

Elettronica In

www.elettronica.in.it

€ 6,90 - Anno XXVIII - n. 265 - GIUGNO 2022



Incubatrice automatica con ESP32-CAM

■ Timer Start & Stop

■ Misuriamo la velocità
del suono

■ Amplificatori audio

■ Serra domotica per
coltura idroponica

■ Comandiamo
la serranda
da smartphone

■ Autoradio hi-tech
dall'aspetto vintage

■ APP per domotica
con micro:bit

Alimentatore duale regolabile



Power Shake per DashCam



CONOSCERE
E USARE



Node-RED

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



Ci siamo

L'evoluzione della tecnologia ci porta a sperimentare nuovi componenti e piattaforme. Nel settore elettronico il progresso tecnologico consente di superare spesso i vecchi limiti.

Come Elettronica In (dove In sta per Innovativa) cerchiamo sempre di presentarvi nuove soluzioni o progetti consolidati rivisti in chiave moderna.

Ma l'innovazione della Vostra rivista non avviene solamente tra queste pagine.

Abbiamo infatti il piacere di annunciarvi il lancio del nuovo Portale online che ospiterà, oltre al blog della rivista (sia in versione Maker che in versione Pro) anche tutta una serie di servizi che vanno dalla nostra Academy, allo shop, ai progetti disponibili gratuitamente per

tutti.

Lo definiamo Portale in quanto, come riportato anche da Wikipedia, *costituisce un punto di partenza, o porta di*

ingresso, ad un gruppo consistente di risorse informative.

Ed è questo che il nuovo sito **futuranet.it** vuole diventare: il punto di riferimento per chiunque vuole avvicinarsi al mondo dell'elettronica, o vuole approfondire e migliorare le proprie conoscenze in questo settore.

Il portale futuranet.it si presenta con un layout fresco ed è ricco di nuove categorie e contenuti che non mancheranno di crescere anche grazie al vostro aiuto.

Vi aspettiamo on-line certi che questa nostra innovazione renderà ancora più In la vostra Elettronica.

Futuranet.it diventa il punto di riferimento per chi vuole avvicinarsi al mondo dell'elettronica

ON LINE

elettronica.in

SI RINGRAZIA

Futura Elettronica
IOTHINGS
Mouser
Rutronik



In copertina:
Questo mese vi presentiamo un'innovativa incubatrice automatica gestita da ESP32CAM e controllata da remoto tramite un Bot Telegram.

Boris Landoni
 boris.landoni@elettronica.in



Elettronica In

Rivista mensile, anno XXVIII n. 265

GIUGNO 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.

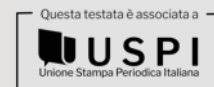
via Bettola 18 - 20092 Cinisello Balsamo (MI)

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



Audio

**Autoradio hi-tech
dall'aspetto vintage**

71



19

ARTICOLI

HOME PROJECT

19 Timer Start & Stop

Permette di attivare per un predefinito tempo, tramite il relé di bordo, un utilizzatore ed anche di disattivarlo manualmente prima che l'intervallo stabilito sia trascorso.

IN AUTO

25 DashCam Power

Circuito basato su un piccolo ATtiny85 e un sensibile sensore di movimento che permette di alimentare una dashcam in modo semplice senza dover modificare l'impianto elettrico dell'auto.

Home Project

Timer Start & Stop



19

In Auto

DashCam Power



25

Misure

Misurare la velocità del suono in aria



39

Hi-Tech

Serra domotica per coltura idroponica



47

Networking**Comandiamo la serranda da smartphone****63****Domotica****Smart Home application****89****Didattica****Amplificatori Audio****105****39****MISURE****Misurare la velocità del suono in aria**

Utilizziamo un economico modulo ad ultrasuoni per misurare la velocità del suono con una scheda Arduino UNO.

83**Laboratorio****Alimentatore duale e regolabile****Didattica****Conoscere e usare Node-RED****117****47****HI-TECH****Serra domotica per coltura idroponica**

Costruiamo una serra domotica (comandata da Blynk e relativa app) con illuminazione artificiale a LED specifici per ottimizzare la crescita delle piante contenute, specie utilizzando la tecnica idroponica. Seconda e ultima puntata.

LABORATORIO**83****Alimentatore stabilizzato duale e regolabile**

Con l'aggiunta di un trasformatore dotato di presa centrale, consente di fornire tensioni stabilizzate simmetriche da 3 a 15 Vcc e correnti di 1 A. Ideale come apparato da banco, può costituire lo stadio di alimentazione di amplificatori BF.

54**STRUMENTAZIONE****Incubatrice automatica con ESP32-CAM**

Incubatrice per uova di gallina completamente automatizzata con controllo di temperatura, umidità e rotazione delle uova; il sistema è monitorabile tramite un bot Telegram con l'invio di messaggi e immagini.

89**DOMOTICA****Smart Home application**

Realizziamo l'APP con Android Studio, per la gestione e il controllo del sistema di domotica micro:bit.

63**NETWORKING****Comandiamo la serranda da smartphone**

Un progetto smart per poter aprire e chiudere comodamente dallo smartphone la serranda di casa, con possibilità di ricevere e-mail sullo stato.

105**DIDATTICA****Amplificatori audio dall'aspetto vintage**

Con prove di laboratorio andiamo a verificare i principali parametri di funzionamento di alcuni diffusi e pratici amplificatori audio in classe AB e D. Seconda puntata.

71**AUDIO****Autoradio Hi-Tech dall'aspetto vintage**

Riproponiamo in chiave moderna le affascinanti autoradio primordiali che equipaggiavano le auto di una volta. Seconda e ultima puntata.

117**DIDATTICA****Conoscere e usare Node-RED**

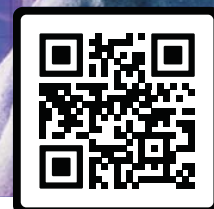
Impariamo ad usare Node-RED, un tool di flow-based programming orientato all'IoT ed alla connettività, originariamente sviluppato dall'IBM Emerging Technology Services team e adesso parte della JS Foundation. In questa puntata parliamo dell'integrazione degli assistenti vocali con Node-RED. Ultima Puntata.

IOT THINGS WORLD

CONFERENCE | EXHIBITION | MEETINGS | NETWORKING



15-16 GIUGNO 2022
MEDIOLANUM FORUM + ON-LINE



SCAN ME

DAL 2002 L'EVENTO ITALIANO PIÙ IMPORTANTE DEDICATO ALLE TECNOLOGIE E SOLUZIONI IOT

IOT THINGS

Scenari evolutivi, Digital Transformation, IOT Data Analytics, Industry 4.0, IoT Networks nella conferenza da sempre leader su M2M e IoT

IOENERGY

Efficienza energetica ma non solo nella conferenza che esplora l'aspetto sistemico di Internet of Energy

ITALIA5G

Progetti innovativi abilitati dalle reti 5G per accelerare la trasformazione digitale del sistema-Paese e la diffusione di applicazioni inedite

EDGE FORUM

Opportunità e vincoli delle architetture di edge computing nel panorama dell'evoluzione tecnologica e applicativa di IoT

IOT ANALYTICS

Processare in maniera intelligente i Big Data ma anche effettuare analisi sofisticate in real-time

CYBER IOT

Area parte di un percorso multi-evento dedicato ad esplorare il tema della cybersecurity da tutte le sue angolazioni

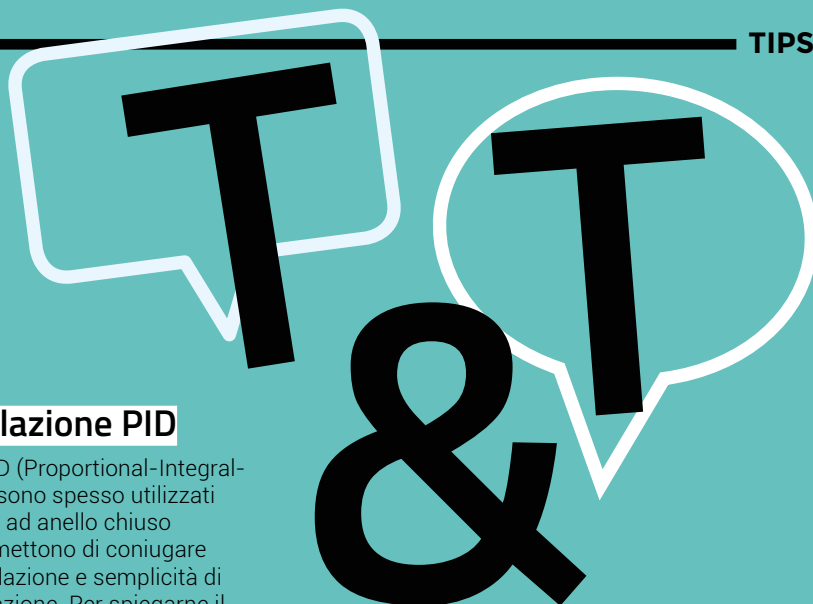
L'ACCESSO ALL'EVENTO È GRATUITO CON PRE-REGISTRAZIONE
WWW.IOT THINGS.WORLD

Brought to you by



Matching Powered by





La regolazione PID

I sistemi PID (Proportional-Integral-Derivative) sono spesso utilizzati nei controlli ad anello chiuso perchè permettono di coniugare buona regolazione e semplicità di implementazione. Per spiegarne il funzionamento e farvene apprezzare la validità confrontiamoli con la regolazione on/off in una tipica applicazione: il mantenimento della temperatura di un riscaldatore a resistenza. La soluzione più semplice consiste nell'applicare un termistore, collegarlo ad un comparatore e confrontare la tensione fornita con una di riferimento: se è più bassa del richiesto si alimenta la resistenza, se è più alta la si spegne. Con questa regolazione on/off la temperatura non si stabilizza mai al punto richiesto, ma continua ad oscillare tra un valore superiore ed uno inferiore; questo perchè il riscaldatore e la resistenza stessa hanno un'inerzia termica non indifferente, e rispondono in ritardo alle variazioni di potenza. Se per una resistenza il problema è relativo, per il posizionamento dei motori diventa rilevante: ad esempio per uno stepper-motor si può avere un'ondeggiamento intorno a una posizione, invece che una posizione stabile. Qualcosa si può migliorare con il Controllo Proporzionale, dove più il parametro da regolare è lontano dal punto desiderato e più potenza si fornisce, riducendola man-mano che ci si avvicina al punto desiderato. Tuttavia anche in questo caso l'inerzia disturba e il fatto di annullare la corrente una volta raggiunto il valore di destinazione implica che quest'ultimo non verrà mai raggiunto. Qui entra in gioco il sistema PID: abbiamo visto che il sistema Proporzionale (P) agisce in base all'errore tra la posizione richiesta e quella corrente ma non tiene conto di come avviene la variazione della

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.

posizione o del valore; se ci stiamo avvicinando rapidamente o lentamente alla posizione desiderata è influente, l'unica cosa considerata è la differenza tra l'attuale e la posizione finale.

Nel caso di un motore, è facile intuire che se stava girando velocemente, l'inerzia meccanica tenderà a far superare la posizione richiesta anche se gli si toglie corrente al momento giusto, mentre se girerà lento, si impiegherà molto tempo a raggiungere la posizione di destinazione.

Qui interviene la parte Derivative (D), che in italiano può tradursi in Differenziale.

Questo fattore tiene in conto della rapidità con cui diminuisce l'errore nel tempo: se ci stiamo avvicinando rapidamente, la diminuzione è rapida e quindi sappiamo di dover "frenare prima". Inserendo questo parametro possiamo compensare quasi totalmente i problemi dovuti all'inerzia meccanica dei motori o termica dei riscaldatori, per esempio.

La parte Integrata (I) del sistema è un po' più complessa da spiegare, perchè tiene conto della somma degli errori nel tempo ed è in grado, ad esempio, di risolvere il problema dell'annullamento della corrente

nel motore al raggiungimento della posizione richiesta. Facendo un esempio semplificato, se il motore raggiunge la posizione richiesta, senza la componente integrale la potenza va a zero, mentre con la componente integrale il sistema "ricorda" quanta fatica ha fatto a raggiungerla e compenserà aumentando la potenza fornita.

A questo punto vediamo come mettere insieme i tre sistemi; in teoria è sufficiente:

- memorizzare l'errore corrente di posizione (parte P);
- memorizzare la differenza tra l'errore corrente e quello precedente (parte D);
- mantenere una somma degli errori man mano che procediamo (parte I);
- mettere i numeri risultanti in una formula appositamente calibrata che restituisce il valore di potenza da applicare ai motori o riscaldatori e in generale agli attuatori del sistema di regolazione.

Nella pratica bisogna affrontare ancora alcuni problemi pratici e in particolare occorre lavorare con numeri interi, perchè quelli a virgola mobile rallentano i calcoli e l'intervento della regolazione e inoltre è necessario dare un limite ai valori misurati, giacché non è pensabile applicare all'attuatore una potenza infinita. Nella componente Integrata del PID è inutile, anzi, dannoso, memorizzare



una storia degli errori troppo "vecchia" perché è opportuno acquisirla dinamicamente.

In definitiva, è possibile scrivere una formula, che è semplice, anche se la determinazione dei relativi parametri non lo è affatto:

$$P = C_p \cdot p + C_d \cdot d + C_i \cdot i$$

Dove C_p , C_d e C_i sono tre costanti da determinare, p è l'errore sulla posizione, d è la sua variazione nel tempo, i è la "somma" degli errori nel tempo; P è la potenza da applicare all'attuatore della regolazione.

La determinazione delle costanti non è facile e non lo è nemmeno la gestione della parte integrale degli errori; per le costanti, esistono dei metodi più o meno empirici, mentre la seconda parte dipende molto dalla dimensione dei numeri interi nel controller, dalla tipologia del sistema da regolare e via di seguito.

Come funziona la sonda Lambda

Ormai tutte le automobili a motore tradizionale sono piene di elettronica e parte di questa è finalizzata al controllo dei parametri di combustione per limitare l'impatto ambientale, ridurre i consumi e migliorare l'efficienza. Uno dei primi sensori adottati e probabilmente quello cruciale è la "sonda lambda" la

quale basa su una misurazione comparativa dell'ossigeno residuo nei gas di scarico e presente nell'aria ambiente. Esistono due tipi di sonda lambda: quella a caduta di tensione e quella a variazione resistiva; la prima è costituita da un elemento in ceramica elettrolitica in biossido di zirconio cavo a forma di dito; tale elettrolito solido è penetrabile per ioni ossigeno a una temperatura superiore a 300°C. Entrambi i lati della ceramica sono rivestiti con un sottile strato poroso di platino che funge da elettrodo. Le emissioni di scarico scorrono oltre l'esterno dell'elemento ceramico, mentre l'interno è riempito con aria di riferimento. Le proprietà dell'elemento ceramico fanno sì che la diversa concentrazione di ossigeno su entrambi i lati provochi una migrazione di ioni di ossigeno, che a sua volta genera una tensione. Questa tensione viene amplificata e utilizzata come segnale per la ECU, che modifica il rapporto aria-carburante in base al contenuto di ossigeno residuo nei gas di scarico. La misurazione del contenuto di ossigeno residuo viene ripetuta più volte al secondo.

Invece per la sonda a variazione resistiva l'elemento ceramico è realizzato in biossido di titanio utilizzando la tecnologia a film spesso multistrato; il biossido di titanio cambia la sua resistenza in proporzione alla concentrazione di ossigeno. Con un contenuto di ossigeno maggiore (miscela magra $\lambda > 1$) è meno conduttivo, con un contenuto di ossigeno inferiore (miscela ricca $\lambda < 1$) diventa

più conduttivo. La sonda siffatta non necessita di aria di riferimento, ma deve essere alimentata con una tensione di 5 V dalla ECU tramite una combinazione di resistenze. Il segnale richiesto per l'unità di controllo è generato dalla caduta di tensione sui resistori.

Le moderne sonde lambda sono dotate di riscaldatore del sensore, il che permette di installarle lontano dal motore, così da non esporle all'elevato carico termico. I sensori lambda riscaldati hanno un tempo di risposta più breve, che rende più pronta la regolazione ad anello chiuso.

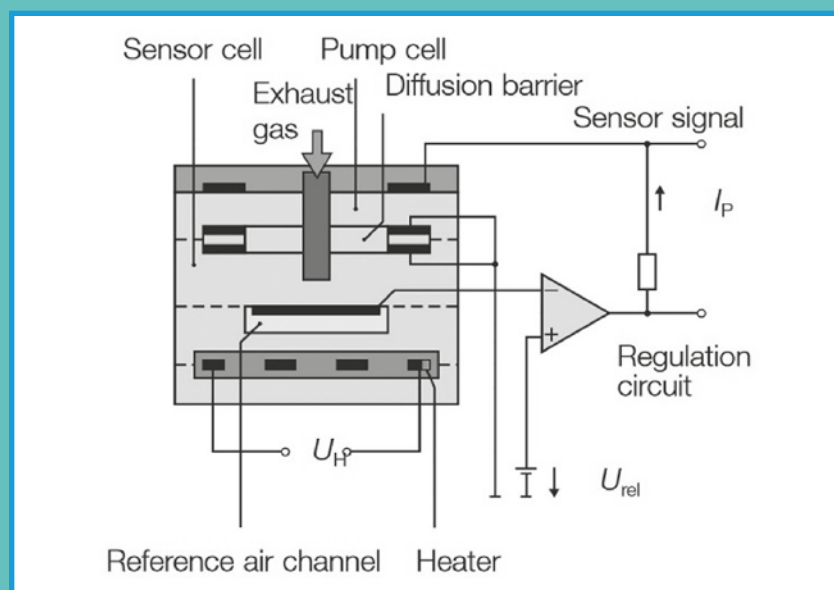
Esistono infine sonde lambda a banda larga che permettono di misurare un rapporto preciso dell'aria sia nell'intervallo magro ($\lambda > 1$) che ricco ($\lambda < 1$). Come una sonda convenzionale, il sensore lambda a banda larga è costruito con aria di riferimento. Ha anche una seconda cella elettrochimica: la cella della pompa (riferitevi alla figura in queste pagine).

Le emissioni di scarico entrano nell'area di misurazione attraverso un piccolo foro nella cella della pompa, noto come gap di diffusione. Per impostare il rapporto dell'aria (λ), qui la concentrazione di ossigeno viene confrontata con la concentrazione di ossigeno dell'aria di riferimento.

Per ricevere un segnale misurabile per l'unità di controllo, viene applicata una tensione alla cella della pompa. Con questa tensione, l'ossigeno può essere pompato dalle emissioni di scarico all'interno o all'esterno del gap di diffusione. L'unità di controllo regola la tensione della pompa in modo tale che il rapporto del gas sia costantemente a $\lambda = 1$ nel traferro di diffusione. Se la miscela è magra, l'ossigeno viene pompato verso l'esterno dalla cella della pompa. Ne risulta una corrente di pompa positiva. Se la miscela è ricca, l'ossigeno viene pompato verso l'interno dall'aria di riferimento. Il risultato è una corrente negativa della pompa. A $\lambda = 1$ nel gap di diffusione, non viene trasportato ossigeno; la corrente della pompa è zero.

I supercondensatori: questi conosciuti

Si tratta di condensatori elettrici che hanno elevatissime capacità (decine di farad) e che possono quindi, sottoposti a tensioni elevate, accumulare ingenti quantità di carica elettrica.



↑ Schematizzazione del funzionamento della sonda lambda riscaldata a banda larga.

Il supercondensatore più usato è l'ECDL, che si basa sul principio del doppio strato elettrico; questo consiste in una struttura composta da due elettrodi immersi in un liquido elettrolitico dove il trasferimento delle cariche avviene su entrambi gli elettrodi, uno con ioni positivi e l'altro con ioni negativi. I due elettrodi sono solitamente in alluminio, rivestiti con carbonio attivo o nanotubi di carbonio per avere un'ampia superficie di scambio con l'elettrolito; un separatore immerso nell'elettrolito evita il cortocircuito degli elettrodi ma permette il passaggio degli ioni. Il rivestimento in nanotubi di carbonio sembra essere la tecnologia d'elezione perché conferisce elevata conducibilità; i nanotubi sono strutture cristalline del carbonio ottenute solitamente per fusione di due elettrodi di grafite con un arco elettrico da cui poi vengono rimosse le impurezze.

La loro struttura cristallina è simile a quelle del fullerene e del grafene, che sono dei microaggregati di atomi di carbonio, come per il diamante, però il nanotubo di carbonio presenta una superficie piana composta sempre da microaggregati di atomi di forma esagonale che si avvolge però per rilassamento su se stessa, formando superfici cilindriche di diametro dell'ordine di 2,2 nm e di lunghezza molto maggiore rispetto al diametro. È possibile ottenere diverse strutture di nanotubi a seconda dell'orientazione degli esagoni sulla superficie cilindrica ma soprattutto è possibile ottenere nanotubi di tipo SWNT (Single Wall Nano-Tube) che hanno la semplice forma a cilindro oppure MWNT (Multi Wall NanoTube) cioè più cilindri concentrici distanti 0,34 nm l'uno dall'altro, disposti a formare un unico nanotubo.

Gli MWNT solitamente hanno maggiore superficie porosa che si traduce in maggiore capacità per i supercondensatori. La capacità dei supercondensatori a elettrodo in nanotubi di carbonio varia ed è stabile a basse frequenze mentre all'aumentare delle frequenze diminuisce e aumenta la ESR.

La tensione ai capi degli elettrodi dei supercondensatori dipende dal dielettrico interposto e quindi dall'elettrolito usato. Gli elettroliti possono essere in soluzione acquosa come H_2SO_4 KOH con alta conduttività (1 S/cm) ma basse tensioni (1V) oppure elettroliti organici come $(\text{C}_2\text{H}_5)_4\text{NBF}_4$ con tensione fino a 3,5V ma bassa conducibilità (0,015 S/cm). L'elettrolito più diffuso è una soluzione di sali di tetraetilammonio in



↑ Una batteria di supercondensatori con relativo circuito di carica fra un elemento e il successivo.

acetonitrile (CH_3CN).

Le maggiori densità di energia si ottengono con elettroliti in soluzione acquosa. È quindi importante sia la scelta dell'elettrodo che dell'elettrolito. Quanto al separatore degli elettrodi, ne esistono sostanzialmente due tipi: separatori in polimero e fibra di cellulosa, usati per gli elettroliti organici, e separatori in fibra di vetro o ceramici, utilizzati per elettroliti in soluzione acquosa.

Oggigiorno si realizzano batterie di supercondensatori (immagine in queste pagine) in serie e parallelo capaci di accumulare grande energia e bilanciare la quantità di carica fra i vari elementi.

Il bus I²S

Noto anche come Inter-IC Sound, Integrated Interchip Sound, o IIS, è un bus seriale usato per interconnettere dispositivi audio digitali, trasmettendo dati audio in PCM tra integrati, tipicamente all'interno di uno stesso dispositivo elettronico.

Nei lettori CD-DVD può interconnettere la testina di lettura al DAC.

Nella versione base, è formato da tre linee: clock, word select (WS) e linea dati multiplexata; a queste possono aggiungersi un Master clock e una linea dati multiplexata per il feedback.

Il clock commuta per ogni elemento di scarto di dati sulle linee dati e la relativa frequenza è il prodotto della frequenza di campionamento per numero di bit per canale per il numero di canali. Il clock di selezione parola (word select clock) permette al dispositivo di sapere se in un dato momento viene trasmesso il canale 1 o il canale 2, dato che i dati sono multiplexati.

Le specifiche I²S prevedono che il canale audio sinistro sia trasmesso durante il livello basso del clock word mentre il destro sia trasmesso sul livello alto.

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



1	VCC	OUT	8
2	IN	S/H	7
3	COMMON	NC 1	6
4	NC	VEE	5

1	1	2	2
3	3	4	4
5	5	6	6
7	7	8	8
9	9	10	10
11	11	12	12
13	13	14	14
15	15	16	16
17	17	18	18
19	19	20	20
21	21	22	22
23	23	24	24
25	25	26	26
27	27	28	28

Scarica GRATUITAMENTE

- Simboli schematici
- Footprint su PCB
- Modelli 3D



mouser.it/ecad



MOUSER
ELECTRONICS

iRobot non smette di sorprendere e propone un robot aspirapolvere e lavapavimenti, che pur essendo strutturato come elettrodomestico è una vera e propria piattaforma di programmazione espandibile e configurabile per sperimentare l'implementazione di nuove funzioni e personalizzare il robot.

Create 3, così si chiama, è basato su una piattaforma Roomba compatibile con ROS 2, il software che moltissimi robot utilizzano. Create 3 è dotato di WiFi, host Ethernet su USB e Bluetooth, nonché di una IMU, un sensore

› Il robot domestico insegna a programmare



ottico di rilevamento del pavimento, encoder per le ruote e sensori IR per applicazioni di localizzazione, navigazione e telepresenza.

Inoltre, il robot include il rilevamento di dislivelli, urti e scivolate, luci a LED e un altoparlante. ROS 2 consente anche a maker e studenti di accelerare lo sviluppo dei loro progetti concentrandosi sull'applicazione, piuttosto che sulla piattaforma. Create 3 supporta anche Python e Ignition Gazebo e si può programmare in ROS 2, con Python SDK o con l'app di iRobot.

<https://edu.irobot.com>

› Test cloud-based di software ADAS

Bosch acquisce Five, leader nel campo della guida automatizzata, che svolgerà un ruolo chiave nella divisione Bosch Cross-Domain Computing Solutions, il principale player SAE Level 4 d'Europa, grazie ad un team di esperti in software cloud, robotica e apprendimento automatico, all'avanguardia nello sviluppo di software e soluzioni basate sull'intelligenza artificiale per l'ADAS di livello 4. La startup ora si concentra principalmente su una piattaforma di sviluppo e test basata su cloud per il software delle auto a guida autonoma.

La piattaforma analizza dati reali da una flotta di veicoli di prova, creare scenari di test avanzati e costruire un ambiente di simulazione che consente di valutare e validare il comportamento del sistema su scala iperattiva.

Bosch continua a sviluppare solu-

zioni per veicoli privati con riguardo per l'ADAS e i sistemi parzialmente e condizionalmente automatizzati (Livelli SAE da 1 a 3), lavorando anche a soluzioni per livelli di automazione più elevati, con particolare attenzione ai veicoli della flotta e ai nuovi

modelli operativi; a livello 4, Bosch ha sviluppato il servizio di parcheggio e riconsegna auto automatizzato, la prima funzione di guida pronta per la produzione che non richiede conducente.

www.bosch-presse.de



› Il sistema Smart Home IoT che rispetta la privacy

Rilevando l'attività umana e regolando di conseguenza le impostazioni ambientali, i sistemi di casa intelligente potrebbero aiutare a creare edifici più efficienti dal punto di vista energetico e sostenibili; tuttavia si sono posti problemi di privacy quando questi sistemi che monitorano l'attività delle persone e i sistemi di casa devono gestire dati riconducibili alle persone, in maniera non sicura. Un nuovo sistema intelligente, soprannominato Chameleon, nasce per risolvere questo problema; arriva dal Media Lab del MIT ed è basato sull'anidride carbonica passiva e sui sensori a infrarossi per gestire la smart-home nel rispetto delle privacy. Il sistema è stato recentemente testato in due diversi ambienti nel corso

di un mese e potrebbe prevedere l'attività umana con un'accuratezza dell'87-99% dopo solo una settimana di addestramento.

Il segreto sta nell'aver costruito il sistema attorno ai sensori di anidride carbonica e infrarossi passivi (PIR) gestiti da un microcontrollore dotato di transceiver WiFi. I dati sull'anidride carbonica e dei PIR non sono intrinsecamente invadenti, mentre i sistemi basati su telecamere o altri sistemi ottici possono sollevare problemi di privacy.

L'anidride carbonica e il rilevamento a infrarossi potrebbero essere utili per stimare una varietà di scenari, come il numero di persone in una stanza e quanto respirano pesantemente,

› AI + Quantum Computing in un Memristor

Negli ultimi anni l'informatica si è mossa in due direzioni: lo sviluppo di algoritmi che migliorano automaticamente attraverso l'esperienza e ricerca sui computer quantistici. Le due potrebbero confluire in un dispositivo noto come memristor quantistico, combinando l'intelligenza artificiale con il calcolo quantistico.

Un memristor, o resistore di memoria, è una sorta di mattone per i circuiti elettronici che gli scienziati avevano previsto

da circa mezzo secolo, ma che hanno creato per la prima volta solo poco più di un decennio fa. Questi componenti sono essenzialmente interruttori elettrici che possono ricordare se sono stati attivati o disattivati dopo lo spegnimento. In quanto tali, assomigliano a sinapsi, i collegamenti tra i neuroni nel cervello umano, la cui conduttività elettrica si rafforza o si indebolisce a seconda di quanta carica elettrica li ha attraversati in passato.

In teoria, i memristori possono agire come neuroni artificiali in grado sia di calcolare che di archiviare dati e i computer neuromorfici costruiti utilizzando i memristori eseguirebbero in maniera ottimale le reti neurali, che sono sistemi di apprendimento automatico che utilizzano versioni sintetiche di sinapsi e neuroni per imitare il processo di apprendimento nel cervello umano.

Mentre i classici computer accendono o spengono i transistor per creare i livelli 1 e 0, i computer quantistici utilizzano bit quantistici (qubit), che possono trovarsi in uno stato di sovrapposizione, tanto da essere 1 e 0 contemporaneamente.

Più qubit sono collegati tra loro in un computer quantistico, maggiore è la sua potenza di calcolo.

Normalmente, il comportamento memristivo e gli effetti quantistici non dovrebbero coesistere, perché i memristori funzionano misurando i dati che fluiscono al loro interno, ma gli effetti quantistici sono notoriamente vulnerabili a qualsiasi interferenza esterna come la misurazione.

I ricercatori hanno superato questa apparente contraddizione creando interazioni all'interno del dispositivo, abbastanza forti da consentire la memristività, ma abbastanza deboli da preservare il comportamento quantistico.

<https://medienportal.univie.ac.at>

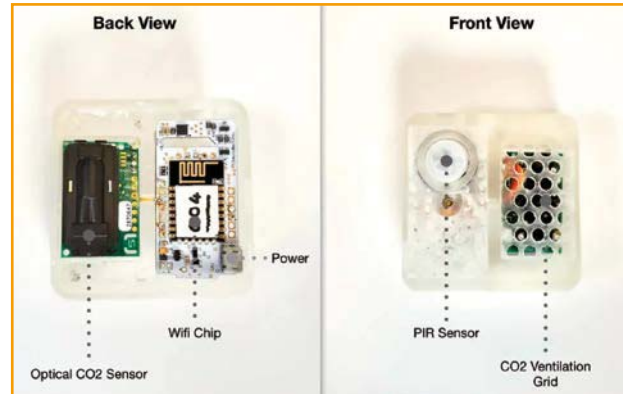


che possono indicare stati di riposo o attivi. Ogni scenario ha firme uniche di anidride carbonica e infrarossi che possono essere analizzate utilizzando l'apprendimento automatico.

Il sistema di casa intelligente Chameleon utilizza algoritmi di apprendimento automatico sia supervisionati che non supervisionati, pertanto parte con alcune conoscenze generali su come valutare qualsiasi ambiente, ma può anche imparare a distinguere le variabili uniche del suo nuovo ambiente; tale combinazione può ridurre i tempi e i costi associati alla calibrazione e alla manutenzione del sistema.

Chameleon è stato messo alla prova dai ricercatori del MIT in due ambienti: un ufficio con pochi dipendenti e un'aula con 20 occupanti.

www.media.mit.edu



› L'IoT cellulare controlla le autocisterne



Un nuovo sistema di sensori intelligenti sviluppato dalla startup tedesca BeST SENSOR consente il monitoraggio delle autocisterne per le acque reflue: BeST Sensor Pump Monitor JROV2201 monitora l'utilizzo della pompa delle acque reflue e fornisce una panoramica in tempo reale dello stato delle pompe collegate, nonché un log degli eventi di flusso.

Il prodotto comprende un sensore di temperatura, umidità e pressione, oltre a un accelerometro e un rilevatore di orientamento; montato su una tubazione tra pompa e serbatoio da riempire o svuotare, il sistema può essere utilizzato per monitorare e registrare il processo di pompaggio e valutare la qualità della procedura utilizzando un algoritmo proprietario

che esegue il riconoscimento del modello.

Una valutazione codificata dello stato della pompa viene trasmessa in modalità wireless tramite l'IoT cellulare LTE-M/NB-IoT abilitato per SiP nRF9160 di Nordic a un broker MQTT connesso, che a sua volta trasmette il messaggio a un server cloud per ulteriori elaborazioni e rapporti.

Il SiP fornisce un sistema operativo in tempo reale (RTOS) con multi-tasking, ampio supporto con software driver per componenti aggiuntivi come sensori e LED e componenti open source. Il sistema comprende un monitor di pompa singolo o una combinazione di diverse centinaia di unità con copertura LTE-M o NB-IoT in tutta l'Europa centrale. L'edge computing viene eseguito sul SiP nRF9160 per analizzare il flusso di smaltimento, con le informazioni quindi trasmesse a una piattaforma cloud tramite LTE-M.

Oltre alla connettività LTE-M e NB-IoT, le caratteristiche chiave del SiP nRF9160 sono il consumo energetico ultra-basso in modalità standby con riattivazione immediata dopo l'interruzione, memoria e possibilità di

integrare l'elaborazione del segnale e algoritmi di riconoscimento del modello in tempo reale e l'integrazione di interfacce di bordo come I²C-Bus.

Dal punto di vista del software, il SiP fornisce un sistema operativo in tempo reale (RTOS) con multi-tasking, ampio supporto con software driver per componenti aggiuntivi come sensori e LED e componenti open source. L'ambiente di sviluppo è disponibile per Linux, macOS e Windows.

Il processore applicativo dedicato Arm Cortex-M33 a 64 MHz del nRF9160 SiP fornisce una potenza di calcolo sufficiente non solo per supervisionare la connettività cellulare del modem LTE-M/NB-IoT, ma anche tutte le altre funzionalità del prodotto. La flash da 1 MB di SiP e la RAM da 256 kB supportano una risposta rapida e un software applicativo complesso.

Il BeST Sensor Pump Monitor funziona senza manutenzione per un massimo di tre mesi prima che le sue sei batterie AA Ni-MH da 7,2 V, 2000 mAh, AA Ni-MH debbano essere ricaricate o sostituite.

www.nordicsemi.com

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.

Controller B&R certificato TÜV SÜD Rail

TÜV SÜD Rail ha certificato il sistema X90 di B&R per l'uso in applicazioni ferroviarie. In conformità con le norme EN 50155, EN 50657 e EN 45545-2, tale controller può essere utilizzato per applicazioni come l'illuminazione e il controllo della temperatura.

I prodotti X90 offrono una resistenza particolarmente elevata a condizioni ambientali difficili e soddisfano elevati requisiti di compatibilità elettromagnetica. La certificazione TÜV SÜD Rail lo conferma, oltre a verificare l'elevata qualità del software B&R e un elevato grado di protezione antincendio.

B&R ha progettato la sua serie X90 con la protezione IP69K, grazie alla quale i dispositivi possono funzionare in un intervallo di temperatura da -40°C a +85°C e sono resistenti agli urti e alle vibrazioni.

B&R è un'innovativa azienda di auto-



mazione, diventata una business unit del Gruppo ABB. B&R combina la tecnologia all'avanguardia con l'ingegneria avanzata per fornire ai clienti di quasi tutti i settori soluzioni complete per

l'automazione di macchine e di fabbrica, il controllo del movimento, l'interfaccia operatore e la tecnologia di sicurezza integrata.

www.br-automation.com

Da Toshiba, nuove MCU ARM Cortex-M3



Toshiba ha iniziato a produrre nuovi microcontrollori M3H della famiglia TX2+ Advanced Class, con processo a 40 nm; basati su core ARM Cortex-M3 con clock fino a 120 MHz, codice Flash integrato da 512 kB e Flash dati da 32 kB con durata del ciclo di scrittura di 100 kB e 64 kB di RAM, hanno funzioni di comunicazione avanzate integrate come UART, TSPI, I²C e un controller DMA a 2 unità (DMAC) oltre ad un driver LCD.

Questi dispositivi suppor-

tano una varietà di applicazioni di rilevamento con un convertitore analogico/digitale (ADC) a 12 bit ad alta velocità e precisione che consente di impostare tempi di campionamento e mantenimento individuali per ciascuno dei canali di ingresso da 12 a 21 ADC. Inoltre, un driver programmabile (A-PMD) on chip sincronizzabile con l'ADC a 12 bit, permette la gestione di motori CA e CC brushless. Nelle MCU troviamo anche un DAC a doppio canale a 8 bit e fino a 135 porte di ingresso/uscita (I/O) dedicate.

I nuovi micro funzionano con una tensione compresa tra 2,7 e 5,5 VCC e sono alloggiati in contenitori LQFP ad alta densità da 64 a 144 pin.

www.toshiba.com

Buffer e Multiplexer per PCIe Gen6

Renesas presenta i primi clock buffer e multiplexer conformi PCIe Gen6: 11 nuovi clock buffer e 4 multiplexer, che supportano le implementazioni PCIe Gen5, si affiancano ai generatori di clock a basso jitter di Renesas 9SQ440, 9FGV1002 e 9FGV1006 per applicazioni data center/cloud computing, networking e per applicazioni industriali ad alta velocità. Lo standard PCIe Gen6 supporta velocità dati fino a 64 GT/s, richiedendo livelli del jitter di clock inferiori a 100 fs RMS. I nuovi clock buffer RC190xx ed i multiplexer RC192xx di Renesas hanno specifiche di jitter additivo PCIe Gen6 di soli 4fs RMS, il che li rende praticamente ininfluenti nella catena di temporizzazione e quindi utilizzabili dai clienti anche in futuri progetti conformi con le prossime generazioni di standard del set-

tore. Tra le caratteristiche dei Clock Buffer e dei Multiplexer PCIe Gen6, jitter additivo PCIe Gen6 di 4 fs, ritardo in-out di 1,4 ns, skew in uscita di 35ps e PSRR di -80 dB a 100KHz, indirizzi SMBus selezionabili, uscita Loss-Of-Signal (LOS) per il monitoraggio e la ridondanza del sistema, interfaccia Side-Band a 4 fili per l'abilitazione/disabilitazione dell'uscita seriale ad alta velocità ed il collegamento in daisy chain. I buffer RC190xx sono offerti in configurazioni a 4, 8, 13, 16, 20 e 24 uscite. I multiplexer RC192xx includono versioni a 2, 4, 8 e 16 uscite.

www.renesas.com/pcietiming



Da Microchip, Flash seriale a 64 Mb resistente alle radiazioni



Le nuove memorie RT della famiglia SuperFlash hanno tolleranza della dose ionizzante totale (TID) di ben 50 kilorad (krad). La SuperFlash S5T26L-F064RT è una quad I/O NOR Flash a 64 Mbit per impiego in ambienti aerospaziali e di difesa, offre le migliori prestazioni TID sul mercato con una soluzione di memoria quad I/O

seriale da 64 Mbit che funziona insieme a qualsiasi FPGA basato su SRAM in diverse applicazioni spaziali. È l'ideale per i sistemi utilizzati nelle costellazioni Low Earth Orbit (LEO) e in altri ambienti con elevato livello di radiazioni. I prodotti Microchip SuperFlash sono integrati da RT FPGA (RTG4 e RT PolarFire) e dal microprocesso-

re (MPU) SAMRH71 rad-hard e dal microcontrollore (MCU) SAMRH707. Microchip seleziona i dispositivi più rilevanti dalla sua famiglia di prodotti qualificati per il settore automobilistico o industriale e aggiunge miglioramenti nel processo del silicio per renderli più resistenti alle radiazioni.

www.microchip.com

Da Keysight, software di debug CAN

Keysight Technologies ha rilasciato la prima soluzione di trigger/decodifica di protocolli automobilistici basata su oscilloscopio (D9010AUTP) che copre CAN XL (Controller Area Network eXtra Long) e che consente agli ingegneri di verificare ed eseguire il debug dei protocolli bus seriali automobilistici a bassa velocità, semplificare lo sviluppo e la risoluzione dei problemi di sistemi, inclusi CAN/CAN FD (Flexible Data Rate) e CAN XL.

CAN XL è lo standard CAN di terza generazione sviluppato dal gruppo CAN in Automation (CiA) che migliora il precedente protocollo CAN FD con la possibilità di inviare un maggior numero di dati in maniera più veloce. Nel bus CAN XL la velocità supera i 10 Mbps, a seconda delle capacità del ricetrasmittitore e dei componenti del livello fisico; inoltre il CAN XL ha due modalità di funzionamento: modalità veloce e modalità SIC (lenta). Il software del decodificatore automobilistico di Keysight attiva e decodifica correttamente il segnale in modalità SIC nella fase di arbitraggio, nonché il segnale in modali-



tà veloce nella fase dati. Gestisce anche i protocolli CAN XL implementati con i ricetrasmittitori CAN HS/FD/SIC. Il software D9010AUTP di Keysight è compatibile con gli oscilloscopi Infiniium di Keysight

che utilizzano la versione software 11.30 o successiva per i modelli MXR/EXR/UXR e la versione 6.72 o successiva per altri modelli di oscilloscopi Infiniium.

www.keysight.com

STM combina Hi-F e classe G

SSTM presenta l'amplificatore audio per impiego automobilistico TDA7901 che integra un controller buck per la commutazione dell'alimentazione a supporto dell'audio ad alta definizione in Classe G. Gli alimentatori in classe G, quando il segnale di uscita raggiunge il livello di clipping, aumentano il valore della tensione

di alimentazione in modo da abbassare la distorsione ed aumentare il range dinamico. Il rendimento degli amplificatori in classe G è dell'ordine dell'85%. Il regolatore buck del TDA7901 ottimizza la tensione fornita allo stadio di potenza BTL (bridge-tied load) a seconda del livello del segnale audio. Il suono analogico che

ne risulta ha un'efficienza vicina alla classe D a livelli di ascolto normali. Con il suo ingresso digitale I²S e l'uscita quad-BTL smart, il TDA7901 offre 4 x 43 W di potenza audio (a 4 Ω di carico, uscita saturata a 14,4 V). L'amplificatore ha una larghezza di banda di 80 kHz.

www.st.com



Snapdragon Digital Chassis: tecnologia Qualcomm per l'auto del futuro



Qualcomm annuncia l'utilizzo del suo Snapdragon Digital Chassis nei 14 marchi automobilistici di Stellantis, a partire dal 2024. Sfruttando le piattaforme Snapdragon Cockpit e le capacità 5G per i sistemi telematici, Stellantis avrà la capacità di offrire aggiornamenti OTA per tutta la vita di un veicolo e di soddisfare le aspettative in evoluzione dei clienti per esperienze personalizzate e all'avanguardia che sono continuamente aggiornabili.

L'accordo faciliterà il piano di Stellantis di unire tutti i domini software in computer ad alte prestazioni, sfruttando le

piattaforme automobilistiche Snapdragon ad alte prestazioni e bassa potenza in tutti i principali domini dei veicoli e contribuirà a proteggere la catena di approvvigionamento di Stellantis su componenti strategici. Stellantis afferma che utilizzerà le piattaforme Snapdragon Cockpit per alimentare i sistemi di comunicazione e infotainment in auto per STLA SmartCockpit, che è stato progettato e progettato insieme ad Amazon e Foxconn. La prima applicazione sarà nel marchio Maserati.

Le piattaforme Snapdragon Cockpit verranno utilizzate anche per migliorare STLA Brain, contribuendo a migliorare le capacità dell'assistente personale di bordo con funzionalità di intelligenza artificiale (AI) intuitive.

www.qualcomm.com

Ricarica Wireless per le soluzioni Wacom Active ES Pen

Renesas integra la sua tecnologia di ricarica wireless nelle soluzioni Active ES Pen di Wacom; il circuito integrato del ricevitore di potenza wireless single chip ha dimensioni ridotte ed elevata efficienza rispetto alle tipiche soluzioni di ricarica ad alta frequenza, tanto da poter entrare in una penna digitale ricaricabile mentre è collegata in modalità wireless al tablet. L'integrazione della ricarica wireless a bassa frequenza nelle penne Active ES di Wacom apre nuove possibilità. Questo concetto nuovo e innovativo porta la comodità della ricarica wireless agli accessori, consentendo agli utenti di esprimersi in modo creativo su smartphone e tablet. Un'esclusiva funzio-



ne di rilevamento ping fornisce un'indicazione precoce della connessione del carica batterie wireless e migliora le prestazioni termiche durante ed al termine della carica completa della batteria. Inoltre, il processore integrato a 32 bit, offre un elevato livello di programmabilità e parametri di gestione del sistema che possono essere facilmente configurati per funzionare con bobine di piccole dimensioni che si adattano alle penne digitali.

www.renesas.com

Nuovi device CoolSiC M1H 1200V da Infineon

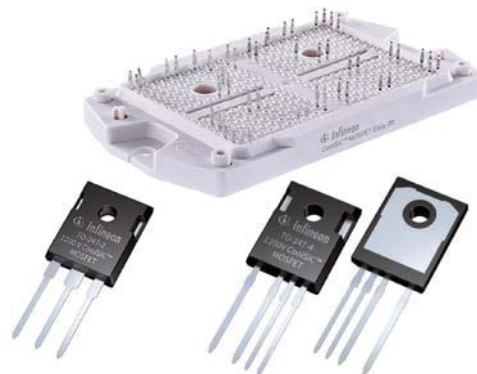
Infineon introduce la famiglia CoolSiC denominata M1H, insieme a soluzioni discrete che utilizzano la tecnologia di interconnessione .XT. Il chip M1H offre un'elevata flessibilità ed è adatto per gli inverter degli impianti fotovoltaici, che devono soddisfare picchi di notevole intensità; è ideale anche per applicazioni di ricarica rapida nei veicoli elettrici, sistemi di accumulo di energia e altre applicazioni industriali.

Gli ultimi progressi della tecnologia di processo CoolSiC consentono una finestra operativa di gate molto più ampia che migliora il valore della resistenza in conduzione a parità di dimensioni del chip; allo stesso tempo, la più ampia finestra operativa di gate fornisce un'elevata robustezza contro i picchi di tensione legati al driver e al layout, senza alcuna restrizione, anche a frequenze di commutazione più elevate. Insieme alla tecnologia del silicio M1H, è stata aggiornata anche la tecnologia del packaging e sono state adottate numerose varianti di package per consentire densità di potenza più elevate con maggiori opzioni per i progettisti che potranno adottare la

soluzione più adatta alle loro esigenze.

Con il chip M1H, la resistenza in conduzione dei moduli può essere notevolmente migliorata, rendendo i dispositivi più affidabili ed efficienti. Inoltre, con una temperatura di giunzione massima di 175°C, aumenta anche la capacità di sovraccarico, consentendo una maggiore densità di potenza e una migliore risposta in caso di guasto. Rispetto al suo predecessore, l'M1H implementa nei moduli una differente RG interna, consentendo di ottimizzare facilmente il comportamento di commutazione. Oltre alla famiglia di moduli Easy, il portafoglio CoolSiC MOSFET 1200 V M1H include nuovi MOSFET in package TO247-3 e TO247-4 con resistenza di ON bassissima, 7 mΩ, 14 mΩ e 20 mΩ. I nuovi dispositivi sono facili da integrare, grazie alle nuove possibilità di overshoot e undershoot offerti dalla tensione di gate, con i nuovi livelli gate-source che possono arrivare a ~10V; anche per quanto riguarda la risposta all'effetto valanga e ai corto circuiti, i nuovi dispositivi offrono garanzie superiori.

www.infineon.com





ABBONAMENTO
ANNUALE



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



Visita: www.elettronica.in.it/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



Il buongiorno si sente dal mattino

NEW

Wooden Clock - Sveglia digitale con temperatura e calendario

Orologio digitale con display a LED rossi in un elegante contenitore effetto legno. Dispone di allarme sonoro, batteria di backup (due LR44 incluse), visualizzazione temperatura ambiente e calendario. Alimentazione tramite 3 batterie minitipo AAA o tramite alimentatore USB (acquistabile separatamente) e cavetto da USB a plug DC (compreso). Dimensioni 180x80x80 mm, peso 270 grammi.



€15,00

cod. WC233

Prezzi IVA inclusa.

Sveglia digitale a campanelli meccanici con doppio allarme

- » Involucro verde fluo e bianco
- » Possibilità di impostare due differenti orari di allarme
- » Retroilluminazione display verde
- » Dimensioni 110x105x74 mm
- » Alimentazione tramite 3 batterie minitipo AAA (acquistabili separatamente)
- » Peso 200 grammi



€11,90

cod. WC206

Sveglia digitale a specchio con display a led bianchi

- » Display a specchio a led bianchi con visualizzazione 12/24 h
- » Allarme con snooze
- » Regolazione di luminosità (normale/notte)
- » Visualizzazione temperatura (°C - °F)
- » Alimentazione tramite 3 batterie minitipo AAA o alimentatore (acquistabili separatamente)
- » Dimensioni 82x82x35 mm
- » Peso 88 grammi

Attenzione! Necessita di presa USB.



€12,90

cod. WC214

Radiosveglia con proiezione dell'ora e radio FM

- » 2 sveglie
- » Allarme con radio o segnale acustico
- » Radio FM 87,5 - 108 MHz con 10 stazioni memorizzabili
- » Proiezione dell'ora con regolazione manuale dell'angolazione e della nitidezza
- » Distanza di proiezione da 0,9 a 2,7 metri
- » Alimentazione 220 V AC ~ 50 Hz, 5 W
- » Dimensioni 160x65x57 mm
- » Peso 367 grammi



€27,90

cod. CRP717BLACK

Sveglia radiocontrollata

- » Display LCD con retroilluminazione BLU
- » Data e orario radiocontrollati tramite segnale radio DCF77
- » Mostra le fasi lunari
- » Funzione sveglia con buzzer
- » Funzione termometro per temperatura interna
- » Alimentazione tramite due batterie da 1,5 volt tipo AAA (acquistabili separatamente)
- » Dimensioni 82x2,50x82 mm
- » Peso 90 grammi



€11,90

cod. REC34BLACK

TIMER START & STOP

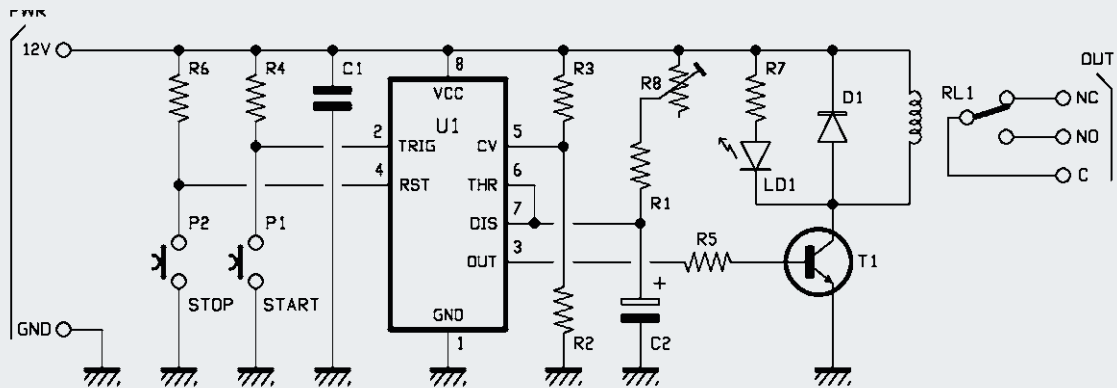


di **BORIS LANDONI**

Permette di attivare per un predefinito tempo, tramite il relé di bordo, un utilizzatore ed anche di disattivarlo manualmente prima che l'intervallo stabilito sia trascorso.

I temporizzatori trovano impiego in numerosissime applicazioni, come il controllo di elettropompe e ventilatori, azionamenti, illuminazione; proprio in tema di luci, il classico timer è quello che fa accendere a tempo le lampadine di atri e scale negli edifici, dietro comando di un pulsante.

Quello siffatto è un circuito che triggerato dal pulsante (o da un ingresso di attivazione) avvia una temporizzazione e, tramite un relé elettromeccanico o allo stato solido, alimenta un utilizzatore sconnettendolo allo scadere del tempo impostato; ma esistono altri tipi di temporizzatore che potremmo considerare "più completi" e che sono utili, se non indispensabili, in altre situazioni: ci riferiamo ai timer start/stop, dei quali un buon esempio è il progetto che vi proponiamo in queste pagine. Si tratta di circuiti che attivano un utilizzatore alla pressione di un pulsante di avvio (start) e che in assenza di ulteriori comandi lo disconnettono (sempre



mediante relé o interruttore allo stato solido...) allo scadere di un intervallo di tempo preimpostato, ma che dispongono di un secondo pulsante (stop) premendo il quale è possibile porre fine istantaneamente alla temporizzazione e quindi sconnettere l'utilizzatore anche prima dello scadere dell'intervallo di tempo impostato. Vediamo dunque come funziona il timer start/stop analizzando il nostro circuito.

SCHEMA ELETTRICO

Come potete vedere, si tratta di qualcosa di piuttosto semplice, che si avvale del popolare e immancabile timer "tuttofare" 555, qui utilizzato con i comandi di trigger e di reset separati e assegnati, rispettivamente, al pulsante START (P1) e al pulsante STOP (P2) della temporizzazione; per l'esattezza, il nostro 555 (chiamato U1 nello schema elettrico) funziona nella configurazione da multivibratore monostabile, però con la possibilità

di gestire manualmente il piedino di reset allo scopo di azzerare in qualsiasi momento la temporizzazione e caricare istantaneamente il condensatore elettrolitico C2.

Per farvi comprendere cosa significhi ciò nella pratica, diamo insieme uno sguardo all'interno dell'integrato 555: questo integrato contiene (**Fig. 1**) due comparatori aventi in comune un partitore di tensione multiplo che polarizza, per l'operazionale superiore l'ingresso invertente e per quello inferiore, il non-invertente; dei comparatori vengono resi accessibili dall'esterno gli ingressi non-invertente di quello superiore (i termini superiore e inferiore si riferiscono al potenziale di riferimento che ricevono dal partitore multiplo, quindi quello superiore riceve la tensione più elevata), corrispondente al piedino 6 (THRES, ossia Threshold) e l'invertente di quello superiore, corrispondente al piedino 2 (TRIG, ossia Trigger).

Il nodo tra la prima e la seconda (considerate

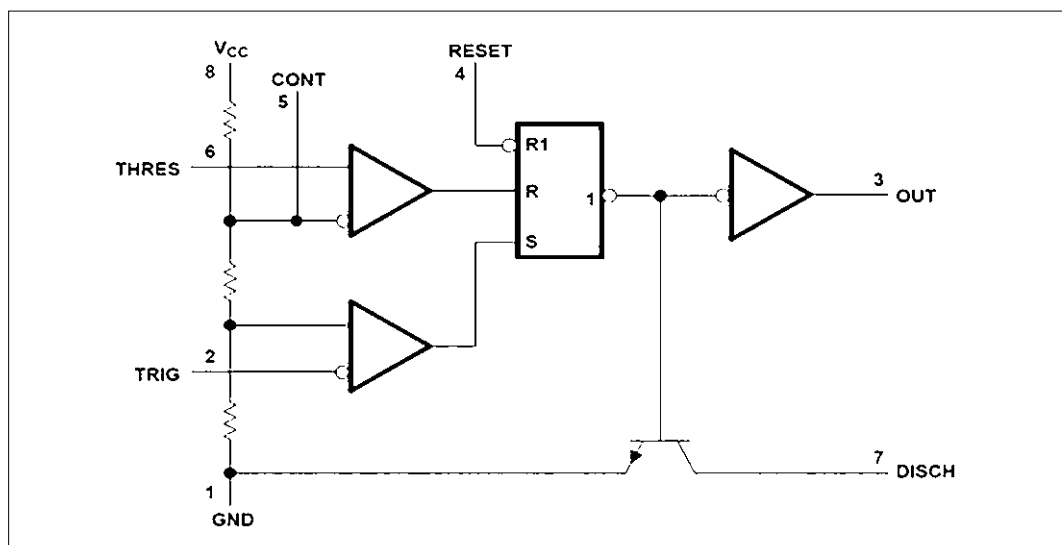


Fig. 1
Schema a blocchi
del 555.



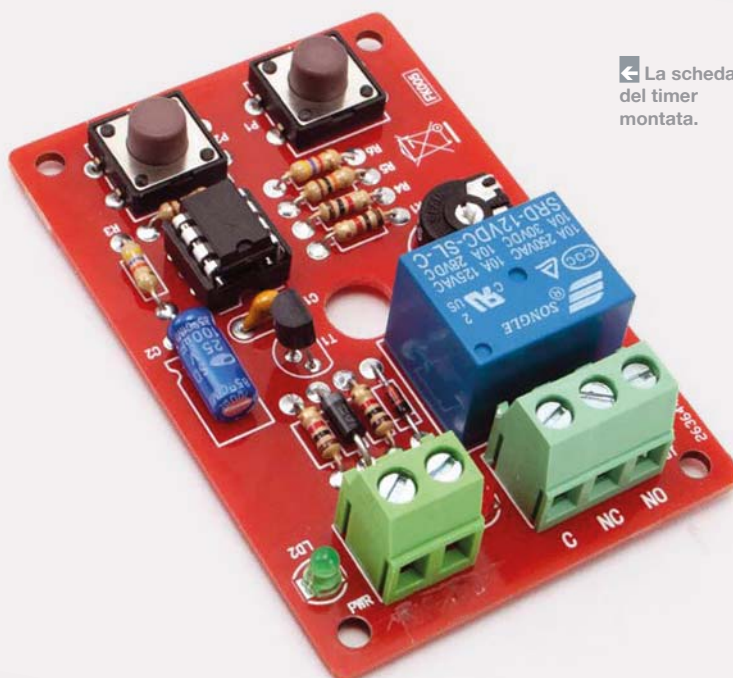
dall'alimentazione positiva) resistenza del partitore multiplo di riferimento dei due comparatori (alimentato, come il resto degli stadi interni al 555, dal piedino 8) viene portato all'esterno tramite il piedino 5 (CONT, Control Voltage) e ciò permette di alterare le tensioni di riferimento dei comparatori, così da controllare con una tensione esterna la frequenza di lavoro nella configurazione astabile (operando lo shift di frequenza come si fa nei VCO, ossia gli oscillatori controllati in tensione) oppure la durata degli impulsi in quella da timer (monostabile). Più esattamente, nella configurazione astabile, abbassando la tensione sul piedino 5 si accorcia il periodo della commutazione dell'uscita e quindi si incrementa la frequenza generata, viceversa lasciando salire la tensione fino al livello caratteristico (a vuoto, ossia senza alcun circuito collegato al piedino 5) o incrementandola si allunga il periodo e si riduce la frequenza prodotta. Nella configurazione monostabile, invece, la tensione altera la durata degli impulsi, il che permette, pilotando il trigger con impulsi a frequenza costante, di ottenere un modulatore PWM. Nel nostro caso, comunque, tale funzione non ci serve perché lavoriamo a singolo impulso. Le uscite dei comparatori del 555 entrano una nel RESET (quella del comparatore superiore) e nel SET (comparatore inferiore) di un flip-flop di tipo RS, che è un circuito logico la cui uscita diretta (Q) va a livello alto (corrispondente a circa il potenziale del piedino 8) quando il SET è a livello alto, ovvero assume lo zero logico se SET è posto a circa zero volt; l'uscita presenta un comportamento esattamente opposto quando ad essere stimolato è l'ingresso RESET, il quale, posto a livello logico alto, resetta il flip-flop, intendendo con ciò che ne azzerà lo stato di Q. Il flip-flop ha anche un'uscita complementata ($\bar{Q}$), la cui condizione logica è sempre l'inverso di quella Q; nel 555 è collegata, tramite una resistenza, alla base di un transistor NPN configurato ad open collector il cui emettitore è connesso a massa e il collettore al piedino DISCH (Discharge, 7). Il transistor può commutare una corrente di collettore di 200 mA, utilizzato in modo sink (ad assorbimento di corrente). L'uscita diretta, Q, è invece collegata al piedino OUT (3) dell'integrato attraverso un buffer interno push-pull, capace di erogare un massimo di 200 mA. Il negativo di alimentazione, vale a dire la massa di riferimento del 555 corrisponde al piedino 1, mentre al 4 è connesso il reset a logica invertita del flip-flop: questo piedino permette di resettare forzatamente il circuito dall'esterno applicando un livello logico basso.

