da uno dei valori predefiniti presenti in "Table". È sufficiente una pressione lunga di P1 per memorizzare tale valore e trovarselo stabilmente aggiunto ad ogni lettura effettuata.

Un esempio chiarirà meglio questa opzione: poniamo di aver necessità di aggiungere 1.000 Hz ad ogni lettura del frequenzimetro; la prima operazione da effettuare è quella di applicare un segnale da 1000 Hz al frequenzimetro, quindi si

entra nel menu, si seleziona l'opzione "Add" e poi una pressione lunga di P1, da questo momento qualunque sia la frequenza letta mostrerà 1000 Hz in più, quindi 10 kHz diventeranno 11.000 kHz, 1 MHz diventerà 1.0010 MHz, ecc.

Se invece la frequenza da aggiungere rientra in quelle memorizzate nel microcontrollore, e disponibili nel sottomenu "Table", allora non serve alcun segnale in ingresso, basta selezionare il valore in "Table" e poi una pressione lunga di P1, il menu mostra la voce "Add", una nuova pressione lunga e la programmazione sarà fatta.

Da "Add" una pressione breve di P1 porta alla voce "Sub", per la quale vale tutto quanto spiegato per "Add" con la sostanziale differenza che le frequenze saranno sottratte alla lettura reale.

Da "Sub" una pressione breve di P1 porta alla voce "Zero", il cui scopo è quello di azzerare l'eventuale offset aggiunto o sottratto in precedenza.

Nota bene: potrebbe capitare che il frequenzimetro, in assenza di circuiti di test, mostri un valore diverso da zero, probabilmente perché riporta un offset inserito in fase di test del microcontrollore. Per eliminarlo sarà sufficiente usare la funzione "Zero".

Dalla funzione "Zero" una pressione breve del pulsante P1 porta alla voce "Table" che, come detto, contiene una serie di valori prememorizzati

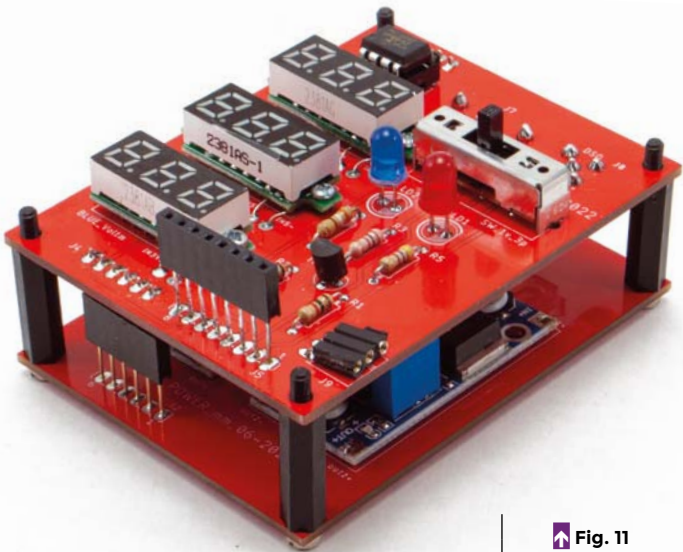
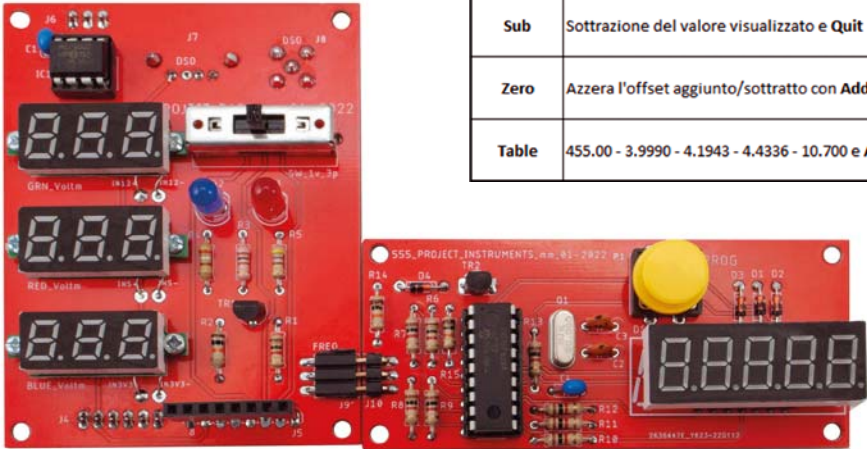


Fig. 11
I PCB alimentazione e base assemblati tramite colonnine distanziali.

in EEPROM, raggiungibili con una pressione lunga di P1 e precisamente: 455.0 (kHz), 3.9990 (MHz), 4.1943 (MHz), 4.4336 (MHz), 10.700 (MHz); una pressione breve di P1 permette di navigare tra i cinque valori, una pressione lunga di P1 seleziona il valore visualizzato.

A questo punto il menu si porta automaticamente sulla voce "Add", se si vuole aggiungere la frequenza selezionata alla lettura basta una pressione lunga di P1, se invece la si vuole sottrarre serve

Fig. 12
Collegamento del frequenzimetro alla base.



P1 breve	P1 lungo	Display
	Prog - apertura del menu	8888
Quit	uscita dal menu	8888
PSAVE	NoPSAVE (o viceversa) e Quit	8888 8888
Add	Aggiunta del valore visualizzato e Quit	8888
Sub	Sottrazione del valore visualizzato e Quit	8888
Zero	Azzerare l'offset aggiunto/sottratto con Add/Sub e Quit	8888
Table	455.00 - 3.9990 - 4.1943 - 4.4336 - 10.700 e Add/Sub	8888

Fig. 13
Menu di programmazione.

Fig. 14
Cavetti di collegamento del mini DSO138.



prima una pressione breve per passare a "Sub" e poi una pressione lunga per attivare il valore; infine, se non si vuole applicare il valore selezionato, basteranno una serie di pressioni brevi su P1 fino a visualizzare "Quit" e poi una pressione lunga per uscire dal menu.

COLLEGAMENTO DEGLI STRUMENTI ESTERNI

Se il comportamento degli indicatori (display e LED) all'accensione è corretto, bisogna togliere l'alimentazione e collegare il voltmetro esterno, inserendo lo spinotto JST nel connettore J3 ed il mini DSO, collegando il cavo USB-micro e quello BNC M - SMA M nei connettori J4 e J5. In **Fig. 14** potete vedere i due cavetti necessari per il collegamento del mini DSO.

Come già spiegato, il cavo BNC M - SMA M può essere sostituito da un più comune BNC M-M, a patto di montare sul connettore J8 un adattatore SMA M - BNC F.

Ora si può collegare nuovamente l'alimentazione, il voltmetro esterno mostrerà dei valori a caso, a prescindere dalla posizione dello switch SW1, mentre sul display del DSO apparirà una linea continua; in caso contrario bisognerà controllare i collegamenti.

USO DEL MINI DSO 138

Per poter visualizzare le varie tipologie di segnali che produrremo con i circuiti test, occorre imparare ad impostare correttamente il mini DSO, quindi facciamo una carrellata delle funzionalità dello strumento, impostabili mediante i tre switch posti sul lato posteriore (quello del BNC d'ingresso) ed i quattro pulsanti posizionati alla base del display. Naturalmente diamo per scontato che si conosca l'uso basilare di un oscilloscopio digitale,

non essendo uno strumento di uso immediato per il neofita.

Switch:

- CPL: accoppiamento del segnale d'ingresso
- DC: continua, il segnale viene letto assieme alla sua componente continua;
- AC: viene letta la sola componente alternata;
- GND: ingresso chiuso a massa.
- SEN1: risoluzione dell'asse verticale: 10 mV, 100 mV, 1 V; è il valore base di ogni quadratino.
- SEN2: moltiplicatore dell'asse verticale: x1 (= SEN1), x2 (2 x SEN1), x5 (5 x SEN1); questo selettore moltiplica il valore impostato con SEN1, quindi la risoluzione verticale può variare da un minimo di 10 mV (10 mV x 1) ad un massimo di 5 V (1 V x 5).

Pulsanti:

- SEL: selezione del parametro da impostare.
- - e +: variazione del valore del parametro selezionato.
- OK: modalità di visualizzazione: RUN (esecuzione) o HOLD (stop).

Parametri:

- Timebase (risoluzione asse orizzontale): 500 s, 200 s, 100 s, 50 s, 20 s, 10 s, 5 s, 2 s, 1 s, 0.5 s, 0.2 s, 0.1 s, 50 ms, 20 ms, 10 ms, 5 ms, 2 ms, 1 ms, 0.5 ms, 0.2 ms, 0.1 ms, 50 μs, 20 μs, 10 μs. Una pressione prolungata dei pulsanti fa scorrere rapidamente i valori del Timebase.
- Modalità di Trigger: Auto, Normal, Single.
- Fronte del Trigger: falling (discesa) o rising (salita).
- Sorgente del Trigger: interna o esterna.
- Livello del Trigger: posizionamento indicatore di livello (la freccetta rosa sul lato destro del display diventa di colore azzurro): una pressione lunga sui pulsanti -/+ muove il cursore a scatti; la pressione prolungata provoca lo scorrimento lineare del cursore, una pressione breve permette un singolo movimento di precisione; la pressione per tre secondi del pulsante OK porta la barra esattamente al centro verticale del segnale.
- Indicatore posizione orizzontale del segnale (la barra verde sul lato superiore del display diventa di colore azzurro): una pressione lunga sui pulsanti -/+ muove la barra a scatti; una pressione breve permette un singolo movimento di precisione; la pressione per tre secondi del pulsante OK porta la barra esattamente al centro orizzontale del segnale.

- Indicatore posizione verticale del segnale (la freccetta gialla sul lato sinistro del display diventa di colore azzurro): una pressione lunga sui pulsanti +/- muove il cursore a scatti; la pressione prolungata provoca lo scorrimento lineare del cursore, una pressione breve permette un singolo movimento di precisione; la pressione contemporanea di entrambi i pulsanti +/- (e non del tasto OK!) porta il cursore esattamente sullo zero verticale.

Funzioni aggiuntive SEL/OK:

- Misure: selezionando con SEL il parametro Timebase, una pressione di circa 3 secondi del pulsante OK attiva la misura dei principali parametri del segnale; una nuova pressione di 3 secondi nasconde i parametri:
- Freq: frequenza;
- Cycl: periodo;
- PW: larghezza impulso;
- Duty: Duty Cycle;
- Vmax: livello di tensione massimo;
- Vmin: livello di tensione minimo;
- Vavr: livello di tensione medio;
- Vpp: tensione picco-picco;
- Vrms: valore di tensione efficace (RMS).
- Taratura zero verticale: selezionando con SEL il cursore verticale, una pressione di circa 3 secondi del pulsante OK attiva la procedura di set della posizione zero del cursore; bisogna spostare gli switch sulla posizione richiesta dal valore lampeggiante.
- Salvataggio segnale: premendo contemporaneamente SEL e +, il segnale visualizzato viene memorizzato nella EEPROM, un eventuale segnale precedentemente memorizzato viene sovrascritto.
- Caricamento segnale: premendo contemporaneamente SEL e -, il segnale attualmente memorizzato nella EEPROM viene mostrato sul display.

CONTENITORE

Per il montaggio di questo strumento abbiamo utilizzato e lavorato il contenitore plastico Teko PULT362, sostituendo il pannello originale in alluminio con uno che abbiamo appositamente disegnato e stampato in 3D, col risultato visibile in **Fig. 15**. Se vi piace questa soluzione troverete il file stl nella sezione download del sito www.elettronicaonline.it; se non avete la possibilità di stampare in autonomia questo pannello potete richiederlo al nostro service, che provvederà a realizzarlo per voi.



Fig. 15
L'Analizzatore del timer 555 montato nello specifico contenitore.

Naturalmente nulla vieta di lavorare il pannello metallico manualmente, facendo i fori con una fresa a mano o con il trapano, rifinandoli poi con delle lime di varia grana e dimensione in base alle necessità.

CONCLUSIONI

Arrivati a questo punto possiamo concludere la prima puntata, che è stata la presentazione del nostro Analizzatore, avendo studiato lo schema elettrico, effettuato il montaggio e visto il funzionamento di massima dei moduli che utilizzeremo. Nelle prossime puntate collauderemo il nostro Analizzatore con una serie di piccoli circuiti già pronti, da inserire nel connettore header di test e, per ognuno di essi, forniremo ampie spiegazioni sul funzionamento, in base alla tipologia (monostabile, bistabile, astabile), in modo che possano essere validi spunti per applicazioni personalizzate. Sfrutteremo quindi l'Analizzatore per leggerne i segnali durante i test.

Per il momento vi auguriamo buon lavoro e vi diamo appuntamento alla prossima puntata. ■



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica, all'interno dell'articolo abbiamo menzionato i codici da cercare all'interno del sito www.futurashop.it.

I circuiti stampati possono essere realizzati 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**

Fai il pieno di energia!

€ 12,⁹⁰
Cod. UL45-6



6V-4,5AH

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.

12V-1,3AH



€ 13,⁹⁰
Cod. UL13-12



12V-7AH

€ 24,⁹⁰
Cod. UL7-12

6V-1,3AH



€ 7,⁹⁰
Cod. UL13-6



6V-3,4AH

€ 11,⁹⁰
Cod. UL34-6



Batterie ricaricabili al Ni-MH VARTA "Ready To Use"



€ 12,⁰⁰

Cod. HR03AAA800



€ 13,⁰⁰

Cod. HR6AA2100

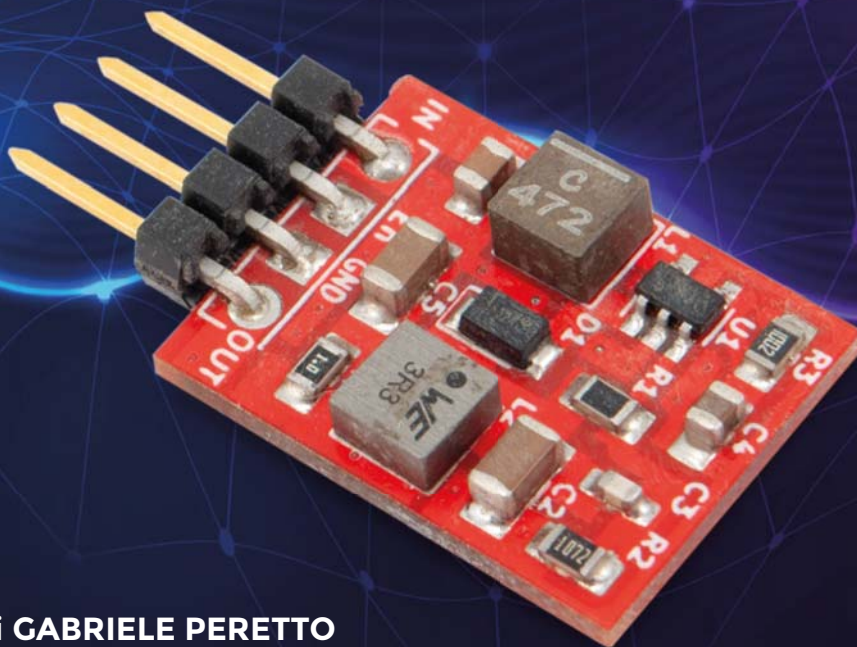


€ 13,⁸⁰

Cod. HR14C3000

Prezzi IVA inclusa.

BOOST CONVERTER IN FORMATO TO3-P



di **GABRIELE PERETTO**

Minuscolo ed efficiente, dotato di funzione True-Shutdown, può erogare fino a 5 W con un'efficienza compresa tra il 70 e l'80%.

A

nche se in commercio si trovano molti moduli alimentatori step up/boost, utilizzabili per le più svariate applicazioni, alcuni dei quali presentati sulle pagine della nostra rivista (fascicolo n° 248 di ottobre 2020), raramente si riesce a reperire un modulo che disponga di un efficiente controllo SHUT-DOWN per mettere in stand-by il modulo ed evitare inutili assorbimenti di corrente senza staccare il carico e/o la fonte di alimentazione primaria. Questa caratteristica è dovuta alla particolare struttura del circuito e si può mediare soltanto utilizzando qualche componente aggiuntivo. Analizzando lo schema semplificato di un alimentatore step-up (**Fig. 1**), si può chiaramente vedere, che anche se il MOS risulta interdetto, la tensione primaria prelevata per esempio da una batteria da 3,6V, può continuare ad alimentare il carico, attraverso l'induttore L e il diodo di recupero (Fast recovery) D.

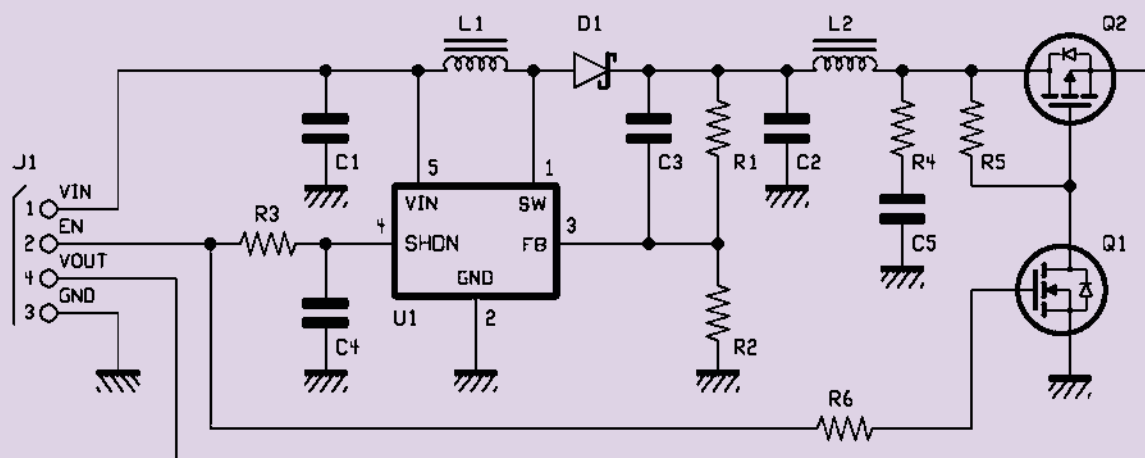


Fig. 1
Schema
concettuale di un
DC/DC converter
tipo Boost.

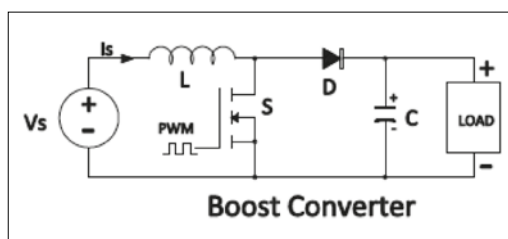


Fig. 2
Piedinatura
dell'integrato
utilizzato nel
progetto.

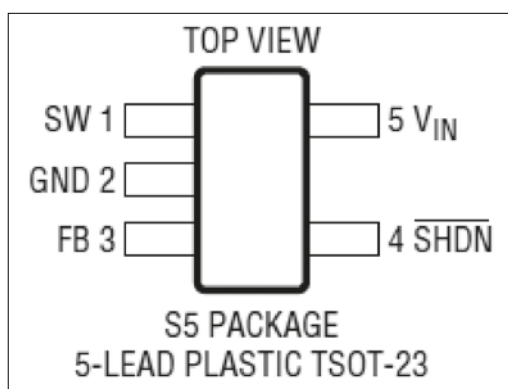


Tabella 1
Caratteristiche
elettriche
dell'integrato
LT1935.

Pertanto se il nostro circuito in funzionamento normale eroga 12V su un carico da 24 ohm, consumando 500 mA, quando viene spento il MOS, la tensione di 3,6V della batteria fluirà comunque attraverso L e D subendo una attenuazione di circa

Vin min	V	2.3	Vin	Vout	A
Vin max	V	16	3,3	5	1
Vout max	V	38	5	12	0,6
Sw. Freq	MHz	1.2	3	5	0,9
Efficiency	%	82-87	3,6	12	0,45

0,5V dovuta alla caduta di tensione (drop-out) del diodo e su un carico da 24 ohm si avrà comunque un assorbimento a vuoto, non desiderato, di 130 mA.

Per ragioni di spazio non possiamo approfondire il funzionamento di questo particolare dispositivo switching, quindi per i dettagli vi rimandiamo alla lettura dell'articolo sul fascicolo n° 248 di Elettronica In.

L'INTEGRATO LT1935

Per questa realizzazione, abbiamo scelto il circuito integrato **LT1935** della Linear Technology (Analog Device) che, anche se costa un po', offre in cambio notevoli prestazioni utilizzando di base soltanto 6 o 7 componenti passivi esterni ed essendo realizzato in contenitore SOT23-5, risulta molto compatto e versatile. In **Fig. 2** è visibile la piedinatura con le varie funzioni assegnate ad ogni piedino. Inoltre tutto il circuito può essere progettato e testato validamente utilizzando il programma (free) **LTspice** (scaricabile dall'indirizzo <https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html>) poiché questo componente è già precaricato nella libreria e risulta quindi molto facile testare varie configurazioni. Nella **Tabella 1**, sono elencati i principali parametri di funzionamento dichiarati dal costruttore e da noi verificati in fase progettuale mediante simulazione con lo strumento **LTspice**.

La compattezza di questo circuito integrato e i pochi componenti che richiede per funzionare, ci hanno permesso di realizzare un circuito piccolo come un MOSFET in package TO3-P, come si vede bene nella **Fig. 3**, dalla quale è possibile fare il confronto e vedere le dimensioni dell'integrato rispetto a un MOSFET.

SCHEMA ELETTRICO

Come abbiamo anticipato, tutto lo schema è stato progettato e testato mediante l'ausilio di *LTspice* per poi essere trasferito su programma CAD dedicato quale può essere EAGLE o KICAD, per la creazione del PCB.

Il circuito, come si può notare è molto semplice ed è realizzato veramente con pochi componenti esterni. Una particolare attenzione la dobbiamo comunque dedicare alla scelta di alcuni componenti critici. In particolar modo dobbiamo porre cura nella scelta dell'induttore **L1** e al diodo di recupero **D1**. Per quanto riguarda L1, dovremo scegliere un induttore di potenza per alta frequenza, con corrente di saturazione di almeno 2A e con resistenza DC inferiore a 100 mΩ. Una scelta oculata del valore dell'induttore si traduce in un'ondulazione di corrente (ripple) pari a 1/3 della corrente massima di commutazione: $L1 = 3 (Vin/Vout) \cdot (Vout - Vin)/(Imax \cdot f)$, dove $Imax$ è la corrente massima di commutazione di 2A e f è la frequenza di commutazione. Con duty-cycle di lavoro inferiore al 70%, si può utilizzare un valore leggermente inferiore in modo da poter utilizzarne uno dimensionalmente più piccolo. Per quanto riguarda il diodo D1, bisogna scegliere un diodo Schottky da 1A (meglio 2A) e con "tensione di reverse voltage" (V_{RRM}) di 20V. Per uscite con tensioni di 24 – 30V bisognerà ovviamente scegliere un diodo con V_{RRM} superiore intorno a 30-40V. Per quanto riguarda i condensatori, dovranno essere tutti a basso ESR (*equivalent series resistance*). Si raccomanda quindi l'utilizzo di condensatori ceramici con dielettrico X5R o meglio X7R. Il condensatore di ingresso C1, deve avere una capacità di almeno 4.7 uF e con tensione adeguata all'utilizzo. Lo stesso vale per il condensatore di uscita C2 che dovrà essere di almeno 22 uF e con tensione di lavoro di 35VL (se non si superano i 30 Vcc di uscita).

Ovviamente se il modulo viene utilizzato per tensioni di uscita da 5V, può essere utilizzato anche un condensatore con V_L inferiore, ad esempio 10VL. Passiamo ora ai due componenti relativi alla programmazione della tensione di uscita e che è affidata alle resistenze **R1** e **R2**. La formula di calcolo è la seguente:

$$V_{out} = 1.265V(1 + R1/R2)$$

Assumendo di fissare il valore di R2 (nel nostro caso 10,7 kohm), per calcolare R1, avremo:

$$R1 = [(V_{out}/1,265) - 1] \cdot R2$$

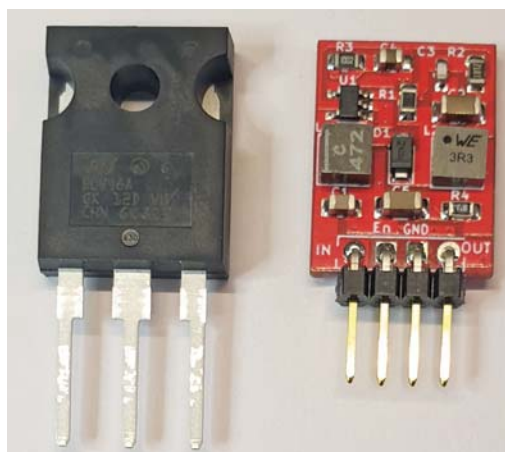
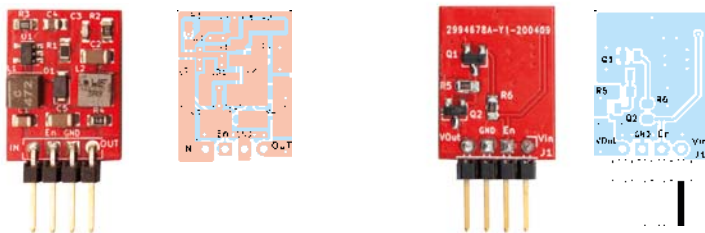


Fig. 3
Il nostro circuito
a confronto con
un MOSFET
TO3-P.

Quindi per realizzare un alimentatore con V_{out} da 12 volt abbiamo:

$$R1 = [(12/1,265) - 1] \cdot 10,7 = 8,486 \cdot 10,7 = 90,8K$$

che arrotondiamo a 91 kohm. Il valore di R2 deve essere scelto intorno ai 10K. Sostituendo per esempio R1 con una resistenza di valore di 33K, si ottiene in uscita 5,2 V. Va da sé che utilizzando un trimmer da 100K si potrebbe prelevare in uscita da un minimo di 4 V a circa 13,1V in modo continuo. Noi abbiamo comunque, volutamente scelto di inserire resistori di valore fisso, perché riteniamo più affidabile nel tempo la stabilità di uscita e poiché una volta inserito nel circuito, l'alimentatore non è più soggetto a variazioni di tensione che potrebbero anzi influire negativamente sul sistema. Il successivo filtro composto da L2, R4 e C5 è un ulteriore miglioramento, volutamente inserito per ridurre ulteriormente il ripple (ondulazione) di uscita, che comunque risulta già di per sé molto buono aggirandosi sui 30+35 mV. Infine, i due MOSFET Q1 e Q2 insieme alle resistenze di polarizzazione R5 e R6 servono per ottenere lo SHUT-DOWN che in questo caso risulta "reale" e cioè con assorbimento "zero" del circuito anche sotto carico. In questo modo possiamo efficientemente salvaguardare la vita della batteria di alimentazione. In pratica, collegando il terminale "En" (enable) dalla tensione positiva della batteria o ad altra fonte di alimentazione come ad esempio un microcontrollore, purché con massa comune al circuito e con tensione minima di 1,8V e massima di 15V, si ottiene il funzionamento del circuito, ovvero l'erogazione della tensione di uscita. Il mosfet Q1 può essere di qualsiasi tipo a canale N per piccoli segnali, mentre il MOSFET Q2 a canale P dovrà essere in grado di erogare almeno



Elenco Componenti:

R1: 90,9 Kohm - 0,1 % (33 Kohm per versione 5 V)	D1: DFLS220L (alt. SBR2A40P1)
R2: 10,7 Kohm - 0,1 %	Q1: BSS138
R3: 10 kohm	Q2: Si2369DS
R4: 1 ohm	L1: 4,7 μ H - 2 A
R5, R6: 47 kohm	L2: 3,3 μ H 2,8 A
C1: 4,7 μ F - 35 VL ceramico	U1: LT1935
C2: 22 μ F - 35 VL ceramico	J1: strip maschio 4 vie 90°
C3: 47 pF ceramico	
C4: 1 μ F - 35 VL ceramico	Varie
C5: 10 μ F - 35 VL ceramico	- Circuito stampato S1637 (15x20 mm)

2A ed avere una resistenza R_{DSon} intorno ai 30 – 40 m Ω (milliOhm). Lasciando invece il piedino aperto, scollegato dal positivo, il sistema va automaticamente in SHUT-DOWN sull'uscita non vi sarà alcuna tensione e quindi nessuna erogazione di corrente.

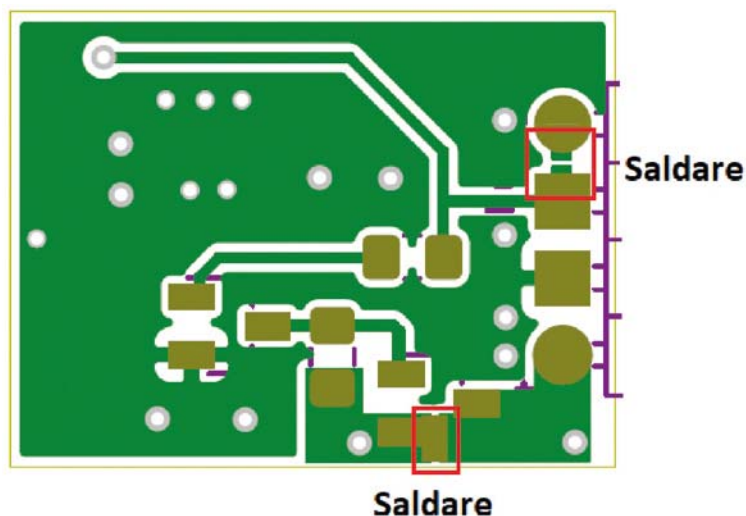
NOTA: se il circuito alimentato dovesse risultare sempre sotto tensione e quindi non servisse utilizzare lo SHUT-DOWN, i componenti R5, R6, Q1 e Q2, possono NON essere montati. In questo caso il terminale En va collegato direttamente al positivo (rispettando sempre un minimo di 1,8V e un massimo 15V). Nel paragrafo dedicato alla realizzazione, spiegheremo come effettuare le varianti. Per quanto riguarda la tensione di ingresso, la

minima applicabile è 3Vcc, ma in questo caso non potremo prelevare in uscita più di 250 mA, mentre utilizzando una batteria ricaricabile al litio da 3,6 V (singola cella) riusciremo a prelevare circa 300 mA, ma dobbiamo comunque considerare che se a carica completa potremmo avere 4,2V, a batteria quasi scarica arriveremo ad avere circa 3,3V. Con questa tensione di alimentazione si può ottenere in uscita max. 280 mA. Alimentando il circuito a 5V ad esempio tramite una porta USB, potremo tranquillamente disporre di 450 mA ottenendo così un'efficienza superiore all'80% e una erogazione di 5,4 Watt. Notate che nel caso utilizzassimo un alimentatore USB, esso dovrebbe poter erogare almeno 2 ampere, ovvero 10 W.

REALIZZAZIONE PRATICA

Dopo la descrizione dello schema elettrico con relativi dettagli funzionali, passiamo ora alla realizzazione fisica della scheda. Tutto il circuito è realizzato su una basetta di soli 15 x 20 x 1,2 mm (spessore di 1,6 mm va comunque bene) a doppia faccia con fori metallizzati, dove con i componenti montati lo spessore totale non supera i 5 mm. In questo progetto, abbiamo volutamente scelto un compromesso tra dimensioni e potenza erogata, pertanto non riusciremo a sfruttare pienamente le caratteristiche dichiarate e ottenute dalla simulazione (elencate nella **Tabella 1**) ottenendo comunque valori di circa il 70% rispetto a quanto considerato. Per ottenere i valori teorici dichiarati ad erogazione continua bisogna costruire un CS più grande oppure adottare una soluzione ad almeno 4 layer, ma questo comporta affrontare una

Fig. 4
I ponticelli da realizzare con lo stagno, se non viene utilizzato il terminale En.



maggiore difficoltà di realizzazione ed un costo decisamente elevato e questo non era il nostro scopo. Tutta la scheda è stata realizzata con componenti SMD, al fine non solo di ridurre le dimensioni, ma soprattutto di posizionare correttamente i componenti per evitare perdite e malfunzionamenti del circuito. Tutti i layout degli alimentatori switching devono essere accuratamente progettati poiché le frequenze di commutazione spesso superano 1MHz, come nel nostro caso. Particolare attenzione bisogna porre soprattutto nel posizionamento dell'induttore L1 e del diodo di recupero D1, i quali vanno sempre posizionati alla distanza minima possibile dai piedini Vin e SW, come pure il condensatore di ingresso C1. Inoltre, per favorire la dissipazione termica risulta utile lasciare quanto più rame possibile collegato alla massa (GND) del circuito. Per un risultato ottimale conviene quindi realizzare il circuito stampato mediante fotoincisione e allo scopo, come sempre, sul nostro sito potrete trovare i files gerber già pronti da utilizzare. In alternativa il servizio di Futura Elettronica PCBPRODUCTION può offrirvi di realizzare il PCB. Il montaggio comunque non risulta particolarmente difficoltoso e con l'aiuto di una buona lente di ingrandimento oppure di un microscopio digitale per riparazioni e con l'ausilio di uno stabile supporto, si può procedere alla saldatura dei vari componenti. L'unica difficoltà sta nel saldare i due induttori, i quali andrebbero saldati per primi in un apposito fornello a rifusione oppure con l'aria calda. A tal proposito si potrebbe procedere mettendo un po' di pasta saldante per smd (stagno a basso punto di fusione) sulle piazzole posizionando poi i due induttori in modo perfettamente centrato. Infine procedere con l'aria calda fino alla saldatura. Tutti gli altri componenti possono essere saldati manualmente avendo sempre cura di utilizzare un ottimo filo di stagno con diametro di 0,5/0,7 mm max. e saldatore a bassa tensione (24V) a punta fine, facendo attenzione ad utilizzare sempre quantitativi minimi indispensabili di stagno, evitando gocciolamenti e cortocircuiti tra i piedini del circuito integrato.

Nel caso non vi interessasse utilizzare la funzione di "SHUT-DOWN" per mettere in stand-by il circuito, ad esempio tramite un microcontrollore, potremo evitare di montare tutti i relativi componenti: R5, R6, Q1 e Q2 che sono stati volutamente inseriti sul retro della scheda proprio a tale scopo, come potete riscontrare nel piano di montaggio pubblicato in queste pagine. Bisogna però tenere in considerazione che in questo caso bisognerà

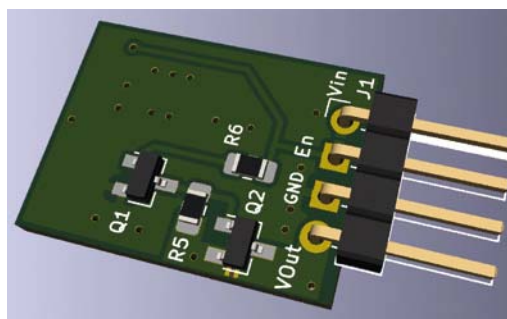


Fig. 5
Vista
posteriore
del modulo
completo
di pin-strip
a 90° con
evidenziata la
piedinatura.

collegare con una goccia di stagno le due piazzole poste sotto Q2 e collegare (sempre con una goccia di stagno) il piedino En con il piedino Vcc, come indicato nella **Fig. 4**, che evidenzia le zone con i collegamenti da effettuare. Infine, potremo montare il connettore a 4 pin a passo 2,54 mm negli appositi fori in fondo alla scheda, ottenendo un risultato come quello mostrato nella **Fig. 5**; noi abbiamo scelto un tipo a 90° per posizionare il circuito in posizione verticale sulla scheda da alimentare, ma nulla vieta di montare un connettore diritto per un posizionamento orizzontale. A questo punto il lavoro è terminato e non resta che collaudare il modulo.

COLLAUDO

La piedinatura a passo 2,54 mm del connettore di uscita, oltre che essere uno standard di montaggio, ci aiuta anche nel collaudo, poiché è sufficiente inserire la scheda in una bread-board per prototipi comunemente presente in tutti i laboratori degli hobbisti; dopo aver collegato i fili di alimentazione, di massa e di uscita, si può iniziare

**Il prototipo
con pin strip
a 90°.**

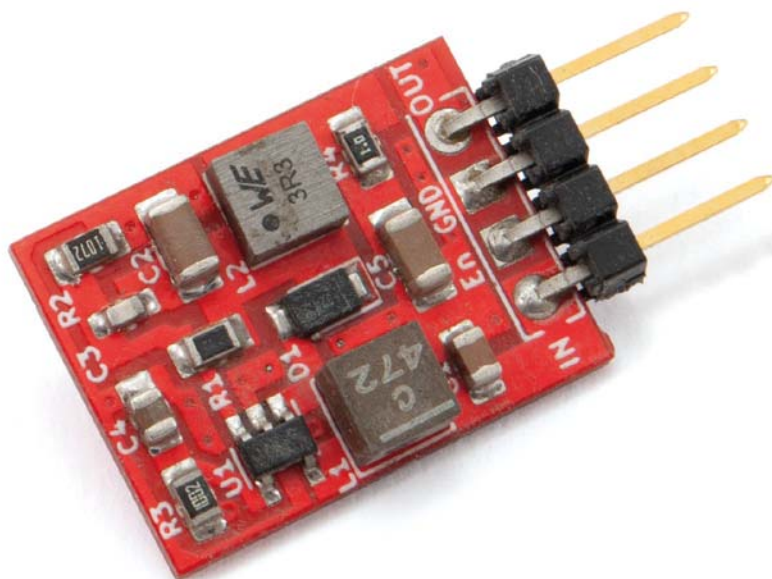


Tabella 2
Risultati dei test sul
prototipo.

Vin	Vout S/C	I- S/C (mA)	Vout	Iin (A)	Iout (A)	Ripple (mV)	Eff. %	Temp. °C
3	12,1	6,2	11,9	1,45	0,25	20	68,4	61
3,3	12,1	6,3	11,9	1,4	0,28	22	72,1	59
3,6	12,09	6,4	11,92	1,3	0,3	21	76,4	60
5	12,08	8,7	11,82	1,25	0,42	19	79,4	61
3	5,2	9	5,06	1,32	0,55	18	70,3	60
3,3	5,2	8	5,03	1,25	0,63	18	76,8	60
3,6	5,2	8	5,03	1,2	0,7	19	81,5	59

il collaudo. Se il circuito è stato montato nella sua versione completa, ovvero per poter funzionare con il controllo "SHUT-DOWN", ricordiamoci sempre di collegare il terminale "En" al positivo di alimentazione, tramite uno spezzone di filo. Nel caso contrario non vi sarà nessuna tensione sul terminale OUT, come ovviamente deve essere. Questo controllo va comunque fatto, perché se trovassimo qualsiasi tensione in uscita senza aver abilitato il piedino "En" significa che Q1 e/o Q2 sono in corto oppure c'è qualche altra anomalia di montaggio. Eseguito questo primo test, possiamo provare a collegare un carico di uscita per valutarne le caratteristiche. Noi abbiamo utilizzato uno strumento di simulazione di carico elettronico e i dati rilevati sono riportati nella **Tabella 2**.

I test sono stati eseguiti con 4 valori di tensione, rispettivamente di: 3V – 3,3V – 3,6V e 5V per il modulo con uscita a 12V e di: 3V – 3,3V e 3,6V per il modulo con uscita a 5V. Nella seconda e terza colonna sono riportati i valori di tensione e assorbimento a vuoto (S/C), mentre nelle successive colonne vi sono i parametri sotto carico. Come abbiamo accennato precedentemente, avendo privilegiato la dimensione della scheda, abbiamo ottenuto valori prossimi al 70% rispetto ai valori teorici, considerando di non superare mai i 60÷62 °C a temperatura ambiente (23 °C) e misurati con termocoppia posizionata sul retro della scheda; poiché il datasheet del componente indica una temperatura di esercizio massima di 85 °C. La tensione di ondulazione come si può notare si attesta sempre sui 20 mV e quindi l'alimentazione risulta veramente stabile e pulita e adatta ad alimentare anche i circuiti più critici e sensibili, sempre nel rispetto della corrente massima erogabile in sicurezza.

CONCLUSIONI

Con questo abbiamo terminato il nostro progetto e sperando di essere riusciti nell'intento di dare tutte le informazioni utili, rimandiamo a ciascuno

di voi la valutazione di utilizzo secondo le proprie esigenze. Chi volesse provare ad aumentare un po' l'efficienza del modulo potrebbe saldare un piccolo lamierino di rame di 15x20 mm piegato a L su una estremità per circa 3-4 mm, saldandolo poi sul bordo superiore del lato posteriore della scheda. Se non vi sono esigenze di spazio si può progettare e realizzare un PCB più grande, in modo da avere più rame di dissipazione sul piano di massa. Ricordiamo solo di verificare sempre la temperatura della scheda in modo che non superi mai i 60÷65 °C ed evitare quindi di surriscaldare eccessivamente e danneggiare i vari componenti. Oltretutto se superiamo questa temperatura, avremo una brusca caduta di tensione in uscita. Riassumendo questi moduli sono in grado di erogare tra i 3 e i 5 watt massimi a seconda della tensione di alimentazione e della tensione di uscita e con un'efficienza variabile tra il 70 e l'82%. Per ora concludiamo qui rimandando l'appuntamento per un prossimo progetto di un modulo di pari dimensioni in tecnologia SEPIC ovvero BUCK-BOOST ma a tensione non invertita, in grado di mantenere una tensione sempre costante di 3,6V, partendo da una batteria al litio da 3,6V a prescindere dallo stato di carica massima di 4,2V a quella quasi scarica di 3V. □



Cosa occorre?

Del progetto è disponibile il file Gerber sul sito www.elettronica.in.it, alla pagina di questo numero. Il PCB può essere realizzato tramite il servizio PCBPRODUCTION disponibile sul sito www.futurashop.it.

Il sito di riferimento è:

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

La TV cambia? Non cambiarla!

Arriva la nuova TV digitale, acquista il decoder senza cambiare la TV!

Decoder TV digitale terrestre DVBT, DVBT2, H.265/HEVC



Cod. GS240T2
€ 26,90

**NUOVO
MODELLO**

DVBT/T2
TERRESTRIAL

**H.265
HEVC**

**FULL HD
1080**

USB

HDMI **mpeg** **XVID**
VIDEO VIDEO

GoSat GS240T2 è un decoder digitale terrestre di nuova generazione compatibile con il nuovo standard di trasmissione DVBT, DVBT2 H.265/HEVC. L'apparecchio può ricevere comunque trasmissioni nei vecchi standard DVB-T H.264 e MPEG-4. Il dispositivo offre una perfetta qualità delle immagini in Full HD 1080p. Grazie alla porta USB, disponibile sulla parte frontale, e alla funzione PVR (Personal Video Recorder), permette la registrazione di programmi TV e radio su chiavetta USB o su hard disk esterno. Può essere inoltre connesso a internet tramite il connettore RJ45 per utilizzare servizi come YouTube e guardare canali IPTV (liste con estensioni M3U). Il telecomando in dotazione, oltre ad avere tutte le funzioni per il controllo del ricevitore, ha la funzione di apprendimento e può essere utilizzato anche per altri dispositivi (TV, ecc.). Il pannello frontale ha un display che mostra l'ora o il numero del programma selezionato.



Prezzi IVA inclusa.

Il tuo telecomando TV è rotto?

Sostituiscilo con i telecomandi universali a infrarossi compatibili con le TV delle maggiori marche prodotte dal 2000 ad oggi. Dispongono di tutte le funzioni del telecomando originale. Subito utilizzabili!

€ 5,90
OGNI PEZZO

**PRONTI
ALL'USO!**



Cod. TVIR-SAMSUNG



Cod. TVIR-SONY



Cod. TVIR-PHILIPS



Cod. TVIR-LG



Cod. TVIR-PANASONIC

DA PROGRAMMARE

Per comandare fino a due dispositivi audio/video.

€ 9,00
Cod. TVFREEDOM2



DISPONIBILE ANCHE 1 in 1
Cod. TVFREEDOM1

€ 8,50

**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

Where are you?

Localizzatori GPS/GSM/GPRS per tutte le esigenze in grado di monitorare in tempo reale l'auto, la moto, la bicicletta, i bagagli e tanto altro.



LOCALIZZATORE GPS/GSM/GPRS CON ASCOLTO AMBIENTALE

Segui in tempo reale, tramite browser e APP gratuita, il percorso di veicoli e persone! Oltre a visualizzare i movimenti del tuo veicolo in tempo reale, potrai controllare la velocità, la direzione e i punti di sosta. Il localizzatore permette anche di gestire una flotta di autoveicoli con un unico account. Dispone di batteria ricaricabile agli ioni di litio da 6.000 mA che fornisce un'autonomia prolungata nel tempo (fino a 30 giorni in standby). Grazie al potente magnete di cui è dotato, è possibile fissarlo su superfici metalliche.



cod. FR763

€ 78,00



MICRO LOCALIZZATORE SATELLITARE GPRS/GSM

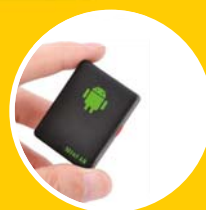
Tutto sotto controllo con il nuovo micro localizzatore GPRS/GSM. Per attivarlo basta inserire una SIM di qualsiasi operatore; inviando un SMS potrai attivare l'ascolto ambientale oppure ricevere un link con le coordinate che indicano la posizione in tempo reale del localizzatore. Dispone di batteria agli ioni di litio ricaricabile. Ideale per tenere sotto controllo la posizione di autoveicoli, persone, cose, animali domestici, bagagli, ecc. La confezione comprende: il micro localizzatore, il cavetto USB-micro USB per la ricarica della batteria e le istruzioni in italiano.

Dimensioni: 43,2x32x14 mm.
Peso: 20 grammi.



cod. A8TRACK

€ 14,00



LOCALIZZATORE TASCABILE GPS/GSM/GPRS

Ideale per la localizzazione personale e veicolare. Può inviare le proprie coordinate (latitudine e longitudine) via SMS verso telefoni cellulari, oppure, tramite tecnologia GPRS, ad un computer opportunamente configurato. Dispone di microfono integrato e batteria ricaricabile.



cod. TK102

€ 54,00

APRICANCELLO GSM MODULARE

di ANDREA MISURACA



Basato su Mercury System, può essere comandato tramite chiamate telefoniche e può essere configurato tramite un numero master impostato alla prima attivazione.

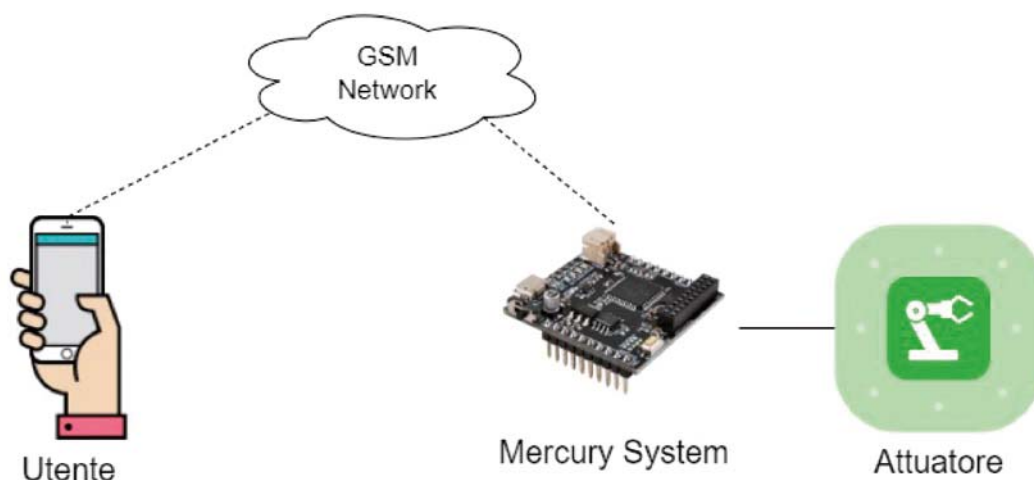


N

egli ultimi anni l'interesse verso la domotica è aumentato in maniera esponenziale; ciò è stato possibile grazie a una serie di fattori quali la diminuzione dei prezzi dei dispositivi intelligenti e la facilità con cui è possibile creare ed interagire con una tecnologia

divenuta ormai di uso comune. L'avvento dell'Internet of Things (IoT) ha permesso la connessione di qualsiasi dispositivo domestico, dagli elettrodomestici (lavatrici, lavastoviglie, forni, ecc.) a strumenti che erano impensabili fino a qualche anno fa, come gli assistenti vocali, pronti a dirci che tempo farà domani o tra quanto passerà l'autobus alla fermata sotto casa. I dispositivi IoT ci hanno semplificato la vita, anche perché sono facilmente controllabili tramite un'applicazione sul nostro smartphone, ma, come si evince dalla stessa definizione, internet è una costante imprescindibile.

→ **Fig. 1**
Schema
concettuale del
sistema.



Eppure, in Italia così come in tante altre parti del mondo, sono ancora molte le zone non coperte da una connessione Internet, in particolar modo le aree rurali di campagna o montagna. Ci sono poi posti in cui la connessione esiste, ma si vuole connettere un dispositivo troppo lontano dal proprio punto di accesso ad internet (il router è troppo distante), o semplicemente non si ha un abbonamento con un provider, come avviene spesso nelle seconde case usate per le vacanze.

Per aggirare il problema vogliamo proporre la realizzazione di un dispositivo intelligente in grado di interagire con la rete GSM dei nostri cellulari, quindi dotato di una normale SIM card e potrà essere controllato mediante chiamate e SMS. Il tipico caso d'uso è quello di un apricancello che si attiva quando l'utente effettua una chiamata al numero del dispositivo, ma tale operazione consiste semplicemente nell'attivazione di un relé, quindi l'utente sarà libero di utilizzarlo per controllare un qualsiasi dispositivo connesso alla rete elettrica. Per la realizzazione del progetto andremo ad utilizzare il Mercury System, un sistema embedded semplice, facile da usare e particolarmente adatto ad applicazioni IoT di questo tipo, anche per via del Modem GSM/GPRS facilmente interfacciabile. Utilizzeremo la BB110 come core di elaborazione, il modem GSM/GPRS FT1308M, la SB111 che contiene il relé, la EB110 per connettere la BB con la SB. Opzionalmente è possibile usare la EB111 per aggiungere la power board PB110, in modo da poter alimentare il sistema con un alimentatore separato e non tramite la connessione USB.

IL PROGETTO

Il sistema che andremo a realizzare per implementare l'apricancello basato su Mercury System è

descritto concettualmente dal disegno riportato in **Fig. 1**. L'architettura è molto semplice: il dispositivo dotato di SIM card si registra sulla rete e l'utente dal proprio cellulare vi può interagire: con una chiamata, il relé si attiverà per qualche secondo (consentendo ad esempio l'apertura di un cancello). Nonostante la semplicità dell'obiettivo, è necessario implementare dei meccanismi che consentano di avere un certo grado di sicurezza per evitare ad esempio che chiunque chiami al numero del dispositivo possa aprire il nostro cancello. L'idea è quindi di utilizzare la memoria interna alla Base Board per memorizzare un numero di telefono primario, in grado di attivare il relé con una chiamata e di salvare ulteriori numeri in memoria, cui consentire l'accesso al dispositivo. Il numero primario (master) potrà poi revocare un permesso ed effettuare un reset completo del sistema. L'interazione tra il numero primario e il nostro dispositivo potrà avvenire tramite l'invio di semplici SMS. Si noti anche come non è necessario avere alcun tipo di piano tariffario particolare nella SIM del dispositivo, visto che non deve rispondere né agli SMS né alle chiamate ricevute, che verranno sempre rifiutate.

SETUP HARDWARE

Come già menzionato, l'hardware del progetto consta di alcuni moduli Mercury System ospitati sulla piastra base; più esattamente, gli elementi della piattaforma qui utilizzati sono:

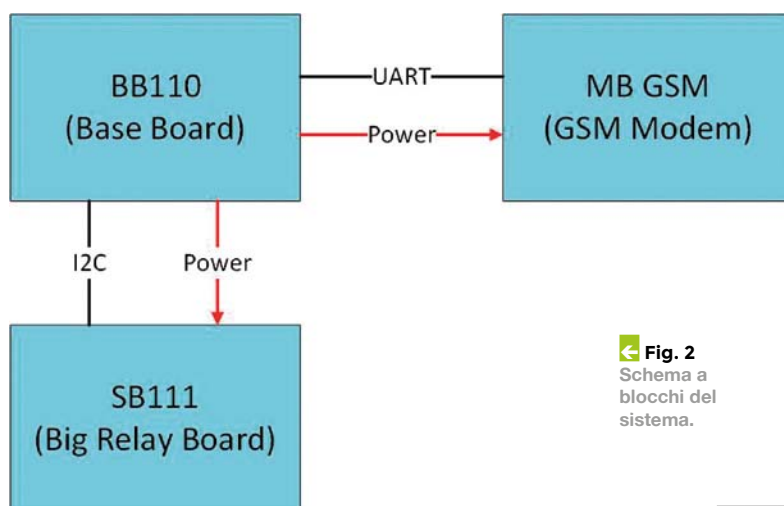
- BB110 - Base Board Model A: unità logica del sistema in cui gira l'applicazione;
- MB GSM - Modem Board GSM: scheda contenente il modulo GSM/GPRS, lo slot per inserire la SIM card e consentire l'interazione con chiamate e messaggi;
- Antenna esterna MMCx/M: antenna per consen-

- tire l'accesso alla rete da parte del modem;
- SB111 – Slave Board Big Relay: scheda contenente il relé;
 - EB110 – Expansion Board Dual: scheda di espansione per consentire il collegamento della BB110 alla SB111.

Opzionalmente sono previsti:

- PB110 – Power Board: per consentire opzionalmente di alimentare il sistema tramite alimentatore piuttosto che via USB;
- EB111 – Expansion Board Quad: per rimpiazzare la EB110 e consentire il collegamento della BB110 alla SB111 e al PB110.

In **Fig. 2** è possibile vedere lo schema della nostra configurazione hardware con evidenziati i bus di comunicazione.



← **Fig. 2**
Schema a
blocchi del
sistema.

Il modem MB GSM utilizzato è connesso alla Base Board tramite UART. La board contenente il relé SB111 è invece connessa alla BB110 tramite I²C. Sempre tramite I²C è connessa la EEPROM 24LC16B già saldata sulla BB110.

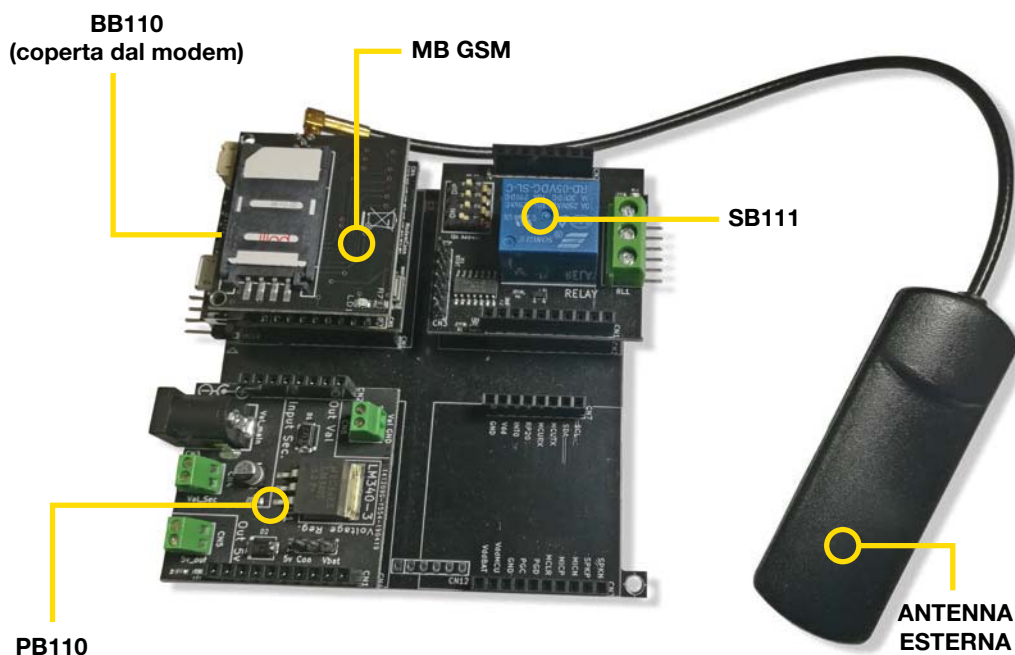
Faremo anche uso di due LED presenti nella BB110 e connessi tramite GPIO che ci consentiranno di avere un effetto visivo sulla riuscita o meno dei comandi inviati al dispositivo.

Per quanto riguarda l'alimentazione da fornire alla board ci sono due alternative, o la si alimenta tramite micro USB, ad esempio con un classico caricabatteria di uno smartphone, o tramite la PB110 alla quale si può connettere un alimentato-

re 12V. Una volta assemblati tutti i componenti il Mercury System dovrebbe apparire come in **Fig. 3**. È necessario ricordarsi di impostare l'indirizzo della scheda SB111, che secondo l'implementazione descritta nel prossimo paragrafo deve essere uguale ad 1; questo viene ottenuto disponendo i dip-switch come mostrato nella **Fig. 4**.

IMPLEMENTAZIONE SOFTWARE

Per quanto riguarda l'implementazione del software embedded, possiamo iniziare con la definizione di alcuni requisiti funzionali del dispositivo riguardanti il numero primario (ossia il numero telefonico



← **Fig. 3**
Disposizione
dei moduli
nel prototipo.

Fig. 4
Impostazione
dei dip-switch.



considerato "master") il quale:

- deve essere impostato tramite USB, utilizzando un'apposita applicazione, o nei primi 180 secondi al primo avvio della board attraverso una chiamata. Il numero primario viene quindi memorizzato in memoria;
- può aggiungere altri numeri in memoria;
- può rimuovere altri numeri dalla memoria;
- può effettuare il reset del dispositivo riportandolo allo stato iniziale.

L'implementazione della nostra applicazione sarà basata sul Mercury Framework, un software che contiene al suo interno i driver di basso livello del Mercury System e ci consente di accedere all'hardware tramite apposite interfacce. È possibile effettuare il download al seguente link: https://store.open-electronics.org/Development_Tools/Mercury/BaseBoardforMercurySystem_7305-BB110, ed una volta installato possiamo procedere con la nostra implementazione. Per maggiori dettagli sul framework è possibile consultare il numero 234 di Elettronica In dove è stato ampiamente descritta la piattaforma Mercury.

CONFIGURAZIONE DI SISTEMA

Alla creazione di un nuovo progetto è necessario configurare il sistema aggiornando il file *sys_cfg.h* dove possiamo impostare il tipo di modem utilizzato (nel nostro caso GSM) tramite la define `MODEM_USED`:

```

/*****
* Modem Type Config
*****/

#define NO_MDM                0u
#define GSM_GPRS_MDM          1u
#define BT_MDM                 2u

```

```

#define WIFI_MDM                3u

#define MODEM_USED              GSM_GPRS_MDM

```

Possiamo poi procedere con il settaggio del tempo con cui il nostro task applicativo verrà chiamato ciclicamente; nel file *app.h* possiamo specificare il valore in millisecondi tramite la seguente define:

```

/*****
* EXPORTED Defines
*****/

/* Task period in ms */
#define MY_APP_TASK_PERIOD_MS  ((UINT16)(100))

```

ARCHITETTURA SOFTWARE

Grazie al Mercury Framework è possibile procedere con lo sviluppo della nostra applicazione senza preoccuparci di implementare i driver di basso livello. Il framework non solo implementa i driver, ma fornisce anche delle interfacce tramite cui è possibile interagire con le periferiche. Si crea così un'architettura a strati in cui abbiamo, partendo dal basso, l'hardware, il Mercury Framework, ed infine il nostro software applicativo. L'applicativo contiene una macchina a stati principale implementata nel file *app_main.c*, ed una serie di macchine a stati necessarie per l'implementazione dei vari comandi (attivare il relé, effettuare un reset, aggiungere un numero, etc.).

Per ognuno dei comandi vi è un file C dedicato:

- *cmd_init.c*: per inizializzare il dispositivo
- *cmd_add.c*: implementazione del comando per aggiungere un numero in memoria
- *cmd_del.c*: per eliminare un numero dalla memoria
- *cmd_res.c*: per effettuare un reset del dispositivo
- *cmd_trigger.c*: per attivare il relé

Vi è poi il file *utils.c* che contiene una serie di funzioni che sono comuni a più comandi, come ad esempio la verifica del numero del chiamante o del numero da cui si è ricevuto un SMS, il controllo della presenza di un numero in memoria, il salvataggio di un numero e l'accensione intermittente di un LED per un certo numero di secondi.

INIZIALIZZAZIONE

Il file *cmd_init.c* contiene l'implementazione delle funzioni che vengono eseguite all'avvio della nostra applicazione. Ci sono delle operazioni comuni ad ogni avvio, come ad esempio l'inizializzazione del modem, e delle operazioni che dipendono dallo

stato del dispositivo.

Ad esempio, alla prima accensione, o eventualmente dopo un reset, la memoria del dispositivo è vuota, ed è necessario quindi impostare un numero primario dal quale effettuare poi le operazioni. Quest'operazione deve essere effettuata solo se la memoria è vuota. Per determinare se la memoria è vuota o meno possiamo utilizzare un preambolo, scrivendo ad esempio uno specifico pattern, come 4 caratteri ASCII, nei primi 4 byte della EEPROM. Se leggendo i primi 4 byte della memoria notiamo che sono uguali al pattern scelto possiamo proce-

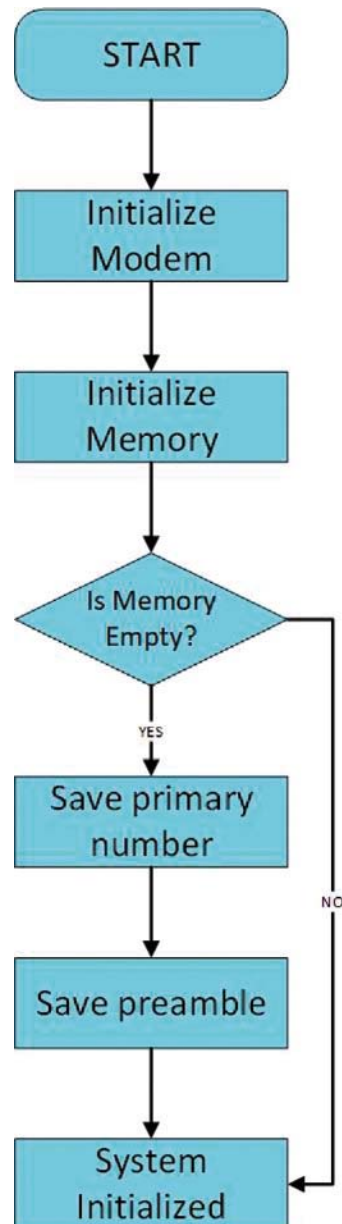
dere con la prossima fase, altrimenti procediamo con l'inizializzazione.

Nel caso non ci sia il preambolo scelto nei primi 4 byte, possiamo metterci in attesa del numero primario, ricevuto o tramite USB o tramite una semplice chiamata. Per ragioni di sicurezza è bene rimanere in attesa della chiamata solo per un tempo limitato, ad esempio 3 minuti, magari notificando lo stato di attesa tramite un LED intermittente, che diventa fisso una volta terminata.

È possibile vedere graficamente il comportamento descritto nel diagramma di flusso proposto in

▼ Listato 1

```
bool initFsm(bool* isCmdSuccessfull)
{
    static initFsmStates currentState = INIT_FSM_MODEM;
    bool isInitialized = false;
    switch (currentState)
    {
        case INIT_FSM_MODEM:
            if (initGprsModem())
            {
                currentState = INIT_FSM_EEPROM;
            }
            break;
        case INIT_FSM_EEPROM:
            if (initEeprom())
            {
                currentState = INIT_FSM_WAIT_CMD;
                setupCallTicks = 0;
            }
            break;
        case INIT_FSM_WAIT_CMD:
            if (isMemoryEmpty)
            {
                if (waitSetupCall())
                {
                    saveNumberInMemory(MASTER_NUMBER_ADDRESS, GetCallerNumber());
                    currentState = INIT_FSM_PREAMBLE;
                }
                if (detectUsbNumber(isCmdSuccessfull))
                {
                    currentState = INIT_FSM_PREAMBLE;
                }
            }
            else
            {
                currentState = INIT_FSM_COMPLETE;
            }
            break;
        case INIT_FSM_PREAMBLE:
            savePreamble();
            currentState = INIT_FSM_COMPLETE;
            break;
        case INIT_FSM_COMPLETE:
            Led_SetLedStatus(LED_1, LED_STS_ON);
            currentState = INIT_FSM_MODEM;
            isInitialized = true;
            break;
        default:
            break;
    }
    return isInitialized;
}
```



← Fig. 5
Flow-chart
dell'inizializzazione.

Fig. 5. Una possibile implementazione in software è quella proposta nel **Listato 1**.

COMANDO AGGIUNGI NUMERO

Usando un pattern simile possiamo implementare la macchina a stati per il comando che ci consente di aggiungere nuovi numeri di telefono nella memoria interna del dispositivo. L'aggiunta di un numero avviene via SMS, ma per soddisfare i requisiti precedentemente definiti è necessario effettuare

un controllo sul mittente del messaggio: infatti solo il numero primario ha la possibilità di effettuare questo comando. Dobbiamo poi controllare che il numero non sia già presente in memoria (utilizzeremo la funzione *isNumberInMemory* definita in *utils.c*). Possiamo poi procedere con la memorizzazione del numero. È possibile trovare a lato una possibile implementazione del diagramma di flusso proposto dalla **Fig. 6**. A questo diagramma di flusso corrisponde il codice contenuto nel **Listato 2**.

▼ Listato 2

```
bool addCmdFsm(uint8_t* receivedNumber, uint8_t* smsText, bool* isCmdSuccessfull)
{
    static addFsmStates currentState = ADD_FSM_CHECK_MASTER;
    bool isComplete = false;
    uint8_t numberInMemory = 0;
    uint8_t masterOpResult = OP_FAILED;
    uint8_t searchNumber = OP_FAILED;
    switch (currentState)
    {
        case ADD_FSM_CHECK_MASTER:
            *isCmdSuccessfull = false;
            masterOpResult = isMasterNumber(receivedNumber);
            if (masterOpResult == OP_SUCCESS)
            {
                currentState = ADD_FSM_CHECK_DUPLICATED;
            }
            else if (masterOpResult == OP_FAILED)
            {
                currentState = ADD_FSM_COMPLETE;
            }
            break;
        case ADD_FSM_CHECK_DUPLICATED:
            searchNumber = isNumberInMemory(smsText + TEXT_OFFSET, &numberInMemory);
            if (searchNumber == SEARCH_FSM_NOT_FOUND)
            {
                currentState = ADD_FSM_SAVE_NEW;
            }
            else if (searchNumber == SEARCH_FSM_FOUND || searchNumber == SEARCH_FSM_ERROR)
            {
                currentState = ADD_FSM_COMPLETE;
            }
            break;
        case ADD_FSM_SAVE_NEW:
            searchNumber = findEmptySpot(&numberInMemory);
            if (searchNumber == SEARCH_FSM_NOT_FOUND)
            {
                saveNumberInMemory(numberInMemory, smsText + TEXT_OFFSET);
                currentState = ADD_FSM_COMPLETE;
                *isCmdSuccessfull = true;
            }
            else if (searchNumber == SEARCH_FSM_FOUND || searchNumber == SEARCH_FSM_ERROR)
            {
                currentState = ADD_FSM_COMPLETE;
            }
            break;
        case ADD_FSM_COMPLETE:
            currentState = ADD_FSM_CHECK_MASTER;
            isComplete = true;
            break;
        default:
            break;
    }
    return isComplete
}
```



▼ Listato 3

```
bool delCmdFsm(uint8_t* receivedNumber, uint8_t*
smsText, bool* isCmdSuccessfull)
{
    static delFsmStates currentState = DEL_FSM_CHECK_
MASTER;
    bool isComplete = false;
    uint8_t numberInMemory = 0;
    uint8_t masterOpResult = OP_FAILED;
    uint8_t searchNumber = OP_FAILED;
    switch (currentState)
    {
        case DEL_FSM_CHECK_MASTER:
            *isCmdSuccessfull = false;
            masterOpResult = isMasterNumber(receivedNumber);
            if (masterOpResult == OP_SUCCESS)
            {
                currentState = DEL_FSM_REMOVE_NUMBER;
            }
            else if (masterOpResult == OP_FAILED)
            {
                currentState = DEL_FSM_COMPLETE;
            }
            break;
        case DEL_FSM_REMOVE_NUMBER:
            searchNumber = isNumberInMemory(smsText + TEXT_
OFFSET, &numberInMemory);
            if (searchNumber == SEARCH_FSM_FOUND)
            {
                saveNumberInMemory(numberInMemory, emptyNumber);
                currentState = DEL_FSM_COMPLETE;
                *isCmdSuccessfull = true;
            }
            else if (searchNumber == SEARCH_FSM_NOT_FOUND ||
searchNumber ==
SEARCH_FSM_ERROR)
            {
                currentState = DEL_FSM_COMPLETE;
            }
            break;
        case DEL_FSM_COMPLETE:
            isComplete = true;
            currentState = DEL_FSM_CHECK_MASTER;
            break;
        default:
            break;
    }
    return isComplete;
}
```

COMANDO ELIMINA NUMERO

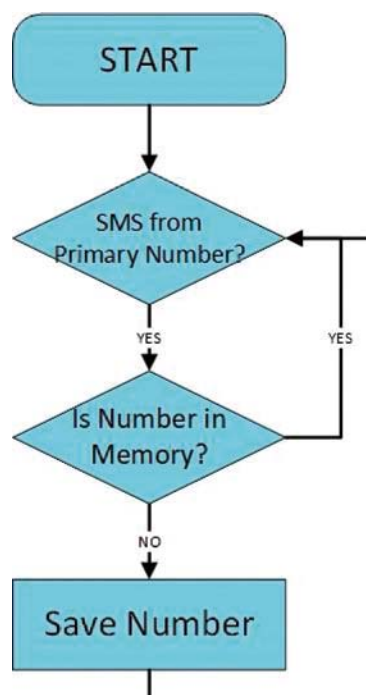
Il comando per eliminare un numero aggiunto in memoria è molto simile al precedente, come evidenziato dal diagramma di flusso proposto nella **Fig. 7** e dall'implementazione software descritta dal **Listato 3**. Se l'SMS è ricevuto dal numero primario, ed il numero che si vuole eliminare è presente in memoria, si può procedere con la cancellazione.

COMANDO DI RESET

Il diagramma di flusso che effettua il reset del dispositivo in **Fig. 8** (come l'implementazione software descritta nel **Listato 4**) è il pezzo più

▼ Listato 4

```
bool resCmdFsm(uint8_t* receivedNumber, bool*
isCmdSuccessfull)
{
    static resFsmStates currentState = RES_FSM_CHECK_
MASTER;
    bool isComplete = false;
    uint8_t masterOpResult = OP_FAILED;
    switch (currentState)
    {
        case RES_FSM_CHECK_MASTER:
            *isCmdSuccessfull = false;
            masterOpResult = isMasterNumber(receivedNumber);
            if (masterOpResult == OP_SUCCESS)
            {
                currentState = RES_FSM_RESET;
            }
            else if (masterOpResult == OP_FAILED)
            {
                currentState = RES_FSM_COMPLETE;
            }
            break;
        case RES_FSM_RESET:
            if (Eeprom_Reset() == 0)
            {
                currentState = RES_FSM_COMPLETE;
                *isCmdSuccessfull = true;
            }
            break;
        case RES_FSM_COMPLETE:
            isComplete = true;
            currentState = RES_FSM_CHECK_MASTER;
            break;
        default:
            break;
    }
    return isComplete;
}
```



◀ **Fig. 6**
Flow-chart della
funzione di
memorizzazione
numeri.

semplice del programma: non fa altro che controllare se l'SMS è stato ricevuto dal numero di telefono primario, e in caso di risposta affermativa cancella tutti i dati presenti nella EEPROM (compreso il preambolo e lo stesso numero primario).

FUNZIONI COMUNI E MEMORIA

Come già accennato le funzioni comuni a più comandi si trovano nel file *utils.c*. La maggior

parte di queste procedure sono necessarie per accedere alla memoria interna del dispositivo. Il Mercury Framework è stato opportunamente esteso per interagire con la EEPROM presente nella board BB110 e le funzioni per salvare o leggere un numero non fanno altro che accedere ad un dato indirizzo di memoria e validare il numero salvato o da salvare.

Fig. 7
Flow-chart della
cancellazione
numeri.

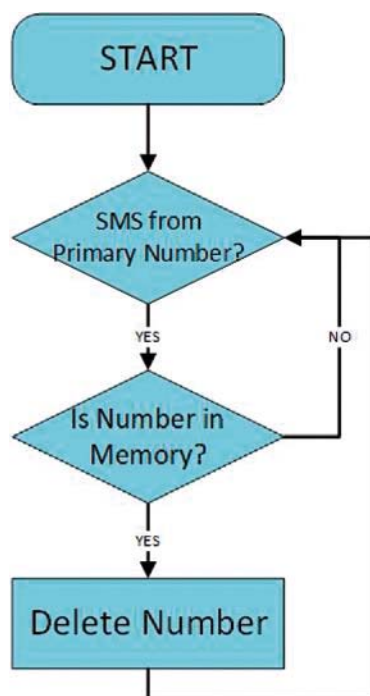
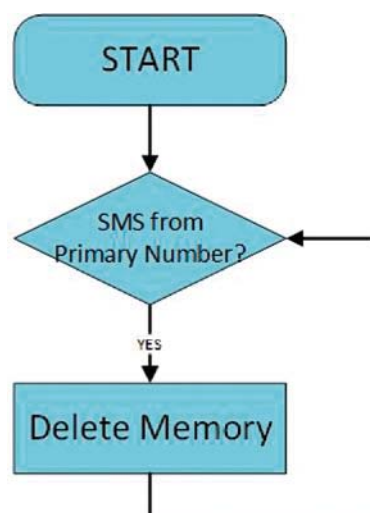


Fig. 8
Flow-chart della
funzione reset di
memoria.



Listato 5

```

bool triggerCmdFsm(uint8_t* receivedNumber)
{
    static triggerFsmStates currentState = TRIGGER_FSM_CHECK_MASTER;
    bool isComplete = false;
    uint8_t numberInMemory = 0;
    uint8_t masterOpResult = OP_FAILED;
    uint8_t searchNumber = OP_FAILED;
    switch (currentState)
    {
        case TRIGGER_FSM_CHECK_MASTER:
            masterOpResult = isMasterNumber(receivedNumber);
            if (masterOpResult == OP_SUCCESS)
            {
                currentState = TRIGGER_FSM_ACTIVATE_RELAY;
            }
            else if (masterOpResult == OP_FAILED)
            {
                currentState = TRIGGER_FSM_CHECK_NUMBER;
            }
            break;
        case TRIGGER_FSM_CHECK_NUMBER:
            searchNumber = isNumberInMemory(receivedNumber, &numberInMemory);
            if (searchNumber == SEARCH_FSM_FOUND)
            {
                currentState = TRIGGER_FSM_ACTIVATE_RELAY;
            }
            else if (searchNumber == SEARCH_FSM_NOT_FOUND || searchNumber == SEARCH_FSM_ERROR)
            {
                currentState = TRIGGER_FSM_COMPLETE;
            }
            break;
        case TRIGGER_FSM_ACTIVATE_RELAY:
            triggerRelay(true);
            secondCounter = 0;
            currentState = TRIGGER_FSM_DEACTIVATE_RELAY;
            break;
        case TRIGGER_FSM_DEACTIVATE_RELAY:
            if (secondsAppTimer(1, &secondCounter, false))
            {
                triggerRelay(false);
                currentState = TRIGGER_FSM_COMPLETE;
            }
            break;
        case TRIGGER_FSM_COMPLETE:
            isComplete = true;
            currentState = TRIGGER_FSM_CHECK_MASTER;
            break;
        default:
            break;
    }
    return isComplete;
}
  
```

Per capire come sono organizzati i dati in EEPROM si può osservare la **Fig. 9**. Come vedete, il preambolo si trova all'indirizzo 0, il primo numero salvato è quello primario, e successivamente, ogni 16 byte, si trovano tutti gli altri numeri. Salvare un numero ogni 16 byte è necessario poiché un numero di telefono contiene solitamente 13 caratteri (ad esempio +390123456789) e non viene fatta alcuna conversione durante il salvataggio: ogni cifra corrisponde quindi ad un carattere, e quindi ad un byte. La cancellazione di un numero consiste semplicemente nel sovrascrivere il numero con una serie di zeri. Per i dettagli implementativi delle singole funzioni è possibile consultare il file *utils.c*. L'implementazione attuale consente la memorizzazione di un massimo di 10 numeri secondari, ma questo limite è facilmente superabile.

COMANDO ATTIVAZIONE RELÉ

A questo punto possiamo procedere con l'implementazione del diagramma di flusso (**Fig. 10**) che rappresenta l'attivazione e la disattivazione del relé con cui verrà comandata la centralina dell'apricancello. Il relé può essere attivato se si riceve una chiamata da un numero salvato, che sia primario o secondario, altrimenti nessuna operazione viene effettuata. Il relé per alcune operazioni come un apricancello deve rimanere attivo per un tempo limitato (specificato nel codice in secondi tramite il primo parametro passato alla funzione *secondsAppTimer*). È anche possibile adattare il codice nel caso si utilizzi il dispositivo per altri fini, quali ad esempio l'accensione di una lampadina, dove è necessario che il relé rimanga attivo per periodi di tempo più lunghi o finché non si riceve un'altra chiamata. L'implementazione software del diagramma di flusso appena descritto si può vedere nel **Listato 5**.

IL TASK APPLICATIVO

L'ultimo passo consiste nel mettere assieme tutte le macchine a stati precedentemente create implementando un'ulteriore funzione che verrà chiamata ciclicamente dal nostro microcontrollore. La macchina a stati principale chiamerà le funzioni di inizializzazione all'avvio e rimarrà quindi in attesa di un comando. Ogni volta che un nuovo comando viene ricevuto, si procederà con la chiamata alla relativa macchina a stati. La procedura appena descritta è rappresentata nel diagramma di flusso proposto nella **Fig. 11**, mentre l'implementazione a livello di software è

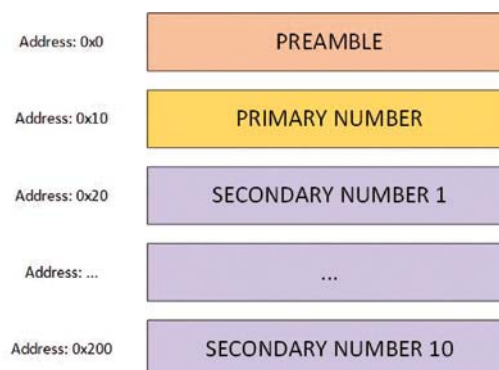


Fig. 9
Organizzazione
della EEPROM.

riportata nel **Listato 6**, dove vedete il programma principale del telecomando apricancello. Si noti infine nel file *app_main.c* come la funzione

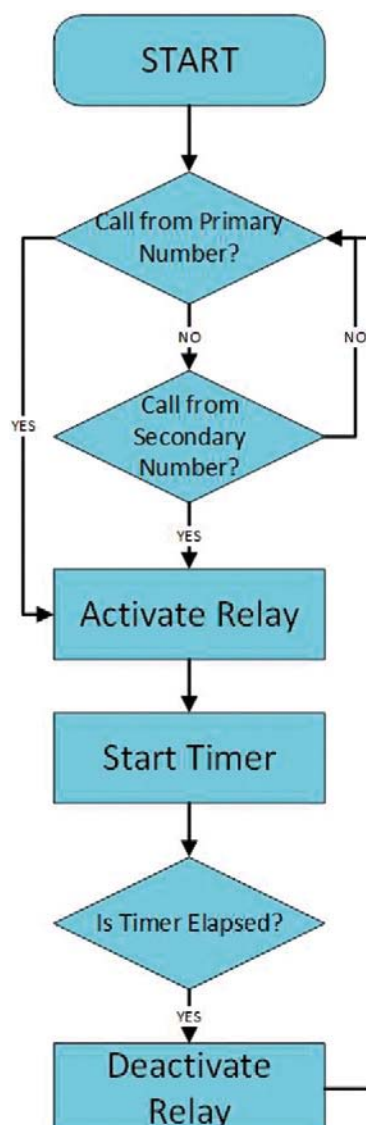
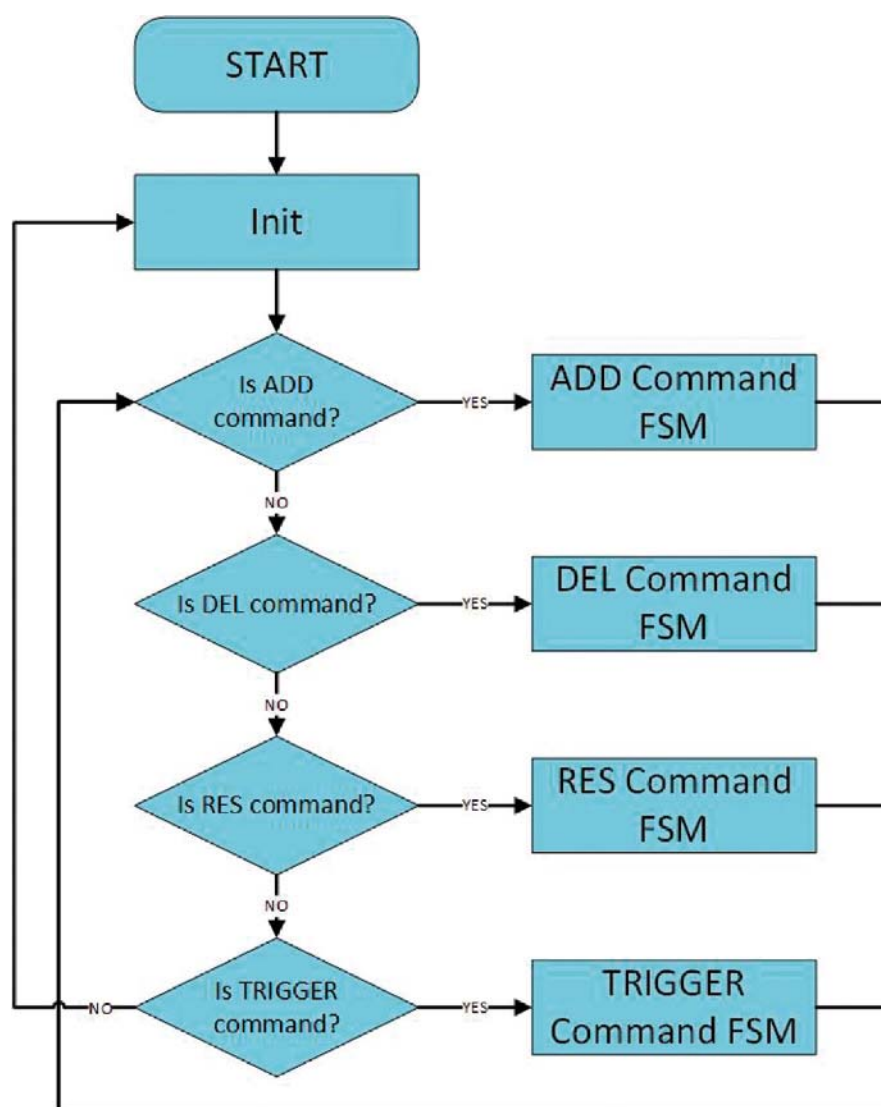


Fig. 10
Flow-chart
della gestione
del relé.

Fig. 11
Flow-chart del task applicativo.



detectCmd faccia in modo di leggere mittente e testo ogni volta che un nuovo SMS è ricevuto, eliminando il messaggio stesso per evitare che la memoria della SIM si riempia. La stessa funzione chiama l'API del Mercury System per riagganciare una chiamata ricevuta dopo aver letto i dati relativi al numero del chiamante. Sempre nello stesso file, è stata implementata la funzione *parseCommand* in grado di identificare un comando all'interno del testo di un SMS, ignorando i comandi sconosciuti. I testi scelti per i comandi sono i seguenti:

- **ADD;**<Numero> -> per aggiungere un nuovo numero;
- **DEL;**<Numero> -> per eliminare un numero;
- **RES;** -> per effettuare il reset;

L'utente è ovviamente libero di modificare il codice in caso volesse usare altri testi o per aggiungere ulteriori comandi.

Per maggiori dettagli è possibile dare uno sguardo al codice disponibile al repository Github raggiungibile all'indirizzo: <https://github.com/andreamisuraca/mercury-gsm-app>.