Bene, ora che sapete cosa c'è all'interno del 555 possiamo spiegare in che modo funziona nel nostro circuito, partendo dalla premessa che è configurato come monostabile, con una variante al circuito canonico, che prevede il RESET disattivato e posto fisso sulla linea di alimentazione; infatti nel nostro caso il RESET è dotato di resistore di pull-up che lo mantiene normalmente a livello logico alto e quindi disattivato, ma può essere portato a livello basso e quindi attivato, mediante il pulsante P2 (STOP). Al solito, abbiamo il trigger (piedino 2) dotato di resistore di pull-up affinché rimanga normalmente a livello alto (lasciando il timer a riposo) e di un pulsante terminante a massa, ossia P1, per far partire la temporizzazione. Inoltre, rispetto allo schema canonico, in questo circuito andiamo ad alterare i potenziali di riferimento dei comparatori, attraverso il partitore formato da R3 ed R2, rispettivamente 4,7 kohm e 10 kohm, in modo da elevare leggermente quello di riferimento sul piedino 5.

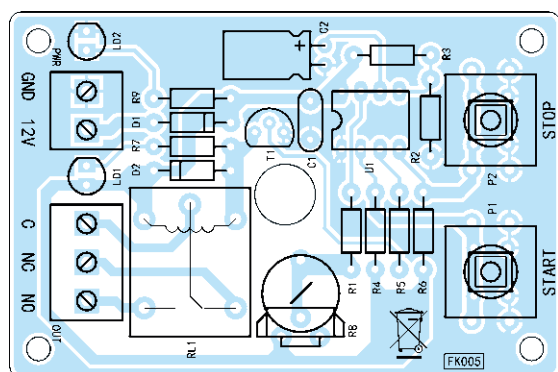
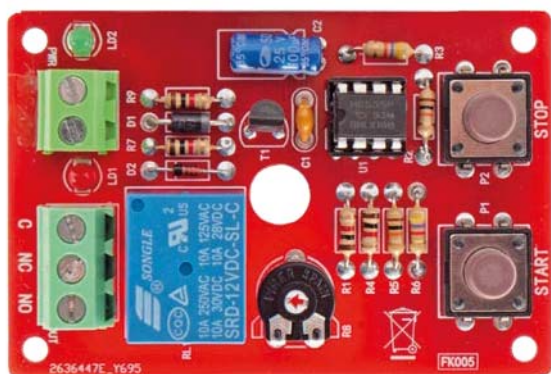
Vediamo dunque come funziona il 555 partendo dall'accensione e supponendo C2 inizialmente scarico, ossia avente una tensione di 0 volt: la condizione iniziale del circuito vede il piedino 6 a circa zero volt e il 2 a livello alto (vi provvede il resistore di pull-up R4) quindi l'uscita del comparatore 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 e lo stesso sul RESET.

CARATTERISTICHE TECNICHE

- **Tensione di alimentazione:** 12 V
- **Corrente assorbita:** 60 mA
- **Temporizzazione:** 0,11÷240 s
- **Uscita a relé con scambio completo**
- **Tensione commutabile:** 250 Vca
- **Corrente commutabile:** 2 A
- **Comando a pulsante di start e stop**



◀ La scheda del timer montata.



Elenco Componenti:

R1, R4, R7, R9: 1 kohm
 R2, R5: 10 kohm
 R3, R6: 4,7 kohm
 R8: Trimmer 2,2 Mohm MO
 C1: 100 nF 63 VL poliestere
 C2: 100 µF 16 VL elettrolitico
 D1: 1N4007
 D2: 1N4148
 T1: BC547
 U1: NE555
 LD1: LED 3 mm rosso

LD2: LED 3 mm verde
 RL1: Relé 12V/10A 1 scambio
 P1, P2: Pulsante 12x12 mm (2 pz.)

Varie:
 -Zoccolo 4+4
 -Morsetto 2 poli passo 5 mm
 -Morsetto 3 poli passo 5 mm
 -Circuito stampato FK005
 (73x49 mm)

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). La situazione rimane questa fin quando non viene premuto il pulsante P1, ovvero lo START del timer; premendo il pulsante e rilasciandolo si fornisce un impulso a zero logico al piedino 2, 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 C2 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 C2 e pone fine alla temporizzazione. Se nessun pulsante è premuto, il circuito rimane come nelle condizioni iniziali, in attesa che venga premuto P1; se viene invece premuto P2 in condizioni di riposo, tale azione non sortisce alcun effetto.

Vediamo dunque cosa succede se durante una temporizzazione, vale a dire nell'intervallo di tempo compreso tra la pressione di P1 e il ripristino del temporizzatore (esaurimento del tempo impostato) si interviene su P2: come già spiegato, quando si preme P1 si triggera il monostabile portando il SET del flip-flop a livello alto e la relativa uscita complementata a zero, ponendo in interdizione il transistor interno e lasciando che C2 si carichi attraverso la solita serie R1-R8; nel frattempo l'uscita del 555 (piedino 3) si trova a livello logico alto. Se in questo frangente si preme P2, si porta a zero logico il pin 4, quindi si attiva il reset esterno del flip-flop R1, con il risultato che l'uscita complementata dello stesso si porta a 1 logico forzando in saturazione il transistor interno e scaricando quasi istantaneamente l'elettrolitico C2; inoltre l'uscita diretta, piedino 3, ritorna a zero logico. La scarica del condensatore porta circa zero volt sul piedino 6 dell'U1, forzando a livello alto l'uscita del comparatore superiore e con essa il reset interno (R) del flip-flop, cosicché anche una volta che P2 venga rilasciato, il flip-flop rimarrà resettato e il monostabile tornerà nella condizione di riposo. 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:

$$T = 1,1 \times (R1+R8) \times C2 = 1,2 \times (1 \text{ k}+2,2\text{M}) \times 100 \mu\text{F}$$

In realtà, questa è la formula suggerita dal costruttore senza alcuna correzione delle tensioni fornite dal partitore multiplo interno al 555; nel nostro caso la tensione dell'ingresso invertente del com-

paratore superiore è stata leggermente innalzata, tanto che si può sostituire il fattore 1,1 con 1,4. Considerato che C8 è un trimmer e che è collegato a reostato, il tempo per cui il relé resterà eccitato dal momento della pressione del pulsante di start e senza alcun intervento sullo stop, varierà quindi da un minimo di 0,14 a un massimo di circa 300 secondi (5 minuti); per l'esattezza, portandone il cursore tutto verso la linea positiva di alimentazione si otterrà la minore durata, mentre spostando il cursore dall'estremo opposto e pertanto su R1, il trimmer avrà inserita la massima resistenza e ciò corrisponderà ad impostare la temporizzazione più lunga. Resta inteso che premendo stop prima, il relé ricade anche se il tempo impostato non è trascorso. 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 l'integrato U1 si serve per commutare l'alimentazione sul carico; rimanendo sullo schema elettrico, il piedino 3 del 555 pilota la base del transistor NPN T1 attraverso la resistenza di polarizzazione R5, mandando in saturazione il BJT quando si trova a livello alto e pertanto per tutta la durata della temporizzazione. In questo intervallo, 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); allo scadere del tempo, 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.

Sempre in parallelo alla bobina del relé abbiamo inserito un bipolo LED-resistenza formato da LD1 ed R7, con lo scopo di segnalare visivamente la condizione di uscita attivata.

Il circuito si alimenta in corrente continua attraverso la morsettiera facente capo a +12V e GND. Ricordiamo che allo scopo vi occorre un alimentatore da rete AC/DC in grado di erogare una tensione continua di 12V, meglio se stabilizzata, ed una corrente di un centinaio di milliamper.

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 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 (tutti a montaggio tradizionale, per la gioia di chi non ha pratica con i componenti SMD) con le resistenze, il diodo al silicio D1 e i condensatori non polarizzati, quindi procedendo con l'integrato 555, che potete montare su uno zoccolo dip 4+4 contatti (in tal caso inserirete l'integrato una volta saldati tutti gli altri componenti). Passate al LED (ricordate che il catodo di quest'ultimo è l'elettrodo dal lato smussato del contenitore) e sistemate poi il transistor NPN, quindi procedete con le morsettiere a passo 5 mm, l'elettrolitico C2 ed il relé.

Per l'orientamento dei componenti polarizzati fate riferimento al disegno di montaggio visibile in queste pagine.

INSTALLAZIONE E UTILIZZO

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. ▣



Cosa occorre?

Il timer start/stop (cod. FK005) è in vendita in scatola di montaggio a Euro 14,00.
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



RICARICA I TUOI DISPOSITIVI OVUNQUE TI TROVI CON I NUOVI PANNELLI SOLARI PORTATILI DA 60 E 80W



PANNELLO SOLARE MONOCRISTALLINO DA 60W

ad alta efficienza (16,85%), pieghevole e portatile. Dispone di uscita DC max 20,9 volt 2,87A, uscita USB-A da 5V 3A, uscita USB-QC: 5V 3A / 9V 2A / 12V 1,5A. Esternamente è realizzato in nylon e dispone di due robuste maniglie con potenti magneti che assicurano una salda chiusura. Ideale per il campeggio, l'escursionismo e altre attività all'aperto. Dimensioni: 530x440x30 mm (chiuso), 880x530x10 mm (aperto), peso: 2.900 grammi.



cod. FOLDINGPAN60

€ 89,00



DISPONIBILE ANCHE NELLA VERSIONE DA 80W - cod. FOLDINGPAN80 **€ 119,00**



USCITA PRESA DC

(5,5x2,1 mm): max 20,9 volt – 2,87 A



USCITA USB-QC
5V 3A / 9V 2A / 12V 1,5A

USCITA USB-A
5V 3A

Prezzi IVA inclusa.

IN AUTO

DASHCAM POWER

di MIRKO UGOLINI



Circuito basato su un piccolo ATtiny85 e un sensibile sensore di movimento che permette di alimentare una dashcam in modo semplice senza dover modificare l'impianto elettrico dell'auto.

A

nche gli automobilisti italiani stanno scoprendo le DashCam e il loro impiego è sempre più diffuso. I progressi nella tecnologia in questo campo sono molto promettenti. Fino qualche anno fa una DashCam di qualità era piuttosto costosa e le dimensioni non passavano proprio inosservate. Ora i prezzi sono molto più abbordabili e le camere hanno raggiunto dei livelli di miniaturizzazione che permettono loro di essere una presenza molto più discreta vicino allo specchietto retrovisore. Rimane il problema del cablaggio della camera. La maggior parte delle camere è provvista di un cavo USB di una certa lunghezza che deve poi essere collegato ad una estremità con un connettore micro-USB alla camera ed alla estremità opposta ad un apposito convertitore 12VDC/5DC (normalmente non fornito di serie) da collocare in prossimità della scatola fusibili.



Fig. 1
Fusibile
sdoppiatore.

Alcuni di questi alimentatori permettono di mantenere la camera accesa per un certo lasso di tempo (configurabile con un dipswitch) anche quando si spegne il quadro generale.

Per ottenere questo funzionamento bisogna collegare l'alimentatore con tre cavetti.

Un cavetto va collegato al positivo di alimentazione sempre sotto tensione, un cavetto va collegato alla massa tramite un capicorda sullo chassis, ed il terzo deve essere collegato ad un punto sotto chiave di accensione in modo tale che quando si accende il quadro o si spegne, questa informazione possa essere recepita dall'alimentatore e secondo la configurazione scelta sul dipswitch, venga rimossa istantaneamente l'alimentazione alla camera oppure sia ritardata di un certo tempo. Il problema è proprio questa terza connessione sotto quadro che molte volte è difficile da individuare oppure difficile da raggiungere, anche perché il quadro fusibili, dobbiamo ammetterlo non è quasi mai in una posizione comoda da raggiungere. Spesso capita inoltre che non ci sia alcun posto libero sotto quadro nella scatola fusibili.

Quindi normalmente bisognerebbe portare l'auto da un elettrauto che andrebbe ad utilizzare un morsetto rubacorrente che funziona andando a perforare il cavo elettrico sul quale passa il segnale sotto quadro, per esempio quello della radio di bordo. Beh, non è un lavoro molto complicato, ma dobbiamo dire che per chi vuole evitare di modificare seppur di poco l'impianto elettrico non è la soluzione ottimale.

Abbiamo aggirato il problema utilizzando un

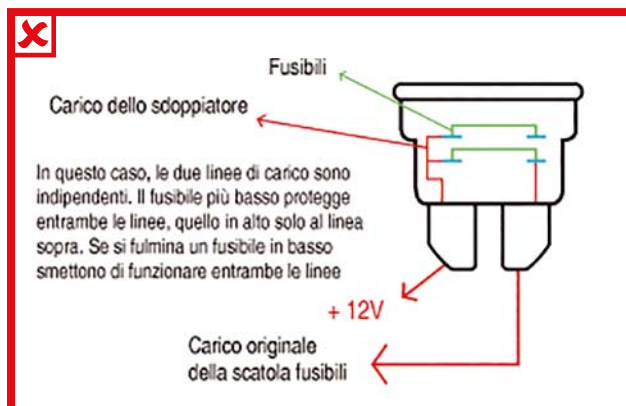
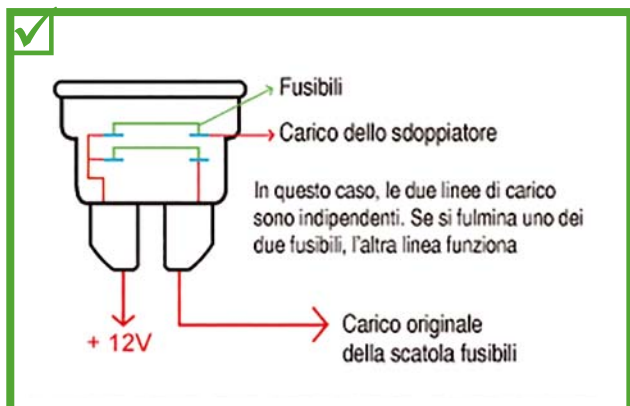
convertitore switching DC/DC che però viene controllato tramite un microcontrollore ATtiny85 ed un microsensore di vibrazione. In pratica un alimentatore con timer monostabile retriggiabile. In questo modo sono necessari solo due cavetti di alimentazione da portare al nostro alimentatore che rimane costantemente sotto alimentazione, ma in modalità LowPower (pochi micro Ampe-re) disattivando completamente il convertitore Switching e portando il microcontrollore in modalità LowPower. Nel momento in cui il sensore recepisce una minima vibrazione, invia un interrupt al microcontrollore il quale viene risvegliato nella modalità di funzionamento normale e conseguentemente abilita il convertitore Switching, per cui quasi istantaneamente la DashCam si accende. Essendo un sistema retriggiabile, se perviene una piccola vibrazione prima che sia scaduto il periodo stabilito, il sistema si retriggi e allunga così la permanenza della camera nello stato ON senza provocare interruzioni di alimentazione alla stessa. Quindi la camera si accenderà nel momento in cui entriamo in macchina e si spegnerà dopo che sarà passato un certo periodo dal momento in cui abbiamo lasciata parcheggiata la vettura. Dopo che sarà scaduto il periodo da noi impostato, sarà sufficiente urtare anche lievemente la carrozzeria per risvegliare istantaneamente la DashCam che provvederà a registrare il responsabile dell'urto.

COME COLLEGARE L'ALIMENTATORE

Come accennato precedentemente, per poter collegare il nostro alimentatore al circuito dell'auto abbiamo necessità di prelevare i 12V della batteria servizi e di un contatto di massa. Per i 12V utilizzeremo un cavetto con sdoppiatore di fusibile, visibile in **Fig. 1**. Come potrete notare il dispositivo ha un alloggiamento per due fusibili nel suo corpo,

Fig. 2
Collegamento
"Corretto"
fusibile
sdoppiatore.

Fig. 3
Collegamento
"Errato"
fusibile
sdoppiatore.



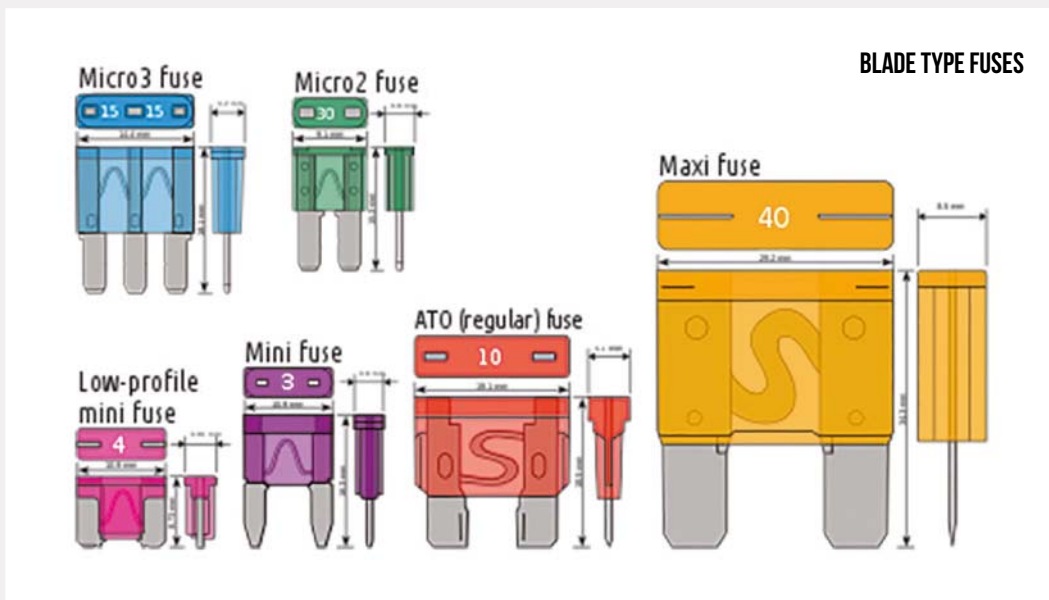


Fig. 4
Tipologie
di Fusibili
Automotive.

quello inferiore per proteggere il circuito originale e quello superiore per proteggere il circuito che fa capo al cavetto di cui è dotato e che termina con un faston.

In pratica il fusibile sdoppiatore, può essere inserito in una posizione libera della nostra centralina fusibili dell'auto ed in questo caso è necessario inserire un solo fusibile superiore. Invece, nel caso in cui tutte le posizioni della scatola portafusibili siano occupate, sarà necessario rimuovere uno dei fusibili già presenti, mettere al suo posto il dispositivo sdoppiatore, riposizionare il fusibile originale per proteggere il circuito ed aggiungere un nuovo fusibile nella posizione superiore per proteggere il nostro alimentatore.

Attenzione ad individuare una posizione fusibili che prima cosa sia sempre sotto tensione e poi che non sia strategica per la sicurezza della vettura. Altro suggerimento è quello di non rimuovere fusibili di alto amperaggio.

Attenzione inoltre ad un'altra cosa: il verso con cui si inserisce il dispositivo sdoppiatore non è indifferente. In **Fig. 2** è possibile vedere il modo corretto di inserirlo, infatti in questo caso il punto comune che unisce i due fusibili sarà collegato al +12V della centralina ed i due fusibili lavoreranno

in modo indipendente. Quello nella posizione inferiore continuerà a proteggere il circuito originale come se fosse al suo posto nella centralina fusibili. Quello superiore invece sarà deputato a proteggere il nostro alimentatore.

Nel circuito di **Fig. 3** invece vediamo che rovesciando il verso di inserimento del portafusibile sdoppiatore, il risultato è molto diverso. Il fusibile in basso protegge la linea collegata normalmente alla scatola fusibili, ma anche la linea del nostro alimentatore, il fusibile superiore protegge solo la linea del nostro alimentatore. Quindi se salta il fusibile inferiore, viene interrotta l'alimentazione anche alla linea dell'alimentatore della DashCam. In **Fig. 4** è possibile vedere che esistono Fusibili a lama di diverse dimensioni. Quindi prima di acquistare il portafusibile sdoppiatore individuate quello per la tipologia di fusibile corretto.

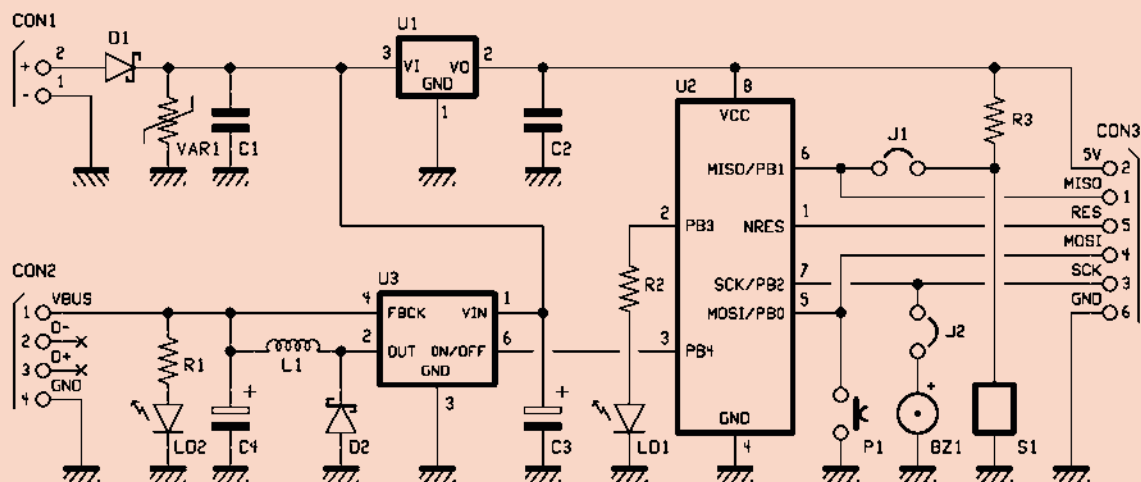
Normalmente sono impiegati fusibili di Tipo ATO oppure Mini Fuse.

SENSORE UTILIZZATO

Il sensore utilizzato in questo progetto è un piccolissimo (3.3 mm x 6.8 mm) dispositivo MEMS (Micro Electro-Mechanical Systems) che possiamo vedere in **Fig. 5**. Quando il dispositivo viene sotto-



Fig. 5
Come si
presenta il
sensore di
vibrazione.



posto ad una seppur lieve vibrazione su qualsiasi asse, si verificano delle variazioni repentine della conducibilità attraverso i suoi terminali. È estremamente importante individuare il tipo di sensore corretto perché la stessa casa produttrice ne realizza diverse tipologie con diverse sensibilità sugli assi: alcuni permettono solamente di individuare quando vengono ruotati ma non intercettano in maniera corretta le vibrazioni su tutti gli assi.

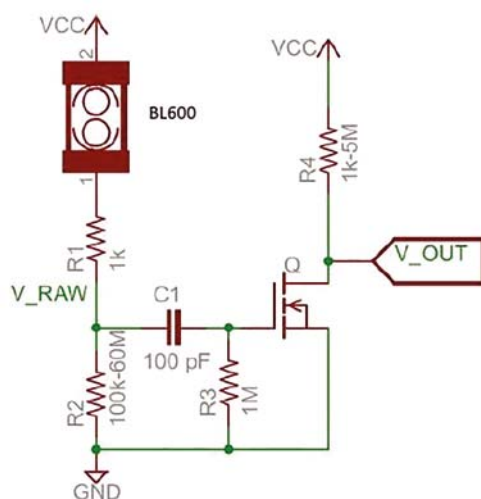
Il sensore che abbiamo utilizzato (cod. VIBSENSBL600) viene impiegato su numerosissimi sistemi commerciali perché ha un consumo estremamente ridotto ed è completamente passivo e per ultimo ma non meno importante presenta una sensibilità estremamente elevata. Pensate che mentre facevamo dei test, un nostro

collaboratore che si trovava a circa un metro e mezzo di distanza ha chiuso la custodia di plastica degli occhiali ed il click ha triggerato un evento sul sensore. In **Fig. 6** è riportato un tipico impiego del sensore ed in **Fig. 7** viene mostrata la registrazione dei due segnali, quello grezzo (RAW) presente tra la massa ed il punto (V_RAW) e quello presente sul DRAIN del MOSFET (V_OUT).

La scelta dei valori dei componenti possono variare la sensibilità alle vibrazioni, tuttavia la stessa casa produttrice, nel DataSheet indica una soluzione circuitale molto più semplice per collegare il dispositivo al microcontrollore. Vediamo infatti che in **Fig. 8** è stato impiegato un solo resistore e il punto V_RAW è stato collegato ad un PIN di Interrupt ad alta impedenza del microcontrollore.

SCHEMA ELETTRICO

Partendo dalla sinistra vediamo che è stato inserito un diodo (D1) in serie al positivo dei 12V della alimentazione per prevenire eventuali inversioni di polarità quando si effettua il cablaggio, inoltre potete notare la presenza di un Varistore (VAR1) a proteggere il dispositivo da impulsi spuri che potrebbero essere presenti sulla linea di alimentazione. Abbiamo poi il regolatore U1 con i relativi condensatori di entrata e di uscita, si tratta di regolatore di tensione (U1) MCP1799 LDO ad uso automotive che permette di alimentare il microcontrollore (U2) ATtiny85. Il regolatore MCP1799 accetta tensioni di ingresso fino a 45V, ha una tensione di uscita di 5V fissa ed una corrente massima erogabile di 80 mA ed un range di temperatura di lavoro da -40°C a 150°C. Noterete che il microcontrollore non ha alcun quarzo esterno, infatti abbiamo utilizzato



→ Fig. 6
Circuito completo
per intercettare le
vibrazioni.

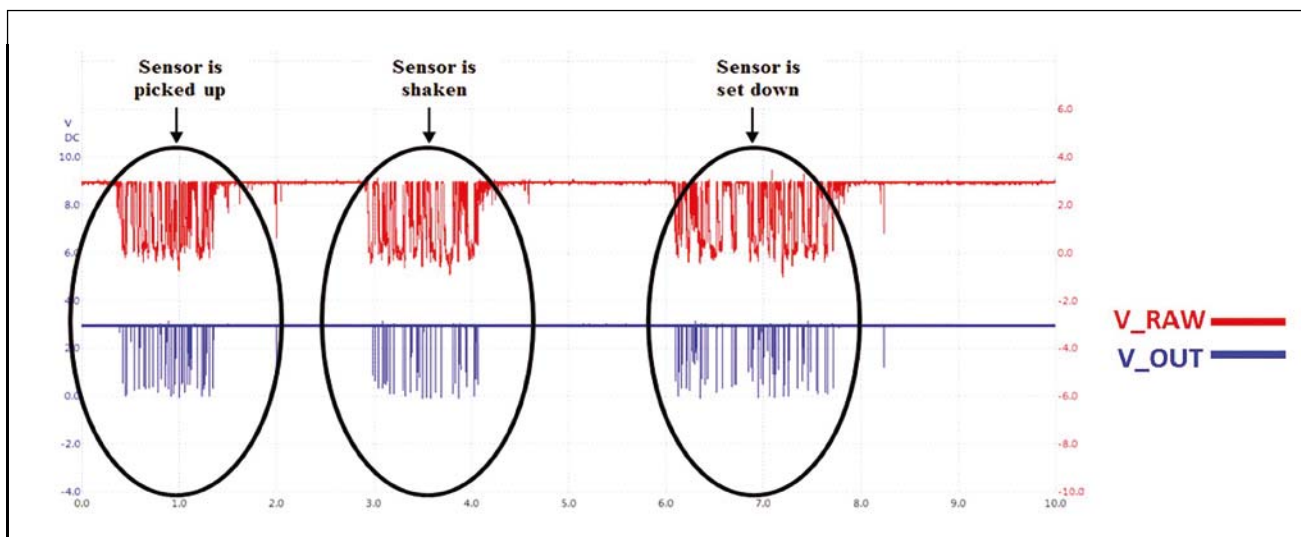


Fig. 7
Forme d'onda
presenti nei due
punti del circuito
di Figura 6.

l'oscillatore interno, questo ci permette di ridurre la complessità del circuito ed al contempo utilizzando una frequenza di clock bassa, la corrente assorbita durante il funzionamento sarà molto ridotta. Visto che stiamo utilizzando un microcontrollore della famiglia ATtiny, il numero di pin è molto scarso ed alcuni sono condivisi con i pin utilizzati nell'interfaccia di programmazione ISP.

Affinché non insorgano problemi o addirittura l'impossibilità a programmare il chip, abbiamo inserito dei jumper (J1, J2) che devono essere rimossi nella fase di programmazione del microcontrollore e poi riposizionati quando sarà terminata. J1 controlla l'applicazione del segnale proveniente dal sensore di vibrazione, mentre J2 interrompe il segnale che viene portato al buzzer piezoelettrico.

Abbiamo dotato il dispositivo di un pulsante (P1) che ci permetterà, come vedremo poi nella spiegazione del software, di definire la durata del periodo retriggrabile del Timer realizzato con il microcontrollore. Sono presenti due LED: LD1 visualizza le vibrazioni intercettate dal sensore MEMS avremo così un riscontro visivo della sua sensibilità, LD2 essendo collegato in parallelo (tramite un resistore di limitazione R1) alla tensione di uscita del regolatore switching indica l'attivazione del regolatore in seguito all'applicato al pin (ON/OFF) del segnale proveniente dal microcontrollore U3. Quindi quando il LD2 è acceso significa che il regolatore U3 è abilitato e avremo quindi i 5V sulla porta USB di uscita che andrà ad alimentare la DashCam. Il LD1 invece viene impiegato per visualizzare le vibrazioni intercettate dal sensore.

Il regolatore U3 è impiegato per convertire i 12V della batteria nei 5V necessari per alimentare la

Dashcam. Naturalmente in questo caso abbiamo adottato un regolatore di tipo switching. Questo ovviamente aumenta un poco la complessità del circuito, a beneficiarne però è il consumo del dispositivo. Come potete vedere dallo schema elettrico sono richiesti pochissimi componenti: un paio di condensatori, un induttore ed un diodo Schottky. L'uscita del regolatore viene portata al connettore (CON2) USB tramite la quale alimenteremo la DashCam.

IL CODICE

Per poter utilizzare l'IDE di Arduino con i controllori della famiglia ATtiny, dobbiamo aggiungere dei

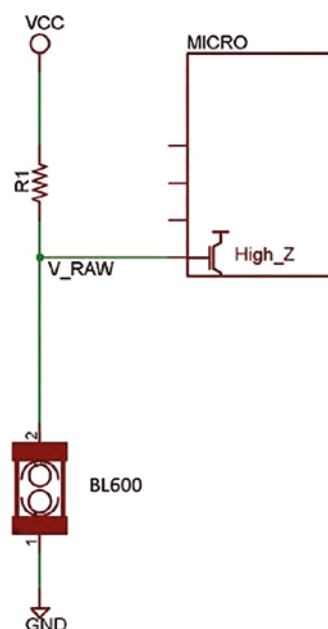


Fig. 8
Circuito
semplificato
per collegare
sensore al
microcontrollore.

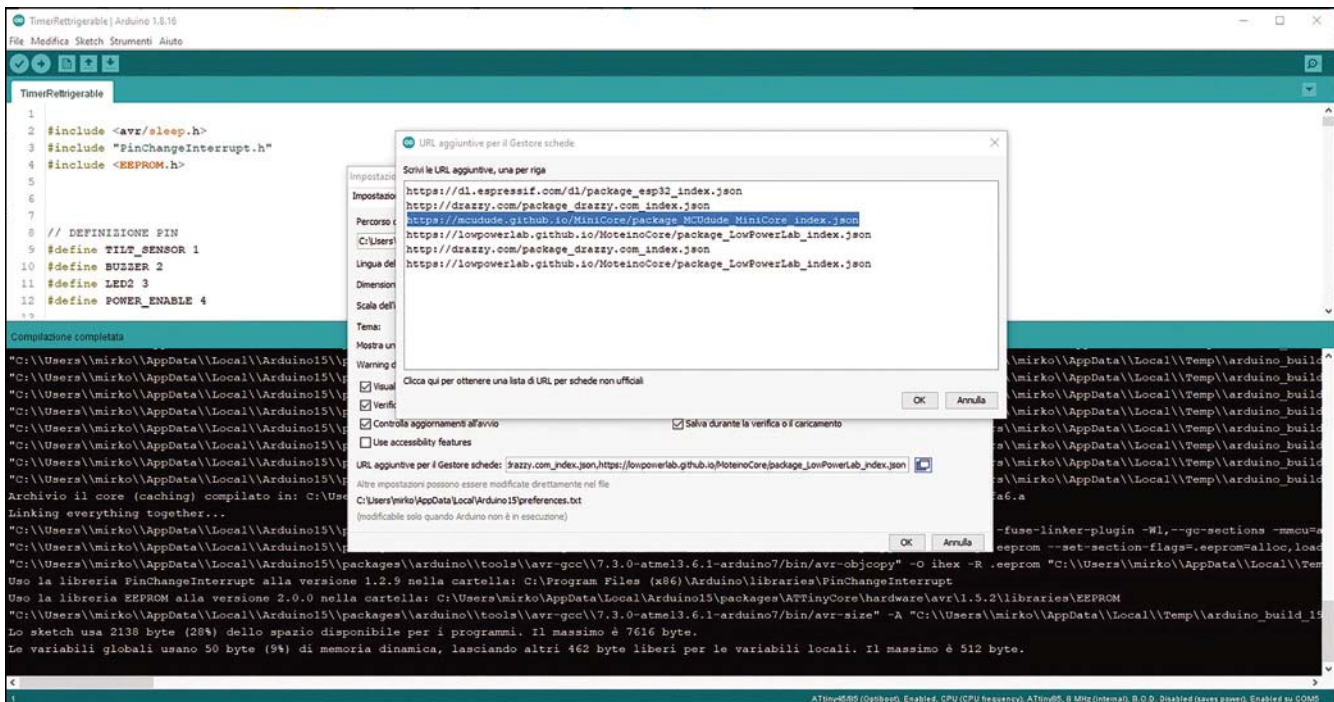


Fig. 9
Come aggiungere il supporto per microcontrollori ATtiny su IDE Arduino

nuovi riferimenti nella lista delle URL che permettono di scaricare le definizioni dei nuovi controllori che intendiamo utilizzare. Apriamo quindi il menu le voci "File/Impostazioni" poi individuiamo la text box con la dicitura "URL aggiuntive per il Gestore schede", clicchiamo sul pulsante alla destra ed si aprirà un finestra di popup come in **Fig. 9**, aggiungiamo quindi alla lista eventualmente già presente questo url: https://mcudude.github.io/MiniCore/package_MCUdude_MiniCore_index.json Salviamo e chiudiamo il popup. Ora, dal da Menu

"Strumenti" posizioniamoci sulla prima voce "Scheda: ..." e selezioniamola. Apparirà (**Fig. 10**) la maschera di selezione delle schede disponibili. Digitiamo nella textbox il testo ATtinyCore e nella zona sottostante dovrebbe apparire la libreria di "Spence Konde". Installiamola. Fatto questo, riavviamo l'IDE e poi torniamo a selezionare le schede disponibili. Dovremmo avere una nuova famiglia di controllori disponibili nella lista, come in **Fig. 11**. Dalla lista sulla destra scegliamo quindi la scheda

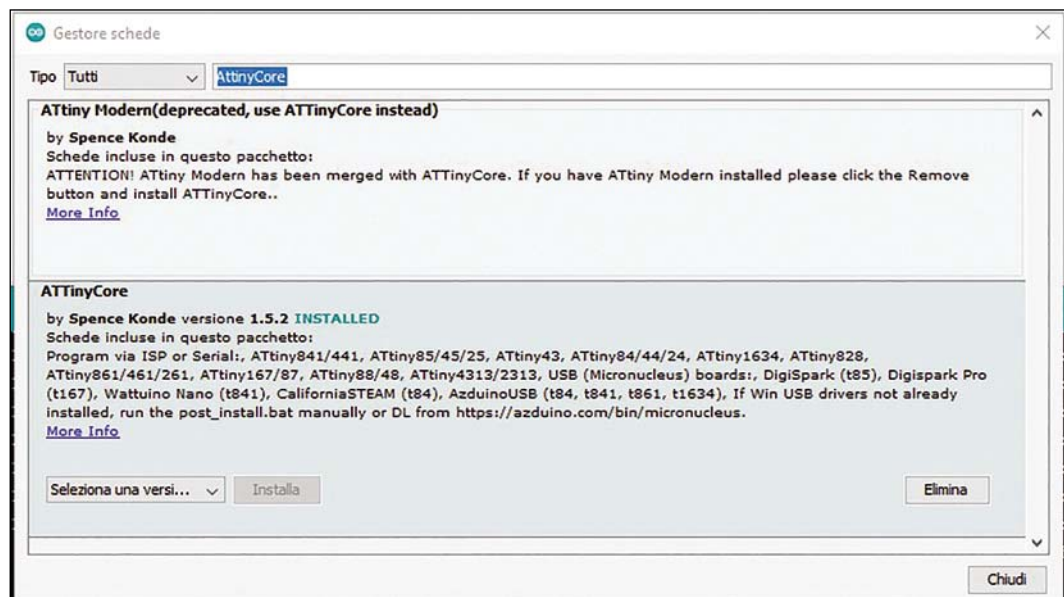
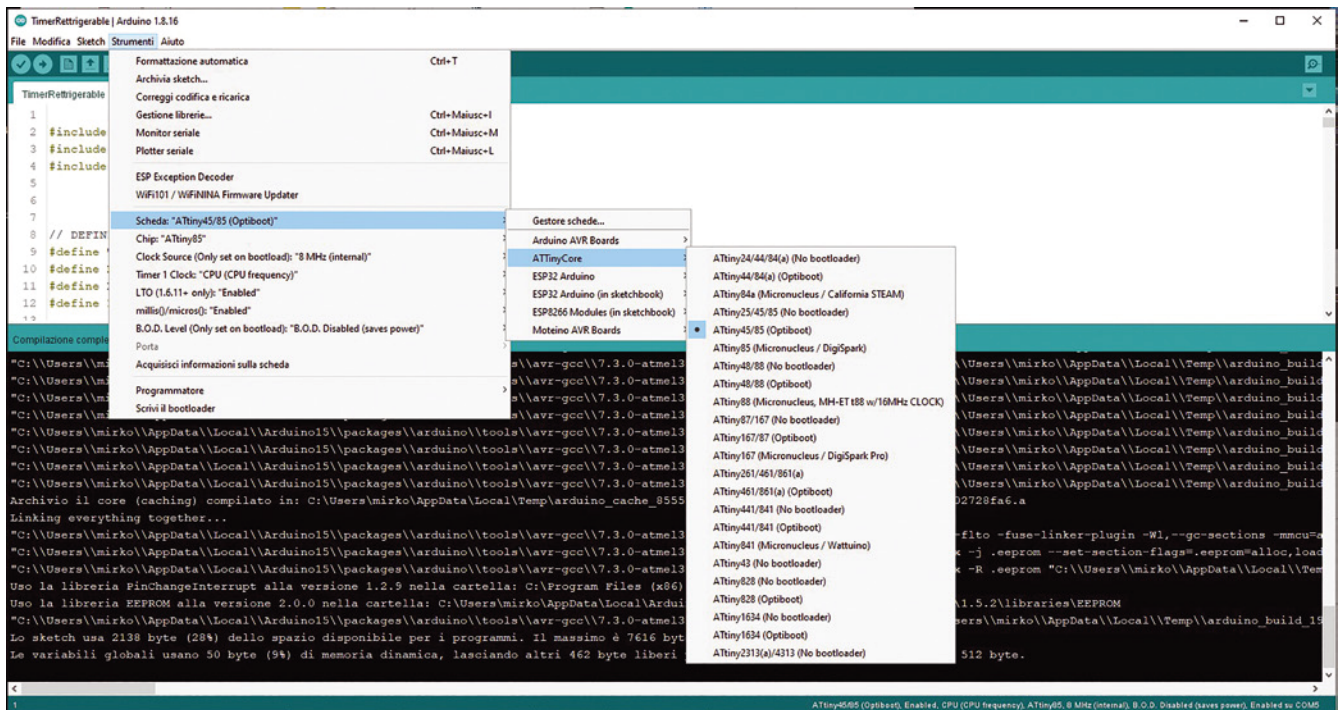


Fig. 10
Selezione della Estensione per le schede ATtiny.



"ATtiny45/85 Optiboot". Una volta selezionato il microcontrollore è necessario anche selezionare in modo corretto alcune opzioni.

In **Fig. 12** vediamo quelle che abbiamo adottato per il nostro alimentatore.

- Clock Source: 8MHz (internal)
- Timer1 Clock: CPU (CPU frequency)
- LTO: Enabled
- millis()/micros(): Enabled
- B.O.D.: Disabled (saves power)

La prima opzione "Clock Source" permette di selezionare la frequenza di clock con cui fare lavorare il microcontrollore. La seconda Opzione "Timer1 Clock" definisce che il timing per Timer 1 è derivata dalla frequenza di clock della CPU. L'opzione LTO (Link Time Optimization) permette di ottenere dei file binari di minore dimensione, ma può essere utilizzato solo se la versione dell'IDE è superiore all 1.6.11. L'opzione "mills()/micros()" è assolutamente indispensabile nel nostro progetto e deve essere abilitata. L'opzione "B.O.D." (B.O.D. è l'acronimo di Brown-out Detection) l'abbiamo disabilitata perché non c'è il pericolo che la tensione di alimentazione scenda sotto una certa soglia e possa mettere in crisi il microcontrollore. Ricordiamo infatti che questa opzione permette nel caso di circuiti alimentati con batterie di piccola capacità di tenere sotto controllo la tensione minima di funzionamento del microcontrollore e nel caso di resettare lo stesso.

PROGRAMMIAMO IL CHIP

Il Programmatore che andremo ad utilizzare è disponibile presso Futura Elettronica ed ha codice prodotto USBPROGAVR.

In **Fig. 13** vediamo come si presenta. E' un dispositivo molto versatile perché oltre ad essere un programmatore ISP è anche un convertitore USB-UART. Inoltre è dotato di un software a corredo ed è ben supportato ed aggiornato. Insomma è preferibile rispetto ai vari programmatori sottocosto che si trovano su internet.

Fig. 11
Selezione della scheda corretta nell'ambito di quelle proposte.

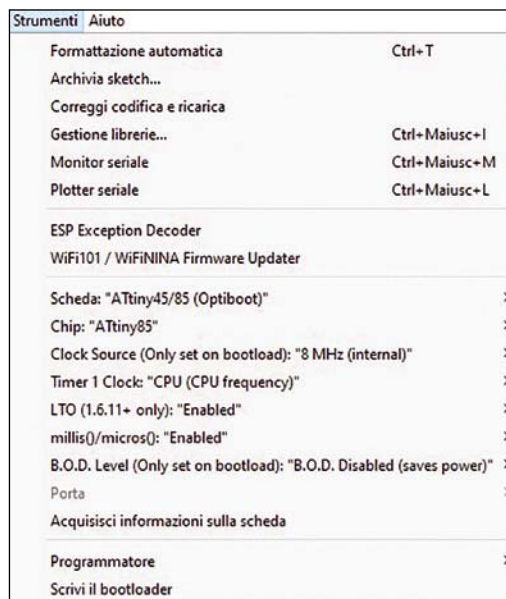


Fig. 12
Selezione dei parametri corretti relativamente al microcontrollore ATtiny85

Fig. 13
Il program-
matore ISP
Pololu.

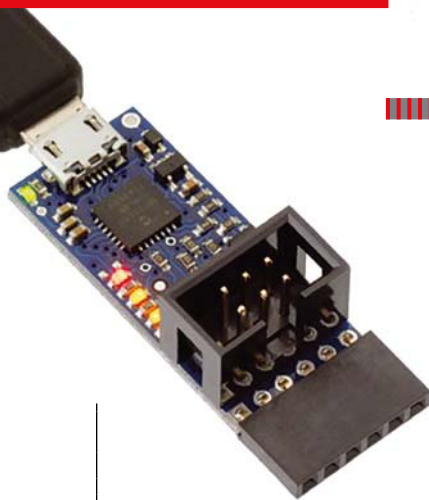
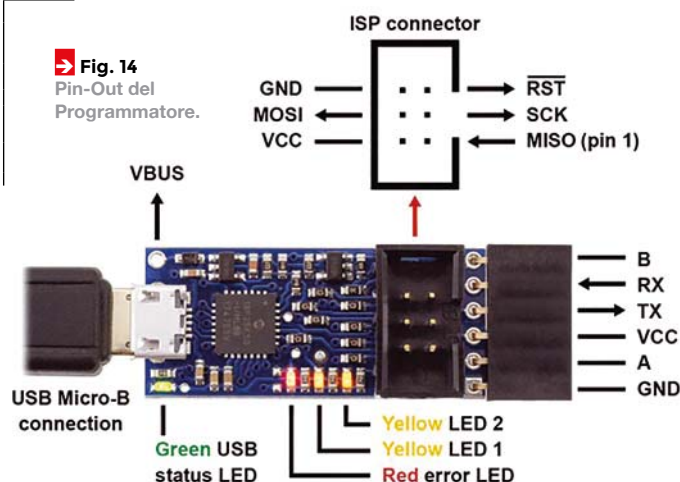


Fig. 14
Pin-Out del
Programmatore.



In **Fig. 14** vediamo il Pinout del programmatore. Il programmatore ISP che vi abbiamo proposto è il vero "coltellino svizzero" che non dovrebbe mancare nel cassetto di un Maker. Infatti non si tratta solamente di un programmatore ISP, include infatti anche un convertitore USB-UART. Per poter utilizzare il programmatore è necessario configurarlo tramite una utility Windows che è possibile scaricare dal sito di Futura elettronica parimenti al PDF del manuale completo di utilizzo. In **Fig. 15** riportiamo come si presenta la finestra di configurazione. Come potrete vedere il programmatore verrà individuato dal vostro sistema operativo come due porte separate: una porta "Programming Port" ed una "TTL Port", nel nostro caso rispettivamente sulla COM19 e sulla COM20. Quindi andremo a selezionare o indicare la COM19 nel caso volessimo caricare del firmware su dei dispositivi senza Bootloader, collegandoci quindi alla porta ISP come nel caso del nostro progetto. Oppure potremmo utilizzare la porta COM20 nel caso in cui sia necessario comunicare con un device per debuggare tramite seriale oppure per caricare del firmware su device che abbiano il BootLoader. Il programmatore AVR USB Pololu ha diverse impostazioni che influiscono su come la tensione viene regolata, distribuita e monitorata. Queste impostazioni possono essere visualizzate e modificate tramite il programma di configurazione "Configuration Utility". Queste impostazioni sono memorizzate nella memoria non volatile del programmatore, quindi persistono anche quando il programmatore non è alimentato.

Opzione "Vcc Output"

L'impostazione dell'uscita VCC determina se il pin VCC (che si trova sia sul connettore ISP che sull'interfaccia seriale, vedi **Fig. 14**) sarà un ingresso

o un'uscita. Per impostazione predefinita, l'uscita VCC è disabilitata, quindi il pin VCC sarà un input. Se si imposta il pin VCC come uscita, è necessario impostare anche la modalità del regolatore su 5 V o 3,3 V: tale impostazione specifica sia la tensione a cui opererà il programmatore sia la tensione che fornirà al pin VCC. L'uso del pin VCC come uscita può essere utile per alimentare sistemi a bassa corrente da USB. Naturalmente la tensione di uscita selezionata deve essere conforme alla tensione massima supportata dal microcontrollore che andremo a programmare.

Opzione "Regulator Mode"

- **Auto** (impostazione predefinita): in questa modalità, il programmatore funzionerà a 3,3 V all'avvio e passerà a 5 V se la tensione su VCC supera i 4096 mV. Il programmatore tornerà a 3,3 V se la tensione su VCC scende al di sotto di 3712 mV.
- **5 V**: il programmatore funzionerà a 5 V durante il normale funzionamento. Continuerà a funzionare brevemente a 3,3 V quando viene applicata per la prima volta l'alimentazione e continuerà a funzionare a 3,3 V quando è in modalità bootloader (che viene utilizzata per gli aggiornamenti del firmware).
- **3,3 V**: Il programmatore funzionerà a 3,3 V.

LIMITI QUANDO SI ALIMENTA UN CIRCUITO TRAMITE IL PROGRAMMATORE

In genere, quando si programma un AVR o si utilizza l'interconnessione seriale da USB a TTL del programmatore, il dispositivo con cui ci si interfaccia deve essere alimentato dalla propria alimentazione. Il programmatore non fornisce alimentazione ai

suoi pin VCC per impostazione predefinita. Tuttavia, se si desidera alimentare un carico a bassa corrente e bassa capacità, è possibile come visto nel paragrafo precedente configurare VCC come uscita utilizzando il software di configurazione del programmatore. Quando VCC è un'uscita, la sua tensione sarà 3,3 V o 5 V, e ciò è determinato dall'impostazione "Regulator mode" del programmatore. Se la tua scheda di destinazione presenta un condensatore con un valore troppo alto su VCC, potrebbe causare il brown-out (caduta di tensione) del programmatore o potrebbe causare interruzioni nella comunicazione USB. Il Pololu USB AVR Programmer ha una sezione di alimentazione migliorata con una velocità di risposta controllata che evita un transitorio molto repentino della tensione di uscita, quindi può alimentare schede target fino a circa 33 µF su VCC. Il programmatore AVR USB Pololu potrebbe comunque subire un ripristino del brown-out quando lo si collega a un target, ma dopo il brown-out è spesso in grado di recuperare e fornire alimentazione al target. Non collegare VCC all'uscita di nessun altro dispositivo, poiché ciò potrebbe causare un cortocircuito. Inoltre Non possono essere erogati più di 80 mA di corrente da VCC. In alternativa, si può utilizzare il pin VBUS del programmatore per alimentare dispositivi esterni. Questo pin è collegato direttamente alla linea di alimentazione del cavo USB. Come con l'uscita VCC, il circuito target non dovrebbe assorbire più di 80 mA e non dovrebbe avere troppa capacità in parallelo al carico. In genere è più sicuro utilizzare l'uscita VCC che utilizzare VBUS. L'alimentazione fornita all'uscita VCC passa attraverso un diodo, che impedisce alla corrente di rifluire nella porta USB, e il regolatore lineare del programmatore, che ha la limitazione di corrente. N.B. Il pin VBUS non fornisce protezioni, quindi c'è un rischio maggiore di danneggiare accidentalmente qualcosa.

RIPRISTINARE IL FUNZIONAMENTO DI UN CHIP CHE È STATO BLOCCATO PER ERRATA CONFIGURAZIONE DEI FUSEBIT

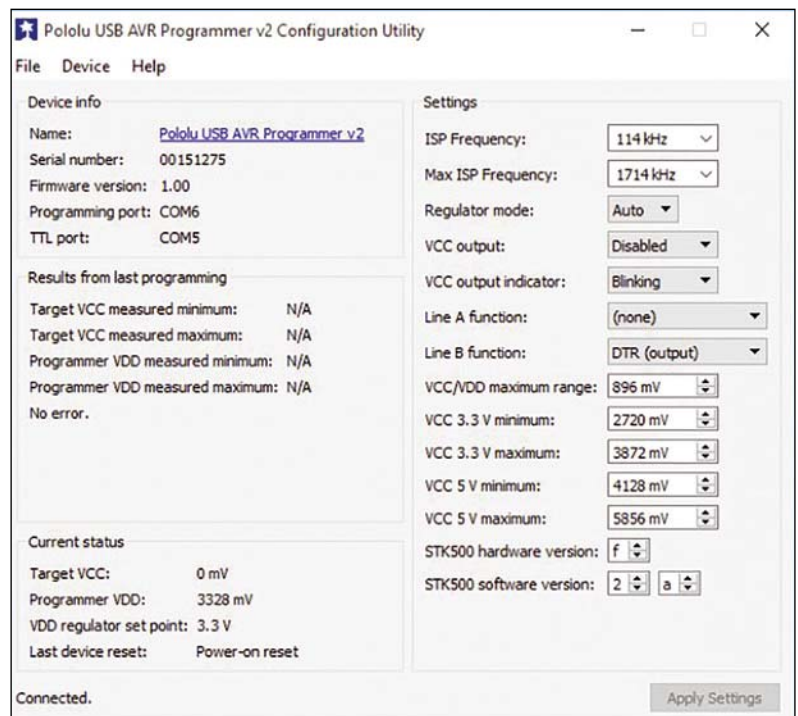
Alcune volte può capitare che la programmazione errata dei fusebit di un microcontrollore comporti l'impossibilità di riprogrammarlo semplicemente tramite i tradizionali programmatori seriali. Questo succede quando si settano male i fusebit relativi alla configurazione della sorgente di clock e quindi il micro diventa inutilizzabile perché si attende una sorgente di clock esterna. Il programmatore Pololu ha la possibilità di definire il Pin B (Fig. 14),

attestato sul connettore seriale, in modo che eroghi un segnale di clock a 100KHz che può essere collegato al pin XTAL2 del chip AVR per fornirgli il clock che ci permetterà di poter riprogrammare in modo corretto i fusebit. Per fare questo è possibile (Fig. 15) modificare l'opzione "Line B function" e settarla come "Clock (output)"

IL CODICE

Nel Listato 1 vediamo, nella parte iniziale del codice, l'inclusione delle tre librerie che utilizzeremo nel progetto. Abbiamo la libreria *avr/sleep.h* per la gestione della modalità Low Power del microcontrollore, *PinChangeInterrupt.h* che ci consente di utilizzare un pin di interrupt per risvegliare il micro dalla modalità a basso consumo ed infine *EEPROM.h* che ci consentirà di memorizzare delle impostazioni che persistano durante i riavvii del circuito. Subito sotto l'inclusione delle librerie abbiamo la definizione dei Pin GPIO. Il pin PB1 che gestisce anche l'interrupt è assegnato al sensore di vibrazione, PB2 al Buzzer, PB3 al led che notifica la vibrazione mentre PB4 al comando (ON/OFF) che abilita il convertitore Switching U3. Infatti quando il pin PB4 si trova nello stato logico 0 il pin ON/OFF si troverà nella condizione logica che abilita il convertitore DC/DC ad operare. Sotto la definizione dei GPIO troviamo diverse variabili, quelle degne di nota sono *m_status*, *n_in-*

Fig. 15
Utility di configurazione del programmatore ISP Pololu.



▼ Listato 1 - Parte iniziale codice

```
#include <avr/sleep.h>
#include <PinChangeInterrupt.h>
#include <EEPROM.h>

// DEFINIZIONE PIN
#define TILT_SENSOR 1
#define BUZZER 2
#define LED1 3
#define POWER_EN 4

#define PRESSED LOW
#define NOT_PRESSED HIGH
const unsigned long shortPress = 100;
const unsigned long longPress = 2000;
long blinkInterval = 500;
unsigned long previousBlink=0;
unsigned int m_status;
unsigned int n_interval=1;
bool power_status = false;
bool led_tilt_status = false;
unsigned long timer = 0;
unsigned long timer_led_tilt = 0;
unsigned long timer_buzzer = 0;
unsigned long countDown = 1000L;

typedef struct Buttons {
    const byte pin = 0;
    const int debounce = 10;
    unsigned long counter=0;
    bool prevState = NOT_PRESSED;
    bool currentState;
} Button;

// create a Button variable type
Button button;
```

- 0 = modalità normale di funzionamento
- 1 = modalità di configurazione

La variabile *countDown* definisce l'intervallo minimo della durata del monostabile retriggerabile, mentre la variabile *n_interval* il numero di intervalli *countDown* che è possibile impostare quando si entra nella modalità di programmazione del dispositivo. Sotto le variabili vediamo la definizione di una *Strunct* che definisce un nuovo tipo di dato. In pratica abbiamo definito un nuovo tipo di dato che abbiamo nominato Buttons e che è costruito aggregando diversi tipi di dati di base. Poi nella riga sotto abbiamo assegnato alla variabile button questa tipologia di dato.

Passiamo ora ad analizzare il blocco di Setup del codice, quello che come lascia intendere il nome viene eseguito una sola volta al boot del dispositivo. Da notare che contrariamente a quello che siamo abituati a fare su i chip arduino basati su ATmega328p o superiori, dotati di porte UART, in questo caso non abbiamo inizializzato alcuna porta seriale di comunicazione perchè il chip ATtiny85 non ne è dotato.

Avremo come di consueto, la definizione della modalità di funzionamento dei Pin utilizzati tramite l'istruzione *pinMode*, poi il settaggio dello stato logico iniziale di tali pin tramite la funzione *digitalWrite*. Abbiamo poi l'inizializzazione di alcune variabili allo stato iniziale.

Degno di nota è il punto in cui andiamo ad estrarre il valore della variabile *n_interval* dalla EEPROM. Se il valore memorizzato in EEPROM è diverso da 255 acquisiamo questo valore e lo salviamo nella variabile *n_interval* altrimenti il suo valore sarà quello di default e pari a 1.

Nella riga successiva vediamo come viene calcolato il ritardo del monostabile retriggerabile. Vediamo quindi che il valore sarà pari a *n_interval * 3600 * 1000* quindi avremo multipli di un'ora.

Poi sotto inizializziamo delle variabili che serviranno ad effettuare dei lampeggi del led e dei beep del buzzer senza ricorrere alla funzione *delay()* ma utilizzando *millis()*, naturalmente questo per evitare di usare del codice bloccante. Infatti vi ricordiamo che la funzione *delay()* è utilissima ma blocca in effetti il funzionamento del codice per tutto il tempo per cui si richiede il ritardo ed in alcuni casi questo è un comportamento che non possiamo tollerare.

Nel **Listato 3** vediamo il blocco di *Loop* in cui notiamo due sezioni di codice, la prima *Handle_Config()* gestisce la configurazione del dispositivo, mentre la seconda *Handle_Status()* gestisce la funzio-

▼ Listato 2 - Sezione di Setup del codice

```
void setup() {
    pinMode(POWER_ENABLE, OUTPUT);
    digitalWrite(POWER_ENABLE, HIGH);
    pinMode(LED1, OUTPUT);
    digitalWrite(LED1, LOW);
    pinMode(button.pin, INPUT_PULLUP);
    pinMode(TILT_SENSOR, INPUT);
    pinMode(BUZZER, OUTPUT);
    digitalWrite(BUZZER, LOW);
    m_status = 0;
    n_interval = 1;

    if (EEPROM.read(0) != 255){
        n_interval = EEPROM.read(0);
    }

    countDown = (unsigned long) n_interval * 3600
    * 1000L;
    notify_delay();
    attachPCINT(digitalPinToPCINT(TILT_SENSOR),
    irs_sub , FALLING);
    timer = millis();
    timer_led_tilt = millis();
    timer_buzzer = millis();
}
```

terval, *countDown*.

- *m_status* definisce lo stato in cui si trova il dispositivo. Gli stati possibili sono:

nalità normale dell'alimentatore, intercettando le vibrazioni del sensore e ritriggerando il timer monostabile e quindi attivando il regolatore di tensione Switching oppure quando scade il periodo di attivazione e non pervengono altre vibrazioni, portando il dispositivo in modalità LowPower. Nel **Listato 4** vediamo il codice che serve ad individuare la pressione prolungata o breve del pulsante (P1) di configurazione che è posto in prossimità del connettore USB di uscita.

Nel caso di pressione prolungata viene invocato il relativo gestore *handleLongPress()*. Nel caso di pressione breve viene invocato il gestore *handleShortPress()*.

Nel **Listato 5** vengono riportate le due funzioni di gestione della durata della pressione del pulsante. In *handleLongPress()* vediamo che se lo stato di funzionamento è quello normale (*m_status==0*) viene emesso un beep prolungato ed un blink prolungato, poi l'intervallo del timer retriggerabile viene impostato ad 1 e lo stato del sistema viene posto in configurazione. In pratica se l'alimentatore si trova nella modalità di funzionamento normale e vogliamo riconfigurare il periodo del timer ritriggerabile dobbiamo mantenere il pulsante il pulsante per un tempo superiore alla variabile *longPress* (5 secondi) entreranno quindi nella modalità di configurazione.

Se siamo nella modalità di configurazione entra quindi in gioco la funzione *handleShortPress()* ed infatti una pressione breve del pulsante: inferiore al valore della variabile *shortPress* (0.1 secondi) incrementerà il valore della variabile *n_interval* ad ogni pressione ed emetterà un breve segnale acustico e luminoso.

Quando riteniamo di aver settato il valore desiderato del Timer è necessario eseguire una pressione lunga del pulsante ed in questo caso il numero settato viene notificato tramite la funzione *notify_delay()*, che effettuerà un numero di lampeggi e segnali acustici brevi pari al numero dell'intervallo impostato, poi il valore impostato viene salvato nella EEPROM e lo stato viene riportato in modalità di funzionamento normale.

Nel **Listato 6** si può vedere la funzione di gestione dell'evento interrupt scatenato da una vibrazione percepita dal sensore. In pratica viene forzato a true la variabile *power_status* che è la variabile che determina l'attivazione del convertitore Switching, poi vengono settate alcune variabili al valore corrente di *millis()* che servono a gestire la durata del tempo di attivazione dell'alimentatore quindi la durata del timer retriggerabile ed altre due variabili

❏ Listato 3 - Sezione di Loop del codice

```
void loop() {
    Handle_Config();
    Handle_Status();
}
```

❏ Listato 4 - Parte del codice che gestisce la configurazione tramite pulsante

```
void Handle_Config(){
    button.currentState = digitalRead(button.pin);
    if (button.currentState != button.prevState) {
        delay(button.debounce);
        button.currentState = digitalRead(button.pin);
        if (button.currentState == PRESSED) {
            button.counter = millis();
        }
        if (button.currentState == NOT_PRESSED) {
            unsigned long currentMillis = millis();
            if ( (currentMillis - button.counter >=
shortPress) && !(currentMillis - button.counter
>= longPress) ) {
                handleShortPress();
            }
            if ((currentMillis - button.counter >= longPress)) {
                handleLongPress();
            }
        }
        button.prevState = button.currentState;
    }
}
```

❏ Listato 5 - Le due funzioni che gestiscono la pressione breve e prolungata del pulsante

```
void handleLongPress() {
    if (m_status == 0) {
        beepLong();
        blinkLong();
        n_interval = 1;
        m_status = 1;
    }
    if (m_status == 1 && n_interval >= 1) {
        notify_delay();
        EEPROM.write(0, n_interval);
        m_status = 0;
    }
}

void handleShortPress() {
    if (m_status == 1) {
        n_interval = n_interval + 1;
        beepShort();
        blinkShort();
    }
}
```

che vengono utilizzate per la gestione della durata della notifica del buzzer e del led di notifica della vibrazione, tutto questo per evitare la funzione *delay* che per sua natura bloccherebbe l'esecuzione

Elenco Componenti:

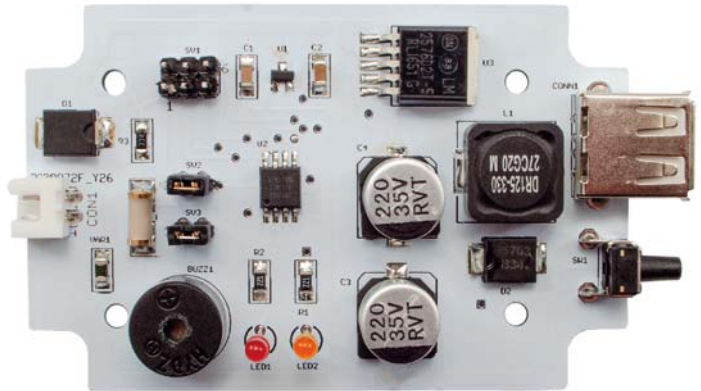
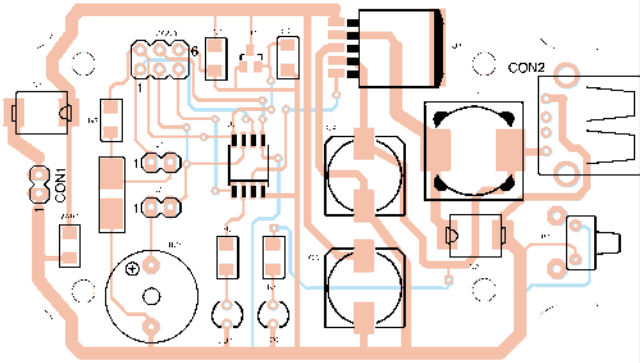
R1, R2: 100 ohm (1206)
 R3: 220 kohm (1206)
 C1: 1 μ F 25 V ceramico multistrato (1206)
 C2: 10 μ F 10 V ceramico multistrato (1206)
 C3, C4: 220 F 35 V elettrolitico
 D1, D2: Schottky SMC403
 LD1: LED giallo 3 mm
 LD2: LED rosso 3 mm
 L1: DR125-330-R 33 H 3 A

S1: Sensore vibrazione BL600
 VR1: Varistore ZV14K1206201NIR1HT
 BZ1Buzzer 12 mm - 5 V
 P1: Pulsante da c.s. 90°
 CON1: JST 2 poli M 90° P2,54
 CON2: Connettore USB A - Femmina da c.s.
 CON3 pin strip 2x3 M
 J1, J2 pin strip 2 vie M

U1 MCP1799T-5002H/TT
 U2 ATTINY85V-10SUR
 U3 LM2592HVS-5.0/NOPB

Varie:

- Jumper
- JST 2 poli F P2,54 con cavetti 20 cm
- Circuito stampato S1626 (80x50 mm)



del codice. Nel **Listato 7** si può vedere la porzione di codice che gestisce il funzionamento normale dell'alimentatore. Infatti come prima cosa viene verificato se la variabile `m_status == 0`, in tal caso se la variabile `power_status` è vera portato il pin `POWER_EN` ad un valore logico basso e questo abilita l'alimentatore switching (nel caso invece in cui la variabile non fosse =0, il pin `POWER_EN` viene portato alto e quindi il regolatore Switching verrebbe disattivato). Poi viene gestita la visualizzazione della notifica della vibrazione, ma cosa più importante viene verificato il tempo passato dall'ultima vibrazione percepita e se questo valore è superiore al massimo periodo (countDown) di durata del monostabile retriggevole il convertitore Switching viene disattivato ed il microcontrollore viene posto in modalità LowPower.

SCRIPT DI PROGRAMMAZIONE

Per agevolarvi nell'utilizzo immediato del progetto abbiamo pensato di rilasciare i precompilati dei codici e permettervi di caricarli direttamente, senza dovervi preoccupare di installare tutte le librerie necessarie, ed effettuare il setup dell'IDE di Arduino. Per caricare il firmware precompilato, apriamo la cartella "Script Programmazione", colleghiamo il programmatore al PC tramite il cavo USB in dota-

zione (attenzione, alcuni cavi che vengono usati per caricare i mobile sono solo cavi Power e quindi non sono adatti a questo scopo).

Nel momento in cui lo colleghiamo, dovremmo udire il classico segnale acustico associato al rilevamento di nuovo hardware, attendiamo il completamento della installazione degli eventuali driver. Completata l'installazione dei driver, colleghiamo il connettore maschio a 6 poli del programmatore Pololu al connettore CON3 maschio presente sul PCB dell'alimentatore. Prestare attenzione al verso di inserimento del connettore.

individuiamo quindi nella cartella lo script upload. Se il programmatore è stato installato con successo, lo script propone la lista delle porte COM trovate. Quella che interessa a noi è quella associata al dispositivo "Pololu AVR Programmer v2.1 Programming Port".

Digitate il numero di quella corretta e continuate nella selezione delle varie opzioni fino al completamento della procedura. Lo script fa uso del software AVRDUDE che normalmente dovremmo avere già installato e raggiungibile senza problemi da uno script DOS come quello che stiamo utilizzando. Lo script carica nel chip ATtiny85 il contenuto del file binario *TimerRetriggerable.ino.hex* che si trova all'interno della stessa cartella.

RICOMPILAZIONE DEL SORGENTE

Per chi volesse cimentarsi nella ricompilazione lo Sketch tramite IDE di Arduino, consigliamo di compilare il sorgente, poi tramite la voce di menu "Sketch/Esporta sketch compilato" vi troverete dentro la cartella dove è presente il sorgente anche l'eseguibile compilato. Il file ha lo stesso nome del sorgente ma termina con una estensione .hex ora questo file possiamo utilizzarlo per rimpiazzare quello che abbiamo fornito noi e che si trova dentro la cartella Script Programmazione, a questo punto è sufficiente eseguire la procedura indicata nel paragrafo precedente per riprogrammare il microcontrollore con la vostra versione del firmware.

REALIZZAZIONE PRATICA

Passiamo ora costruzione pratica: è necessario ottenere i PCB scaricando il file da www.elettronica.in.it. Potrete poi utilizzare il servizio di realizzazione PCB di Futura Elettronica.

Il PCB è a doppia ramatura ed essendo i componenti utilizzati di tipo SMD, la realizzazione pratica richiede un minimo di esperienza e una certa attrezzatura, vale a dire:

- Una stazione saldante che abbia un potenza almeno di 40-50W.
 - Una punta che, contrariamente a quello che si potrebbe essere portati a pensare non è necessario sia sottilissima. La cosa importante è che sia in buono stato (non ossidata) e che abbia la parte terminale corta, questo consente di trasferire velocemente calore dalla punta alla piazzola ed evita la realizzazione di saldature fredde.
 - Della lega di stagno (preferibile Sn-Ag) con flussante, da 0.5 mm o 1 mm.
 - Una pinzetta per prelevare e posizionare i componenti.
 - Una treccia dissaldante. Molto importante nel caso in cui vengano a crearsi dei ponticelli tra i pin più ravvicinati dei componenti.
 - Una lente di ingrandimento.
 - Per ultimo ma non meno importante... tanta pazienza ed una mano calda.
- Al solito, iniziate il montaggio dai componenti a basso profilo e proseguite in ordine di altezza.

CONCLUSIONI

Bene, abbiamo concluso l'esposizione dell'articolo dell'alimentatore per DashCam.

Speriamo che possa esservi utile nella installazione sulla vostra autovettura. Ma anche se non avete intenzione di usarlo per questo scopo, il codice

▼ Listato 6 - Porzione di codice che definisce la gestione dell'interrupt

```
void irs_sub(void) {
    power_status = true;
    led_tilt_status = true;
    timer = millis();
    timer_led_tilt = millis();
    timer_buzzer = millis();
}
```

▼ Listato 7 - Porzione di codice che gestisce la funzionalità in modalità normale

```
void Handle_Status(
    if (m_status == 0) {
        if (power_status){
            digitalWrite(POWER_EN , LOW);
        }
        if (led_tilt_status){
            digitalWrite(LED1, HIGH);
        }else {
            digitalWrite(LED1, LOW);
        }
        if (millis()- timer_led_tilt >= 10 ){
            led_tilt_status=false;
            timer_led_tilt = millis();
        }
        if ((millis() - timer) >= countDown) {
            power_status=false;
            sleep();
        }
    }else {
        digitalWrite(POWER_EN , HIGH);
        digitalWrite(LED1, LOW);}
}
```

e lo schema elettrico possono essere lo spunto per l'impiego in altri ambiti.

Il codice e le librerie utilizzate nell'articolo sono disponibili al seguente url di Github:

<https://github.com/open-electronics/shakepower> 



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. Il sensore di vibrazione (cod. VIBSENSBL600) è in vendita a Euro 2,50, il programmatore per AVR (cod. USBPROGAVR) è disponibile a Euro 17,00. Il PCB può essere realizzato utilizzando il servizio PCBPRODUCTION disponibile sul sito www.futuranet.it. 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



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



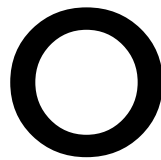
Trasformano
la notte 
in  giorno!



MISURARE LA VELOCITÀ DEL **SUONO** IN **ARIA**

di **VALERIO CONICELLA**

Utilizziamo un economico modulo ad ultrasuoni per misurare la velocità del suono con una scheda Arduino UNO.



Oggi gli ultrasuoni trovano largo impiego in ambito medico ed industriale e non solo. Vengono utilizzati nelle ecografie mediche; in ambito ferroviario per testare l'integrità delle rotaie; in molti apparecchi per la pulizia e, tra i diversi utilizzi, anche il sonar impiega intervalli di frequenze che rientrano nella

gamma degli ultrasuoni.

In queste pagine vedremo come misurare la velocità del suono in aria attraverso il modulo **HC-SR04**, un sensore ultrasonico per schede **Arduino UNO**, che si può acquistare al costo di pochi euro e che consente di misurare la distanza da un oggetto, attraverso l'emissione di un segnale sonoro, misurandone il tempo di risposta.

MA COSA SONO GLI ULTRASUONI?

Gli ultrasuoni o onde ultrasoniche sono delle onde meccaniche sonore non percepibili dall'orecchio umano, la cui frequenza è

Tabella 1
Esempi della velocità del suono in diversi materiali.

MATERIALI	VELOCITÀ DEL SUONO IN M/S
Aria	343
Acqua	1484
Vetro	5770
Acciaio	5900
Piombo	6300
Titanio	6100

convenzionalmente fissata al di sopra dei **20 kHz**. Questo permette di distinguere le onde ultrasoniche dalle onde soniche, la cui frequenza è appunto inferiore ai 20 kHz.

In generale, un'onda sonora è un'oscillazione compiuta dalle particelle (atomi e molecole) in un mezzo fisico di propagazione (es. aria). Le particelle, "disturbate" da una sorgente, si trasmettono energia a vicenda, propagandola.

Un aspetto molto importante è caratterizzato dalla velocità di propagazione dell'onda sonora, ovviamente data dalla distanza percorsa dall'onda per unità di tempo. La velocità di ogni onda dipende dalle proprietà del mezzo in cui essa viaggia.

Facendo riferimento alla **Tabella 1** si può notare come la velocità dell'onda sia direttamente proporzionale alla densità del mezzo di propagazione. Per esempio, un'onda sonora viaggerà circa quattro volte più velocemente nell'acqua che nell'aria.

Tabella 2
Velocità del suono in aria alle diverse temperature.

TEMPERATURA °C	VELOCITÀ DEL SUONO IN M/S
-10	325,4
-5	328,5
0	331,5
+5	334,5
10	337,5
15	340,5
20	343,4
25	346,3
30	349,2

Questo aspetto è dovuto essenzialmente al fatto che, nei mezzi più densi, le particelle che lo costituiscono sono più strettamente legate e, dunque, una sollecitazione meccanica si trasmette più facilmente. Un aspetto da non trascurare è il fatto che la velocità di un'onda sonora nell'aria, dipende anche dalla temperatura e dall'umidità. Nell'aria, infatti, la velocità del suono è di 331.2 m/s a 0 °C e di 343.1 m/s a 20 °C.

È possibile infatti, conoscendo la temperatura, ricavare con buona approssimazione la velocità dell'onda sonora, come espresso dalla seguente formula, dove la temperatura (T) è espressa in Kelvin.

$$v_T(T) = \sqrt{401,8 \cdot T} \frac{m}{s}$$

La **Tabella 2** fornisce un quadro della variazione della velocità del suono nell'aria al variare della temperatura.

SENSORI AD ULTRASUONI

I sensori a ultrasuoni servono per rilevare con affidabilità la posizione di un oggetto, misurare distanze o riconoscere sostanze solide, polverulenti o liquidi. Poiché i sensori a ultrasuoni rilevano gli oggetti utilizzando il suono anziché la luce, sono adatti ad applicazioni in cui i sensori fotoelettrici non potrebbero funzionare. I sensori a ultrasuoni rappresentano una soluzione ottimale per il rilevamento di oggetti trasparenti e per la misurazione del livello di liquidi, tutte applicazioni in cui i sensori fotoelettrici riscontrano difficoltà a causa della traslucenza del bersaglio. Di contro, può essere un problema un qualsiasi rumore acustico, con una frequenza alla quale il sensore a ultrasuoni sia sensibile e che può interferire e causare malfunzionamenti e imprecisioni di lettura.

CARATTERISTICHE DEL SENSORE HC-SR04

Ora vediamo brevemente le caratteristiche del modulo ultrasonico HC-SR04 (**Fig. 1**) disponibile da Futura Elettronica con il codice MISDIST04.

Questo sensore è composto da due trasduttori piezoelettrici: uno funge da trasmettitore (altoparlante) ed emette delle onde sonore ad una frequenza di circa **40 kHz**, mentre l'altro funge da microfono per ricevere le onde ultrasoniche rimbalzate su un eventuale oggetto.

Il sensore dispone di quattro pin, di cui due per l'alimentazione (**Vcc** e **Gnd**), uno per l'ingresso degli impulsi di trigger (**Trig**) e un pin per l'uscita degli impulsi (**Echo**). Fornendo un impulso positivo della

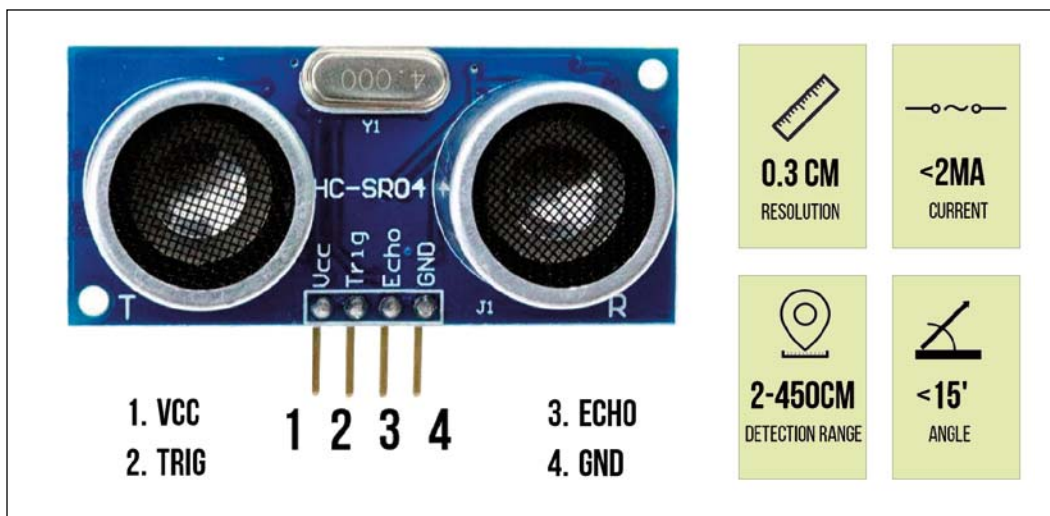


Fig. 1
Il sensore
HC-SR04.

durata di 10 μ s al pin di ingresso "Trigger", il sensore genera un treno di otto impulsi ad ultrasuoni a 40 kHz. Questi impulsi vengono inviati all'ostacolo, tramite la capsula trasmittente e la capsula ricevente rileva l'eco sul pin di uscita "Echo", dove sarà presente un impulso di durata variabile - da 150 μ s (2 cm) a 25 ms (4,5 m) - proporzionale alla distanza dell'oggetto. Se il sensore non rileva alcun ostacolo entro i 4,5 m, produce un impulso di durata di circa 38 ms.

ERRORI DI MISURA

Prima di entrare nel vivo dell'esperimento, è necessario introdurre qualche concetto base sugli errori di misura. Ogni misurazione è necessariamente accompagnata da errori; imprecisioni; limitazioni degli strumenti o errori commessi nel procedimento di calcolo. L'errore è per definizione la differenza tra il valore reale e il valore misurato della grandezza in esame. Gli errori di misura vengono suddivisi in due categorie: **errori casuali**, variano in modo imprevedibile da una misura all'altra, influenzando il risultato a volte per eccesso e a volte per difetto; gli **errori sistematici**, invece, avvengono sempre nello stesso senso: o sempre per eccesso, o sempre per difetto. Questi errori possono dipendere dalle qualità dello strumento di misura, da errori di metodo di misurazione o da errori che commette colui che misura. In ambito scientifico gli errori vengono definiti **incertezze**. Possiamo dire che, a causa dell'incertezza, il valore ottenuto da una misurazione è solamente una stima del valore reale di una grandezza fisica, pertanto, il risultato della misurazione non è un unico valore, bensì un insieme di valori derivati dalla misurazione diretta o indiretta della grandezza fisica.

L'incertezza può essere suddivisa in due categorie:

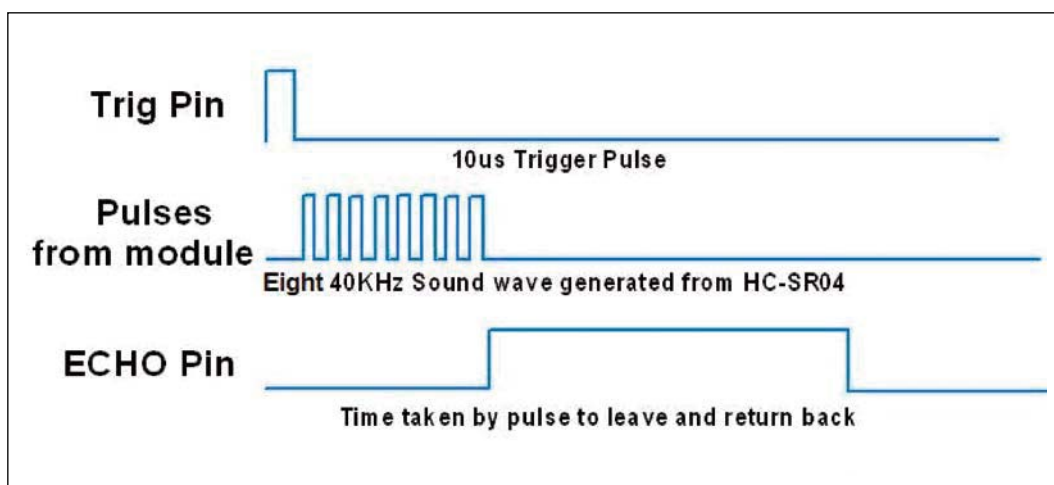
- **Incertezza di tipo A (u_A)**: viene stimata attraverso analisi statistica mediante misure ripetute;
- **Incertezza di tipo B (u_B)**: non può essere dedotta da misure ripetute, ma da altre informazioni (conoscenza del comportamento degli strumenti; specifiche tecniche del costruttore; dati forniti in certificati di taratura; ecc.).

In generale, quando si effettua la misura di una grandezza fisica X, il **risultato** si esprime come il valore medio dei valori osservati (X_m), tenendo conto dell'incertezza (s) e dell'unità di misura (u.m.):

CARATTERISTICHE TECNICHE HC-SR04



Fig. 2
Diagramma
temporale.



$$X = (X_m \pm s) u. m.$$

Senza andare troppo nello specifico, supponiamo di voler conoscere una grandezza fisica X , effettuando una serie di misure ripetute nelle stesse condizioni sperimentali. Indichiamo con N il numero di valori misurati ($x_1, x_2, \dots, x_k, \dots, x_n$), dove x_k è il generico valore di ogni x misurato. Definiamo allora, il **valore medio** per l'insieme dei valori misurati come segue:

$$X_m = \frac{1}{N} \sum_{k=1}^N x_k$$

Per poter stimare gli scostamenti (s), viene utilizzata la formula dell'**incertezza sperimentale** o **scarto tipo sperimentale**, che rappresenta un indice della dispersione delle misure attorno al valore medio:

$$s(X_m) = u(X_m) = \sqrt{\frac{1}{N(N-1)} \sum_{k=1}^N (x_k - X_m)^2}$$

SETUP DI MISURA

Prima di effettuare le misurazioni, è utile annotare la **temperatura** della stanza dove si sta svolgendo l'esperimento. Nel nostro caso misuriamo una temperatura pari a $T = 23,5^\circ\text{C}$.

In tali condizioni, secondo la relazione scritta in precedenza per ricavare la velocità dell'onda sonora, la velocità teorica del suono dovrà essere pari a

$$v_T(T) = 344,95 \text{ m/s}$$

A questo punto, possiamo procedere con la misura

Listato 1

```
int trigPin = 7;
int echoPin = 8;
int i = 0;
long t_sum = 0;
long t_sum_q = 0;
long t = 0;
float sigma = 0.0;
int nread = 40;
void setup()
{
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  digitalWrite(trigPin, LOW);
  Serial.println("Misura velocità del suono");
}
void loop()
{
  t = 0;
  t_sum = 0;
  t_sum_q = 0;
  for (i = 1; i <= nread; i++) {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(10);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    long duration = pulseIn(echoPin, HIGH);
    t = duration;
    t_sum = t_sum + t;
    t_sum_q = t_sum_q + t * t;
    delay(100);
  }
  t = (t_sum / nread) / 2;
  sigma = (sqrt(nread * t_sum_q - t_sum * t_sum) / nread) / 2;
  Serial.print(" Media sul tempo: ");
  Serial.print(t);
  Serial.print(" Incertezza: ");
  Serial.println(sigma, 1);
  delay(100);
}
```

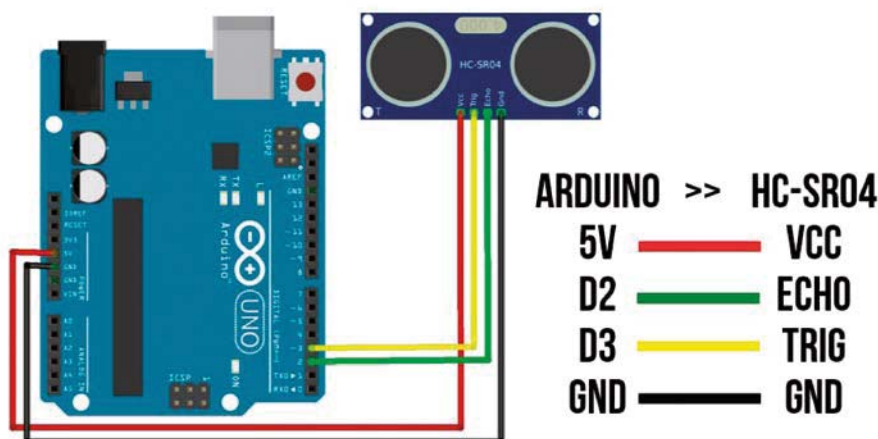


Fig. 3
Schema di
cablaggio.

del suono. Posizioniamo il sensore HC-SR04 a distanze sempre maggiori a partire da una parete piana e libera da ogni ostacolo. Attraverso il monitor seriale di Arduino UNO, leggiamo il valore del tempo di propagazione del segnale sonoro e l'incertezza associata.

Se durante la misura, notate che i valori acquisiti sono molto fluttuanti, potrebbe essere utile collegare in parallelo all'alimentazione del sensore, un condensatore elettrolitico di capacità non inferiore a 100 μF - 25 V.

Misuriamo, dunque, la distanza tra la parete e il sensore e appuntiamo i valori su una tabella. Ripetiamo le misure per almeno 10 distanze diffe-

renti, in modo da avere un numero di dati sufficienti a stimare la velocità del suono in aria in maniera più accurata e precisa possibile.

SCHEMA DI CABLAGGIO

Per l'esperimento servono i seguenti materiali:

- Scheda Arduino UNO;
- Sensore ultrasonico HC-SR04;
- Metro a nastro con portata di 1 m e sensibilità 1 mm;
- Cavi ponticello.

Il circuito può essere assemblato su una bread board, attraverso dei cavi ponticello maschio/femmina seguendo lo schema di cablaggio (**Fig. 3**) e

Tabella 3
Dati sperimentali raccolti.

	TEMPO T (S)	SCARTO TIPO (S)	DISTANZA D (M)	INCERTEZZA SULLA DISTANZA UD (M)	VELOCITÀ V (M/S)	INCERTEZZA SULLA VELOCITÀ UV (M/S)
1	0,000311	0,000001	0,10	0,002	321,54	6,51
2	0,000594	0,000001	0,2	0,002	336,70	3,41
3	0,000868	0,000001	0,3	0,002	345,62	2,34
4	0,001153	0,000004	0,4	0,002	346,92	2,11
5	0,001434	0,000002	0,5	0,002	348,68	1,48
6	0,001712	0,000004	0,6	0,002	350,47	1,43
7	0,001998	0,000005	0,7	0,002	350,35	1,33
8	0,002289	0,000005	0,8	0,002	349,50	1,16
9	0,002576	0,000004	0,9	0,002	349,38	0,95
10	0,002864	0,000001	1	0,002	349,16	0,71
				Media della velocità	344,8	

alimentando arduino con una tensione stabilizzata di 12VDC.

IL FIRMWARE

Commentiamo brevemente il codice sorgente che viene riportato interamente nel **Listato 1**.

Per prima cosa sono definite e inizializzate le variabili utilizzate nel codice. Nella funzione **setup()** viene inizializzata la porta seriale **Serial.begin()** e impostati i pin di I/O.

Nella funzione **loop()**, è implementato l'algoritmo che consente di calcolare il tempo e la deviazione standard. Attraverso il ciclo **for()**, la scheda Arduino effettua 40 misurazioni del tempo ad intervalli di 100 millisecondi.

A fine ciclo, viene assegnato alla variabile **t** il valore medio delle misure e alla variabile **sigma** il valore della deviazione standard. Successivamente vengono inviati sulla porta seriale di Arduino i valori ottenuti, attraverso la funzione **Serial.print()**.

ELABORAZIONE DATI

La prima cosa da fare dopo aver raccolto tutti i dati su un foglio di calcolo, sarà quella di procedere calcolando la velocità del suono per ogni singola distanza, attraverso la seguente formula:

$$v = \frac{d}{t}$$

Dove **d** sarà la distanza tra sensore e ostacolo e **t** il tempo di propagazione dell'onda sonora.

La valutazione dell'incertezza della velocità viene fatta applicando la *regola dell'incertezza scarto tipo* sul valore della velocità del suono, calcolato per ogni singola distanza. In questo modo avremo:

$$u_v = v \sqrt{\left(\frac{u_d}{d}\right)^2 + \left(\frac{u_t}{t}\right)^2}$$

dove u_v indica l'incertezza della misurazione della velocità, e u_d indica l'incertezza della distanza, posta dall'operatore pari a 0,002 m e infine u_t indica l'incertezza relativa al tempo. Facciamo un esempio prendendo i seguenti valori dalla **Tabella 3**:

$$d = (0,010 \pm 0,002) \text{ m}$$

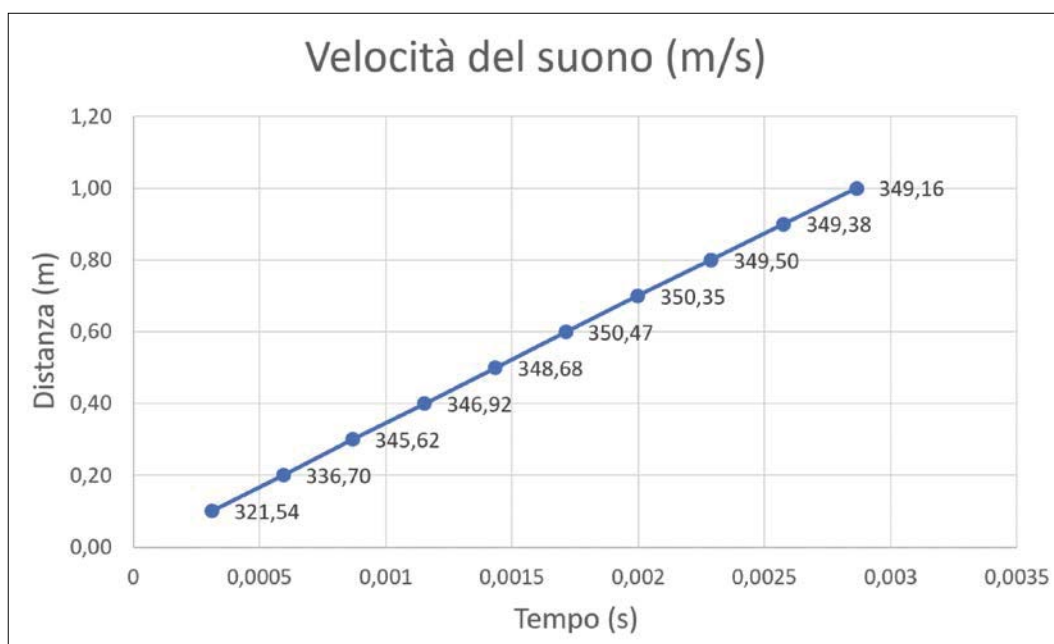
$$t = (311 \pm 1) \text{ us.}$$

Applicando la formula per il calcolo della velocità, otterremo:

$$v = (321,54 \pm 6,51) \text{ m/s}$$

Dalla **Tabella 3**, riportiamo graficamente nella **Fig. 4**, sull'asse delle ascisse (x) il tempo **t**, espresso in secondi e sull'asse delle ordinate (y) la distanza **d**, espressa in metri.

Fig. 4
Grafico dati sperimentali.



Notiamo che la relazione tra le due variabili è di tipo lineare, dato che l'andamento è una retta. Questo conferma ciò che ci aspettavamo, poiché la formula della velocità è una relazione lineare. Inoltre, tra le due grandezze, vediamo che la proporzionalità è diretta, questo implica che il loro rapporto è costante:

$$\frac{y}{x} = m \rightarrow \frac{d}{t} = v$$

Vediamo come il coefficiente angolare della retta **m** risulta essere proprio la velocità.

Facendo la media delle 10 osservazioni della velocità per le diverse distanze, si può osservare che il valore medio (344,8 m/s) ottenuto, si avvicina molto al valore teorico calcolato in precedenza

$$v_T(T) = 344,95 \text{ m/s}$$

Si potrebbe notare come le prime misurazioni del valore della velocità del suono siano più basse rispetto al valore teorico, in corrispondenza di una distanza inferiore ai 20 cm. Questo aspetto deriva dal fatto che probabilmente gli errori di misura hanno una maggiore incidenza su distanze e tempi brevi.

CONCLUSIONE

Lo studio degli ultrasuoni rappresenta un ambito molto interessante, proprio per i molteplici utilizzi in cui viene applicato.

Il fine ultimo di questo approfondimento, è stato quello di stimare la velocità del suono attraverso un semplice esperimento. La stima è stata effettuata utilizzando un sensore ultrasonico e una scheda Arduino, che ha permesso di stimare la velocità del suono in maniera molto precisa e accurata, tenendo conto di aspetti fondamentali per la misurazione, come la temperatura dell'ambiente in cui si è svolto l'esperimento, dato che, come abbiamo potuto vedere, la velocità di ogni onda dipende dalle proprietà del mezzo in cui essa viaggia. Proprio perché, nella misurazione della velocità del suono, gli aspetti che potrebbero incidere sono molteplici (ad esempio, oltre alla temperatura, anche il livello di umidità), si è doverosamente introdotto anche il concetto imprescindibile degli errori di misura. L'esperimento ci ha permesso di approfondire l'aspetto fisico che sta alla base del funzionamento dei sensori ad ultrasuoni, e di descriverlo nella maniera più semplice possibile. □


RUTRONIK
 ELECTRONICS WORLDWIDE





DDR5 SODIMM INDUSTRIAL MEMORY MODULE

- Next-generation DDR5 SODIMM industrial memory module, with a memory transfer rate of up to 4800 MT/s
- Uses original high-quality next-generation DDR5 DRAM lcs
- Equipped with power management IC (PMIC) (5V) on DIMM to efficiently control system power load
- On-die ECC error correction mechanism improves reliability
- 262-pin, DDR5 small outline dual in-line memory module
- VDD = VDDQ = 1.1V
- VPP = 1.8V
- 32 internal banks; 8 groups of 4 banks each
- Lead-free (RoHS compliant)
- Halogen free
- Conformal Coating / Underfill (optional)
- Anti-sulfuration (Apacer patented) (optional)

More information about DDR5 SODIMM:
 Phone +49 7231 801 4613 | richard.klauser@rutronik.com

www.rutronik.com





COMMITTED TO EXCELLENCE
 Consult | Components | Logistics | Quality

**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

SERRA DOMOTICA PER COLTURA IDROPONICA

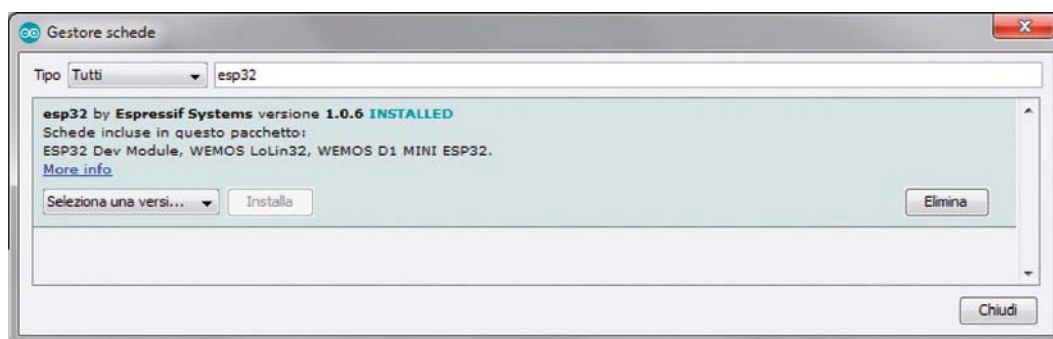


di **SAMUELE BRUNINI**

Costruiamo una serra domotica (comandata da Blynk e relativa app) con illuminazione artificiale a LED specifici per ottimizzare la crescita delle piante contenute, specie utilizzando la tecnica idroponica. Seconda e ultima puntata.

Ci siamo lasciati, nel fascicolo precedente, dopo aver presentato il nostro sistema di coltivazione idroponica con illuminazione allo stato solido ottenuta da LED che combinano varie luci per ottimizzare la crescita dei vegetali. Abbiamo spiegato che la nostra serra si basa su un controller M5Stack Basic Core IoT KIT programmabile con l'IDE Arduino e dotato di display LCD grafico a colori da 320x240 pixel, oltre che di transceiver WiFi che ci consentirà di gestire il progetto da smartphone mediante un'interfaccia utente creata e appoggiata a Blynk. Bene, descritta la parte di configurazione di Blynk e la creazione della nostra interfaccia su smartphone, procediamo con la programmazione del controller, che avverrà creando e caricando uno sketch tramite l'IDE Arduino, che andrà preliminarmente impostato per il nostro scopo.

Fig. 1
Aggiunta del driver
di terze parti del
controller.



PREPARAZIONE IDE ARDUINO

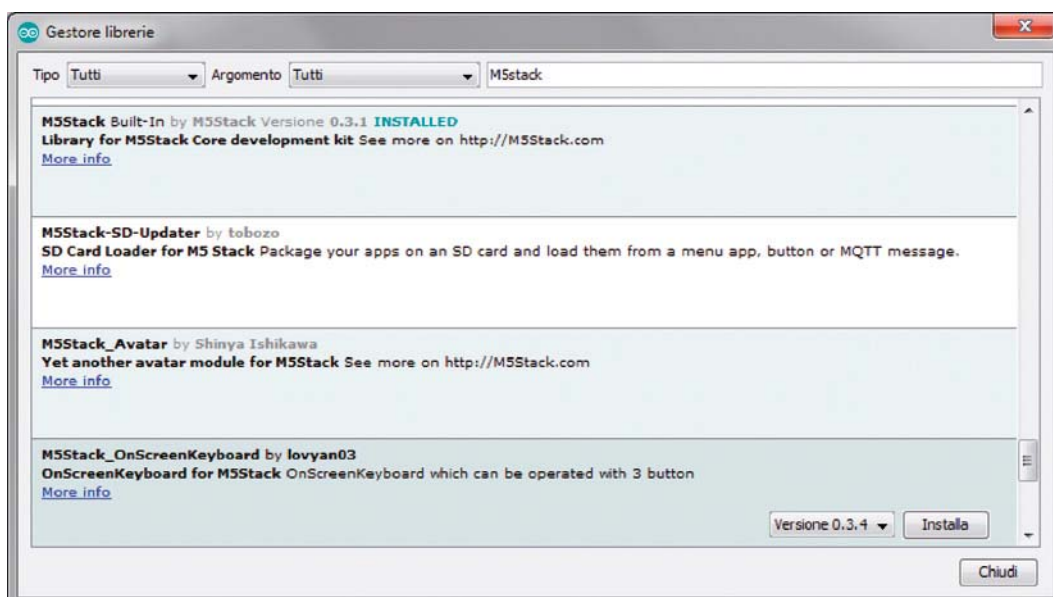
Prima di aprire l'IDE di Arduino bisogna inserire le librerie che verranno utilizzate per compilare il progetto, quindi è necessario copiare le librerie contenute nella cartella "#lib" in "C:\Users\utente\Documents\Arduino\libraries". Prima di caricare lo sketch è necessario preparare la memoria Flash del dispositivo, che verrà partizionata per ospitare sia lo sketch che le immagini. Per fare ciò si andrà a installare un add-on che farà questo lavoro per noi; dunque, come prima operazione copiare la cartella "ESP32FS" che si trova nello zip del progetto, nella cartella di installazione dell'IDE di Arduino sotto a "tool", ottenendo così: "C:\Program Files (x86)\Arduino\tools\ESP32FS\tool\esp32fs.jar". Apriamo *File > Impostazioni* e andiamo a inserire l'URL https://dl.espressif.com/dl/package_esp32_index.json nella casella *URL aggiuntive per il gestore schede* in modo da indicare il percorso di ricerca dei driver. Una volta confermata la configurazione, dal menu "Strumenti > Scheda > Gestore Schede", andate a

cercare "esp32" e procedete all'installazione cliccando sul pulsante *Installa* (**Fig. 1**). Ora da "Sketch > #include libreria > Gestione Librerie" cercate "M5stack" e procedete all'installazione, come mostrato nella **Fig. 2**. Ad installazione completata, da "Strumenti > Scheda" selezionate "M5Stack-Core-ESP32", quindi impostate "Upload Speed" a "921600".

INIZIALIZZAZIONE FILE SYSTEM

Prima di aprire lo sketch, dovete verificare che all'interno della cartella "GrowingStation" vi sia la cartella "data" con all'interno tre immagini: questa cartella servirà all'add-on installato nello step precedente per formattare e caricare le immagini nella memoria flash del dispositivo. Adesso è possibile aprire il file "GrowingStation.ino", cliccare su "strumenti" e poi "ESP32 Sketch data Upload". Fate attenzione che il monitor seriale non sia aperto, perché in tal caso andrebbe a bloccare la porta COM e il processo non andrà a buon fine.

Fig. 2
Installazione del
controller nell'IDE
Arduino.



CONFIGURAZIONE FIRMWARE

Tramite IDE di arduino (1.8.9 o superiore) aprite lo Sketch "GrowingStation.ino" facendo attenzione che insieme ad esso venga aperto anche "Free_Fonts.h" e "OTA.h". Se ciò non accadesse, dovete verificare di avere la corretta struttura della cartella dove c'è il firmware, che è quella proposta nella **Fig. 3**. Una volta aperto correttamente lo sketch, sarà molto importante personalizzare 5 righe di codice, procedura obbligatoria per consentire la corretta connessione del dispositivo alla rete WiFi, e quindi di conseguenza al servizio Blynk. Come prima cosa dovete sovrascrivere le prime tre righe del codice con quelle ottenute in precedenza tramite la scheda "Device Info" della Dashboard. Successivamente andate a modificare anche la variabile "ssid" e "password" con i dati di accesso corretti per potere accedere via WiFi alla connessione internet (**Fig. 4**). Ora tutto il codice necessario è stato modificato, pertanto si potrà procedere con la compilazione e quindi il caricamento del firmware sul proprio dispositivo. Assicuratevi quindi che il cavo USB sia collegato al proprio PC e la porta COM risultante sia stata selezionata dal menu "Strumenti > Porta" dell'IDE. Come scheda selezionate "M5Stack-Core-ESP32" (**Fig. 5**). Fatto ciò, cliccate sul secondo tasto della barra menu dell'IDE per consentire la compilazione e il caricamento del firmware sulla scheda. Ora che avete programmato il controller potete passare agli aspetti hardware del progetto e successivamente alla messa in esercizio.

PREPARAZIONE DELL'HARDWARE

Prima di realizzare il cablaggio dell'insieme è opportuno calibrare il DC/DC converter step-down, perché tale modulo è regolabile e ci occorre che in uscita fornisca una determinata tensione e che la corrente erogata non superi i 4A. Quindi collegate pure tutti gli elementi del progetto ma lasciate momentaneamente disconnessa l'uscita del DC/DC.

#lib	16/02/2022 10:24
data	11/02/2022 10:29
Free_Fonts.h	03/02/2021 16:34
GrowingStation.ino	16/02/2022 12:20
OTA.h	03/02/2021 16:34

Fig. 3
File necessari alla configurazione firmware.

```
GrowingStation | Arduino IDE
File Modifica Sketch Strumenti Auto

GrowingStation.ino
1 #define BLYNK_TEMPLATE_ID "TMPL6cdW0Gc"
2 #define BLYNK_DEVICE_NAME "Serra"
3 #define BLYNK_AUTH_TOKEN "gUCdKlhj1ZFPwVFKlWNLyAkUcV98D"
4 #define BLYNK_FIRMWARE_VERSION "0.1.0"
5
6 #define DEBUG //commenti di debug
7 #ifdef DEBUG
8 #define DPRINT(...) Serial.print(__VA_ARGS__)
9 #define DPRINTLN(...) Serial.println(__VA_ARGS__)
10 #else
11 #define DPRINT(...) //now defines a blank line
12 #define DPRINTLN(...) //now defines a blank line
13 #endif
14
15 #include <M5Stack.h>
16 #include <FS.h>
17 #include <SPIFFS.h>
18 #include <Free_Fonts.h>
19 #define BLYNK_PRINT Serial
20 #include <EEPROM.h>
21 #include <WiFi.h>
22 #include <ESP8266.h>
23 #include <WiFiUdp.h>
24 #include <WiFiClient.h>
25 #include <BlynkSimpleEsp32.h>
26
27 #include <TimeLib.h>
28 #include <Time.h>
29 #include <ArduinoOTA.h>
30
31 char ssid[] = "ssid";
32 char pass[] = "password";
33 #include <OTA.h>
34
35 //define pin led
```

Fig. 4
Le righe da sovrascrivere.

A questo punto prendete un multimetro disposto sulla misura di tensioni continue con fondo-scala di almeno 20V e toccate con i suoi puntali i morsetti di uscita del modulo step-down, quindi date tensione all'alimentatore principale e ruotando il trimmer P1 del circuito fate in modo che il tester legga una tensione più prossima possibile ai 10V. Togliete tensione al sistema e disponete il multimetro alla misura di correnti con fondo-scala di 10 ampere, quindi collegate i puntali ancora ai morsetti di uscita in modo da fare cortocircuito; alimentate di nuovo il tutto e ruotando il trimmer



Fondo della vasca che conterrà l'acqua: a sinistra vedete l'elettropompa, che recupera l'acqua dal fondo e la immette nel diffusore.

Tabella 1
Collegamenti del controller.

M5STACK	DRIVER BOARD LED (CONTROL)	DRIVER BOARD LED (LED SIDE)
GND	1	
16	2	1 – LED BIANCO
17	3	2 – LED UV
5	4	LED ROSSOBLU
26	5	LED BLU

P2 impostate l'output del modulo a 4A. Ora potete togliere nuovamente tensione e sconnettere il tester, connettendo i morsetti d'uscita del DC/DC al resto del circuito, secondo quanto previsto dallo schema di cablaggio.

La regolazione descritta serve per garantire ai LED la giusta tensione e soprattutto limitare la corrente per evitare che possano assorbire più del dovuto e conseguentemente degradarsi/bruciarsi alla svelta.

Per quando riguarda il cablaggio del controller con alimentazione e driver Colibrì, vale la **Tabella 1**.

AVVIO DEL SISTEMA

Terminato il caricamento dello sketch e fatti i collegamenti elettrici secondo lo schema di cablaggio, è possibile avviare il progetto di Blynk. Per poter creare un collegamento diretto al dispositivo sulla schermata del proprio Smartphone tenere il dito

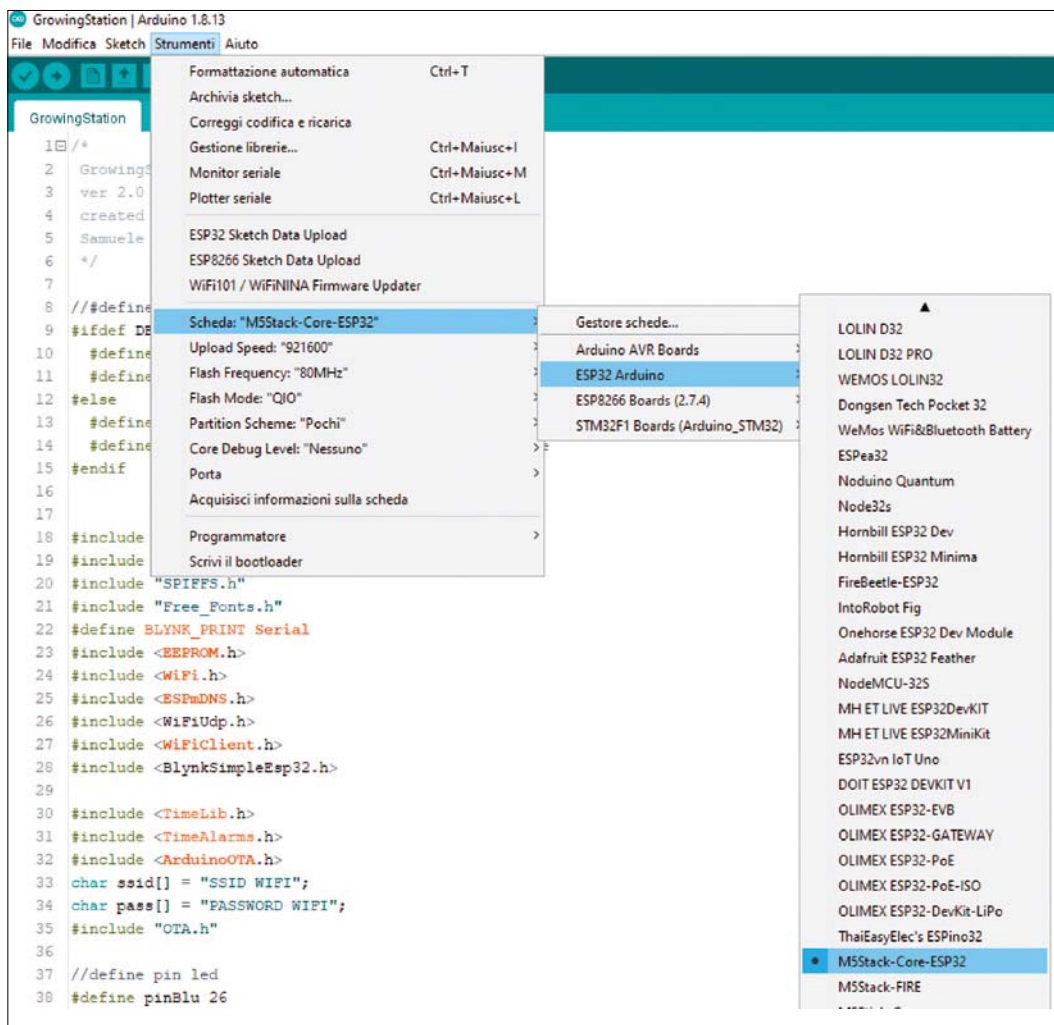


Fig. 5
Selezione della scheda da programmare.



← Vasca con applicato il corpo superiore nel quale si trova il diffusore (è il tubolare trasversale che si vede a destra) che spruzza l'acqua nei canali della piastra nera dove si appoggeranno le radici.

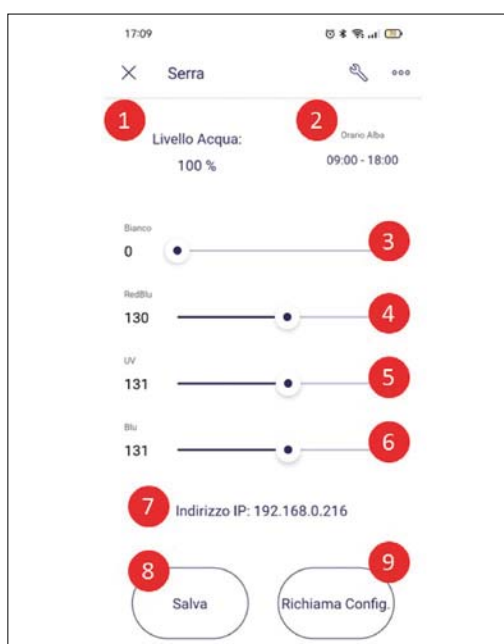
premuto sull'icona Blynk IoT, quindi non appena compare il menu a comparsa, rilasciare e premere sull'icona "Serra" e trascinarla dove si desidera. Da ora, si potrà pigiare sull'icona "Serra" per lanciare l'interfaccia di gestione. A riguardo, di seguito riportiamo i dettagli delle funzioni disponibili. Con riferimento alla **Fig. 6**, le aree dell'interfaccia utente sono le seguenti:

- 1 Indicazioni del livello di acqua (0-100%)
- 2 Regolazione dell'orario accensione luci
- 3 Regolazione luce Bianca
- 4 Regolazione luce RedBlu
- 5 Regolazione luce UV
- 6 Regolazione luce Blu
- 7 Visualizzazione indirizzo IP locale
- 8 Permette di salvare il livello di luci da eseguire durante il periodo Alba
- 9 Richiama la configurazione delle luci per la fase Alba per procedere alla eventuale modifica.

Il dettaglio delle schermate di configurazione e gestione della serra in locale del controller è riportato nell'apposito riquadro, dove descriviamo le sei pagine. Nel circuito è inserito un sensore di misura a ultrasuoni (cod. JSN-SR04T) che serve a rilevare quando il livello dell'acqua nella vasca è basso e va ripristinato. Questo sensore dev'essere posizionato in maniera che segnali un allarme al controller, il quale invierà a display una segnalazione per avvertirci di aggiungere acqua, ad evitare che le radici delle nostre colture idroponiche rimangano con poca irrigazione. Nel nostro caso il sensore è stato applicato al tetto del mobile e si affaccia verso il basso attraverso un tubo di plastica (voluti per evitare che le piante, crescendo, possano mettersi in mezzo al flusso di ultrasuoni disturbando il rilevamento) fin dentro la vasca; il segnale di distanza è poi elaborato dal firmware in modo che corrisponda alla minima altezza dell'acqua nella vasca. Il livello viene peraltro mostrato in alto a destra nel display del controller, in forma percentuale: laddove scenda sotto il minimo, il controller darà un avviso locale e

uno remoto sotto forma di notifica sull'app Blynk. Il modulo a ultrasuoni è in grado di rilevare distanze fino a 4,5 metri con una precisione di 0,5 cm (quindi più è distante, meno errore commette) e la sua uscita fornisce un impulso, quando il dispositivo viene triggerato dal controller, la cui durata esprime la distanza: 150 μ s corrispondono a 2 cm e 25 ms a 4,5 m. Se non rileva alcun ostacolo entro i 4,5 m, produce un impulso di circa 38 ms. Per calcolare la distanza in cm dovete dividere la durata dell'impulso in μ s per 58, mentre in pollici la divisione è per 148. In base alla distanza alla quale porrete il trasduttore a ultrasuoni (la capsula nera) dalla superficie dell'acqua che considererete come livello normale nella vaschetta dovete agire sullo sketch, alla riga:

```
sens=map(duration, 2350, 2200, 0, 100);
```



← **Fig. 6**
Dettagli dell'app per il controllo della serra.