L'APPLICAZIONE PER PC

Nel nostro sistema apricancello è possibile impostare il numero primario utilizzando l'apposito software applicativo (disponibile nel medesimo repository succitato e da esso scaricabile) da installare su Personal Computer. Allo scopo, dopo aver collegato il sistema Mercury al

▼ Listato 6

```
void MyApp_Task (UINT8 Options)
{
    static appMainFsmStates currentState = APP_MAIN_INIT;
    static bool isCmdSuccessfull = true;
    cmdVisualEffet(isCmdSuccessfull);
    switch (SystemState)
    {
        /* System Initialization Phase */
        case InitializationState:
        /* Make app init. if necessary */
        Led_SetLedBlinkTime(LED_1,100,900);
        Led_SetLedBlinkTime(LED_2,100,900);
        break;
        /* System Normal operation Phase */
        case RunningState:
        switch (currentState)
        {
            case APP_MAIN_INIT:
            if (initFsm(&isCmdSuccessfull))
            {
                currentState = APP_MAIN_WAIT_EVENT;
            }
            break;
            case APP_MAIN_WAIT_EVENT:
            if (detectUsbNumber(&isCmdSuccessfull))
            {
                triggerVisualEffect();
            }
            currentState = detectCmd();
            break;
            case APP_MAIN_ADD_CMD:
            if (addCmdFsm(receivedNumber, smsText, &isCmdSuccessfull))
            {
                currentState = APP_MAIN_WAIT_EVENT;
                triggerVisualEffect();
            }
            break;
            case APP_MAIN_DEL_CMD:
            if (delCmdFsm(receivedNumber, smsText, &isCmdSuccessfull))
            {
                currentState = APP_MAIN_WAIT_EVENT;
                triggerVisualEffect();
            }
            break;
            case APP_MAIN_TRIGGER_CMD:
            if (triggerCmdFsm(receivedNumber))
            {
                currentState = APP_MAIN_WAIT_EVENT;
            }
            break;
            case APP_MAIN_RESET:
            if (resCmdFsm(receivedNumber, &isCmdSuccessfull))
            {
                currentState = APP_MAIN_INIT;
                triggerVisualEffect();
            }
            break;
            default:
            break;
        }
        break;
        /* Default */
        default:
        break;
    }
}
```

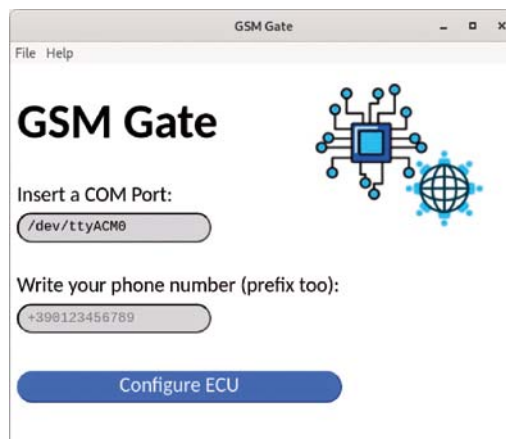


Fig. 12
Interfaccia
utente
dell'applicazione
per PC.

PC tramite USB è sufficiente inserire il numero di telefono nella casella della schermata proposta in **Fig. 12** (Write your phone number) e cliccare sul pulsante *Configure ECU*. Questa è una delle funzioni supportate dal programma per PC, ma è quella che attualmente ci interessa vedere.

CONCLUSIONI

Nonostante la facilità di sviluppo, il Mercury System consente di creare un'applicazione funzionale e sicura, implementando quelli che potrebbero essere i requisiti di un qualsiasi apricancello commerciale. Il design modulare dell'hardware, che ci consente di aggiungere alla Base Board estensioni come il modem GSM o il relé, e del software, grazie al Mercury Framework, consente di focalizzarci sul nostro applicativo senza badare all'implementazione dei driver di basso livello.

Ci si può quindi sbizzarrire creando i dispositivi IoT più disparati. ■



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili nello store di Futura Elettronica. Il modulo base (cod. BB110), è in vendita a Euro 23,00, il modulo GSM con SIM800C (cod. FT1308) è disponibile a Euro 29,00, il modulo relé (cod. SB111) costa Euro 17,00, il modulo alimentatore (cod. PB110) è in vendita a Euro 17,00, il modulo espansione per quattro moduli (cod. EB111) è disponibile a Euro 12,00. 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



**SONO ARRIVATI I NUOVI
SET DI COMPONENTI
CON ARDUINO UNO!**

LIVELLO 2



€ 42,90

Cod. **ARDUKITLEVEL2**

**SET con Arduino Uno Rev 3
e i componenti elettronici
per sviluppare applicazioni
con il linguaggio Arduino**

LIVELLO 1



€ 32,90

Cod. **ARDUKITLEVEL1**

Disponibili anche....

SET PER ARDUINO UNO REV3

€ 65,00

Cod. **ARDUKITV6**

**IL PIÙ
COMPLETO!**



Il Set comprende: il libro "l'ABC di ARDUINO", la board Arduino UNO Rev3, e tutti i componenti per effettuare esperimenti descritti nel libro.

**SET CON ARDUINO UNO REV 3
CON CAVO USB**

Contiene il necessario per utilizzare Arduino Uno Rev 3; per iniziare basta connettersi a un computer tramite il cavo USB o alimentarla con un trasformatore AC/DC o una batteria.



€ 25,50

Cod. **COMB06G**

Prezzi IVA inclusa

AMPLIFICATORI AUDIO

di **GABRIELE PERETTO**

Analizziamo in laboratorio le caratteristiche tecniche di moduli amplificatori commerciali fino a 2x50W. Terza e ultima puntata.

Ci avviaamo alla conclusione del nostro articolo-tutorial dedicato allo studio teorico e all'analisi pratica degli amplificatori di potenza. Dopo aver affrontato nella prima puntata la tecnica e teoria, trattando i parametri, le classi di funzionamento e le tipologie di misurazione che vengono normalmente effettuate per la caratterizzazione degli amplificatori audio e descritto (e provato) nella seconda alcuni moduli di bassa potenza, fino a 3W max., in questo articolo conclusivo andremo a verificare moduli con potenze più importanti, che possono arrivare fino a 50/60 watt. Anche in questo caso, parliamo di "moduli commerciali", che sono reperibili on-line presso www.futuranet.it. Ugualmente, per ogni modulo testato, abbiamo raccolto i dati più significativi all'interno di una tabella.

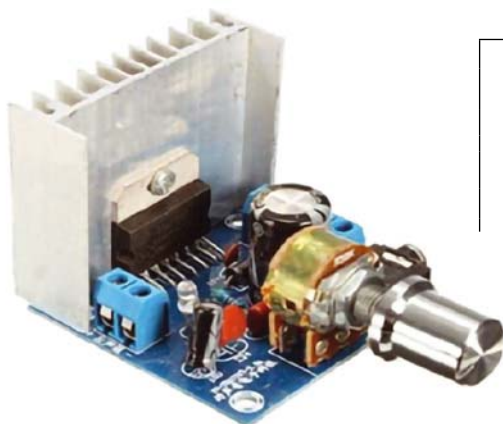
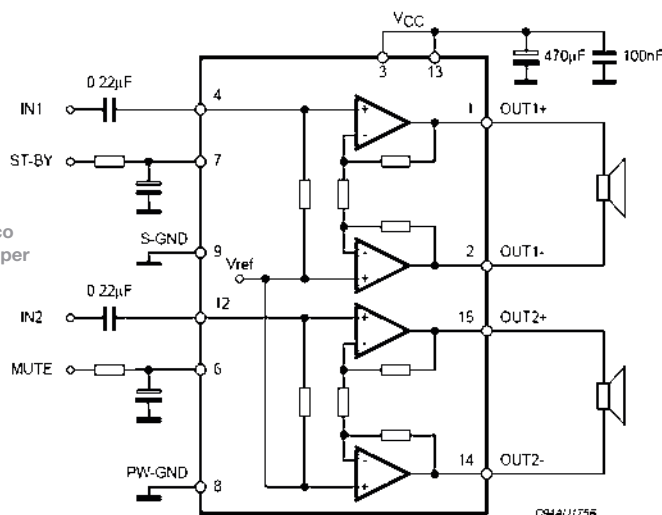


Fig. 1
Modulo
amplificato-
re STEREO
2X15 W con
TDA7297
AMPLI2X15W

Fig. 2
Schema elettrico
base di utilizzo per
TDA7297.



Per comprendere il significato dei parametri contenuti nelle tabelle ed evitare di ripeterlo ogni volta per ciascuna, qui di seguito riportiamo la convenzione utilizzata:

- **V_ALIM** = Tensione di alimentazione
- **P (W)** = Potenza assorbita in watt
- **Vpp In** = Tensione di ingresso in volt pk/pk
- **Vout_rms** = Tensione di uscita sotto carico
- **PO_8Ω (W)** = Potenza di uscita in watt rms
- **AV (dB)** = Guadagno in tensione espresso in dB
- **THD %** = Distorsione armonica totale
- **THD % 3a** = Distorsione della terza armonica
- **rend. %** = Rendimento percentuale.

Ribadiamo qui il principio che, siccome normalmente l'udibilità umana non percepisce distorsioni superiori al 5% purché la terza armonica non superi il 2%, abbiamo, anche in questo contesto, cercato di rilevare parametri nell'intorno di questi valori, eccedendo in taluni casi fino al 10% (parametro spesso utilizzato dai produttori per indicare la distorsione alla massima potenza), soltanto per illustrare come si presentano forma d'onda e spettro armonico in presenza di elevata distorsione e comprendere se l'architettura dei progetti impostati dai progettisti delle "board" e quindi realizzati, rispettano massimamente le condizioni dichiarate dai costruttori dei "chip integrati" che rappresentano il cuore del progetto. Tutte le prove sono state eseguite (come da manuale) su carichi composti da resistori di potenza e quindi hanno valenza puramente tecnica. La valutazione della dinamica, come abbiamo descritto nei capitoli precedenti, dovrebbe essere correttamente fatta con un carico induttivo quale è l'altoparlante per la riproduzione, anzi meglio nella sua apposita cassa acustica correttamente progettata e accordata e all'interno del locale di diffusione sonora.

Per questo motivo non troveremo qui i dati relativi al fattore di smorzamento (damping factor) ovvero: il rapporto dell'impedenza del carico, diviso l'impedenza dell'amplificatore, proprio perché questa misura va fatta tra amplificatore e altoparlante di uscita, il quale dispone di caratteristiche proprie peculiari di impedenza e di dinamica in funzione della membrana, del magnete, della dimensione, della cassa acustica ecc. I parametri che andremo a riportare, dovrebbero comunque essere sufficienti per indirizzare i possibili utilizzatori ad operare una scelta mirata alle proprie esigenze.

STRUMENTAZIONE

Come fatto per le precedenti prove, anche in questo percorso di test abbiamo utilizzato strumentazioni di vario genere, che abbiamo ampiamente illustrato nella prima e nella seconda puntata. Inoltre in questo caso abbiamo eseguito varie misure comparative mediante l'utilizzo di un PICOSCOPE serie 2000 (www.picotech.com) che pur essendo un prodotto posizionato su una fascia di prezzo contenuta, dispone di funzioni matematiche sui due canali di ingresso, evitando così l'utilizzo di costose sonde differenziali e dispone anche di un ottimo analizzatore di spettro, in grado di fornire un'ampia gamma di informazioni sulle varie tipologie di distorsione, banda passante ecc. Inoltre il SW di gestione dispone di filtro passa basso di ingresso, impostabile alla frequenza desiderata (normalmente per misure sugli amplificatori Hi-Fi, si imposta sui 30kHz) che permette di eliminare la componente HF di modulazione senza dover impiegare filtri RC aggiuntivi, come illustrato nella prima puntata. Infine, esso dispone pure di un generatore integrato di segnali compresa la funzione "sweep",

che possono essere prelevati tramite l'apposita uscita BNC ed applicate all'ingresso dell'amplificatore da testare. Quindi una serie di vantaggi non indifferenti per chi vuole effettuare delle analisi su apparecchiature BF senza dover acquistare svariati e costosi strumenti. Basta questo piccolo apparato e un PC dotato di porta USB ed abbiamo tutto quanto ci serve.

MODULO CON TDA7297

Questo primo modulo (Fig. 1) che abbiamo verificato (cod. AMPLI2X15W), utilizza un integrato monolitico TDA7297 in package Multiwatt15, prodotto da STMicroelectronics e classificato come 15+15W DUAL BRIDGE AMPLIFIER. Questo integrato in configurazione stereo che nasce nei primi anni 2000 e risulta progettato e realizzato in architettura di classe AB e che troviamo tutt'ora in produzione, è stato largamente utilizzato prima dell'avvento della nuova generazione di amplificatori, in svariate applicazioni soprattutto portatili. E proprio per la sua versatilità, la sua notorietà e la sua collaudata funzionalità viene tutt'ora ancora ampiamente utilizzato. In (Fig. 2) riportiamo lo schema elettrico tipico, tratto dal datasheet. Nella Tabella 1, abbiamo riportato, come nelle precedenti recensioni, i principali parametri dichiarati dal fornitore (ST). Poiché questo modulo dispone anche di potenziometro per il controllo di volume, abbiamo regolato lo stesso in modo da ottenere una sensibilità di ingresso sempre uguale e da noi fissata a 500 mVpp. Comunque l'ingresso può accettare segnali ben oltre 1 Vpp, purché il potenziometro non sia regolato per la sua minima sensibilità circa 100 mV. I dati rilevati sono riportati in Tabella 2 e come sempre sono riferiti ad un solo canale.

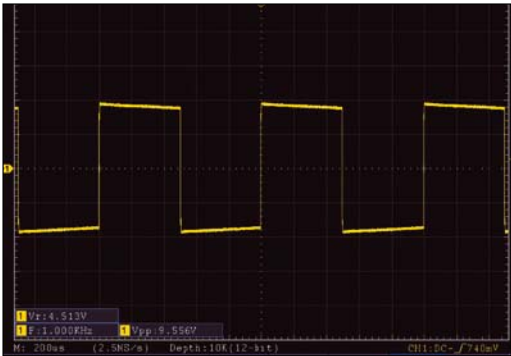


Fig. 3
Forma d'onda
quadra tipica
a 1KHz del
modulo
AMPLI2X15W.

Risultando l'altro, praticamente uguale, per conoscere la potenza totale assorbita bisogna ovviamente, moltiplicare per 2 il valore P(W). Abbiamo osservato che i dati dichiarati dal costruttore (ST) risultano abbastanza allineati relativamente alle distorsioni e al guadagno e noi, coerenti a quanto ci siamo prefissati circa i livelli massimi di distorsione accettabili, abbiamo riscontrato che per un utilizzo ottimale, non bisogna comunque prelevare in uscita più di 5 – 6 watt per canale, che tutto sommato per una vasta gamma di applicazioni risulta più che sufficiente. La banda passante risulta abbastanza lineare su tutta la gamma che si estende ben oltre i 20 kHz, con una leggera progressiva attenuazione dei bassi sotto i 250 Hz. In (Fig. 3) è rappresentata la forma d'onda quadra tipica a 1000 Hz che rende abbastanza bene l'idea dell'andamento anzi descritto circa l'attenuazione delle frequenze più basse. Il rendimento di questo modulo si posiziona attorno al 50% e comunque nei parametri di funzionamento classico degli amplificatori di classe AB. Questo prodotto, benché relativamente vetusto e

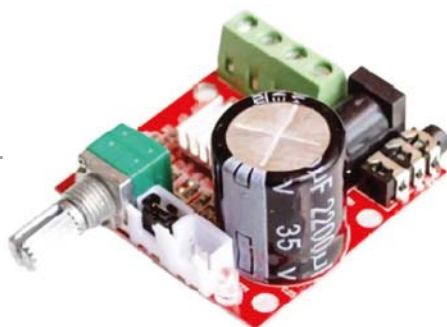
Tabella 1
Dati del costruttore relativi al C.I. TDA7297.

Parametri	Condizioni di test / note	Min	tip	max	Unità di misura
Vs tensione di alimentazione		6.5		18	V
Impedenza altoparlante			8		Ω
Guadagno Gv /Av		31	32	33	dB
Potenza di uscita	16.5V – 1KHz- THD 10%	13+13		15+15	W
Distorsione THD	16.5V – 1KHz - 1W – 8Ω	0.1		0.3	%

Tabella 2
Parametri rilevati sul modulo AMPLI2X15W.

V_ALIM	P (W)	Vpp In min (*)	Vout_rms	PO_8Ω (W)	AV (dB)	THD %	THD % 3a	rend. %
12	11,2	0,5	6,5	5,281	31,3	1,66	0,1	47
12	12,5	0,5	7,1	6,301	32,1	3,8	3,67	50

→ Fig. 4
Modulo PAM8610.

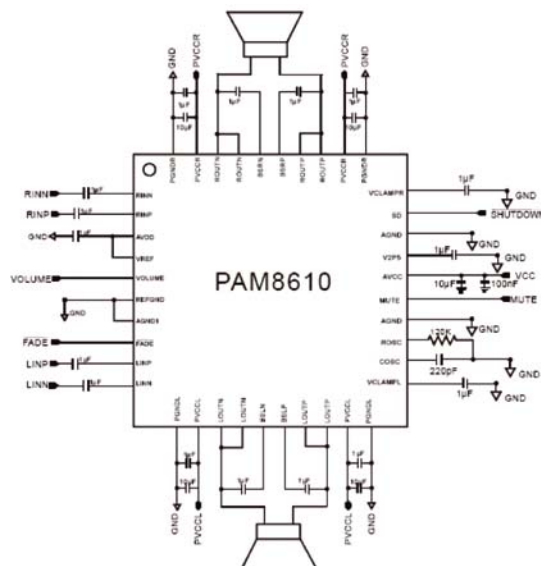


affiancato da moduli di nuova generazione, risulta comunque ancora valido per qualità / prezzo / dimensioni.

MODULO CON PAM8610

Il secondo amplificatore integrato di cui ci occuperemo in questa trattazione (Fig. 4), è un piccolo modulo di dimensioni 40x40x23 mm, dotato anch'esso di potenziometro per la regolazione del volume di ingresso, alimentazione tramite plug da 5,5 mm e ingressi audio tramite jack stereo. Inoltre presenta due connettori: il primo utile a collegare un interruttore di alimentazione, senza doverla scollegare fisicamente e l'altro per la funzione di "muting". Si basa su un chip monolitico che lavora in classe "D" tipo PAM8610 prodotto da Diodes Inc. In Fig. 5 è riportato lo schema tipico applicativo, suggerito dal costruttore, mentre in Tabella 3, sono riportati i dati principali dichiarati sempre dal costruttore e relativi all'integrato stesso. Abbiamo notato che se andiamo a scaricare il datasheet, questo "chip" viene considerato obsoleto per nuovi progetti e si consiglia l'utilizzo del PAM8406. Questo significa che pur essendo ancora in produzione, nei prossimi anni potrebbe progressivamente essere abbandonato.

In Tabella 4, troviamo quindi i valori da noi rilevati. Specifichiamo che abbiamo provato a testarlo a 4Ω, ma la piccola aletta dissipatrice di cui è dotato il circuito, non è sufficiente a garantire la dissipazione



→ Fig. 5
Schema elettrico
base di impiego del
C.I. PAM8610.

termica tale da superare 6-7 watt e il circuito va continuamente in protezione termica. Pertanto, ci siamo concentrati sui rilievi tipici a 8Ω di impedenza. Possiamo senz'altro affermare che i valori dichiarati, in questo progetto vengono rispettati. Si può notare che fino a 7 watt, la distorsione è molto contenuta, per poi salire bruscamente fino ad arrivare a circa 10% a 9-10 watt, come correttamente dichiarato dal costruttore. La banda passante (Fig. 6), come quasi tutti gli amplificatori in classe "D", risulta molto lineare e va ben oltre i 100 kHz, che a noi non servono, ma potrebbe essere interessante per l'utilizzo di amplificatori di potenza ad ultrasuoni. Sulle frequenze basse, sotto i 300 Hz troviamo una leggera attenuazione, ma sempre molto contenuta di massimo 2dB a 20 Hz. In questo caso abbiamo rendimenti superiori al 75% tipici di questa classe di amplificatori.

→ Tabella 3
Dati del costruttore relativi al C.I. PAM8610.

Parametri	Condizioni di test / note	Min	tip	max	Unità di misura
Vs tensione di alimentazione		7	12	15	V
Impedenza altoparlante		4		8	Ω
Guadagno Gv /Av	Con regolazione volume	-75		+32	dB
Potenza di uscita	1KHz- 8 Ω THD+N = 1%	8	+	8	W
Potenza di uscita	1KHz- 8 Ω THD+N = 10%	10	+	10	W
P. U. (Dissipatore necessario)	1KHz- 4 Ω THD+N = 0,1%	15	+	15	W
Distorsione THD	12V - 1KHz - 1W - 8Ω	0.1			%

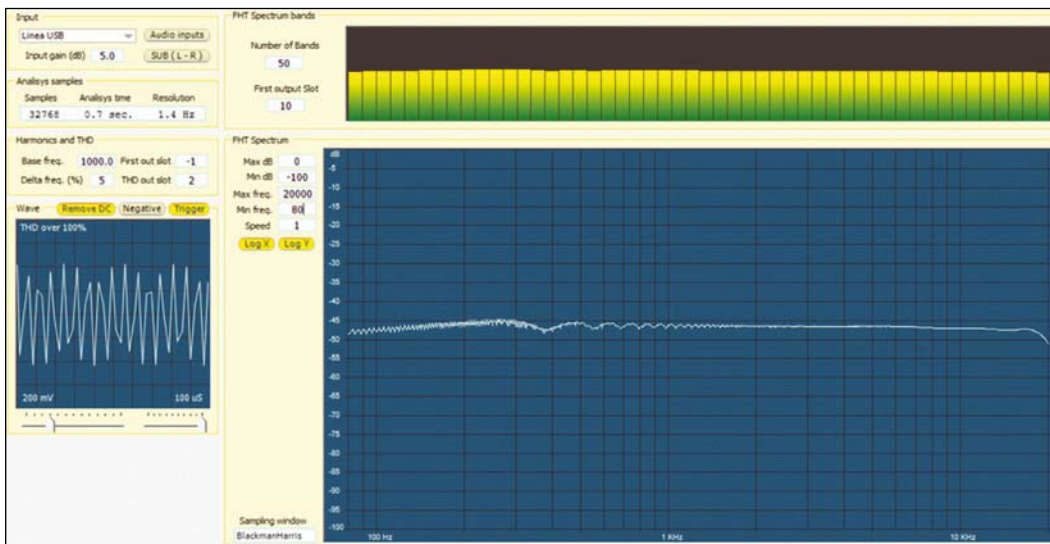


Fig. 6
Curva di
risposta
della banda
passante
del modulo:
PAM8610.

Tabella 4
Parametri rilevati sul modulo 1606-PAM8610.

V_ALIM	P (W)	Vpp In	Vout_rms	Po_ 8Ω (W)	Av (dB)	THD %	THD % 3a	rend. %
12	19	1,28	7,56	7,1	24,5	0,17	0,07	75
12	22,8	1,66	8,58	9,2	23,3	9,8	9,2	81

In conclusione, riteniamo questo modulo veramente interessante per qualità / prezzo per potenze fino a 7-8 watt per canale. Possiamo quindi definire questo amplificatore, come "un buon competitor" del precedente, soprattutto se desideriamo una maggior resa sulle frequenze basse e un miglior rendimento e quindi una minore dispersione termica, ricordando però che con questa piccola aletta di raffreddamento, risulta valido solo per impedenze di 8 ohm.

MODULO CON TDA7492

Stiamo ora salendo di potenza e andremo ad analizzare un modulo di dimensioni leggermente maggiori (71x59x21), dotato di dissipatore di temperatura ben dimensionato e basato su TDA7492 di STM in grado di erogare, secondo datasheet, fino a 50+50W con alimentazione a 25V.

Tabella 5
Dati del costruttore relativi al C.I. TDA7492.

Parametri	Condizioni di test / note	Min	tip	max	Unità di misura
Vs tensione di alimentazione		8	25	36	V
Impedenza altoparlante		6		8	Ω
Guadagno Gv /Av	Con regolazione volume	20		34.6	dB
Potenza di uscita	25V - 1KHz- 6 Ω THD= 1%		40 + 40		W
Potenza di uscita	25V - 1KHz- 6 Ω THD= 10%		50 + 50		W
Distorsione THD	25V - 1KHz - 1W – 8Ω		0.1	0.4	%
Efficienza	Po = 40W + 40W	80	80		%

In **Fig. 7** è visibile l'architettura della scheda dove si vedono i componenti relativi ai due canali stereo speculari, con i due morsetti per il collegamento degli altoparlanti e il morsetto centrale per il collegamento alla fonte di alimentazione. Sul lato opposto troviamo un connettore a vaschetta a 3 poli a passo 2.54 mm per collegare i due segnali di ingresso con massa (GND) comune. Anche in questo caso siamo di fronte ad un amplificatore in classe "D" dual BTL (Bridge-tied load) ovvero in configurazione a ponte senza trasformatore con uscite differenziali singole per ogni canale (ovvero NON a massa comune).

La **Fig. 8** rappresenta lo schema interno del circuito, relativo ad un solo canale, dove si può notare il classico schema a doppio ponte adottato, mentre nella **Tabella 5**, sono riportati, come sempre i dati salienti del produttore. Purtroppo, la maggior parte

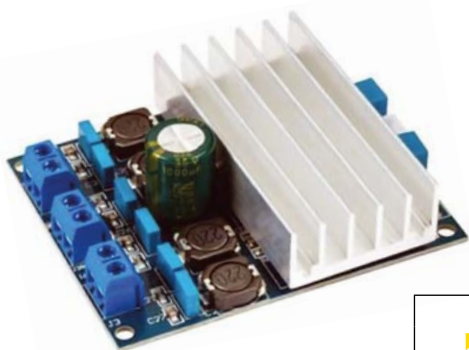


Fig. 7
Modulo
amplificatore
stereo 2X50W
con TDA7492
AMPLI2X50W.

Fig. 8
Schema
tipico
applicativo
relativo
al chip
TDA7492.

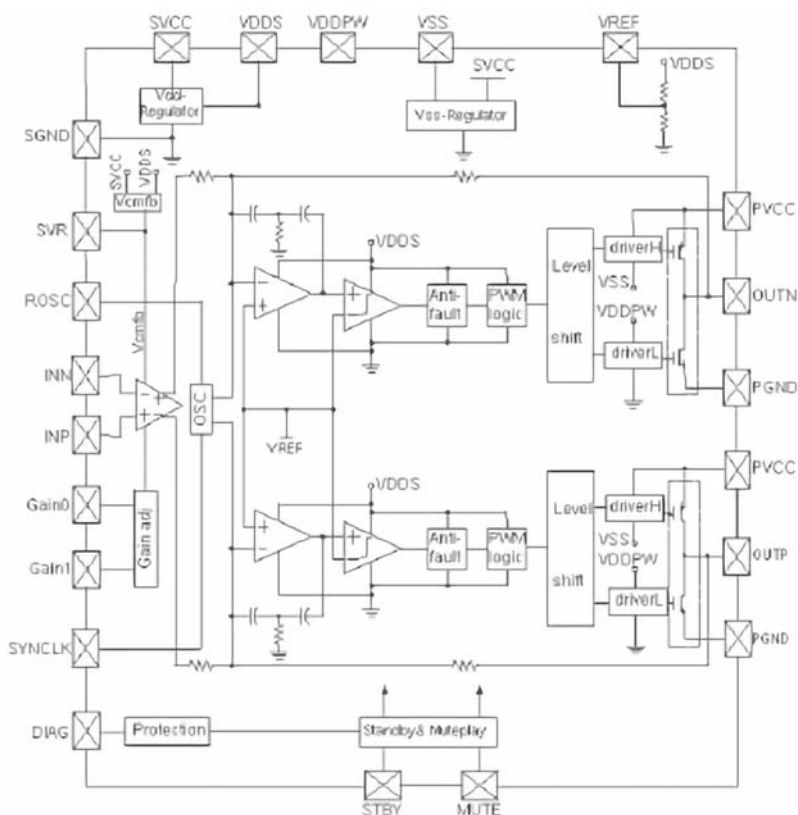
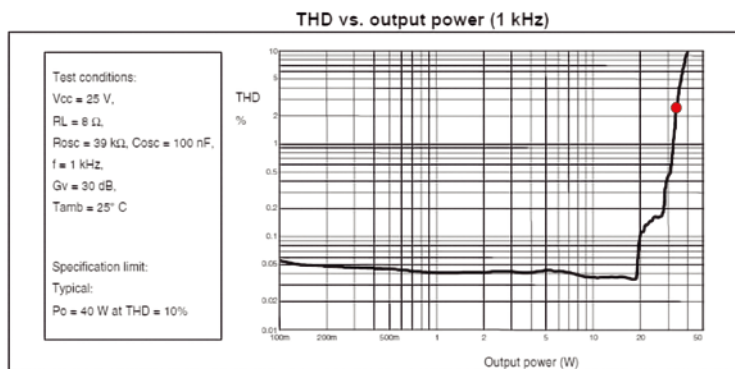


Fig. 9
Curva del
rapporto potenza
/distorsione
relativa al TDA7492
(data-sheet
costruttore).



dei parametri riportati dal costruttore si riferisce ad una impedenza altoparlante di 6Ω, che risulta poco utilizzata in campo audio e nemmeno tanto facile da ottenere.

Per arrivare a questa configurazione bisogna utilizzare 3 altoparlanti da 4 ohm, di cui due collegati in parallelo per ottenere 2 ohm e l'altro in serie per ottenere 2+4 = 6 ohm. Per cui noi abbiamo preferito eseguire dei test a soli 8Ω, con tensione di alimentazione a 24Vcc, ottenendo ovviamente potenze erogate leggermente inferiori. Abbiamo poi voluto vedere se anche a 4 ohm (non contemplati dal costruttore), poteva funzionare altrettanto

bene e in sicurezza, scoprendo che si può utilizzare tranquillamente, purché la tensione di alimentazione sia di soli 12Vcc.

Nella **Tabella 6**, abbiamo riportato i dati rilevati alle differenti condizioni e poiché abbiamo riscontrato i parametri, molto coerenti con quelli dichiarati, in **Fig. 9** abbiamo scelto di riportare il grafico che rappresenta il rapporto potenza / distorsione fornito dal costruttore, dove si può notare come la distorsione a 33 W su 8 Ω, corrisponda esattamente a quanto da noi rilevato, ovvero THD 2,24%. Bisogna tuttavia notare che la distorsione di 3a armonica, si avvicina molto al valore di distorsione totale.

In **Fig. 10**, dove è illustrato lo spettro di frequenza relativo a tutte le armoniche superiori e multiple dei 1kHz si possono infatti notare, oltre all'alto livello di questa armonica ovvero la terza, anche elevati livelli di armoniche dispari superiori (5a, 7a, 9a ecc.). La banda passante risulta perfettamente lineare tra 150Hz e 5kHz circa, per subire poi da entrambi i lati una progressiva ma molto contenuta attenuazione fino a circa -2 dB.

Possiamo quindi utilizzare questo amplificatore con un ottimo rendimento acustico su tutta la banda audio e con linearità assoluta tra 150Hz e 5 kHz e per potenze fino a 30 watt rms, su 8 Ω con alimentazione a 24 Vcc.

Utilizzando una alimentazione di soli 12 Vcc, non potremo pretendere più di 12 – 13 watt rms, per ogni canale, che tuttavia è un ottimo risultato e che fino a qualche anno fa era un traguardo irraggiungibile se non con complessi artifici. Questo amplificatore, come ci aspettavamo, raggiunge un'efficienza molto elevata che si attesta intorno all'85% minimo con alimentazione a 12Vcc e 11 W, fin oltre al 90% con alimentazione a 24Vcc a 30W di uscita di potenza.

A nostro avviso questo modulo si presta già molto bene a realizzare un discreto impianto di riproduzione con un costo quasi irrisorio di poco più di 20 Euro.

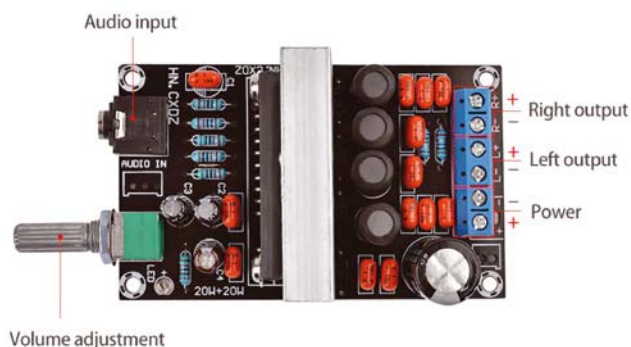
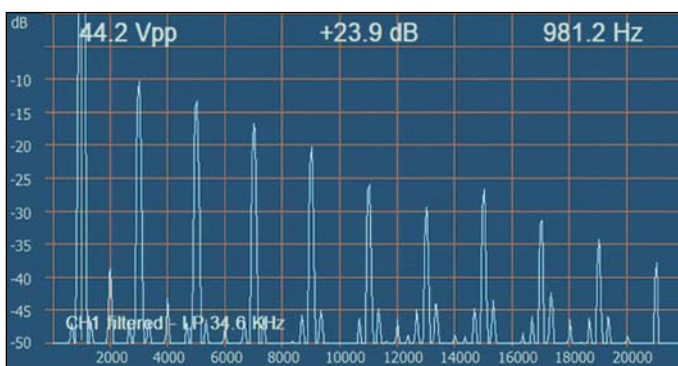


Fig. 10
Spettro di
frequenza delle
armoniche a 33W
su 8Ω del modulo
AMPLI2X50W.

Fig. 11
Modulo
amplificatore stereo
10+10W in classe T
AMPLITA2020.

MODULO CON TA2020-020

Ora ci occuperemo di un amplificatore ad architettura un po' particolare e cioè di classe "T" (Fig. 11) che non risulta molto diffuso ed è anche poco conosciuto. Come avevamo visto nella prima parte, l'amplificatore di classe "T", risulta sostanzialmente simile ad un amplificatore in classe "D", con una copertura di brevetto da parte del costruttore "Tripath" che produce quindi una serie prioritaria dedicata di chip per svariate applicazioni a differenti potenze di erogazione. In questo caso il progettista del modulo ha utilizzato un chip denominato TA2020. Questo, funziona con una modulazione ad alta frequenza variabile tipicamente tra 100 kHz e 1 MHz detta a "spettro esteso" e che varia in funzione del segnale audio applicato. Questa tipologia di modulazione, variabile, a

differenza del classe "D" che ha frequenza fissa, non altera né distorce il segnale di ingresso, ma introduce comunque alcune componenti di frequenza elevata che si posizionano oltre i 100 kHz. Dette componenti sono per lo più filtrate dai filtri passa basso di uscita, ma in misura seppur modesta e molto attenuata, riescono comunque a sovrapporsi al segnale BF, che essendo comunque posizionate oltre la soglia udibile, non influenzano minimamente la qualità del segnale.

La Fig. 15 illustra abbastanza chiaramente la forma d'onda di uscita con la sovrapposizione di piccole porzioni di alta frequenza che riescono a superare i filtri di uscita. Nella Fig. 12, riportiamo la curva tipica dichiarata dal costruttore, del rapporto potenza / distorsione (THD+N), mentre nella Tabella 7, sono sintetizzati i principali valori forniti dallo

Tabella 6
Parametri rilevati sul modulo AMPLI2X50W.

V _{ALI} M	P (W)	V _{pp} In	V _{out_rms}	P _{o_8Ω} (W)	A _v (dB)	THD %	THD % 3a	rend. %
24	24,7	0,8	13,4	22,4	33,5	0,17	0,17	91
24	37	1	16,3	33,2	33,3	2,24	2,02	90
V _{ALI} M	P (W)	V _{pp} In	V _{out_rms}	W _{rms_4Ω}	A _v (dB)	THD %	THD % 3a	rend. %
12	13,8	0,42	6,8	11,6	33,2	0,4	0,38	84
12	16,8	0,5	7,75	15,0	32,8	3,39	3,58	89

Tabella 7
Dati del costruttore relativi al C.I. TA2020-020.

Parametri	Condizioni di test / note	Min	tip	max	Unità di misura
Vs tensione di alimentazione		8,5	13,5	14,6	V
Impedenza altoparlante		4		8	Ω
Potenza di uscita	13.5V - 1KHz- 8Ω THD= 0.1%		7 + 7		W
Potenza di uscita	13.5V - 1KHz- 4Ω THD= 0.1%		11 + 11		W
Distorsione THD+N	13.5V - 1KHz - 10W - 4Ω		0.03		%
Efficienza	Po = 20W 4 Ω - 12W 8 Ω	81		88	%

Tabella 8
Parametri rilevati sul modulo AMPLITA2020.

V_ALIM	P (W)	Vpp In	Vout_rms	Po_ 8Ω (W)	Av (dB)	THD %	THD % 3a	rend. %
12	8	0,6	7,15	6,4	30,6	2,4	0,8	80
12	11	0,8	8,74	9,5	29,8	7.9	6.7	87
V_ALIM	P (W)	Vpp In	Vout_rms	Wrms_ 4Ω	Av (dB)	THD	THD % 3a	rend. %
12	14,2	0,6	6,47	10,5	29,7	4,21	0,78	74
12	17,5	0,6	7,43	13,8	30,9	5	2,3	79

Fig. 12
Curva del rapporto potenza/distorsione relativa al TA2020-020 (data-sheet costruttore).

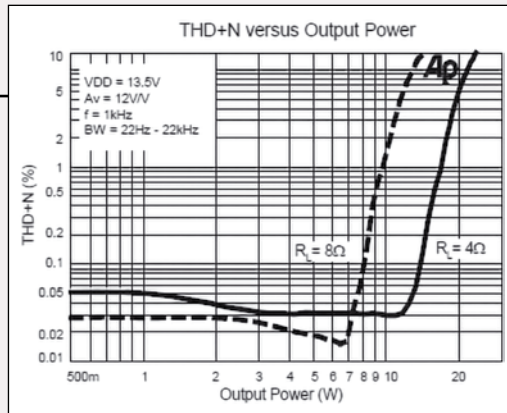


Fig. 13
Forma d'onda fortemente "clippata" a 9.5W- 8Ω e THD 7.9% del modulo AMPLITA2020.

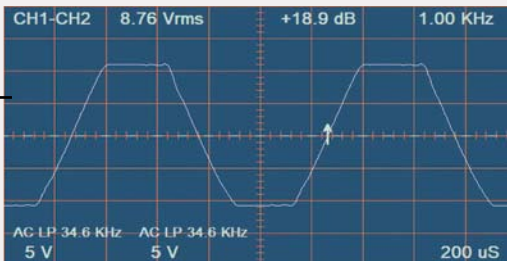
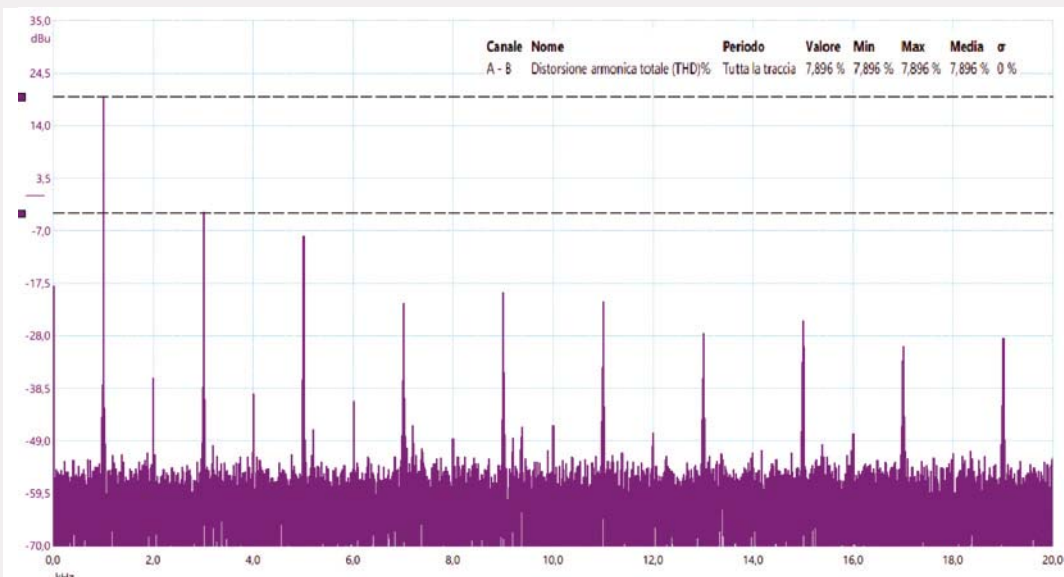


Fig. 14
Spettro della fondamentale e delle armoniche a 9.5W- 8Ω del modulo AMPLITA2020.



stesso e tratti quindi dal datasheet del componente. I nostri rilievi, elencati in **Tabella 8**, in questo caso, non danno purtroppo gli stessi riscontri, ma crediamo sia sostanzialmente un problema di layout progettuale e non imputabile al chip stesso, oppure che sia stato introdotto un errore di trascrizione dei dati nel datasheet, perché i valori risultano di almeno un ordine superiore e nel caso di uscita a 10W con alimentazione a 12 V – 1kHz sono addirittura di due ordini superiori, ovvero: invece di 0,03% abbiamo riscontrato valori intorno al 3%. Il rendimento è comunque buono e si attesta intorno all'80%, come dichiarato, mentre la banda passante risulta lineare tra i 70 Hz fino a 4 kHz, rispettando i parametri dichiarati, anche in questo caso. Comunque al di sotto dei 70 Hz e fino a 10 Hz e sopra i 4kHz e fino a 20 kHz la caduta risulta di soli -0,5 dB e quindi veramente trascurabile. Possiamo quindi tranquillamente utilizzare questo amplificatore considerandolo eccezionalmente lineare su tutta la banda audio. Nelle immagini allegate, possiamo osservare alcune forme d'onda e relative distorsioni, a prova

di quanto affermato. In particolare, in (Fig. 13) notiamo la forte "tosatura" a $8\Omega - 9,5$ watt, che genera una distorsione pari a 7,9% (Fig. 14). In Fig. 15 vediamo quindi, la forma d'onda in uscita per ogni singolo canale con la risultante matematica A-B, relativamente alla condizione 12V di alimentazione, 4Ω e 10,5W out, dove sono evidenziate le componenti HF anzi descritte, che riescono a passare dai filtri. In questo caso abbiamo volutamente utilizzato solo il filtro RC e non quello SW dello strumento, proprio per mettere in risalto il segnale applicato agli altoparlanti. In Fig. 16 troviamo le componenti armoniche alle stesse condizioni. In questo caso, notiamo però che le ampiezze maggiori sono date rispettivamente dalla seconda e dalla quarta armonica, per cui la distorsione non si presenta fastidiosa essendo quasi tipico di una classe "A" ed oltremodo la terza armonica è anche contenuta entro lo 0,8%. In conclusione, questo amplificatore, anche se non rispetta perfettamente le caratteristiche dichiarate dal costruttore, fondamentalmente per quanto dichiarato circa la distorsione armonica, presenta comunque buone caratteristiche audio fino a $6 + 6$ W su 8Ω e fino a $10 + 10$ W su 4Ω e può risultare sicuramente un'ottima scelta se si privilegia la linearità di risposta su tutta la banda audio e se si vuole valutare una nuova esperienza di riproduzione da confrontare con altri modelli di architettura differente, magari anche con un classe A, perché no?! Ci riserviamo in futuro di reperire e testare qualche altro prodotto in classe "T" perché sembra un sistema veramente interessante.