LE SCHERMATE DEL MODULO CONTROLLER



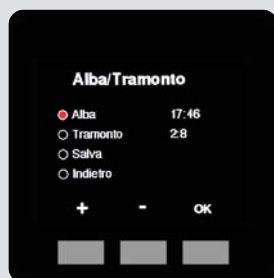
Questa è la schermata di accensione del dispositivo, una volta che esso si è collegato al wifi passerà automaticamente alla schermata successiva



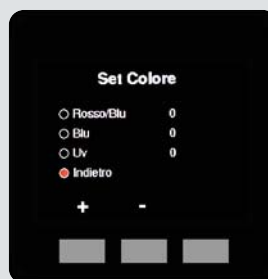
Questo è il menu principale che è così composto:

- Alba/Tram
- Colore
- Luminosità
- Luce

Da questo menu ci si può spostare dall'alto verso il basso e viceversa con i tasti + e - e selezionare i menu con il tasto OK (il primo tasto a destra)



Questo è il menu **Alba/Tramonto** da esso è possibile impostare l'ora dell'alba (orario di accensione di led) e l'ora del tramonto (orario di spegnimento dei led). Una volta impostati i valori desiderati spostarsi su "Salva" e cliccare ok per rendere definitive le impostazioni, invece se si volessero eliminare le modifiche spostarsi su "Indietro" e cliccare "OK"

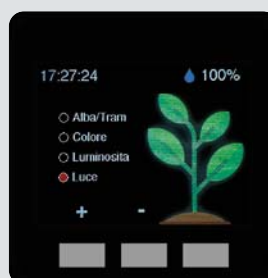


Questo è il menu **Set Colore** da esso è possibile impostare l'intensità del colore dei led, mentre si stanno modificando questi parametri i led si accenderanno in base al valore attuale impostato per permettere di impostare la luce il più preciso possibile. Una volta terminati i vari settaggi spostarsi su "Indietro" e cliccare il tasto "OK" per confermare i cambiamenti



Questo è il menu **SET LUMINOSITÀ** da questo menù è possibile impostare la luminosità del display con i tasti + e -

Una volta scelta la luminosità desiderata cliccare il tasto indietro per confermare l'impostazione (il parametro viene mantenuto anche al riavvio).



L'ultima voce del menù principale è: **Luce** una volta che ci si è spostati su di essa alla pressione del tasto ok corrisponderà un livello di intensità della luce bianca fino a 5 al sesto clic la luce si spegne. La luce bianca è utilizzata come luce di compagnia

della sezione *int lettura_Acqua()*. A 2200 corrisponde il 100% del range (25 ms = 4,5m) e a 2350 il minimo (150 ms, ossia 2 cm). Nel nostro prototipo la vasca piena ha l'acqua a 35 cm dal sensore, quindi 35 cm = 2200; se fate metà distanza impostate 1100 e via di seguito. Concludiamo con alcune note riguardanti la vasca contenente l'acqua, che abbiamo realizzato per il nostro progetto; come accennato nella prima puntata, utilizziamo la tecnica del riciclo, nel senso che facciamo ricircolare l'acqua dal fondo della vasca alla parte superiore. La vasca ha una parte inferiore dove "pesca" un tubetto di gomma collegato all'ingresso dell'elettropompa; l'uscita spinge l'acqua con un altro raccordo in gomma all'interno di un diffusore o erogatore, che è il tubo trasversale proposto nella foto in alto in questa pagina, il quale la spruzza sul letto scanalato nero che costituisce un microcircolo dove si appoggeranno le radici per trarre l'irrigazione. Da qui l'acqua in eccesso ricade sul fondo per essere ripescata. □

Cosa occorre?

I componenti principali utilizzati in questo progetto sono disponibili presso Futura Elettronica. Il modulo con display M5Stack Basic Core IoT (cod. M5ESP32BASICKIT) è disponibile a Euro 42,00, il sensore di misura ad ultrasuoni (cod. JSN-SR04T) è in vendita a Euro 17,50, il driver di potenza per LED (cod. COLIBRI) costa Euro 24,00, il LED grow da 10 W BLU (cod. LEDGROWBLU) è disponibile a Euro 3,80, quello UV (cod. LEDGROWUV) è in vendita a Euro 7,00, quello rosso/blu (cod. LEDGROWGY) è disponibile a Euro 3,80, quello bianco (cod. LED10WBIANCO) è in vendita a Euro 3,80, il driver a mosfet (cod. MOSDRIVER) costa Euro 2,90, la pompa per l'acqua (cod. MINIPOMPA12V) è disponibile a Euro 16,90, il convertitore DC/DC (cod. MODULODCDC) è in vendita a Euro 4,30, il modulo Step-Down da 5V (cod. MP1584ENDCDC) è disponibile a Euro 2,50, quello di potenza da 5A (cod. STEPD5V32) è in vendita a Euro 5,90, l'alimentatore in grado di gestire tutta la serra (cod. AP12M5) costa Euro 22,00. 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

MINI TRAPANI per ogni esigenza

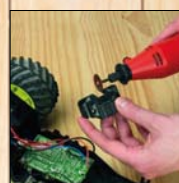
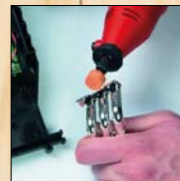


Mini Trapano con set per incisione - 162 Pezzi

cod. **VTHD0051**
€ 47,00



Indispensabile per piccoli lavori di rifinitura. Fornito in una pratica valigetta plastica comprendente una prolunga flessibile ed un ricco set di accessori utili per forare, smerigliare e lucidare. Alimentazione 230 VAC/50 Hz; potenza 135 W; velocità regolabile 10.000-32.000 rpm; interruttore ON/OFF.



Prezzi IVA inclusa.

Mini Trapano con accessori



cod. **TP212**
€ 29,00

Mini trapano con velocità di rotazione di 18.000 rpm adatto per lavori di precisione. Diametro testa mandrino: da 0,5 a 3 mm.

Avvitatore/trapano con torcia e impugnatura ruotabile

Compatto e potente avvitatore/trapano con batteria ricaricabile agli ioni di litio da 3,6 volt 1300 mAh ricaricabile tramite caricabatteria da rete incluso. Corredato di inserti, punte e prolunga per inserti.



cod. **WT71880**
€ 23,90

Mini Trapano con alimentazione USB



cod. **VTSUSB5**
€ 13,90

Mini trapano con velocità di rotazione di 15.000 giri/minuto, 6 differenti punte e alimentazione USB.

Set di accessori per incidere, forare, smerigliare, fresare, avvitare



Pratica confezione comprendente 188 accessori per incidere, forare e smerigliare. Da utilizzare con i trapani della serie VTHD.

cod. **VTHDS6**
€ 22,50



cod. **VTHDS2N**
€ 5,80

Set 20 frese diamantate di varie forme
- diametro
codolo: 3 mm
- lunghezza:
45 mm.



Valigetta contenente un set di 29 punte giravite in acciaio S2, supporto magnetico a sgancio rapido, contenitore con clip rotante da cintura.

cod. **VTBT17**
€ 15,00

INCUBATRICE AUTOMATICA CON ESP32-CAM

Incubatrice per uova di gallina completamente automatizzata con controllo di temperatura, umidità e rotazione delle uova; il sistema è monitorabile tramite un bot Telegram con l'invio di messaggi e immagini.



di **GABRIELE BIANCHI**

È

la natura che controlla l'uomo o l'uomo che controlla la natura? Creare la vita con la tecnologia è possibile?

Magari leggendo queste domande vi sarà balzata alla mente l'immagine di

uno scienziato che nella luce soffusa del suo laboratorio crea un mostro urlando frasi del tipo "Si può fare!"; purtroppo (o per fortuna) non vi stiamo proponendo di emulare le imprese del dottor Frankenstein dell'omonimo romanzo di Mary Shelly; proveremo invece, con la tecnologia, a replicare quello che una chioccia fa da migliaia di anni, ovvero covare le uova per dare alla luce dei pulcini. Impresa di certo meno grandiosa, ma con un livello di rischio notevolmente inferiore! L'incubazione non è affatto un processo innovativo, in commercio sono presenti macchine di ogni tipo atte a questo scopo, più o meno professionali; noi abbiamo tentato di costruirne una che fosse la più tecnologica e smart possibile. L'equazione è semplice: entrano uova fecondate, acqua, corrente elettrica ed escono dei pulcini. In realtà in questa formula manca un termine che non va trascurato, il tempo: 21 giorni per la precisione, periodo in cui non faremo altro che aspettare e monitorare il processo.

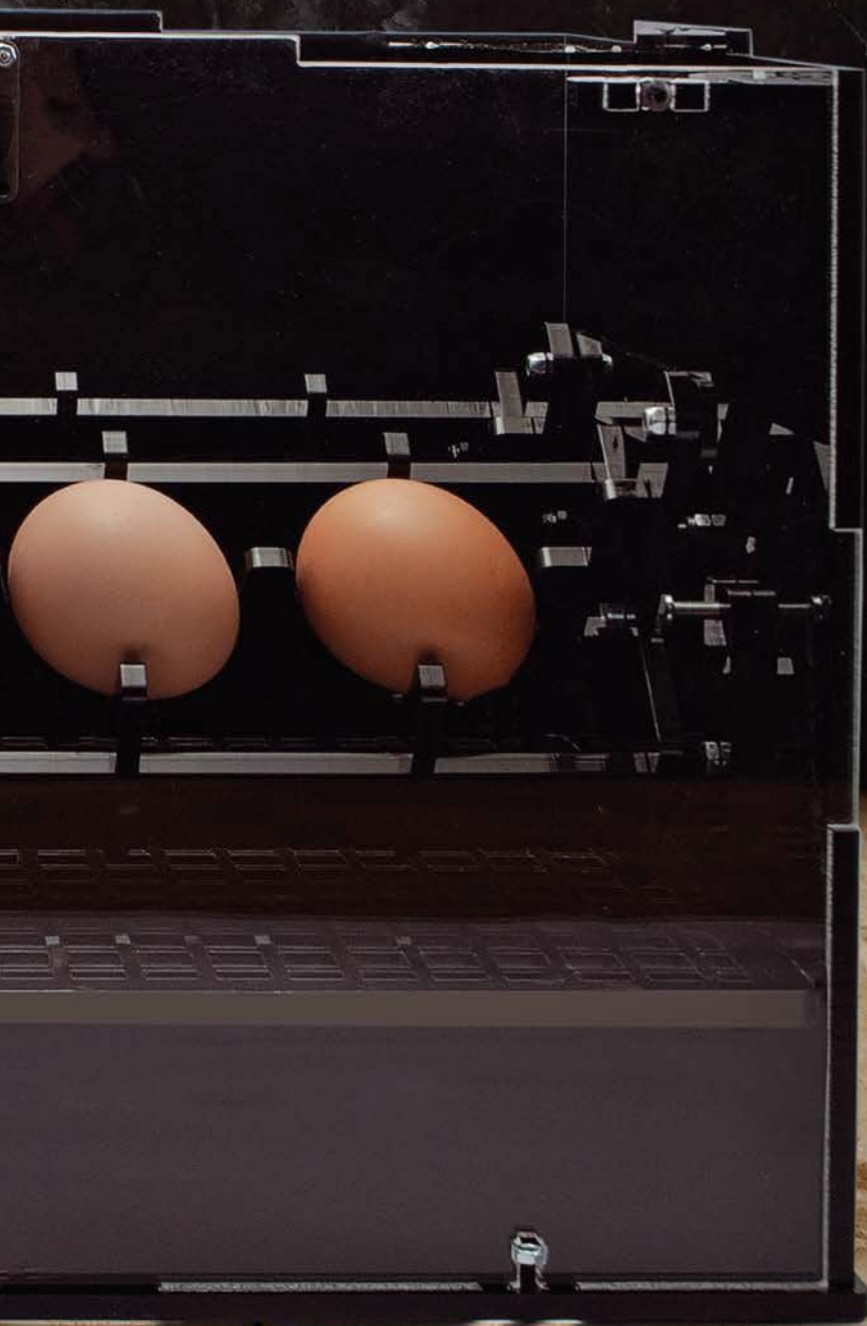
Nota molto importante da tenere in considerazione: bisogna assicurarsi che le uova siano fecondate, quindi a monte servono per forza un gallo e una gallina; per quanto l'incubazione sia tecnologica e automatica non può ancora prescindere del tutto dalla natura.

COME FUNZIONA

La struttura si presenta come una scatola in plexiglass con una parete trasparente attraverso cui è possibile "sbirciare" l'eventuale nascita dei pulcini.

Per capire cosa deve fare un'incubatrice è necessario analizzare il comportamento della chioccia che vogliamo replicare artificialmente.

Il processo di cova è relativamente semplice: la gallina si siede sulle uova fecondate e resta in posizione per 21 giorni (lascia il nido solo per brevi periodi), girando regolarmente le uova per



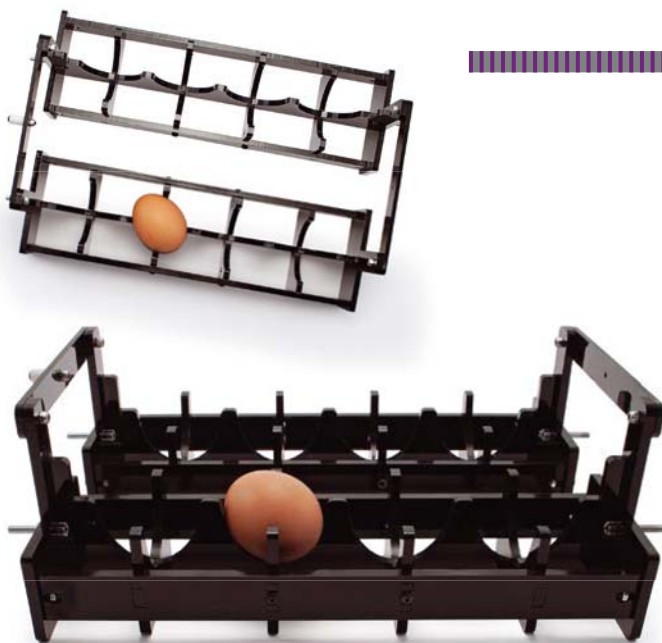


Fig. 1
Portauova
estratto
dall'incuba-
trice con un
uovo all'in-
terno nella
posizione
corretta.

evitare che l'embrione si attacchi alla membrana del guscio. Dopo 18 giorni i pulcini tenderanno di forzare il guscio dall'interno e ne impiegheranno circa 3 o 4 per uscire.

Dobbiamo quindi ricreare l'ambiente in cui stanno le uova sotto alla chioccia e inoltre simulare una rotazione delle stesse. La temperatura da mantenere è di circa $37,7^{\circ}\text{C}$ con un'umidità attorno al 55%. Nella fase di schiusa (dal giorno 18 in poi) è indicato, nell'incubazione artificiale, abbassare la temperatura a circa 37°C e alzare l'umidità portandosi attorno al 70%; questo è l'unico momento in cui è necessario intervenire manualmente. Ora che abbiamo analizzato teoricamente le condizioni richieste, vediamo come funziona nella pratica l'incubatrice.

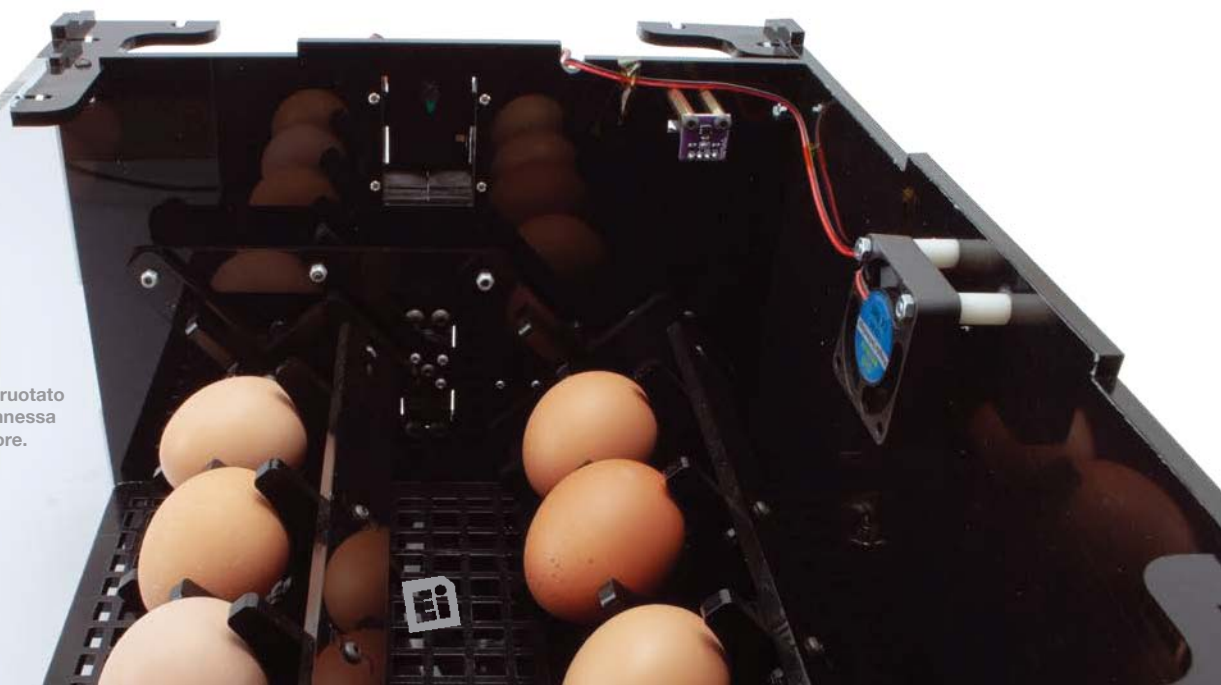
La macchina permette di incubare 8 uova che vengono posizionate verticalmente con la "punta" rivolta verso il basso in un apposito portauova rotante come si può vedere in **Fig. 1**.

Ogni 3 ore questo effettuerà una rotazione di circa 40 gradi rispetto all'asse verticale. Per mantenere l'umidità è presente una vaschetta sul fondo che viene riempita automaticamente pescando acqua da una sorgente posta all'esterno; è possibile comunque regolarne il valore manualmente (almeno la prima volta e nel periodo di schiusa) aprendo o chiudendo una piccola finestrella atta allo scopo. Basta un minimo spiffero di aria dall'esterno per modificarne sensibilmente lo stato.

I dati di temperatura e umidità sono monitorabili attraverso un bot Telegram tramite cui è possibile accendere un piccolo LED interno montato sul microcontrollore (un ESP32-CAM) e scattare una fotografia; quindi, anche se fuori casa, si può tenere la situazione sotto controllo.

All'inizio della fase di schiusa la macchina va aperta, si tolgono le uova dal portauova e si appoggiano delicatamente sulla grata di fondo (anch'essa in plexiglass), dopo di che si può regolare l'umidità chiudendo leggermente la finestrella sul lato ed eventualmente si può intervenire sul codice per modificare il set point di temperatura portandolo da $37,7^{\circ}\text{C}$ a 37°C (azione in realtà non strettamente necessaria). Nei 3 giorni successivi i pulcini ben formati inizieranno a rompere il guscio e si sentirà il loro pigolio. Non vi aspettate però che appena nati siano paffuti e morbidi, per quello bisognerà attendere qualche ora, una volta asciutti li aspetterà sicuramente un bel book fotografico!

Fig. 2
Il portauova ruotato
con asta connessa
al servomotore.



LA STRUTTURA

L'intera incubatrice, modellata in 3D con SketchUp, è costituita da parti in plexiglass con spessore 3mm e 5mm tagliato al laser.

Il montaggio non richiede colla, ma solamente incastri e in alcuni casi viti per rendere la struttura più stabile.

Analizziamo ora nel dettaglio le componenti principali dell'incubatrice.

Nella **Fig. 2** possiamo vedere il portauova con la massima inclinazione che può raggiungere; l'abbiamo realizzato in modo tale che fosse estraibile con semplicità in quanto nella fase di schiusa non sarà più necessario. È costituito da due parti connesse che si muovono simultaneamente grazie ad un servomotore, ciascuna può contenere 4 uova. Sul fondo troviamo la vaschetta che conterrà l'acqua per ottenere l'umidità desiderata, la superficie della vaschetta è stata scelta per mantenerne il corretto valore, ma, come detto sopra, è comunque possibile regolarla aprendo o chiudendo un pertugio sul lato.

Sollevata di pochi centimetri dal fondo c'è la rete su cui nasceranno i pulcini.

Il coperchio invece presenta l'elemento riscaldante e, sulla parete laterale, troviamo una piccola ventola con lo scopo di creare un ambiente in cui la temperatura sia la più omogenea possibile.

L'elettronica è tutta confinata all'interno di una doppia parete sul lato corto dell'incubatrice. Nella **Fig. 3** possiamo vedere un'immagine di quest'area, lo scopo è isolare i componenti dall'ambiente caldo umido in cui nasceranno i pulcini e allo stesso tempo, con la seconda parete, non lasciare nulla allo scoperto evitando l'accesso dall'esterno.

L'HARDWARE

Iniziamo dal microcontrollore scelto, un ESP32-CAM Development Board della A.I. Thinker. Si tratta di un modulo che integra un Wi-Fi 802.11 b/g/n e un Bluetooth dual-mode (classico e BLE). Presenta poi una piccola telecamera da 2 Megapixel, un LED (attenzione perché se acceso per troppo tempo scalda notevolmente la scheda), uno slot per una memoria mini SD fino a 4 GByte e 9 pin I/O. Abbiamo già parlato in altri articoli di questo microcontrollore, ma ricordiamo velocemente come si programma: il modulo è privo di una porta USB per cui abbiamo due possibilità, o si utilizza un adattatore USB/Seriale (un FTDI), oppure si può utilizzare un Arduino UNO. Consigliamo il primo approccio e nella **Tabella 1** possiamo vedere i collegamenti necessari in questo caso.

Il PID è un regolatore che permette, tramite tre azioni di controllo, di contrastare l'errore dovuto a vari disturbi e avvicinarsi il più possibile al set point. Vediamo ad alto livello e abbastanza intuitivamente come funziona.

Il regolatore proporzionale (la P di PID) applica una correzione che sarà tanto più intensa quanto più grande è l'errore secondo un coefficiente moltiplicativo K_p . Da solo però questo controllo non garantisce una buona stabilità, abbiamo bisogno di altri due elementi.

L'integrale (la I di PID) è la somma, nel tempo, di tutti gli errori; è come se "ricordasse" quelli precedenti, quindi anche se l'errore in un determinato istante è nullo, il valore dell'azione integrale po-

trebbe non esserlo. Nella formula di questa correzione è presente la costante K_i da settare.

La derivativa (la D di PID) non lavora sull'errore in termini assoluti ma sulla sua tendenza alla crescita o alla diminuzione; se per esempio vediamo che l'errore sta crescendo molto rapidamente possiamo dare una correzione più intensa rispetto a quella che darebbe la sola azione proporzionale.

Qui è la costante K_d che va settata correttamente.

L'output sarà una combinazione di questi 3 elementi, la parte complessa è trovare i valori adeguati di K_p , K_i e K_d e in genere questo avviene empiricamente con test ed esperimenti (almeno a livello amatoriale).

Durante la programmazione il pin 100 deve essere connesso a GND, questo ponticello deve essere tolto a programmazione avvenuta.

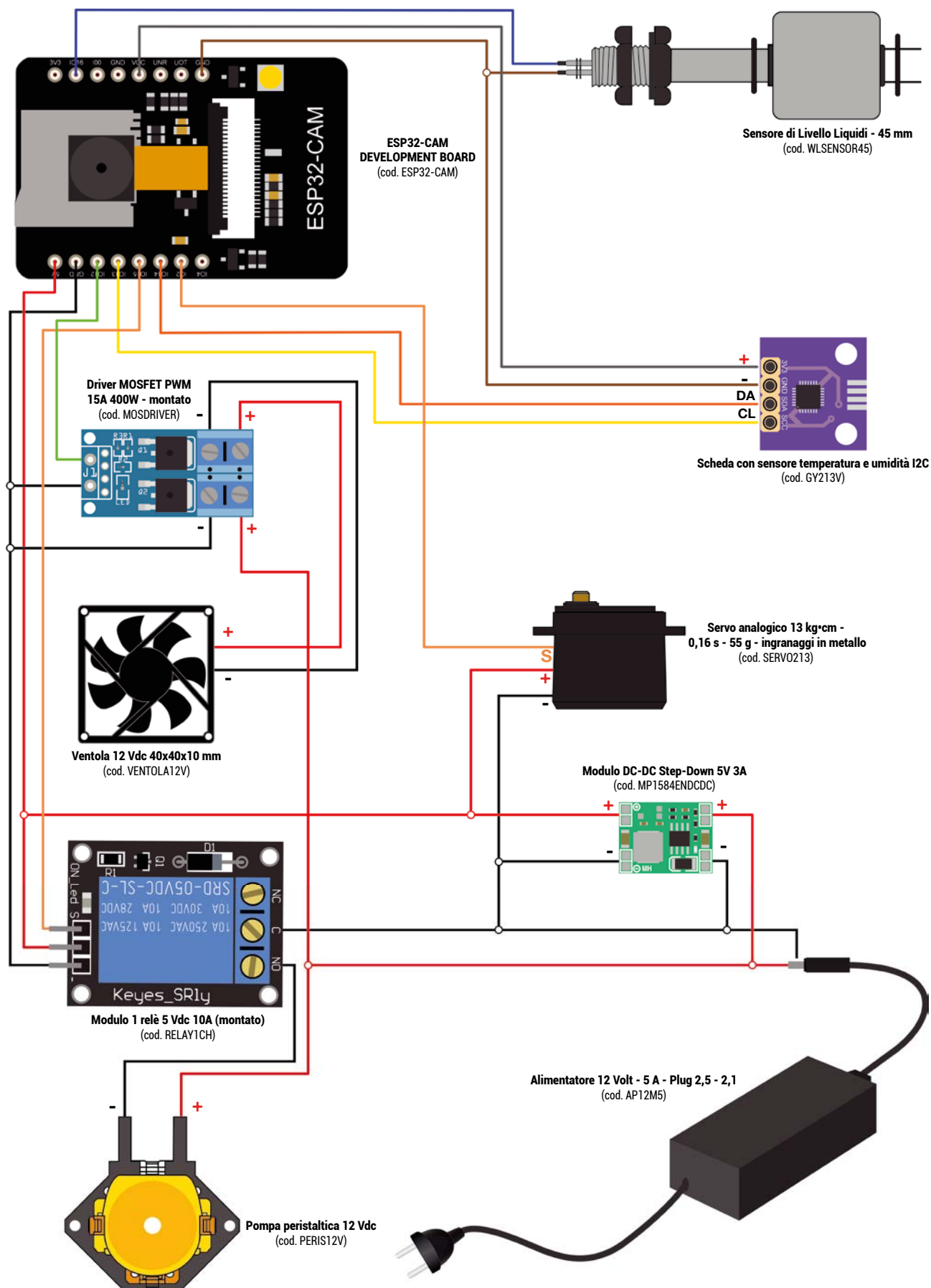
Per scrivere il codice si può utilizzare il classico IDE Arduino in cui è però necessario inserire il driver per l'ESP32-CAM. Il modulo viene poi alimentato con 5V.

Passiamo ora al sensore temperatura, abbiamo

Fig. 3
Doppia parete in cui abbiamo posizionato i componenti elettronici.



schema di CABLAGGIO



scelto un modulo basato sul HTU21D con interfaccia I²C. Ha una precisione di $\pm 0,3^{\circ}\text{C}$ per quanto riguarda la temperatura e di $\pm 2\%\text{RH}$ per quanto riguarda l'umidità, quindi è adatto ai nostri scopi. L'alimentazione è di 3,3V che prenderemo dal pin VCC dell'ESP32-CAM.

Per la gestione della vaschetta con l'acqua utilizziamo un galleggiante (fissato alla rete e inserito nella vaschetta) e una pompetta peristaltica attivata da un relé.

Il sensore di livello è dotato di due fili, il contatto è normalmente chiuso e, quando il livello dell'acqua sale, viene aperto. La pompa presenta un motore 12V DC e un tubo in silicone per trasferire l'acqua da una sorgente esterna alla vaschetta interna. La rotazione del portauova avviene grazie a un servo analogico con coppia di torsione 13 kg•cm e ingranaggi in metallo.

Il servo è alimentato con 5V.

L'elemento riscaldante è un piatto utilizzato nelle stampanti 3D, viene alimentato a 12V e controllato da un driver MOSFET (con in ingresso un segnale PWM) per carichi fino a 15A.

L'alimentazione principale di tutto il sistema è a 12V, questi vengono utilizzati direttamente dall'elemento riscaldante, la ventola (sempre accesa) e dalla pompetta peristaltica.

Gli altri componenti necessitano di 5V, abbiamo quindi inserito un modulo DC-DC stepdown.

Nella **Tabella 2** vediamo i pin utilizzati dal microcontrollore ESP32-CAM con i relativi collegamenti e, nella **Fig. 4**, possiamo vedere lo schema elettrico completo.

Tabella 1
Collegamenti ESP32-CAM con adattatore USB/Seriale.

PIN ADATTATORE USB/SERIALE	PIN ESP32-CAM
VCC	3V3
GND	GND
TX	U0R
RX	U0T

IL SOFTWARE

Possiamo suddividere il codice in due sezioni: la prima parte si occupa di gestire tutti i componenti dell'incubatrice mantenendo la temperatura adeguata, il livello dell'acqua e controllando la rotazione del servo; la seconda parte si occupa dell'interfaccia con l'utente, ovvero l'invio di informazioni al bot Telegram (dati di temperatura, umidità e immagini) tramite Wi-Fi.

Analizziamo ora la prima sezione, in particolare la gestione della temperatura. Abbiamo utilizzato una regolazione PID (ovvero proporzionale, integrativo, derivativo) rispetto all'errore che si presenta al comparatore dovuto alla differenza tra il valore di set point (valore di temperatura impostato) ed il valore della grandezza rilevata. Abbiamo incluso la libreria "PID_v1.h" che lavora su 5 parametri: i coefficienti Kp, Ki e Kd, un input e un output.

I coefficienti li abbiamo scelti dopo qualche test per ottenere una regolazione ottimale, l'input è il set

Tabella 2
Pin di ESP32-CAM usati.

PIN ESP32-CAM	COMPONENTE CONNESSO	PIN COMPONENTE
5V	Modulo DC-DC stepdown	5V
GND		GND
IO2	Servo	PWM
IO12	Mosfet	PWM
IO13	Sensore temperatura umidità	SCL
IO14		SDA
VCC		3.3V
IO15	Relé	SEGNALE
IO16	Galleggiante	

↓ Listato 1

```
if(timer_temp()){
    humd = umid_temp.readHumidity();           //lettura temperatura da sensore
    temp = umid_temp.readTemperature();        //lettura umidità da sensore
    input = temp;
    setpoint = 37.7;
    myPID.Compute();

    int pwm = constrain((int)output, 0, 200);
    ledcWrite(5, pwm);                          //output sul mosfet (channel 5) per controllo piatto

    t_temp = millis();
}
```

↓ Listato 2

```
int timer_servo()
{
    int ret_servo = 0;

    diff_servo = millis() - t_servo;
    if(diff_servo > TEMPO_ROTAZ)
        ret_servo = 1;

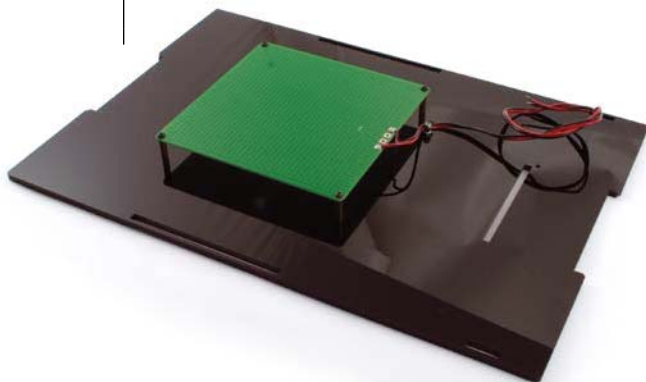
    return ret_servo;
}
```

↓ Listato 3

```
const char* ssid = " Nome Wi-Fi ";
const char* password = " Password Wi-Fi ";
String chatId = " User id ";
String BOTtoken = " Token del bot ";
```

point (fissato inizialmente a 37,7°C), mentre l'output è il valore di PWM generato che noi limitiamo a 200 per evitare un eccessivo assorbimento di corrente da parte del piatto riscaldante. Nel **Listato 1** possiamo vedere la porzione di codice che gestisce la regolazione di temperatura.

↓
Il riscaldatore.



Come si può vedere all'interno del primo "if" viene chiamata la funzione "timer_temp()"; si tratta di un timer non bloccante che sfrutta la funzione "millis()" di Arduino. Abbiamo utilizzato timer non bloccanti in tutto il codice per gestire pompa, sensore e servomotore in modo da tenere traccia del tempo e intanto svolgere i vari compiti.

Nel **Listato 2** ne vediamo un esempio, in questo caso quello relativo al servomotore che deve effettuare una rotazione ogni 3 ore. La costante "TEMPO_ROTAZ" viene definita all'inizio e vale "3*60*60*1000", appunto 3 ore scritto in millisecondi. Vediamo ora il controllo della pompa. Siccome non vogliamo che si accenda per oscillazioni della superficie dell'acqua che fanno muovere il galleggiante e magari aprire il circuito, abbiamo fissato delle soglie in termini di tempo; in particolare il galleggiante deve restare nello stato "vaschetta vuota" per più di 10 secondi per far attivare la pompa e, iniziato il riempimento, deve restare nello stato "vaschetta piena" 5 secondi per spegnere la pompa. Anche in questo caso abbiamo utilizzato timer non bloccanti.

Come spiegato in precedenza, il sistema sfrutta il collegamento Wi-Fi per interagire con l'utente tramite un bot Telegram. Abbiamo già parlato in altri articoli di come si crea un bot, ne ricordiamo ora brevemente i passaggi.

Dall'app di Telegram si cerca "BotFather", chat tramite cui si può creare il bot; dopo la creazione avrete a disposizione il token. Siccome chiunque conosca il nome del vostro bot può interagire con esso, facciamo in modo che gli unici possibili utilizzatori siate voi grazie al vostro User ID. Cercate quindi "IDBot" e scrivendo "/getid" avrete il vostro identificativo. Ora che abbiamo creato il Bot, agendo sul codice, dobbiamo fare in modo che l'ESP32-CAM si colleghi ad esso.

Nel **Listato 3** vediamo i campi da compilare con le

▼ Listato 4

```
void handleNewMessages(int numNewMessages){
  Serial.print("Gestisco nuovo messaggio: ");
  Serial.println(numNewMessages);

  for (int i = 0; i < numNewMessages; i++){
    // Chat id del mittente
    String chat_id = String(bot.messages[i].chat_id);
    if (chat_id != chatId){
      bot.sendMessage(chat_id, "Utente non autorizzato", "");
      continue;
    }

    // Stampa il messaggio ricevuto
    String text = bot.messages[i].text;
    Serial.println(text);

    String fromName = bot.messages[i].from_name;

    if (text == "/flash") {
      flashState = !flashState;
      digitalWrite(FLASH_LED_PIN, flashState);
    }
    if (text == "/photo") {
      sendPhoto = true;
      Serial.println("Richiesta di nuova fotografia");
    }
    if (text == "/readings"){
      String readings = "Temperatura: " + String(temp) + "°C " + "\nUmidità: " + String(humd) + "%";
      bot.sendMessage(chatId, readings, "");
    }
    if (text == "/start"){
      String welcome = "Ecco una lista di comandi per controllare la tua incubatrice:\n";
      welcome += "/photo : scatta una fotografia\n";
      welcome += "/flash : accendi o spegni il flash LED\n";
      welcome += "/readings : richiedi temperatura e umidità\n\n";
      bot.sendMessage(chatId, welcome, "Markdown");
    }
  }
}
```

informazioni sul Wi-Fi e sul bot; utilizziamo inoltre la libreria "UniversalTelegramBot". Quando l'utente scriverà in chat "/start" gli verranno presentati un elenco di comandi tra cui la possibilità di accendere il LED, scattare una fotografia e leggere i valori di temperatura e umidità.

Nel **Listato 4** vediamo la funzione che gestisce l'arrivo di un nuovo messaggio; inizialmente ci si chiede se il mittente è effettivamente l'utente di cui abbiamo inserito l'identificativo, in caso affermativo si legge il messaggio e si agisce di conseguenza.

Nel caso si tratti della richiesta di una fotografia, la variabile "sendPhoto" diventa "true" e questo fa sì che entri in funzione "sendPhotoTelegram()" in cui è presente tutta la procedura per scattare una fotografia e inviarla in chat al bot.

Abbiamo scelto di controllare l'arrivo di nuovi messaggi ogni 5 secondi per evitare di stressare troppo il microcontrollore che potrebbe surriscaldarsi.

CONCLUSIONI

Con questo progetto abbiamo quindi cercato di riprodurre artificialmente il lavoro che una chioccia fa da migliaia di anni; mamma corrente elettrica e papà PID faranno un buon lavoro? Non vi resta che provare e scoprire se veramente la tecnologia può (entro certi limiti) sostituire la natura. □



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. Tra i componenti principali, la board ESP32-CAM (cod. ESP32-CAM) è in vendita a Euro 16,50, il sensore di livello liquidi (cod. WLSSENSOR45) è disponibile a Euro 6,00, il sensore temperatura e umidità (cod. GY213V) è disponibile a Euro 7,50. 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



MAKER UNO

La scheda **Cytron** Technologies compatibile
con il mondo Arduino®

Starter kit con Maker Uno
e tutto quello che serve
per iniziare
a programmare
con il linguaggio
Arduino!

+

50
video
tutorial



€ 58,00

cod. MAKERKITBASE

**FUTURA
ELETTRONICA®**

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

Prezzo IVA inclusa

Caratteristiche tecniche
e vendita on-line su:

www.futurashop.it

COMANDIAMO LA SERRANDA DA SMARTPHONE

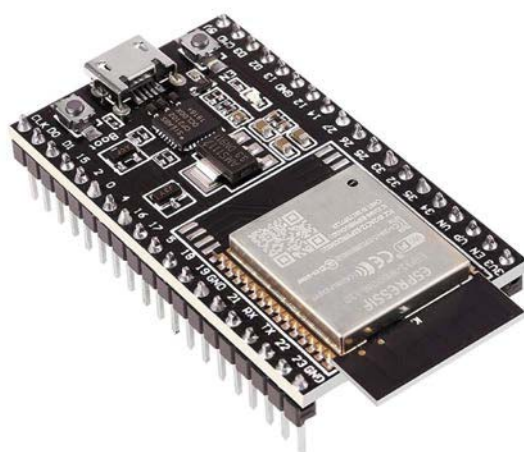
degli ingg. **MASSIMO BERTUCCI**
e **FRANCESCO NOTO**



Un progetto smart per poter aprire e chiudere comodamente dallo smartphone la serranda di casa, con possibilità di ricevere e-mail sullo stato.

Le nostre case sono sempre più automatizzate e la domotica si conferma sempre più la tecnologia degli anni a venire; del resto molti di noi desiderano automatizzare sempre più cose, dalle luci interne ed esterne, ai cancelli e alle tapparelle, per arrivare talvolta ad ambiti molto specifici. Il progetto qui proposto, da noi battezzato *Serrandino*, va in tale direzione e consente il controllo remoto, con tanto di feedback, della serranda o basculante di un'autorimessa, mediante uno smartphone e le connessioni WiFi (per l'accesso a Internet) e Bluetooth LE. Per l'esattezza, il progetto proposto consiste in un dispositivo smart apriserranda, in grado di collegarsi via Bluetooth ad uno smartphone e, tramite un'apposita app, inviare comandi di apertura e chiusura. Tale livello di apertura e chiusura espresso in percentuale, viene poi inviato tramite e-mail dal dispositivo stesso.

Fig. 1
La Board ESP32.



Il tutto viene realizzato con l'aiuto di una board ESP32, dotata di connettività wireless sia Bluetooth che WiFi; tale scheda (**Fig. 1**) commercializzata dalla Futura Elettronica (www.futuranet.it) è basata su un SoC MCU a 32 bit ed è nata per aggiungere connettività wireless ai sistemi elettronici e per la progettazione di sistemi IoT low cost.

Questa board ha una potente capacità di elaborazione, grazie alla quale è possibile collegare, attraverso i suoi pin GPIO, diverse periferiche tra le quali dei sensori ed altri dispositivi compatibili. L'hardware che compone la scheda è basato su un chip ESP32 (**Fig. 2**) dual core (240 MHz e 520Kb di memoria SRAM) dotato di 48 pin dotati di diverse funzionalità. Le periferiche del modulo includono canali ADC e DAC, interfacce seriali (UART, SPI), canali PWM e interfacce I2C.

Tale chip supporta un range di frequenza compreso tra 2,4GHz e 2,5GHz, standard 802.11 b/g/n, tre diverse modalità di utilizzo ed una velocità massima dati 150 Mbps@11n HT40, 72 Mbps@11n

Fig. 2
ESP32 dual core chip.



HT20, 54 Mbps@11g e 11 Mbps@11b.

La breakout board dispone di un connettore micro USB, un pulsante di Boot e pulsante di reset.

Può essere impiegata in una vasta gamma di applicazioni come: dispositivi indossabili, reti di sensori a bassa potenza, cuffie audio, prese intelligenti, domotica, reti mesh, controllo wireless industriale. Le applicazioni possibili sono smart power plug, home and building automation, reti di tipo Mesh, reti di sensori e sistemi di posizionamento WiFi beacon.

Le tre modalità di utilizzo del modulo ESP32 sono:

- 1. Station:** il dispositivo funziona come un normale dispositivo di rete;
- 2. Access Point:** il dispositivo funziona come un punto di accesso;
- 3. Access Point + Station:** modalità ibrida (AP + Station) (**Fig. 3**).

IL NOSTRO SISTEMA

Per poter azionare i motori della nostra serranda occorrono due relé: uno per la salita ed uno per la discesa, per poter aprire e chiudere la serranda stessa; questi relé, insieme a un minimo di elettronica di contorno, prendono posto su un circuito stampato che abbiamo progettato appositamente e che è descritto dallo schema elettrico che trovate in queste pagine.

Tale circuito viene interfacciato alla scheda con ESP32 la quale, tramite le linee impostate come uscite, pilota i relé e relativi LED, mentre con gli I/O si interfaccia ad una presa USB.

Ogni relé viene pilotato attraverso un transistor NPN configurato a emettitore comune, che funge da switch.

Per rendere più compatto l'insieme, abbiamo previsto di montare il modulo ESP32 sullo stesso PCB dei relé, sul quale prenderà posto anche il connettore microUSB.

Sul modulo ESP32 o in generale su una qualsiasi scheda Arduino è presente un pulsante che può essere utilizzato per far eseguire una porzione di codice quando il pulsante viene premuto, ovvero in base ad un evento; nella nostra ESP32, di pulsanti ne abbiamo due e più avanti vi spiegheremo che funzione svolgono nel progetto.

Siccome quando i contatti di un pulsante si usurano è possibile che ad ogni pressione e rilascio si generino più chiusure e aperture (note come "rimbalzo") si ricorre a tecniche di Debouncing o antirimbalo, ossia soluzioni per mitigare questo effetto ed evitare che il circuito rilevi più commutazioni per ogni pressione.

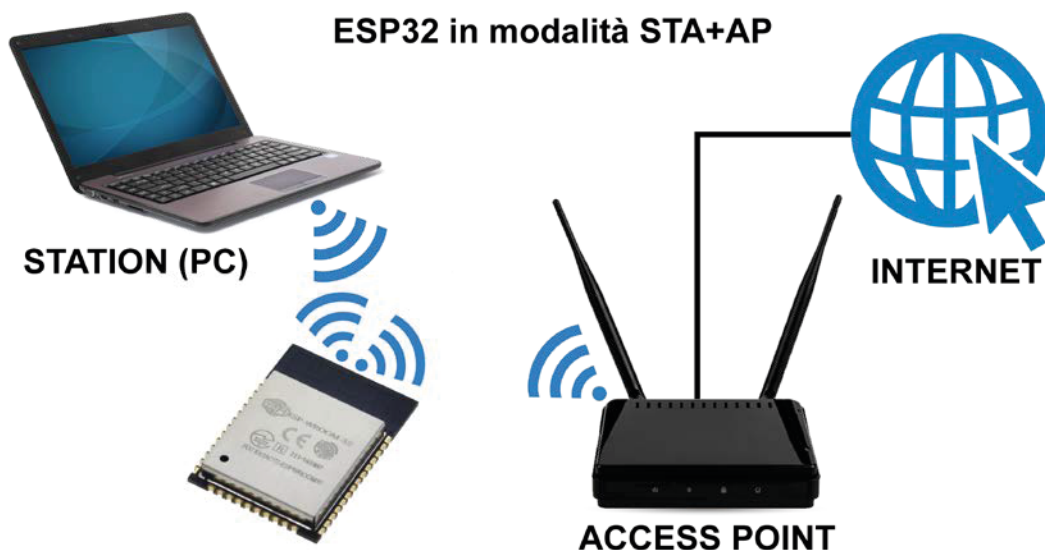


Fig. 3
Modulo WiFi
in modalità
doppia
station
access point.

Esistono due modi per implementare l'antirimbalo di un pulsante:

- **Hardware**, filtrando il segnale in ingresso;
- **Software**, aggiungendo un tempo morto dopo il primo rilevamento.

L'antirimbalo hardware è una soluzione preferibile rispetto a quello software. Fondamentalmente, questo può essere implementato aggiungendo un filtro passa basso (circuitto con resistenza e capacità) tra il pulsante e il GPIO corrispondente.

SCHEMA ELETTRICO

Diamo dunque uno sguardo allo schema del circuito che ospita l'ESP32, siglato U1, i relé, i transistor che li pilotano e i relativi LED di segnalazione. Del modulo abbiamo utilizzato la linea GPIO D13 (pin 16) come uscita per il controllo del relé RL1, tramite il transistor T1 pilotato in base attraverso la resistenza R3 e la D12 (pin 14) per pilotare la base di T2, attraverso R4, in modo da controllare l'attività del relé RL2. Lo stato di ciascun relé viene segnalato da un LED, che è LD1 per T1 e LD2 per T2 e la bobina di ciascun relé ha un diodo in antiparallelo per proteggere la giunzione di collettore del rispettivo BJT. I GPIO D4 (pin 26) e D5 (pin 29) sono stati impostati, invece, come ingressi per leggere le condizioni di discesa (DOWN) e salita (UP) della serranda, desumendole dallo stato dei pulsanti che localmente consentono il comando manuale. Il relativo collegamento sarà descritto più avanti in questo stesso articolo; a riguardo non preoccupatevi, perché non andremo a collegare gli input (decisamente delicati) della ESP32 alla linea

elettrica dei pulsanti, ma sconetteremo gli stessi per inviarne lo stato al modulo U1 e fare in modo che questo li ripeta attraverso i relé di cui dispone. Il circuito del nostro controllo a distanza per serranda trae alimentazione dalla linea a 5 volt, collegata a un connettore plug, che nello schema elettrico è siglato PWR; la presenza dei 5V è segnalata dal LED LD3 sull'alimentazione Vcc.

FUNZIONAMENTO DEL SISTEMA

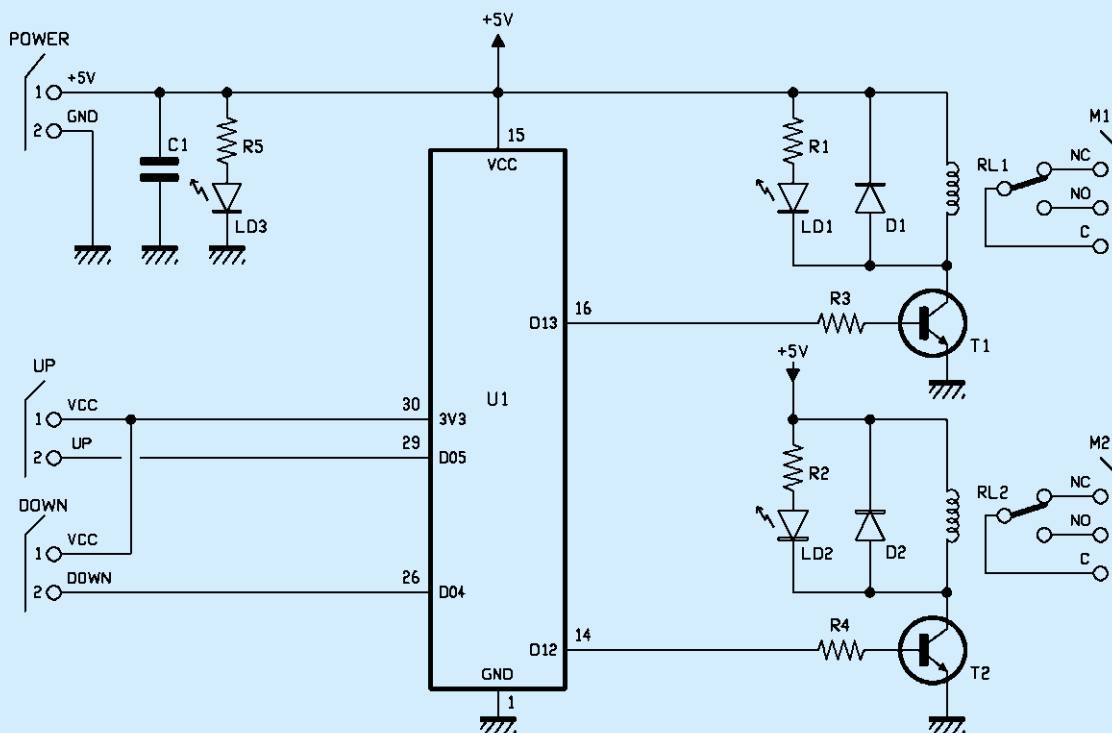
Il progetto Serrandino prevede l'invio dei comandi di apertura e chiusura tramite il bluetooth integrato sul modulo ESP32, attraverso l'APP "Bluetooth Terminal" scaricabile dai principali APP store. Prima di usare l'APP, però, bisogna configurare il dispositivo.

CONFIGURAZIONE

Per la configurazione, bisogna accendere la scheda e cercare tra le reti WiFi disponibili la rete "Serrandino" (Fig. 4). Collegatevi dunque in wireless al dispositivo (SSID: Serrandino) ed immettere la password predefinita "123456789". Utilizzando un Browser (consigliato Google Chrome), digitate nella barra degli indirizzi l'IP address **192.168.4.1**. Fatto ciò, dovete procedere con la scelta dei parametri da impostare al dispositivo utilizzando l'apposito menù a tendina, inserire le informazioni corrette. Se la procedura va a buon fine comparirà il messaggio di corretta configurazione.

Di seguito spieghiamo i parametri da inserire:

- **WiFi SSID**: nome della rete su cui si vuole collegare il dispositivo;
- **WiFi Password**: password della rete su cui si vuole collegare il dispositivo;



- **Tempo Salita:** espresso in secondi, è il tempo che impiega la serranda ad aprirsi tutta (percen-

tuale di apertura 100%);

- **Tempo Discesa:** espresso in secondi, è il tempo che impiega la serranda a chiudersi tutta (percentuale di apertura 0%);
- **Nome Bluetooth:** nome del dispositivo Bluetooth;
- **Password Bluetooth:** è la password che viene chiesta quando si fa il pairing col Bluetooth;
- **E-mail di notifica:** indirizzo di posta elettronica di riferimento per ricevere la notifica di apertura/chiusura e relativa percentuale.



Fig. 4
Serrandino è presente tra le reti WiFi.

Nel momento in cui uno di questi valori non viene settato, l'utente verrà avvisato su quale dei parametri è stato omesso, per una corretta configurazione bisogna inserire tutti i parametri. Nel caso in cui ne manca uno verrà visualizzato il messaggio di segnalazione visibile in **Fig. 5**.

Nel caso in cui la configurazione vada a buon fine il dispositivo avviserà a video il corretto salvataggio dei dati. Una volta configurato il dispositivo, si potrà interagire con l'APP precedentemente descritta; per fare questo bisogna effettuare la ricerca (Discovery) dei dispositivi Bluetooth Low Energy nelle vicinanze e collegarsi, effettuando così il pairing, al dispositivo con il nome scelto in fase di configurazione: come mostra la **Fig. 6**, in questo caso è "SerrandinoBLE".

Se il "pairing" del dispositivo è andato a buon fine,

per iniziare la comunicazione bisogna scrivere un qualsiasi carattere sul terminale. Nel momento in cui ci si collega sul dispositivo bisogna inserire la password corretta, settata in fase di configurazione. Si hanno tre tentativi a disposizione, altrimenti il sistema si blocca.

Una volta inserita la password corretta premere invio per avviare la comunicazione da terminale (**Fig. 7**). Come si può notare dallo screenshot in **Fig. 7**, è stata inserita la password corretta all'ultimo tentativo ed il sistema ha dato il benvenuto con il valore di percentuale attuale di apertura, precedentemente salvato (% 50). Scrivendo 100 sul nuovo valore da terminale, verrà aperta la serranda fino al 100% (apertura completa). Alla fine si ottiene il messaggio "APERTO". Al cambio di percentuale in base all'evento di chiusura o apertura verrà inviata una mail di notifica, come descritto nel paragrafo successivo.

INVIO COMANDI MANUALI E RESET DEL DISPOSITIVO

Per poter aprire e chiudere la serranda, ci sono due modi possibili: usare il terminale dallo smartphone oppure usare i tasti su e giù e quindi comandare il dispositivo manualmente.

Nel momento in cui si piglia il tasto su, la serranda inizia ad alzarsi e continua finché non si preme un'altra volta il tasto su (o giù è indifferente) per fermarla. Lo stesso vale per la chiusura.

Se si volesse resettare il dispositivo, perché lo si vuole agganciare ad un'altra serranda, oppure è cambiato il gestore di internet, quindi si deve rifare la configurazione, bisogna aprire il terminale dall'App e scrivere il comando "RESET" (oppure "reset", upper case o low case è indifferente per questa operazione). Una volta resettato, il dispositivo è pronto per una nuova riconfigurazione.

CONFIGURAZIONE PER RICEVERE LA MAIL DI NOTIFICA

È possibile inviare e-mail di notifica dal dispositivo, ovvero messaggi di posta elettronica sugli eventi di apertura e chiusura della serranda; a riguardo vi consigliamo di creare un nuovo account e usarlo solo a questo scopo per motivi di sicurezza, ovvero di non usare l'indirizzo e-mail personale!

Nel nostro esempio è stato utilizzato un account Gmail creato ad hoc per inviare le e-mail di notifica, ma si può utilizzare qualsiasi altro provider di posta elettronica; l'account e-mail del destinatario dei messaggi può invece essere tranquillamente uno personale.

Fig. 5
Configurazione parametri e Warning campo obbligatorio.

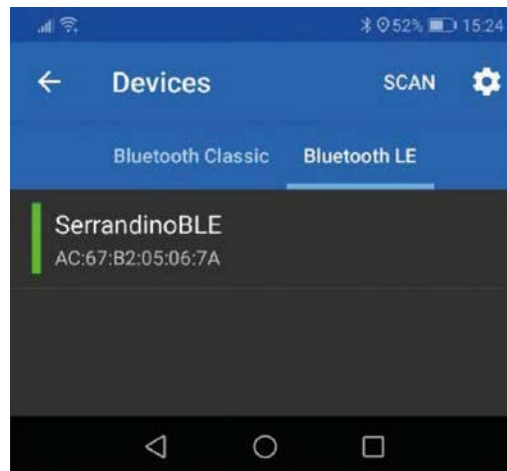


Fig. 6
Discovery dispositivi bluetooth dall'App.

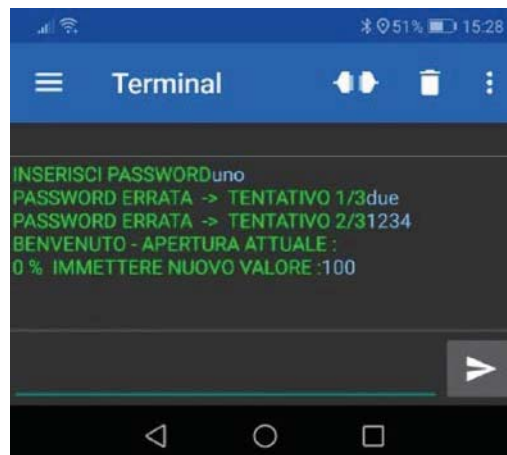


Fig. 7
Esempio apertura serranda.

Elenco Componenti:

R1, R2, R5: 470 ohm
 R3, R4: 3,3 kohm
 C1: 100 nF mutistrato
 LD1, LD2: LED 3 mm rosso
 LD3: LED 3 mm verde
 T1, T2: BC547
 D1, D2: 1N4148
 RL1, RL2: Relé 5V
 U1: ESP32

Varie:

- Morsetto 2 poli passo 5 mm (2 pz.)
- Morsetto 3 poli passo 5 mm (2 pz.)
- Strip femmina 15 poli passo 2,54 mm (2 pz.)
- Plug DC
- Circuito stampato S1595 (100x50 mm)

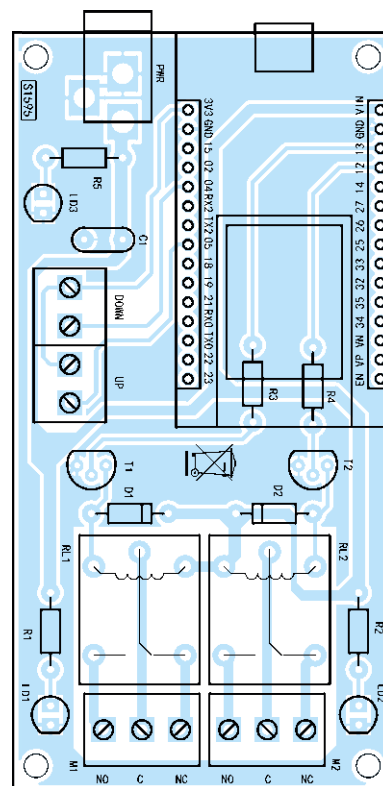
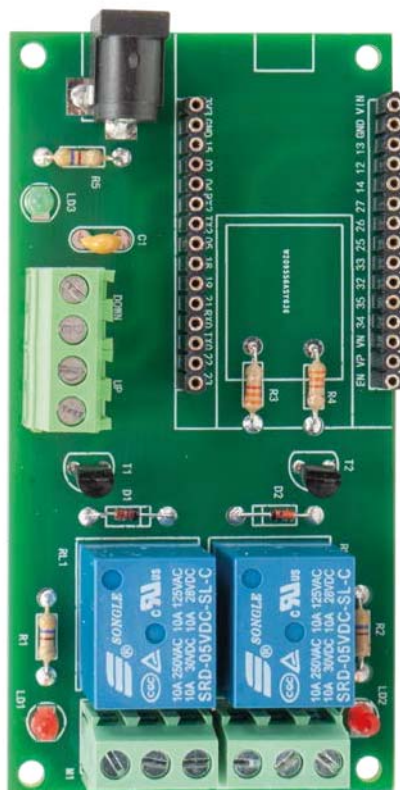


Fig. 8
 Permessi
 App meno sicure
 in Gmail.

Per creare un nuovo account di posta elettronica su Gmail, si può accedere al seguente link web e www.google.com/gmail/about/ e una volta creato l'account bisogna consentire, alle App meno sicure, di accedere ad esso, in modo da poter inviare e-mail. Per modificare questa informazione visitate questo link web: <https://myaccount.google.com/intro/security>, vi troverete in una pagina web come quella mostrata nella **Fig. 8**. Nella **Fig. 9** trovate un esempio di ricezione e-mail per apertura (in rosso) e chiusura (in blu).