MODULO YDA138

L'ultimo modulo di cui ci occuperemo in questo capitolo dedicato ai moduli di potenza, è un insolito modulo che utilizza un chip monolitico YAMAHA con sigla: YDA138 (Fig. 17). Il costruttore lo definisce in architettura: "Pure Pulse Direct Speaker Drive Circuit", con modulazione bilanciata a circa 500 kHz; il che lo farebbe rientrare negli amplificatori di classe "D". Dalle forme d'onda (semplici positive e negative) che abbiamo rilevato Fig. 18 a noi sembra più simile ad un classe "H" ovvero una classe "AB digitale".

La classe H è un miglioramento della classe G e funziona variando continuamente la tensione di alimentazione quando il segnale audio supera un particolare livello di soglia. La tensione di alimentazione tiene traccia del livello di picco del segnale in modo da essere solo leggermente superiore al valore istantaneo dell'onda audio, tornando al suo livello inferiore una volta che il valore di picco del segnale scende di nuovo al di sotto del livello di soglia funzionale. Ciò che rende molto interessante questo dispositivo è che dispone altresì di una apposita uscita per cuffia funzionante effettivamente in classe AB con potenza massima di 50 mW (30mW ad alta fedeltà) per ogni canale, con centrale a massa su un carico di 32Ω . Quando si inserisce il jack nella presa cuffia, vengono inibite automaticamente le uscite di potenza, come su un amplificatore professionale e molto inusuale per un prodotto di questa classe per utilizzazione consumer. Altra peculiarità che lo rende particolarmente versatile, è che le due uscite possono

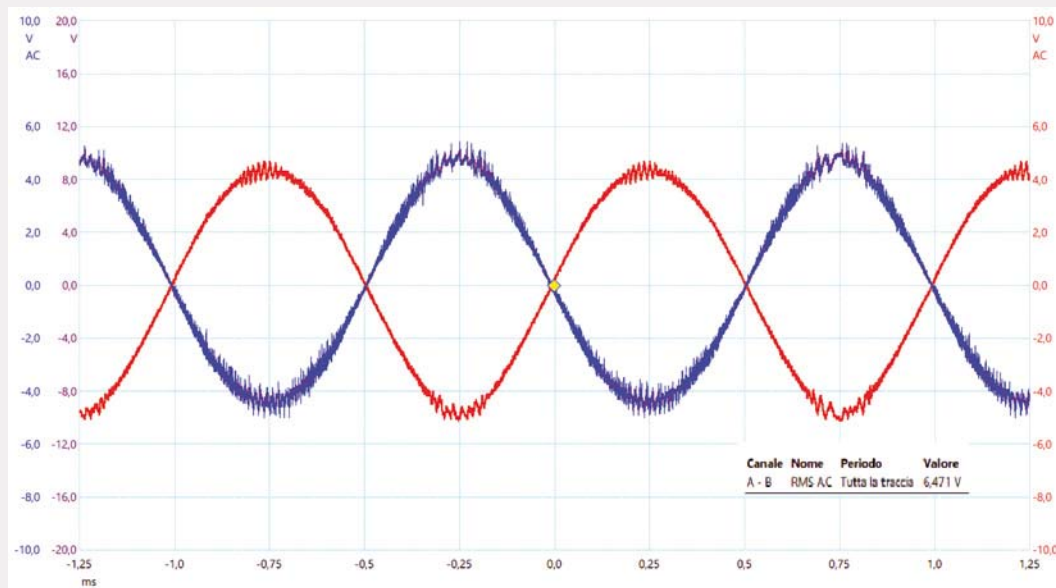
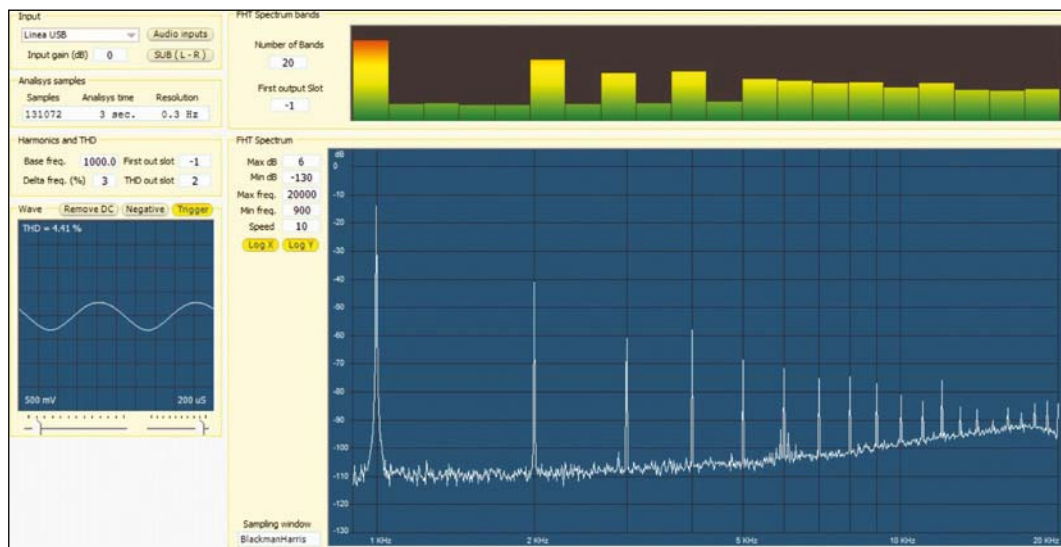


Fig. 15
Forme d'onda su una singola uscita e segnale a 10,5W- 4Ω del modulo AMPLITA2020

Fig. 16
Distorsione
armonica del 4,2%
a 10,5W -4Ω, del
modulo
AMPLITA2020.



essere collegate a ponte su un carico di 4Ω, ottenendo quindi un'uscita mono con potenza doppia. Anche questo amplificatore dispone, on-board, di controllo di volume e il guadagno massimo può arrivare fino a 36 dB.

Nella **Tabella 9** sono riassunti, come sempre, i dati fondamentali dichiarati dal costruttore. Infine, nella **Tabella 10**, troviamo i dati rilevati dalle varie prove, in differenti condizioni. Possiamo constatare che i dati forniti dal costruttore sono abbastanza coerenti con quanto riscontrato e pertanto il progetto, in questo caso, risulta ben

fatto. Pertanto possiamo prelevare senz'altro 7+7 watt a 8 Ω e 15 watt mono (in configurazione a ponte). La risposta in frequenza di questo amplificatore è veramente ottima, risultando molto lineare a partire da 100 Hz, fin oltre i 100 kHz. In **Fig. 19** riportiamo un dettaglio dello spettro di frequenza dei due canali fino a 2kHz, dove si può osservare la linearità di riproduzione sulle basse frequenze. La riproduzione in cuffia risulta molto buona fino a 30 mW, poi inizia a distorcere, fino a raggiungere livelli non più accettabile intorno ai 50 mW, come d'altra parte, dichiarato dal costruttore.

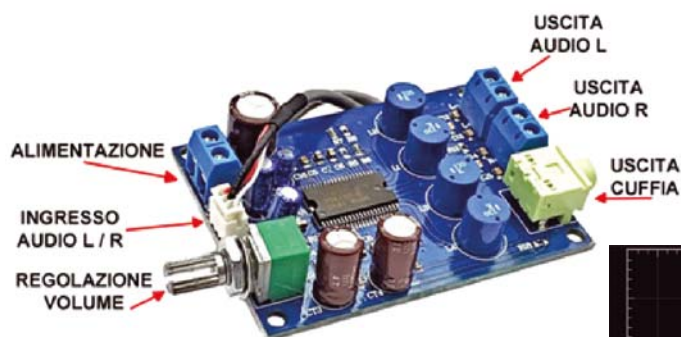
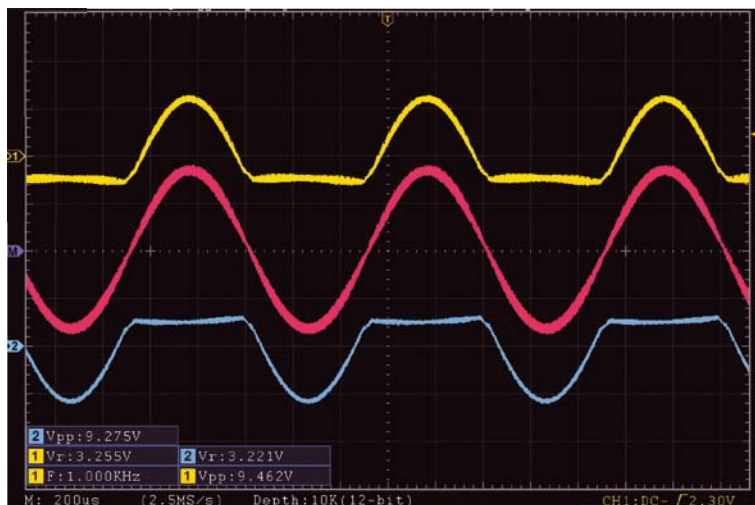


Fig. 17
Modulo
AMPLYDA138 con
relativi collega-
menti.

Fig. 18
Forme d'onda presenti
su un solo canale e
onda risultante dalla
differenza matemati-
ca relativa al modulo
AMPLYDA138.



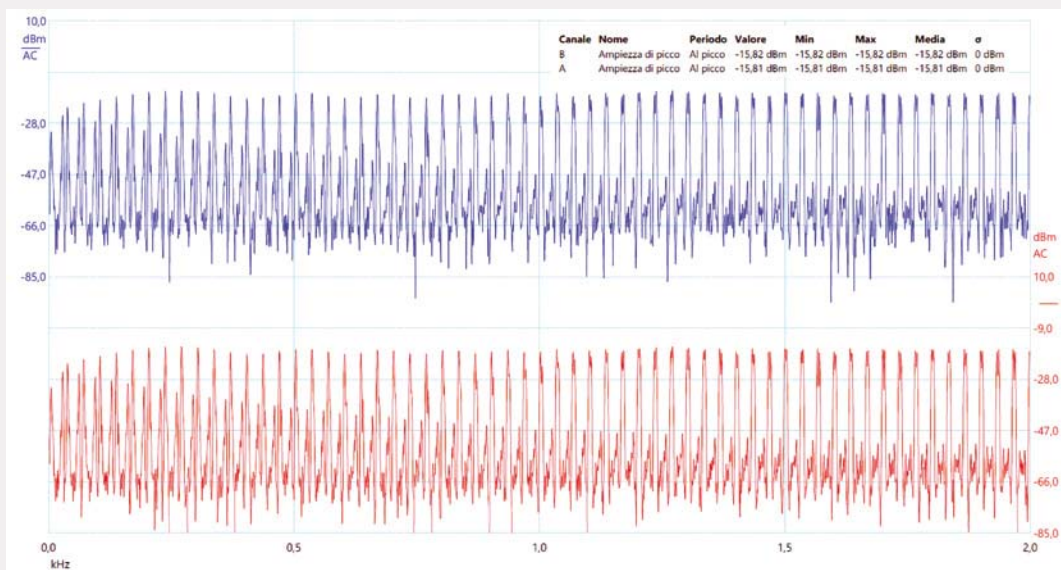


Fig. 19
Spettro della banda passante dei due canali (L-R) relativo alla risposta di frequenza fino a 2KHz del modulo AMPLYDA138

Però 30 mW in cuffia sono più che buoni, per ascoltare senza lesionarci i timpani. L'unica cosa strana che abbiamo riscontrato è lo sbilanciamento di segnale, quando si prelevano 50 mW. Nelle ultime due righe della tabella possiamo infatti osservare, che a parità di segnale in ingresso, il segnale di uscita e quindi la potenza erogata, risulta

inferiore su un canale rispetto all'altro e paradossalmente il canale con meno potenza presenta anche maggior distorsione. Nelle **Fig. 20** e **Fig. 21** si può osservare chiaramente questo dettaglio, che tutto sommato risulta non significativo se non si superano i 30 mW di uscita, perché in questo caso risulta tutto perfettamente speculare e lineare.

Tabella 9
Dati del costruttore relativi al C.I. YDA138.

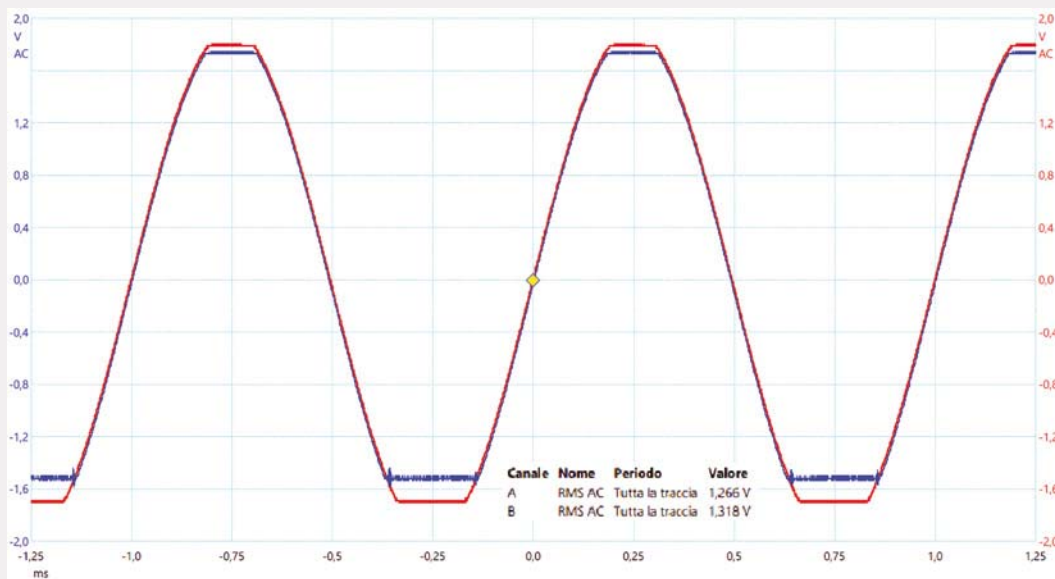
Parametri	Condizioni di test / note	Min	tip	max	Unità di misura
Vs tensione di alimentazione		9	12	13,5	V
Impedenza altoparlante	(*) a ponte - 32Ω presa cuffia	4(*)	8	32	Ω
Guadagno Gv /Av	Con regolazione volume	18		36	dB
Potenza di uscita	12V - 1KHz- 8Ω THD= 10%		10+10		W
Potenza di uscita a ponte 1 CH	12V - 1KHz- 8Ω THD= 10%		20		W
Distorsione THD+N	12V - 1KHz - 1W – 8Ω		0.02		%
Potenza di uscita cuffia	12V – 1KHz - 32Ω – THD+N 0.01%		30		mW
Potenza di uscita cuffia	12V – 1KHz - 32Ω – THD+N 10%		50		mW
Efficienza	Po = 10W - 8Ω			88	%

Tabella 10
Parametri rilevati sul modulo AMPLYDA138.

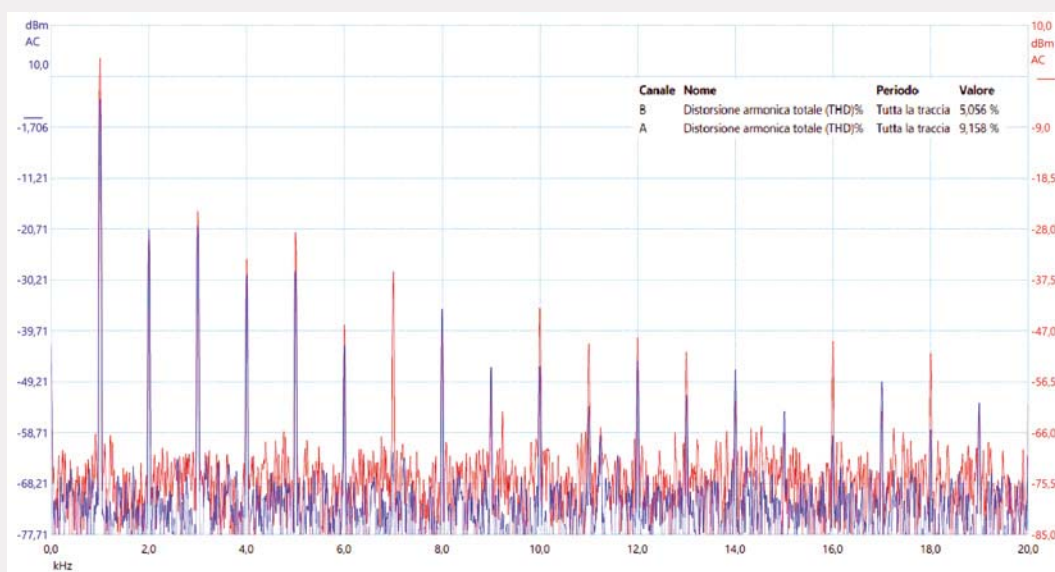
V_ALIM	CH	P (W)	Vpp In	Vout_rms	Po_8Ω (W)	Av (dB)	THD %	THD % 3a	rend. %
12	A/B	8,15	1	7,48	7,0	26,5	0,35	0,11	86
12	A/B	10,8	1,2	8,6	9,2	26,1	10	8,5	86
V_ALIM	CH	P (W)	Vpp In	Vout_rms	Po_4Ω* (W)	Av (dB)	THD	THD % 3a	rend. %
12	Bridge	17	0,6	7,69	14,8	31,2	1,02	0,31	87
12	Bridge	21	0,7	8,57	18,4	30,8	8,05	7,34	87
V_ALIM	CH	P (mW)	Vpp In	Vout_rms	Po_32Ω (mW)	Av (dB)	THD %	THD % 3a	rend. %
12	A/B	400	0,73	1	30,3	11,8	0,32	0,06	18
12	A	510	1	1,27	50,4	11,1	8,7	6,05	23
	B	510	1	1,32	54,5	11,4	4,8	3,42	25



➔ **Fig. 20**
Differenza
del segnale di
uscita tra i canali
stereo a 32Ω (cuffia)
a 50 mW $\approx$ del modulo
AMPLYDA138.



➔ **Fig. 21**
Differenza di
distorsione tra i
canali stereo a 32Ω
(cuffia) a 50 mW $\approx$
del modulo
AMPLYDA138.



CONSIDERAZIONI FINALI

Con questa terza parte dedicata ai moduli amplificatori di potenza media con il miglior rapporto qualità prezzo, termina la nostra recensione sugli amplificatori audio.

Abbiamo cercato di essere il più sintetici possibile, senza però tralasciare le informazioni e i dati più rilevanti al fine di poter indirizzare ciascuno necessiti di un amplificatore di bassa / media potenza, di poter effettuare la propria scelta in modo oculato e selettivo. Non mancheremo in futuro di recensire nuovi moduli, man mano che saranno disponibili sul mercato, scegliendo tra quelli che riterremo maggiormente interessanti e innovativi e che utilizzeranno nuove tecnologie, sicuramente sempre più integrate e performanti. ■



Cosa occorre?

I moduli presentati in questo articolo sono disponibili presso Futura Elettronica. L'amplificatore con TDA7297 (cod. AMPLI2X15W) è in vendita a Euro 7,00, il modulo con PAM8610 (cod. PAM8610) è disponibile a Euro 8,50, la scheda con TDA7492 (cod. AMPLID2X50W) costa Euro 25,00, l'amplificatore con TA2020-020 (cod. AMPLITA2020) è disponibile a Euro 22,00, il modulo con YDA138 (cod. AMPLYDA138) è in vendita a Euro 12,00. 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

SMART HOME COMFORT CONTROL

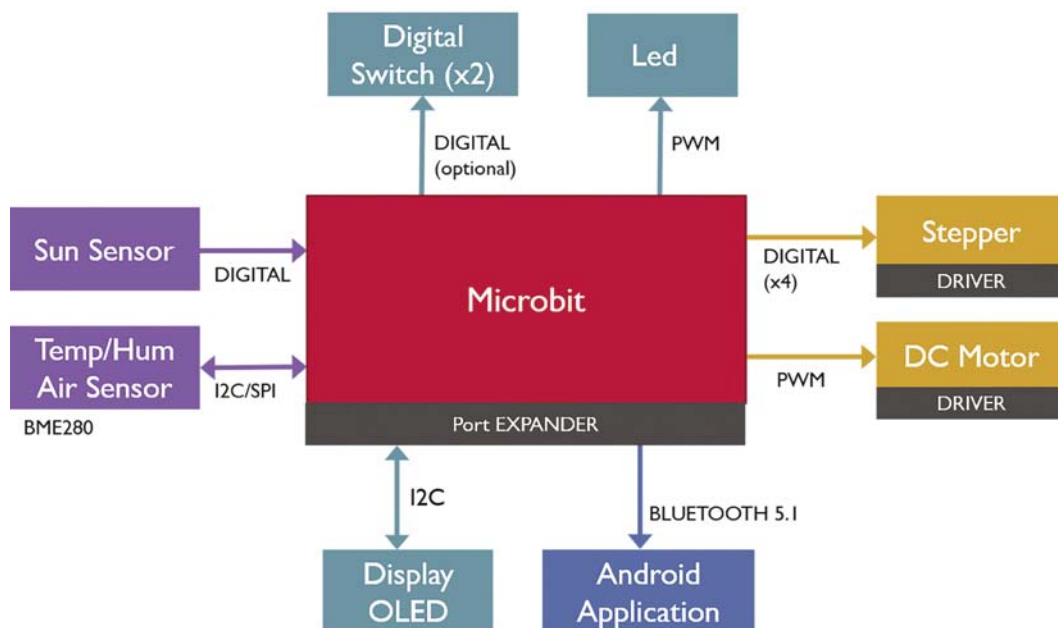
di MARCONATALE PARISE
e VINCENZO GERMANO



Aggiungiamo funzionalità al nostro progetto di domotica sviluppato con micro:bit e in questo appuntamento concentriamoci sulla gestione del comfort domestico.

La domotica è lo studio e l'applicazione di tecnologie basate sull'ingegneria informatica/elettronica, che mirano alla realizzazione di dispositivi integrati in grado di automatizzare e facilitare varie operazioni solitamente svolte in una casa. Tutto ciò per ottimizzare i consumi e/o le funzionalità di un appartamento, l'automatizzazione di alcune attività o anche la gestione ad alto livello di sistemi che rendono l'utente finale capace di controllare diversi strumenti attraverso tecnologie dedicate; perciò la domotica raccoglie quell'insieme di automazioni e tecnologie domestiche, che puntano a migliorare l'abitabilità e la funzionalità delle nostre case. Com'è facile immaginare la domotica è orientata a rendere intelligenti i dispositivi, gli impianti elettrici e i sistemi degli appartamenti; in quest'ottica, a partire dal fascicolo n° 264 stiamo proponendo il presente sistema domotico basato sulla scheda micro:bit per guidarvi

Fig. 1
Schema a Blocchi.



nel controllo di funzionalità domestiche attraverso la gestione di differenti tecnologie.

Nei precedenti progetti di domotica con micro:bit abbiamo spiegato come controllare in modo automatico o manuale l'impianto di illuminazione domestico facendo, per semplicità di trattazione, un'equivalenza con l'accensione di un singolo LED, ma chiarendo che al posto di esso è possibile inserire per esempio la gestione di un relé o una board elettronica che pilota una o più strisce LED, lampade, neon oppure un intero impianto d'illuminazione domestico. Non solo, abbiamo realizzato e sviluppato un'applicazione dedicata, analizzato come modificarla ed evolverla a proprio piacimento per centralizzare la gestione del sistema di domotica.

A questo punto continuiamo ad aggiungere funzionalità al sistema mettendo in pratica un'altra equivalenza come fatto per la gestione LED, ovvero aggiungiamo la gestione di una ventola per il ricircolo d'aria forzato nell'appartamento che può essere gestita manualmente o in automatico a seconda delle necessità dell'utente e di parametri ambientali come la temperatura e/o l'umidità. In aggiunta consideriamo anche la gestione di tapparelle/persiane utilizzando un motore stepper, per poi visualizzare le informazioni su un display LCD. Prima di passare all'implementazione software su micro:bit e alla presentazione dell'app Android, analizziamo tutto l'hardware necessario, illustrandone le caratteristiche di ogni componente del progetto.

SETUP DI SISTEMA

Per una maggiore comprensione del sistema, in **Fig. 1** riportiamo uno schema a blocchi nel quale è immediatamente visibile il micro:bit al centro di tutto il sistema, a cui vengono connessi il sensore di temperatura/umidità (per rilevare le condizioni ambientali), un motore DC (corrente continua o Direct Current) che viene pilotato mediante un driver (utile per la gestione di una ventola per il ricircolo forzato dell'aria in un appartamento), il motore stepper (o passo-passo) gestito mediante il corrispondente driver (che può essere pensato per aprire/chudere una tapparella o saracinesca) ed eventualmente due switch di fine corsa associati al motore stepper per permettere al micro:bit di rilevare le due battute di fine corsa. Infine, mediante Bluetooth comunica con il proprio smartphone e tramite LCD vendono visualizzati i messaggi di sistema. Nello schema a blocchi vengono messi in risalto anche le tipologie di comunicazione o i particolari protocolli utilizzati per lo scambio di dati tra i vari blocchi. Per agevolare il reperimento del materiale di questo progetto e per evitare la realizzazione di board dedicate, mostreremo alcune soluzioni di schede elettroniche integrate "pronte all'uso" a costi contenuti che permettono la realizzazione delle funzioni d'interesse, reperibili facilmente sul sito futurashop.it.

LA SCHEDA D'ESPANSIONE

Cominciamo la trattazione dei componenti aggiuntivi con la scheda d'espansione.

Purtroppo, in contrapposizione con i suoi tanti vantaggi, il connettore del micro:bit è un elemento un po' troppo vincolante rispetto ad altre schede equivalenti "Open" tipo Arduino, Raspberry Pi e simili, che facilmente si riescono a connettere con board aggiuntive mediante *pin header* (con passo 2,54mm e singolo pin da 1mm); perciò, per permetterne un facile utilizzo e sfruttare a pieno tutte le sue caratteristiche di potenza computazionale, flessibilità sui protocolli di comunicazione grazie all'astrazione hardware delle librerie di programmazione ma anche e soprattutto i pin di input/output, si ha necessità di una scheda d'espansione.

Il connettore, pur rimanendo molto compatto (riportato in **Fig. 2** con anche visibili le funzionalità dei vari pin) non è standard e quindi è doveroso parlare di una delle schede d'espansione presenti in commercio, che agevola le connessioni, permettendo una facile interconnessione verso il "mondo esterno".

Tra le varie soluzioni disponibili, quella più compatta e flessibile è sicuramente un "Port Expander" della ALSTROBOT (acquistabile sullo store futurashop.it col codice MICROBITPROTO) in grado di rendere di facile utilizzo tutte le porte di input/output della board.

In **Fig. 2** è visibile il TOP della scheda e si comprende rapidamente come il connettore del micro:bit possa essere reso usabile/accessibile mediante pin header colorati in vario modo, equivalentemente a quanto accade con Arduino, Raspberry Pi o altro.

Oltre l'immediata possibilità di riportare la connessione in un ambito "standard", ci sono vari vantaggi aggiuntivi, come la possibilità di essere alimentata tramite i due connettori rossi, con tensioni variabili fino a 24V o fino a 6V e poter sfruttare uno dei due regolatori stabilizzati di tensione da 5V o 3.3V; oppure la possibilità della scheda di essere alimentata direttamente tramite la porta USB di micro:bit.

Riguardo i pin header con passo 2,54 mm, si hanno le seguenti funzionalità:

- colore giallo e blu per le porte di I/O e protocolli di comunicazione (I²C, SPI e Seriale);
- colore blu parzialmente per il riferimento GND;
- colore rosso per le tensioni d'alimentazione.

Un ulteriore vantaggio è dato dalla disponibilità di un interruttore a due posizioni (on/off) che permette di accendere e spegnere direttamente la scheda. Con questo è facile immaginare come l'utilizzo di questa board risulta essenziale quando si vogliono connettere immediatamente componenti esterni al connettore micro:bit, come ad esempio schede di sensori che richiedono specifici collegamenti fisici magari complessi da collegare direttamente al micro:bit.

IL SENSORE SOLARE

A questo punto per rilevare l'intensità luminosa, tra i vari moduli disponibili sul mercato la scelta è ricaduta sulla scheda, visibile in **Fig. 3** (cod. LIGHTSENSOR), acquistabile sullo store futurashop.it.

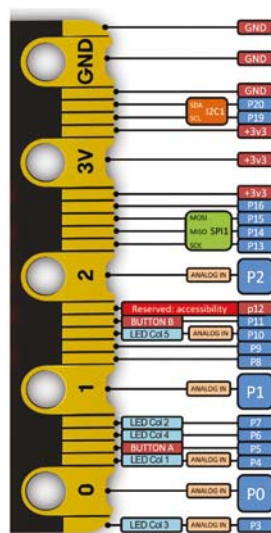
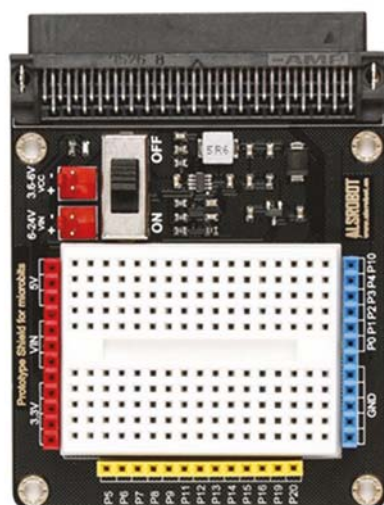
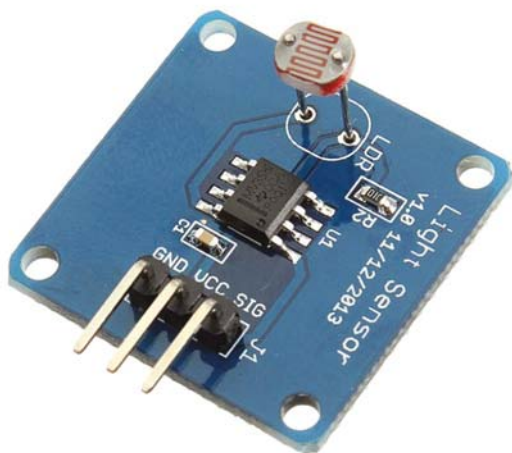


Fig. 2
Port
Expander e
pinout.

Fig. 3
Modulo Sensore
Solare.



Questa board ha una resistenza fotosensibile (la fotoresistenza o il fotoresistore è una resistenza composta da uno strato semiconduttore che reagisce alla radiazione luminosa, sia emessa dal sole che da altra fonte luminosa) che consente di rilevare la luminosità della luce ambientale. Il funzionamento di una fotoresistenza, è molto semplice perché varia il valore da 0Ω alla sua resistenza massima 10, 50, 100K Ω , in base alla tipologia di fotoresistenza, quando passa da una condizione di illuminazione piena al buio. In pratica se la luce è sufficientemente forte il suo valore in ohm è tendente a zero, mentre man mano che la luce diminuisce, il suo valore aumenta fino alla resistenza massima. Quindi riassumendo è un componente elettronico la cui resistenza è inversamente proporzionale alla quantità di luce che lo colpisce e ciò comporta che la corrente elettrica che transita attraverso tale componente è proporzionale all'intensità di una sorgente luminosa. In tale maniera si realizza una sorta di

potenziometro attuabile tramite la luce, anziché tramite forze meccaniche o segnali elettrici. Come apprezzabile dalla **Fig. 3**, la scheda ha un pin header da 3 pin, ovvero:

- **VCC**: alimentazione 3,3+5V;
- **GND**: zero volt;
- **SIG**: uscita del segnale da 0V (valore) a 5V (valore alto).

Tale segnale è possibile collegarlo direttamente al micro:bit per poter rilevare i cambiamenti dell'intensità della luce ambientale. Questa scheda, come altre, ha anche dei comodi fori di fissaggio per una facile installazione mediante delle viti.

IL DISPLAY OLED

Come analizzato nello schema a blocchi di **Fig. 1**, uno dei componenti "gestiti" da micro:bit è il display OLED (Organic Light-Emitting Diode), ovvero diodo organico ad emissione di luce visibile in **Fig. 4**, che a differenza di un comune display LCD (Liquid Crystal Display), non necessita di una fonte luminosa esterna per essere visibile ma è direttamente in grado di emettere luce propria; ne consegue che possono essere realizzati display dalle ridottissime dimensioni e peso, come in questo caso circa 29 x 27 x 5 (mm) e solo 3 grammi. Questo, oltre ad altre caratteristiche, lo rende un ottimo sostituto al display LCD. Può essere acquistato con due risoluzioni/dimensioni differenti, 128 x 64 e 128 x 32, angolo di visione circa 180°, dotato di un buon contrasto; naturalmente la risoluzione da 32 avrà la parte del display dimezzata. In più, viene prodotto con differenti colorazioni, sia modelli monocromatici (maggiormente diffusi ed economici), che modelli bicromatici. Questi ultimi, a differenza dei primi che presentano una sola colorazione, hanno una striscia ben definita nella parte alta del display di un colore differente rispetto la sottostante (i colori generalmente sono giallo e blu). Possono essere utili in svariate applicazioni, come ad esempio nel caso in cui si vogliano differenziare le informazioni da mostrare nella parte alta - si pensi alla data, allo stato di una batteria e via dicendo - per lasciare la parte inferiore ai contenuti della particolare applicazione. Tuttavia, per il nostro scopo abbiamo selezionato un display monocromatico 128x64 di colore blu. Nella maggior parte dei casi, questi display OLED utilizzano il driver SSD1306 on-board per l'interfacciamento, che mette a disposizione due protocolli di comunicazione possibili, SPI (Serial Peripheral Interface) e I2C

Fig. 4
Display OLED.



(Inter Integrated Circuit), nel nostro caso I²C; con tensione di alimentazione dai 3,3V ai 5V circa. Perciò tolti i pin di alimentazione e zero volt (GND), si riesce a pilotarlo agevolmente soltanto mediante gli altri due pin dell'I²C, SDA (Serial Data) ed SCL (Serial Clock).

IL SENSORE DI TEMPERATURA

Passiamo all'analisi della componentistica utilizzata iniziando da uno dei più complessi, ovvero il BME280 (cod. GY-BME280), che verrà impiegato per rilevare la temperatura d'ambiente. Questo (**Fig. 5**) è un sensore di pressione barometrica assoluta, particolarmente adatto per applicazioni mobili, le sue piccole dimensioni e il suo basso consumo energetico ne consentono l'implementazione in dispositivi alimentati a batteria come telefoni cellulari, moduli GPS o orologi. Si basa sulla comprovata tecnologia di sensori di pressione piezoresistivi di Bosch, caratterizzata da elevata precisione e linearità, nonché stabilità a lungo termine ed elevata robustezza EMC (Electromagnetic Compatibility o compatibilità elettromagnetica). Perciò possiamo considerarlo come un nuovo sensore ambientale per temperatura, umidità e pressione barometrica che insieme al BMP280, sono l'evoluzione della precedente serie BMP085/BMP180/BMP183. Questo prodotto è in grado di effettuare misurazioni molto "fini" con un'accuratezza assoluta di ± 1 hPa per la pressione, $\pm 3\%$ per l'umidità e di $\pm 1.0^\circ\text{C}$ per la temperatura, grazie alle quali può essere utilizzato anche come altimetro con un'accuratezza di 1m. Numerose opzioni di funzionamento del dispositivo garantiscono la massima flessibilità, in più è ottimizzato in termini di consumo energetico, risoluzione e prestazioni. Per dare un'idea, il chip consente di effettuare letture con un campionamento selezionabile e una risoluzione che varia da $0,0050^\circ\text{C}$ a $0,0003^\circ\text{C}$, con consumi dipendenti da una delle seguenti possibili modalità operative:

- **Sleep mode:** non vengono effettuate misure e i consumi sono ridotti al minimo;
- **Force mode:** viene effettuata una singola misura con successiva transizione in sleep mode;
- **Normal mode:** consiste in cicli di lettura continui interrotti da un periodo di inattività.

Senza addentrarci troppo in tecnicismi, dovendolo utilizzare come sensore di temperatura ambientale, possiamo rapidamente al suo utilizzo ovvero alla possibilità di connetterlo con il micro:bit.

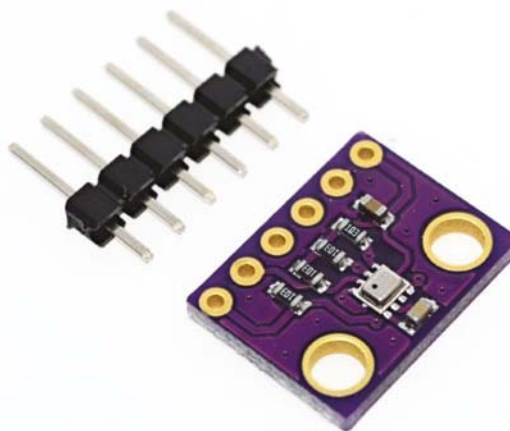


Fig. 5
Breakout
board del
sensore
BME280.

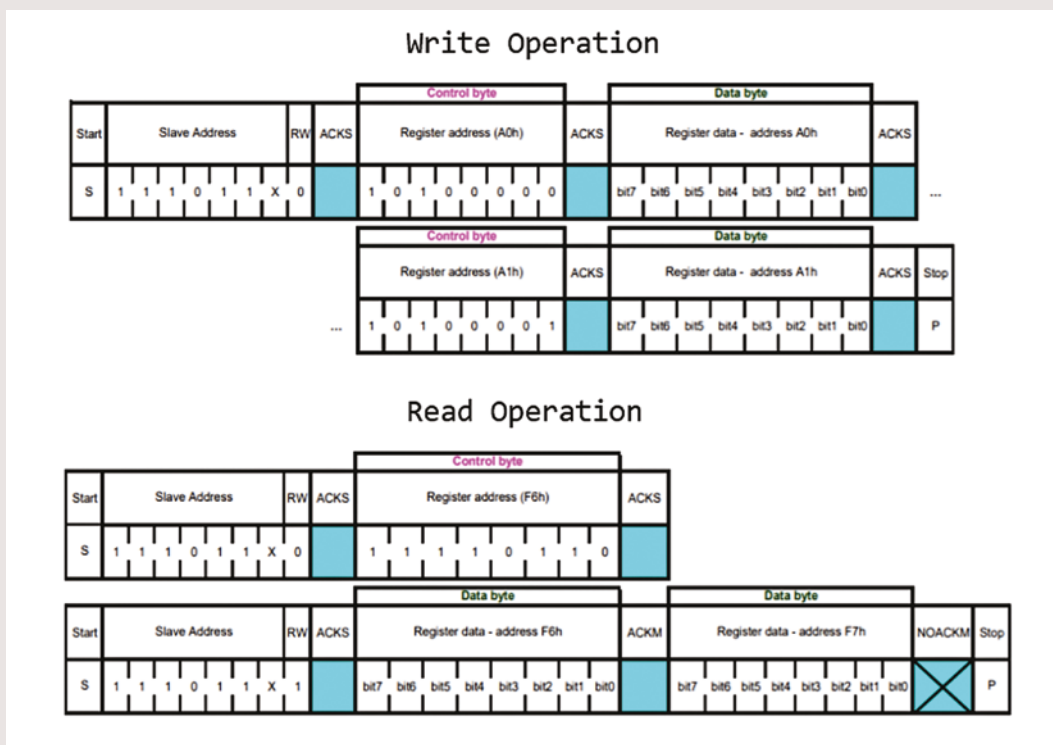
In commercio esistono varie schede già realizzate e acquistabili con costi ridotti (attraverso i più diffusi canali di vendita) come quella in **Fig. 5**. La scheda presenta un regolatore a 3.3V per l'alimentazione stabilizzata del sensore e un convertitore di livello per funzionare sia con logica a 3.3V che a 5V; è possibile effettuare letture di pressione e temperatura abilitabili individualmente, che vengono rese disponibili tramite opportuni registri al quale è possibile accedere mediante i protocolli di comunicazione SPI (Serial Peripheral Interface) oppure I²C (Inter Integrated Circuit).

Per questa implementazione è stata scelta l'interfaccia I²C, abilitabile tramite il pin CS (Chip Select), grazie ad un opportuno collegamento alla tensione di alimentazione tramite una resistenza del valore di 1K Ω . Come in tutti i dispositivi basati su questo protocollo di comunicazione, è possibile interrogare il dispositivo tramite un determinato indirizzo (indirizzo slave) inserito nella stringa; l'indirizzo slave si compone di 7bit (111011x) di cui 6 statici e uno dinamico che varia in base al desiderato, ovvero collegando il pin SDO a GND si genera l'indirizzo 0x76 (1110110) oppure collegandolo alla tensione di alimentazione (ad esempio tramite una resistenza di 1K Ω) si genera l'indirizzo 0x7 (1110111). Nel nostro caso l'indirizzo scelto del dispositivo sarà 0x76. Per completezza, le linee SCK (Serial Clock) ed SDI (Serial Data Input) dovranno essere collegate ai rispettivi pin lato micro:bit, con l'accortezza di connettere la linea SDI alla tensione di alimentazione tramite una resistenza (va bene anche un valore dell'ordine di 1K Ω). I frame di lettura e scrittura tramite protocollo I²C sono illustrate in **Fig. 6**. Vediamoli brevemente entrambi.

L'operazione di scrittura consiste nell'inviare:

- l'indirizzo del dispositivo (slave) con il bit RW impostato a 0;

➔ **Fig. 6**
Protocollo I²C.



- la coppia di registri denominati "Register address" che rappresenta il registro che si vuole scrivere e il registro "Register data" che dovrà contenere il dato che dovrà essere scritto nel "Register address".

La procedura termina quando è intercettata la condizione di stop (P) caratterizzato da una transizione da basso ad alto del SDA mentre il SCL è alto.