REALIZZAZIONE E INSTALLAZIONE

Andiamo quindi alla parte pratica del progetto, iniziando con la costruzione del circuito di controllo e concludendo con il cablaggio nell'impianto elettrico. Per il circuito stampato, scaricate lo ZIP contenente tutti i layer in formato Gerber (si tratta di un doppia faccia quindi troverete i file relativi a entrambi i lati del PCB) dal nostro sito www.elettronica.in.it (le trovate nella sezione Download relativa ai file di questo numero della rivista) e dopo averle stampate su carta da lucido o acetato, potete procedere all'esposizione UV e all'incisione di una basetta ramata presensibilizzata a doppia ramatura, quindi allo sviluppo e all'incisione.

Cosa occorre?

I componenti principali utilizzati in questo progetto sono disponibili presso Futura Elettronica. Il modulo con ESP32 (cod. YB555) è in vendita a Euro 13,50. Il PCB può essere realizzato utilizzando il servizio PCBPRODUCTION disponibile sul sito www.futuranet.it. 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

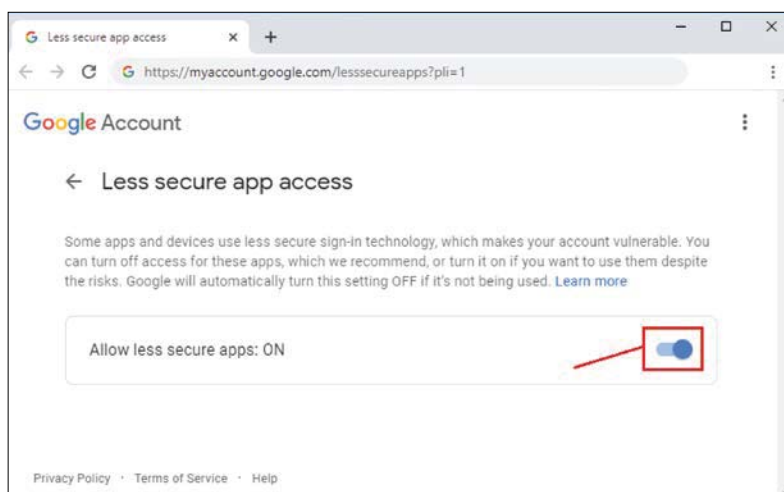




Fig. 9
Accesso
Serrandino.

Forate quindi il PCB e iniziate a montare i componenti del caso in ordine di altezza, concludendo con i due relé; stagnate da entrambe le facce le piazzole di interconnessione e dove serve realizzate delle "vie" tra le due facce mediante avanzi di reofori dei componenti o sottili spezzoni di filo in rame, che stagnerete da entrambe le parti (questo non serve se vi fate realizzare il PCB dal servizio dedicato dalla Futura Elettronica, *PCBPRODUCTION*, giacché vi verrà fornito uno stampato con fori metallizzati). Per l'orientamento dei componenti polarizzati, ossia transistor e diodi, ma anche la scheda ESP32, seguite l'orientamento indicato nel piano di montaggio pubblicato in queste pagine; a proposito della ESP32, montatela utilizzando due strip di pin a passo 2,54 mm da 15 contatti ciascuna. Il montaggio del dispositivo è semplice, ma bisogna fare attenzione in quanto è necessario interfacciarsi con la rete elettrica della propria abitazione, che ad alta tensione: 230 Vca. Lo schema che vi guiderà nel cablaggio è quello proposto nella **Fig. 10**.

Operate quindi seguendo questa procedura:

1. posizionate la scheda in prossimità della

tapparella, o comunque in una zona dove sono accessibili i cablaggi elettrici pre-esistenti;

2. collegate il tasto Sali-Scendi presente all'interno della cassetta 503 come indicato in **Fig. 10**, cioè ad ogni estremità dei pulsanti vanno collegati i fili che portano il segnale fino ai connettori SALI e SCENDI sul PCB;
3. connettete i fili del motore come segue:
 - filo blu a NEUTRO;
 - filo Verde-Giallo a terra
 - fili marrone e nero alla scheda sui contatti NO dei due relé (prima del montaggio controllare quale dei colori è associato ad ognuna delle due direzioni);
4. collegate la fase ai connettori COM.

Fatti tutti i collegamenti e verificato che siano esatti e che non abbiate erroneamente creato cortocircuiti, potete alimentare la board inserendo il plug di un alimentatore a 5V nell'apposita presa presente nel PCB.

A questo punto potete procedere alla configurazione, come descritto in precedenza.

Buon lavoro!

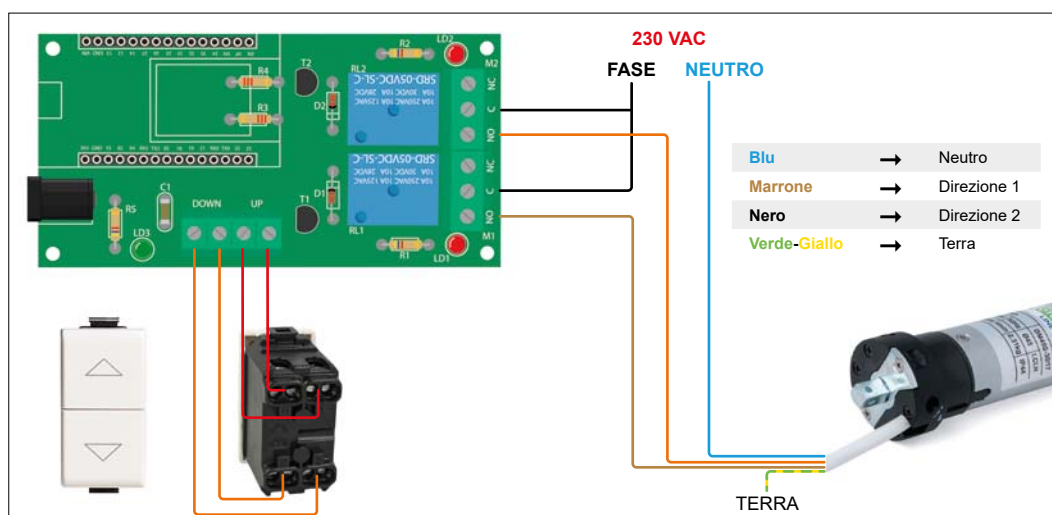


Fig. 10
Schema
di collegamento
del sistema.

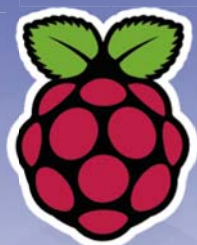
RASPBERRY PI 400!

La tastiera con il computer integrato

€ 146,00

Cod. PI400KIT-ITA

IVA inclusa



**Manuale
in italiano!**

Prezzo IVA inclusa.

Kit completo con computer Raspberry Pi 400 accessori e manuale in Italiano.



Raspberry Pi 400 è un personal computer completo che integra nella tastiera (italiana), una scheda Raspberry Pi 4 da 4Gb con processore quad-core a 64 bit, Wi-Fi, Bluetooth, doppia uscita video (micro HDMI) con una risoluzione fino a 4K, Gigabit Ethernet, USB 2.0 e USB 3.0.

Leggero, portatile, semplice da usare. È ideale per navigare sul web, creare e modificare documenti, guardare video e imparare a programmare utilizzando l'ambiente desktop del sistema operativo Raspberry Pi. I pin GPIO di Raspberry Pi rimangono comunque accessibili, permettendo di connettere componenti e prototipare progetti.

La confezione di Raspberry Pi 400 contiene tutto il necessario (ad eccezione di una TV o di un monitor): computer Raspberry Pi P400, alimentatore da rete, mouse, scheda microSD da 16GB pre-programmata con Raspberry Pi OS, cavo da Micro HDMI a HDMI A lungo 1 metro, un manuale illustrato a colori di 247 pagine completamente in Italiano.

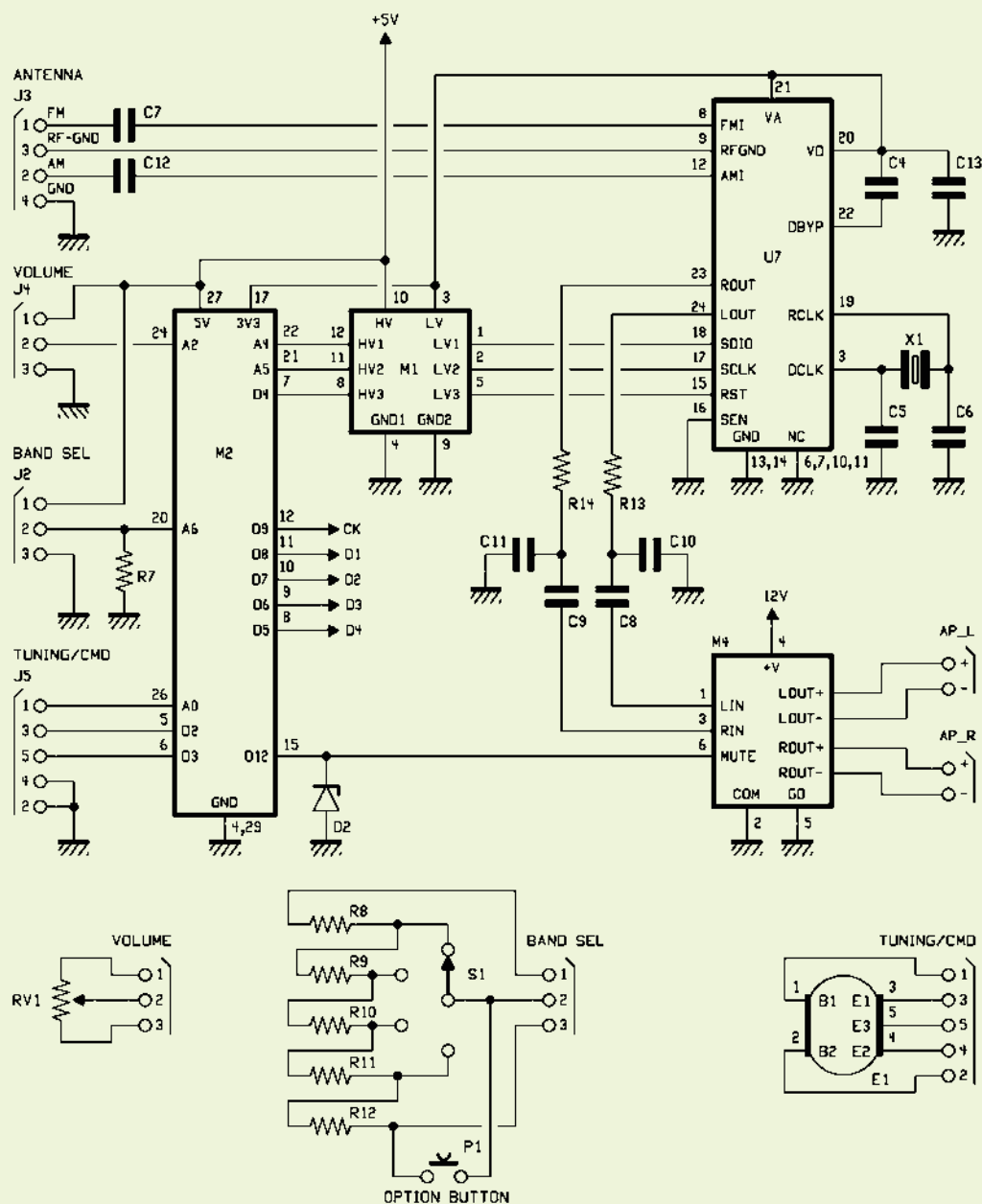
AUTORADIO HI-TECH DALL'ASPETTO VINTAGE

di STEFANO PURCHIARONI



Riproponiamo in chiave moderna le affascinanti autoradio primordiali che equipaggiavano le auto di una volta. Seconda e ultima puntata.

Esistono componenti che hanno fatto la storia dell'elettronica e sicuramente quelli primordiali, storici, sono per eccellenza le valvole termoioniche -precorritrici dei moderni transistor- e le Nixie, utilizzate come visualizzatori e vere e proprie antenate dei moderni display. Per rievocare il fascino "old style" dei tubi Nixie (talvolta impropriamente chiamati "valvole" Nixie) abbiamo deciso di dedicare loro questo articolo, facendo un mix di antichità e modernità per creare una moderna radio ricevente per auto che si ispira alle prime autoradio della storia, dove utilizziamo un'elettronica hi-tech gestita da microcontrollore per la ricezione e la demodulazione, nonché per l'amplificazione audio, affiancandole delle Nixie restyling prodotte ancora con una tecnologia che richiama quella delle originali.

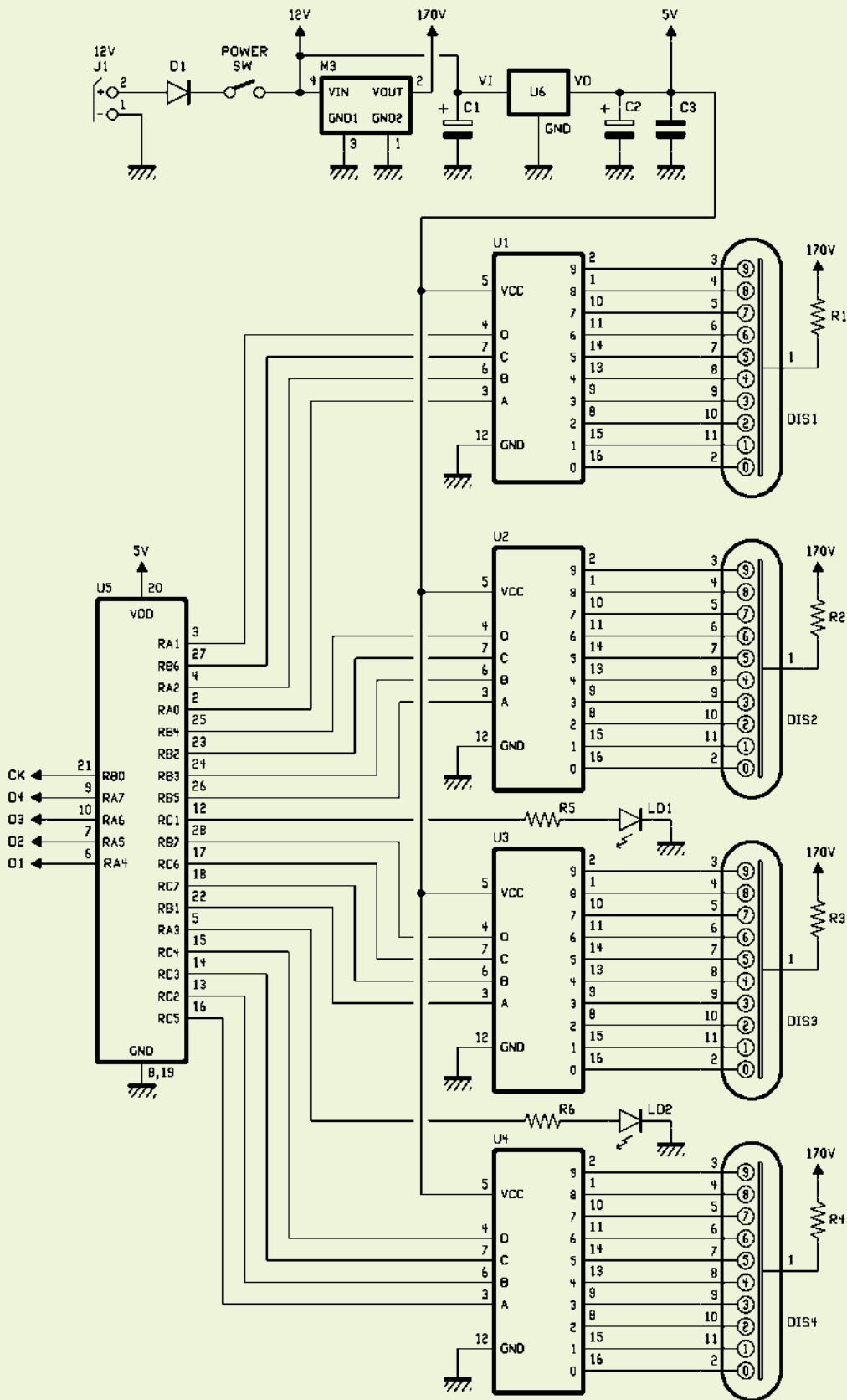


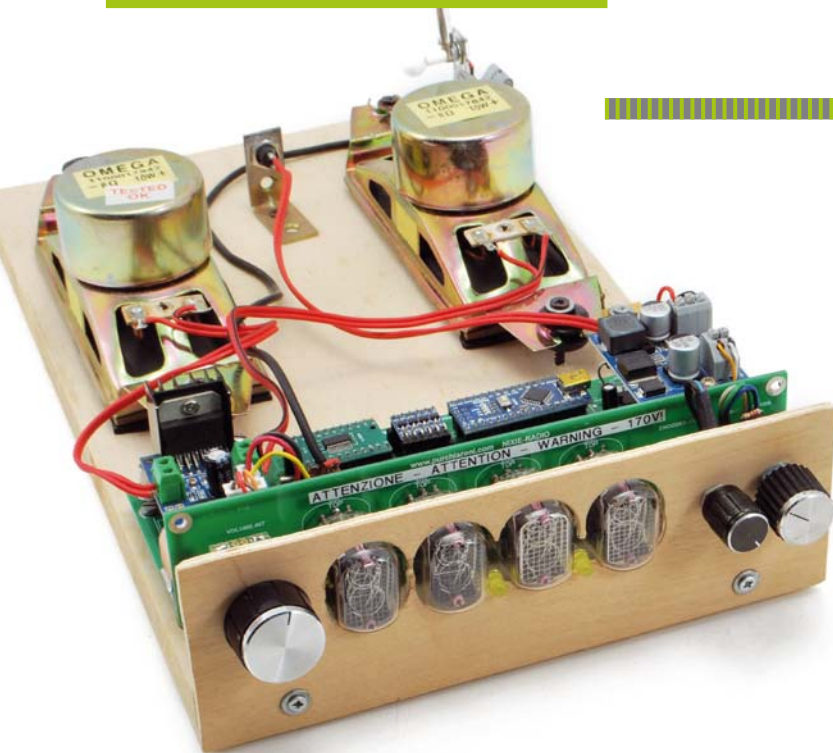
Nella prima puntata di questo articolo vi abbiamo accennato l'hardware della nostra autoradio, abbozzando la gestione a microcontrollore, che è distinta per il blocco radioricevente e per quello display: vi sono quindi due unità, una facente capo alla radio, che trasmette a quella che governa le Nixie i dati da visualizzare. A riguardo, abbiamo descritto la gestione firmware, le impostazioni e le funzioni, nonché la parte display, governata da una board Arduino Nano. Adesso è giunto il momento di analizzare, attraverso lo schema elettrico, come sono composte le

parti dell'elettronica e lo facciamo descrivendo lo schema elettrico del circuito, che le descrive meglio d'ogni altra cosa.

SCHEMA ELETTRICO

Andiamo ora all'analisi dei circuiti componenti il progetto, tutti ospitati su un PCB del quale riportiamo lo schema elettrico in queste pagine; iniziamone l'analisi dal Sottosistema Display. Lo schema mostra i collegamenti tra il microcontrollore PIC 18F2626 ed i Tubi Nixie, mediato dai driver ad alta tensione 74141 U1+U4.





Il nostro prototipo al completo, con gli altoparlanti sul fondo del mobile.

La tensione anodica di 170V è fornita dal modulo Step-Up M3, alimentato direttamente dalla linea di alimentazione esterna proveniente dalla batteria dell'auto. Si noti la presenza del diodo D1 da 3A a protezione in caso di inversioni accidentali di polarità. I 12V, filtrati dai disturbi presenti nell'impianto elettrico dell'auto da un elettrolitico da 2200 uF, sono usati anche per alimentare l'amplificatore Audio nello schema mostrato più avanti. L'integrato U6 (7805) fornisce una tensione stabilizzata di 5V ed ulteriormente filtrata da

disturbi RF tramite C3. Tale tensione è necessaria al funzionamento dei microcontrollori e dei driver Nixie. Linee di output dedicate pilotano i LED di separazione decimale tra le cifre del display. Il PIC riceve dati ed istruzioni tramite le linee D1+D4, letti al fronte di salita della linea CK. Tali linee provengono da Arduino Nano, che gestisce tutte le periferiche, incluso il sottosistema Display visto prima, tramite le linee D1..D4 e CK. Oltre al display abbiamo un encoder munito di pulsante (E1) per la sintonia e per la gestione delle varie funzionalità dell'apparecchio radio, un potenziometro (RV1) per la gestione del volume audio, un commutatore a quattro posizioni (S1) per la selezione delle bande di frequenza, ed infine un pulsante Opzioni per la gestione delle bande laterali e per entrare nel modo Impostazioni. Il commutatore definisce un partitore che fornisce, per ogni posizione selezionata, una diversa tensione al pin analogico A8 di Arduino (M2), che discrimina così la posizione selezionata. Il pulsante pone a zero tale tensione, inducendo Arduino a rilevarne l'azionamento. Anche il potenziometro è letto da un pin analogico, ed ogni variazione del livello di tensione provocato da una manovra sul suo perno genera un comando diretto al modulo Radio per la definizione di un nuovo livello audio in uscita. L'amplificazione del segnale audio stereofonico emesso dal Chip Si473X è affidata ad un modulo

Listato 1 - Inclusione delle librerie

```
//-----
// Libraries
//-----

#include <SI4735.h>
#include <Rotary.h>
#include <EEPROM.h>
#include "patch_init.h" // SSB patch for SSB RX initialization
```

Listato 2 - Parametri relativi ad una banda

```
// Indexes for the band's parameters starting from BandsIni + Band * NumParamsPerBand enum E_BandParams {
  E_ModulationType=0,           // Modulation type (1: FM, 2: AM)
  E_ModulationSBmode,           // Modulation SB mode (1: no ssb)
  E_LastUserFrequency,           // Last user frequency
  E_InitialFrequency,            // Initial frequency
  E_FinalFrequency,              // Final frequency
  E_FrequencyStep,               // Frequency step
  E_Bandwidth,                   // Bandwidth applied at demodulation
};
```

▼ Listato 3 - Funzioni per la comunicazione col sottosistema Display

```
void NixSend(uint8_t nib) {
// Present a nibble to the Nixie display subsystem interface
uint8_t x;

    digitalWrite(PIN_CK,LOW);          // Put Clock line to low
    delayMicroseconds(1);              // Wait for output stabilisation

    // Set the Data pin composing the specified nibble value
    x = bitRead(nib,0); digitalWrite(PIN_D4,x);
    x = bitRead(nib,1); digitalWrite(PIN_D3,x);
    x = bitRead(nib,2); digitalWrite(PIN_D2,x);
    x = bitRead(nib,3); digitalWrite(PIN_D1,x);

    // Trigger the Nibble reading with a 0->1 level transition on the Clock pin
    delayMicroseconds(1);              // Wait digital output stabilisation
    digitalWrite(PIN_CK,HIGH);         // Low->High triggers IOC interrupt in he PIC
    delay(TnixCk);                     // Wait the PIC acquire the nibble
    digitalWrite(PIN_CK,LOW);          // Let the Clock line to Low
}

void NixAttr(uint8_t leds, uint8_t digTim, uint8_t ledTim) {
// Set Leds and set Digits and Leds blinking semi-period (ms)

    NixSend(14);                      // Leds and blinking semiperiods definition command
    NixSend(leds);                    // Leds 1, 2 or both (3)
    NixSend(digTim);                  // Digits' blink semi-period (x100ms)
    NixSend(ledTim);                  // Leds' blink semi-period (x100ms)
}

void NixReset() {
// Reset Leds and stop blinking
    NixSend(15); // Nixie reset command for the PIC
}

void NixDigits(uint8_t dig3, uint8_t dig2, uint8_t dig1, uint8_t dig0) {
// Send the specified digits to Nixie subsystem
int i;
uint8_t curdig;
uint8_t x;

    NixSend(13); // Digits setting command for the PIC
    for (i=0;i<NumDigits;i++) {
        switch (i) {
            case 0: curdig=dig0; break;
            case 1: curdig=dig1; break;
            case 2: curdig=dig2; break;
            case 3: curdig=dig3; break;
        }
        NixSend(curdig);
    }
}
```

pre-assemblato (M4). Tra Arduino ed il modulo Radio si interpone un modulo Level Shifter (M1) che si occupa di adattare i livelli di tensione sulle linee dati traslandoli verso 5V oppure verso 3.3V se diretti al Si473X, visto che tale circuito integrato lavora con questa tensione. Oltre alle linee dati di comunicazione tra Arduino e Si473X, il modulo radio necessita di una alimentazione di 3,3V, che è offerta in output da un pin di Arduino, ed è diretta anche al Level Shifter.

L'antenna esterna si collega al connettore J3, sul pin AM o FM in funzione della banda. Se l'auto non dispone di doppia antenna, si può collegare la stessa ad entrambe le entrate.

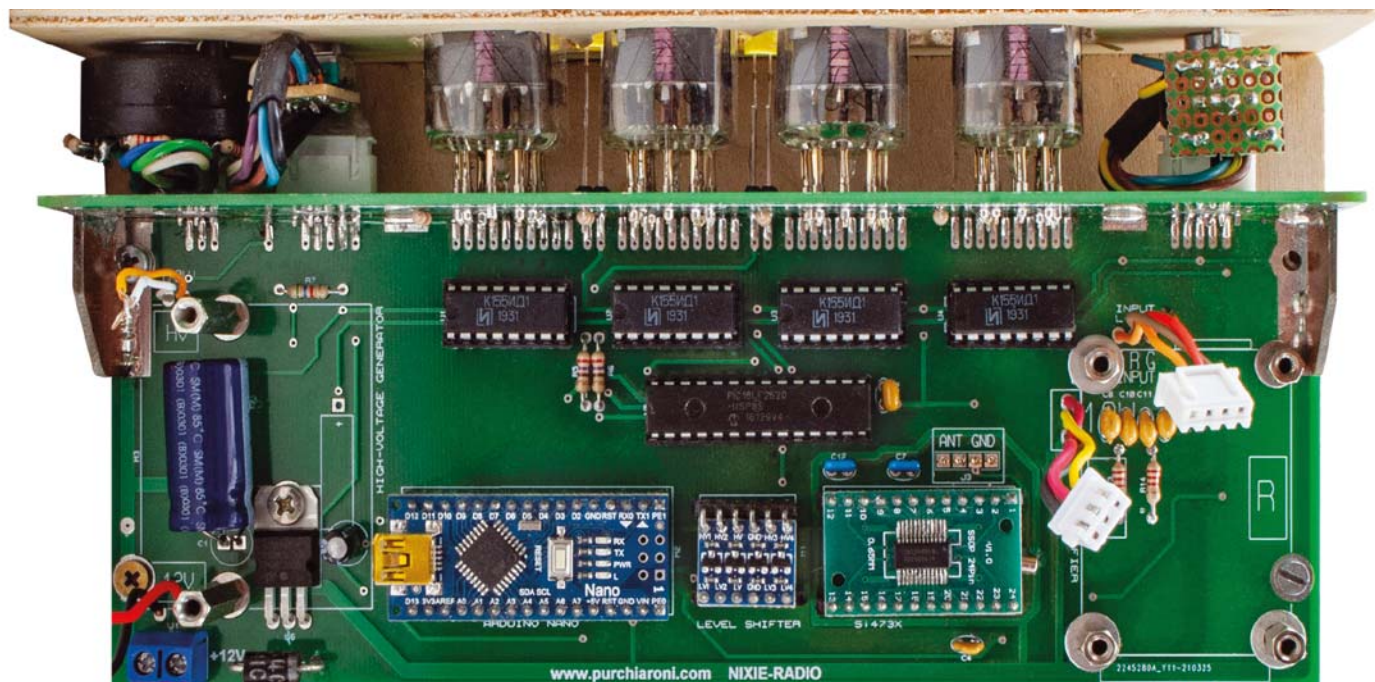
Le masse RF-GND vanno collegate al piano di massa della/delle antenne utilizzate, mentre GND è da collegare con la scocca del veicolo. Anche in tal caso è possibile collegare le due masse assieme, se non è possibile separarle a livello di antenna.

SKETCH ARDUINO

Il codice di gestione dell'apparecchio fa uso di una libreria Open Source sviluppata da Ricardo Lima Caratti, disponibile sul noto portale GitHub (<https://github.com/pu2clr/SI4735>).

Le chiamate alle funzioni di tale libreria permettono di pilotare il Chip Radio Si473X, dall'accensione





Elenco Componenti:

R1, R2, R3, R4: 8,2 kohm
 R5, R6: 220 ohm
 R7: 5,6 kohm
 R8, R9, R10, R11, R12, R13, R14: 2,2 kohm
 RV1: 50 kohm potenziometro lineare con interruttore
 C1: 2200 μ F 25V elettrolitico

C2: 100 μ F 16 V elettrolitico
 C3, C13: 100 nF multistrato
 C4: 22 nF multistrato
 C5, C6: 22 pF ceramico
 C7, C12: 470 pF multistrato
 C8, C9: 2,2 μ F multistrato
 C10, C11: 10 nF multistrato
 LD1, LD2: LED giallo 3 mm

D1: 1N5408
 D2: zener 5,1 V - 0,25 W
 X1: quarzo per orologi 32768 Hz
 DIS1, DIS2, DIS3, DIS4: tubo Nixie IN12
 U1, U2, U3, U4: 74141
 U5: PIC18F2620
 U6: 7805
 U7: Si4730 o Si4735

all'operatività run-time.

Lo sketch *NixieRadio.ino*, disponibile per il download, inizia come si vede nel **Listato 1**, con l'inclusione della suddetta libreria, nonché di quelle necessarie alla gestione della EEPROM, per il salvataggio delle memorie utente e delle impostazioni, e dell'encoder rotativo. Viene inoltre inclusa una Patch per il firmware del modulo, da caricare in RAM ad ogni accensione, necessaria per gestire la demodulazione delle Bande Laterali.

Le bande radio sono definite con le porzioni dedicate dei parametri di lavoro visti nella Tabella 3, i cui puntamenti sono definiti da un "Enumeral" mostrato nel **Listato 2** dedicato più un offset dipendente dalla banda in uso. Sono definiti tutti i parametri relativi alla modulazione, all'intervallo di frequenze ed alla larghezza di banda. È presente anche una flag che indica se è possibile utilizzare la modalità di demodulazione SSB.

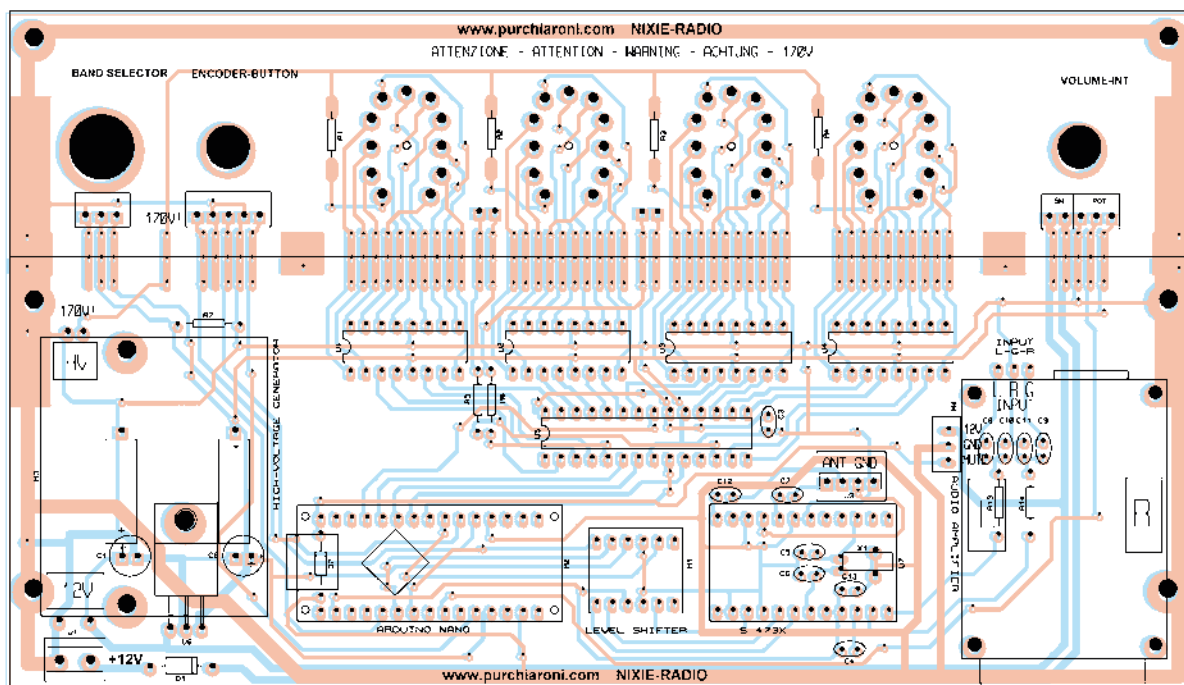
La visualizzazione della frequenza e dei parametri

di lavoro è affidata al Sottosistema Display, col quale lo sketch dialoga grazie alle funzioni riportate nel **Listato 3**.

La gestione dei pulsanti (Encoder e Opzioni) prevede l'uso di un automa a stati che discrimina la semplice pressione, dal doppio click, alla pressione prolungata. I valori temporali per la distinzione tra le tre modalità di utilizzo dei pulsanti sono configurabili nella modalità Impostazioni. La funzione GetButton mostrata nel **Listato 4** gestisce lo stato del pulsante indicato.

La funzione SetBand legge la tensione in uscita dal partitore definito dalla posizione corrente del Commutatore di Banda, ed invoca la funzione SetRadio se viene confermato due volte lo stesso valore analogico diverso dal precedente. La chiamata a SetRadio invia i parametri di banda al Chip Radio per impostare la nuova Banda (**Listato 5**).

La funzione SetRadioParameters mostrata nel **Listato 6** invia al chip Si473X i parametri di funzio-



M1: level shifter 5 V to 3,3 V bidirezionale
M2: Arduino Nano
M3: NCH6100HV
M4: DW-0425 (vedere box)
E1: encoder rotativo con pulsante
P1: pulsante N.A.
J1: morsetto 2 poli 5 mm
J2: connettore JST XH 3 poli M 90° - 2,54

mm (vedi testo)
J3: strip femmina 4 vie
J4, J5: connettore JST XH 5 poli M - 2,54 mm
Varie:
- connettore JST XH 3 poli F 2,54 mm con cavetti 10 cm
- strip maschio 4 vie

- connettore JST XH 5 poli F 2,54 mm con cavetti 10 cm
- altoparlante 8 ohm 10 W
- circuito stampato S1591 (194 x 111 mm)

**PIANI DI MONTAGGIO
RIDOTTI AL 80%.**

namento per la banda scelta.

Se necessario carica nella RAM del chip anche la patch per la gestione delle bande laterali.

Il loop principale di gestione opera come automa a stati e, a seconda dello stato corrente e delle azioni utente, gestisce le transizioni verso un nuovo stato

ed attua l'operazione richiesta.

Il **Listato 7** riporta il codice di gestione per il solo stato nominale.

Ricordiamo che i sorgenti completi sono scaricabili dal sito della rivista, come anche i compilati per il caricamento diretto sui microcontrollori impiegati.



Fig. 1
Frontalino.

↓ Listato 4 - Gestione dei pulsanti

```
void GetButton(int ButIdx, int Event, int Dbl) {
// Detect button's changes and set its status
// If Dbl=1 then Wait for the "double click" timeout (otherwise assume a single click)

// Set new event
but[ButIdx].But = Event;

// Manage button changes. ButSts must be reset to BUT_SBY by main after treatment
switch (but[ButIdx].ButSts) {
case BUT_SBY:
    if ((but[ButIdx].PrvBut == 0) && (but[ButIdx].But == 1)) { // Button pressed now
        StartTimer(ButIdx); // Start the Timer at presson time
        but[ButIdx].ButSts = BUT_PRESSED; // Declare first presson occurred
    }
    break;
case BUT_PRESSED:
    if (but[ButIdx].ButElapsed > *ButLong[ButIdx]) {
        but[ButIdx].ButSts = BUT_LONG;
        Serial.print("LONG bt"); Serial.println(ButIdx);
    } else if ((but[ButIdx].PrvBut == 1) && (but[ButIdx].But == 0)) {
        but[ButIdx].ButSts = BUT_PRESSED_SHORT;
    }
    break;
case BUT_PRESSED_SHORT:
    if (but[ButIdx].ButElapsed > *ButDouble[ButIdx]*Dbl) {
        but[ButIdx].ButSts = BUT_SHORT;
        Serial.print("SHORT bt"); Serial.println(ButIdx);
    } else if ((but[ButIdx].PrvBut == 0) && (but[ButIdx].But == 1)) {
        but[ButIdx].ButSts = BUT_DOUBLE;
        Serial.print("DOUBLE bt"); Serial.println(ButIdx);
    }
    break;
}

but[ButIdx].PrvBut = but[ButIdx].But; // Backup used event
if (but[ButIdx].ButSts < BUT_SBY) {
    but[ButIdx].ButElapsed = GetTimer(ButIdx); // Collect time elapsed in temporary status
}
}
```

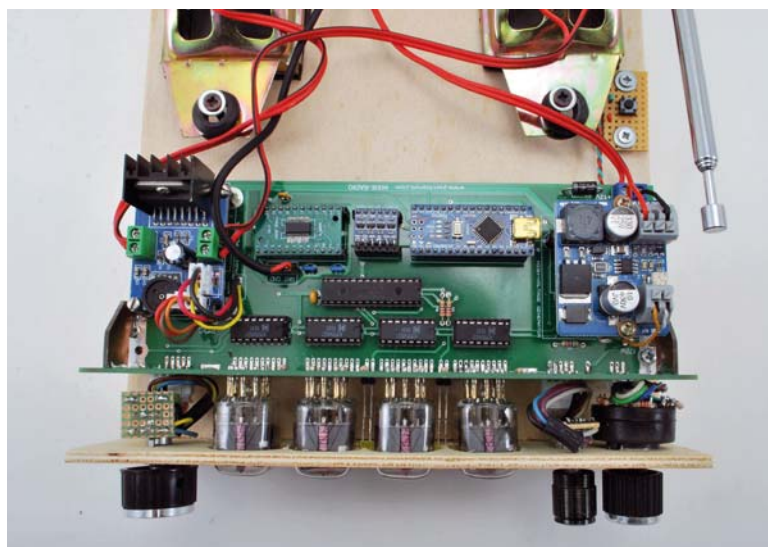
↓ Listato 5 - Gestione del commutatore di banda

```
void SetBand() {
// Send the new band to the Radio module
// Do it only on a manual change of the switch position
// Promote the proposed band only after a second confirmation
int v;
int8_t n, i, ip, b[2];

for (n=0;n<2;n++) { // Need two confirmations for band selection
    v = analogRead(PIN_BND); // Read voltage divider level
    for (i=0;i<NumBands;i++) {
        if ((v>vBand[i]-vGapBand)&&(v<vBand[i]+vGapBand)) break;
    }
    if ((i>=NumBands) || (i == bandPrv)) break; // No changes, or Search failed
    else if (n==0) {
        ip = i; // Store first reading for comparison
        delay(5); // Wait a little bit
    }
}
// Change band if different, after two confirmations
if (ip<NumBands) {
    if ((i<NumBands) && (n == 2) && (i == ip) && (i != bandPrv)) {
        Serial.print("SetBand:"); Serial.println(i);
        curband = i;
        SetRadio(); // Send the new band settings to the radio module
        bandPrv = i;
    }
} else {
    Serial.print("BndErr "); Serial.println(v);
}
}
```

REALIZZAZIONE PRATICA

Passiamo adesso alla costruzione dell'apparecchio radio, che prevede la preparazione di un circuito stampato che dopo l'incisione e la foratura va diviso in due parti, da unire a 90 gradi sul bordo dotato di piazzole appositamente disegnate, come mostrato nel piano di montaggio che trovate in queste pagine. Per realizzare il PCB dovete scaricare dalla sezione Download del nostro sito www.elettronicaain.it, tra i file di questo progetto, le due tracce lato rame, stamparle per ottenere le pellicole e procedere quindi con la fotoincisione. Ottenuto e separato lo stampato, potete montarvi i componenti occorrenti iniziando da quelli a più basso profilo; per quelli polarizzati, come condensatori elettrolitici, diodi, integrati ecc. dovete seguire scrupolosamente l'orientamento indicato nel piano di montaggio. Per rinforzare il collegamento meccanico sono state predisposte ulteriori ampie sezioni da stagnare. Un ulteriore consolidamento è dato da "spallette" laterali e, alla fine del montaggio, il risultato sarà molto robusto e funzionale. La parte verticale del circuito stampato ospita i quattro Tubi Nixie, inseriti in zoccoli con contatti dorati, e le resistenze R1+R4 per la limitazione della corrente anodica. Le piste, una volta unita la



parte verticale con quella orizzontale, proseguono su quest'ultima verso il microcontrollore PIC che controlla il display, e verso il generatore della tensione anodica destinata ad illuminare le cifre da visualizzare. La parte verticale del CS ospita anche i connettori per il collegamento dei dispositivi di comando, ovvero potenziometro con interruttore,

 Vista dall'alto dell'inserzione delle Nixie nel frontalino.

Listato 6 - Impostazioni del chip radio

```
void SetRadioParameters() {
// Set the radio working parameters on change, where current value is different from default
// Call it before to write the changes to EEPROM
int i;
uint16_t x, hi, lo;

for (i=RadioIni;i<=BandsIni;i++)
if ((parVal[i].sts == 2) && (parVal[i].value != GetDefault(i))) {
x = parVal[i].value;
hi = x >> 8;
lo = x & 0x00FF;
Serial.print("SetRadioPar "); Serial.print(i); Serial.print(": "); Serial.println(x);
switch (i) {
case E_AM_AGC_OVERRIDE      : rx.setAutomaticGainControl(hi, lo); break;
case E_FM_AGC_OVERRIDE      : rx.setAutomaticGainControl(hi, lo); break;
case E_AM_CHANNEL_FILTER    : rx.setBandwidth(lo, hi); break;
case E_AM_SEEK_SNR_THRESHOLD : rx.setSeekAmSrnrThreshold(x); break;
case E_FM_SEEK_TUNE_SNR_THR  : rx.setSeekFmSrnrThreshold(x); break;
case E_AM_SEEK_RSSI_THRESHOLD : rx.setSeekAmRssiThreshold(x); break;
case E_FM_SEEK_TUNE_RSSI_THR  : rx.setSeekFmRssiThreshold(x); break;
case E_AM_AUT_VOL_CONT_MAX_GAIN : rx.setAvcAmMaxGain(x); break;
case E_FM_BLEND_STEREO_THR    : rx.setFmBlendStereoThreshold(x); break;
case E_FM_BLEND_MONO_THR      : rx.setFmBlendMonoThreshold(x); break;
case E_FM_BLEND_SNR_STEREO_THR : rx.setFmBlendSnrStereoThreshold(x); break;
case E_FM_BLEND_SNR_MONO_THR   : rx.setFmBlendSnrMonoThreshold(x); break;
case E_FM_BLEND_MULTIP_STER_THR : rx.setFmBlendMultiPathStereoThreshold(x); break;
case E_FM_BLEND_MULTIP_MONO_THR : rx.setFmBlendMultiPathMonoThreshold(x); break;
}
}
}
```


▼ Listato 7 - Automa di gestione (caso STS_RUNNING)

```

case STS_RUNNING: // Standard mode -----

    // Show the current frequency if changed
    if (SomethingChanged) {
        SomethingChanged = false;
        Display(bndFrequ(curband)/bndFrequDiv(curband), bndFrequDot(curband), RIGHT_JUSTF, NO_BLINK,
parVal[E_InitialZeroes].value);
    }

    GetButton(ENCODER_BUTTON, (digitalRead(PIN_ENC_BT)==LOW), 1); // Get encoder button pressed status
    (with double click management)
    GetButton(OPTION_BUTTON, (Button1Pressed), 0); // Get option button pressed status

    // Manage encoder's push button
    switch (but[ENCODER_BUTTON].ButSts) {
    case BUT_SHORT:
        but[ENCODER_BUTTON].ButSts = BUT_SBY; // Reset button status
        SeekStation(); // Search for the next station on current band
        break;
    case BUT_DOUBLE:
        but[ENCODER_BUTTON].ButSts = BUT_SBY; // Reset button status
        Status = STS_MEMSELECT;
        // Set frequency from memory (initial setting)
        StartTimer(WheelTimer); // Start timeout timer for wheel moving
        EncInc = 0; // Reset encoder counts
        Serial.println("-> MEMSELECT");
        SomethingChanged = true; // Force display refresh
        break;
    case BUT_LONG:
        but[ENCODER_BUTTON].ButSts = BUT_SBY; // Reset button status
        Status = STS_MEMSET;
        StartTimer(WheelTimer); // Start timeout timer for wheel moving
        SomethingChanged = true; // Force display refresh
        break;
    }
    // Manage the encoder rotations to change frequency
    GetEncoder(3); // Get encoder rotation steps (all speeds)
    if (EncInc != 0) { // Wheel moved
        // Update the frequency value using the encoder's increments
        bndFrequ(curband) += EncInc * bndFstep(curband);
        // Truncate the frequency inside the foreseen boundaries
        if (bndFrequ(curband) < bndFrequL(curband)) bndFrequ(curband) = bndFrequL(curband);
        else if (bndFrequ(curband) > bndFrequH(curband)) bndFrequ(curband) = bndFrequH(curband);
        Serial.print("SetFrq "); Serial.println(bndFrequ(curband));
        rx.setFrequency(bndFrequ(curband)); // Change the tuning frequency
        SomethingChanged = true; // Force display refresh
    }

    // Manage the option button
    switch (but[OPTION_BUTTON].ButSts) {
    case BUT_SHORT: // Change demodulation mode
        SomethingChanged = true;
        but[OPTION_BUTTON].ButSts = BUT_SBY; // Reset button status
        Status = STS_SHOWMODE; // Enter Show Modulation Mode status up to a timeout or action
        StartTimer(ShowModeTimer); // Start timer to show demodulation mode
        prvmode = curmode; // Save the current mode, to be restored on timeout
        break;
    case BUT_LONG:
        SomethingChanged = true;
        but[OPTION_BUTTON].ButSts = BUT_SBY; // Reset button status
        StartTimer(WheelTimer); // Start timeout timer on wheel moving
        Status = STS_CONF; // Enter configuration mode
        break;
    }
    break;

case STS_SHOWMODE: // Show demodulation mode Norm,LSB,USB -----

    Display(curmode, DOT_NONE, LEFT_JUSTF, NO_BLINK, NO_ZEROES); // Show the current demodulation mode

    GetButton(ENCODER_BUTTON, (digitalRead(PIN_ENC_BT)==LOW),0); // Get encoder button pressed status
    GetButton(OPTION_BUTTON, (Button1Pressed), 0); // Get option button pressed status

    // Manage encoder's push button
    switch (but[ENCODER_BUTTON].ButSts) {
    case BUT_SHORT:
    case BUT_LONG:

```

```

    but[ENCODER_BUTTON].ButSts = BUT_SBY; // Reset button status
    Status = STS_RUNNING;                // Back to nominal mode on push
    SetRadio();                          // Send the new setting to the Radio module
    break;
}

// Manage the encoder rotations to change mode
GetEncoder(1);                          // Get encoder rotation steps
if (EncInc != 0) {                       // Wheel moved
    SomethingChanged = true;             // Force display refresh
    StartTimer(ShowModeTimer);          // Reset timer
    curmode += EncInc;                  // increase/Decrease current mode
    InBounds(&curmode, 0, 3);            // Fit mode between the foreseen limits
    //if ((curmode<4) && (EncInc>0)) curmode++;
    //else if ((curmode>0) && (EncInc<0)) curmode--;
    Serial.print("SSB mode "); Serial.println(curmode);
}

// Manage timeout
if (GetTimer(ShowModeTimer) > parVal[E_ShowModeTimeout].value) {
    curmode = prvmode;                  // Ignore change and restore the saved value
    SomethingChanged = true;            // Force display refresh
    Status = STS_RUNNING;               // Back to default state
}
break;

```

commutatore per la selezione delle bande, encoder con pulsante. Il pulsante Opzioni è derivato direttamente dai piedini del commutatore, ed andrà posizionato sul frontalino o, come nel nostro caso, su un lato del mobile contenitore.

La parte orizzontale del CS ospita anche la sezione di alimentazione, sovrastata dal modulo generatore M3 della tensione anodica che si eleva di qualche centimetro sulla superficie grazie a colonnine metalliche. Questo modulo può essere sostituito con altri modelli ma dovrà infine disporre di collegamenti filari verso il CS, sia dal lato 12V che dal lato 170V. Occorrerà predisporre un adattatore ad-hoc per ottenere ciò. Troviamo poi connettori strip femmina per l'inserimento dei moduli Level Shifter (M1), Arduino Nano (M2) ed Amplificatore (M4). Questo montaggio modulare facilita gli interventi e le evoluzioni. Volendo sostituire il Chip Radio Si4730 con un Si4735, basteranno operazioni di disinserimento e reinserimento della nuova parte. Ovviamente questi chip SSOP vanno saldati su adattatori PDIP. Sempre sulla parte orizzontale troviamo i connettori per l'alimentazione a 12V (J1) e per l'antenna (J5) già descritta precedentemente. Sotto all'adattatore del Chip Radio troviamo il Quarzo a 32.765 Hz che fornisce il clock per le operazioni, con i suoi due condensatori di carico da 22 pF. Sotto all'amplificatore troviamo le reti RC formate da R13, C8, C10 sul canale audio sinistro e R14, C9, C11 sul canale destro. Gli altoparlanti vanno collegati direttamente al modulo amplificatore. L'impiego di moduli aventi un layout differente richiederà un adattamento meccanico ed elettrico.

IL CONTENITORE DELLA RADIO NIXIE

Per l'utilizzo pratico è necessario inserire il circuito con l'elettronica all'interno di un contenitore dal cui pannello frontale saranno accessibili comandi e visualizzatore Nixie; per darvi qualche suggerimento pubblichiamo alcune immagini del prototipo realizzato dall'autore, che ha un aspetto decisamente vintage e adatto al progetto. In **Fig. 1** potete vedere il relativo frontalino, che misura 6x19 cm, appartenente proprio a detto prototipo. Il mobile, che si sviluppa dietro a tale frontalino, potrà ospitare direttamente gli altoparlanti se l'auto è sprovvista di tali elementi, un filtro anti-disturbo sull'alimentazione se "sporca", un pre-amplificatore d'antenna (opzionale) ed infine la presa per l'antenna e per i 12V dell'alimentazione.

Proprio la presenza dell'amplificatore e degli altoparlanti consente l'utilizzo autonomo della radio, anche in casa. Bene, con questi suggerimenti abbiamo terminato; certi che farete un ottimo lavoro, non possiamo che augurarvi un buon ascolto! 



Cosa occorre?

Alcuni dei componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. In particolare Arduino nano (cod. ARDUINONANO) è in vendita a Euro 25,00 mentre l'amplificatore con TDA7297 (cod. AUDIOMODAMPLI) è disponibile a Euro 8,00. 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



INDISPENSABILE IN LABORATORIO!



L'alimentatore è lo strumento più importante di ogni laboratorio elettronico da cui si esige prima di tutto la possibilità di regolare con buona risoluzione la tensione e la corrente di uscita, requisiti soddisfatti da questi nuovi alimentatori. Mantengono i valori impostati in modo affidabile e indipendentemente dalle fluttuazioni della tensione di alimentazione o del carico collegato.



DISPONIBILE ANCHE NELLE VERSIONI:

DA 0-30 VDC / 0-10 A

COD. AL3010 € 128,00

DA 0-60 VDC / 0-5 A

COD. AL6005 € 118,00

€ 89,00

Cod. AL3005N



ALIMENTATORE LINEARE CON USCITA REGOLABILE 0-30 VDC / 0-5 A

Alimentazione: 110/220 VAC, 50/60 Hz

Uscita duale regolabile:

- Tensione: da 0 a 30 VDC

- Corrente: da 0 a 5 A

Risoluzione della tensione: 10/100 mV

Risoluzione della corrente: 1/10 mA

Precisione visualizzazione della tensione: $\pm 1\%$ +1 cifra

Precisione visualizzazione della corrente: $\pm 1\%$ +2 cifre

Dimensioni: 220x165x85 mm

Peso: 1,350 kg



Prezzi IVA inclusa.

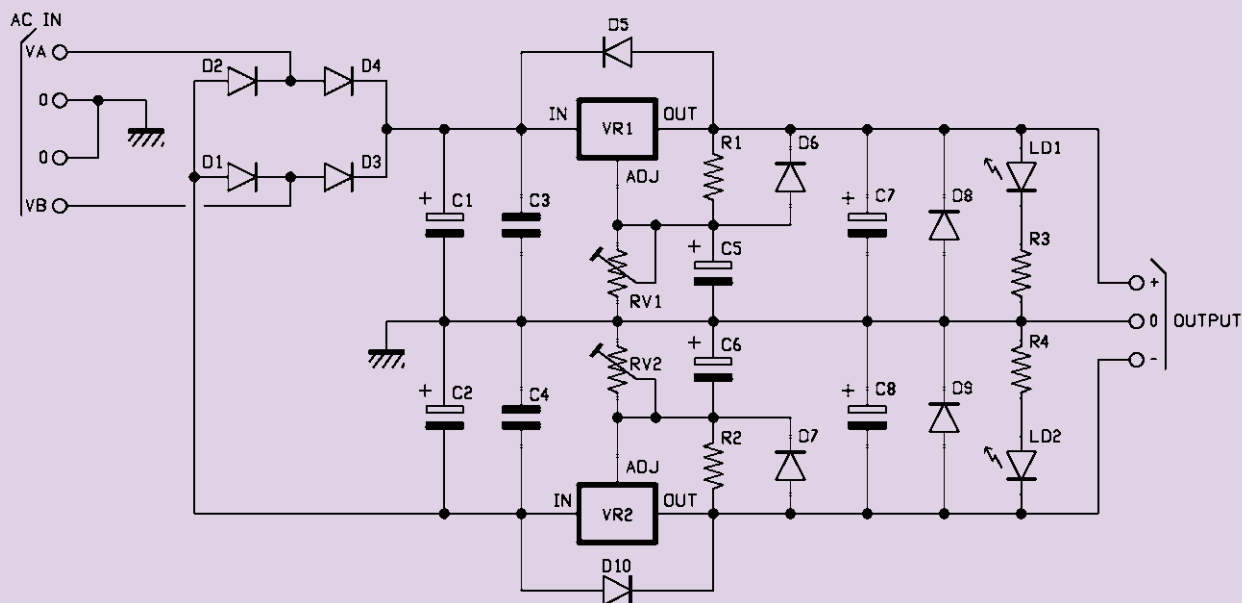
ALIMENTATORE STABILIZZATO DUALE E REGOLABILE



di **DAVIDE SCULLINO**

Con l'aggiunta di un trasformatore dotato di presa centrale, consente di fornire tensioni stabilizzate simmetriche da 3 a 15 Vcc e correnti di 1 A. Ideale come apparato da banco, può costituire lo stadio di alimentazione di amplificatori BF.

Tra i classici circuiti dell'elettronica spicca e continuerà ad essere presente l'alimentatore stabilizzato, sia esso AC/DC (ossia capace di trasformare una tensione alternata in continua stabilizzata) che DC/DC (vale a dire un regolatore di tensione) ma comunque lineare; infatti, sebbene oggi vada per la maggiore la circuitazione switching, esistono ambiti applicativi nei quali è determinante utilizzare circuiti lineari per mettersi al riparo dai disturbi -irradiati e residui del filtraggio della componente continua- che gli switching malgrado tutto implicano. Ci riferiamo, per esempio, agli amplificatori a bassissimo segnale d'ingresso, a quelli per trasduttori e per l'impiego nel settore medico e della misura. In generale, l'alimentazione lineare è da preferire quando si tratta di amplificare segnali analogici, non ultimi quelli audio, specie se di basso livello e quindi dovendo operare notevoli



amplificazioni, che oltre che quello del segnale, eleverebbero anche il livello dei disturbi, rendendo inaccettabile ciò che risulta poi all'uscita. Proprio per l'importanza che riveste l'alimentatore stabilizzato lineare, abbiamo pensato proporvene un nuovo progetto, che di fatto è un evergreen in quanto si basa su una circuitazione canonica, quasi scolastica (senza quasi...) e su una coppia di circuiti integrati regolatori di tensione, complementari, che sono gli storici LM317 (per il ramo positivo) ed LM337 (per quello negativo). Dalla presenza di due regolatori complementari avrete capito, prima ancora di gettare uno sguardo allo schema elettrico del nostro circuito, che quello proposto è un alimentatore stabilizzato duale, capace, cioè, di fornire una tensione simmetrica, positiva e negativa, alle proprie uscite. E dal fatto che utilizziamo due regolatori a tensione d'uscita regolabile, avrete compreso che quello qui descritto è un alimentatore variabile, che si presta quindi anche a realizzare un alimentatore da banco ad uscita regolabile entro un certo range di tensioni, ma anche lo stadio d'alimentazione di un preamplificatore o amplificatore audio con uscita accoppiata in continua, dove è necessaria un'alimentazione simmetrica per fare in modo che in assenza di segnale l'uscita sia circa a 0 volt. La simmetria e l'accompagnamento dei valori di uscita negativi a quelli positivi è ottenuto, nel nostro circuito, mediante due trimmer, che consigliamo di sostituire con un potenziometro doppio per essere certi che la variazione sia identica, nel valore assoluto, tra le due uscite.

SCHEMA ELETTRICO

Ma andiamo dunque a vedere di cosa si tratta, analizzando lo schema elettrico del circuito, riportato in queste pagine: la presenza di un ponte raddrizzatore discreto a quattro diodi (il classico Ponte di Graetz) ci fa capire che si tratta di un alimentatore AC/DC, ossia capace di accettare all'ingresso le tensioni alternate fornite da un trasformatore da rete con primario 220/50 Hz e secondario a presa centrale, di cui gli estremi vanno ai morsetti Va e Vb indifferentemente, mentre la presa centrale si connette ai punti 0.

Per quanto sia nativamente un alimentatore AC/DC, al circuito può essere applicata tensione sia continua che alternata; nel caso della continua, la presenza del ponte raddrizzatore fornisce il vantaggio di poterla applicare senza riguardo per la polarità. Dunque, analizziamo il funzionamento di questo circuito supponendo di alimentarlo con il secondario di un trasformatore a presa centrale: la tensione alternata viene raddrizzata dal ponte composto da D1, D2, D3, D4 in questo modo: quando il secondario ha polarità positiva su Va e negativa su Vb, la corrente fluisce dall'anodo del diodo D4 e attraverso il catodo raggiunge i condensatori di filtro C1 (elettrolitico) e C3, caricandoli e attraversandoli per poi passare in C2 e C4; più esattamente, la tensione presente nella metà di secondario connessa tra Va e 0 determina il flusso di corrente attraverso C1 e C2 e la massa, mentre la componente alternata dell'altra metà del secondario fluisce in C2-C4 e passa in Vb tramite il diodo D1.