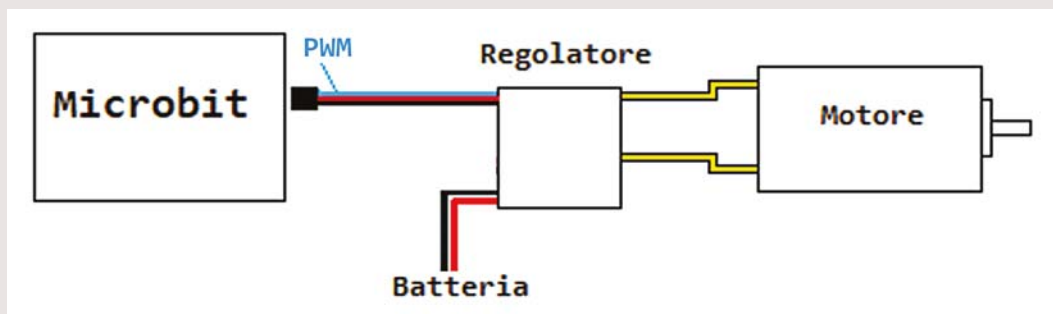
Invece per l'operazione di lettura avviene questo:

- si configura il bit RW a 0 e si invia il registro denominato "Register address" da leggere;
- con una nuova condizione di start "S", si configura il bit RW a 1;
- successivamente il dispositivo invierà il dato del registro;

- la procedura terminerà con un NOACKM ed una condizione di stop (P).

Le diciture ACKM, ACKS sono definite come *Acknowledge* lato master oppure lato slave che può indicare l'inizio di un'operazione indifferenziata da una forzatura bassa della linea dati durante l'impulso alto del clock (SCL). I registri interessati nel caso della lettura della temperatura sono 0xFA, 0xFB e 0xFC. Per i meno esperti, fortunatamente, come vedremo successivamente le librerie software utilizzate ci astrarranno da tutti questi dettagli e impostazioni, permettendoci di semplificare enormemente lo sviluppo software e rendendo la sua configurazione immediata, illustrando le funzioni più importanti per implementare la lettura dei dati d'interesse.

➔ **Fig. 7**
Schema controllo motore pompa acqua.



IL MOTORE PER LA VENTOLA E IL DRIVER

Considerando la funzionalità di una ventola che possa forzare il ricircolo dell'aria in un appartamento, il motore più indicato è un motore DC, che necessita di una circuiteria aggiuntiva per il suo pilotaggio. A tale scopo è stato scelto un driver comunemente utilizzato anche per il modellismo dinamico, ovvero basato su MOSFET (*Metal Oxide Semiconductor Field Effect Transistor*), nient'altro che un circuito elettronico integrato in grado di regolare la velocità del motore in base ad un segnale di pilotaggio, ma vediamo meglio come funziona.

Le interconnessioni possono essere schematizzate come nella **Fig. 7**, ovvero per semplicità possiamo dire che il blocco driver ha a bordo l'elettronica che pilota (mediante le due fasi) il motore e ne varia la velocità a seconda del segnale d'ingresso proveniente dal micro:bit, perciò sono previsti:

- un'alimentazione esterna proveniente generalmente da una batteria a 12V in grado di erogare fino a 1A, con la quale viene alimentato direttamente il motore;
- un connettore con un segnale di controllo e le alimentazioni per l'interfaccia digitale.

Il segnale di controllo è identificato come **PWM** (*Pulse Width Modulation*), una tecnica di modulazione che consente di variare la larghezza dell'impulso con periodo costante; perciò il driver acquisisce questo segnale e genera un segnale d'uscita destinato a regolare la velocità del motore che consentirà di forzare il ricircolo dell'aria. L'ingresso PWM del regolatore dovrà essere collegato al micro:bit che si occuperà del controllo e quindi della generazione del PWM come vedremo a breve nel setup di sistema. Per completezza e per dare un'idea di quanto appena illustrato, si riporta in **Fig. 8** il driver di esempio.

DRIVER STEPPER E MOTORE

Passando oltre e pensando a come poter aprire/chiedere una tapparella, la soluzione migliore è sicuramente un motore stepper e non un DC, perciò tra i vari moduli e motori disponibili sul mercato la scelta è ricaduta su quanto visibile in **Fig. 9**, acquistabile sullo store *futurashop.it* con il codice SM2003DB. Questo perché permette una soluzione tutta in uno, ovvero una scheda per la gestione del motore ed in più il motore associato; essendo poi uno stepper (passo-passo) e non un DC, si ha la possibilità di avere un controllo puntuale sulla posizione della tapparella in modo

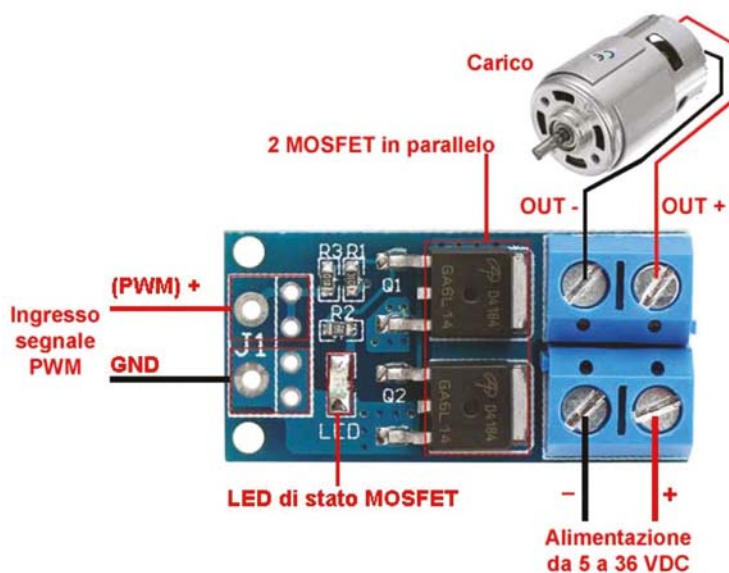


Fig. 8
Driver MOSFET
in PWM.

preciso e direttamente legato agli step/passi di rotazione impostati.

Ma vediamo qualche dettaglio in più.

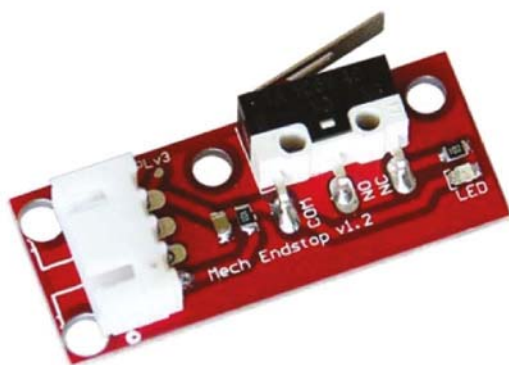
Come appena anticipato, il set acquistabile è composto da una scheda di controllo basata su un driver 7 canali (Darlington NPN) per motori passo-passo mediante il chip ULN2003 e da un motore unipolare stepper di tipo 28-BY48. Questa tipologia di motori non sono altro che dei dispositivi elettromagnetici che si muovono con passi discreti ben definiti; internamente vi sono varie bobine con un albero centrale su cui sono agganciati una serie di magneti, le bobine vengono alternativamente eccitate con correnti che generano campi magnetici, i quali attraggono o respingono i magneti sull'albero, facendo ruotare il motore.

Dotato di cinque fili e quattro bobine, ha le connessioni centrali di quest'ultime tutte



Fig. 9
Stepper
Driver e
motore
stepper.

Fig. 10
Switch di fine corsa.



interconnesse tra loro e utilizzate come alimentazione e il nome unipolare deriva proprio dal fatto che la potenza arriva sempre su questo polo. Considerando la scheda, l'unico integrato presente è l'ULN2003A dedicato alla gestione del motore (per maggiori dettagli si faccia riferimento al datasheet), poi troviamo 4 diodi LED con le corrispondenti resistenze di polarizzazione, che corrispondono alle quattro fasi (A – B – C – D) del motore passo-passo e vari connettori. Per poter utilizzare lo stepper, le quattro fasi di pilotaggio verso la scheda devono essere collegate ai pin-header J1; mentre attraverso il connettore J4 a 5 poli si deve collegare il motore alla scheda facendo accortezza alla polarità di connessione. Infine, il pin-header J3 serve per l'alimentazione, ovvero, se si vuole alimentare il motore a 5V si dovrà montare il jumper affinché la stessa tensione arrivi all'integrato U1 e al motore. Dato che un motore di questo tipo ha un assorbimento rilevante, potrebbe essere preferibile fornire l'alimentazione da una scheda apposita e non dalla scheda controllo. Si tenga presente che questa tipologia di motori può anche essere acquistata singolarmente in due versioni, a 5V e a 12V, a seconda della necessità; forniscono entrambe una buona coppia anche in stato di stop ma hanno un assorbimento importante da tener presente. In ogni caso, acquistando il set direttamente da *futurashop.it* si ha tutto il necessario per il progetto.

SWITCH DI FINECORSO

Infine, per permettere il rilevamento del finecorsa di una serranda/tapparella/saracinesca o altro, si ha la possibilità di utilizzare uno switch di finecorsa, visibile in **Fig. 10**, grazie al quale il micro:bit comprende quando essa arriva ad essere totalmente aperta o totalmente chiusa; per quanto scontato, se si vogliono rilevare entrambe

le precedenti posizioni ci vorranno due finecorsa, uno per ogni estremo. I finecorsa (cod. LMSW) non sono altro che dispositivi elettromeccanici che richiedono il contatto fisico di un oggetto con l'interruttore per far cambiare lo stato di un segnale rilevabile dall'elettronica. La scheda che fa al caso nostro è acquistabile sul sito di *futurashop.it* ed è costituita da una board con un micro switch e un LED di stato (generalmente utilizzata per rilevare i finecorsa di stampanti 3D o CNC), ha delle dimensioni ridotte ed è completa di cavetto con relativi connettori per le connessioni (il filo rosso è l'alimentazione, quello nero il GND e infine il filo verde il segnale di stato). Vista la semplicità del suo utilizzo viene lasciato come spunto di espandibilità del progetto ma successivamente non sarà integrata la sua connessione con micro:bit.

DESCRIZIONE CONNESSIONI

A questo punto vediamo brevemente le interconnessioni fisiche presentate a livello logico nel diagramma a blocchi di **Fig. 1** e il setup di sistema tra il micro:bit e le varie periferiche. Ripartiamo dal considerare il *port expander*, con riferimento all'immagine di **Fig. 2**, abbiamo scelto di utilizzare la tensione di alimentazione V1, il riferimento G ed i connettori gialli per le periferiche. Per quanto riguarda le tensioni di alimentazione è importante impostare la linea V1 su 5V, collegando i due jumper verdi come mostrato in figura. Per semplicità i pin d'interesse destinati alle periferiche vengono riportati in

Tabella 1 - Connessioni fisiche tra i pin header delle varie schede.

PIN	FUNZIONI
P0	Sun Sensor (Digital)
P1	Led
P8	PWm Fan
P12	Digital IN1 - Stepper
P13	Digital IN2 - Stepper
P14	Digital IN3 - Stepper
P15	Digital IN4 - Stepper
P19	I2C SCL - Display e BME280
P20	I2C SDA - Display e BME280

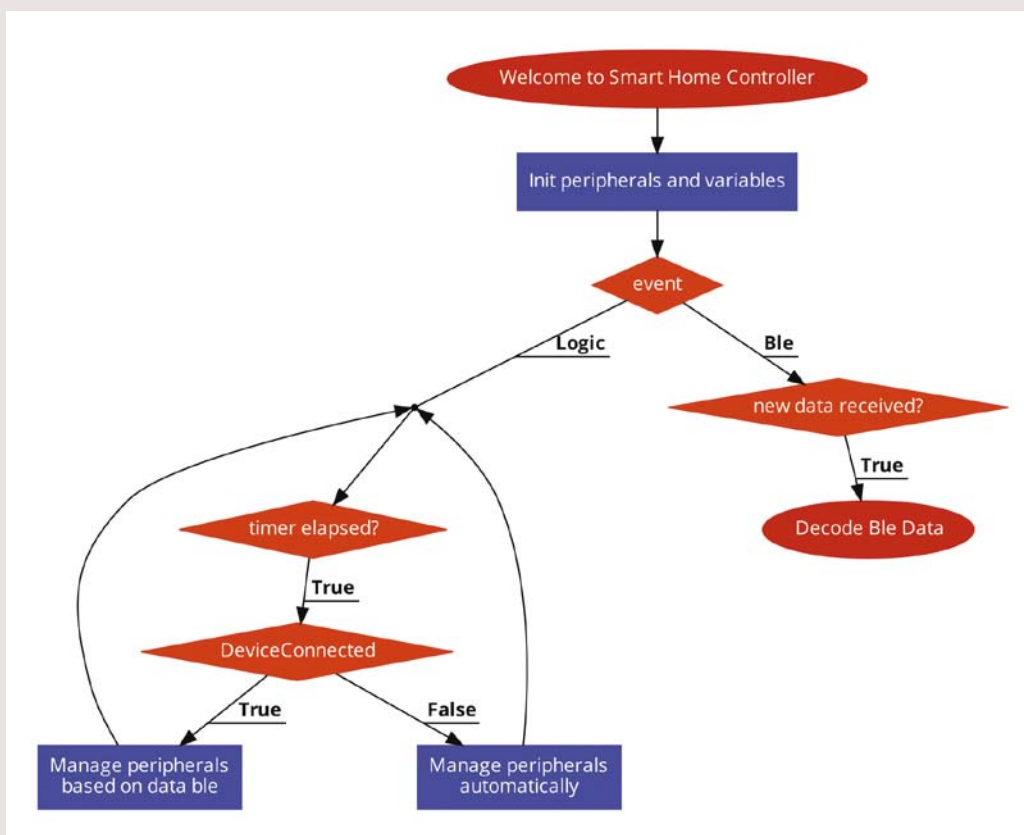


Fig. 11
Il diagramma di flusso del software.

Tabella 1 ed esplicitate le varie connessioni tra schede attraverso i loro pin-header, ma vediamo insieme. Le connessioni principali sono quelle che vanno da micro:bit al display OLED, alla scheda del sensore solare e al LED. Il primo collegamento prevede alimentazione, zero volt (GND) e protocollo di comunicazione I2C; mentre la seconda connessione oltre alle alimentazioni si ha solo una lettura digitale, perciò:

- **Vin e GND:** che rappresentano alimentazione e zero volt;
- **SCK e SDA:** necessari per la comunicazione (particolare attenzione per SDA che dovrà essere collegato in pull-up con una resistenza di 1KΩ);
- **Pin per:** lettura digitale;
- **Pin per:** gestione led in PWM;
- **3V:** non collegato.

Infine, per quanto riguarda il collegamento tra micro:bit e LED, basta prevedere una resistenza da 1kΩ per avere una corretta gestione. Per i meno esperti, come vedremo successivamente, le librerie software utilizzate ci astrarranno da tutti questi dettagli e impostazioni, permettendoci di semplificare enormemente lo sviluppo software e rendendo la sua configurazione immediata,

illustrando le funzioni più importanti per implementare la lettura dei dati d'interesse.

DESCRIZIONE LOGICHE DI FUNZIONAMENTO

La descrizione dell'implementazione software del sistema comfort di domotica richiede ulteriori dettagli sulle logiche implementative che verranno descritte attraverso il diagramma di flusso di **Fig. 11**.

La prima fase dell'implementazione riguarda l'inizializzazione della periferica bluetooth, del protocollo I2c che verrà utilizzato per la gestione del display OLED e del sensore di temperatura BME280, del driver stepper e delle variabili di appoggio per una corretta e leggibile implementazione.

La logica di funzionamento si basa su due eventi:

1. Ricezione dati bluetooth e successiva decodifica dei dati.
2. Attesa della scadenza del timer per gestire le funzionalità di illuminazione e comfort.

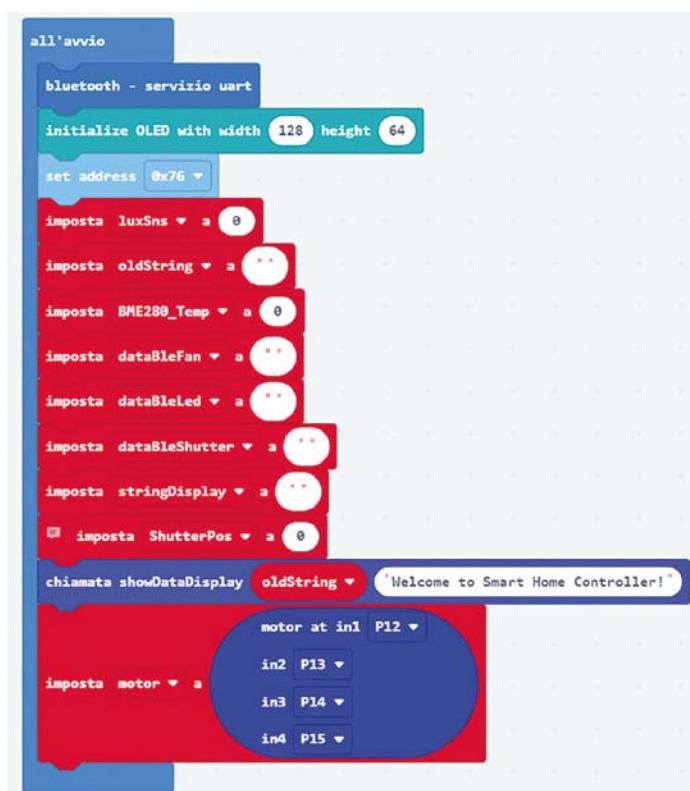
La decodifica dei dati bluetooth prevede l'estrazione dei dati relativi allo stato dell'illuminazione, del fan e della tapparella nel caso di gestione remota tramite opportuna

Fig. 12
Avvio software.

applicazione.
Le informazioni che il micro:bit riceverà avranno il seguente formato: **#LxFyySzz;**
Dove:

'L' = Led
'F' = Fan
'S' = Shutter
'x' potrà essere 0 o 1
'yy' avrà valore da 0% a 99%
'zz' avrà valore da 0% a 99%

Con il timer scaduto verrà valutato lo stato di connessione e nel caso di dispositivo connesso, le periferiche verranno gestite in base alle informazioni decodificate altrimenti, illuminazione e comfort verranno gestite in base al sensore solare e di temperatura.



IMPLEMENTAZIONE IN MICRO:BIT

La piattaforma che verrà utilizzata è il Make Code Block, ma prima di addentrarci nella scrittura del codice è necessario importare le librerie, quindi selezionare "Estensioni" e digitare le parole chiave

"Bluetooth" e "ssd1306" per il display OLED, facendo attenzione che la versione sia la 2.0.9 tramite il tasto approfondisci, "stepper-motor" per il motore ed infine "BMP280" per il sensore di temperatura. Cominciamo con il definire blocchi principali all'avvio

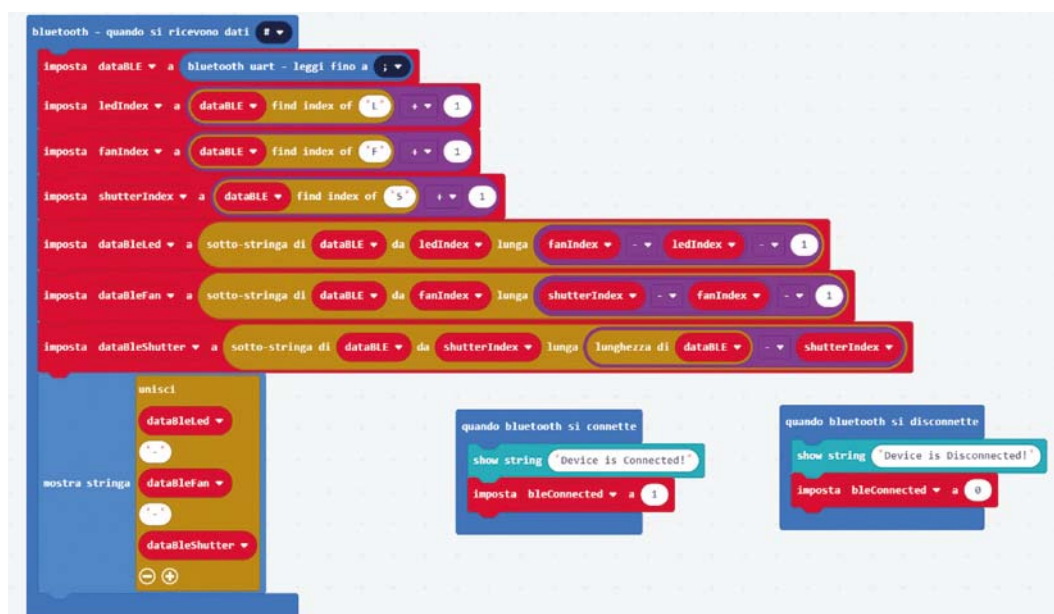


Fig. 13
Gestione BLE

del nostro codice e quindi l'inizializzazione della periferica Bluetooth del protocollo I²C relativo al display OLED e sensore BME280 e delle variabili per la gestione delle logiche di funzionamento.

Il primo blocco riguarda l'inizializzazione delle variabili e delle periferiche come mostrato in **Fig. 12**:

- inizializziamo la periferica Bluetooth tramite il blocco *bluetooth – servizio uart*
- *Initialize OLED with width 128 height 64* inizializza il display che abbiamo deciso di utilizzare con risoluzione 128x64
- le variabili *dataBleLed*, *dataBleFan*, *dataBleShutter* che rappresentano i dati bluetooth ricevuti e ripuliti
- la variabile *BME280_Temp* per i dati di temperatura acquisiti dal sensore
- *luxSns* per i dati acquisiti dal sensore solare
- *ShutterPos* per avere memoria della posizione della tapparella
- *stringDisplay* è la stringa che dovrà essere mostrata su display OLED
- *oldString* è la stringa tampone che evita di scrivere le stesse informazioni su display OLED
- *Chiamata showDataDisplay* consente di mostrare le informazioni su display OLED (in questo caso la stringa "Welcome to Smart Home Controller!")

Per quanto riguarda l'acquisizione dei dati Bluetooth utilizziamo il blocco "*Bluetooth - quando si ricevono dati*" (**Fig. 13**) che verifica opportunamente quando l'utente decide di assumere il controllo del sistema di illuminazione mediante l'applicazione Android.

Per verificare quando il dispositivo è connesso utilizziamo i blocchi "*Bluetooth – quando si connette*" e "*Bluetooth - quando si disconnette*" mostrando su display OLED l'informazione relativa sullo stato della connessione e tramite la variabile *bleConnected* andiamo a valutare quando il dispositivo è realmente connesso o disconnesso.



Fig. 14
Funzioni per gestione LED.

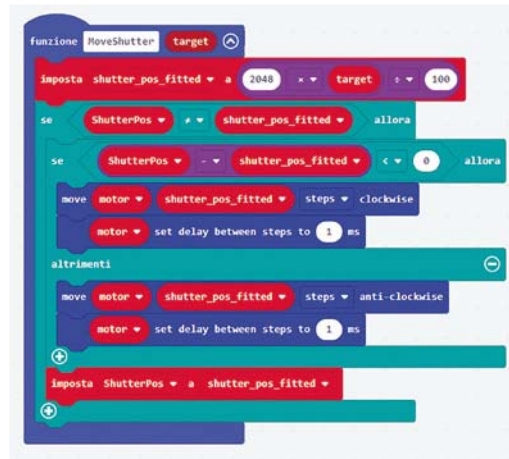


Fig. 15
Funzioni per gestione stepper.

Nel momento in cui riceviamo nuovi dati dall'applicazione risulta necessario interpretarli e per semplicità abbiamo deciso di utilizzare formato "#LxFySzz;" descritto in precedenza.

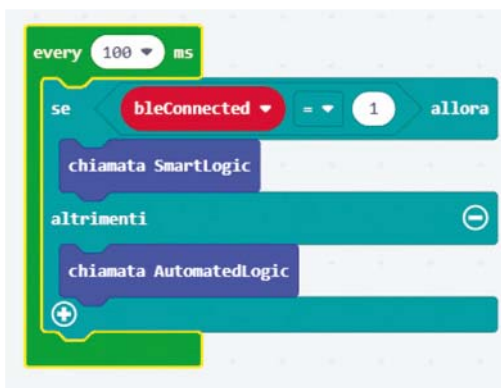
Il compito del codice in esame sarà quello di estrapolare il dato o meglio i caratteri compresi tra le lettere 'L', 'F' e 'S' tramite il quale l'utente deciderà mediante l'applicazione se accendere o spegnere il LED.

A tale scopo creiamo le variabili *dataBleLed*, *dataBleFan*, *dataBleShutter* ed acquisiamo il dato in base alla posizione dei rispettivi caratteri 'L', 'F' e 'S', memorizzandoli nelle variabili *ledIndex*, *fanIndex*



Fig. 16
Funzioni per gestione ventola.

Fig. 17
Funzione "every 100ms".



e shutterIndex.
Prima di addentrarci nell'implementazione delle logiche definiamo le funzioni, come di seguito:

- *turnOnLed – turnOffLed*: verranno utilizzate per accendere/spegnere il led tramite un

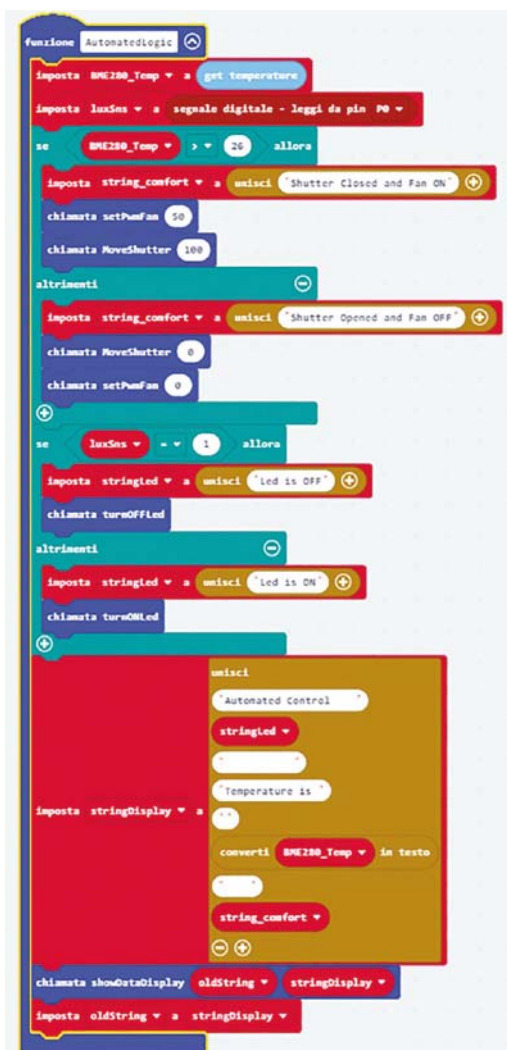


Fig. 18
Funzione per logica automatica.

segnale PWM (**Fig. 14**); per creare questo segnale utilizziamo il blocco "segnale analogico - imposta periodo su P1 a 1000 us", attraverso il quale definiamo un segnale con periodo di un millisecondo e quindi frequenza di 1kHz; mentre attraverso il blocco "segnale analogico-scrivi su P1 con 512" andiamo a definire il duty cycle con fondo scala di 1024, pertanto decidiamo di utilizzare un duty cycle del 50% (la funzione turn off display utilizza la stessa libreria con la differenza che il duty cycle è impostato allo 0%).

- *MoveShutter*: consente di movimentare il motore e quindi chiudere o aprire la tapparella tramite il target definito (**Fig. 15**). La variabile "ShutterPos" è usata per tenere memoria della posizione attuale e valutare se un nuovo target è richiesto.
- *setPwmFan*: utile per il pilotaggio della ventola consente di regolare la velocità tramite il duty cycle definito automaticamente oppure dall'utente (**Fig. 16**). Verrà utilizzato il blocco "segnale analogico - imposta periodo su P1 a 1000 us", attraverso il quale definiamo un segnale con periodo di un millisecondo e quindi frequenza di 1kHz; mentre attraverso il blocco "segnale analogico- scrivi su P8 con duty_fitted" andiamo a definire il duty-cycle.

In base quest'ultimo, il motore verrà azionato per ottenere il movimento orario (100% - tapparella chiusa) oppure antiorario (0% - tapparella aperta) attraverso le funzioni messe a disposizione dalla libreria del motore stepper per micro:bit. Una volta che abbiamo verificato ed acquisito il dato, andiamo ad aggiungere la funzione "every 100ms" attraverso la quale definiamo le logiche illustrate in **Fig. 17**.

Tramite il blocco *if-else* andiamo a verificare se il dispositivo è connesso via bluetooth e quindi attuare il dato ricevuto, oppure implementare la logica automatica in base i dati acquisiti dai sensori.

Per semplicità di lettura codice, verranno definite le funzioni chiamata *SmartLogic* e chiamata *AutomatedLogic* illustrate nelle **Fig. 18** e **Fig. 19**. Nel caso di dispositivo connesso via Bluetooth verranno applicate le richieste utente tramite le funzioni descritte in precedenza.

Per la funzione *MoveShutter* applichiamo direttamente il dato acquisito *dataBleShutter* convertito in numero e medesime considerazioni vengono applicate anche per la funzione

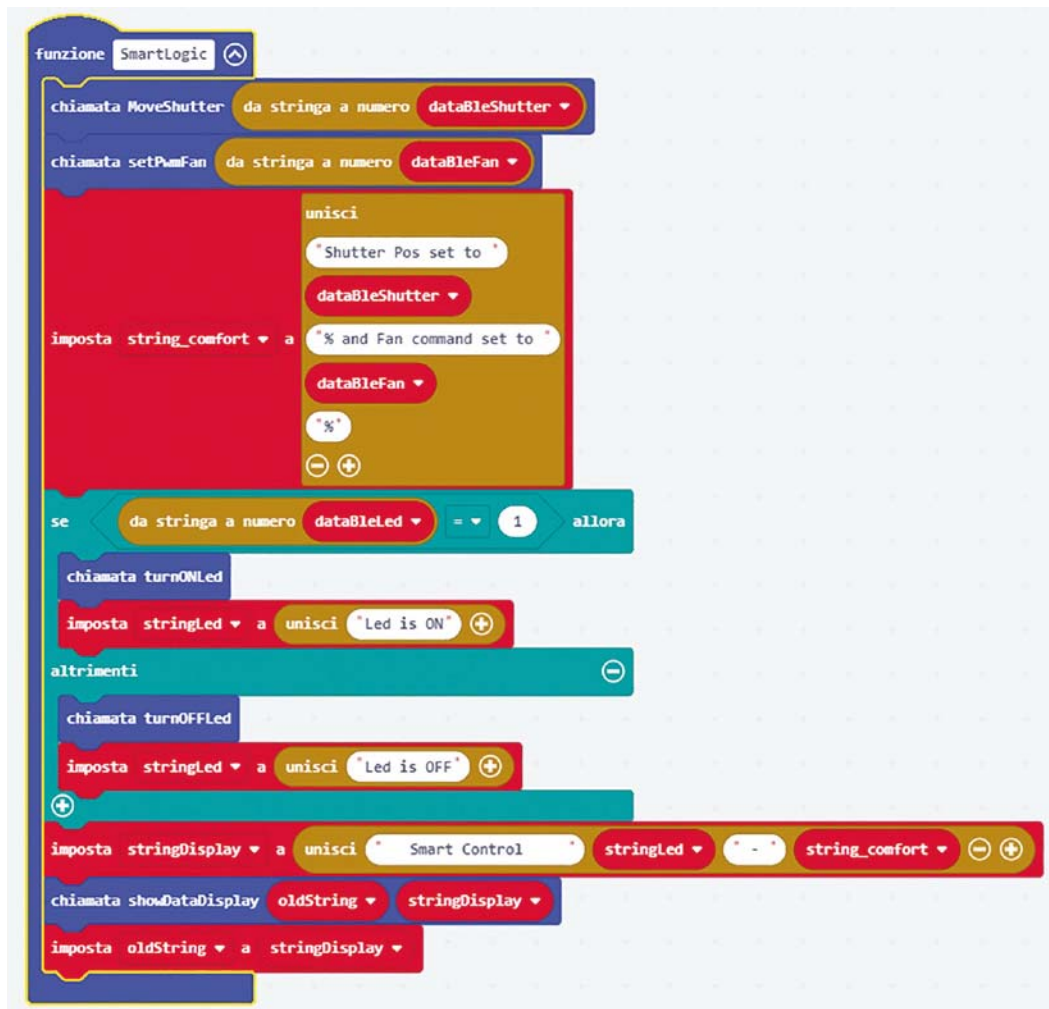
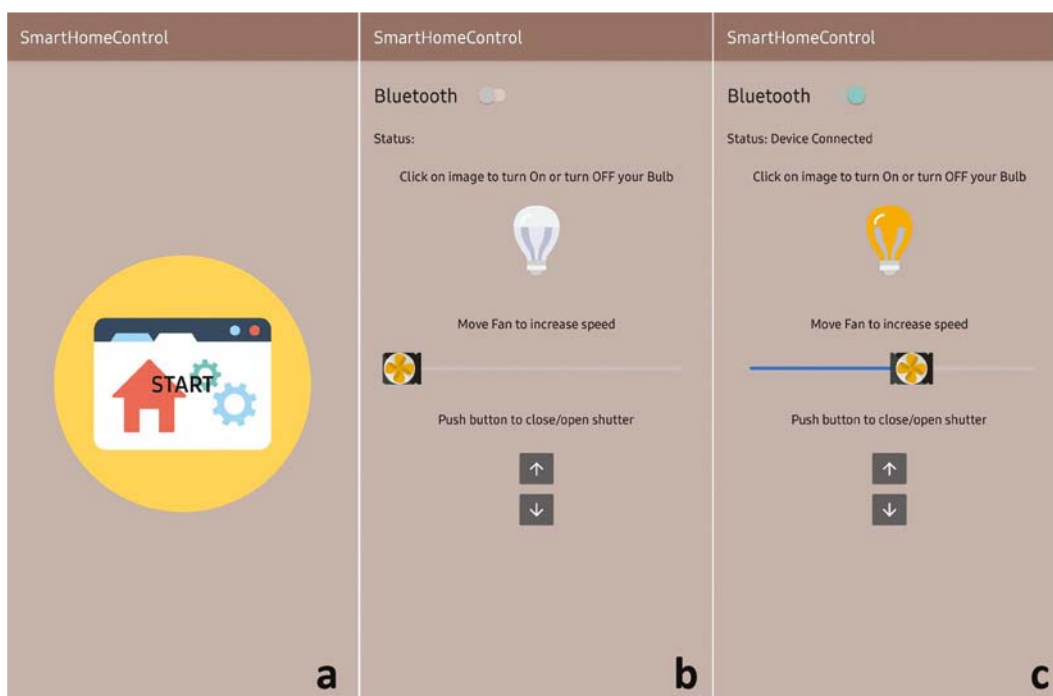


Fig. 19
Logiche
smart.

setPwmFan con la stringa *dataBleFan* con conseguente aggiornamento della stringa *string_comfort* che verrà mostrata su display. Il dato relativo all'illuminazione verrà analizzato e se quest'ultimo risulta essere uguale a 1 allora verrà acceso il led attraverso la funzione *turnONLed* ed inizializzata la stringa *stringLed*, così da utilizzarla per mostrare le informazioni sul display. Al contrario se il dato è uguale a zero allora il led dovrà essere spento tramite la funzione *turnOFFLed* con conseguente aggiornamento della stringa da mostrare sul display. Infine utilizziamo la funzione *showDataDisplay* per comunicare all'utente lo stato del dispositivo, dell'illuminazione e dei dispositivi di comfort. Nel caso in cui il dispositivo non sia connesso via Bluetooth, le funzionalità verranno gestite automaticamente tramite il segnale acquisito dal sensore solare e del sensore di

temperatura; pertanto tramite la variabile *luxSns* verrà memorizzato il dato fornito dal sensore crepuscolare sul pin digitale P0 mentre con la variabile *BME280_temp* verrà acquisita l'informazione sulla temperatura (mediante I²C-Bus con *get temperature*). La variabile *luxSns* consente di valutare se nell'area circostante è presente luce con conseguente spegnimento del led tramite la funzione *turnOFFLed* oppure accensione del LED tramite la funzione *turnONLed*. *BME280_Temp* verrà utilizzata per valutare se il limite di temperatura (in questo caso 26°C) è stato superato, con conseguente accensione della ventola e chiusura della tapparella oppure decidere di non intervenire con lo spegnimento del fan e l'apertura della tapparella. Infine la funzione *showDataDisplay* consente di comunicare all'utente che il sistema non è connesso via Bluetooth ma sarà gestito automaticamente.

Fig. 20
Applicazione
Android.



APPLICAZIONE ANDROID

In questo paragrafo verrà descritta l'applicazione Android che consentirà all'utente di gestire a suo piacimento i componenti presenti nel sistema tramite semplici azioni. L'applicazione si presenta con la schermata (anzi, le schermate) mostrata in **Fig. 20**, dove attraverso un semplice tap-in (pressione a schermo) è possibile raggiungere la pagina di gestione componenti, quest'ultima proposta nella **Fig. 20b**. Da quest'ultima pagina è possibile abilitare la connessione Bluetooth tramite un opportuno switch (cursore a video) che nella sua apparente semplicità consente di cercare ed accoppiarsi con il dispositivo micro:bit, mostrando lo stato della connessione tramite l'etichetta "Status:".

L'etichetta "Status: Device Connected", evidenzia come l'applicazione abbia trovato e stabilito una comunicazione con il dispositivo e come l'utente possa interagire con gli oggetti disponibili tramite le informazioni fornite. La pressione (tap-in) sull'immagine della "lampadina" oppure muovere l'icona della ventolina sulla barra di scorrimento (slide bar) ed infine la pressione dei tasti, sono azioni che consentono di aggiornare la stringa Bluetooth che ciclicamente viene fornita alla nostra scheda micro:bit.

La semplice pressione della lampadina consentirà l'accensione o spegnimento dell'illuminazione

mentre il movimento dell'icona della ventola consente di incrementare o ridurre la velocità con una risoluzione del 1% ed infine la pressione tasti permette la movimentazione della tapparella sempre con risoluzione 1%.

CONCLUSIONI

Arriviamo così al termine della descrizione del progetto di espansione della gestione domotica, con il quale abbiamo visto come si possa facilmente aggiungere componenti di "comfort" al sistema micro:bit.

Tuttavia il nostro viaggio non termina qui, perché nei prossimi fascicoli di Elettronica In vi spiegheremo come sia possibile espandere ulteriormente questo progetto, implementando nuovi scenari di controllo domotico. □



Cosa occorre?

La board micro:bit V2 è disponibile con lo starterkit (cod. Micro:bitQSKIT) a Euro 45,00.
I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

ARDUINO

SERIALCMD

di GUGLIELMO BRACUGLIA



Comandiamo le schede Arduino via seriale grazie ad una potente e completa libreria.

M

olto spesso, gli utenti di Arduino devono affrontare il problema di inviare, via seriale (normalmente usando il monitor seriale o altro emulatore) dei comandi ad una scheda per eseguire alcune funzioni come posizionare un servo ad un

certo angolo, accendere o spegnere dei LED, attivare o disattivare dei relé, inviare informazioni su un display, ecc. La gestione della porta seriale, su Arduino, richiede una certa esperienza e, specie per quelli che iniziano, si presenta come un bell'ostacolo; se a questo si aggiunge anche la problematica di interpretare ciò che si riceve, separando un comando dai suoi eventuali parametri e, di conseguenza, eseguire delle operazioni, ci si rende conto che la difficoltà cresce esponenzialmente e chi è alle prime armi, spesso si arrende o cerca il classico "copia/incolla", a volte incom-

prensibile per lui, che faccia, almeno in parte, ciò che deve fare, spesso con ... pessimi risultati. Per questo motivo abbiamo voluto realizzare un'apposita libreria che supera questo ostacolo. La libreria in questione l'abbiamo chiamata **SerialCmd** (installabile dal "Gestore di Librerie" di Arduino) e risolve questo problema (e, come vedremo, anche altri più complessi) permettendo di definire, in modo molto semplice, da dove arrivano i comandi, quali comandi riconoscere e, una volta riconosciuti, cosa fare per ciascuno di loro. A corredo della libreria ci sono vari esempi che mostrano come utilizzarla per eseguire comandi in arrivo da seriale (sia fisica che software), letti da un file di testo su SD e inviati da programma (ovvero è l'applicazione stessa che invia un comando, magari ricevuto via WiFi). Oltre che tramite l'IDE di Arduino, è possibile scaricare la suddetta libreria, e la documentazione che l'accompagna, da GitHub, puntando dal browser all'indirizzo <https://github.com/gpb01/SerialCmd>.

Le origini di questa libreria vanno ricercate nel lavoro svolto, prima da Steven Cogswell nel 2011 e poi da Stefan Rado nel 2012. A partire dal 2013, Guglielmo Braguglia ha ripreso detti lavori ed ha continuato sino ad oggi a modificarla ed ad implementare nuove funzionalità.

La libreria è personalizzabile ed adattabile alle proprie esigenze, così da minimizzare l'impatto sulla memoria da essa utilizzata. Purtroppo, a causa di una limitazione del funzionamento del IDE di Arduino (non è possibile predeterminare l'ordine di compilazione del codice e quindi NON è possibile cambiare, da programma .ino, i valori delle #define che vengono utilizzate dalle librerie), la personalizzazione va effettuata direttamente sui sorgenti della libreria. Scaricata quindi la libreria, ed installata nella apposita cartella delle librerie dell'utente (normalmente la cartella "libraries" che si trova nella cartella di "sketchbook" dell'utente), all'interno della cartella "SerialCmd" sottocartella "src" si troveranno i sorgenti della libreria. Il file interessato dalle personalizzazioni è quello di nome "SerialCmd.h"; esso andrà aperto con un editor di testo (quindi NON con un editor che introduce caratteri di formattazione) e, a seconda delle proprie esigenze, si potranno modificare le seguenti righe che si trovano nella parte iniziale del suddetto file:

```
#define SERIALCMD_MAXCMDNUM 8
// Max Number of Command
```

```
#define SERIALCMD_MAXCMDLNG 6
// Max Command Length
#define SERIALCMD_MAXBUFFER 30
// Max Buffer Length
```

Il significato delle tre righe è quello di seguito descritto.

- SERIALCMD_MAXCMDNUM: indica il numero **massimo** di "comandi" che la libreria potrà riconoscere. Ogni comando viene memorizzato in un array, assieme ad altre informazioni e, il numero di elementi di detto array, è gestito tramite tale voce.
- SERIALCMD_MAXCMDLEN: indica il numero massimo di "caratteri" di cui è composto, **al massimo**, un "comando". Anche questa voce viene utilizzata per ridurre al minimo l'occupazione di memoria della libreria, allocando solo lo spazio necessario.
- SERIALCMD_MAXBUFFER: indica il numero massimo di "caratteri" di cui è composto, **al massimo**, il buffer di ricezione dei comandi, inclusi tutti i parametri che si vogliono passare. Andrà dimensionato in funzione del tipo e del numero di parametri che seguiranno il comando, nella riga comando.