In questa semionda, D2 e D3 sono interdetti, perché si troverebbero il catodo positivo rispetto all'anodo. Vediamo invece cosa accade nella semionda negativa, quando la polarità della tensione al secondario è positiva su Vb rispetto a Va, ovvero in metà secondario la polarità è positiva su Vb e nell'altro su 0: in questo caso ad andare in conduzione sono i diodi D2 e D3, perché l'anodo di quest'ultimo è positivo rispetto al catodo e la corrente fluisce in D3 per andare a caricare C1, C2, C3, C4 sempre con la stessa polarità della semionda precedente (il bello del ponte raddrizzatore è proprio questo: rende unidirezionale la tensione in uscita) e dal negativo dell'elettrolitico C2 raggiunge l'anodo del D2 per chiudersi su Va.

Volendo ragionare sulle metà del secondario, in semionda negativa quello che ha polarità positiva su Vb eroga corrente attraverso il diodo D3, caricando C1 e C3 e quindi chiudendo il circuito verso massa su 0 e la presa centrale; invece il mezzo secondario positivo su 0, eroga corrente da massa verso il positivo dell'elettrolitico C2, caricando quest'ultimo e C4, quindi la corrente si chiude attraverso l'anodo del diodo D2 verso Va.

Per effetto di quanto appena spiegato, avremo due tensioni riferite a massa: una positiva sul + dell'elettrolitico C1 (ovvero sui catodi di D3 e D4) ed una negativa sul - dell'elettrolitico C2 (ovvero sugli anodi di D1 e 2); siccome le metà del secondario del trasformatore erogano tensioni alternate uguali in ampiezza, le componenti positiva e negativa a valle del ponte di Graetz sono, teoricamente, di pari valore e tale valore (V) è dato dalla formula approssimata:

$$V = V_{eff} \times 1,4142 - 0,6$$

dove Veff è la tensione efficace di metà secondario, mentre 0,6V è la caduta di ogni diodo del ponte raddrizzatore a basse correnti dirette (può salire anche a 0,8V ad 1 ampere). La tensione efficace è quella normalmente dichiarata nel trasformatore, nel senso che quando si dice che un trasformatore ha il secondario da 12V, si intende 12Veff. Il fattore 1,4142 corrisponde al rapporto tra la tensione di picco dell'onda sinusoidale restituita dal secondario del trasformatore e il relativo valore efficace e rientra nella formula perché di fatto i condensatori, al netto delle cadute sui diodi raddrizzatori, vengono caricati con una tensione pari a quella di picco. Ciò detto, sappiamo cosa accade prima del blocco regolatore di tensione, quindi dobbiamo vedere cosa fa questo blocco, costituito dall'integrato

LM317 (VR1) per il ramo positivo e dall'LM337 (VR2) per quello negativo. Entrambi questi circuiti integrati sono regolatori di tensione, ossia ricavano e stabilizzano la differenza di potenziale alla loro uscita (OUT) a prescindere (entro certi limiti) dall'ampiezza della tensione all'ingresso, applicata tra il terminale IN e massa; il valore assoluto (prescindendo dal segno) della tensione d'uscita per ciascuno, si determina in base al potenziale che un partitore resistivo applica al piedino VADJ. Più esattamente, la tensione di uscita (Vo) si calcola con la formula:

$$V_o = V_{ref} (1 + R_2 / R_1) + (I_{adj} \times R_2)$$

considerando che Vref è una tensione di riferimento ricavata internamente all'integrato, pari a 1,25V e che tipicamente IADJ ammonta a 50 µA ed è praticamente trascurabile.

R1 ed R2 sono le resistenze collegate rispettivamente fra il terminale Output e ADJ e tra quest'ultimo e massa, pertanto nel caso di VR1 (ramo positivo dell'alimentatore) sono la resistenza R1 e il trimmer RV1, mentre per VR2 (ramo negativo) sono rispettivamente R2 ed RV2. Siccome vogliamo che il nostro alimentatore sia duale e fornisca quindi un'uscita simmetrica, i due trimmer sono di pari valore e così è anche per R1 ed R2.

Quindi, sostituendo i valori della sezione positiva nella formula sopra esposta, abbiamo una Vo (tensione fra OUT e massa) dell'LM317 pari a:

$$V_o = V_{ref} (1 + RV1 / R1) + (I_{adj} \times RV1)$$

Essendo RV1 un trimmer montato a reostato semifisso, il suo valore può variare tra 0 e 2,2 kohm, che nel nostro caso è il valore tra gli estremi del trimmer; con il trimmer tutto cortocircuitato (curso- re tutto a massa) la tensione di uscita è pari alla Vref, ossia 1,25V.

Il drop-out minimo degli integrati utilizzati è 3V, quindi nel fare i calcoli considerate che la tensione da applicare al terminale Input del regolatore (indifferentemente dal ramo considerato) deve essere almeno 3V superiore a quella voluta in uscita; essendoci di mezzo un ponte a diodi, la cui caduta di tensione è almeno 1,2V, la componente continua complessiva agli ingressi del circuito deve essere almeno 1,2V maggiore della somma dei valori assoluti delle tensioni d'uscita, ovvero ciascuna componente alternata fornita agli ingressi del circuito deve superare in valore massimo (questo si ottiene dividendo il valore efficace, che è quello normalmente

CARATTERISTICHE TECNICHE

Tensione di ingresso minima:
2 x 4 Vca

Tensione di ingresso massima:
2 x 24 Vca

Tensione di uscita minima:
2 x 12 Vcc

Tensione di uscita massima:
2 x 24 Vcc

Corrente massima erogabile:
2 x 1 A

Regolazione simmetrica
della tensione d'uscita



La coppia di integrati che abbiamo utilizzato nel progetto descritto in questo articolo è una delle più note dell'elettronica analogica, essendo sul mercato da decine d'anni; si tratta di due componenti identici, solo che uno opera su tensioni positive rispetto a massa (LM317) e l'altro su tensioni negative (LM337). Hanno entrambi tre terminali che fungono, uno da entrata, uno da uscita ed uno da ingresso della tensione di riferimento che determinerà il potenziale d'uscita. In Fig. A vedete lo schema a blocchi applicabile all'LM317.

Come vedete, all'interno dell'integrato, che è un regolatore serie con Darlington NPN che fa cadere su di sé la tensione in eccesso rispetto a quella impostata tramite il piedino ADJ, c'è un amplificatore

operazionale d'errore che sulla base del riferimento applicato, appunto, al piedino ADJ, pilota la base del Darlington e ne rileva il potenziale d'emettitore per compensare eventuali abbassamenti dovuto al carico, in modo da elevare la precisione di regolazione. Una resistenza in serie al pin di uscita (OUT o Output) consente di rilevare la corrente erogata, in modo da far intervenire la protezione Over Current che cortocircuita base ed emettitore del Darlington.

L'integrato LM317 si presta a fare anche da caricabatteria, ovvero da alimentatore limitato a un valore di corrente determinabile inserendo tra il piedino Out e il partitore di retroazione un resistore, come mostrato nello schema di Fig. B.

Fig. A
Schema a blocchi dell'LM317.

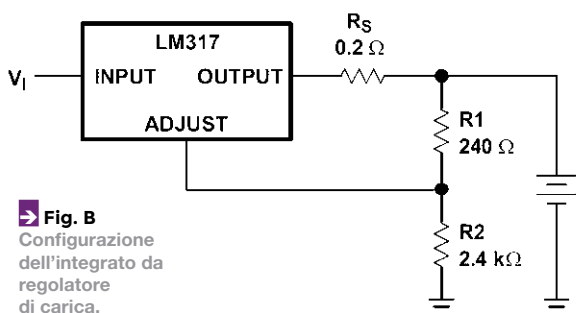
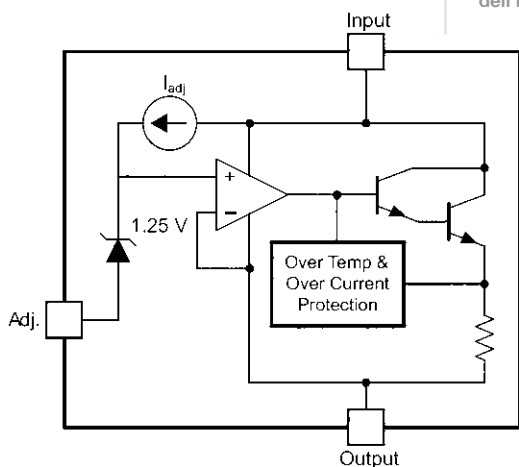


Fig. B
Configurazione dell'integrato da regolatore di carica.

dichiarato per il trasformatore, per 0,707...) 0,6V+3V quella continua desiderata all'uscita del regolatore. Se ai morsetti di ingresso si applica una componente continua simmetrica, ogni tensione deve essere 3,6V superiore a quella attesa alla rispettiva uscita. Per fare un esempio, se vogliamo avere 5V all'uscita positiva e altrettanti a quella negativa, applicando al ponte una componente alternata, questa dovrà avere valore efficace di circa 6+6V; se partiamo da una tensione continua simmetrica, ai morsetti Va e Vb andranno applicate due tensioni, riferite a massa e chiaramente opposte di segno, di 8,6 Vcc. Stabilito ciò, procediamo con l'analisi dello schema elettrico facendovi notare che per livellare ulteriormente la tensione disponibile a ciascuna uscita, ogni piedino di riferimento è filtrato da un condensatore elettrolitico, la cui funzione è attenuare quanto possibile eventuali disturbi e residuo di alternata che potrebbero introdursi nel piedino ADJ ed essere amplificati, determinando la sovrapposizione di una certa quantità di disturbo alle uscite del nostro alimentatore stabilizzato, ma anche esponendo al rischio di autoscillazione del circuito interno al regolatore.

Per ciascun regolatore sono stati previsti due diodi di protezione dall'inversione di polarità, per spiegare il cui funzionamento facciamo riferimento alla sezione facente capo all'LM317, fermo restando che quanto spiegato varrà, invertendo le polarità di tensioni e correnti, per quella negativa, costruita intorno all'LM337: il diodo D5, connesso tra l'uscita e l'entrata del regolatore, normalmente è interdetto perché la tensione sul piedino IN è maggiore di quella sull'OUT, però quando si spegne il circuito, ovvero si toglie alimentazione al ponte raddrizzatore, l'elettrolitico C7 ed eventuali condensatori nel circuito alimentato dal nostro alimentatore potrebbero rimanere carichi e polarizzare inversamente lo stadio d'uscita del VR1; in questo caso D5 entra in conduzione fa in modo che la corrente scavalchi l'LM317.

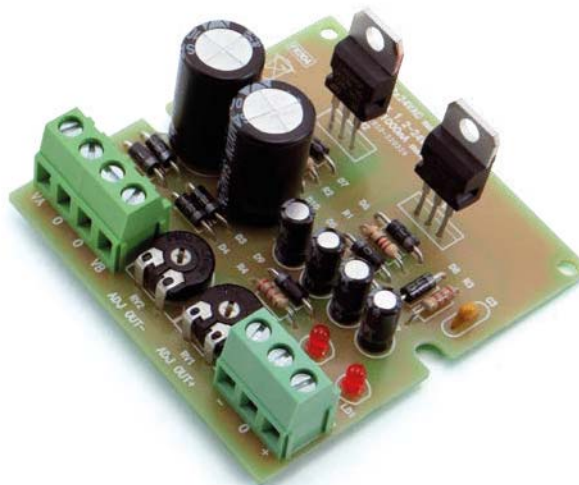
Quanto al diodo D6, serve per fare in modo che il condensatore di filtro del piedino ADJ si scarichi rapidamente allo spegnimento del circuito. Per completare la panoramica sui diodi, nel circuito d'uscita troviamo D8 e D9, collegati rispettivamente in parallelo alla linea positiva e a quella negativa, la cui funzione è proteggere i regolatori nel caso il circuito alimentato produca tensioni inverse (ad esempio se ha elementi reattivi come le induttanze, oppure se contiene motori elettrici) oppure se per errore si collegano le uscite dell'alimentatore a linee alimentate. Tali diodi rappresentano un'utile

protezione anche nel caso vada in cortocircuito un diodo del ponte raddrizzatore d'ingresso, allorché una sezione verrebbe polarizzata in entrambe le semionde dell'alternata; in questa evenienza la soppressione della componente inversa si otterrebbe in collaborazione con il diodo posto a cavallo tra IN e Out del regolatore.

Ciascuna delle uscite dell'alimentatore dispone di un LED di stato (LD1 per la positiva e LD2 per quella negativa) con resistenza di limitazione della corrente assorbita: il diodo luminoso indica, accendendosi, la presenza della tensione sulla rispettiva linea. Ciascuna uscita è completata da un condensatore elettrolitico di filtro del ripple e di eventuali disturbi sfuggiti alla restante parte del circuito: si tratta di C7 per la linea positiva e di C8 per quella negativa. Le tensioni fornite dal nostro alimentatore stabilizzato sono rese disponibili su una morsettiera d'uscita tripolare (OUTPUT) che riporta quella positiva (+) e quella negativa (-) riferite a massa (0).

REALIZZAZIONE PRATICA

Bene, ora che abbiamo descritto il circuito nel suo funzionamento, è il caso di vedere come si può costruirlo e utilizzarlo; al solito, per la realizzazione è stato previsto un PCB da ottenere per fotoincisione (o altra tecnica a voi più congeniale) dopo aver scaricato la traccia lato rame dalla sezione Download del nostro sito Internet www.elettronicaonline.it, dai file di questo numero della rivista; il circuito stampato è alla portata di tutti perché è a singola faccia e quindi non richiede allineamenti, doppia esposizione e via di seguito. Una volta che lo avete inciso e forato, potete disporvi i pochi componenti occorrenti, iniziando dalle resistenze e dai diodi al silicio (che dovrete orientare come indicato nel piano di montaggio, rammentando che la fascetta colorata sul loro corpo indica il catodo) quindi realizzando i ponticelli di interconnessione richiesti mediante spezzoni di filo in rame rigido del diametro di 0,5÷1 mm o avanzi di reofori tagliati da resistenze e diodi. Proseguite il montaggio con i due trimmer e i condensatori, prestando attenzione alla polarità degli elettrolitici, poi passate ai due LED, il cui lato smussato dovrà essere rivolto come indicato nel piano di montaggio che trovate in queste pagine. In ultimo montate i due regolatori integrati LM317 ed LM337, da inserire nei rispettivi fori del circuito stampato, entrambi in modo che l'aletta metallica sia rivolta al bordo della basetta, come mostrato nel solito piano di montaggio. Completate il montaggio inserendo e saldando le due morsettiere bipolari per l'ingresso dal trasfor-



Il circuito stampato montato.

matore con secondario a presa centrale e quella tripolare dell'uscita.

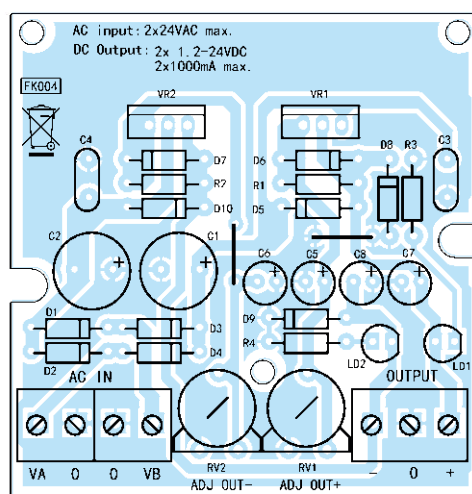
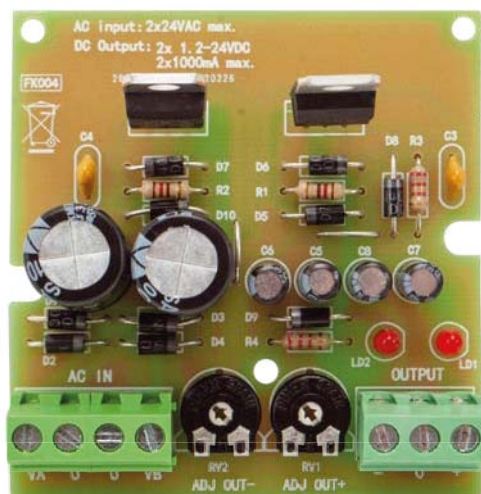
Per i regolatori VR1 e VR2 prevedete dissipatori ML26 TO-220 per regolatori di tensione, laddove vogliate prelevare correnti superiori a 500 mA con drop-out (ossia differenza di tensione fra ingresso e uscita) superiore a 7V. Più esattamente, la potenza massima dissipabile da LM317 ed LM337 in contenitore TO-220, come quelli da noi utilizzati, è di circa 3,5 W a 30 °C di temperatura ambiente e senza alcun radiatore di calore applicato; considerato che la resistenza termica tra giunzione e contenitore è di circa 4 °C/W, il dissipatore, ovvero la sua resistenza termica (θ_{ra}) si calcola in funzione della potenza da dissipare, imponendo una temperatura ambiente di 30 °C e conoscendo la massima temperatura di giunzione dell'integrato, che è 150 °C:

$$\theta_{ra} = (150 - 30) ^\circ\text{C} / P_d$$

dove P_d è la potenza da dissipare. Quest'ultima è data dalla caduta di tensione fra ingresso e uscita del componente, moltiplicata per la corrente erogata, che con buona approssimazione possiamo ritenere pari a quella in entrata. Supponendo di voler prelevare in uscita una tensione duale di $\pm 5\text{V}$, avendo all'ingresso 11V, il drop-out (caduta di tensione fra ingresso e uscita) è pari a 6 volt; volendo prelevare la massima corrente di uscita, ossia 1 ampere, questi valori si traducono in una potenza dissipata sui regolatori LM317 ed LM337 pari a: $P_d = 6\text{V} \times 1\text{A} = 6\text{W}$

Lavorando nelle condizioni di temperatura suaccennate, ossia avendo un salto termico massimo ammissibile di 120 °C tra chip e ambiente, la resistenza termica complessiva sarà di:

$$\theta_{ja} = 120 ^\circ\text{C} / 6\text{W} = 20 ^\circ\text{C/W}$$



Elenco Componenti:

- | | |
|---|---|
| R1, R2: 120 ohm | VR1: LM317 |
| R3, R4: 2,2 kohm | VR2: LM337 |
| RV1: Trimmer 2,2 kohm MO | LD1, LD2 LED: 3 mm rosso |
| RV2: Trimmer 2,2 kohm MO | |
| C1, C2: 1000 µF 35 VL elettrolitico | Varie: |
| C3, C4: 100 nF multistrato | - Morsettiera 2 poli passo 5 mm (2 pz.) |
| C5, C6, C7, C8: 10 µF 63 VL elettrolitico | - Morsettiera 3 poli passo 5 mm |
| D1, D2, D3, D4, D5: 1N4007 | - Circuito stampato FK004 (62x64mm) |
| D6, D7, D8, D9, D10: 1N4007 | |

Avendo una θ_{jc} , ossia una resistenza termica fra chip e contenitore, di 4 °C/W e trascurando la resistenza termica di accoppiamento tra la parte metallica (aletta) del regolatore e superficie del dissipatore (a patto di interporre della pasta termica in silicone o a base d'argento...) si ottiene una resistenza termica del dissipatore di:

$$\theta_{ra} = \theta_{ja} - \theta_{jc} = (20 - 4) \text{ °C/W} = 16 \text{ °C/W}$$

Con il nostro dissipatore standard ML26 a forma di "U", siamo al limite ma dovremmo farcela.

Naturalmente questo è solo un caso d'uso, ma l'esempio vale, sostituendo i valori che di volta in volta vi serviranno, per ogni condizione operativa e vi permette di fare i dovuti calcoli termici.

Per evitare eccessive perdite e dissipazioni di calore da parte degli integrati regolatori, se l'applicazione cui destinerete il circuito prevede una tensione di alimentazione nota e fissa, raccomandiamo di scegliere un trasformatore la cui tensione efficace di ogni metà secondario, moltiplicata per 1,4142, non superi di oltre 6V la tensione che desiderate avere su ciascuna uscita.

Infine, una nota riguardante i due trimmer; se volete realizzare un alimentatore da banco per alimentare i circuiti in prova, vi conviene montare, al posto dei due trimmer RV1 ed RV2, un doppio potenziometro, possibilmente di precisione, in modo che agendo sul perno si possano variare simmetricamente le tensioni presenti all'uscita, in maniera il più possibile uguale. Se utilizzate un doppio potenziometro avete anche il vantaggio di non dovervi preoccupare, prima di collegare l'utilizzatore, di equilibrare le tensioni d'uscita; questo vale principalmente se dovete alimentare un circuito che prevede un'alimentazione simmetrica con poca tolleranza. Se invece mantenete i due trimmer, ricordatevi di regolare RV1 ed RV2 ponendo un multimetro all'uscita corrispondente (+ per RV1 e - per RV2) e facendo in modo che le due tensioni siano uguali in valore assoluto (trascurando il segno) prima di collegare il circuito da alimentare. Rammentate che per misurare la tensione dell'uscita positiva dovete applicare il puntale positivo (rosso) del tester sul + della morsettiera OUTPUT e il negativo su 0, mentre per quella negativa il negativo andrà sul - della morsettiera e il positivo su 0. ■



Cosa occorre?

La scatola di montaggio dell'alimentatore duale (cod. FK004) è in vendita a Euro 12,00. I regolatori sono disponibili anche separatamente, in particolare l'LM317 (cod. LM317) è disponibile a Euro 0,70 e l'LM337 (cod. LM337) costa Euro 1,60. 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**

SMART HOME APPLICATION



di VINCENZO GERMANO
e MARCONATALE PARISE

Realizziamo l'APP
con Android Studio,
per la gestione
e il controllo del
sistema di domotica
micro:bit.



A

bbiamo introdotto nel precedente articolo (numero 264 di Elettronica In) il tema della domotica, in pieno sviluppo e fermento in questi ultimi anni, associata al crescente fenomeno "Internet of Things" (IoT) ovvero, l'Internet delle Cose, una rete che permette di instaurare una "connessione" tra gli oggetti d'uso quotidiano, permettendo la loro comunicazione e gestione. La domotica è l'esempio più attuale e calzante dell'IoT, infatti parlandone si pensa immediatamente a lampade controllate dallo smartphone oppure a sistemi di videosorveglianza, impianti di riscaldamento gestiti da remoto e così via. A oggi rappresenta vari processi e tecnologie tese a migliorare la qualità della vita nelle case e negli edifici, infatti grazie alla domotica si riescono a espandere le prestazioni ma anche le funzionalità degli impianti che compongono un'abitazione, andando ad ottimizzare i consumi mediante il risparmio energetico ma anche fornendo

strumenti aggiuntivi per la sicurezza, le comunicazioni, il controllo e il comfort di un'abitazione. Se si pensa a un impianto "tradizionale", esso può essere gestito mediante apparecchiature che sono incapaci di comunicare tra loro, basti pensare ad un comune interruttore che accende/spegne una lampadina alla sua pressione, mentre in un sistema domotico generalmente si ha un "punto di controllo" (come può essere un tablet, uno smartphone o altro) con il quale si gestiscono le varie funzionalità dei dispositivi connessi.

Tuttavia per permettere una comunicazione con i dispositivi (o anche tra essi) è necessario utilizzare un protocollo di comunicazione specifico, cablato, wireless, o anche misto.

Tutto ebbe inizio negli anni '70 con l'uscita del protocollo di comunicazione X10 su cablaggio, ma era affetto da varie problematiche intrinseche, perciò si è dovuto attendere gli anni 2000 per una diffusione sostanziale della domotica, anni in cui le tecnologie innovative Zigbee e Z-Wave arrivarono sul mercato, unendo il cablaggio elettronico e i segnali wireless; successivamente si sono aggiunti svariati protocolli come gli attuali Ethernet, WiFi e Bluetooth.

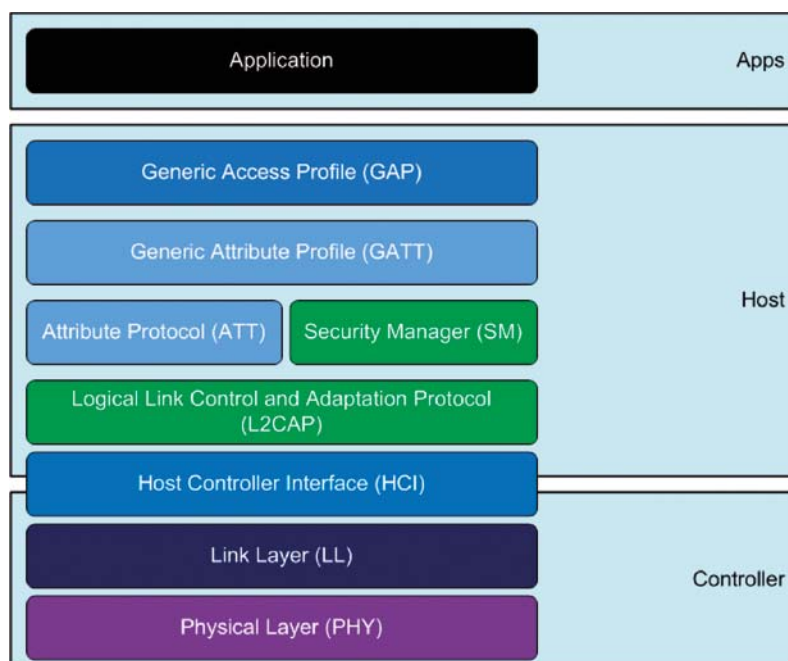
Oggigiorno i dispositivi utilizzati in quest'ambito possono avere a disposizione più di un protocollo, tuttavia il Bluetooth è di facile utilizzo e permette di avere elevate prestazioni ma anche bassi consumi, ecco perché lo abbiamo scelto per la nostra implementazione domotica, che ha necessità di

un punto di controllo ovvero quanto proposto e sviluppato nel corrente articolo, lo "Smart Home Application", un'applicazione installabile su smartphone che permette di centralizzare l'utilizzo delle funzionalità della casa domotica. Tuttavia prima di addentrarci nel dettaglio di sviluppo dell'applicazione mediante *Android Studio*, approfondiamo alcune nozioni di base del protocollo Bluetooth che ci serviranno successivamente.

ARCHITETTURA DEL BLUETOOTH

Il Bluetooth è un protocollo di comunicazione integrato in una grande varietà di prodotti di consumo, basti pensare a cuffie o casse audio, controller di videogiochi e molto altro; standardizzato per l'invio e la ricezione di dati tramite un collegamento wireless a 2,4 GHz, opera nella stessa banda di frequenza ISM (Industriale, Scientifica e Medica) in cui esistono anche protocolli radiofrequenza come ZigBee e WiFi, è sicuro e perfetto per le trasmissioni wireless a corto raggio (sotto i 100 metri), a bassa potenza e a basso costo tra dispositivi elettronici. In continua evoluzione da quando è stato concepito negli anni '90, per quanto alcune delle versioni precedenti sono ancora ampiamente utilizzate, il vero passo avanti lo ha compiuto con la versione v4.0, in cui si sono differenziate tre categorie: classico, ad alta velocità e a bassa energia. Il reale punto di forza di Bluetooth v4.0 è Bluetooth Low Energy (BLE), che ha portato a una massiccia revisione delle specifiche e dello standard, mirata ad applicazioni a bassissima potenza, infatti sacrifica la portata (portandola a 50 m invece di 100 m) e il "throughput" dei dati permettendo un notevole risparmio di consumo energetico. Il BLE è rivolto a dispositivi periferici che funzionano a batterie e non richiedono velocità di dati elevate o trasmissione dati costante, infatti è nato dalla necessità di connettere molteplici dispositivi all'interno di un sistema/rete come può essere in ambito domotica. Per quanto negli ultimi anni la versione di Bluetooth rilasciata sia arrivata alla 5 (ci sono anche ulteriori sotto-versioni come la 5.2) che apporta netti miglioramenti riguardo distanza, consumi e funzionalità, trattiamo alcune linee generali sul BLE, non legate alle versioni specifiche, che è doveroso esplicitare come dettaglio architetture per capire quelle che saranno le funzioni che andremo ad utilizzare all'interno della nostra applicazione, cercando di esplicitare solo ed esclusivamente le linee generali. Quanto riportato in **Fig. 1** sono gli strati del protocollo Bluetooth, partendo dal basso, i primi due sono *Physical Layer (RF and PHY)* e *Link*

Fig. 1
Architettura
Tecnologia BLE.



Layer, identificati come Controller (Controllo) mentre i blocchi superiori sono identificati come Host (o anche Server) ad esclusione dell'*HCI* (Host-to-Controller Interface) a cavallo tra i due che gestisce la comunicazione tra Controller e Host. Il *Physical Layer* è il componente fisico (come può essere l'antenna) che consente di trasmettere e ricevere fisicamente i dati; mentre il secondo componente è rappresentato dal *Link Layer* ovvero il primo "livello di controllo" che si occupa di gestire la ricezione e trasmissione pacchetti dati ma anche di gestire operazioni di *advertising*, *scanning* e *connection*. Iniziamo dall'*advertising*, che è una delle principali operazioni nella tecnologia BLE che fornisce ai dispositivi un mezzo o metodo per segnalare la loro presenza oppure permettere di stabilire una connessione. Mentre con il Bluetooth 4.0 all'interno del range di frequenza erano disponibili 40 canali dove solo 3 erano di advertising (utilizzati per stabilire una connessione o riconoscere la presenza di altri dispositivi), con l'evoluzione del Bluetooth 5.0 questi 3 canali sono diventati canali primari di advertising mentre i rimanenti 37 sono stati definiti come canali di advertising secondari così che la trasmissione dati diventasse più robusta.

Semplificando di molto lo standard, la **Fig. 2** mostra un esempio di un dispositivo (*Advertiser*) che segnala la sua presenza inviando pacchetti dati su uno o più canali di advertisement che altri dispositivi (*Scanner*) possono catturare. Lo *scanning* invece si basa sul principio dove uno scanner (quindi un altro dispositivo) è in ascolto di nuovi advertisement per scoprire nuovi dispositivi, stabilire una connessione oppure semplicemente per ricevere pacchetti dati. Questo può avvenire in modo "passivo" dove l'*advertiser* provvede ad inviare i segnali e lo scanner cerca di catturarli, oppure in modo "attivo" nel quale la ricerca del dispositivo si basa su

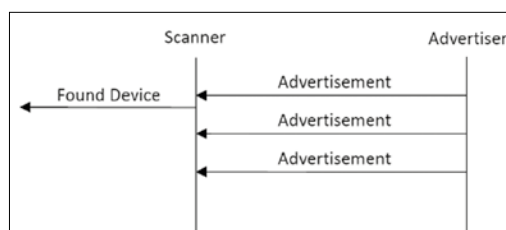


Fig. 2
Advertisement.

uno scambio di informazioni tra i dispositivi, il tutto avviato dal segnale di advertisement. Infine, la *connection* è la connessione tra i vari dispositivi, che tra le altre cose permette una trasmissione robusta ed affidabile attraverso l'uso di CRC (Cyclic Redundancy Check – Controllo di Ridondanza Ciclico) e ritrasmissione dati, per assicurarne la corretta consegna anche mediante procedure di criptaggio e de criptaggio. Passando ai layer superiori, il *L2CAP* (Logical Link Control and Adaptation Protocol) non ci dilungheremo in quanto agisce come protocollo per la gestione dei pacchetti dati; mentre il *GAP* (Generic Access Profile) fornisce accesso alle operazioni di link layer descritte in precedenza attraverso specifici requisiti per la trasmissione/ricezione dati e definisce le modalità e le procedure di ricerca, connessione ed accoppiamento (bonding) dispositivo. Mentre il protocollo *Attribute Protocol* (ATT), gestisce le informazioni su attributi che contengono lo stato del dispositivo e quindi la comunicazione tra server e client attraverso il canale di L2CAP. Esso definisce come il server dovrà memorizzare i dati di uno o più attributi oppure fornire risposte al Client che dovrà leggere le informazioni. In modo molto semplificato, la **Fig. 3** mostra un esempio di comunicazione Client/Server dove il primo può accedere ai vari attributi del server attraverso una richiesta che genera una risposta da parte di quest'ultimo.

Come esempio, un attributo può essere rappre-

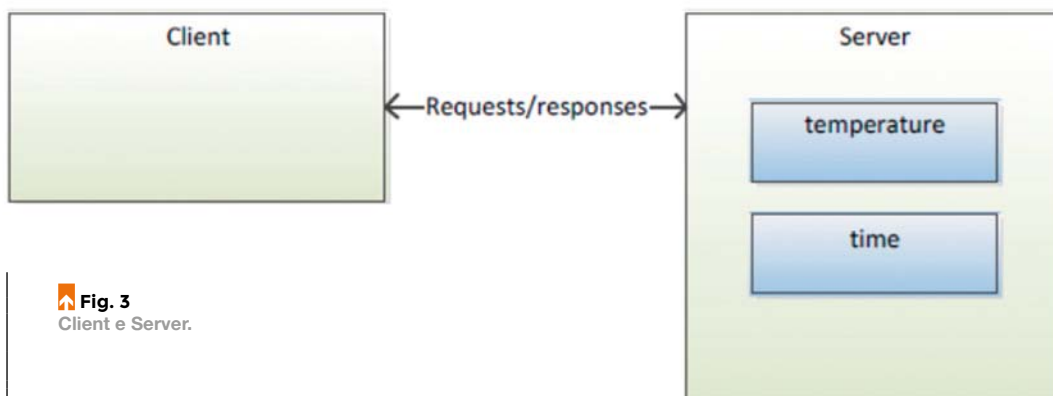


Fig. 3
Client e Server.

Tabella 1

HANDLE	UUID	VALUE
0x001	0x1804	0x0000
T0x0002	0x2a00	0x426c75656769676120546563686e6f6c6f6769657

sentato come in **Tabella 1**, dove:

- *Handle* è una chiave che consente al client di richiedere l'attributo
- *UUID* (Universally Unique Identifier) determina il tipo di attributo che si vuole usare
- *Value* è il dato che può avere una dimensione da 0 a 512 bytes

Infine il *GATT* (Generic Attribute Profile) utilizza il protocollo ATT e stabilisce una struttura comune per il trasporto e la memorizzazione dei dati. Quindi possiamo fare distinzione tra GATT server e GATT client dove il server memorizza dati trasportati su ATT ma può anche accettare ed inviare risposte al GATT client. I dati condivisi su ATT sono definiti attributi (formattati come servizi e caratteristiche). Come esplicitato in **Fig. 4**, nella quale viene rappresentato un classico esempio di struttura GATT server, i servizi sono collezioni di dati con una o più caratteristiche allo scopo di definire una specifica funzione (la frequenza cardiaca nell'esempio); mentre una caratteristica è un valore usato in un servizio che consente di mostrare e/o scambiare dati ma può contenere anche descrittori che consentono di configurare notifiche o indicazioni della caratteristica.

A questo punto possiamo procedere oltre avendo definito le linee generali di funzionamento del protocollo BLE, passando alla piattaforma Android Studio e alla realizzazione dell'applicazione Android per il nostro sistema di domotica.

ANDROID STUDIO

Come più volte anticipato, per la gestione di un sistema di domotica conviene avere un punto di controllo, perciò abbiamo pensato di sviluppare un'applicazione per smartphone con tutte le funzionalità del caso. Nella seguente parte dell'articolo presenteremo solo la gestione dell'illuminazione che si interfaccia con il sistema Microbit presentato nel precedente articolo (numero 264 di Elettronica In) e permette di accendere/spegnere l'illuminazione in base agli input ricevuti, ovvero il sensore crepuscolare o il dato ricevuto tramite protocollo bluetooth. Proprio per quest'ultima funzionalità abbiamo illustrato un'applicazione in grado di accoppiarsi al dispositivo Microbit ed inviare i dati necessari. In questo paragrafo vi guideremo nella realizzazione dell'applicazione presentando la piattaforma di programmazione *Android Studio*. Quest'ambiente di sviluppo integrato (IDE), open source (quindi disponibile gratuitamente) ideato da Google nel 2013, consente la programmazione di applicazioni Android in vari linguaggi come Java, Kotlin e C++. La piattaforma è disponibile per siste-

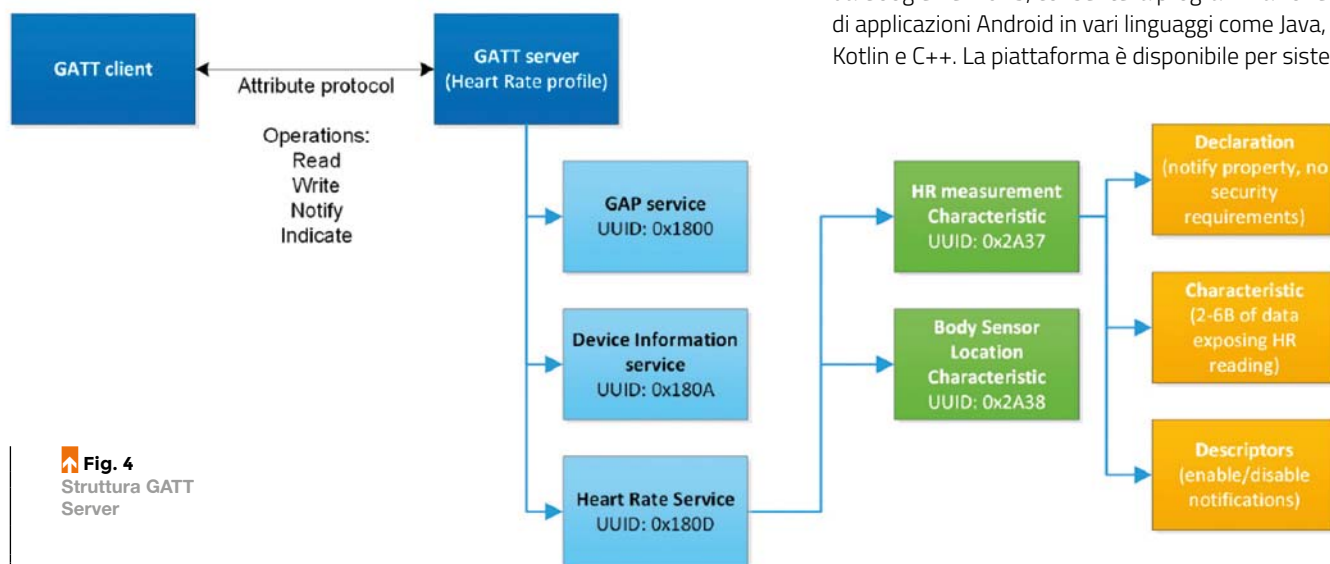


Fig. 4
Struttura GATT Server

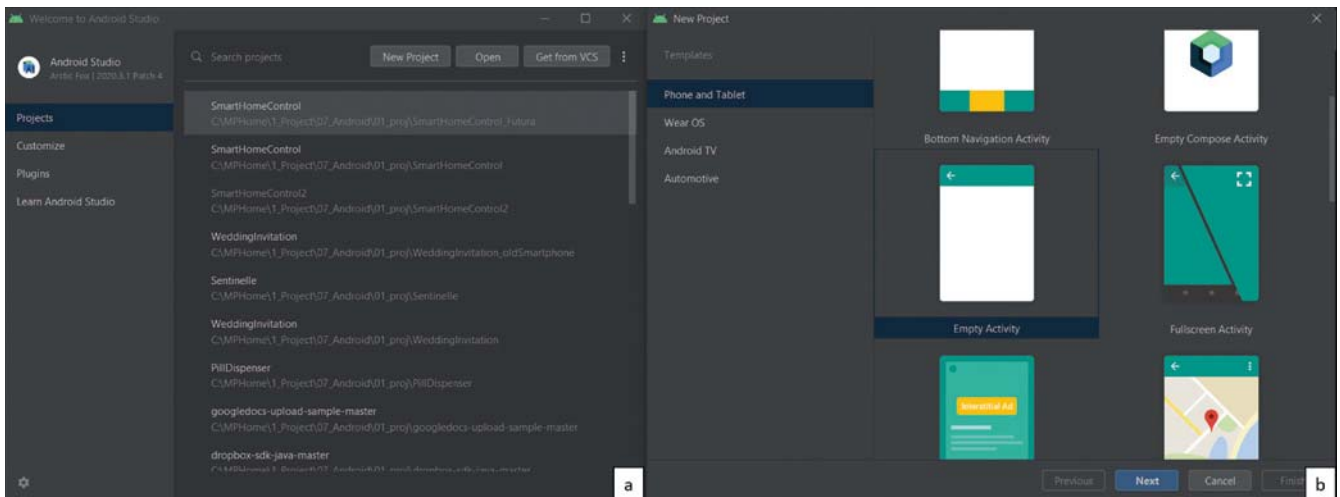


Fig. 5
Avvio
Programma.

mi operativi Windows, Mac e Linux ed è scaricabile tramite il seguente link ufficiale: <https://developer.android.com/studio#downloads>

Negli anni sono state introdotte molte migliorie che hanno reso l'ambiente di sviluppo sempre più user-friendly (di facile utilizzo) nella gestione delle pagine (layout) di cui si compone un'applicazione, come vedremo, per rendere l'attività di scrittura codice più leggibile e semplificare le attività di testing mediante l'emulatore di smartphone. A questo punto dopo aver scaricato l'ultima versione Android presente sul sito ufficiale, procedete con l'installazione, sulla quale non ci dilungheremo vista la facilità e i passi obbligati. Completata la procedura, avviate il tool Android Studio così da creare il progetto che si presenterà come mostra-

to nella **Fig. 5**. Cliccate su *New Project* (**Fig. 5a**) e successivamente selezionate "Empty Activity" e cliccate su *Next* (**Fig. 5b**).

A questo punto è necessario definire i parametri iniziali del nostro progetto, riportati in **Fig. 6**:

- *Name*: scegliamo *SmartHomeController*;
- *Package name*: verrà automaticamente compilato;
- *Save location*: il percorso dove verrà creato il progetto;
- *Language*: scegliate *Java*;
- *Minimum SDK*: rappresenta la minima versione Android dove potrà girare l'applicazione e la scelta ricade su *API 21: Android 5.0 (Lollipop)* che rappresenta un ottimo compromesso per quanto riguarda librerie disponibili e dispositivi

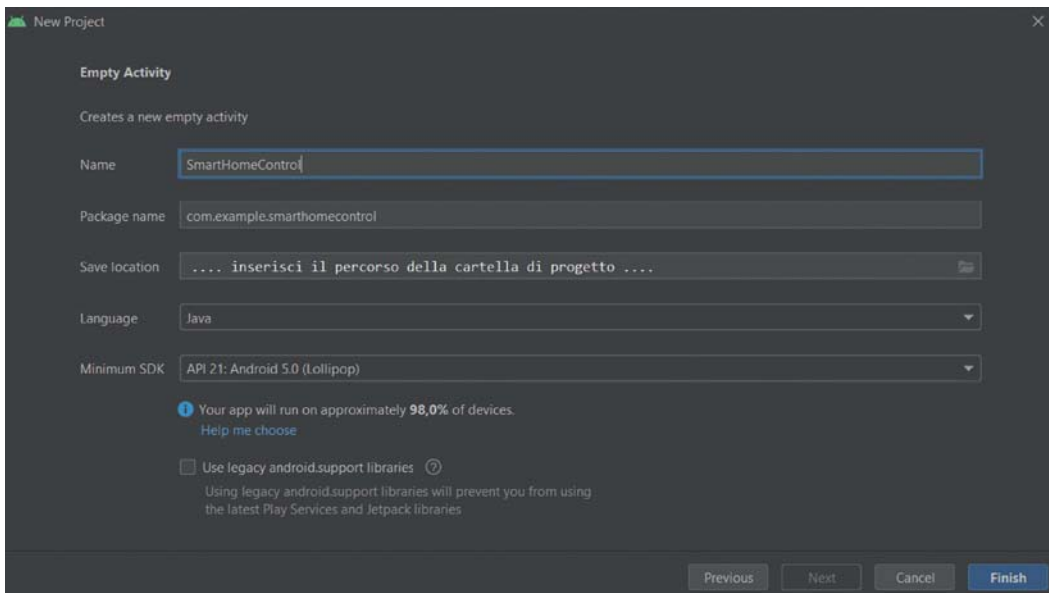


Fig. 6
Parametri
Iniziali
Programma.

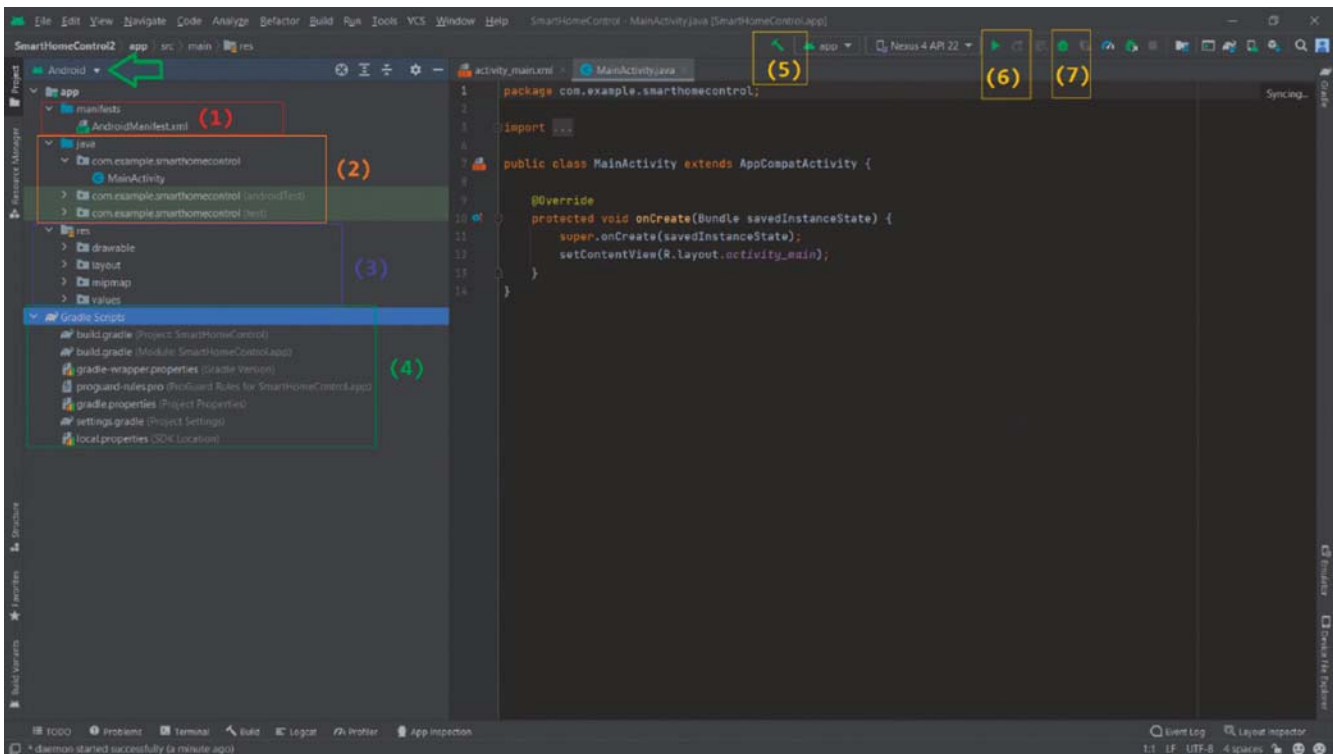


Fig. 7
Strumenti del
programma.

sul quale potrà essere installata (il tool suggerisce il 98%).

Alla pressione del tasto "Finish" il tool si presenterà come mostrato in **Fig. 7**. Android Studio consente di visualizzare il progetto in diversi formati e a tal proposito utilizziamo la visualizzazione *Android* indicata dalla freccia verde in **Fig. 7**. Passiamo all'analisi delle varie sezioni principali.

Sezione (1) – Android Manifest

Ogni progetto si compone sempre del file *AndroidManifest.xml* che è richiesto per dichiarare le seguenti informazioni:

- Componenti dell'app che include attività e servizi;
- I permessi di cui l'applicazione ha bisogno per accedere a parti protette del sistema oppure ad altre app;
- Hardware e software caratteristiche che l'app richiede e che influenzano i dispositivi che possono installare l'applicazione.

Sezione (2) – Java file

La visualizzazione permette di evidenziare il percorso principale del progetto raggruppando le attività (file) con estensione *.java* che definiranno le logiche della nostra applicazione. Come è possibile

notare sono presenti delle cartelle con le diciture (*androidTest*) e (*Test*) che vengono utilizzate per testare il codice realizzato ma che non sarà oggetto di trattazione.

Sezione (3) – res directory

La cartella di *res* contiene una serie di sottocartelle:

- *Drawable*: in ottica grafica raggruppa tutti una serie di formati immagine come bitmap file (*.png, *.jpeg), xml file oppure vettoriali in formato *.svg
- *Layout*: definisce le pagine della nostra applicazione in formato .xml
- *Mipmap*: è usata per definire l'icona dell'app
- *Values*: contiene vari file xml come string.xml, themes.xml oppure colors.xml, che consentono di modificare lo stile dell'applicazione.

Sezione (4) – Gradle

Questa sezione non sarà oggetto di modifica ma è importante tenerla in considerazione, perché rappresenta un set di strumenti di costruzione avanzata per automatizzare e gestire il processo di compilazione per la creazione dell'applicazione. Il processo di compilazione include i processi di *compiling* (per la traduzione del linguaggio ad alto livello in linguaggio macchina e per la generazione di un eseguibile), *linking* (che integra librerie e sottopro-

grammi nel linguaggio macchina) e l'*impacchettamento* del codice per la creazione del file .apk. Sempre all'interno di questa sezione è possibile definire un insieme di risorse e codice che potranno essere usati per tutte le versioni dell'applicazione.

Sezioni (5) – (6) – (7)

Le icone evidenziate rappresentano rispettivamente le procedure di *build*, *run* e *debug* dell'applicazione che verrà eseguita su dispositivo simulato/fisico oppure per analizzare e correggere eventuali errori. Analizzeremo tutte le sezioni descritte in dettaglio durante la realizzazione del progetto, pertanto procediamo con il primo passo che riguarda la creazione del layout (le varie pagine) dell'applicazione ovvero il contenitore di oggetti che consentiranno all'utente di interagire con l'applicazione per implementare le nostre logiche.

CREAZIONE DEL LAYOUT

Dalla sezione layout, con doppio click su "*activity_main.xml*" accediamo alla configurazione delle pagine della nostra applicazione; in **Fig. 8** sono disponibili gli strumenti principali che andremo ad utilizzare. La visualizzazione (layout) è configurabile tramite l'opzione "Split" evidenziata nell'area (2) che consente di mostrare la pagina in formato grafico (Design) ma anche il codice .xml (Code). All'interno del layout è possibile aggiungere i vari

oggetti disponibili nella *Palette* (3) come *TextView* (1) che potrà essere modificato attraverso la scrittura del codice direttamente su file ".xml" oppure mediante le proprietà disponibili nell'area *Attributes* (4). Procediamo con la realizzazione del nostro layout inerente alle funzionalità dell'applicazione e mostrato in **Fig. 9** dove vogliamo dare la possibilità all'utente di:

- Attivare la funzionalità Bluetooth tramite uno switch;
- Aggiungere un'immagine per simulare l'accensione o spegnimento della lampadina.

Quindi selezioniamo dall'area di *Design* l'oggetto *TextView* così da rimuoverlo tramite il tasto CANC ed aggiungere i nuovi oggetti. Dalla *Palette* cerchiamo l'oggetto switch che selezioniamo e trasciniamo all'interno dell'area di layout, inizialmente saranno presenti alcuni errori all'interno dell'area di codice dovuti alla mancanza di vincoli (constraint), ma questo è normale perché dobbiamo ancora definirli. Tramite la proprietà *text*, disponibile nell'area di codice, modifichiamo il nome dello switch in *Bluetooth* e modifichiamo la sezione l'id in "*@+id/Sw_ble*", proprietà importante che utilizzeremo in seguito. Torniamo sulla definizione di vincolo, ogni oggetto aggiunto all'interno del layout richiede la definizione di quella che sarà la sua posizione e questo può essere definito attraverso il *Constraint*

Fig. 8
Layout.

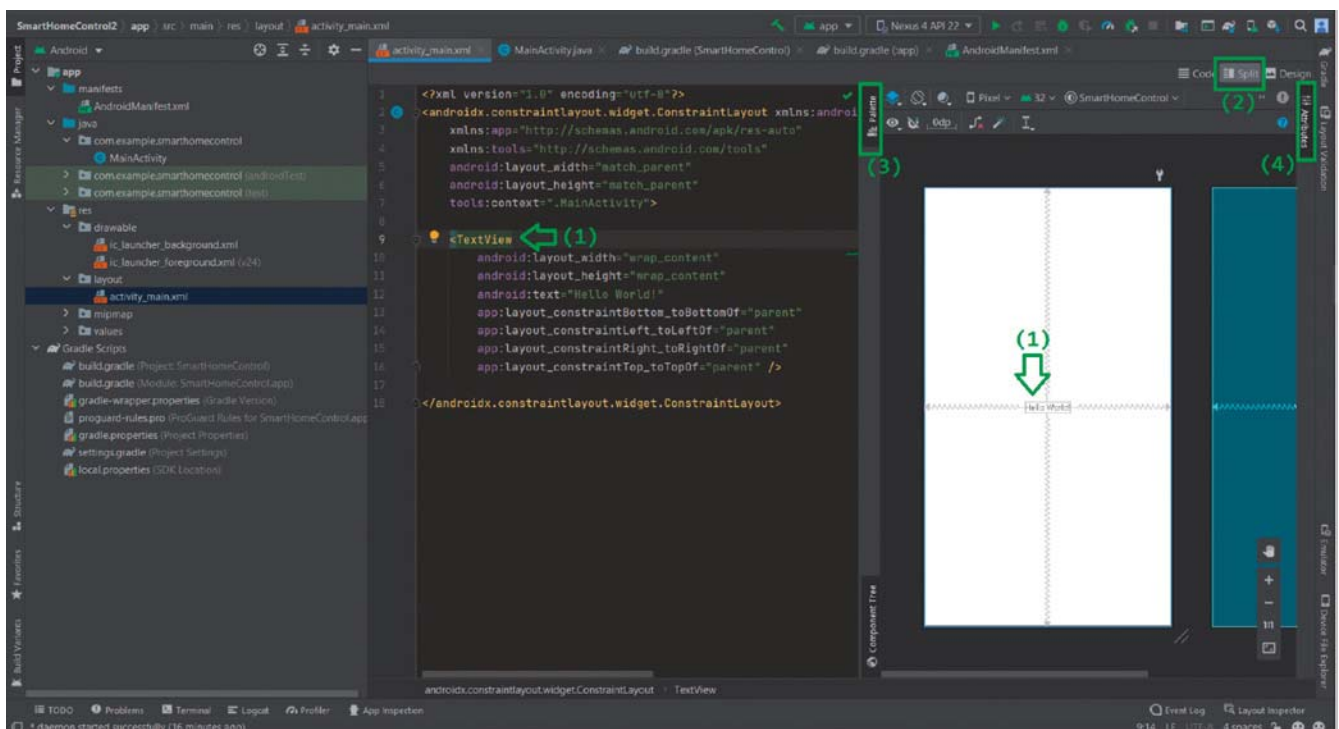


Fig. 9
Layout.



visibile nella sezione *Attributes* dove i pallini azzurri consentono di aggiungere i vincoli (**Fig. 10a**).

Eseguiamo la procedura per il lato sinistro ed il top dell'oggetto ed automaticamente verranno imposti i vincoli rispetto la nostra pagina identificata come *parent*. Modifichiamo il contenuto delle caselle con 16 per il lato superiore e 8 per il lato sinistro per imporre la distanza rispetto il parent, mentre nella sezione *Code* appariranno le diciture `android:layout_marginStart="8dp"` e `android:layout_marginTop="16dp"` come mostrato in **Fig. 10b**, dove *dp* (Density-independent Pixels) indica la distanza tra oggetti in termini di pixel.

Procediamo con aggiungere una nuova *TextView* che trasciniamo all'interno del layout ed associamo il nome "Status" con id "@+id/TV_status_connection", che utilizzeremo per aggiornare lo stato della connessione Bluetooth, infine definiamo i vincoli nella medesima modalità del precedente oggetto con la differenza che in questo caso il vincolo superiore non sarà *parent* ma bensì l'oggetto switch e quindi l'id `SW_ble` come mostrato in **Fig. 11**.

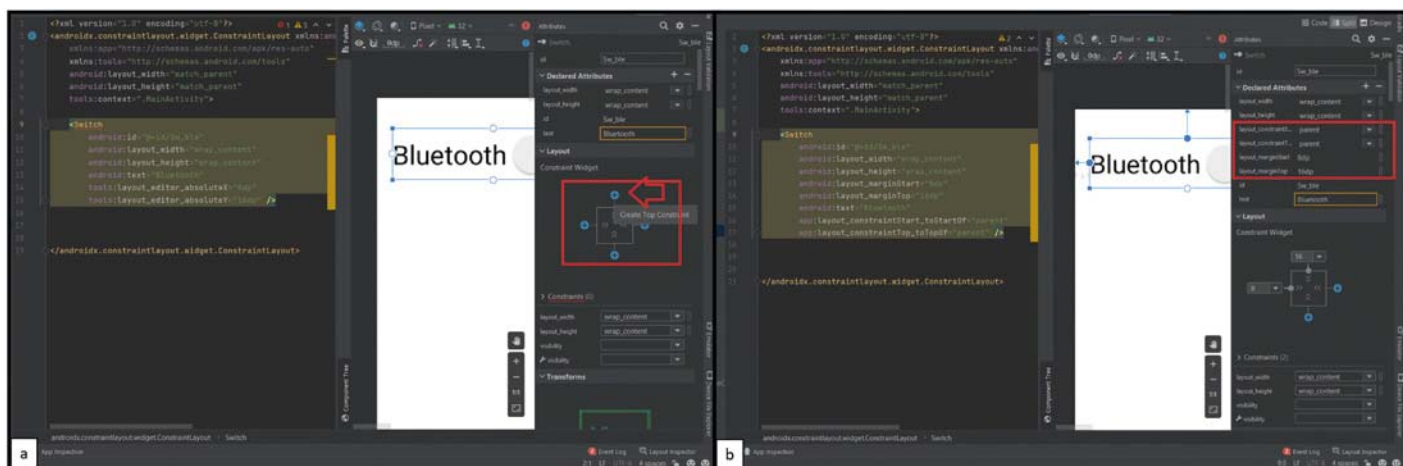
Fig. 10a/10b
Vincoli.

In linea con la **Fig. 9**, aggiungiamo un'ulteriore *TextView*, per guidare l'utilizzatore dell'applicazione nelle possibili azioni da eseguire, con vincolo alla base dell'etichetta *Status* e posizionamento al centro della pagina aggiungendo il vincolo destro e sinistro (rispetto al *parent*) con distanza di *Odp*. Associamo l'id "@+id/TV_info" ed il testo "click on image to turn Off or turn On your bulb".

Infine, per rendere più gradevole l'applicazione e fornire un colpo d'occhio riguardo lo stato dell'illuminazione, aggiungiamo un'immagine che rievoca una lampadina; a tale scopo è importante caricare all'interno del progetto i file che andremo ad utilizzare attraverso il direttorio *drawable*. Tramite il tasto destro del mouse sulla cartella d'interesse, selezionare *New – Vector Asset* e comparirà una finestra come visibile nella **Fig. 12**, da cui è possibile selezionare il file vettoriale *.svg (il file in **Fig. 12** sarà scaricabile dal sito ufficiale di Elettronica In insieme a tutto il resto del materiale) ed infine terminiamo l'operazione tramite il tasto *Finish*; comparirà all'interno della cartella di *drawable* il file in formato *.xml. Ritorniamo sul nostro layout e all'interno della *palette* selezioniamo l'oggetto *ImageView*, lo trasciniamo all'interno del layout in modo tale da far comparire una nuova finestra che ci inviterà a selezionare l'immagine:

- scegliamo l'immagine appena caricata `ic_bulboff`;
- vincoliamo l'oggetto alla base dell'etichetta con id "@+id/TV_info" con distanza di 24dp
- centriamo l'immagine aggiungendo un vincolo a destra e sinistra del *parent* con distanza *Odp*
- modifichiamo la dimensione dell'immagine con altezza di 100dp a larghezza di 80dp.

Per completezza riportiamo nel **Listato 1** il codice



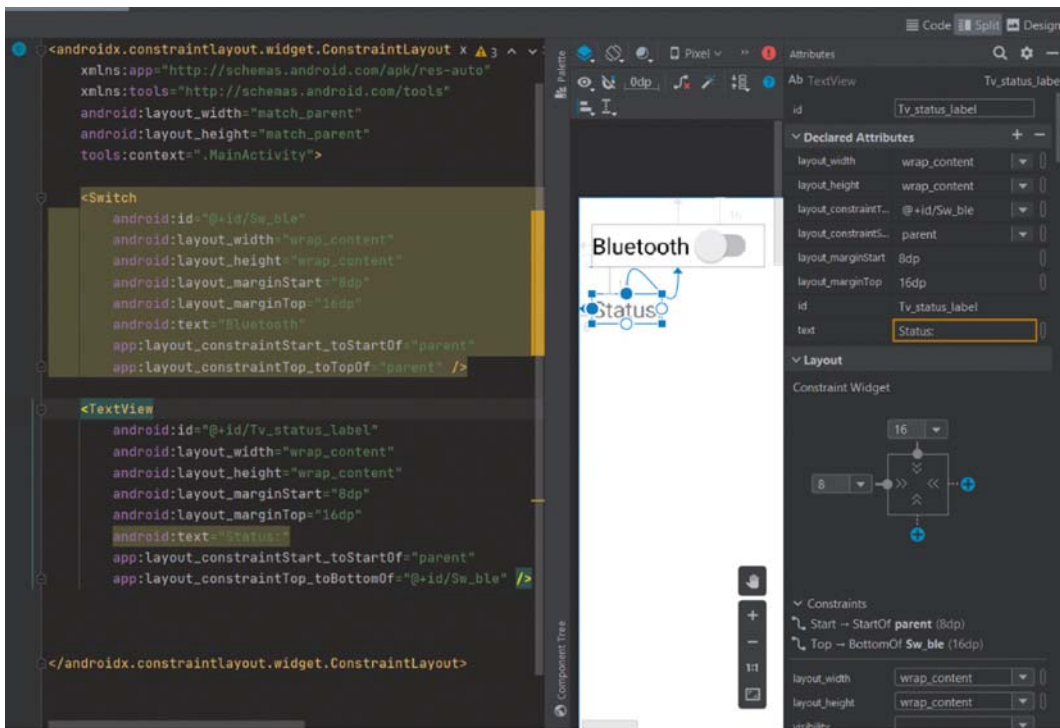


Fig. 11
Vincoli tra
oggetti.

finora definito in modo tale da poter avere un confronto della configurazione sin qui realizzata.

CICLO VITA APPLICAZIONE

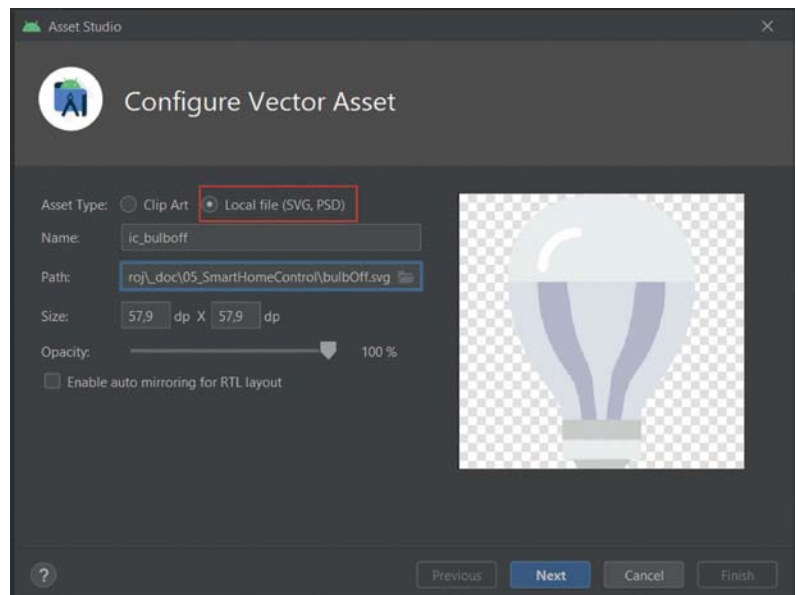
Per quanto semplificato estremamente come discorso, prima di procedere oltre bisogna avere alcune nozioni utili sul ciclo vita di un'applicazione o nello specifico di un'attività (activity) come lo è il *MainActivity* e gli stati che può assumere a seconda delle azioni dell'utente. Partiamo dalla definizione, ovvero il ciclo vita di un'applicazione è un insieme di stati che un'attività può assumere durante il suo ciclo vita; da quando viene creata a quando viene distrutta, il sistema "reclama" le sue risorse e quando un'utente naviga attraverso l'applicazione effettua delle transizioni tra differenti stati del ciclo vita. In **Fig. 13** vengono mostrati gli stati di un'attività all'interno di un'applicazione Android alla quale vengono associate le corrispondenti funzioni:

- la creazione dell'attività che richiama la funzione *onCreate()*;
- l'avvio dell'applicazione richiama la funzione *onStart()*;
- onResume()* viene eseguita con la riesecuzione dell'applicazione o a seguito di una pausa dell'attività;
- la pausa dell'attività è uno stato che intercorre quando l'attività non è più in primo piano (*foreground*) *onPause()*;

- l'attività entra nello stato di stop quando un'altra attività è avviata e quindi la funzione *onStop()*;
- onDestroy()* è una funzione che viene richiamata prima che l'attività venga distrutta o chiusa (ad esempio la rotazione del dispositivo).

È possibile sovrascrivere queste funzioni per eseguire specifiche azioni utente, tuttavia vedremo qualche dettaglio in più durante il proseguo della trattazione.

Fig. 12
Aggiungere
immagini vettoriali.



Listato 1

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/an-
droid"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Switch
        android:id="@+id/Sw_ble"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:text="Bluetooth"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/ TV_status_connection "
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:text="Status:"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/Sw_ble" />

    <TextView
        android:id="@+id/TV_info"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:text="click on image to turn Off or turn On your bulb"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/ TV_status_connection " />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="80dp"
        android:layout_height="100dp"
        android:layout_marginTop="24dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/TV_info"
        app:srcCompat="@drawable/ic_bulboff" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

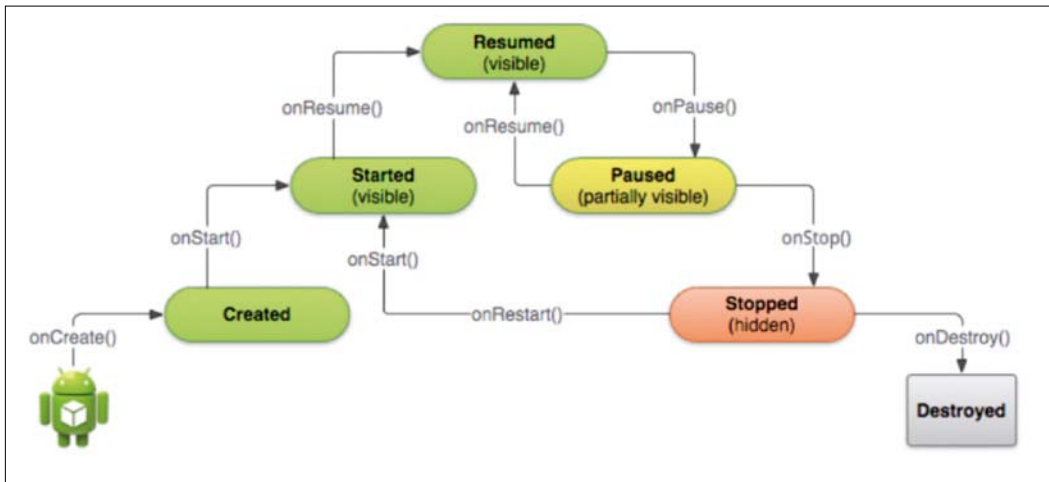
IMPLEMENTAZIONE CODICE

A questo punto, terminata la creazione del layout dell'applicazione, possiamo implementare il codice così da definire le logiche di interazione tra l'utente e applicazione. È importante fare una premessa in quanto non sarà possibile entrare nel dettaglio della descrizione di alcune librerie (come quelle Bluetooth) perché non è lo scopo di questo articolo e quindi ci limiteremo a descrivere lo scopo delle funzioni. L'oggetto delle nostre modifiche è il file *MainActivity.java*, che come suggerisce l'estensione del file viene scritto in linguaggio *Java*. In linea con la descrizione del ciclo vita della nostra attività

andiamo a definire o meglio, a sovrascrivere, le funzioni tramite l'annotazione *@Override*. La funzione *onCreate()* è resa disponibile all'apertura del programma in quanto la sua definizione è una condizione necessaria per il funzionamento dell'applicazione, mentre le altre funzioni si possono anche non richiamare se non utilizzate.

Per il questo progetto, sarà importante gestire la connessione Bluetooth pertanto decidiamo di sovrascrivere le funzioni di ciclo vita, in una nuova riga digitiamo *@Override* e successivamente sovrascriviamo le funzioni come mostrato nella **Fig. 14**. Completata questa fase, dichiariamo gli oggetti





◀ Fig. 13
Ciclo Vita
applicazione
Android.

che abbiamo già definito all'interno del layout come variabili globali così da riutilizzarle all'interno del nostro codice:

```
Switch ble_connection;
TextView status_ble;
ImageView imageBulb;
```

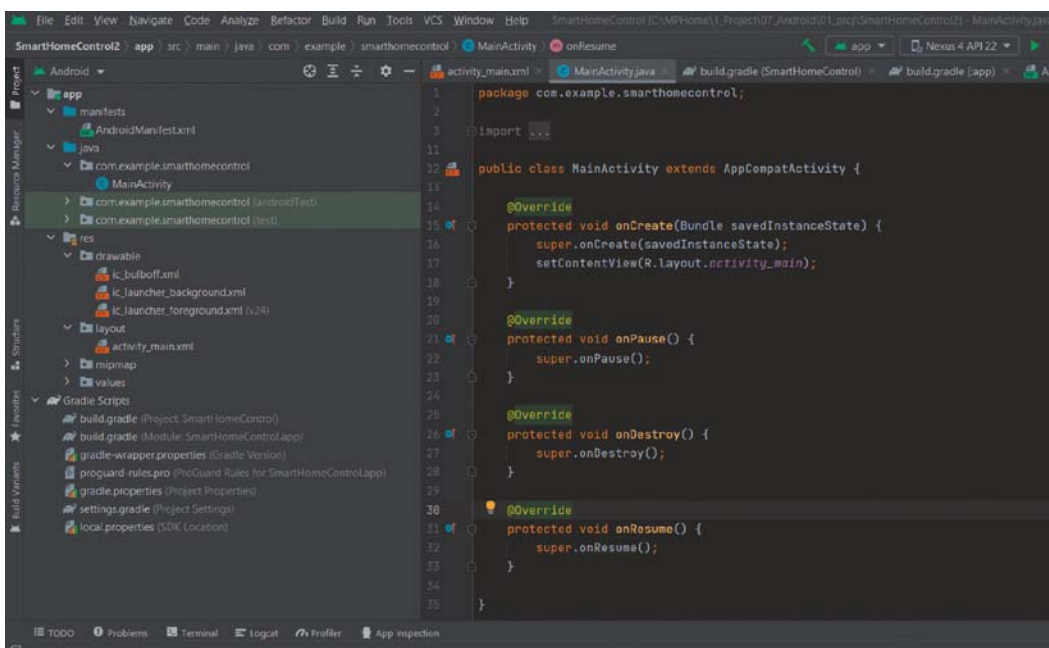
Android mette a disposizione differenti metodi per collegare i vari oggetti al contenuto del layout ed uno di questi è l'utilizzo della funzione *findViewById* che appunto richiama l'oggetto sulla base dell'id definito:

```
status_ble = findViewById(R.id.TV_status_connection);
```

```
ble_connection = findViewById(R.id.Sw_ble);
imageBulb = findViewById(R.id.IV_bulb);
```

Ogni oggetto si distingue in base alle sue proprietà e i metodi che lo sviluppatore può utilizzare per interagire con essi, in questo caso particolare vogliamo consentire il cambio stato dello switch e rendere l'immagine cliccabile e quindi procediamo come segue:

```
ble_connection.setOnCheckedChangeListener(new Com-
poundButton.OnCheckedChangeListener() {
    public void onCheckedChanged(CompoundButton but-
tonView, boolean isChecked) {
        if(isChecked){
```



◀ Fig. 14
Ciclo Vita
applicazione
Android.

```

        }else{
        }
    }
});
imageBulb.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
    }
});

```

Successivamente andremo a definire i contenuti ma prima descriviamo brevemente le librerie che andremo ad utilizzare in base alle funzioni che Android mette a disposizione per il protocollo Bluetooth. Come già descritto in precedenza, il protocollo Bluetooth e nello specifico il Bluetooth Low Energy (BLE) si basa sulla definizione (lato server) di servizi e quindi caratteristiche per la condivisione dati (con il client). Nel nostro progetto l'applicazione Android svolgerà il ruolo di Client pertanto analizziamo le funzioni definite all'interno dei file *BleActivity.java* e *MicrobitGattAttributes.java* che verranno rese disponibili con l'articolo sul sito ufficiale di Elettronica In.

BleActivity contiene tutte le funzioni necessarie per la gestione del servizio Bluetooth e nello specifico:

- *public final BluetoothGattCallback mGattCallback* è una funzione di callback ovvero consente di gestire gli eventi generati dalla richiesta del client
 1. *public void onConnectionStateChange(BluetoothGatt gatt, int status, int newState)* tramite questa funzione è possibile verificare quando lo stato della connessione BLE cambia
 2. *public void onServicesDiscovered(BluetoothGatt gatt, int status)* consente di avere accesso ai servizi trovati
 3. *public void onCharacteristicRead(BluetoothGatt gatt, BluetoothGattCharacteristic characteristic, int status)* permette di eseguire delle azioni a seguito della lettura caratteristica
 4. *public void onCharacteristicWrite(BluetoothGatt gatt, BluetoothGattCharacteristic characteristic, int status)* permette di eseguire delle azioni quando aggiorniamo il valore della caratteristica
 5. *public void onCharacteristicChanged(BluetoothGatt gatt, BluetoothGattCharacteristic characteristic)* permette di eseguire delle azioni quando la caratteristica cambia stato
- *public BluetoothLeScanner getBLEScanner()* restituisce l'oggetto *BluetoothLeScanner* per la ricerca di dispositivi

- *public boolean initialize()* inizializza la periferia Bluetooth
- *public boolean connect(final String address)* consente la connessione ad uno specifico Server Gatt in base all'indirizzo
- *public void disconnect()* se esiste una connessione al Server Gatt, è possibile disconnettersi dal server
- *public void close()* se esiste una connessione al Server Gatt, consente di chiudere la connessione
- *public void readCharacteristic(BluetoothGattCharacteristic characteristic)* permette la lettura dei dati all'interno della caratteristica
- *public boolean writeCharacteristic(BluetoothGattCharacteristic characteristic, String data)* permette di modificare la caratteristica attraverso il dato definito (in formato String)
- *public void setCharacteristicNotification(BluetoothGattCharacteristic characteristic, boolean enabled)* abilita le notifiche per la caratteristica del server Gatt
- *public List<BluetoothGattService> getSupportedGattServices()* restituisce la lista dei servizi presenti all'interno del Gatt Server.

Il file *MicrobitGattAttributes.java* contiene invece i vari UUID delle caratteristiche per le operazioni di lettura, scrittura e notifiche.

Come già anticipato non entreremo nel dettaglio delle funzioni ma saranno comunque disponibili e reperibili con il progetto direttamente nel percorso specifico, ovvero il seguente: ...\\LaTuaCartella\\SmartHomeControl\\app\\src\\main\\java\\com\\example\\smarthomecontrol.

Nell'introduzione ad *Android Studio* abbiamo citato il file *AndroidManifest.java* che consente di definire permessi per accedere ad altri servizi come nel nostro caso il Bluetooth, pertanto accediamo al file e definiamo i seguenti permessi subito prima della parola chiave "<application":

```

<uses-permission android:name="android.permission.
BLUETOOTH" />
<uses-permission android:name="android.permission.
BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.
ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.
ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.
BLUETOOTH_CONNECT" />
<uses-permission android:name="android.permission.
BLUETOOTH_SCAN" />

```



Infine aggiungiamo, prima di `</application>`, anche il permesso per utilizzare il file *BleActivity* come servizio aggiuntivo per l'applicazione:

```
<service android:name="com.example.smarthomecontrol.
BleActivity" android:enabled="true"></service>
```

Questa funzionalità risulta necessaria in quanto vogliamo un'iterazione continua tra l'attività principale (*MainActivity*) e le funzionalità Bluetooth che saranno disponibili nell'attività *BleActivity*.

A questo punto possiamo spostare la nostra attenzione sul *MainActivity.java* dove è necessario descrivere brevemente alcune funzioni che andranno dichiarate in questa fase affinché si possa interagire direttamente con gli elementi aggiunti all'interno del layout:

- *Private void scanLeDevice(final boolean enable)* consente di avviare o interrompere la richiesta di scanning
- *private final ScanCallback leScanCallback*, tale funzione viene richiamata quando nuovi dispositivi (server) segnalano (advertisement) la loro presenza
- *private final ServiceConnection mServiceConnection*, definisce il ciclo vita con il servizio *BleActivity*
- *private final BroadcastReceiver mGattUpdateReceiver*, intercetta le informazioni inviate dal servizio *BleActivity* riguardo la comunicazione Bluetooth
- *private void displayGattServices(List<BluetoothGattService> gattServices)*, rende disponibile il contenuto del server GATT e tutti i servizi disponibili ma anche le corrispondenti caratteristiche
- *private void BlePermissionVerification()*, verifica i permessi per accedere alla periferica Bluetooth e se necessario invio una richiesta all'utente
- *private static IntentFilter makeGattUpdateIntentFilter()*, definisce le possibili interazioni (intent) tra l'applicazione e periferica BLE.

La descrizione di queste funzioni è propedeutica per quello che sarà il contenuto del nostro programma che sarà necessario importare all'interno del *MainActivity* per garantire una corretta compilazione del codice.

Dichiariamo all'interno del file le seguenti variabili:

```
BleActivity mBluetoothActivity;
private boolean bleConnSts = false;
BluetoothLeScanner bluetoothLeScanner;
ArrayList<BluetoothDevice> mLeDevices;
boolean bulbStsOn = false;
```

```
String deviceAddress = "NA";
```

- L'oggetto *mBluetoothActivity* verrà utilizzato per accedere al servizio *BleActivity*;
- *bleConnSts* verrà utilizzata per avere informazioni sullo stato della connessione Bluetooth, se il valore sarà *true* allora il dispositivo sarà connesso;
- *bluetoothLeScanner* consente di istanziare l'operazione di scanning;
- *mLeDevices* contiene tutti i dispositivi riconosciuti dall'operazione di scanning;
- *bulbStsOn* è una variabile che utilizzeremo per gestire la pressione dell'immagine (*ImageView*) *imageBulb*;
- *deviceAddress* è l'indirizzo del dispositivo al quale l'applicazione si conatterà.

All'interno della funzione *onCreate()* andiamo a definire:

```
BlePermissionVerification();
Intent gattServiceIntent = new Intent(this, BleActivity.class);
getApplicationContext().
bindService(gattServiceIntent, mServiceConnection,
getApplicationContext().BIND_AUTO_CREATE);
```

Riguardo la prima funzione è già stata descritta in precedenza, mentre le successive consentono di richiamare e connettere l'applicazione al servizio *BleActivity*.

Definiamo il contenuto del metodo per la gestione dello switch:

```
ble_connection.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if(isChecked){
            if(mBluetoothActivity!=null){
                mLeDevices.clear();
                bluetoothLeScanner = mBluetoothActivity.getBleScanner();
                scanLeDevice(true);
            }
        }else{
            if(mBluetoothActivity!=null){
                if(bleConnSts)mBluetoothActivity.disconnect();
            }
        }
    }
});
```



Nel caso di switch attivo allora sarà necessario verificare che il servizio *BleActivity* è stato creato correttamente tramite il controllo *mBluetoothActivity!=null*, cancellare il contenuto della lista di dispositivi riconosciuti nel caso di procedure di scanning precedenti con *mLeDevices.clear()* e quindi avviare la procedura di ricerca con la funzione *scanLeDevice(true)*. In caso contrario l'applicazione verrà disconnessa dal dispositivo con la funzione *mBluetoothActivity.disconnect()*. Quindi andiamo ad agire sull'immagine e come già effettuato in precedenza andiamo ad aggiungere la seconda immagine *ic_bulbon* ed il rispettivo codice:

```
imageBulb.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
if(bulbStsOn){
if(bleConnSts){
imageBulb.setImageResource(R.drawable.ic_bulboff);
mBluetoothActivity.writeCharacteristic(myGattCharacteristic, "#B0;");
bulbStsOn =false;
}
}else{
if(bleConnSts) {
imageBulb.setImageResource(R.drawable.ic_bulbon);
mBluetoothActivity.writeCharacteristic(myGattCharacteristic, "#B1;");
bulbStsOn =true;
}
}
}
});
```

Alla pressione dell'immagine dobbiamo verificare lo stato della stessa tramite la variabile *bulbStsOn* ma anche lo stato della connessione bluetooth con la variabile *bleConnSts* e nel caso di immagine con lampadina spenta andiamo ad inviare il comando via bluetooth con la scrittura del dato della caratteristica *mBluetoothActivity.writeCharacteristic(myGattCharacteristic, "#B1;")*, aggiorniamo lo stato della lampadina con *imageBulb.setImageResource(R.drawable.ic_bulbon)* ed infine lo stato della variabile *bulbStsOn*. A questo punto non rimane altro che gestire il ciclo vita dell'applicazione:

```
@Override
protected void onPause() {
super.onPause();
unregisterReceiver(mGattUpdateReceiver);
```

```
Log.d(TAG, "onPause");
}
```

La funzione *unregisterReceiver(mGattUpdateReceiver)* svincola l'applicazione dal servizio *BleActivity*.

```
@Override
protected void onResume() {
super.onResume();
registerReceiver(mGattUpdateReceiver, makeGattUpdateIntentFilter());
if (mBluetoothActivity != null) {
if (deviceAddress != "NA"){
final boolean result = mBluetoothActivity.
connect(deviceAddress);
Log.d(TAG, "Connect request result=" + result);
}
}
Log.d(TAG, "onResume");
}
```

Il contenuto della funzione *OnResume()*, al contrario della precedente, ristabilisce un collegamento con il servizio *BleActivity* attraverso la funzione *registerReceiver(mGattUpdateReceiver, makeGattUpdateIntentFilter())* e se il dispositivo è già stato trovato (*deviceAddress != "NA"*) ristabilisce la connessione con *mBluetoothActivity.connect(deviceAddress)*. Nel caso di applicazione chiusa permanentemente (*OnDestroy()*) andiamo a disconnettere e chiudere la connessione Bluetooth ma anche a disconnettere il servizio *BleActivity* dall'applicazione tramite la funzione *unbindService(mServiceConnection)*;

```
@Override
protected void onDestroy() {
super.onDestroy();
mBluetoothActivity.disconnect();
mBluetoothActivity.close();
unbindService(mServiceConnection);
Log.d(TAG, "onDestroy");
}
```

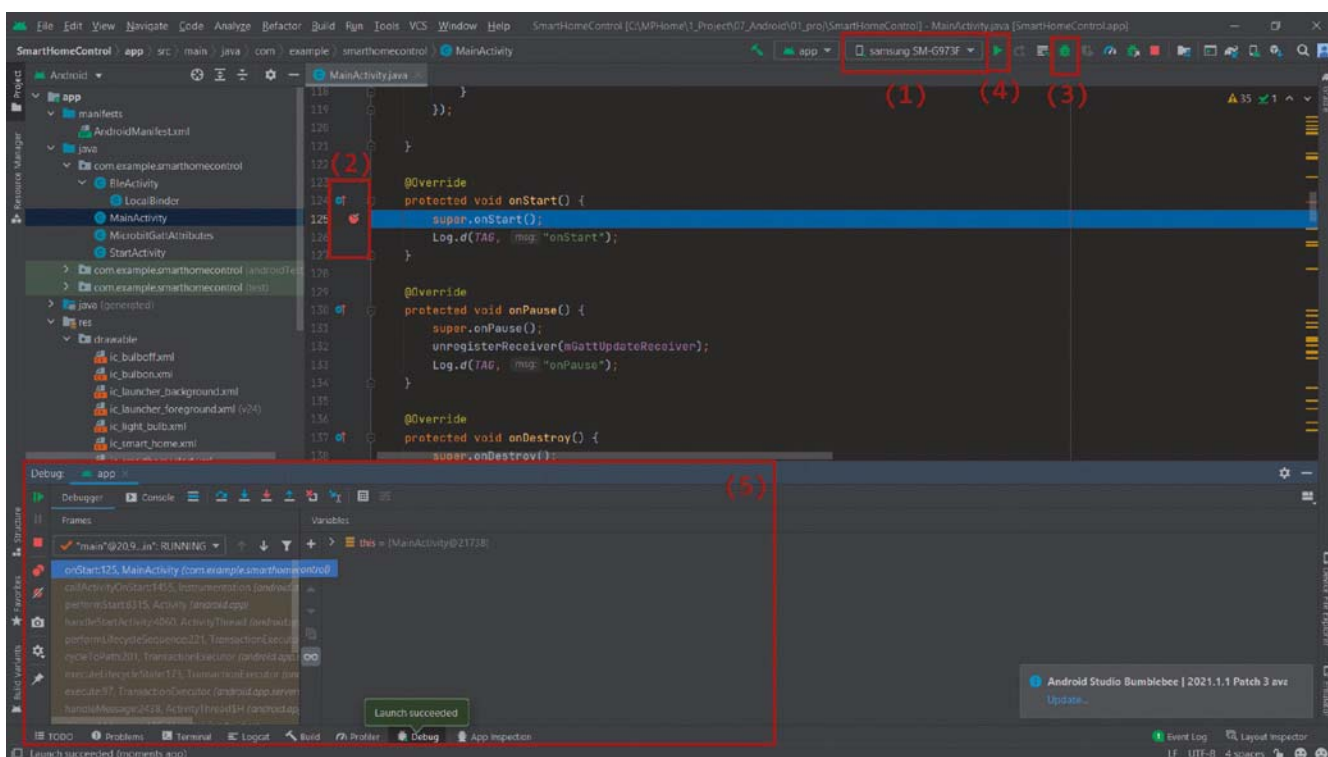
Infine per modificare lo stile dell'applicazione in linea con la **Fig. 9** è necessario:

- aggiungere i colori d'interesse nella sezione *res – values – color.xml*
- modificare lo stile dell'applicazione tramite il file *res – values – themes – themes.xml* utilizzando i nuovi colori definiti

DEBUG E VERIFICA APPLICAZIONE

Adesso che il nostro programma è completo





possiamo procedere con la verifica e il debug; il primo passo da seguire è il collegamento usb dello smartphone al PC con l'accortezza di abilitare il debug USB dalle impostazioni dello stesso smartphone. A questo punto il dispositivo verrà riconosciuto come mostrato in **Fig. 15** punto (1). L'ambiente *Android Studio* permette di verificare il codice tramite la modalità di debug al quale è possibile accedere aggiungendo un pallino rosso sulla riga di codice dove si vuole effettuare la verifica (2) e successivamente premere sul tasto di debug (3) con conseguente risultato mostrato nella stessa **Fig. 15** punto (5). In alternativa è possibile anche non collegare lo smartphone ed utilizzare Android Emulator che consente di simulare l'applicazione realizzata ma che sconsigliamo per questo progetto in quanto la periferica Bluetooth non può essere simulata. Se non necessaria la sessione di debug, si può procedere direttamente con l'esecuzione dell'applicazione tramite in tasto (4), che consentirà di installare ed eseguire automaticamente l'applicazione sullo smartphone. Bene, arrivati a questo punto possiamo procedere con la verifica della comunicazione Bluetooth tramite il dispositivo di *SmartHomeControl* realizzato con Microbit di cui abbiamo ampiamente discusso nell'articolo precedente sul numero 264 di *Elettronica In*.

CONCLUSIONI

Arriviamo a termine di questo articolo aggiungendo un secondo passo verso la realizzazione di un sistema di domotica che ha come cuore la scheda Microbit. Sin qui abbiamo analizzato in dettaglio come implementare un'applicazione con Android Studio che ci permetta di centralizzare il controllo del sistema d'illuminazione, al quale si aggiungeranno nuove funzionalità nei prossimi articoli, per arrivare ad un sistema completo per la gestione di svariate funzionalità all'interno della propria abitazione. ■

Fig. 15
Verifica
Applicazione.



Cosa occorre?

La board micro:bit V2 è disponibile con lo starterkit (cod. MICROBITQSKIT) 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

HDMI
↓
VGA

€ 11,50

Cod. **BOXHDMITOVGA**

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.

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.

Convertitori di segnale

Permettono di convertire un segnale digitale HDMI in VGA, AV in HDMI, VGA in HDMI, HDMI in AV.

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**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.



AMPLIFICATORI AUDIO

di **GABRIELE PERETTO**

Con prove di laboratorio andiamo a verificare i principali parametri di funzionamento di alcuni diffusi e pratici amplificatori audio in classe AB e D. Seconda puntata.

Dopo aver affrontato, nella prima puntata di questo articolo, la parte tecnica e teorica, quale i parametri, le classi di funzionamento e le tipologie di misurazione che vengono normalmente effettuate per la caratterizzazione degli amplificatori audio, in questa seconda parte, andremo a verificare con prove di laboratorio, i principali parametri di funzionamento di alcuni moduli commerciali, che come sempre possiamo reperire presso Futura Elettronica. Tratteremo in particolare i piccoli moduli con potenze fino a 3W massimi per canale e che sono molto utili per applicazioni, dove sono richieste modeste potenze in spazi ridotti e/o con basse tensioni di alimentazione, quali possono risultare anche una semplice batteria al litio da 3.6V oppure una presa USB (5V). Questi moduli possono inoltre essere facilmente interfacciati con

schede Arduino o Raspberry Pi risultando quindi alquanto versatili. Per ogni modulo che andremo a proporre, abbiamo raccolto i dati più significativi all'interno di una tabella. 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
- $V_{pp In}$ = Tensione di ingresso in Volt pk/pk
- V_{out_rms} = Tensione di uscita sotto carico
- $PO_{_8\Omega} (W)$ = 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.

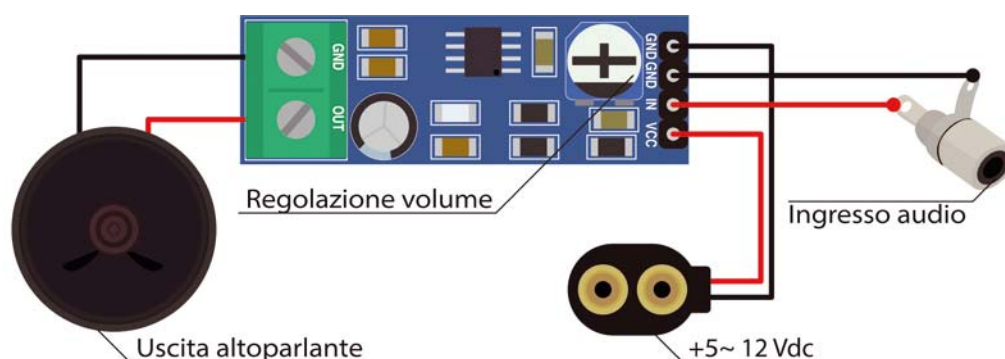
Poiché come descritto nella prima puntata di questo tutorial, normalmente l'udibilità umana non percepisce distorsioni superiori al 5% a patto che la terza armonica non superi il 2%, abbiamo volutamente valutato dati all'intorno di questi parametri. In alcuni casi abbiamo volutamente ecceduto per mostrare come si presenta la forma d'onda e lo spettro armonico in presenza di elevata distorsione. Le immagini che andremo ad allegare alla spiegazione serviranno a fare chiarezza su molti aspetti, spesso poco conosciuti. Facciamo comunque una dovuta precisazione. Tutte le prove sono state fatte (come da manuale) su carichi composti da resistori di potenza e quindi hanno valenza puramente tecnica. La valutazione della dinamica, dovrebbe essere fatta con un carico induttivo quale è l'altoparlante per la riproduzione, ma poiché ognuno di essi dispone di caratteristiche proprie e uniche, questo in laboratorio non è possibile farlo. Le indicazioni che qui andiamo ad illustrare, dovrebbero essere sufficienti affinché

ognuno poi possa fare le proprie prove personalizzate e finalizzate al proprio utilizzo, con i propri altoparlanti e trarne le dovute considerazioni.

STRUMENTAZIONE

Gli strumenti necessari per le varie verifiche, li abbiamo elencati nella prima parte dell'articolo e li abbiamo utilizzati quasi tutti, ma in particolare abbiamo voluto sperimentare la versatilità di un paio di programmi open source e che ci hanno dato ottimi riscontri su tutta la banda audio tra 20 e 20000 Hz, anche se al momento a causa della mancata disponibilità di mercato di schede audio con banda passante superiore a basso prezzo, non è possibile analizzare segnali oltre i 20 KHz, benché l'applicazione permetterebbe di arrivare fino a 192 kHz. Questo comunque non è un grosso intoppo, poiché le frequenze oltre 20 KHz non sono udibili e pertanto sono trascurabili, almeno per i nostri fini. Questi programmi (DAA digital audio analyzer) e (AA audio analyzer) hanno numerosi pregi, quali quello di disporre di analisi armonica, di poter utilizzare funzioni matematiche, come la sottrazione dei segnali (per non utilizzare sonde differenziali) e inoltre di disporre di filtri digitali di ingresso programmabili (DAA). Tutto questo con la modica spesa di pochi euro per l'acquisto della sola scheda audio di ingresso USB da collegare al PC, poiché i software sono freeware e open source; inoltre non richiedono installazione, perché sono dei cosiddetti "Programmi Portatili", semplici eseguibili da lanciare. Per disinstallarli, basta cancellare la cartella. Per chi fosse interessato ad approfondire l'argomento e a utilizzarli invitiamo a collegarsi al sito: www.thereмино.com/downloads/uncategorized e all'interno cercare l'applicazione: DDA (Digital Audio Analyzer) e/o AA (Audio Analyzer) seguendo poi l'articolo e le spiegazioni dettagliate per l'utilizzo di questi ottimi e versatili strumenti.

Fig. 1
Modulo
amplificatore
MONO con LM386
HR02147.



Parametri	Condizioni di test	Min	tip	max	Unità di misura
Vs tensione di alimentazione	LM386	4		12	V
Impedenza altoparlante	LM386	4		32	Ω
Guadagno Gv /Av	LM386	26		46	dB
Potenza di uscita	9Vs - 8 Ω - THD 10%	500		700	mW
Distorsione THD	6V - 8 Ω - 125mW - 1KHz		0.2		%

Tabella 1
Dati del costruttore relativi al C.I. LM 386.

V_ALIM	P (W)	Vpp In (*)	Vout_rms	Po_8 Ω (W)	Av (dB)	THD %	THD % 3a	rend. %
9	0,56	0,11	1	0,125	28,2	0,46	0,1	22
12	1,8	0,31	2,5	0,781	27,2	3,59	1,9	43
V_ALIM	P (W)	Vpp In (*)	Vout_rms	Wrms_4 Ω	Av (dB)	THD %	THD % 3a	rend. %
9	1,1	0,15	1	0,250	25,5	2,35	0,71	23
6	0,8	0,29	1,15	0,331	21,0	6,63	0,6	41

Tabella 2
Parametri rilevati sul modulo HR02147.

MODULO CON LM386

Il primo modulo (**Fig. 1**) che andiamo a trattare è basato su un chip molto diffuso prodotto da Texas Instruments da quasi 20 anni, cioè il noto LM386. La versatilità di questo piccolo amplificatore monolitico (disponibile sul sito www.futuranet.it con il codice HR02147) è ben nota e il componente può essere utilizzato anche come oscillatore a ponte di Wien a bassa distorsione, nonché come generatore di onde quadre e per amplificare il segnale BF proveniente di piccoli moduli ricevitori AM e FM. Risulta altresì possibile adattare il circuito come amplificatore "BASS BOOST" di piccola potenza, in modo da ottenere un piccolo amplificatore con buona riproducibilità delle frequenze basse (subwoofer amp) con frequenza di taglio intorno ai 100 Hz. In questo caso otteniamo un guadagno di circa 20 dB a 20Hz, per salire fino a 25 dB a 100 Hz, 21 dB a 500 Hz, scendendo poi a 20 dB tra 5 e 20 KHz.

Se volessimo utilizzare due moduli, possiamo quindi realizzare un piccolo ma efficiente sistema a 2 vie con subwoofer. Sul data sheet di TI possiamo trovare tutti i possibili schemi applicativi relativi a quanto descritto. In **Fig. 2** è rappresentato lo schema classico di utilizzo come amplificatore BF, per potenze fino a 250 mW e che massimamente corrisponde al circuito del modulo rappresentato. Nella **Tabella 1**, possiamo vedere i principali dati dichiarati dal costruttore. Il circuito che abbiamo testato è a singola uscita (MONO) e dispone di trimmer per la regolazione della sensibilità del segnale di ingresso. Con la regolazione alla massima sensibilità, ovvero massimo volume, possiamo già ottenere 125 mW su 8 Ω , con soli 100 mV di segnale di ingresso. Nella **Tabella 2**, sono riportati i dati rilevati dalle nostre prove di laboratorio.

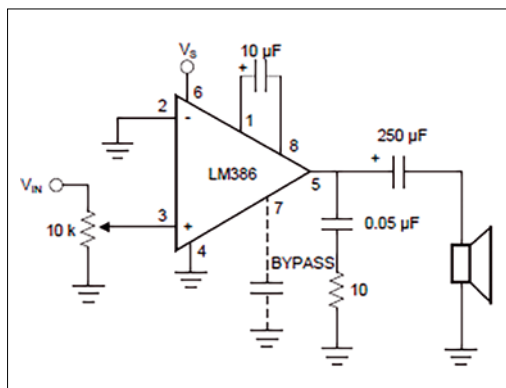


Fig. 2
Schema di utilizzo classico per LM386.

Fig. 3
Curva di risposta tipica della banda passante del modulo HR02147.

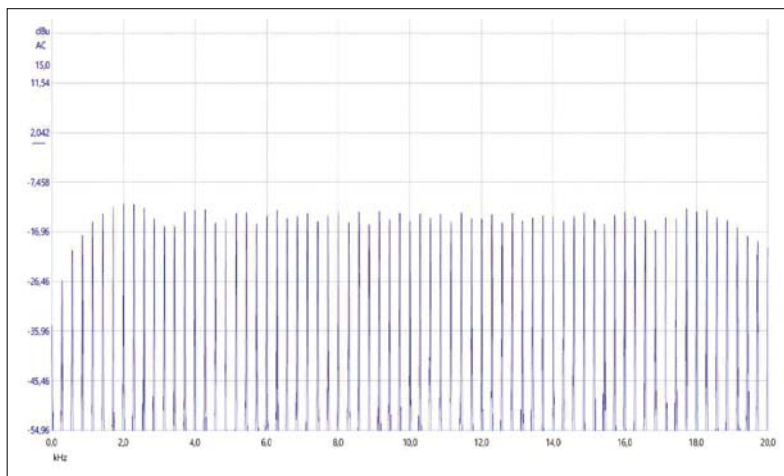
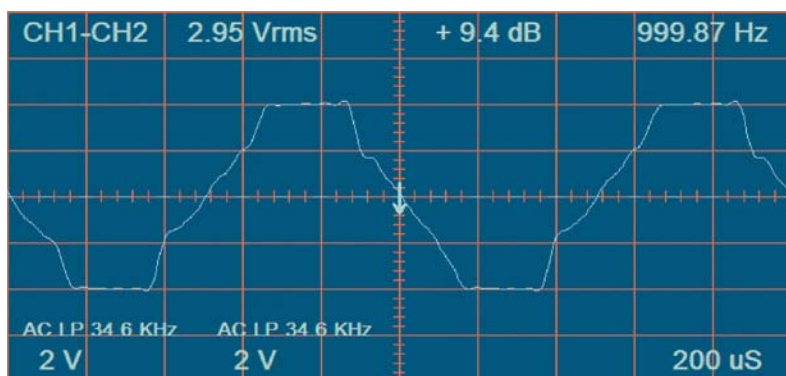


Fig. 4
Modulo PAM8302A.

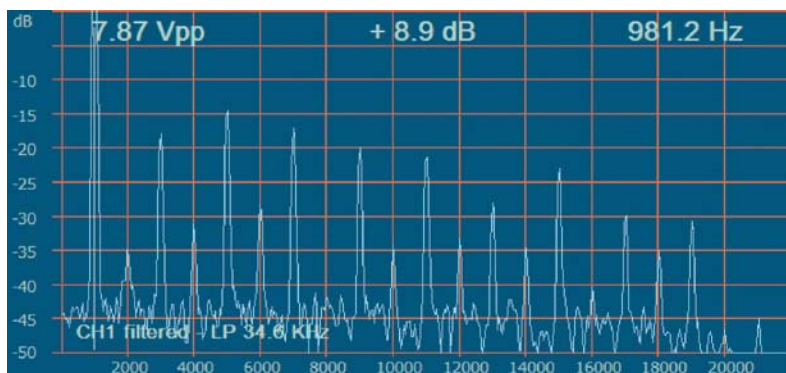
→ **Tabella 3**
Dati del
costruttore relativi
al C.I. PAM8302.

Parametri	Condizioni di test	Min	tip	max	Unità di misura
Vs tensione di alimentazione	PAM 8302	2		5.5	V
Impedenza altoparlante	PAM 8302	4		8	Ω
Guadagno Gv /Av	PAM 8302		23.5		dB
Potenza di uscita	5Vs - 8 Ω - THD 10%	1.35	1.5		W
	5Vs - 4 Ω - THD10%	2.25	2.5		
Distorsione THD	5V - 4 Ω - 0.5W - 1 KHz		0.4	0.45	%



↑ **Fig. 5**
Distorsione
7.9% con
uscita 2.24 W
a 5V su 4
Ohm relativa
al modulo
PAM8302A.

↓ **Fig. 6**
Spettro
armonico
della
distorsione
armonica
al 7.9% del
modulo
PAM8302A.



Per eventuali ulteriori dettagli circa questa configurazione, si può fare riferimento al data sheet del costruttore.

MODULO PAM8302

Il secondo modulo che abbiamo inserito nelle nostre prove (**Fig. 4**), è basato su chip PAM8302 prodotto da DIODES Inc e distribuito da Futura Elettronica con il codice PAM8302A ed è un circuito monolitico integrato, funzionante in classe D; trattasi di un modulo a singola uscita (mono) e alcuni dati salienti del costruttore sono stati da noi riportati nella **Tabella 3**. Questo piccolo amplificatore che misura soltanto 15,3x24,5x3 mm e come abbiamo anzidetto, funzionando in architettura classe D, presenta una buona efficienza che abbiamo rilevato fino a circa il 75% in condizioni ottimali e cioè con alimentazione a 5V su carico da 8 Ω dove si riescono ad ottenere circa 1,1W rms con una distorsione dello 0,7 %. Il massimo che siamo comunque riusciti ad ottenere in condizioni ottimali, si attesta intorno a 2,2W su 4 Ω , sempre con alimentazione a 5V ma con una distorsione che raggiunge quasi l'8% e con distorsione di 3° armonica pari a 4,5%. La **Fig. 5** evidenzia l'elevata deformazione dell'onda sinusoidale in uscita alla frequenza di 1KHz. In **Fig. 6** troviamo invece rappresentato lo spettro armonico alle stesse condizioni, dove si può notare l'elevato contenuto di armoniche soprattutto dispari e di ampio livello su tutta la gamma fino a 20 KHz. Come abbiamo menzionato nel capitolo 1° le armoniche dispari purtroppo sono quelle che distorcono significativamente il segnale in modo "fastidioso" quando presenti a livelli elevati, come riscontrabile in questo caso. In **Tabella 4**, abbiamo riportato i dati rilevati e relativi a due differenti tensioni di alimentazione; ovvero 3.6V, tipico delle batterie al litio a tensione nominale e 5V, quale potrebbe essere la tensione prelevata da una porta USB di un PC o da un alimentatore tipo carica batterie per smartphone. Rispettivamente con uscite a 8 Ω e 4 Ω . Da notare che sul modulo è presente un trimmer per la regolazione del livello del segnale in ingresso.

La banda passante risulta abbastanza buona e discretamente lineare da 300 Hz fino a circa 20 KHz. Sotto i 300 Hz, abbiamo una progressiva caduta ma contenuta in massimo 2 - 3 dB fino a 20 Hz. Sostanzialmente ci troviamo di fronte ad un buon modulo per applicazioni che non richiedono potenze superiori a 1÷1,5 watt, con una buona fedeltà di riproduzione su tutta la banda audio e molto versatile, viste le ridotte dimensioni.

MODULO PAM8403

Questo piccolissimo modulo (**Fig.7**) che misura appena 18x21x3.2 mm, si basa su un chip monolitico anch'esso prodotto dalla DIODES Inc. e commercializzato con la sigla MICROAMPLID2X3W. Pur essendo un chip datato, risulta ancora in produzione o quantomeno esistono ancora stock di magazzino che lo rende ancora largamente utilizzato, anche se è ragionevole pensare che in futuro sarà del tutto rimpiazzato. Al di là di queste considerazioni, il nostro modulo, anche se di dimensioni ridottissime, è un amplificatore ad alta efficienza in classe D, tipo stereo (2 canali). Come prima abbiamo riportato i principali dati forniti dal costruttore, nella **Tabella 5**. In **Fig. 8** è rappresentato lo schema di utilizzo tipico tratto dal data sheet, dove si può notare il ristretto numero di componenti passivi esterni necessari che si riduce a 7 condensatori e appena due resistori. Questo denota la complessa architettura interna voluta dal costruttore al fine di semplificare le realizzazioni, miniaturizzarle e renderle quindi molto versatili alle più svariate applicazioni, anche in

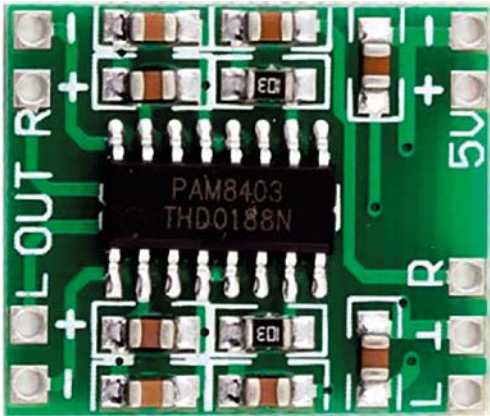


Fig. 7
Modulo
MICROAM-
PLID2X3W

presenza di spazio limitato.
Le varie misurazioni effettuate, sono riportate in **Tabella 6**, dove possiamo notare che in questo

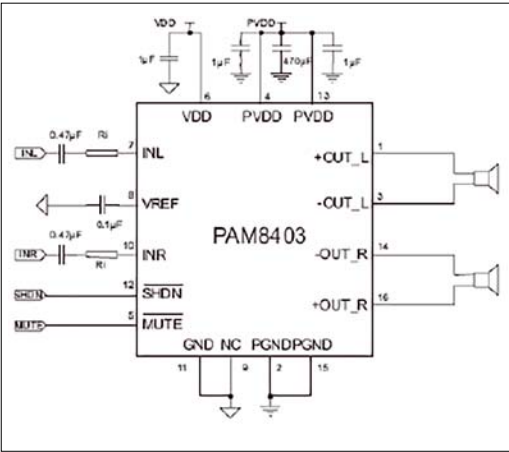


Fig. 8
Schema
tipico
applicativo
relativo
al chip
PAM8403.

V_ALIM	P (W)	Vpp In	Vout_rms	Po_8Ω (W)	Av (dB)	THD %	THD % 3a	rend. %
5	1,5	0,61	3	1,1	22,9	0,7	0,44	75
5	1,16	0,5	2,49	0,8	23,0	0,55	0,33	67
3,6	0,69	0,4	1,99	0,5	23,0	0,6	0,52	72
V_ALIM	P (W)	Vpp In (*)		Wrms_ 4Ω	Av (dB)	THD %	THD % 3a	rend. %
5	1,14	0,36	1,42	0,5	21,0	3,39	1,84	44
5	3,1	0,74	2,99	2,24	21,2	7,9	4,52	72
3,6	1,2	0,4	1,81	0,8	22,1	2,44	1,02	68

Tabella 4
Parametri
rilevati sul
modulo
MICROAM-
PLID2X3W.

Parametri	Condizioni di test	Min	tip	max	Unità di misura
Vs tensione di alimentazione	PAM 8403	2		5.5	V
Impedenza altoparlante	PAM 8403	4		8	Ω
Guadagno Gv /Av	PAM 8403		24		dB
Potenza di uscita	5Vs - 8Ω - THD 10%		1.8		W
	5Vs - 4Ω - THD10%		3.2		
Distorsione THD	5V - 8 Ω - 1W – 1KHz		0.15		%

Tabella 5
Dati del
costruttore relativi
al C.I. PAM8403.

Tabella 6
Parametri rilevati
sul modulo
MICROAMPLID2X3W

V_ALIM	P (W)	Vpp In	Vout_rms	Po_8Ω (W)	Av (dB)	THD %	THD % 3a	rend. %
3,6	1,66	0,3	2,35	0,69	26,9	0,43	0,17	83
5	3,2	0,42	3,3	1,36	26,9	0,41	0,17	85
5	3,45	0,45	3,45	1,49	26,7	2,11	1,82	86
V_ALIM	P (W)	Vpp In	Vout_rms	Wrms_4Ω	Av (dB)	THD	THD % 3a	rend. %
3,6	3,02	0,3	2,18	1,19	26,3	0,71	0,18	79
5	5,9	0,39	3,07	2,36	27,0	0,42	0,2	80
5	6,4	0,43	3,23	2,61	26,5	2,2	1,8	82

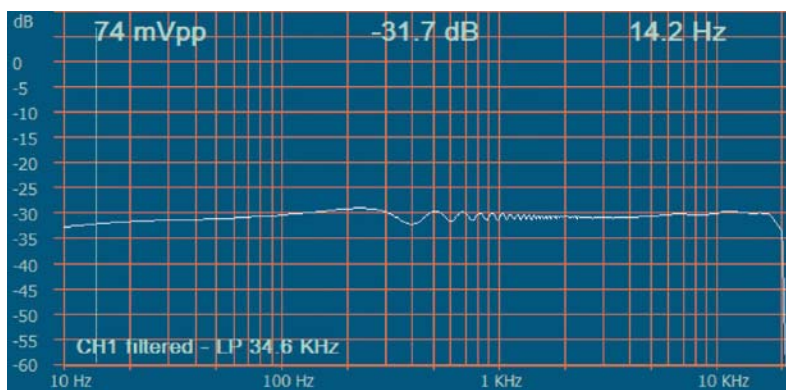
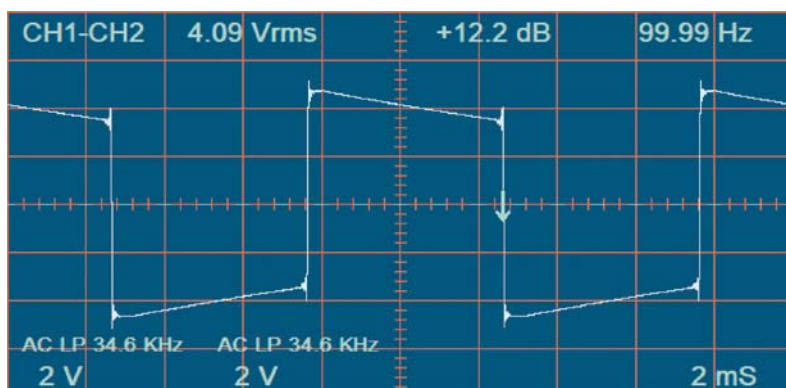


Fig. 9
Curva di risposta
della banda passante
relativa al modulo
MICROAMPLID2X3W.

caso siamo di fronte ad un modulo in grado di erogare fino a 2,6W per canale, con una distorsione intorno al 2,2% e 3^a armonica a 1,8%. L'efficienza risulta molto buona raggiungendo valori intorno all' 82% con alimentazione a 5V e su carico da 4Ω. Anche la banda passante di questo modulo (**Fig. 9**) è eccellente e risulta molto lineare su tutta la gamma di frequenze da 20 a 20.000 Hz, con una sola attenuazione di circa 2÷3 dB sotto 250 Hz. Nella (**Fig.10**), abbiamo riportato la forma dell'onda quadra ma a 100 Hz (della quale abbiamo trattato nello specifico nel precedente articolo), dove si può notare la leggera attenuazione dei bassi, ma considerando che con questa

Fig. 10
Onda quadra a 100
HZ relativa alla
BW del modulo
MICROAMPLID2X3W.



analisi scendiamo ad analizzare fino a 5 Hz, si può immediatamente capire quanto sia lineare l'uscita anche a frequenze molto basse. In conclusione possiamo asserire che questo modulo è ottimo per tutte le applicazioni con potenze fino a 2W e tensioni d'alimentazione fino a 5Vcc.

MODULO PAM8403 CON REGOLAZIONE VOLUME

Questo modulo amplificatore (**Fig. 11**) è sostanzialmente la versione precedente che utilizza sempre un "core" PAM8403 e cioè un chip in contenitore SOIC 16 ma, in questo caso con implementato un interruttore di alimentazione e relativo potenziometro doppio, integrato per la regolazione del volume dei due canali (cod. PAM8403POT).

Questa configurazione permette maggior flessibilità di utilizzo, poiché non dobbiamo più preoccuparci di "dosare" il segnale di ingresso che viene gestito dal doppio potenziometro.

In **Fig. 12** è rappresentato lo schema elettrico completo di tale circuito. Non riporteremo qui i dati del costruttore relativamente al PAM8403, poiché uguali al precedente e quindi già riportati in **Tabella 5**.

Abbiamo comunque nuovamente controllato e misurato tutto per capire se il progetto rispettava

Fig. 11
Modulo PAM 8403 con regolatore di volume
PAM8403POT.



V_ALIM	P (W)	Vpp In	Vout_rms	Po_8Ω (W)	Av (dB)	THD %	THD % 3a	rend. %
3,6	1,73	0,46	2,27	0,6	22,9	0,32	0,14	74
5	3,36	0,64	3,18	1,3	23,0	0,32	0,14	75
5	3,5	0,67	3,3	1,4	22,9	1,57	0,77	78
V_ALIM	P (W)	Vpp In	Vout_rms	Wrms_4Ω	Av (dB)	THD	THD % 3a	rend. %
3,6	3,03	0,42	2,03	1,0	22,7	0,34	0,16	68
5	5,94	0,59	2,86	2,0	22,7	0,39	0,2	69
5	7	0,71	3,2	2,6	22,1	6,42	6,24	73

Tabella 7
Parametri
rilevati sul
modulo
PAM8403POT

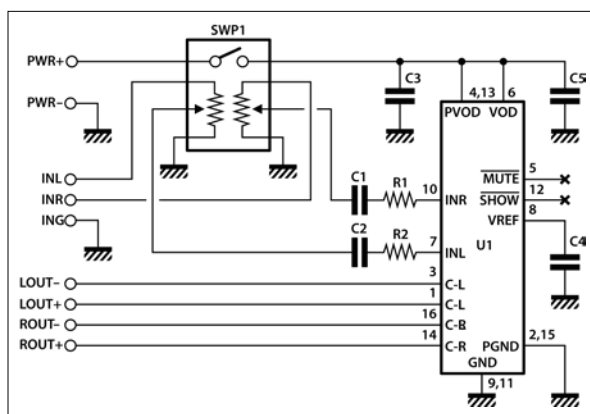


Fig. 12
Schema elettrico del circuito PAM8403
con potenziometro volume e interruttore di
alimentazione.

massimamente quanto rilevato in precedenza e in **Tabella 7**, sono quindi elencati tutti i valori delle misurazioni effettuate e calcolate. Anche in questo caso i test sono stati effettuati con due differenti alimentazioni, rispettivamente a 3,6V e 5V. Da segnalare che in questo caso abbiamo trovato il guadagno (circa 23 dB), più allineato alle specifiche e quindi il circuito meno "tirato", ma con un rendimento leggermente inferiore. Comunque con alimentazione a 5V, si riescono ad ottenere tranquillamente 2+2 Watt su 4Ω, con una distorsione dello 0,39% ed un'efficienza del 69%. In **Fig. 13** abbiamo riportato la forma d'onda fortemente "clippata" alle condizioni di 2,6 + 2,6 W su 4Ω, on distorsione del 6,4%, mentre in **Fig. 14**, possiamo vedere la distorsione armonica alle stesse condizioni dove si evidenzia in modo netto cosa succede quando si entra in questa situazione. In pratica la potenza non sale più, ma la distorsione cresce esponenzialmente. Possiamo quindi concludere che anche questo moduli, di dimensioni comunque ridotte, risulta

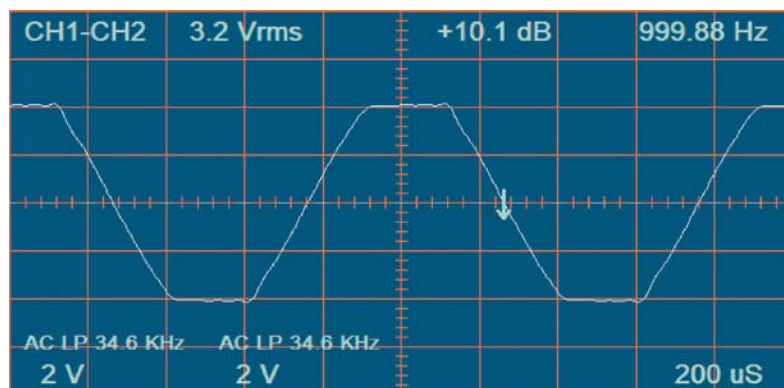


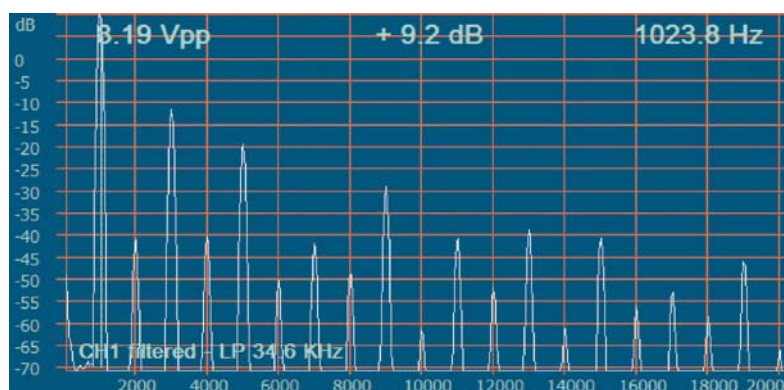
Fig. 13
Forma d'onda
fortemente
"clippata" ovvero
a distorsione 6.4%
a 2.6+2.6W output
relativa al modulo
PAM8403POT.

valido e con una riproduzione molto fedele e lineare per applicazioni fino ad un massimo di 2 + 2,5 Watt per canale. Quanto dichiarato dal costruttore è comunque reale e si riescono effettivamente ad ottenere in uscita 3+3 Wrms con una distorsione del 10%, ma va detto che la distorsione è data tutta dalla 3° armonica, per cui il segnale risulta molto distorto. Questo vale per tutti i moduli che utilizzano il PAM8403.