Ovviamente, per minimizzare al massimo ed ottimizzare l'occupazione di memoria (*nelle MCU dove questo ha realmente senso*), si dovranno dimensionare i suddetti parametri per lo stretto necessario in funzione delle proprie esigenze. Un ulteriore parametro che è personalizzabile, ma che non è legato all'occupazione di memoria, è il seguente:

```
#define SERIALCMD_FORCEUC 0
// If set to 1 force uppercase for serial command.
```

- SERIALCMD_FORCEUC: se lasciato a "0", non interviene sul "comando" ricevuto, se messo ad "1" forza la conversione, del solo "comando", esclusi quindi i parametri, da minuscolo a maiuscolo. Occorre tenere presente che tale cosa è vera, solo per i comandi ricevuti dallo *streaming* di dati (*seriale hardware, seriale software, ecc.*) e non per quelli richiamati direttamente dall'applicazione.

Questi sono i **solli** parametri di configurazione che debbono essere personalizzati mettendo le mani nel sorgente della libreria; vedremo invece

in seguito, altre personalizzazioni, che è possibile avere dinamicamente da programma applicativo. Analizziamo quindi come fare per utilizzare la libreria in un nostro programma Arduino. La prima cosa da fare, in testa al programma, assieme alle altre eventuali `#include`, sarà quello di aggiungere la seguente riga:

```
#include <SerialCmd.h>
```

In questo modo la libreria sarà inclusa tra quelle utilizzate dal nostro programma. Occorrerà poi indicare da quale Stream arriveranno i dati; tale Stream potrà essere una porta seriale fisica, un porta seriale virtuale (ad esempio creata con la libreria `SoftwareSerial`) o un flusso di dati proveniente, ad esempio, dalla lettura di una SD-Card. Dovremo quindi creare un'istanza della classe `SerialCmd`.

In tale fase, oltre ad indicare lo Stream di provenienza dei caratteri, potremo, opzionalmente indicare ulteriori due informazioni: quale carattere indica che la riga del comando è terminata e che quindi può essere elaborata (di base il carattere che la libreria si attende è il "carriage return", ovvero il CR o 0x0D) e quale carattere separa gli eventuali parametri tra di loro e dal comando stesso (di base il carattere che la libreria si aspetta è la virgola ovvero ", ").

Per facilitare la programmazione ed evitare errori, la libreria predefinisce delle costanti; per il termi-

natore di riga si può utilizzare una delle seguenti:

- `SERIALCMD_CR`: è il carattere "carriage return", normalmente abbreviato in CR ed equivalente al carattere esadecimale 0x0D;
- `SERIALCMD_LF`: è il carattere "line feed", normalmente abbreviato in LF ed equivalente al carattere esadecimale 0x0A;
- `SERIALCMD_NULL`: è il carattere "null", normalmente indicato come NULL ed equivalente al carattere esadecimale 0x00.

Notate che per il carattere che separa i vari parametri tra di loro ed essi dal comando, si può utilizzare una delle seguenti espressioni:

- `SERIALCMD_COMMA`: è il carattere "virgola" ovvero i parametri vanno separati tra di loro e dal "comando" con ',' (virgola);
- `SERIALCMD_SEMICOL`: è il carattere "punto e virgola" ovvero i parametri vanno separati tra di loro e dal "comando" con ';' (punto e virgola);
- `SERIALCMD_DOT`: è il carattere "punto" ovvero i parametri vanno separati tra di loro e dal "comando" con '.' (punto);
- `SERIALCMD_SPACE`: è il carattere "spazio" ovvero i parametri vanno separati tra di loro e dal "comando" con ' ' (spazio).

Quindi, ad esempio, per istanziare la nostra classe, prima del `setup()` e del `loop()`, indicando che i caratteri arriveranno dalla "Serial" (ovvero dalla seriale fisica collegata alla USB della nostra

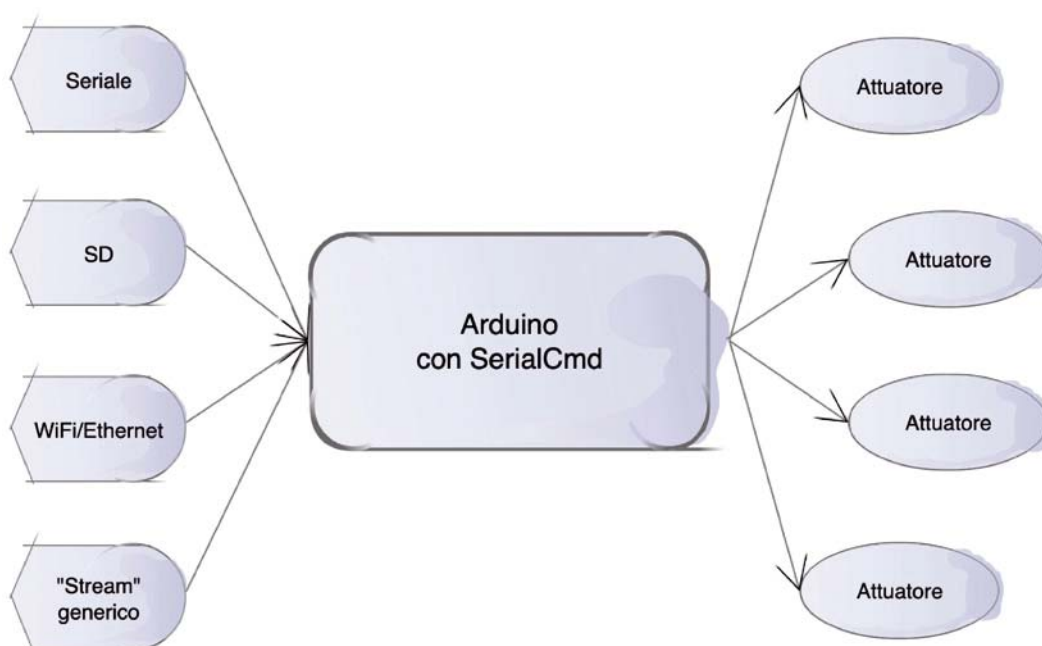


Fig. 1
La libreria `SerialCmd` permette di ricevere comandi da diversi input e può controllare più attuatori.

scheda Arduino) e creando un "istanza" della classe di nome "mySerCmd", inseriremo una riga simile a questa:

```
SerialCmd mySerCmd( Serial );
```

Volendo, in aggiunta, specificare che la nostra riga di comando non si concluderà con il CR (*che è il valore che, di base, la libreria si aspetta*), ma con il carattere LF, dovremo semplicemente passare un parametro in più:

```
SerialCmd mySerCmd( Serial, SERIALCMD_LF );
```

La stessa cosa dovremmo fare se volessimo cambiare il carattere separatore. Una volta istanziata la classe, la libreria ci mette a disposizione una serie di "metodi" sia per configurare i vari comandi che essa dovrà riconoscere, sia per poter passare dei comandi direttamente da applicativo, sia per poter verificare se è arrivato un comando e quindi eseguirlo. Cominciamo con il chiarire che un singolo comando è composto da una serie di caratteri terminati con il carattere terminatore che abbiamo scelto. Ad esempio, per accendere e spegnere il LED che è su ogni scheda Arduino, potremmo immaginare di creare due comandi:

```
LEDON  
LEDOF
```

Il primo di essi richiamerà una nostra funzione in cui accenderemo il LED ed il secondo, un'altra nostra funzione, in cui spegneremo il LED. Questa frase ci fa capire che, associato ad ogni "comando" che definiremo, dovremo scrivere una "funzione" che esegue il comando.

Ad esempio, per i due comandi che abbiamo appena definito, dovremo scrivere due funzioni, che potremo chiamare `f_ledOn` ed `f_ledOf` che andranno ad accendere e spegnere il LED sulla scheda:

```
...  
( void ) f_ledOn ( void ) {  
    digitalWrite( LED_BUILTIN, HIGH );  
}  
( void ) f_ledOf ( void ) {  
    digitalWrite( LED_BUILTIN, LOW );  
}  
...
```

Per associare un comando ad una nostra funzione, useremo il metodo della `SerialCmd` "AddCmd()":

```
uint8_t AddCmd ( const char *command,  
    char allowedSource, void ( *function ) () );
```

Tale metodo richiede tre parametri, prima di tutto una stringa costante che rappresenta il comando, poi un parametro per indicare da dove esso può arrivare ed, infine, quale funzione il comando deve chiamare.

Il secondo parametro può assumere solo tre valori che, per facilitare la programmazione, sono predefiniti:

- `SERIALCMD_FROMSERIAL`: indica che il comando è valido solo se arriva dallo *Stream* seriale e non è valido se arriva dalla funzione che permette di richiamare dei comandi da programma;
- `SERIALCMD_FROMSTRING`: indica che il comando è valido solo se arriva dalla funzione che permette di richiamare comandi da programma e non è valido se arriva dallo *Stream* seriale;
- `SERIALCMD_FROMALL`: indica che il comando è valido indipendentemente da dove arriva.

Quindi, proseguendo con il nostro esempio di accensione e spegnimento del LED, nel `setup()` chiameremo due volte il metodo `AddCmd()` in questo modo:

```
...  
mySerCmd.AddCmd("LEDON", SERIALCMD_FROMALL, f_ledOn);  
mySerCmd.AddCmd("LEDOF", SERIALCMD_FROMALL, f_ledOf);  
...
```

Questo "dice" alla libreria di aggiungere due comandi, di permetterne l'esecuzione sia che essi arrivino dallo *Stream* seriale, sia che provengano da comando da programma; indica inoltre di associare "LEDON" alla funzione che abbiamo chiamato `f_ledOn()` e "LEDOF" alla funzione che abbiamo chiamato `f_ledOf()`. Notate che i nomi delle funzioni associate vanno passati **senza** le parentesi.

Il metodo `AddCmd()` ritorna un `uint8_t` che può essere verificato per sapere se la chiamata ha avuto successo (valore ritornato 1/true) o non ha avuto successo (valore ritornato 0/false), come nel caso si cerchi di aggiungere più



comandi di quanti previsti nel valore costante `SERIALCMD_MAXCMDNUM` che abbiamo definito all'inizio.

Il metodo `AddCmd()` può inoltre essere chiamato più volte per lo stesso comando; questo permette sia di cambiare la funzione associata durante l'esecuzione del programma, sia di disabilitarla passando come funzione il valore `NULL`.

Sarà sempre possibile riabilitarla in seguito, chiamando di nuovo il metodo `AddCmd()` con il corretto nome di funzione da eseguire.

A questo punto, nella funzione `loop()` del nostro programma, non dovremo fare altro che richiamare in continuazione il metodo `"ReadSer()"` con:

```
int8_t ReadSer( );
```

Tale metodo non richiede alcun parametro e ritorna un `int8_t` che, volendo, può essere verificato per controllare se non ha ancora ricevuto alcun comando (non ha ancora incontrato il carattere terminatore), caso in cui ritorna il valore `-1`, se ha ricevuto un comando valido ed ha richiamato la funzione associata, caso in cui ritorna `1`, o se ha ricevuto un comando non riconosciuto e quindi non ha eseguito nulla, caso in cui ritorna `0`. Come spiegato, tale metodo va richiamato in continuazione (quindi può essere, ad esempio, messo come prima, o come ultima, istruzione nella nostra funzione `loop()`).

Esso ha il compito di andare a leggere lo *Stream* seriale che abbiamo definito nel momento dell'istanza della nostra classe, recuperare i caratteri, individuare il carattere terminatore, separare il comando ed infine, se trovato un comando conosciuto, mettere in esecuzione la funzione associata.

Un semplice estratto di codice che spiega come utilizzare il metodo è il seguente:

```
void loop() {
    int8_t ret;
    // ...
    // qui vanno altre istruzioni
    // che vanno eseguite nel loop()
    // ...
    ret = mySerCmd.ReadSer();
    if ( ret == 0 )
        mySerCmd.Print ( ( char * ) "ERRORE\r\n" );
}
```

In questo piccolo estratto abbiamo utilizzato un altro metodo della classe `SerialCmd`, metodo che

non abbiamo ancora descritto, il metodo `"Print()"`. Tale metodo permette di inviare, allo *Stream* seriale che abbiamo definito nel momento dell'istanza della classe, sia stringhe di caratteri, sia valori numerici, un po' come si fa con il metodo `"print()"` della classe `Serial` di Arduino.

Abbiamo detto che oltre ai comandi è possibile passare anche dei parametri, infatti, ad esempio, potremmo non solo voler accendere e spegnere un LED, ma potremmo volerne regolare la luminosità (tramite la funzione di Arduino `analogWrite()`) e quindi dobbiamo avere il modo di indicare alla funzione che si occuperà della cosa che valore di luminosità utilizzare.

Immaginiamo di modificare la nostra funzione, che abbiamo visto sopra, relativa al comando `LEDON` così da poter anche leggere con che intensità accendere il LED.

Il comando che dovremo inviare dovrà essere composto dal "comando" vero e proprio seguito dal carattere separatore dei parametri e dai necessari parametri (nel nostro esempio uno solo, la luminosità).

Quindi, volendo ad esempio impostare la luminosità a `128`, da tastiera dovremo digitare (dando per scontato che il carattere separatore è quello predefinito, ovvero `;`) il comando:

LEDON, 128

... e dovremo modificare la nostra funzione `f_ledOn()` per accettare tale nuovo parametro. La lettura dei parametri si fa tramite un altro metodo della classe `SerialCmd`, il metodo `"ReadNext()"`:

```
char* ReadNext( );
```

Tale metodo ritorna un `"char *"`, ovvero un puntatore ad un array di caratteri (*terminato da `0x00`*), quindi una stringa classica del 'C', che contiene il solo parametro.

Nel caso di più parametri, ogni chiamata a tale metodo ritorna il parametro successivo sino a quando, alla fine, non essendoci più parametri, ritorna `NULL`.

Nel nostro esempio, chiamato una sola volta il metodo (*abbiamo un solo parametro*), esso ci ritornerà la stringa `"128"` che sarà nostra cura convertire in valore numerico prima di chiamare la `analogWrite()`. Per fare questo useremo la funzione standard del 'C' `"atoi()"` che converte un valore numerico intero, espresso da una stringa di caratteri, nel numero intero equivalente.



Per cui avremo:

```
...
( void ) f_ledOn ( void ) {
    char * cPar;
    uint8_t intensita;
    //
    cPar = mySerCmd.ReadNext( );
    if ( cPar == NULL ) {
        mySerCmd.Print ( ( char * )
            "ERRORE: Parametro omissso. \r\n" );
        return;
    }
    intensita = atoi ( cPar );
    analogWrite( LED_BUILTIN, intensita );
}
...
```

In quest'esempio si è fatto solo un semplice controllo per verificare che l'utente abbia inserito almeno un valore.

Nella realtà è bene sempre effettuare controlli più accurati sui vari parametri, ad esempio, oltre a verificarne l'esistenza, verificare che rientrino in intervalli di valore ammessi per ciò che si deve fare.

Come abbiamo già accennato, la libreria *SerialCmd* è inoltre in grado di ricevere i comandi, invece che dallo Stream seriale, direttamente da una riga del programma.

Se questa cosa può sembrare inutile (difatti, se sono sotto il controllo del programma, potrei fare direttamente quello che il comando fa fare alla funzione), in realtà questa cosa è estremamente utile e permette di ricevere il comando ed i parametri da qualche cosa di diverso che uno Stream seriale e poi passare il tutto alla libreria per la verifica e l'esecuzione del comando ricevuto.

Proprio a dimostrazione di questa cosa, a corredo della libreria, vengono forniti un paio di esempi (*ESP_SerialCmd* e *CH376S_SerialCmd*); il primo dei due riceve i "comandi" e i parametri attraverso una HTTP GET tramite web browser e li passa poi alla libreria *SerialCmd*, il secondo va a leggere, riga per riga, i comandi da una SD-Card, non direttamente collegata ad Arduino e (letta tramite la libreria SD), ma gestita da un chip CH376S e li passa poi alla *SerialCmd*. Il metodo che permette di fare questo è "ReadString()":

```
uint8_t ReadString ( char * theCmd,
    uint8_t fValidate = false );
```

Tale metodo accetta due parametri; il primo è una stringa classica del 'C' che contiene il "comando" e gli eventuali suoi parametri, così come li avremmo scritti se li avessimo dovuti inviare da Stream seriale, il secondo permette solo di verificare se il comando che vogliamo eseguire è tra i comandi riconosciuti o meno, senza doverlo mettere in esecuzione (in pratica permette di effettuare una validazione a priori).

I suindicati esempi, a corredo della libreria, ne mostrano esaurientemente l'utilizzo, fateli vostri e sperimentate partendo da questi codici. Prima di concludere, occorre ancora indicare che, per risparmiare memoria sulle piccole MCU AVR, alcuni metodi della classe *SerialCmd* esistono anche in una versione che permette l'uso della nota "macro F()", macro che permette di memorizzare le stringhe costanti (come lo sono le varie stringhe che identificano i comandi) nella memoria flash delle MCU AVR classiche (Arduino UNO, Arduino Leonardo, Arduino MEGA, Arduino Nano, ecc.).

Le varianti ai metodi già illustrati che, se ci si trova su MCU AVR (e quindi a livello del pre-processore è definita la costante `__AVR__`) si hanno in più a disposizione, sono le seguenti:

```
uint8_t AddCmd( const __FlashStringHelper *command,
    char allowedSource, void ( *function )() );
```

```
uint8_t ReadString ( const __FlashStringHelper *
    theCmd, uint8_t fValidate = false )
```

Per ulteriori informazioni e tutti i dettagli sull'implementazione dei vari metodi, suggeriamo un'attenta lettura del "readme" che accompagna la libreria e lo studio dei vari esempi che vengono forniti assieme ad essa.

Un'ultima nota: quando si usa un terminale seriale, come può essere il "monitor seriale" dell'IDE di Arduino, ricordarsi sempre di impostare, coerentemente con ciò che l'istanza nel vostro programma della *SerialCmd* si aspetta, il carattere terminatore che, di base, è il CR (*carriage return*).

Bene, siamo arrivati alla fine di questo articolo, dove vi abbiamo illustrato questa completa e versatile libreria per Arduino.

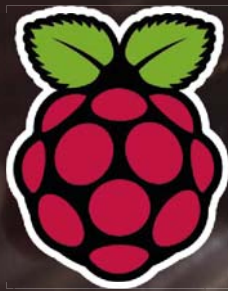
Non vi resta che sperimentare partendo dai numerosi esempi e controllare così la vostra scheda di prototipazione rapida in maniera semplice e veloce.

Buon lavoro.



GITEA IL VERSIONING
OPEN-SOURCE

di EMANUELE SIGNORETTA



Configuriamo una Raspberry Pi per farle ospitare un'istanza della versione self-hosted e open-source di Git, il popolare tool di versioning per software e firmware.

S

i dice, ma pare non sia solo una leggenda metropolitana degli sviluppatori di codice (i famigerati Developer o Dev, come si usa chiamarli oggi per contrapporgli agli Ops o Operations...), che il 95% del lavoro di chi scrive codice sia dedicato al debug e solo il 5% alla scrittura. Dai gruppi di sviluppo per le grandi imprese al maker che si avvicina per la prima volta al mondo del coding, c'è una costante: tenere traccia delle modifiche apportate al codice. In questo articolo parleremo delle necessità degli sviluppatori in termini di affidabilità, velocità e scalabilità e procederemo all'installazione di un'istanza dedicata al code versioning su una Raspberry Pi 4.

GIT

Git è il principale software che viene utilizzato per il code

versioning. Il “padre” di questo progetto è **Linus Torvalds**, l’inventore del kernel Linux. Lo sviluppo iniziò nell’aprile del 2005 quando non venne rinnovata la licenza di Bitkeeper agli sviluppatori del kernel. Basti pensare che per le primissime versioni del kernel, le varie modifiche venivano condivise utilizzando gli archivi compressi (tarball) inviati tramite e-mail a circa 250 colleghi. Il processo era molto macchinoso e molto lento, per non parlare delle eventuali modifiche apportate “al volo” da uno sviluppatore mentre magari riceveva una e-mail con le modifiche di un collega. Le alternative non erano molte: utilizzare **SVN**, **CVS** (detestati da Torvalds per la loro macchinosità e lentezza) oppure scrivere di suo pugno un nuovo software. Per questo motivo il 3 aprile del 2005 iniziò lo sviluppo di Git. Tra le caratteristiche principali di Git possiamo annoverare le elevate performance (circa 6 o 7 patch applicate al kernel Linux in un secondo), la possibilità di avere un flusso di lavoro non centralizzato, dunque distribuito e che proteggesse i sorgenti in caso di corruzione dei dati o di eliminazione degli stessi, sia essa accidentale o no. Il 16 giugno 2005 è stata rilasciata la prima versione del kernel Linux mediante Git. Tra le molte funzionalità abbiamo la possibilità di accedere ai repository utilizzando i protocolli http, FTP, ssh, rsync o il protocollo git stesso. I repository possono essere suddivisi in branches per poi essere riuniti con un merge, e ad ogni commit corrisponde un hash, codice univoco delle modifiche apportate. In più è possibile installare nuovi moduli, come ad esempio un’interfaccia web, un visualizzatore di documenti, ecc.

SOLUZIONI CLOSED SOURCE, OPEN CORE, OPEN-SOURCE E GITEA

Come per tutti i software, possiamo suddividere i fork di Git in diverse macroaree, fornendo alcuni esempi:

- Closed source: software che non possono essere scaricati o eseguiti sui propri server. **GitHub** è uno di questi, offre l’archiviazione di un numero infinito di repositories sui propri server, ma i sorgenti che “muovono” il tutto sono closed.
- Open Core: i software che possono essere scaricati ed eseguiti liberamente ma i cui sorgenti sono closed source. Ne è un esempio **GitLab**, che permette il download dei file binari o dei pacchetti per poterne eseguire l’installazione sui propri server.
- Open source: software scaricabili ed eseguibili liberamente, i cui sorgenti sono aperti alla

comunità per migliorarne lo sviluppo. Tra questi possiamo elencare Gogs, Gitarella, Gitweb, etc. Gitea è un **fork di Gogs** e ne eredita tutte le specifiche principali: è scritto in Go, è così lightweight da poter essere eseguito su una Raspberry Pi e ha un’interfaccia grafica web quasi del tutto simile a quella di GitHub. Supporta l’autenticazione con LDAP, openID, la gestione di utenti e organizzazioni, ecc. Alcune di queste funzionalità non sono disponibili gratuitamente in servizi come GitHub e perciò si presta molto all’utilizzo in ambienti lavorativi di piccole dimensioni o team studenteschi.

HARDWARE NECESSARIO

Per questo progetto sono necessari una Raspberry Pi 4 con una scheda SD da almeno 64 GB (o un SSD nel caso in cui desiderassimo avere più affidabilità), il relativo case, l’alimentatore e il dissipatore. Neessiteremo anche di un cavo di rete per collegare il Raspberry Pi al gateway di casa. È possibile acquistare il kit completo presso il sito www.futuranet.it (cod. 7300-OKIT8GPI4).

PREPARAZIONE DEL SISTEMA OPERATIVO

Il primo step di questa guida è preparare il sistema operativo per la Raspberry Pi.

Scarichiamo il tool Raspberry Pi Imager dal sito ufficiale della Raspberry Pi Foundation e una volta installato avviamolo. Apparirà una schermata come in **Fig. 1**; per prima cosa selezioniamo il sistema operativo cliccando nell’ordine le voci **Scegli S.O.** → **Raspberry Pi OS (other)** → **Raspberry Pi OS Full (32 bit)**.

Selezioniamo adesso la scheda SD (oppure un SSD, la procedura cambia leggermente) attraverso la voce **Scegli scheda SD**. Una volta fatto possiamo configurare un’eventuale connessione WiFi cliccando sulla ruota dentata in basso a sinistra e riempiendo accuratamente i campi SSID e Password. Abilitiamo il servizio SSH per poter usare la Raspberry Pi anche senza monitor durante la prima configurazione (**Fig. 2**). Una volta fatte le nostre scelte, avviamo il flashing del sistema operativo sulla scheda SD e quando terminato inseriamola



Fig. 1
Scelta del sistema operativo per Raspberry Pi.

nella Raspberry Pi. Avviamolo e dopo qualche minuto avviamo una scansione della LAN alla ricerca dell'indirizzo IP assegnato alla Raspberry Pi. Potete usare strumenti come Advanced IP Scanner oppure collegarvi all'interfaccia web del vostro gateway di casa.

Una volta trovato l'indirizzo IP (nel nostro caso 192.168.178.82) colleghiamoci in SSH (la porta di ascolto di default è la 22) e usiamo come username **pi** e come password **raspberrypi**. Quando collegati cambiamo la password dell'utente pi utilizzando il comando `sudo passwd pi`: inseriamo e confermiamo una nuova password quando richiesto.

Ciò fatto, sempre da terminale, diamo il comando `sudo raspi-config` e nella schermata che si aprirà scegliamo l'opzione **Display Options** → **VNC Resolution** → **1920x1080**. Una volta confermata questa opzione, abilitiamo il protocollo VNC (per poter usufruire dell'interfaccia grafica, ma senza utilizzare un monitor) selezionando l'opzione **Interface Options** → **VNC** → **Y**.

Usciamo dal tool e quando richiesto riavviamo la scheda Raspberry Pi.

È necessario assegnare al Raspberry un indirizzo IP statico e per farlo useremo il tool grafico presente nel Desktop Environment del sistema. Per farlo ci collegheremo allo schermo "virtuale" configurato precedentemente usando il protocollo **VNC**.

Sul PC apriamo dunque il programma VNC Viewer (Scaricabile al link www.realvnc.com/en/connect/download/viewer/) e inseriamo l'indirizzo IP della Raspberry Pi. Confermiamo le chiavi di sicurezza e una volta inseriti username e password, ci apparirà una schermata come in **Fig. 3**.

Spostiamo il cursore in alto a destra, sull'icona dell'interfaccia di rete e dopo aver fatto un click con il tasto destro del mouse selezioniamo **"Wired & wireless network setting..."**, selezioniamo l'interfaccia eth0 e inseriamo i parametri di rete come proposto nella **Fig. 4**, prestando attenzione alla subnet mask, diversa in base alla rete LAN di appartenenza. Una volta fatto ciò, salviamo le impostazioni ed usciamo da VNC, per tornare alla shell SSH.

Iniziamo con il creare un utente con determinati requisiti, lanciando il comando:

```
sudo adduser \
  --system \
  --shell /bin/bash \
  --gecos 'Git Version Control' \
```

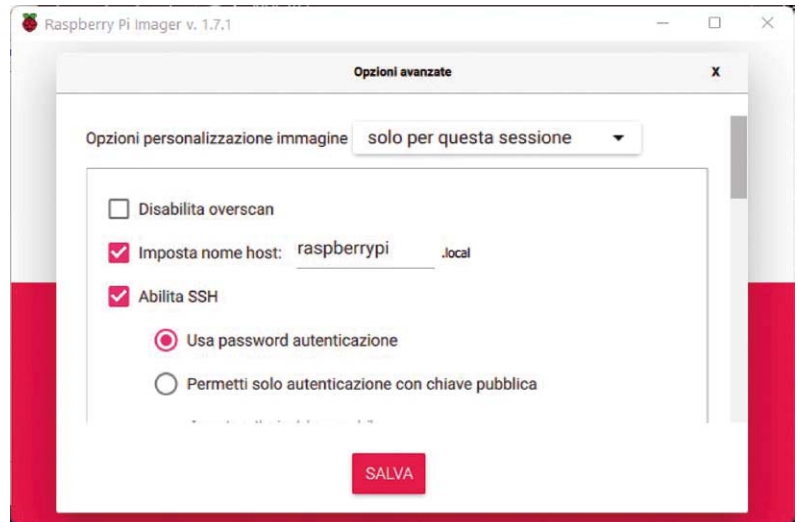


Fig. 2
Abilitazione
del servizio
SSH.



Fig. 3
Schermata di
welcome del
sistema.

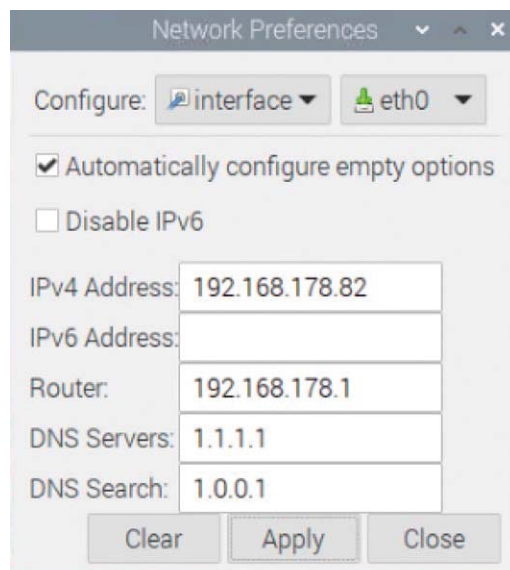


Fig. 4
Inserimento
dei parametri
di rete.

```
--group \
--disabled-password \
--home /home/git \
git
```

e creiamo l'apposita struttura per cartelle con i relativi permessi, con i comandi:

```
sudo mkdir -p /var/lib/gitea/{custom,data,log}
sudo chown -R git:git /var/lib/gitea/
sudo chmod -R 750 /var/lib/gitea/
sudo mkdir /etc/gitea
sudo chown root:git /etc/gitea
sudo chmod 770 /etc/gitea
```

Procediamo adesso all'apertura delle porte 80 e 443 del firewall per consentire connessioni dall'esterno, successivamente spiegheremo nel dettaglio perché. Procediamo con i comandi:

```
sudo ufw allow 80
sudo ufw allow 443
```

per aprire le porte e con i comandi *sudo ufw enable* e *sudo ufw reload* rispettivamente per abilitare il firewall e aggiornare le regole.

PORT FORWARDING E DDNS

Per poter accedere all'istanza di Gitea al di fuori

Fig. 5
Impostazioni di abilitazione del Port Forwarding della porta 443.

Creare abilitazione

☐ Abilitazione MyFRITZ!
☒ Abilitazione porte

Applicazione

Server HTTPS

Protocollo

TCP

Porta del dispositivo

443

fino a porta

443

Porta desiderata esternamente

443

☒ Attivare abilitazione

OK

Annulla

della rete domestica è necessario impostare il port forwarding e in base alle proprie disponibilità, usare un servizio di **DDNS** oppure un IP pubblico statico. Procediamo con ordine: per poter accedere al webserver utilizzeremo la porta 80 per il protocollo http e la 443 per il protocollo https.

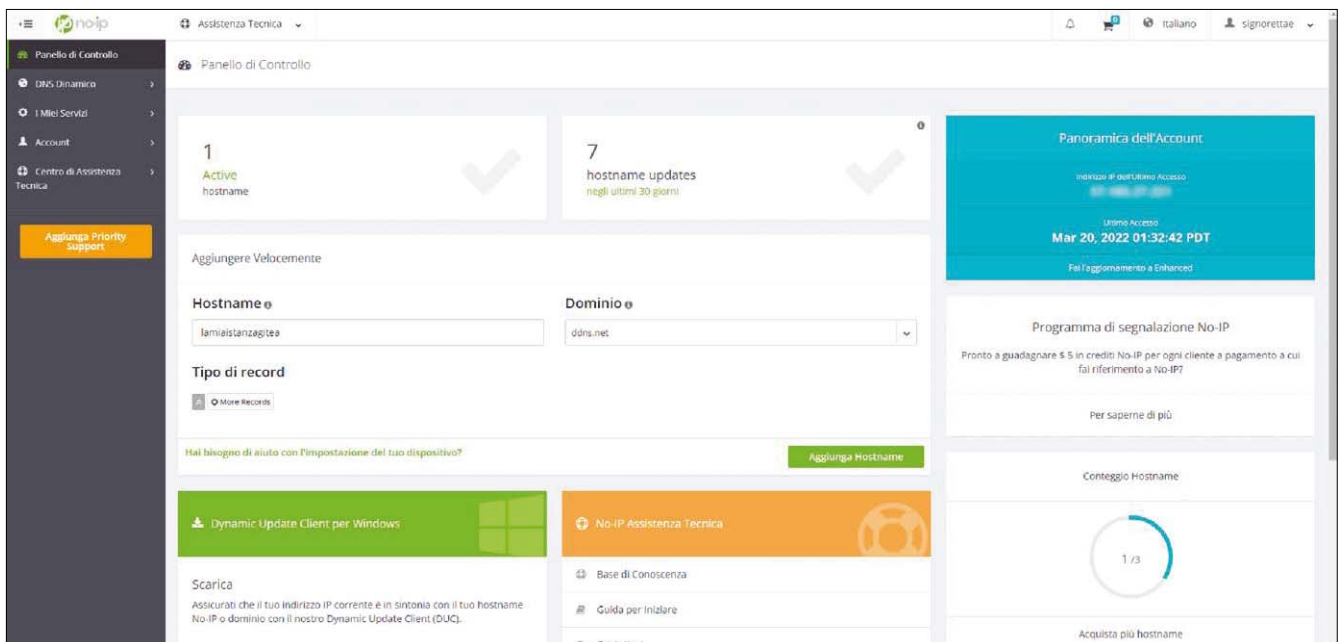
Ma quali sono le differenze? Https, a differenza di http trasmette i dati in formato criptato in modo tale da rendere impossibile lo sniffing dei pacchetti. Dobbiamo dunque creare una tabella di inoltro sul nostro gateway in modo tale che quando riceverà delle richieste per le porte 443 e 80: queste ultime verranno inoltrate all'indirizzo IP interno della Raspberry Pi (**Fig. 5**).

La procedura varia in base al modem a disposizione di ciascuno di noi, ma noi ci limiteremo a riportare come esempio quella di un Fritz!Box 7590. Rechiamoci sull'interfaccia web del router digitando l'indirizzo IP dello stesso sulla barra degli indirizzi (in questo caso è 192.168.178.1).

Logghiamoci e andiamo sull'opzione **"Internet"** → **"Abilitazioni"** → **"Abilitazione porte"** → **"Aggiungi dispositivo per abilitazioni"**. Selezioniamo adesso il dispositivo e una volta fatto ci compariranno l'indirizzo locale e l'indirizzo MAC. Clicchiamo su **"Nuova abilitazione"** e infine sul radio button **"Abilitazione porte"**. Selezioniamo **"Server HTTP"** e spuntiamo la voce **"Attivare abilitazione"**. Una volta premuto **"OK"** ripetiamo la stessa procedura per l'applicazione **"Server HTTPS"** e a questo punto la Raspberry Pi sarà accessibile dall'esterno sulle porte 80 e 443. Siccome alcuni ISP non forniscono un indirizzo IP pubblico statico, utilizzeremo un servizio di DDNS. Sostanzialmente andremo ad associare all'indirizzo IP pubblico un record del tipo *lamiaistanzagitea.ddns.net*.

Sarà il router di casa a comunicare l'IP inviandolo periodicamente ad un servizio che aggiornerà il record DNS (da qui il termine Dynamic DNS). In questo caso faremo ricorso a No-IP, il quale è un servizio gratuito di DDNS che richiede (solamente) il rinnovo del dominio mensilmente, cliccando su un link che ci verrà inviato in automatico sulla casella di posta. Per creare un dominio su No-IP rechiamoci al link no-ip.com e una volta terminata la registrazione rechiamoci sulla dashboard e creiamo un nuovo record, compilando i campi richiesti dalla pagina come proposto in **Fig. 6**.

Rechiamoci di nuovo sull'interfaccia web del Fritz!Box e andiamo sull'opzione **"Internet"** → **"Abilitazioni"** → **"DynDNS"**. Spuntiamo la voce **"Usare DynDNS"**, selezioniamo come provider **NO-IP** e compiliamo i campi vacanti: sono ne-



cessari nome del dominio, username e password dell'account. Il risultato finale deve essere come quello proposto nella **Fig. 7**. Una volta sicuri delle nostre scelte possiamo cliccare su **"OK"** e le impostazioni verranno applicate.

INSTALLAZIONE E CONFIGURAZIONE DI MARIADB

Prepariamo adesso il database su cui ospitare i dati di Gitea. Noi utilizzeremo **MariaDB**. Per proce-

dere con l'installazione, sempre da SSH impartiamo i seguenti comandi:

```
sudo apt-get update
sudo apt-get upgrade -y
sudo apt-get install mariadb-server pwgen -y
```

Una volta fatto ciò, lanciamo il comando:

```
pwgen -y 20
```

Fig. 6
Creazione di un dominio No-IP.

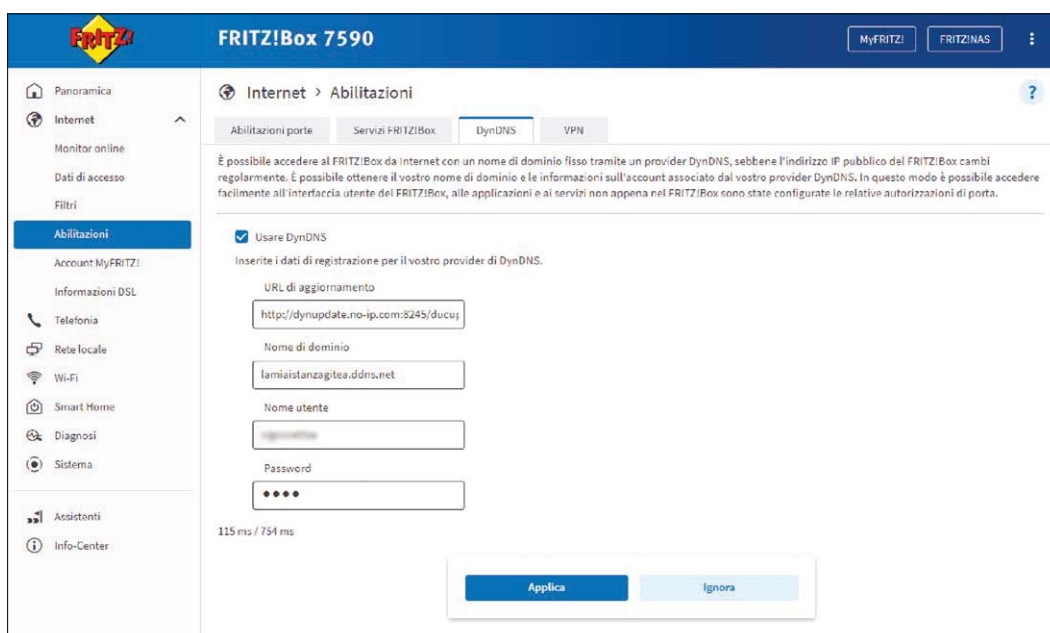


Fig. 7
Impostazione del dominio No-IP completata.

Questo comando genererà entropicamente una serie di password di 20 caratteri, scegliamone due e copiamole in documento di testo, ci serviranno più in avanti. Successivamente possiamo impartire il comando:

```
sudo mysql_secure_installation.
```

Prestiamo molta attenzione all'output del terminale perché ci verranno richieste diverse informazioni per procedere con la configurazione.

Nello specifico, alla domanda **"Set root password? [Y/n]"** rispondiamo con Y e quando richiesto inseriamo e confermiamo la prima delle due password generate precedentemente con pwgen.

Rispondiamo a tutte le richieste **"Disallow root**

login remotely? [Y/n]", **"Remove test database and access to it? [Y/n]"**, **"Reload privilege tables now? [Y/n]"** con Y e attendiamo il termine dell'installazione.

Per configurare il database è necessario loggarci sulla shell di MariaDB. Per fare ciò useremo la shell SSH: diamo il comando `sudo mysql -u root -p` e inseriamo la password dell'utente root creato precedentemente.

Una volta loggati creiamo l'utente che avrà accesso al database con il comando `CREATE USER 'gitea_user'@'localhost' IDENTIFIED BY "PASSWORD";`. Al posto di "PASSWORD" inseriamo la seconda password che abbiamo creato con pwgen.

Per creare il database usiamo il comando `CREATE`

Listato 1 - Configurazione di gitea mediante il file .ini

```
APP_NAME = Gitea: Git with a cup of tea
RUN_USER = git
RUN_MODE = prod

[server]
LOCAL_ROOT_URL      = http://localhost:3000/
SSH_DOMAIN          = localhost
DOMAIN              = lamiaistanzagitea.ddns.net
HTTP_PORT           = 3000
ROOT_URL            = https:// lamiaistanzagitea.ddns.net/
DISABLE_SSH         = true
SSH_PORT            = 22
LFS_START_SERVER    = true
LFS_CONTENT_PATH    = /var/lib/gitea/data/lfs
LFS_JWT_SECRET      =
OFFLINE_MODE        = false

[database]
DB_TYPE = mysql
HOST    = 127.0.0.1:3306
NAME    =
USER    =
LOG_SQL = false
PASSWD  =
SCHEMA  =
SSL_MODE = disable
CHARSET = utf8mb4
PATH    = /var/lib/gitea/data/gitea.db

[security]
INSTALL_LOCK      = false
SECRET_KEY        =
INTERNAL_TOKEN    =
PASSWORD_HASH_ALGO = pbkdf2
LOGIN_REMEMBER_DAYS = 7
COOKIE_USERNAME   = gitea_awesome
COOKIE_REMEMBER_NAME = gitea_incredible
REVERSE_PROXY_AUTHENTICATION_USER = X-WEBAUTH-USER
REVERSE_PROXY_LIMIT = 1
REVERSE_PROXY_TRUSTED_PROXIES = 127.0.0.0/8,::1/128, 192.168.178.82
MIN_PASSWORD_LENGTH = 6

[oauth2]
ENABLE = false
```

`DATABASE gitea_db CHARACTER SET 'utf8mb4'`
`COLLATE 'utf8mb4_unicode_ci';` e per garantire
all'utente "gitea_user" i permessi sul database
usiamo invece: `grant all privileges on gitea_db.* TO`
`'gitea_user'@'localhost';`
Per rendere effettive le modifiche diamo `FLUSH`
`PRIVILEGES;` e successivamente il comando `exit` per
uscire dalla shell di MariaDB.

DOWNLOAD E PREPARAZIONE DI GITEA

Per scaricare Gitea rechiamoci al link dl.gitea.io/gitea e selezioniamo l'ultima versione disponibile. Verremo reindirizzati ad una pagina contenenti i file binari di Gitea compilati per diversi sistemi operativi e per diverse architetture.

Selezioniamo il file binario per i sistemi Linux basati su architettura **arm6** e copiamone il link. Torniamo sulla shell e digitiamo il comando `sudo wget -O gitea` e copiamo il link copiato precedentemente.

Il risultato dovrebbe essere il comando `sudo wget -O gitea https://dl.gitea.io/gitea/1.16.7/gitea-1.16.7-linux-arm-6`. Una volta scaricato diamo il comando `sudo chmod +x gitea` per garantire i permessi di esecuzione. Copiamo adesso il file con il comando `sudo cp gitea /usr/local/bin/gitea`.

Creiamo adesso il file di configurazione con il comando `sudo nano /etc/gitea/app.ini` e copiamoci il contenuto del **Listato 1**.

Salviamo il tutto con **CTRL+X → T e INVIO**. Diamo ora il comando `sudo chown git:git /etc/gitea/app.ini`

↓ Listato 1 (continua)

```
[service]
REGISTER_EMAIL_CONFIRM      = false
ENABLE_NOTIFY_MAIL          = false
DISABLE_REGISTRATION        = false
ALLOW_ONLY_EXTERNAL_REGISTRATION = false
ENABLE_CAPTCHA               = false
REQUIRE_SIGNIN_VIEW        = false
DEFAULT_KEEP_EMAIL_PRIVATE  = false
DEFAULT_ALLOW_CREATE_ORGANIZATION = true
DEFAULT_ENABLE_TIMETRACKING  = true
NO_REPLY_ADDRESS             = noreply.localhost

[repository]
ROOT = /var/lib/gitea/data/gitea-repositories

[mailer]
ENABLED = false

[picture]
DISABLE_GRAVATAR            = false
ENABLE_FEDERATED_AVATAR    = false

[openid]
ENABLE_OPENID_SIGNIN       = false
ENABLE_OPENID_SIGNUP       = false

[session]
PROVIDER = file

[log]
ROOT_PATH      = /var/lib/gitea/log
MODE           = file

; ; Either "Trace", "Debug", "Info", "Warn", "Error", "Critical", default is "Trace"
LEVEL          = Trace
ROUTER         = file
ROUTER_LOG_LEVEL = Info
ENABLE_ACCESS_LOG = true
ACCESS         = file
ACCESS_LOG_TEMPLATE = {{.Ctx.RemoteAddr}} - {{.Identity}} [{{.Start.Format "[02/Jan/2006:15:04:05 -0700]"}]}
                {{.Ctx.Req.Method}} {>
REDIRECT_MACARON_LOG = true
MACARON             = file
```

❏ Listato 2 - Configurazione del servizio con Systemd

```
[Unit]
Description=Gitea (Git with a cup of tea)
After=syslog.target
After=network.target

Wants=mariadb.service
After=mariadb.service

[Service]

RestartSec=2s
Type=simple
User=git
Group=git
WorkingDirectory=/var/lib/gitea/

ExecStart=/usr/local/bin/gitea web --config /etc/gitea/app.ini
Restart=always
Environment=USER=git HOME=/home/git GITEA_WORK_DIR=/var/lib/gitea

[Install]
WantedBy=multi-user.target
```

per impostare l'utente git come proprietario del file e `sudo chmod 770 /etc/gitea/app.ini` per assegnare gli opportuni permessi.

AVVIO AUTOMATICO CON SYSTEMD

Per far avviare Gitea all'avvio della Raspberry Pi è necessario creare un demone con Systemd.

Per fare ciò digitiamo il comando `sudo nano /etc/systemd/system/gitea.service` e una volta sull'editor di testo copiamo il contenuto del **Listato 2**.

Salviamo il tutto **CTRL+X** → **Y** e **INVIO**. Adesso diamo i comandi `sudo systemctl daemon-reload` e `sudo systemctl enable --now gitea`.

Per verificarne lo stato possiamo usare il comando `systemctl status gitea` e se tutto è ok ci comparirà un output come in **Fig. 8**.

FRONT-END CON NGINX

Per impostazione predefinita, il Web Server di Gitea è in ascolto sulla porta 3000, ma per comodità possiamo usare la porta standard per l'HTTP (80)

oppure usare un altro webserver come reverse proxy. Noi abbiamo optato per la seconda opzione, utilizzando NGINX; quest'ultimo è un webserver molto apprezzato per le sue elevate performance e per il basso consumo di risorse, entrambe qualità ottime e apprezzate quando si tratta di eseguirne l'installazione su una board Raspberry Pi. Installiamo Nginx con il comando `sudo apt-get install nginx -y`. Terminato il processo di installazione editiamo il file di configurazione del sito usando il comando `sudo nano /etc/nginx/sites-available/default`.

Nell'editor editiamo la configurazione fino ad ottenerne una simile a quella del **Listato 3**. Nel campo `server_name` sostituiamo "lamiaistanzagitea.com" con il sottodominio che abbiamo scelto su NO-IP. Salviamo la configurazione premendo **CTRL+X** → **Y** e successivamente **INVIO**.

Testiamo la configurazione con il comando `sudo nginx -t` e se non riceviamo errori possiamo riavviare il webserver usando `sudo service nginx restart`.

❏ **Fig. 8**
Esito del comando di verifica dello stato.

```
pi@raspberrypi:~$ sudo systemctl status gitea
● gitea.service - Gitea (Git with a cup of tea)
   Loaded: loaded (/etc/systemd/system/gitea.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2022-03-21 09:57:15 CET; 2 days ago
     Main PID: 844 (gitea)
        Tasks: 11 (limit: 4915)
           CPU: 12min 42.644s
    CGroup: /system.slice/gitea.service
            └─844 /usr/local/bin/gitea web --config /etc/gitea/app.ini

Mar 21 09:57:15 raspberrypi systemd[1]: Started Gitea (Git with a cup of tea).
Mar 21 09:57:19 raspberrypi gitea[844]: 2022/03/21 09:57:19 cmd/web.go:102:runWeb() [I] Starting Gitea on PID: 844
Mar 21 09:57:19 raspberrypi gitea[844]: 2022/03/21 09:57:19 cmd/web.go:150:runWeb() [I] Global init
Mar 21 09:57:19 raspberrypi gitea[844]: 2022/03/21 09:57:19 routers/init.go:106:GlobalInitInstalled() [I] Git Version: 2.30.2, Wire Protocol Version 2 Enabled
Mar 21 09:57:19 raspberrypi gitea[844]: 2022/03/21 09:57:19 routers/init.go:109:GlobalInitInstalled() [I] AppPath: /usr/local/bin/gitea
Mar 21 09:57:19 raspberrypi gitea[844]: 2022/03/21 09:57:19 routers/init.go:110:GlobalInitInstalled() [I] AppWorkPath: /var/lib/gitea
Mar 21 09:57:19 raspberrypi gitea[844]: 2022/03/21 09:57:19 routers/init.go:111:GlobalInitInstalled() [I] Custom path: /var/lib/gitea/custom
Mar 21 09:57:19 raspberrypi gitea[844]: 2022/03/21 09:57:19 routers/init.go:112:GlobalInitInstalled() [I] Log path: /var/lib/gitea/log
Mar 21 09:57:19 raspberrypi gitea[844]: 2022/03/21 09:57:19 routers/init.go:113:GlobalInitInstalled() [I] Configuration file: /etc/gitea/app.ini
Mar 21 09:57:19 raspberrypi gitea[844]: 2022/03/21 09:57:19 routers/init.go:114:GlobalInitInstalled() [I] Run Mode: Prod
```

Listato 3 - Configurazione di Nginx

```
server {
    server_name lamiaistanzagitea.ddns.net;

    location / {
        proxy_pass http://localhost:3000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    listen 80;
}
```

CONFIGURAZIONE DEL PROTOCOLLO HTTPS

Pur avendo abilitato il port forwarding della porta 443, dobbiamo configurare il Web Server per accettare le richieste in ingresso su quest'ultima e, soprattutto, dobbiamo generare i certificati SSL. Possiamo eseguire entrambi i passaggi utilizzando **Certbot**, un software che genera gratuitamente certificati SSL.

Per installarlo riprendiamo la shell SSH ed installiamo/aggiorniamo il package manager snapd con il comando `sudo snap install core; sudo snap refresh core`. Installiamo successivamente certbot con il comando `sudo snap install --classic certbot` e una volta fatto creiamo il collegamento per poterlo

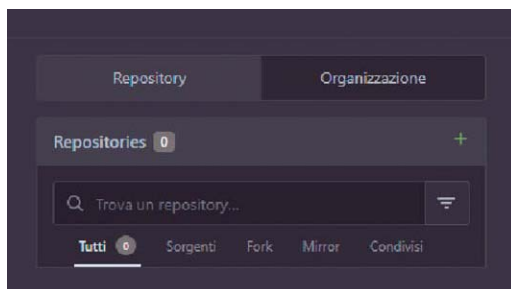
richiamare quando necessario usando `sudo ln -s /snap/bin/certbot/usr/bin/certbot`.

Possiamo generare il certificato usando il comando `sudo certbot --nginx`, accettiamo i termini di contratto quando richiesto e successivamente selezioniamo il sito per il quale richiedere il certificato. Il sistema provvederà automaticamente a richiedere il certificato, installarlo sulla configurazione di Nginx e a riavviare quest'ultimo.

Possiamo ora accedere all'istanza web usando il formato `https://lamiaistanzagitea.ddns.net` anziché `http://lamiaistanzagitea.ddns.net`. Nel caso in cui venisse tentato l'accesso mediante http, verrebbe effettuato in automatico il passaggio ad https.

Fig. 9
Schermata di installazione Gitea completa dei relativi dati.

Fig. 10
GUI di Gitea.



SETUP E CREAZIONE DEL REPOSITORY

Adesso abbiamo completato (momentaneamente) la procedura di configurazione mediante riga di comando. Rechiamoci al link <https://lamiaistanza.gitea.ddns.net> e ci apparirà una schermata per l'installazione. Inseriamo tutti i dati come in Fig. 9 (configurazione iniziale) e poi clicchiamo su "Impostazioni account amministratore" e inseriamo nome utente e password che preferiamo. Per completare l'installazione su "installare Gitea" e il

Fig. 11
Creazione di un nuovo repository.

gioco è fatto! Dopo qualche secondo, ci comparirà la schermata di login e logghiamoci con username e password. L'aspetto della GUI è quasi del tutto simile a quella di GitHub ed è proposta nella Fig. 10, sebbene vi sia un'ovvia differenza per quanto riguarda la gestione del sistema, degli utenti e delle configurazioni extra.

Per creare il primo repository rechiamoci in alto a destra, clicchiamo sul simbolo "+" → "Nuovo repository" e compiliamo i dati come in Fig. 11. Una volta fatto confermiamo e il nostro repository è pronto! È possibile sincronizzarlo caricando manualmente i file oppure usando i client come GitHub Desktop mediante il protocollo HTTPS.

CONFIGURAZIONE DI FAIL2BAN

Possiamo difenderci dagli attacchi di **brute-forcing** usando **Fail2ban**. Fail2ban è un software scritto in Python che si occupa di leggere i log dei vari software installati e, in base ai filtri e alle regole di funzionamento, blocca l'accesso alle porte del sistema usando **iptables**. Per installare Fail2ban torniamo sul terminale e digitiamo il comando:

```
sudo apt-get install fail2ban -y
```

Una volta terminata l'installazione dobbiamo creare un **filtro**, in modo tale che il software possa fare in automatico il parsing dei log e una "jail" per definire le regole del ban. Per creare il filtro digitiamo il comando `sudo nano /etc/fail2ban/filter.d/gitea.conf` e copiamo quanto contenuto di seguito:

```
# gitea.conf
[Definition]
failregex = .* (Failed authentication attempt|invalid credentials|Attempted access of unknown user).*
from <HOST>
ignoreregex =
```

Salviamo il file premendo i tasti **CTRL + X** → **Y** e digitando **INVIO**. Creiamo la **jail** digitando `/etc/fail2ban/jail.d/gitea.conf` e copiando il testo seguente:

```
[gitea]
enabled = true
filter = gitea
logpath = /var/lib/gitea/log/gitea.log
maxretry = 3
findtime = 3600
bantime = 900
action = iptables-allports
```

Salviamo il file premendo i tasti **CTRL + X** → **Y** e digitando **INVIO**. Riavviamo Fail2ban con `sudo`

```
pi@raspberrypi:~$ sudo fail2ban-client status gitea
Status for the jail: gitea
- Filter
  - Currently failed: 1
  - Total failed: 1
  - File list: /var/lib/gitea/log/gitea.log
- Actions
  - Currently banned: 0
  - Total banned: 0
  - Banned IP list:
```



Fig. 12

Esito del buon funzionamento del sistema.

`service fail2ban restart` e verifichiamone il funzionamento con `sudo fail2ban-client status gitea`. Possiamo testare Fail2ban tentando di eseguire un accesso con un set di credenziali errate e digitando di nuovo `sudo fail2ban-client status gitea`. Se il sistema funziona correttamente, ci comparirà un output come in **Fig. 12**.

MODIFICA DEL FILE APP.INI

Una volta terminata l'installazione e accertato che il sistema sia in funzione correttamente è necessario editare di nuovo il file di configurazione, per disabilitare la modalità installazione e la registrazione spontanea di nuovi utenti. Editiamo nuovamente il file `app.ini` nella cartella `/etc/gitea` e sostituite i valori `"INSTALL_LOCK = false"` con `"INSTALL_LOCK = true"` e `"DISABLE_REGISTRATION = false"` con `"DISABLE_REGISTRATION = true"`. Salviamo il documento. Adesso impostiamo i permessi del file in sola lettura con il comando `sudo chmod 640 /etc/gitea/app.ini` e riavviamo Gitea con `sudo systemctl restart gitea`.

CONCLUSIONI

Termina qui la presentazione di questo software open source, molto valido e molto leggero, utile soprattutto nell'ambito hobbistico aziendale, in cui è possibile fornire features di livello enterprise gratuitamente ed efficientemente. 



Cosa occorre?

Sono disponibili diversi starter kit con Raspberry Pi 4 Tipo B già pronti all'utilizzo alla pagina bit.ly/schedeRPI

Il materiale va richiesto a:

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

Dai alle tue piante il giusto nutrimento!

Misuratore di luce-umidità-PH



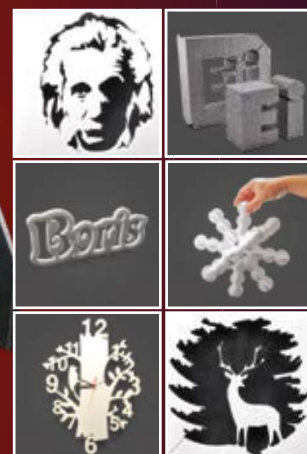
Se ti appresti a coltivare un giardino o un orto, ti sarà utile conoscere la qualità del terreno. Questo strumento 3 in 1 ti aiuterà a capire rapidamente quando le piante hanno bisogno di acqua, controllare il pH e capire se ricevono abbastanza luce. Basta inserire la sonda nel suolo a una profondità 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

DAI FORMA ALLE TUE IDEE!



cod. FOAMCUTTER

€ 320,00

PLOTTER TAGLIA POSTIROLO - IN KIT

Realizza, intagliandole a caldo, sagome di ogni genere partendo da lastre di polistirolo, polistirene ed altri materiali espansi a bassa temperatura di fusione. Collegalo subito al PC e utilizzalo per realizzare tutto quello che vuoi: modelli di vario genere, scritte, scatole, sagome per decorazioni, lettere scatolate, loghi, simboli, puzzle 3D e molto altro!

CARATTERISTICHE TECNICHE:

- **Tecnica di taglio:** a filo caldo
- **Materiali lavorabili:** espansi a bassa temperatura di fusione
- **Dimensioni:**
 - **Plotter:** 775x690x220 mm
 - **Area lavorabile:** 500x480 mm
 - **Spessore lavorabile:** 120 mm
- **Risoluzione:** 0,5 mm
- **Tolleranza X Y:** 0,012 mm
- **Velocità di taglio:** 300 mm/minuto (varia in funzione della tipologia di materiale lavorato)
- **Alimentazione:** 12 VDC / 3 A
- **Temperatura filo:** regolabile da software
- **Gestione:** da PC tramite interfaccia USB

Prezzo IVA inclusa.

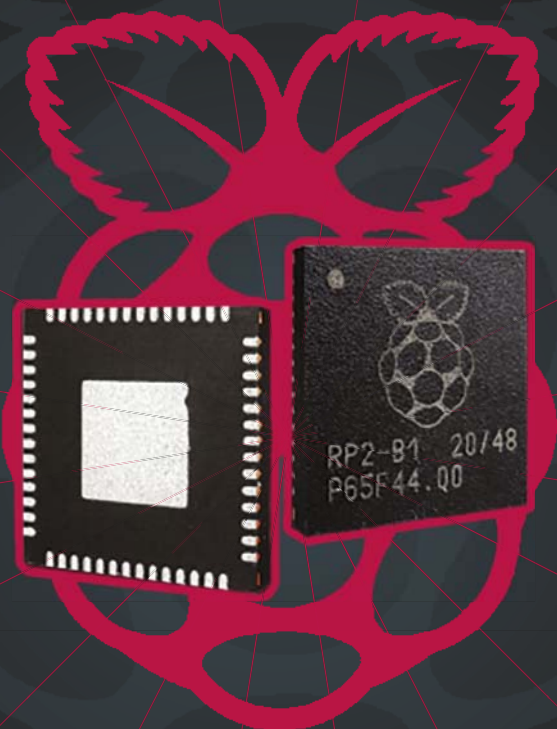
**FUTURA
ELETTRONICA®**

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

www.futurashop.it

Programmare con RP2040

di PIER CALDERAN



PUNTATA N. 1

Conosciamo e impariamo ad usare il nuovo chip RP2040 grazie alla scheda Raspberry Pi Pico e una board custom perfettamente compatibile con l'ambiente di programmazione previsto da Raspberry Pi Foundation.

Le ottime caratteristiche della scheda Raspberry Pi Pico, unite alle dimensioni, ridotte come fa presagire il nome, rendono tale hardware molto interessante per un vasto pubblico. Per questa ragione abbiamo pensato di dedicare un corso all'utilizzo e alla programmazione di tale scheda, ma come siamo soliti fare, invece di un corso sulla scheda Raspberry Pi Pico, abbiamo preferito costruirne una nostra, chiamandola FTR2040.

Si tratta di una scheda costruita attorno allo stesso microcontrollore RP2040 di Raspberry Pi, ma con caratteristiche pensate più per il maker e il progettista di hardware personalizzato.

Le caratteristiche del chip RP2040 sono:

- Dual ARM Cortex-M0+ a 133MHz;

- 264 kB di SRAM su chip in sei banchi indipendenti;
- supporto per un massimo di 16 MB di memoria Flash off-chip tramite bus QSPI dedicato;
- controllore DMA;
- architettura AHB (AMBA High-performance Bus);
- periferiche con interpolatore e divisore intero;
- regolatore LDO (Low Drop Out) programmabile su chip per generare la tensione del core;
- 2 PLL (Phase-Locked Loop) su chip per generare il clock USB e core;
- 30 pin GPIO, di cui quattro utilizzabili come ingressi analogici;
- 2 UART;
- 2 controller SPI;
- 2 controller I²C;
- 16 canali PWM;

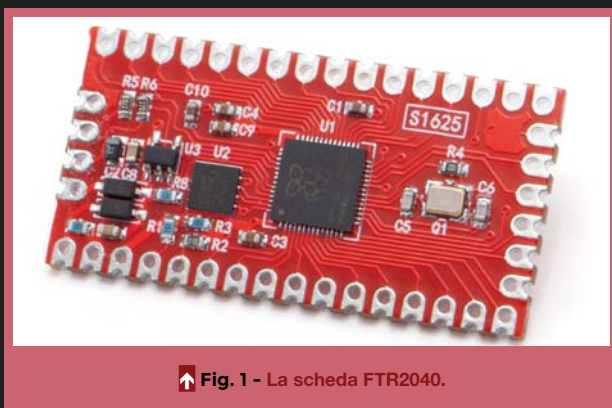


Fig. 1 - La scheda FTR2040.

- controller USB 1.1 e PHY, con modalità host e device;
- 8 PIO state machines.

La scheda FTR2040 (la vedete nella Fig. 1) viene fornita in versione "decidi-tu-cosa" saldare, ovvero con i pin saldabili secondo necessità: per esempio, se volete sperimentare inserendo la scheda in una breadboard, potete saldare i pin in modo che si vedano le serigrafie sul lato superiore. Attenzione però a saldare i pin in alto e in basso sul lato opposto perché altrimenti non riuscirete a infilare la scheda nella breadboard. Per contenere le dimensioni della scheda, viene resa disponibile una piccola breakout board per implementare la connettività USB; in pratica, la breakout ospita il connettore USB tipo A e dispone di quattro pad per collegarlo, mediante altrettanti fili, alla nostra scheda. Così, una volta programmata la scheda per il progetto finale, potete eliminare il breakout USB.

La scheda nuda e cruda potrà essere comodamente saldata su un PCB di vostra creazione.

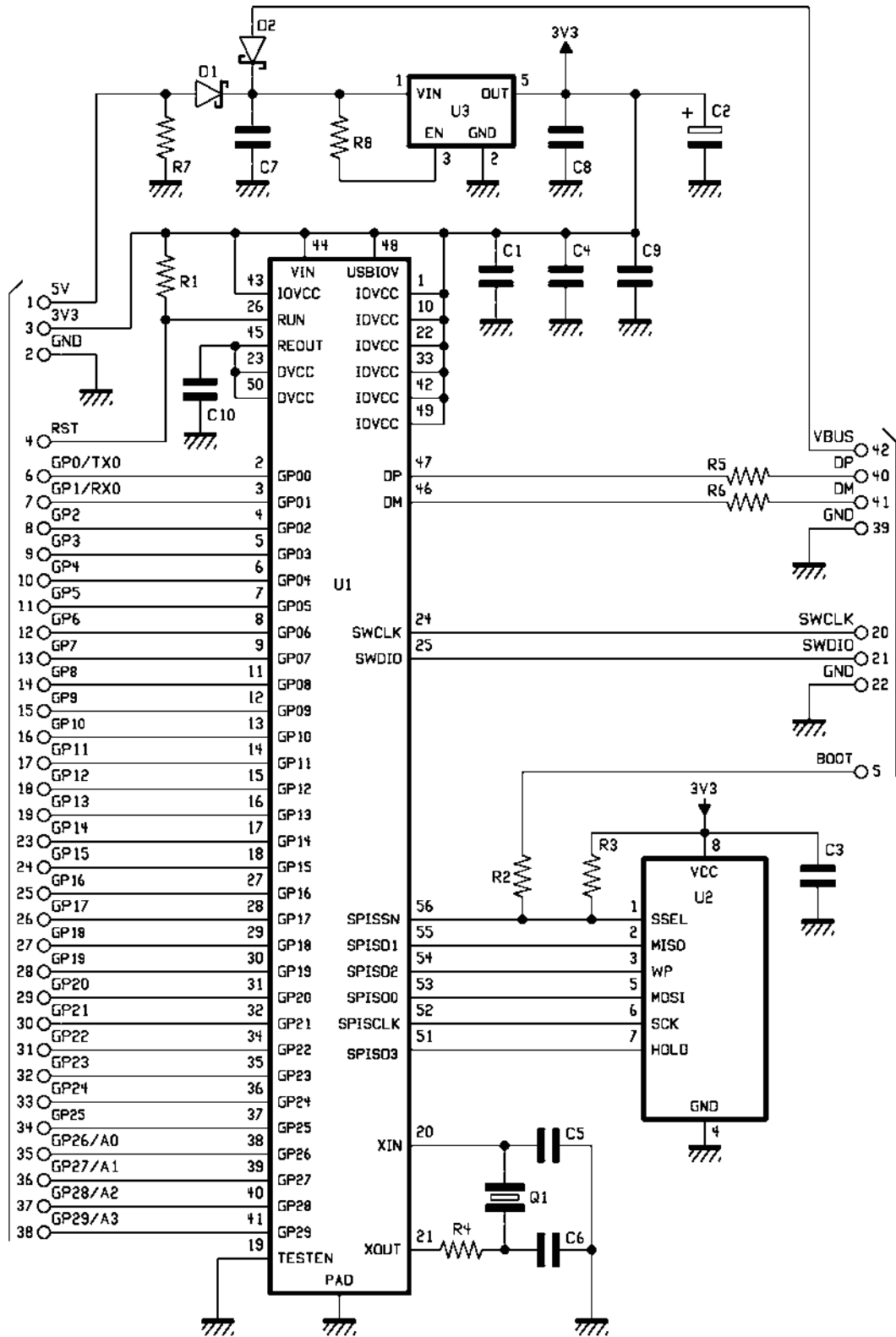
SCHEMA ELETTRICO

Prima di procedere è opportuno spendere qualche paragrafo per esporvi come è fatta la nostra board FTR2040, iniziando con la descrizione dello schema elettrico, il quale è proposto nella pagina qui accanto. Come potete intuire, il cuore della scheda è il chip RP2040, siglato U1 nello schema elettrico; si tratta di un potente microprocessore del quale rendiamo disponibili sui pad laterali della scheda i GPIO e le linee principali, nonché l'USB (pin 39, 40, 41, 42). Il clock che scandisce il funzionamento dell'U1 è ricavato grazie al quarzo Q1 e ai condensatori Q5 e Q6, connessi ai piedini di temporizzazione 29 e 21, che appartengono all'oscillatore principale e cui si connettono i componenti esterni. L'alimentazione per il chip è ricavata dal regolatore a basso drop-out U3, il quale, ricava 3,3V stabilizzati. Grazie ai diodi Schottky (utilizzati per minimizzare la caduta di tensione) D1 e D2, come fonte di alimentazione principale si possono utilizzare i 5V in arrivo dalla connessione USB o quelli provenienti dal contatto 5V. In altre parole, il circuito può essere alimentato attraverso la porta USB o il pin 5V, ma è anche possibile connettere la presa USB (attraverso la specifica breakout board) mentre il circuito è alimentato dai suoi pad di alimentazione, proprio grazie alla presenza degli Schottky, che impediscono ad una tensione di andare sulla linea dell'altra. Completiamo l'analisi dello schema elettrico con l'integrato siglato U2, il quale altro non è che l'indispensabile memoria Flash EPROM da 64Mbit ad accesso seriale, con modalità DUAL e QUAD SPI; come vedete, è interfacciata all'RP2040 attraverso una connessione Serial Peripheral Interface a 4 linee più 2 segnali di controllo. Questa memoria Flash serve a caricare i firmware che scriveremo e a contenere eventuali dati da memorizzare a permanenza, utili alle applicazioni che andremo a realizzare con la nostra board, sia durante questo corso, sia in autonomia.



Fig. 2 - La confezione e il contenuto di Pico Starter Kit.

schema ELETTRICO



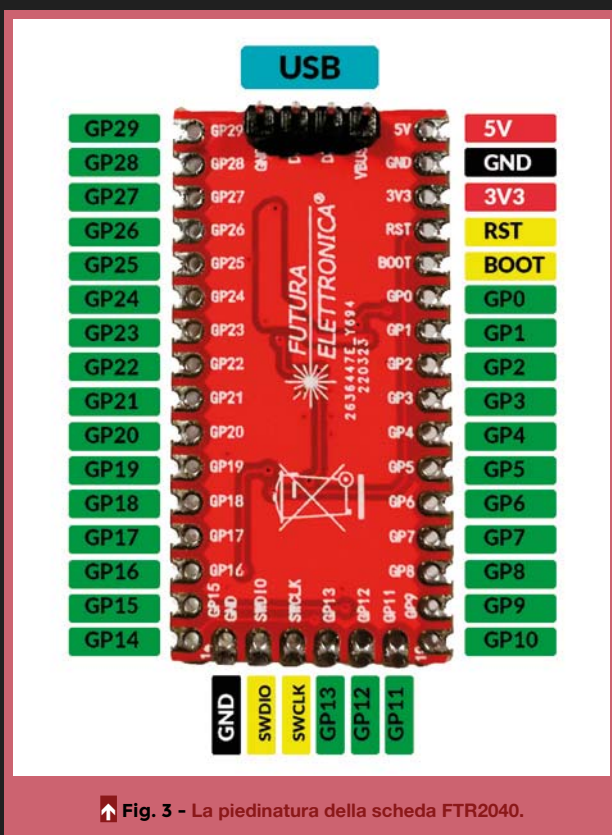


Fig. 3 - La piedinatura della scheda FTR2040.

REALIZZAZIONE DELLA BOARD

Il modulo FTR2040 può essere autocostruito seguendo il piano di montaggio illustrato nella pagina qui accanto e scaricando, per ottenere il relativo PCB, le relative tracce rame dalla sezione Download del nostro sito Internet www.elettronica.in.it, tra i file di questo corso. Il circuito stampato è un doppia faccia che richiede una certa competenza per essere realizzato, ma che potete farvi preparare da un servizio di circuiti stampati on-line come il PCBPRODUCTION di Futura Elettronica. In ogni caso, per il montaggio serve un minimo di attrezzatura, della quale deve

far parte un saldatore a punta molto fine, una stazione ad aria calda per lavorazione di SMD e BGA (la classica ed economica hot-air station in versione basic va sempre bene...) delle pinzette per posizionare i componenti prima della saldatura, del filo sottilissimo di stagno (diametro non superiore a 0,5 mm) o della pasta-salda per SMD, una lente d'ingrandimento per verificare il posizionamento del chip RP2040 prima di saldarlo e tutte le saldature effettuate (alla ricerca di baffi di stagno che potrebbero costituire cortocircuiti) e della pasta flussante.

Il montaggio deve iniziare dal chip RP2040, che è il componente più impegnativo e che va posizionato, rispettando il verso indicato nel piano di montaggio che vedete nella pagina accanto, ben centrato sulle rispettive piazzole, preventivamente cosparse di stagno e poi di pasta flussante; fatto ciò, scaldandolo con la stazione ad aria calda, si salderà quando avrete visto lo stagno brillare, allorché toglierete il getto d'aria calda.

La saldatura della Flash dovrà essere fatta similmente. Per il resto dei componenti, tutti molto più semplici, potrete procedere con il saldatore, sempre dopo aver stagnato e cosperso di flussante le piazzole.

Terminato il montaggio e verificato, con l'aiuto della lente d'ingrandimento che sia tutto a posto, la scheda è pronta per l'uso. Possiamo quindi scordarci l'hardware ed entrare nel vivo della sua programmazione.

Chi, tra voi, non ha l'attrezzatura o non se la sente di costruire da sé il circuito, non deve disperare perché Futura Elettronica (www.futurashop.it) rende disponibile la scheda FTR2040 già montata e collaudata.

LO SVILUPPO SOFTWARE

Similmente a quanto previsto per la board Raspberry Pi Pico, per lo sviluppo del software relativo alle applicazioni della nostra FTR2040, è disponibile l'SDK C/C++ di Raspberry Pi o MicroPython.

La programmazione del caso può essere effettuata utilizzando qualsiasi piattaforma, anche se è consigliabile l'uso di una

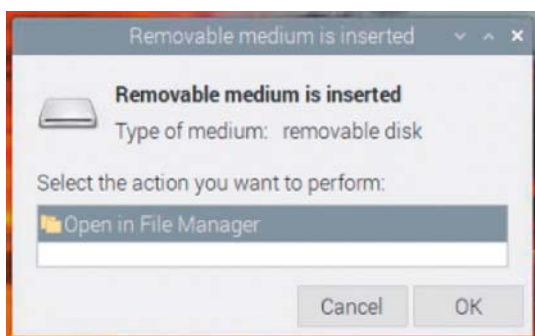


Fig. 4 - La finestra per montare l'unità rimovibile.

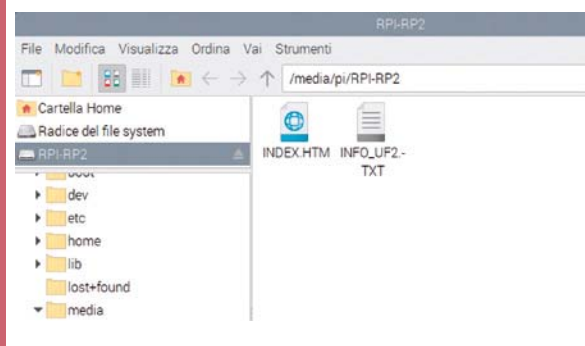
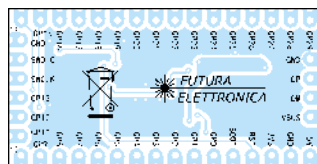
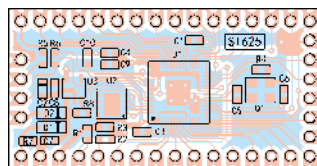


Fig. 5 - La cartella con i due file all'interno della cartella della scheda.

piano di MONTAGGIO



Elenco Componenti:

R1, R7, R8: 10 kohm (0603)
 R2, R4: 1 kohm (0603)
 R5, R6: 27 ohm (0603)
 R3: -
 C1, C3, C4, C10: 100 nF ceramic (0603)
 C9: 1 µF ceramic (0603)
 C2: 10 µF 6,3 VL tantalum (2012)
 C5, C6: 22 pF ceramic (0603)
 C7, C8: 10 µF ceramic (0603)
 D1, D2: MBR0520LT1G (SOD123)
 Q1: Crystal 12MHz (3.2x2.5 mm)
 U1: RP2040
 U2: W25Q64JVXG IQ TR
 U3: MIC5317-3.3YM5-TR

scheda Raspberry Pi, almeno per le prime volte. In questo corso esploreremo MicroPython, l'ambiente C/C++ e anche l'ambiente Arduino, giacché è possibile programmare l'RP2040 tramite l'IDE di Arduino. La possibilità di programmare la scheda tramite il popolare IDE Arduino, semplice e molto utilizzato, è un ulteriore punto a favore del nostro progetto perché lo avvicina anche a chi ha meno dimestichezza con gli altri tool.

RP2040 STARTER KIT

Per facilitarvi a seguire questo corso in maniera costruttiva fin da subito, abbiamo preparato uno speciale Pico Starter kit, che può essere utilizzato con la nostra scheda e qual-

siasi altra scheda basata su chip RP2040 come Raspberry pi Pico (Fig. 2). La confezione (kit) in vendita online su www.futurashop.it con il codice RPICOKIT, comprende una serie di componenti elettronici utili a realizzare gli esperimenti descritti in questo corso, oltre alla scheda Pico.

PREPARAZIONE DELLA SCHEDA FTR2040

Per utilizzare in pratica la board, bisognerà prepararla in base a come verrà impiegata; per prima cosa, vi consigliamo di saldare i pin strip in dotazione nelle piazzole di I/O e alimentazione, per l'inserimento in una breadboard o su un altro circuito stampato. Fatelo stando attenti a saldare sul lato opposto i pin per la connessione con la breakout USB.

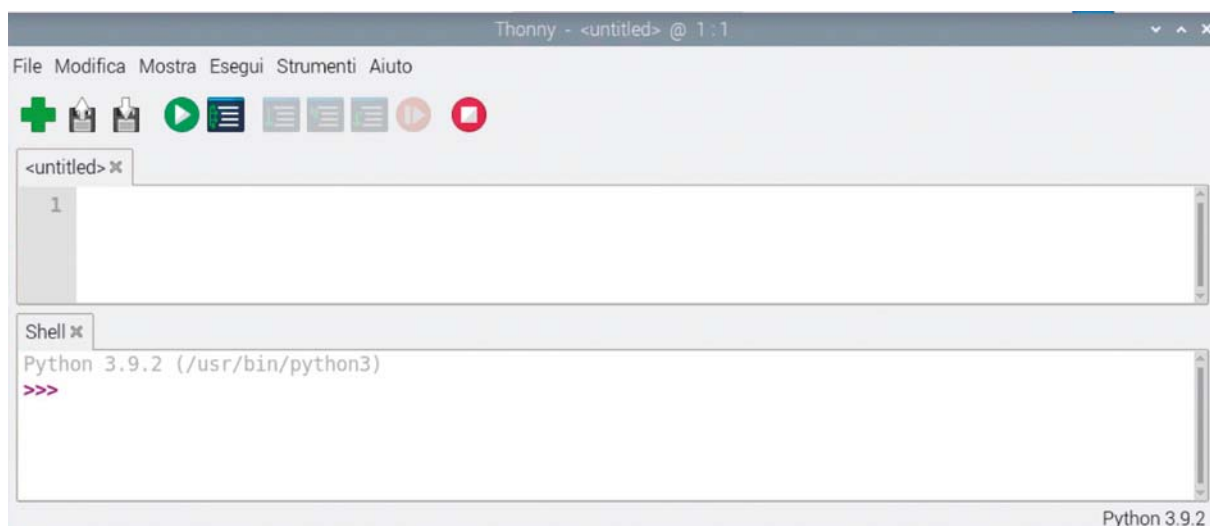


Fig. 6 - La finestra principale di Thonny.



Fig. 7 - La scelta dell'interprete MicroPython per Pico.

Per il collegamento USB, vi consigliamo di usare una prolunga maschio femmina A/A per collegare la scheda al computer. Anche se è possibile programmare la scheda usando un computer qualsiasi, per comodità abbiamo preferito una scheda Raspberry Pi 4 connessa in VNC con un PC.

Questo facilita un po' di cose, soprattutto perché potete lavorare subito con la scheda usando Thonny, ovvero l'editor per Python e MicroPython preinstallato nel sistema operativo Raspberry Pi.

Se guardate la serigrafia della scheda (Fig. 3) individuate la scritta "BOOT" che è l'abbreviazione per "boot selector" ovvero selettore di avvio.

Prima di collegare la presa USB a Raspberry Pi, dovete cortocircuitare il pin BOOT a massa (quindi il pin 5 a un pin GND qualsiasi). In questo modo la scheda viene abilitata alla programmazione.

Dopo aver collegato la scheda FTR2040 alla presa USB apparirà un finestra di notifica che vi avverte di montare il dispositivo come unità rimovibile (Fig. 4). Facendo clic su OK, la scheda verrà montata come un'unità rimovibile e a questo punto potete togliere il corto fra BOOT e GND.

Una volta montata l'unità, apparirà una cartella con all'interno due file, come mostrato nella Fig. 5:

- INDEX.HTM
- INFO_UF2.TXT

Il file INFO_UF2.TXT riporta semplicemente la versione del bootloader e il tipo di scheda, mentre il file INDEX.HTM è un link alla documentazione di Raspberry Pi Pico: <https://raspberrypi.com/device/RP2?version=E0C9125B0D9B>.

INSTALLAZIONE DI MICROPYTHON

Aprite il file INDEX.HTM con Chromium di Raspberry Pi e andate nella pagina della documentazione di Raspberry Pi Pico. Selezionate la scheda "MicroPython" per aprire la sezione specifica di MicroPython. MicroPython è un'implementazione completa del linguaggio di programmazione Python 3 che viene eseguito direttamente su hardware embedded come Raspberry Pi Pico e quindi anche la nostra scheda.

Se non avete mai usato MicroPython potete visitare il link MicroPython Wiki nella stessa pagina (<https://github.com/micropython/micropython/wiki>).

Nella sezione MicroPython selezionate il link "downloadable UF2" per scaricare il firmware MicroPython.

Verrà scaricato nella cartella Download un file UF2 (USB Flashing Format) con un nome simile a "rp2-pico-20220117-v1.18.uf2". Si tratta di un file che serve a installare MicroPython per Pico/RP2040.

Aprite un'altra finestra del file manager e recatevi nella cartella Download. Trascinate il file con l'estensione ".uf2" nella finestra dell'unità rimovibile. Una volta rilasciato il file, verrà installato automaticamente MicroPython e l'unità rimovibile verrà smontata automaticamente. Dopo alcuni secondi la cartella scomparirà dal File Manager.

PROGRAMMAZIONE CON MICROPYTHON

Per semplicità di spiegazione e soprattutto per facilitare l'approccio ai principianti, oggi parleremo solo della programmazione con MicroPython e non C/C++.

Dal suo lancio nel 1991, il linguaggio di programmazione Python è cresciuto fino a diventare uno dei più popolari al mondo. Il linguaggio Python è stato sviluppato per sistemi

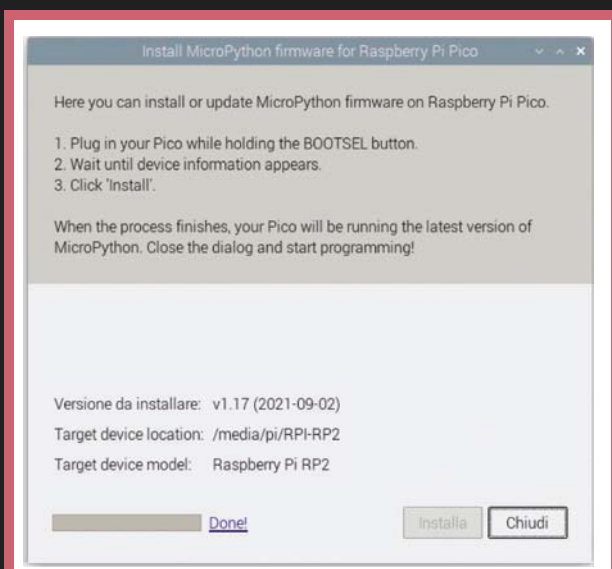


Fig. 8 - L'installazione dell'interprete MicroPython per Pico.

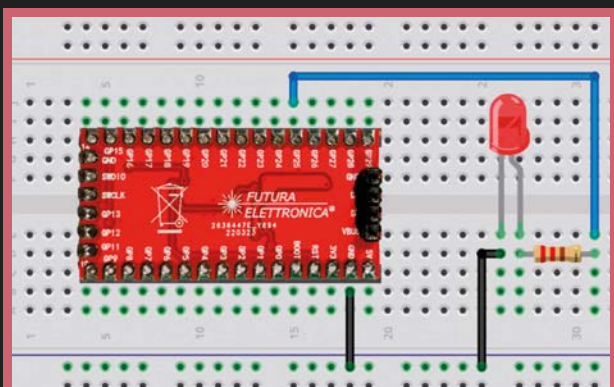


Fig. 9 - Il LED collegato alla scheda FTR2040.

informatici di qualsiasi tipo e grandezza, ma i microcontrollori come RP2040 sono piccoli, più semplici e con una quantità di memoria notevolmente inferiore, il che significa che non possono eseguire lo stesso codice Python delle controparti più grandi.

È qui che entra in gioco MicroPython. Sviluppato originariamente da Damien George e rilasciato per la prima volta nel 2014, MicroPython è un linguaggio di programmazione compatibile con Python, ma sviluppato specificamente per i microcontrollori, da qui il nome. Include molte delle funzionalità di Python, aggiungendone di nuove progettate specificamente per sfruttare le funzionalità disponibili su RP2040. Se avete già programmato con Python, troverete MicroPython immediatamente familiare. In caso contrario, è un linguaggio che vi consigliamo di imparare.

THONNY

Prima di iniziare a programmare Raspberry Pico o l'FTR2040 con MicroPython, dovrete configurare un ambiente di sviluppo integrato (IDE).

Thonny (Fig. 6) è un popolare ambiente IDE per Python e MicroPython ed è precaricato nel sistema operativo Raspberry Pi. Se invece volete utilizzare Pico con una diversa distribuzione Linux, o usare Windows o macOS, potete scaricare dal sito ufficiale "<https://thonny.org>" il programma di installazione del bundle Thonny e Python per il relativo sistema operativo o le istruzioni di installazione per Linux.

Thonny integra tutti gli strumenti necessari per scrivere e sviluppare software in un'unica interfaccia utente. Anche se esistono molti altri ambienti IDE per Python, vi consigliamo di utilizzare solo Thonny per la programmazione di RP2040.

Nella parte superiore dell'interfaccia utente di Thonny potete vedere la barra degli strumenti, che offre un sistema di accesso rapido basato su icone e un menu per l'accesso a varie funzionalità del programma. Il menu appare solo in modalità "Regular" ed "Expert", per cui è consigliabile fare clic su "Switch to regular mode" in alto a destra alla prima apertura di Thonny. In questo modo potrete accedere al menu Tools e cambiare altre opzioni come, per esempio, la lingua dell'edi-

tor. Fra le tante c'è anche l'italiano. Da qui in avanti i riferimenti e le figure sono relativi all'interfaccia utente in italiano e la modalità dell'interfaccia utente "Regular". La scelta della modalità si trova sotto Strumenti > Opzioni > Generali > Modalità UI. L'area centrale dell'editor è dedicata agli script, dove vengono praticamente scritti tutti i programmi in linguaggio Python e/o MicroPython. La parte inferiore è la cosiddetta Python Shell che consente di digitare singole istruzioni che vengono eseguite non appena premete il tasto INVIO e fornisce anche informazioni sugli script in esecuzione, per esempio, le istruzioni print, gli errori di esecuzione eccetera. Thonny può essere avviato dal menu Programmazione di Raspberry Pi. Per impostare Thonny come interprete per Pico, fate clic nella parte inferiore destra della finestra in corrispondenza della scritta con il nome dell'interprete e della versione, per "Python 3.9.2".

Nell'elenco che appare, selezionate "MicroPython (Raspberry Pi Pico)" (Fig. 7). Questa opzione è presente anche in Strumenti > Opzioni > Interprete.

Fatto questo, si aprirà una finestra (Fig. 8) dalla quale installare o aggiornare MicroPython nel Pico.

IL PRIMO PROGETTO

Il piccolo microcontrollore RP2040 è stato progettato pensando al calcolo fisico. I suoi numerosi pin di input/output (GPIO) per uso generico consentono di comunicare con una vasta gamma di componenti, consentendo di creare progetti anche molto complessi.

Non vogliamo soffermarci su print ("Hello World!") e altre istruzioni di Python che già dovrete conoscere. Concentriamoci piuttosto sul primo progetto "fisico" con Pico.

BLINK DI UN LED

Per capire come funziona il tutto, vi consigliamo di scrivere nell'editor di Thonny queste poche righe che servono a far lampeggiare a intervalli di un secondo un LED collegato al pin GP25 (che corrisponderebbe al LED interno di una scheda Pico) (Fig. 9). Ricordatevi di mettere una resistenza da 220 Ω in serie al LED. Per chi non lo sapesse, il codice di Python va indentato seguendo l'esempio seguente:

```
import machine
import utime

led = machine.Pin (25, machine.Pin.OUT)

while True:
    led.value (1)
    utime.sleep (1)
    led.value (0)
    utime.sleep (1)
```

La prima riga di codice import machine è la chiave per lavorare con MicroPython su RP2040. La libreria "machine"

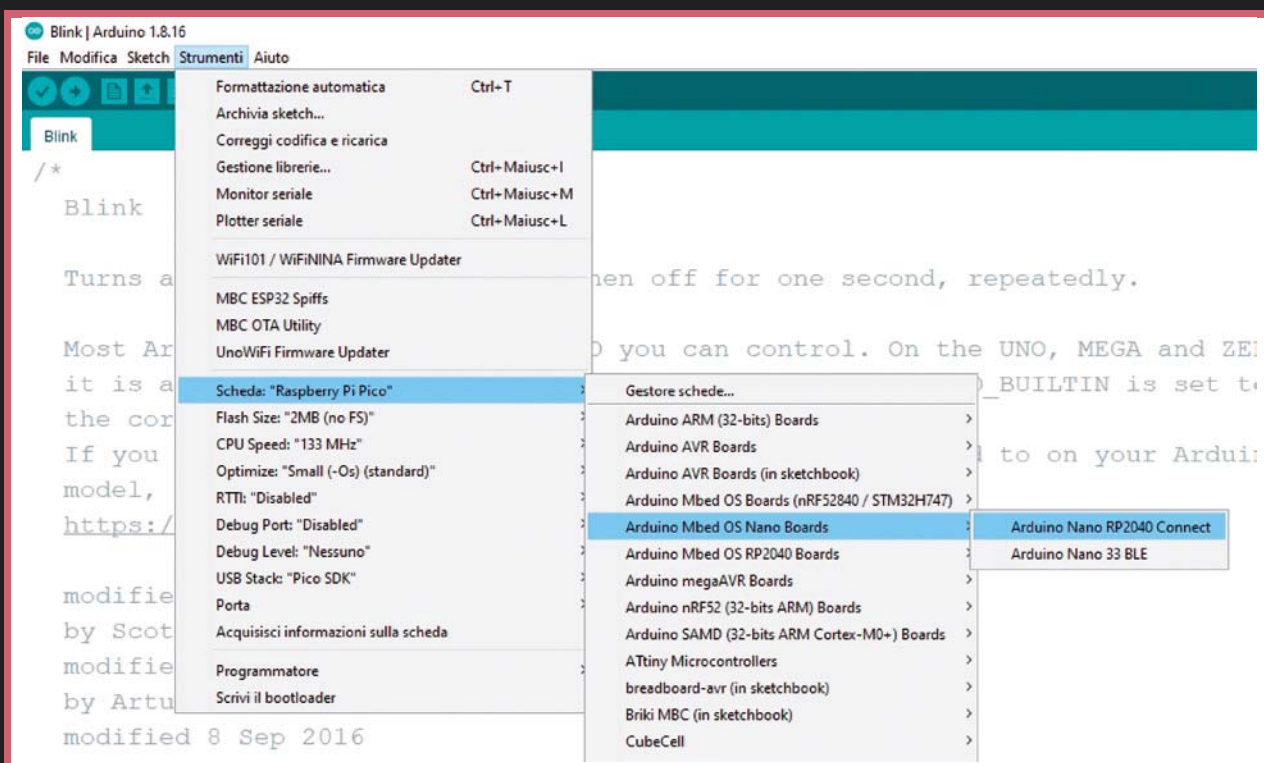


Fig. 10 - Selezione della scheda Arduino Nano RP2040 Connect.

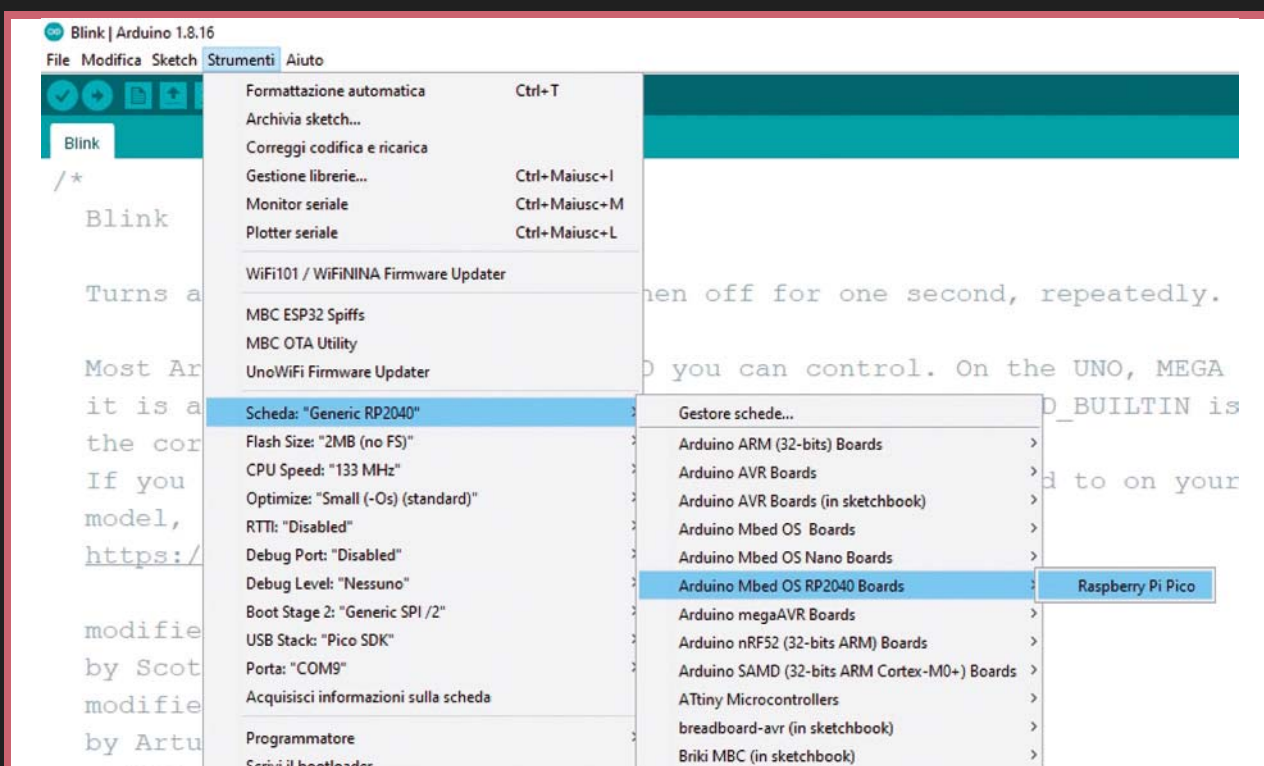


Fig. 11 - Selezione della scheda Raspberry Pi Pico.

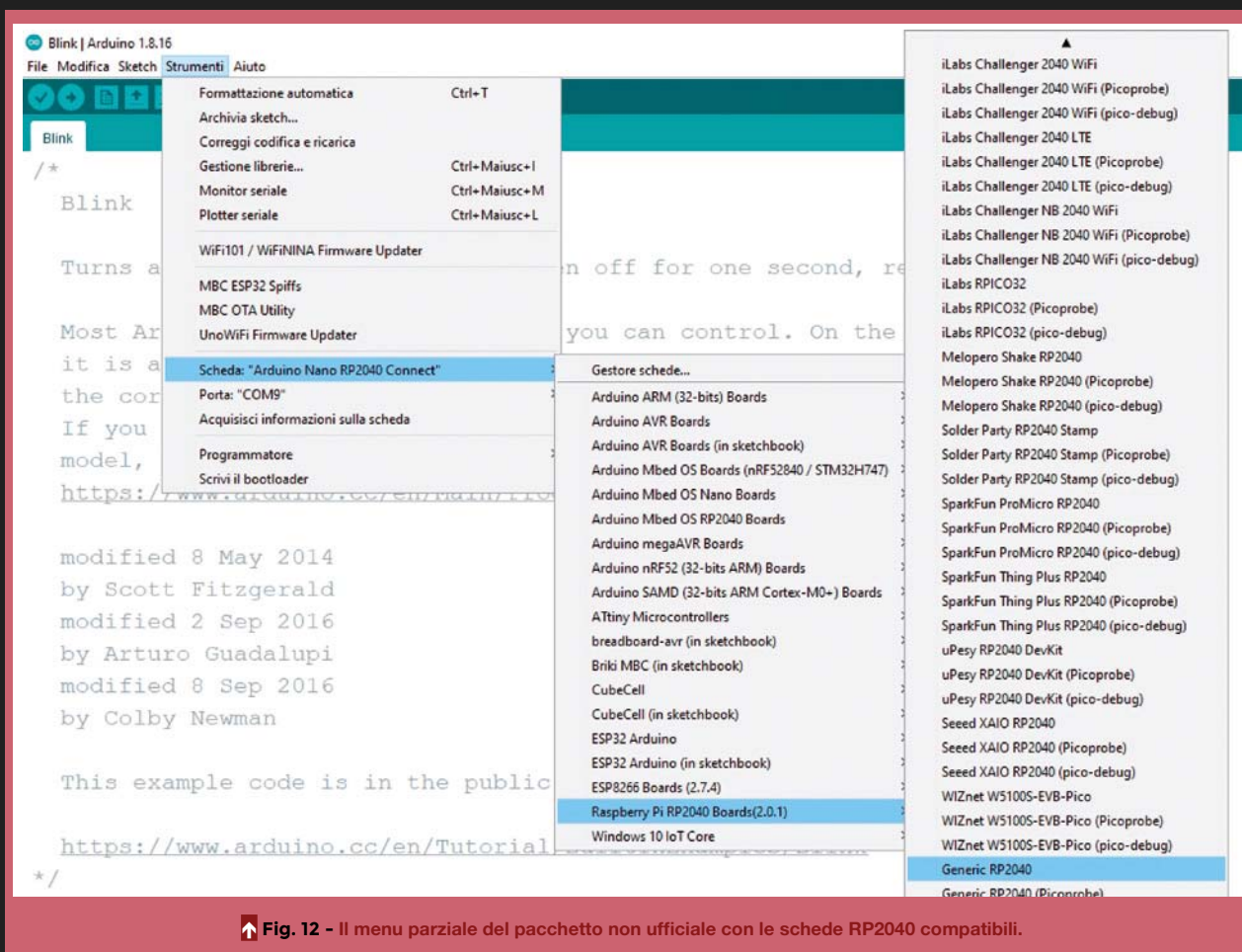


Fig. 12 - Il menu parziale del pacchetto non ufficiale con le schede RP2040 compatibili.

contiene tutte le istruzioni necessarie a MicroPython per comunicare con Pico e altri dispositivi compatibili, estendendo il linguaggio per il calcolo fisico.

Senza questa riga, non sarete in grado di controllare nessuno dei pin GPIO di Pico e, nel nostro caso, non potremo accendere il LED collegato al pin GP25.

Nella riga successiva del programma potete cambiare il pin a piacere collegandolo a un altro pin:

```
led = machine.Pin (25, machine.Pin.OUT)
```

Questa riga definisce un oggetto chiamato led, che è un nome puramente descrittivo. È ovviamente possibile utilizzare qualsiasi nome, ma è meglio attenersi a nomi che descrivono lo scopo della variabile, per rendere il programma più leggibile e comprensibile.

L'istruzione `machine.Pin(25, machine.Pin.OUT)` chiama la funzione della libreria per gestire i pin GPIO della scheda:

- il primo argomento, 25, è il numero del pin che si sta configurando;
- il secondo, `machine.Pin.OUT`, dice al pin deve essere usato come output.

All'interno del ciclo while, l'istruzione `led.value (1)` accenderà il LED collegato, mentre l'istruzione `led.value (0)` lo spegnerà. L'istruzione `utime.sleep (1)` imposta il ritardo a 1 secondo fra i due stati.

LIBRERIA UTIME INVECE DI TIME

Invece di importare la classica libreria `time` di Python per creare temporizzazioni e ritardi, qui viene usata la libreria `utime` (u sta per la lettera μ = micro).

Si fa notare che la libreria `utime` è una versione progettata per microcontrollori e anche se si importa la libreria `time`, MicroPython utilizzerà automaticamente la libreria `utime`.

ESECUZIONE DEL PROGRAMMA

Per eseguire il programma basta fare clic sul pulsante Esegui o F5 da tastiera o dal menu Esegui. Vi verrà chiesto di salvare il programma sul computer locale oppure nella scheda. Salvate il file nella scheda con il nome "blink_test.py", per esempio. Se non avete commesso errori, vedrete il LED lampeggiare a intervalli di un secondo. Premete l'icona Stop per fermare il programma e il lampeggio del LED.

Se volete che il lampeggio del LED continui, eseguite il pro-

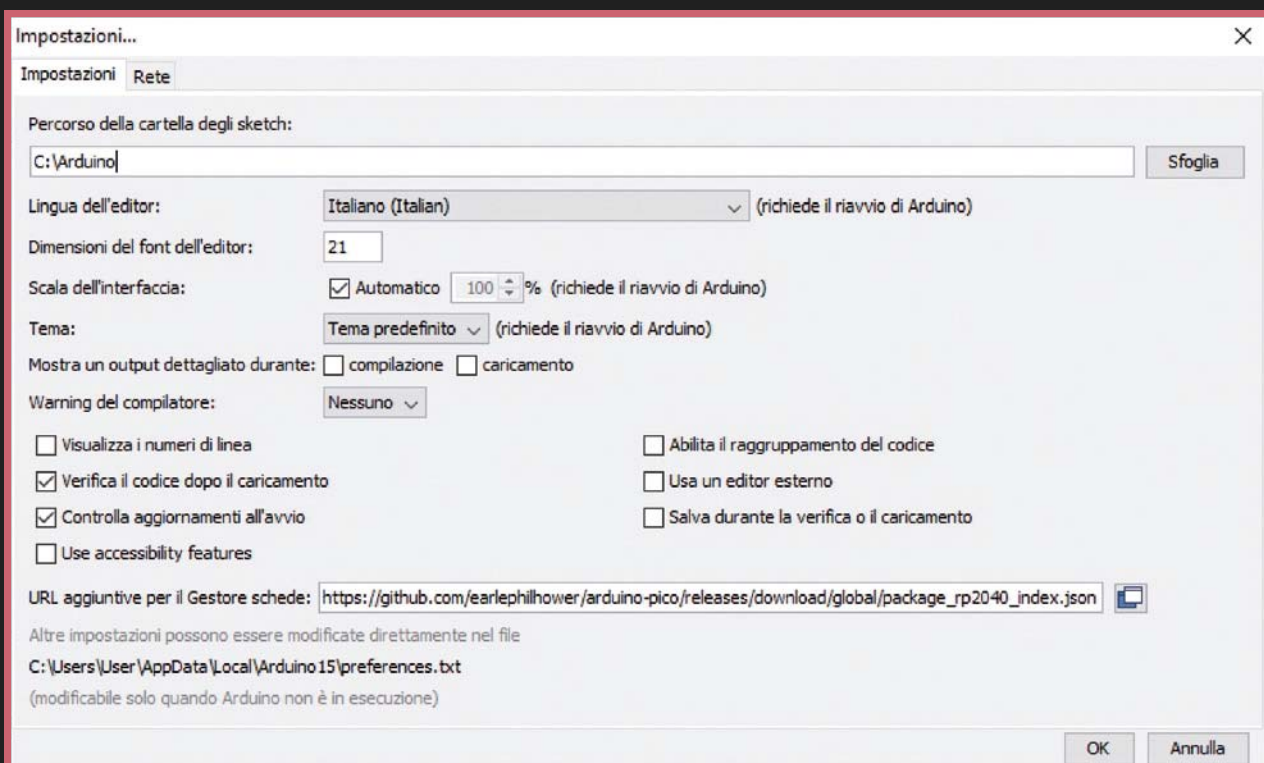


Fig. 13- Casella URL aggiuntivi per il Gestore schede.

gramma senza premere Stop. Potete quindi uscire anche da Thonny.

Se volete che il programma venga eseguito in autostart, alimentandolo anche da un'altra sorgente, dovete salvarlo sempre con il nome "main.py" all'interno della scheda. Attenzione che quando si parla dei pin GPIO di RP2040, di solito vengono indicati con i loro nomi completi: per esempio, GP25 indica il pin GPIO 25.

In MicroPython, le lettere GP devono essere eliminate, quindi nel programma bisogna scrivere "25" anziché "GP25", altrimenti non funzionerà.

BLINK CON TOGGLE

C'è anche un altro modo per gestire il lampeggio del LED. Se si usa la funzione toggle della libreria machine potete ottenere lo stesso risultato con le seguenti righe di codice:

```
import machine
import utime

led = machine.Pin (25, machine.Pin.OUT)

while True:
    led.toggle()
    utime.sleep (1)
```

La funzione toggle() è disponibile su tutti i pin di uscita

digitale e commuta semplicemente tra on e off: se il pin è alto, toggle() lo disattiva; se è basso, toggle() lo attiva.

Basta solo collegare nella breadboard un LED a un pin GPIO qualsiasi.

Per esempio, potete modificare l'istruzione precedente con la seguente per controllare un LED collegato a GP15 scrivendo:

```
led_external = machine.Pin (15, machine.Pin.OUT)
```

PROGRAMMARE L'RP2040 CON L'IDE DI ARDUINO

Il modo più semplice per i nuovi utenti è utilizzare MicroPython. Per gli utenti più esperti, un modo più avanzato è scrivere codice in C/C++.

C'è un terzo modo in cui possiamo scrivere codice per il nostro RP2040, e cioè tramite l'IDE di Arduino, che molti conoscono e che trovano senza dubbio più semplice del C++. Per programmare l'RP2040 con l'IDE di Arduino si può procedere in tre modi, tutti validi, ma con sostanziali differenze:

1. selezionare la scheda Arduino Nano RP2040 Connect;
2. installare il pacchetto ufficiale RP2040;
3. installare il pacchetto non ufficiale RP2040.

Metodo 1

Questo metodo prevede semplicemente la selezione della scheda Arduino Nano RP2040 Connect dal menu Strumenti > Scheda > Arduino Mbed OS Nano Boards (Fig. 10). Si tratta di una scheda Arduino perfettamente compatibile con il chip

RP2040, come quello della nostra scheda.

L'unico neo è che gli esempi del menu File sono relativi solo alla suddetta scheda Arduino.

Metodo 2

Questo metodo prevede l'installazione del pacchetto per RP2040 ufficiale ovvero quello reso disponibile da Arduino (Fig. 11). Dal menu Gestore schede basta cercare Pico e si potrà installare il pacchetto per la scheda Raspberry Pi Pico. Anche in questo caso gli esempi sono dedicati alla scheda Raspberry Pi Pico, ma la cosa non trascurabile è che questo pacchetto risulta deprecato.

Metodo 3

Questo metodo prevede l'installazione del pacchetto per RP2040 non ufficiale. Cosa significa? È un pacchetto sviluppato da Earle F. Philhower III che ha reso disponibile nel suo GitHub l'installazione per (praticamente) tutte le schede esistenti, basate sul chip RP2040 (Fig. 12).

Il pacchetto non ufficiale di Earle F. Philhower III è preferibile rispetto agli altri e sarà quello che useremo in questo corso. A oggi è quello che fornisce più opzioni di programmazione con la gestione di tutte le schede di diverso tipo ma con lo stesso microcontrollore RP2040. Da sottolineare il fatto che il menu File è ricco di esempi pronti all'uso.

Per installare il pacchetto RP2040 non ufficiale dovete seguire la procedura esposta qui di seguito.

1. Aprite l'IDE e andate in File > Impostazioni. Copiate e incollate il seguente link nella casella URL aggiuntivi per il Gestore schede (riferitevi alla Fig. 13): https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json.
2. Se avete altri URL, separateli con una virgola.
3. Fate clic su OK per salvare le impostazioni.
4. Andate in Strumenti > Scheda > Gestore schede per aprire il Gestore schede.
5. Si aprirà la finestra Gestore schede che caricherà anche il riferimento URL che avete appena inserito.
6. Cercate nel campo di ricerca "Pico".
7. Troverete la scheda "Raspberry Pi Pico/RP2040 by Earle F. Philhower III".
8. Selezionate l'ultima versione e fate clic su "Installa".
9. Dopo che l'installazione del pacchetto non ufficiale è terminata, vedrete la voce INSTALLED di fianco al nome del pacchetto.

Adesso nel menù Strumenti > Scheda > Gestore schede troverete il nuovo menù: Raspberry Pi RP2040 Boards (2.0.1) o una versione più aggiornata. Per poter programmare la nostra scheda FTR2040 dovete selezionare la prima voce di menù: "Raspberry Pi Pico". Per programmare la scheda FTR2040 utilizzando l'IDE di Arduino con il pacchetto non ufficiale, seguite quanto spiegheremo qui di seguito.

IL NOSTRO PRIMO SKETCH PER RP2040

Per vedere se tutto funziona come atteso, vi consigliamo di caricare il solito Blink dal menu esempi dell'IDE di Arduino. Quando collegate la scheda FTR2040 con il cavo USB, ricordatevi di cortocircuitare momentaneamente il pin BOOT a massa, come fatto in precedenza. Questo installerà una porta seriale che sarà disponibile nel menu Strumenti. Selezionate questa porta per caricare lo sketch. In un Mac il nome della porta sarà qualcosa di simile a /dev/cu.usbmodem (Raspberry Pi Pico) mentre, nel caso di un computer Windows potrebbe essere ad esempio COM6 (Raspberry Pi Pico).

Premete il pulsante Carica per compilare e caricare il codice nella scheda.

Durante il caricamento sentirete il tipico suono di scollegamento della porta USB. Se tutto è andato bene vedrete il messaggio: "Caricamento completato".

Dopo il caricamento vedrete il LED collegato al pin 25 della scheda che inizia a lampeggiare a intervalli di 1 secondo.

CONCLUSIONI

Bene, per il momento ci fermiamo qui. Con la speranza di essere stati sufficientemente chiari nell'esporre le varie nozioni di questa prima puntata, vi invitiamo a sperimentare i progetti presentati in queste pagine e vi aspettiamo il mese prossimo.

Per vostra comodità, abbiamo reso disponibile tutto il codice degli esempi di questo corso nell'area download del nostro sito www.elettronica.in.it, insieme a file del corso.

Non ci resta che darvi appuntamento alla prossima puntata, augurandovi buon lavoro! 



Cosa occorre?

La board di Futura Elettronica con RP2040 (cod. FTR2040) è disponibile a Euro 13,90 e include i pin strip da saldare.

Dal sito futuraishop.it è possibile acquistare lo starter kit con Raspberry Pi Pico (cod. RPICOKIT) a Euro 49,90 o il solo Raspberry Pi Pico (cod. RPI-PICO) a Euro 6,90. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

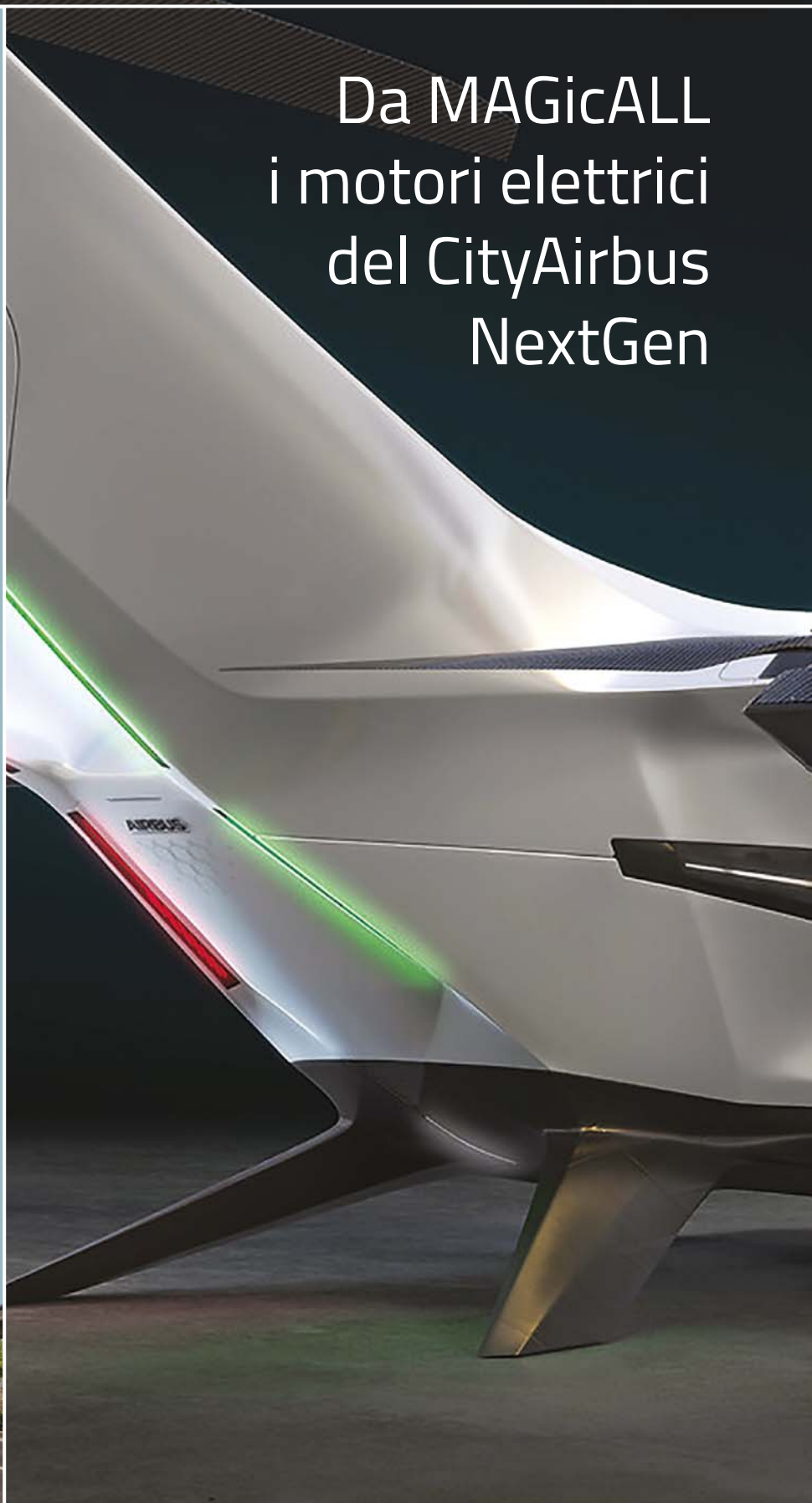
Baidu porterà i taxi senza conducente sulle strade di Pechino

Baidu ha ricevuto i primi permessi che autorizzano a fornire servizi di trasporto senza conducente al pubblico su strade della capitale Pechino. La società afferma che l'approvazione normativa segna una pietra miliare per l'industria del trasporto autonomo in Cina. Con i permessi rilasciati dalla sede centrale della Beijing High-level Automated Driving Demonstration Area (BJHAD), 10 veicoli autonomi senza conducente al volante offriranno passaggi ai passeggeri in un'area designata di 60 km² a Pechino. Le auto si uniranno a una flotta esistente fornita da Apollo Go, il servizio di ride-hailing autonomo di Baidu, nella capitale. Baidu ha la più grande flotta di guida autonoma in Cina. Nell'ampliare i suoi servizi di veicoli senza conducente, l'azienda afferma di aver lavorato per affrontare le sfide tecniche uniche del complesso ambiente del traffico di Pechino. Sono già in corso piani per aggiungere altri 30 veicoli di questo tipo, ampliando la flotta per fornire al pubblico servizi senza conducente più convenienti. La concessione dei permessi avvicina anche Baidu al funzionamento scalabile dei suoi servizi di trasporto a chiamata autonomi a Pechino. Apollo Go si è esteso a nove città in Cina dal suo lancio nel 2020, comprese tutte le città di primo livello del paese (Pechino, Shanghai, Shenzhen e Guangzhou) e altre cinque città (Chongqing, Changsha, Cangzhou, Yangquan e Wuzhen). Un altro operatore di robotaxi è Pony.ai, che ha avviato il programma senza passeggeri da ottobre 2021, ma non con il pubblico; inizialmente, un supervisore siede sul sedile del passeggero anteriore. Circa 300.000 residenti nell'area possono ora chiamare un robotaxi Pony.ai senza conducente utilizzando l'app PonyPilot+ dalle 9:00 alle 17:00. Centinaia di punti di ritiro e riconsegna includono strutture pubbliche come stazioni della metropolitana, parchi e stadi, oltre a centri commerciali centrali e aree residenziali.

www.apollo.auto



Da MAGicALL i motori elettrici del CityAirbus NextGen



Airbus ha scelto MAGicALL per la fornitura dei motori del proprio velivolo elettrico a decollo e atterraggio verticale (eVTOL) CityAirbus NextGen.

Il prototipo eVTOL di Airbus sarà dotato di una versione su misura di MAGiDRIVE, l'ultima generazione di motori elettrici prodotti da MAGicALL.

La personalizzazione dettagliata del sistema di propulsione elettrica di CityAirbus NextGen consentirà al prototipo di beneficiare di caratteristiche che si abbineranno al suo design unico, aumentando le prestazioni e l'affidabilità.

I leggeri motori brushless che saranno integrati in CityAirbus NextGen sono un prodotto di design e produzione innovativi: incorporano l'ultima generazione della tecnologia magnetica ed elettronica di potenza di MAGicALL.

Ciascuno muoverà un'elica del velivolo a quattro posti che è poi un multicottero per brevi spostamenti, capace di un'autonomia di 80 km a una velocità di crociera di 120 km/h.

I motori elettrici sono una componente importante del prototipo, perché l'elevata coppia e il peso ridotto di cui il MAGiDRIVE è capace, nonché il suo design robusto, saranno estremamente vantaggiosi per l'efficienza complessiva del velivolo.

I motori elettrici contribuiranno anche all'architettura senza soluzione di continuità del prototipo eVTOL di Airbus, grazie a una soluzione completamente integrata che comprende sia il motore stesso che il suo sistema di controllo, come parte di un'unità di propulsione completamente chiusa e raffreddata ad aria.

MAGicALL è un fornitore leader di motori elettrici con sede in California. Fondata nel 2004, l'azienda beneficia di una comprovata esperienza in soluzioni personalizzate magnetiche ed elettroniche di potenza per l'industria aerospaziale.

MAGicALL ha anche contribuito allo sviluppo dei motori che hanno alimentato il dimostratore tecnologico UAM ad ala inclinata di Airbus, Vahana.

Dal 2014, Airbus sta esplorando come la propulsione elettrica può aiutare a guidare lo sviluppo di nuovi tipi di veicoli aerei. Nel settembre 2021, la Società ha presentato il suo prototipo eVTOL completamente elettrico, CityAirbus NextGen. Airbus sta sviluppando una soluzione UAM con eVTOL non solo per offrire un nuovo servizio di mobilità, ma anche come passo importante nella sua ricerca per ridurre le emissioni nell'aviazione in tutta la sua gamma di prodotti.

www.airbus.com



Il micro-drone ispirato dal seme dell'acero

Il design relativamente semplice del quadrotor per i droni, enfatizza le prestazioni e la producibilità, ma non l'autonomia, perché quattro motori prosciugano in fretta la carica della batteria. Si pone quindi il problema dell'autonomia di volo, che può essere mitigato con batterie sempre più grandi, ma resta da risolvere il problema dell'efficienza in volo, condizionata dal peso delle batterie, molta energia delle quali verrebbe dissipata per trasportare esse stesse. Ora i ricercatori della City University di Hong Kong hanno creato un drone ispirato ai semi d'acero che pesa meno di 50 grammi ma può mantenere un volo stabile per oltre 24 minuti alimentato da una batteria da 650 mAh. I baccelli di semi d'acero, chiamati anche samaras, sono quelle cose che vediamo roteare cadendo dagli alberi di acero in autunno; sono ottimizzati per il massimo tempo di volo, grazie a un processo evolutivo che favorisce la distanza percorsa dall'albero genitore, risultando in un alto rapporto tra ala e carico utile.

I droni Samara (o monocotteri in generale) sono eccellenti alianti e possono essere dotati di un sistema di propulsione alle estremità dell'ala. Questo particolare design è simmetrico, utilizzando due ali considerevoli, ciascuna con un'elica di punta. L'elettronica, la batteria e il carico utile sono al centro e l'intero velivolo gira a circa 200 giri/min. Il bicottero è intrinsecamente stabile, con le ali che agiscono come ammortizzatori aerodinamici che si traducono in una stabilizzazione del moto passivo. Il drone ha un'apertura alare di circa 60 cm. La chiave dell'efficienza è nel fatto che le eliche non vengono utilizzate per il sostentamento, ma per far girare le ali, le quali sostentano il drone. Il controllo 3D completo si ottiene facendo pulsare le eliche in punti specifici della rotazione per andare in qualsiasi direzione.

www.scmp.com/news

Elettricità da idrogeno rinnovabile

ENERTRAG sta progettando in Uruguay il Tambor Green Hydrogen Hub, un complesso di impianti eolici e solari, abbinati ad un elettrolizzatore con sistemi di conversione per la produzione di idrogeno verde e suoi derivati.

Negli ultimi anni, l'interesse per il potenziale dell'America Latina e del Sud America per la produzione di idrogeno verde è in continuo aumento, date le sue risorse eoliche e solari di livello mondiale.

L'Uruguay ha il potenziale per produrre idrogeno su scala gigawatt. Tambor Green Hydrogen Hub, che nella sua prima fase sarà costituito da impianti di energia rinnovabile (eolico e solare fotovoltaico) con una capacità di 350 megawatt (MW) e un elettrolizzatore in loco, è progettato e sviluppato in collaborazione



con il progettista uruguayano SEG Ingeniería. Le 15.000 tonnellate di idrogeno da produrre all'anno verranno convertite in e-metanolo, che potrebbe compensare parte del metanolo prodotto conven-

zionalmente dal petrolio.

Il metanolo è una materia prima importante per l'industria chimica e può essere utilizzato come vettore energetico. Il gas di sintesi necessario per la produzione di metanolo è tradizionalmente prodotto da materie prime fossili come petrolio, gas naturale e talvolta carbone. Nel progetto Tambor verranno utilizzate solo risorse rinnovabili, pertanto, il prodotto finale sarà e-metanolo rinnovabile, la cui produzione non emette gas serra.

enertrag.com

Wafer grandi per un fotovoltaico più efficiente

Per poter produrre moduli solari più efficienti, l'industria fotovoltaica sta spostando sempre più la sua produzione verso formati di wafer più grandi. Essere in grado di garantire la produzione di queste celle solari con una lunghezza del bordo fino a 210 millimetri con qualità e velocità di ciclo costanti pone nuove sfide per i produttori di apparecchiature. Un team di ricerca del Fraunhofer Institute for Solar Energy System ISE è ora riuscito a implementare un nuovo processo in cui le celle solari di grande formato vengono continuamente lavorate mentre si muovono ad alta velocità sotto uno scanner laser, capace di 500mila cicli al secondo e ha produttività di oltre 15.000 wafer l'ora.

Il nuovo concetto Fraunhofer ISE, invece, è adatto per l'implementazione di formati di wafer di grandi dimensioni come M12 e G12, anche con elevata produttività e dimensioni ridotte della struttura. Il cuore del nuovo sistema è la tecnologia di scansione poligonale di MOEWE Optical Solutions GmbH, in cui una ruota a specchio ad alta velocità devia il raggio laser sul campione a oltre 3.000 chilometri all'ora, circa 20 volte più veloce degli scanner galvanometrici comunemente usati oggi.

Con 15.000 wafer solari all'ora, il throughput del nuovo sistema è significativamente superiore all'attuale standard industriale di 7000 all'ora, mantenendo la stessa efficienza delle celle prodotte attraverso gli attuali standard industriali.

I bassi costi operativi della lavorazione laser porteranno vantaggi in termini di costi.

www.ise.fraunhofer.de/en

Neon Ice: una promettente piattaforma qubit

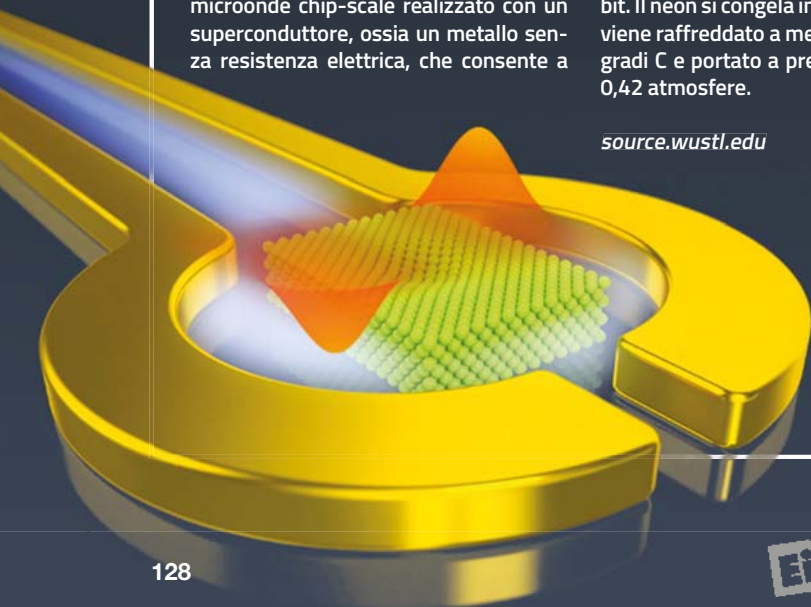
Gli elementi costitutivi fondamentali di un computer quantistico sono i bit quantistici, o qubit, i requisiti di qualità per i qubit sono estremamente elevati e per questo un team guidato dall'Argonne National Laboratory del DOE degli Stati Uniti ha creato una nuova piattaforma qubit dove gli elettroni di un filamento di luce riscaldato atterrano su neon solido, dove ciascuno è intrappolato e manipolato da un circuito quantistico superconduttore. Un componente chiave di tale piattaforma qubit è un risonatore a microonde chip-scale realizzato con un superconduttore, ossia un metallo senza resistenza elettrica, che consente a

elettroni e fotoni di interagire vicino allo zero assoluto con una perdita minima di energia o informazioni.

Il risonatore a microonde concentra l'interazione tra il qubit e il segnale a microonde, permettendo di misurare quanto bene funziona il qubit.

Per creare un qubit protetto dalle interruzioni ambientali, gli scienziati hanno sperimentato il neon, un gas nobile come l'elio che praticamente non reagisce con altri elementi, rendendolo potenzialmente un ospite ideale per un qubit. Il neon si congela in un solido quando viene raffreddato a meno di meno 248,6 gradi C e portato a pressioni superiori a 0,42 atmosfere.

source.wustl.edu





cod. FR830

€ 38,00

**PENNA CON TELECAMERA
AD ALTA RISOLUZIONE**

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



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.

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.



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.

Prezzi IVA inclusa

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



cod. PWMSOLAR30
€ 39,00

30A



cod. PWMSOLAR20
€ 22,00

20A



cod. FR699
€ 108,00

15A



cod. SOLAREG10
€ 16,00

10A

Forniscono alle batterie la quantità di corrente necessaria, proteggendole da un eccesso di carica o scarica.