MODULO MM209

Ci occupiamo ora di un modulo prodotto di Vel-leman e distribuito sempre da Futura Elettronica

Fig. 14
Spettro di
distorsione
armonica al
6.4% del modulo
PAM8403POT.



→ **Tabella 8**
Dati del costruttore
relativi al modulo
MM209.

Parametri	Condizioni di test	Min	tip	max	Unità di misura
Vs tensione di alimentazione	MM 209	2.5		5	V
Impedenza altoparlante	MM 209	4		8	Ω
Guadagno Gv /Av	MM 209		18 adj.		dB
Potenza di uscita	5Vs - 8 Ω - THD N.D. 5Vs - 4 Ω - THD N.D.		1.6 2.4		W
Distorsione THD	5V - 8 Ω - 1W - 1KHz		0.02		%

→ **Tabella 9**
Parametri rilevati
sul modulo
MM209.

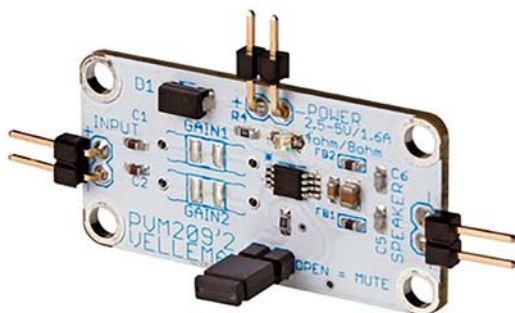
V_ALIM	P (W)	Vpp In	Vout_rms	Po_ 8 Ω (W)	Av (dB)	THD %	THD % 3a	rend. %
3,6	0,6	0,6	1,67	0,3	17,9	0,02	0,02	58
5	1,3	0,92	2,56	0,8	17,9	0,25	0,14	63
5	1,42	1	2,71	0,9	17,7	2,9	2,11	65
V_ALIM	P (W)	Vpp In	Vout_rms	Wrms_ 4 Ω	Av (dB)	THD	THD % 3a	rend. %
3,6	1,44	0,53	1,44	0,5	17,7	0,52	0,31	36
5	2,12	0,82	2,22	1,2	17,7	0,27	0,2	58
5	2,36	0,9	2,37	1,4	17,4	3,07	2,66	60

(Fig. 15) con il codice MM209. Di questo modulo non conosciamo il chip utilizzato e pertanto ci limitiamo a riportare in **Tabella 8** i dati dichiarati avendo comunque verificato che trattasi di un amplificatore ad un canale (MONO) in classe D. A nostro avviso, da quanto emerso dalle misurazioni i Watt dichiarati non sono rms ma di picco (Pk), poiché per ottenere una distorsione THD dello 0.02 % siamo riusciti a prelevare soltanto 300 mW su 8 Ω , mentre, rispettivamente con 1Wrms, su 8 Ω e 1.4Wrms su 4 Ω si ottiene una distorsione di circa il 3% che sarebbe più che accettabile, considerando però che la distorsione di 3° armonica risulta tra 2,1 e 2,6% e quindi superiore a quanto ci siamo prefissati (2%). La riproduzione delle frequenze basse, subisce una progressiva attenuazione di circa 4-5 dB a partire da 300 Hz e quindi la banda passante (BW) risulta nei limiti accettabili. La **Tabella 9**, riporta i dati rilevati su questo modulo, mentre in **Fig. 16** sono evidenziate

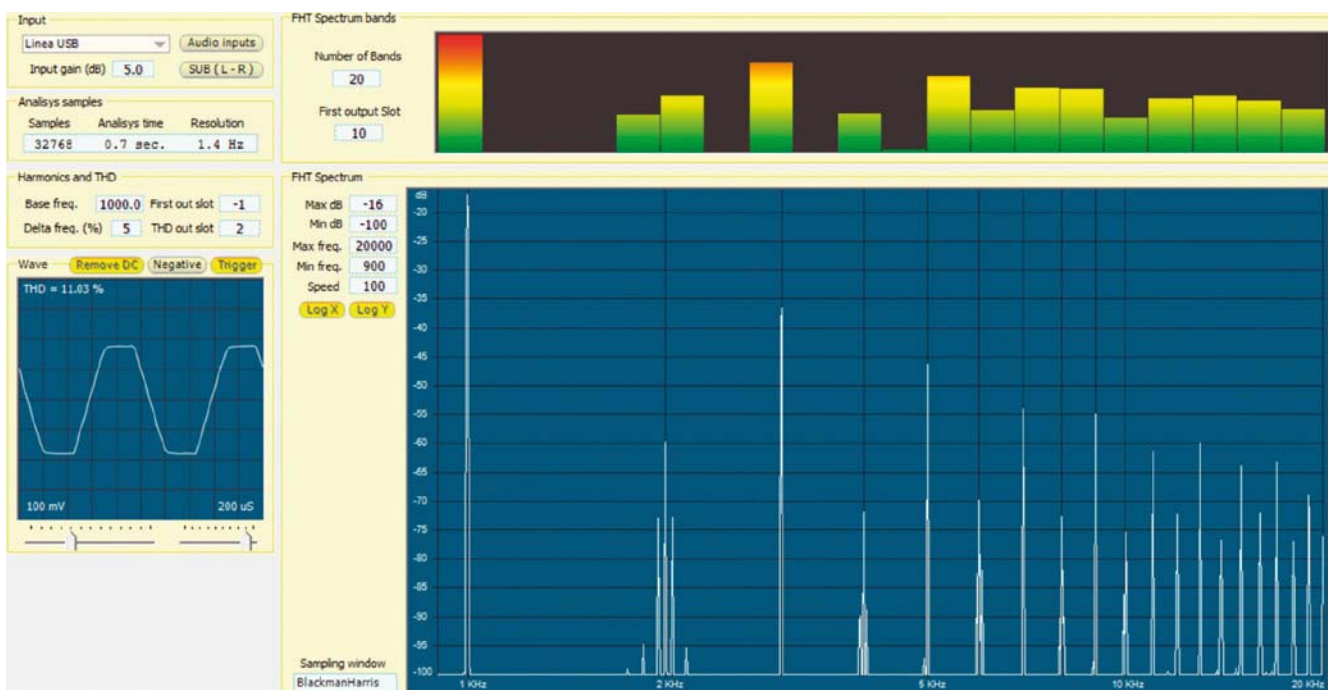
deformazione sinusoidale (clipping simmetrico) con relativa THD (11%) e spettro delle armoniche rilevato a 5V - 4 Ω - 1,7W. In questo caso i 2.4W dichiarati non sono stati dai noi rilevati quanto Wrms e quindi riteniamo che poiché non specificato, si tratti di W/pk, poiché: $(1,7 \times 1,4142 = 2,4)$. Ciò che tuttavia ci ha un po' stupito è il basso rendimento del modulo, che per un classe D dovrebbe quantomeno attestarsi intorno all'80%, ma questo dipende dalla scelta tecnica del progetto, per cui noi riportiamo semplicemente quanto riscontrato. Noi consigliamo l'utilizzo di questo modulo per applicazioni "mono" per esigenze non superiori a $1 \div 1,2$ Watt rms.

MODULO BLUEAMP 2X5W

L'ultimo componente di questa serie che andremo a testare in questa seconda parte del nostro percorso tra i vari "modulini" è un interessante assieme stereo di 5+5W (*dichiarati*) di dimensioni un po' più consistenti (98x63x25), ma che dispone di slot per accumulatore al litio rimovibile, da 3,6 V del tipo 18650, ma soprattutto trovano posto on-board, uno slot per scheda SD dove caricare brani musicali, una porta USB da collegare ad un PC per trasferire la musica e una micro-usb per ricaricare la batteria. E per finire, non poteva mancare un chip Bluetooth per il collegamento diretto con tablet, smartphone ecc.. nonché quattro (4) pulsanti di controllo, quali: *Volume+*, *Volume-*, *Mode* e *Play*. Ci troviamo quindi di fronte ad un vero e proprio sistema di riproduzione completo di tutto (cod. BLUEAMP10W). Basta collegare



→ **Fig. 15**
Modulo
MM209



quattro fili ai due altoparlanti o a due mini casse pe poter immediatamente riprodurre i brani musicali desiderati, siano essi memorizzati sulla SD che provenienti da una fonte esterna quale ad esempio il Bluetooth.

Tramite la porta micro USB, potremo poi tranquillamente ricaricare la batteria, la cui carica viene monitorata dal led di stato dedicato. In **Fig. 17**, sono illustrati i collegamenti alle casse e la varie funzioni disponibili.

Anche questo modulo si basa sul chip PAM8403 per cui anche in questo caso, non andremo a creare una tabella con i dati del costruttore poiché

ci si può comodamente riferire alla **Tabella 5**. In **Tabella 10** invece possiamo trovare i dati rilevati dalle nostre prove di laboratorio. Per i nostri vari test, abbiamo "iniettato" il segnale direttamente sui pin di ingresso tramite un condensatore da 0,47uF in serie ad un resistore da 10 K Ω . In questo modo abbiamo testato solo la sezione "pulita" dell'amplificatore, tralasciando volutamente i possibili disturbi derivanti da una connessione bluetooth che potrebbe non risultare ottimale e falsare quindi i test. I test sono stati effettuati con due tensioni, rispettivamente 3.6V (tensione nominale di batteria) e 4 V tensione su

Fig. 16
Distorsione armonica 11% con V alim. 1.7W out su 4 Ohm del modulo MM209.

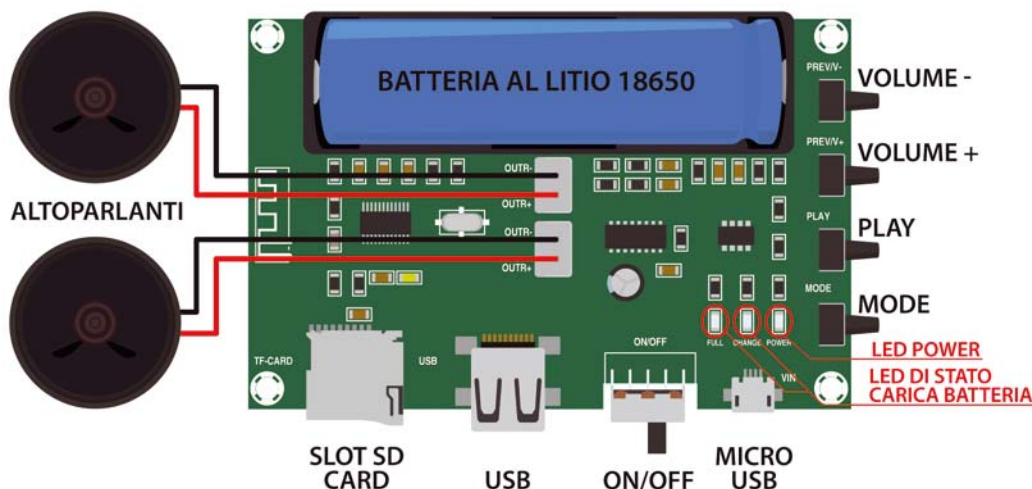


Fig. 17
Modulo BLUEAMP10W con relativi collegamenti.

→ **Tabella 10**
Parametri rilevati
sul modulo
BUEAMP10W.

V_ALIM	P (W)	Vpp In	Vout_rms	Po_8Ω (W)	Av (dB)	THD %	THD % 3a	rend. %
3,6	0,76	0,88	1,94	0,470	15,9	0,62	0,19	62
3,6	0,79	0,92	2	0,500	15,8	2,44	1,86	63
4	0,93	0,82	2,16	0,583	17,4	0,23	0,23	63
4	0,98	0,86	2,24	0,627	17,3	2,55	2,24	64
4,08 (*)	/	0,98	2,53	0,800	17,3	0,07	0,04	/
V_ALIM	P (W)	Vpp In	Vout_rms	Wrms_4Ω	Av (dB)	THD %	THD % 3a	rend. %
3,6	1,19	0,72	1,56	0,608	15,7	1,43	0,46	51
3,6	1,25	0,76	1,61	0,648	15,6	3,42	2,51	52
4	1,48	0,81	1,75	0,766	15,7	1,15	0,45	52
4	1,65	0,95	1,87	0,874	14,9	7,72	7,59	53
4,05 (*)	/	0,95	2,05	1,051	15,7	0,35	0,05	/

batteria al 90% di carica. Inoltre è stata fatta una prova con caricabatteria (micro-usb) inserito dove abbiamo rilevato tra i 4,05 e 4,08 Vcc. Con questo modulo, non siamo riuscite ad andare oltre i 17 dB di guadagno per una efficienza intorno al 60%, senza ottenere distorsioni importanti. La **Fig. 18** illustra molto chiaramente la distorsione ottenuta (7,7%) a soli 0,87 Watt su 4Ω e la risposta in frequenza (BW) è la stessa dei moduli precedenti e quindi buona. Alla fine, considerando il rapporto qualità/costo (poco più di 10 Euro), tutto sommato possiamo dire che questo modulo presenta caratteristiche interessanti, purché non si pretendano livelli sonori superiori a 1 Wrms.

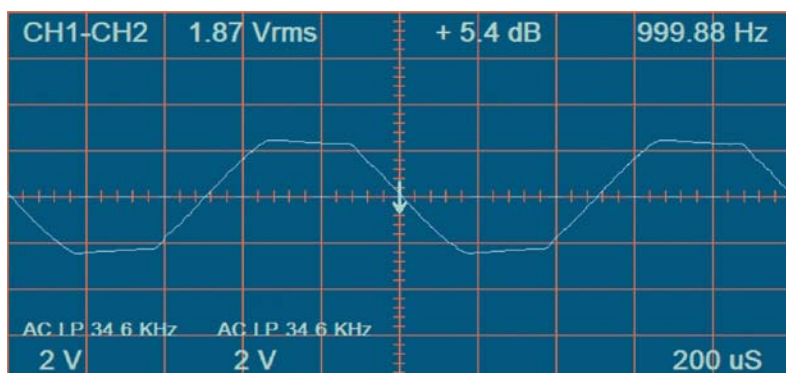
È interessante notare che già solo con alimentazione leggermente superiore (4,05 ÷ 4,08 Vcc) misurata con USB collegata, si ottiene un netto miglioramento delle prestazioni che si traduce in maggiore, seppur lieve, amplificazione e minore distorsione. Pertanto, la massima prestazione la si può ottenere soltanto a batteria completamente carica o con alimentazione USB inserita.

Va comunque considerato che questo componente, pur essendo un ottimo sistema "integrato", risulta ben lontano dall'erogare 5 + 5 Watt (nem-

meno se considerati come $W/Pk/Pk$). D'altra parte non potrebbe essere così, visto che il costruttore stesso dichiara come massimo 3W ma con distorsione 10% e alimentazione a 5V. Comunque, in futuro speriamo di poter trovare altri validi moduli con Bluetooth dei quali occuparci, possibilmente più potenti e maggiormente performanti. Per ora, ma solo per ora, terminiamo qui la nostra panoramica su alcuni dei più noti e versatili mini moduli commerciali presenti sul mercato, rimandando l'appuntamento ad una prossima puntata dedicata a moduli di potenza più performanti, costruttivamente più complessi e adatti quindi ad applicazioni più impegnative.

Non mancheremo comunque anche di tenere in debita considerazione l'evoluzione di questa fascia di prodotti così importanti, per tornare prossimamente ad occuparcene di volta in volta che troveremo qualcosa di interessante e come sempre innovativo, restando fedeli al nostro motto e al nome della nostra rivista: Elettronica INnovativa. □

↓ **Fig. 18**
Forma d'onda
fortemente
"clippata";
distorsione
7.7% a 0.87 W
su 40hm e 4V
alimentazione
relativa al
modulo
BUEAMP10W.



Cosa occorre?

I moduli presentati in questo articolo sono disponibili presso Futura Elettronica. L'amplificatore con LM386 (cod. HR02147) è in vendita a Euro 4,00, il modulo con PAM8302 (cod. PAM8302A) è disponibile a Euro 4,50, la scheda con PAM8403 (cod. MICROAMPLID2X3W) costa Euro 4,00, l'amplificatore con lo stesso modulo PAM8403 ma con anche il potenziometro (cod. PAM8403POT) è disponibile a Euro 6,50, il modulo MM209 (cod. MM209) è in vendita a Euro 15,00 infine l'amplificatore con Bluetooth (cod. BLUEAMP10W) costa Euro 10,50. 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

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.

La Presa su misura per te!

PRESA MULTIPLA SCHUKO 6 POSIZIONI CON PROTEZIONE DA SOVRATENSIONE PROTECTION

€ 21,⁵⁰

Cod. **WTPRESAM6P**



Presse multipla universale con 6 prese Schuko (angolate di 45°) con protezione da sovratensione, interruttore ON/OFF luminoso, struttura robusta con alloggiamento antiurto, LED di sicurezza luminoso. Lunghezza del cavo 1,4 metri con spina Schuko, corrente massima 16 A, potenza massima 3500 W.

SET ESPANDIBILE PRESA DA RETE RADIOCOMANDATA

Set composto da una presa passante radiocomandata e da un radiocomando portatile in grado di attivare (ON/OFF) un carico massimo di 1100 W. Il set è espandibile, con la possibilità di aggiungere ulteriori trasmettitori e abbinare fino a quattro prese, ciascuna associata ai quattro canali operativi del trasmettitore. Frequenza di lavoro 433 Mhz con codifica ad elevato grado di sicurezza, basso consumo energetico. Presa di tipo tedesco Schuko con protezione per bambini. Il radiocomando copre distanze fino a 30 metri e risulta particolarmente adatto per controllare le prese posizionate in punti difficili da raggiungere. Il trasmettitore è alimentato con batteria di tipo 23A inclusa nella confezione.

€ 22,⁹⁰
cod. **WTWRS**



Prodotti aggiuntivi al set WTWRS:

PRESA RADIOCOMANDATA ADDIZIONALE AL SET WTWRS



€ 12,⁹⁰
Cod. **RXWTWRS**

RADIOCOMANDO 4 CANALI ADDIZIONALE AL SET WTWRS



€ 9,⁰⁰
Cod. **TXWTWRS**

PRESA TIMER DIGITALE SETTIMANALE DA ESTERNO 3600W - IP44

Consente di accendere e spegnere apparecchiature elettriche nei giorni e nelle ore della settimana prestabilita. Dispone di pulsante On/Off, 10 programmi di funzionamento per ciascun giorno della settimana con step di regolazione di 1 minuto. Ha inoltre 16 diverse combinazioni giornaliere, batteria di backup, formato dell'orario 12/24 ore, funzione di regolazione automatica dell'ora legale, presa di tipo SCHUKO. Adatto per impieghi all'esterno.



€ 16,⁵⁰
Cod. **WT51301**

PRESA DI CORRENTE SCHUKO DA 16A CON 2 USCITE USB DA 5 V - 3,15 A



€ 11,⁹⁰
Cod. **WT40885**

Presse di corrente passante con due prese USB da 5 V - 3,15 A (totali). Ideale per ricaricare smartphone, lettore MP3, fotocamera digitale o tablet tramite cavo USB.

PRESA DI RETE SCHUKO CON PROTEZIONE DA SOVRATENSIONE



€ 6,⁶⁰
Cod. **WT51270**

Presse in grado di proteggere i dispositivi elettrici dalle sovratensioni come quelle causate dai fulmini. Dispone inoltre di LED di stato che si accende appena viene alimentata.



CONOSCERE E USARE

di FRANCESCO FICILI

8

Node-RED

Impariamo ad usare Node-RED, un tool di flow-based programming orientato all'IoT ed alla connettività, originariamente sviluppato dall'IBM Emerging Technology Services team e adesso parte della JS Foundation. In questa puntata parliamo dell'integrazione degli assistenti vocali con Node-RED. Ultima Puntata.

Come abbiamo già visto ampiamente nella puntata precedente, Node-RED è facilmente interfacciabile con strumenti di utilizzo quotidiano, come ad esempio, per citarne solo alcuni, email, strumenti social ed applicazioni di messaggistica istantanea. Un altro oggetto con cui possiamo interfacciare agevolmente Node-RED sono gli assistenti vocali, un segmento di mercato di oggetti IoT che ha avuto una rapidissima crescita negli ultimi anni. In Italia parliamo di un valore di circa mezzo miliardo di euro, destinato ad aumentare vertiginosamente dal momento che gli assistenti vocali sono un enabler per un settore più ampio che possiamo genericamente definire come Smart Home, che vede la connessione di vari sensori ed attuatori all'interno delle nostre case per abilitare una gestione sempre più Smart dei dispositivi casalinghi. Esistono ormai diversi assistenti vocali disponibili sul mercato, sostanzialmente molto simili tra di

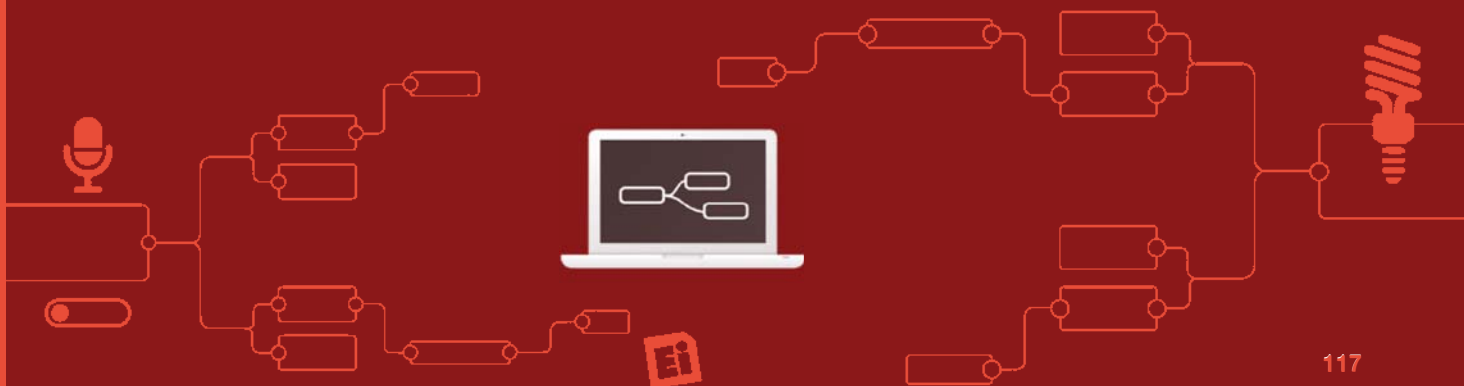
loro, sebbene ognuno con le sua peculiarità; i più comuni sono sicuramente gli assistenti vocali dei colossi Amazon e Google, ossia Alexa e Google Home. In questa puntata noi ci soffermeremo principalmente sui dispositivi della famiglia Alexa, tralasciando il caso di Google Home, che è comunque supportato da Node-RED.

INTEGRAZIONE DI ALEXA IN NODE-RED

Esistono diversi nodi che possono essere utilizzati per integrare Alexa in Node-RED, e tutti questi nodi possono essere divisi sostanzialmente in due categorie:

- Nodi che permettono di emulare dispositivi controllabili via Alexa in Node-RED
- Nodi che permettono di interagire con le API di Alexa

I primi consentono di sfruttare le feature di controllo vocale di Alexa per controllare dei flussi. Si tratta essenzialmente di nodi di input, che, a





 **Fig. 1** - Assistenti vocali di Amazon.

fronte di una determinata configurazione, permettono di triggerare dei flussi tramite controllo vocale. Questa famiglia di nodi è perfetta per realizzare applicazioni domotiche in quanto permette di controllare luci, relè, interruttori, tapparelle elettriche e via dicendo. Il secondo gruppo di nodi permette invece di interagire con la moltitudine di APIs messe a disposizione dall'ecosistema Alexa, quindi sono più orientati alla gestione dei dispositivi Amazon da Node-RED. Con questa famiglia di nodi è possibile quindi controllare Alexa come text-to-speech device (facendo quindi riprodurre vocalmente una stringa), o controllare la riproduzione di file musicali, oppure ancora inviare notifiche push ad un dispositivo. La lista è decisamente non esaustiva, dal momento che le operazioni eseguibili sono davvero tante, quindi non perdiamo tempo e passiamo alla descrizione di una libreria del primo gruppo, che ci consentirà di controllare tramite comandi vocali Node-RED e, di conseguenza, le interfacce hardware del dispositivo su cui lo stiamo facendo girare, ossia nel nostro caso, la nostra fida Raspberry Pi.

NODE-RED COME ALEXA DEVICE

Come accennato precedentemente, esiste una famiglia di nodi che permette sostanzialmente di emulare dispositi-

tivi Alexa tramite Node-RED. Esistono diverse librerie in questa famiglia, ognuna con le sue caratteristiche peculiari ed i suoi pregi e difetti; noi ne consigliamo una che riteniamo decisamente valida e che si basa su una Alexa skill e su un servizio di Bridging per garantire l'interfacciamento.

Il nome della libreria è **node-red-contrib-alexa-home-skill** (realizzata da Ben Hardill), e può essere installata, come di consueto, cercandone il nome tramite la l'opzione "manage palette" di Node-RED.

Affinchè la libreria funzioni correttamente è necessario eseguire i seguenti passi:

1. Installare la libreria in Node-RED
2. Registrarsi al servizio di bridge con la skill Alexa
3. Aggiungere le descrizioni dei dispositivi che si desidera implementare
4. Installare la skill Node-RED sull'app Alexa

Iniziamo quindi installando la libreria, che è composta da due nodi più un configuration node (vedi Fig.2).

Descriviamo brevemente i nodi che compongono la libreria:

alexa-home: questo è il nodo base che permette di intercettare un comando vocale per uno specifico dispositivo. I dispositivi devono essere preventivamente configurati tramite il servizio di bridge con la skill, prima di poter essere utilizzati. La configurazione contiene, oltre al nome del dispositivo stesso, anche le sue proprietà, tra cui ad esempio la tipologia (lampada, interruttore, tapparella, etc..) ed i comandi vocali ai quali il dispositivo può obbedire.

alexa-home-resp: questo nodo permette di inviare una risposta personalizzata in merito all'esecuzione di una determinata operazione (ad esempio per confermare che l'attuazione di una luce è andata a buon fine). Non ci soffermeremo molto su questo nodo, in quanto le medesime operazioni possono essere eseguite in maniera più semplice e completa, tramite alcuni nodi appartenenti al secondo gruppo.

alexa-home-conf: questo è il classico nodo di configurazione che permette di collegare un determinato account del servizio di bridge ad una specifica istanza di Node-RED, e quindi permettere di accedere ai dispositivi configurati per quello specifico account.

Una volta installata la libreria, prima di iniziare ad utilizzarla sarà necessario registrarsi al seguente sito web: <https://alexa-node-red.bm.hardill.me.uk/>, la cui home page è rappresentata in Fig.3. Una volta registrati sul sito possiamo iniziare ad aggiungere i dispositivi che desideriamo emulare tramite Node-RED. Come primo semplice esempio possiamo aggiungere un LED della nostra Raspberry Pi, che poi andremo ad attivare tramite un comando vocale. Spostiamoci quindi nella sezione "Devices" del sito



 **Fig. 2** - Nodi della libreria node-red-contrib-alexa-home-skill.

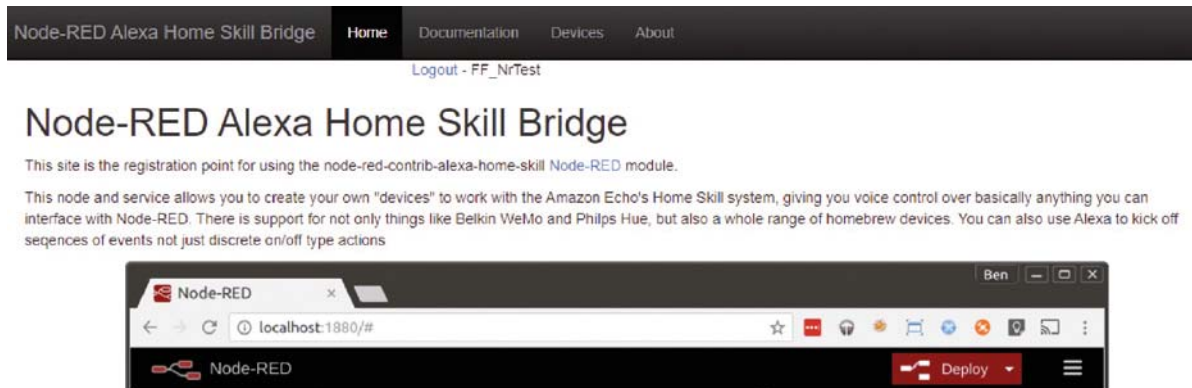


Fig. 3 - Home page del servizio di bridge.

Add New Device

Name:

Description:

Actions

On: ☒ **Off:** ☒

%: ☐ **+%:** ☐ **-%:** ☐

°C/F: ☐ **+°C/F:** ☐ **-°C/F:** ☐

Fig. 4 - Inserimento di un nuovo dispositivo.

a cui ci siamo appena registrati e clicchiamo sul pulsante "Add Device" per configurare un nuovo dispositivo. Verrà aperta una semplice maschera di configurazione che ci permetterà di configurare il nostro device. Trattandosi di un semplicissimo esempio, possiamo inserire la configurazione riportata in Fig. 4.

Al termine dell'operazione, se tutto è andato a buon fine, il dispositivo comparirà nella lista dei dispositivi, come riportato in Fig. 5. A questo punto, per far riconoscere i dispositivi ad Alexa, è necessario installare una skill nell'app Alexa (o tramite il vostro account Amazon) e collegare l'account del servizio di bridging in cui abbiamo configurato il nostro dispositivo. La skill da installare si chiama Node-RED (di Ben Hardill), ed una immagine della pagina descrittiva della skill è visibile in Fig. 6.

La skill si può facilmente trovare cercando la voce "Node-RED" dall'app Alexa o dal vostro account Amazon. Una volta installata la skill va attivata e, dopo l'attivazione, verranno richieste le credenziali che avete inserito al momento della registrazione al servizio. A questo punto il dispositivo che è stato precedentemente aggiunto può essere riconosciuto da Alexa. Possiamo utilizzare il comando vocale "Alexa, trova nuovi dispositivi" per consen-

Devices

Name	Description	Actions
LED	This device will control a RPI LED.	☒ ☐
Edit Delete		
Add Device		

After making any changes remember to ask Alexa to update devices it knows about by saying "Alexa, discover devices"

Fig. 5 - Lista dei dispositivi.



Fig. 6 - Skill Alexa per l'interfacciamento con il servizio di Bridge.

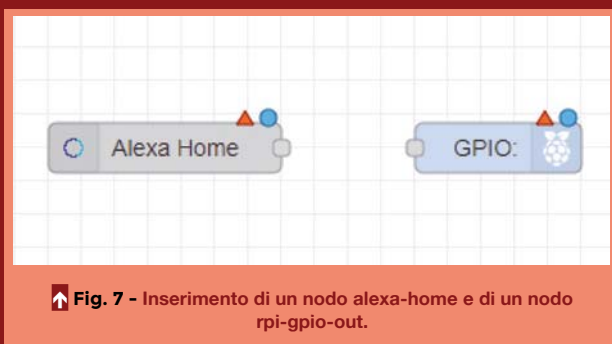


Fig. 7 - Inserimento di un nodo alexa-home e di un nodo rpi-gpio-out.

tire l'associazione. Dopo qualche istante Alexa dovrebbe confermare che un dispositivo denominato LED è stato trovato ed associato.

Possiamo tornare su Node-RED ed andare avanti con la creazione di un semplice flow per testare il nuovo dispositivo. Nella palette cerchiamo i nodi della categoria Alexa ed aggiungiamo un nodo alexa-home, oltre ad un nodo rpi-gpio-out, come illustrato in Fig.7.

Configuriamo, come di consueto, il nodo rpi-gpio-out per controllare un GPIO, e impostiamolo per controllare il GPIO 18, che, usando lo Shield Raspberry I/O

(7100-FT1060M), è collegato al primo LED.

Passiamo ora alla configurazione del nodo alexa-home, facendo un doppio click sul nodo stesso per accedere alla finestra delle proprietà. Qui clicchiamo sull'icona a forma di matita per accedere alla configurazione del configuration node alexa-home-conf, come illustrato in Fig. 8.

Il configuration node richiede semplicemente di inserire le credenziali dell'account creato in precedenza, quindi eseguiamo e clicchiamo su "Update". Una volta tornati sulla finestra delle proprietà del nodo alexa-home clicchiamo su "Done" ed eseguiamo un deploy. A questo punto, aprendo nuovamente la finestra delle proprietà, dovremmo poter vedere i nostri dispositivi (in questo caso, se non ne sono stati aggiunti altri, il dispositivo LED) sul drop-down control "Device". Selezioniamo LED, mettiamo il segno di spunta su Auto Acknowledge e clicchiamo nuovamente su "Done", come illustrato in Fig. 9.

A questo punto possiamo collegare con un wire l'uscita del nodo alexa-home e l'ingresso del nodo rpi-gpio-out (Fig.10) e testare il nostro flow.

Se tutto è stato fatto correttamente, usando il comando vocale "Alexa, accendi LED" si dovrebbe accendere il LED sullo shield, mentre con il comando "Alexa, spegni

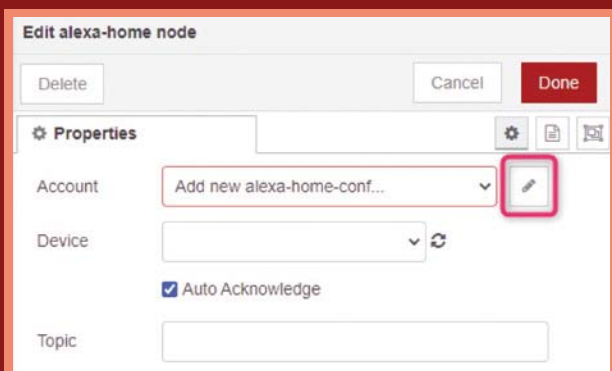


Fig. 8 - Accesso al configuration node.

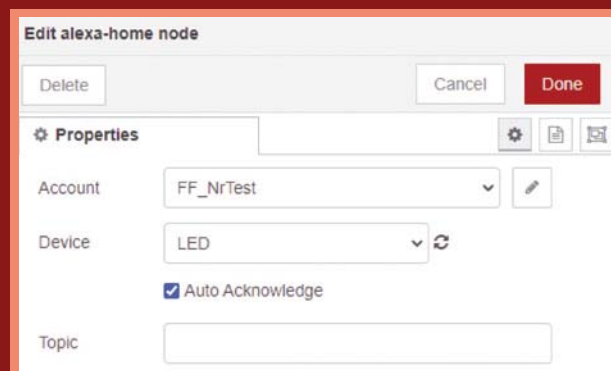


Fig. 9 - Configurazione del nodo alexa-home.

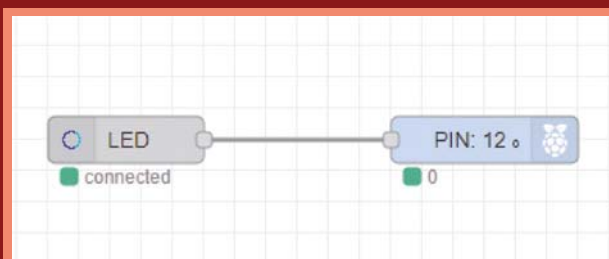


Fig. 10 - Flow per il controllo di un LED tramite comando vocale con Alexa.

LED" il LED dovrebbe spegnersi. Dal momento che è stata utilizzata l'opzione "Auto Acknowledge", Alexa fornirà automaticamente un feedback sul risultato dell'operazione ("OK" se tutto è andato per il verso giusto o un qualche tipo di errore nel caso contrario). Come avrete notato non è necessario nessun tipo di adattamento del messaggio di uscita in questo caso, in quanto il nodo alexa-home restituisce un boolean su msg.payload che riporta lo stato booleano associato al comando vocale impartito (true nel caso di "accendi LED" e false nel caso di "Spegni LED"). Naturalmente nel caso di attuazioni più complesse rispet-

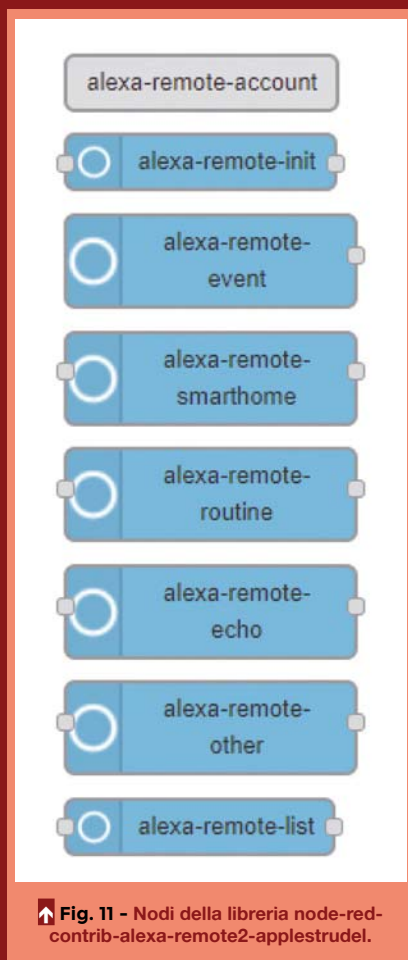


Fig. 11 - Nodi della libreria node-red-contrib-alexa-remote2-applestrudel.

to ad un semplice LED potrebbe essere necessario fare qualche adattamento al messaggio in uscita da alexa-home, ma sostanzialmente partendo da questa semplice base è possibile controllare (almeno con dei comandi di base) virtualmente qualsiasi dispositivo interfacciabile con Node-RED.

INTERFACCIAMENTO CON ALEXA API

Ora che abbiamo sperimentato con il primo gruppo di nodi, passiamo ad analizzare il secondo, ossia i nodi che ci consentono di interfacciarci con le API di Alexa. In questa famiglia di nodi, il rappresentante più noto è sicuramente la libreria node-red-contrib-alexa-remote2-applestrudel, che ha subito vari fork, passando da essere chiamata alexa-remote and alexa-remote2, cakebed ed infine alesxa-remote2-applestrudel, che è la versione che vi raccomandiamo di installare. Una volta installata questa libreria mette a disposizione un ampio set di nodi, riportati in Fig. 11. Come di consueto facciamo una rapida carrellata, per poi passare a qualche dettaglio sulla configurazione e ad un esempio pratico.

alexa-remote-init: nodo utilizzato per effettuare il controllo manuale sull'inizializzazione. Non è strettamente necessario utilizzarlo in quanto esiste una opzione di auto-init nel configuration node.

alexa-remote-event: nodo che permette di intercettare eventi, come errori, modifiche ad alcuni settaggi come il volume ed altro ancora. Non tutti gli eventi vengono però notificati da questo nodo.

alexa-remote-smarthome: nodo per il controllo dei dispositivi di tipo casa intelligente. Questo nodo permette sia di controllare dispositivi standard, come lampadine intelligenti e simili, sia i dispositivi emulati da Node-RED (come il LED dell'esempio precedente).

alexa-remote-routine: questo nodo permette di triggerare vari tipi di routine, tra cui:

- **Speak:** riproduce una stringa di testo.
- **Speak at volume:** riproduce una stringa di testo ad uno specifico volume.
- **Text Command:** permette di inviare tramite una stringa un comando vocale.
- **Wait, Stop:** utilizzati per il controllo di esecuzione nelle sequenze.
- **Prompt, Phrase:** permette di far eseguire ad Alexa propt o frasi standard (ad esempio fornire notizie sul tempo o sul traffico in tempo reale o utilizzare formule di saluto standard).
- **Sound:** riproduce un suono (ad esempio un allarme o il verso di un animale).
- **Volume:** permette di controllare il volume.
- **Music:** Permette di riprodurre musica.
- **SmartHome:** permette di accedere al controllo dei dispositivi per la casa intelligente (sostanzialmente la stessa funzione del nodo alexa-remote-smarthome).

- **Launch Skill:** permette di lanciare una skill di Alexa (naturalmente la skill dev'essere installata).
- **Routine:** permette di eseguire una routine predefinita in Alexa.
- **Push Notification:** permette di inviare una notifica push al dispositivo associato all'account.
- **Node:** permette di implementare una sequenza complessa contenente, in numero ed ordine arbitrario, le funzioni elencate in precedenza.

alexa-remote-echo: questo nodo permette di gestire alcune funzioni fondamentali dei dispositivi, sia amministrative che non (come ad esempio rinominare o cancellare un dispositivo o gestire il pairing o la connessione di dispositivi bluetooth)

alexa-remote-other: questo nodo fornisce un'interfaccia con tutte le altre funzioni (ad esempio varie funzioni per il reperimento di informazioni o la gestione delle notifiche).

alexa-remote-list: questo nodo fornisce un'interfaccia dedicata per la gestione delle liste.

alexa-remote-account: questo è il configuration node che permette di interfacciare il nostro account Amazon con Node-RED ed impostare alcune configurazioni di base.

Ora che abbiamo dato uno sguardo ai vari nodi messi a disposizione, passiamo alla configurazione e ad un semplice esempio. Per cominciare trasciniamo sul workspace un nodo "Routine" e facciamo doppio click per accedere alla finestra delle proprietà. Come di consueto, la prima cosa da fare sarà impostare le proprietà del configuration node, quindi clicchiamo sull'icona a forma di matita (Fig.12). Qui impostiamo:

Nome: il nome che preferite per la vostra configurazione (nell'esempio MyNrConfig).

Auth method: Proxy

This IP: inserite l'IP locale della Raspberry sulla quale state facendo girare Node-RED (nell'esempio 192.168.1.10)

Port: lasciate pure 3456

File Path: qui va indicato il path di un file di testo locale nel quale verranno salvate le credenziali per gli accessi successivi

Refresh: 0

Service Host: alexa.amazon.it

Page: amazon.it

Language: it-IT

Per il resto lasciate sempre i valori di default. Salvate ed eseguite un deploy. A questo punto dovrete notare che sotto al nodo routine che avevate trascinato sul workspace compare il seguente messaggio: "open 192.168.1.10:3456 in your browser" (Fig.13). Naturalmente l'indirizzo IP sarà quello che avete inserito in fase di configurazione, ossia quello della vostra Raspberry nella vostra rete locale. Se aprite il browser all'indirizzo indicato dovrete trovare una pagina di autenticazione sulla quale dovrete inserire le

Fig. 12 - Finestra di configurazione del configuration node di alexa remote2 applestrudfel.

credenziali del vostro account amazon.

Se tutto viene eseguito correttamente dovrete visualizzare un messaggio sul browser che conferma il successo dell'operazione e visualizzare la scritta "ready" sotto il nodo routine. A questo punto i nodi alexa remote sono operativi e possono essere utilizzati.

Per testarli possiamo utilizzare lo stesso esempio utilizzato

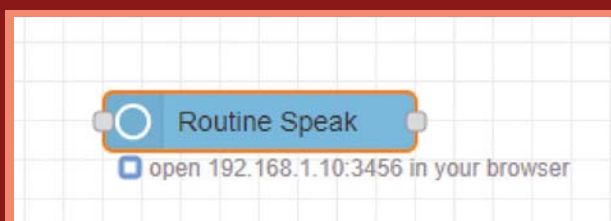


Fig. 13 - Messaggio con l'indirizzo IP della pagina di autenticazione.

Intervista a Nick O'Leary

Nick O'Leary è uno dei creatori di Node-RED, lo abbiamo raggiunto via mail per fargli qualche domanda sul passato, presente e futuro del tool che ha creato.

Elettronica In: Ciao Nick, come puoi immaginare molti dei nostri lettori sono curiosi riguardo a come lo sviluppo di Node-RED ha avuto inizio. Puoi darci qualche dettaglio a riguardo?

Nick: Certamente. Nel 2013 stavo lavorando presso l'IBM Emerging Technology Group. A quel tempo lavoravo su progetti per diversi clienti, spesso mettendo insieme diversi elementi tecnologici per dare vita ai progetti più disparati. Come parte del ruolo eravamo attivamente incoraggiati a spendere parte del nostro tempo usando nuove tecnologie e mettendo insieme proof-of-concepts di nuove idee. Avendo un background in MQTT ho realizzato una semplice demo di un tool che permetteva di connettere insieme nodi che rappresentavano topics MQTT per mezzo di "wires". Non c'è voluto molto ad aggiungere un pulsante che permetteva di generare il codice in grado di implementare il sistema MQTT progettato. A quel punto un collega che stava per iniziare un nuovo progetto per un cliente mi suggerì che sarebbe stato molto utile avere la possibilità di mappare messaggi MQTT su una porta seriale, e così il gruppo di nodi Serial venne aggiunto al PoC. Da quel punto in avanti ogni nuovo progetto aggiunse un nuovo set di requisiti e un nuovo set di nodi al tool. In breve diventò chiaro che non si trattava più solo di MQTT, ma avevamo tra le mani un tool per la progettazione dei sistemi più disparati. La prima versione di Node-RED era nata a quel punto.

Elettronica In: Avresti mai immaginato che Node-RED potesse avere un simile successo?

Nick: Nei primi mesi siamo stati molto impegnati a consegnare progetti per i clienti, spesso usando Node-RED per realizzarli. Sapevamo di avere qualcosa di utile tra le mani, ma la vera domanda era cosa fare con quel qualcosa. Sin dall'inizio abbiamo deciso che la migliore cosa da fare fosse di renderlo open-source, per far sì che la vasta community degli sviluppatori IoT potesse metterci le mani sopra. Dopo essere passati per il processo interno della compagnia abbiamo ricevuto l'approvazione a renderlo open-source e così il progetto Node-RED su GitHub vide la luce. Non avrei mai potuto immaginare quando la community potesse trovarlo utile e quanto potesse farlo crescere, ma è successo. Continuo a volte a vederlo come il mio piccolo side-project, ma da molto tempo ormai è molto più di quello.

Elettronica In: Quali sono i tuoi programmi adesso? Abbiamo visto che hai lasciato IBM, lo hai fatto perché vuoi dedicarti al



100% a Node-RED nel prossimo futuro?

Nick: Sono stato incredibilmente fortunato a trovare un ruolo in IBM, che mi ha concesso di guidare il progetto di Node-RED e renderlo open-source. Avrei potuto continuare felicemente in quel ruolo, ma all'inizio del 2021 mi è stata data l'opportunità di costruire qualcosa di più grande con Node-RED. Quindi in Aprile ho lasciato IBM ed ho fondato la mia compagnia, FlowForge Inc., che ha lo scopo di costruire una piattaforma attorno a Node-RED, così che possa essere più facile per altre compagnie adottarlo.

Elettronica In: Quali compagnie fanno uso di Node-RED?

Nick: C'è una crescente lista di compagnie che hanno adottato Node-RED. E' molto interessante osservare da quali settori provengono. Molte di loro sono società che lavorano nell'IoT, come ci si potrebbe aspettare, ma ci sono altre compagnie che lo usano e che lavorano in altri settori, come l'healthcare, l'insurance e l'automotive.

Elettronica In: Quali sono le funzionalità più interessanti della release corrente (Node-Red 2.0)? E quali sono le più promettenti feature future che puoi anticiparci?

Nick: Nella release 2.0 ci sono due nuovi tools piuttosto interessanti: un flow debugger ed un Flow Linter.

Il debugger permette di inserire breakpoints nei flows, esattamente come si farebbe con un tradizionale debugger a livello di codice. Il Linter è un analizzatore che controlla il flow usando una serie di regole e permette di evidenziare potenziali problemi nel flow. Siamo ancora agli albori per entrambi questi tools, ma in futuro forniranno agli utilizzatori nuovi superpoteri per aiutarli a creare applicazioni con un elevato livello di qualità. Un nuovo elemento sul quale abbiamo iniziato a lavorare è un Flow Testing Framework. Quello che vorremmo fare è fornire agli sviluppatori uno strumento che gli consenta di creare facilmente test per i loro flow, in modo da garantire che funzionino come ci si aspetta.

in precedenza, completandolo con una conferma customizzata alle operazioni di accensione e spegnimento del LED. Per farlo trasciniamo sul workspace un altro nodo "Alexa Routine" ed un nodo Switch.

Configuriamo il nodo switch con due regole, rispettivamente "is true" ed "is false" (vedi Fig.14), per creare due

output alternativi che vengono triggerati in funzione dell'operazione richiesta (ricordate che il nodo alexa-home ritorna appunto un boolean come risultato della richiesta di accensione o spegnimento del LED). Ora configuriamo i due nodi Alexa routine per riprodurre sul dispositivo che vogliamo rispettivamente le due frasi "LED Acceso" e "LED

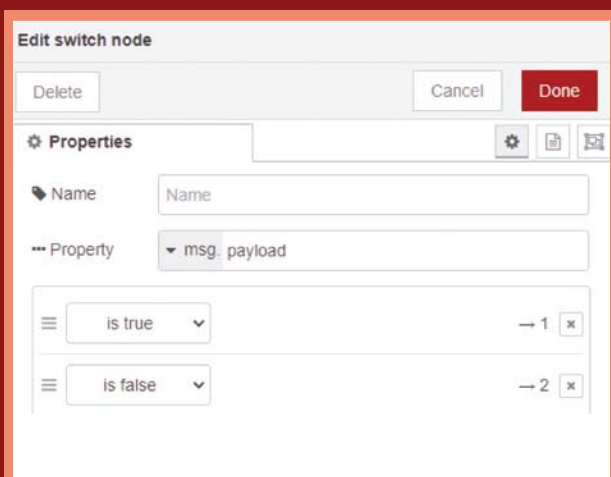


Fig. 14 - Configurazione del nodo Switch.

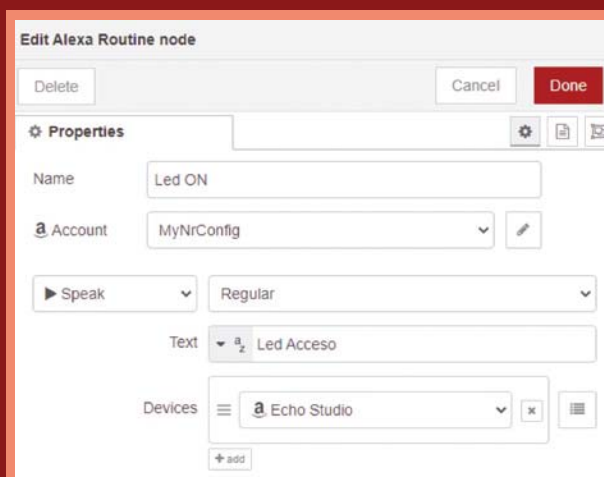


Fig. 15 - Configurazione del nodo Alexa Routine usato per riprodurre il comando vocale "LED acceso".

spento" e colleghiamoli ai due rami "is true" ed "is false" del nodo switch. Ovviamente il nodo che riproduce la frase "LED acceso" sul ramo "is true" e l'altro su "is false". Infine colleghiamo l'uscita del nodo alexa-home all'ingresso del nodo switch. Infine rimuoviamo la spunta sulla checkbox "Auto Acknowledge" del nodo alexa-home. In Fig. 15 è riportata la finestra delle proprietà di uno dei nodi Alexa Routine ed in Fig. 16 un'immagine del flusso completo. A questo punto se ritestate il vostro flow noterete che le risposte di Alexa sono cambiate, infatti invece di rispondere sempre "OK" (funzione di auto acknowledge, che abbiamo disabilitato), risponde con le frasi "LED acceso" e "LED spento" a conferma delle due operazioni. Come avrete potuto facilmente notare, quelli che abbiamo presentato sono solo due semplicissimi esempi delle potenzialità derivanti dalla combinazione di Node-RED ed Alexa, lasciamo ai lettori più interessati ed intraprendenti l'esplorazione delle varie funzioni aggiuntive.

CONCLUSIONI

Siamo arrivati alla conclusione del corso Node-RED. Durante questa avventura, dopo aver gettato le necessarie basi teoriche sulla FBP (Flow Based programming), siamo passati dai primi, semplici, esempi di gestione dell'I/O (GPIO, I2C, UART, 1-wire), ad esempi sempre più complessi, come la connettività (TCP, UDP, http, WebSocket ed MQTT), l'interfacciamento con le app di messaggistica (Whatsapp e Telegram) ed infine l'integrazione con gli assistenti vocali. Ci sarebbero molti altri temi, che per questioni di tempo, non possono essere trattati in questo corso, come il Machine Learning e l'AI, l'interfacciamento con i social (Facebook, Twitter) ed altro ancora, insomma Node-RED è uno strumento potente e flessibile, ma allo stesso tempo, alla portata di tutti, quindi non vi resta altro da fare che prendere una Raspberry (se non ne siete già in possesso) ed iniziare a sviluppare i vostri flow e le vostre automazioni. □

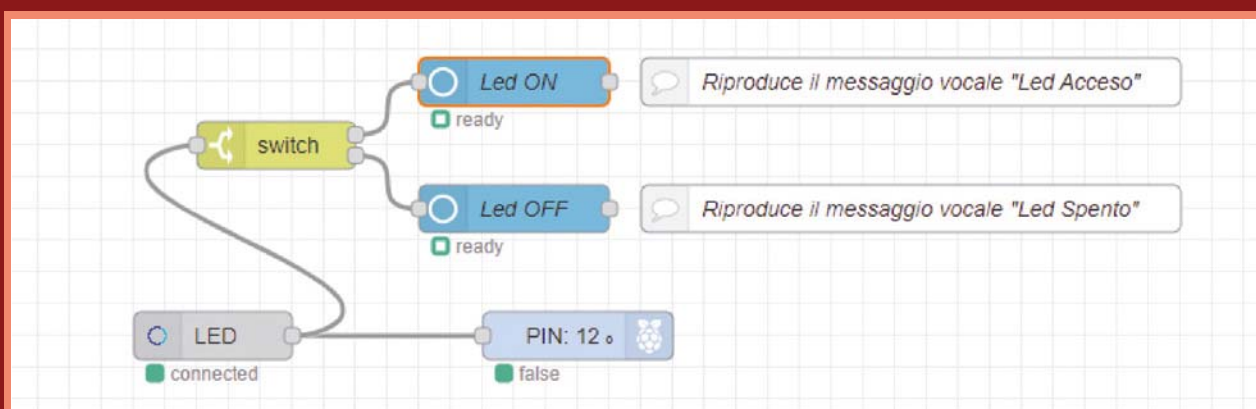


Fig. 16 - Flusso completo con risposte customizzate.

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

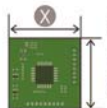
1 Layer

2 Layers

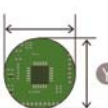
4 Layers

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.



Il sale fuso congela l'energia per una stagione

La decarbonizzazione delle reti elettriche aumenta la necessità di opzioni di stoccaggio a lungo termine per l'energia generata da fonti rinnovabili, in particolare per eolico e solare. Le batterie tradizionali, sebbene siano la soluzione preferita per l'utenza finale, sono poco pratiche su larga scala a livello di rete elettrica, pertanto gli scienziati cercano soluzioni per l'accumulo di grandi quantità di energia, tra cui le batterie a sale fuso. Un recente studio del Pacific Northwest National Laboratory (PNNL) dimostra che tali batterie possono "congelare" la loro carica per mesi, mantenendo il 92% della loro capacità per tre mesi.

Le batterie a sale fuso utilizzano un elettrolito liquido a sale fuso, che si congela a temperatura ambiente, bloccando o quasi l'autoscarica; quando riscaldati, il catodo, l'anodo e gli elettroliti si separano ed avviene lo scambio ionico, ossia la cessione dell'elettricità.

I ricercatori hanno usato nichel e alluminio rispettivamente per il catodo e l'anodo, con tetracloruro di sodio e alluminio (Na-AlCl_4) come elettrolito di sale fuso, tutti materiali relativamente economici e diffusi; l'elettrolito ha un punto di fusione di circa 157 °C e rimane solido a temperatura ambiente.

Ora puntano a sostituire il nichel con un materiale a basso costo come il ferro, oltre a considerare elettroliti a più basso punto di fusione, in modo da facilitarne e renderne economico il riscaldamento.

Allo stato della ricerca, il team del Pacific Northwest National Laboratory ipotizza di arrivare all'applicazione commerciale su larga scala in un periodo di tempo compreso tra il 2030 e il 2035, in particolare per l'utilizzo in sistemi distribuiti e operativi.

<https://www.pnnl.gov>



Recuperare gli stati dei razzi? Ci prova l'elicottero!

Uno dei problemi correlati al lancio di navicelle e satelliti mediante razzi vettori è che questi razzi sono necessariamente composti da stadi booster, parte dei quali viene espulsa, una volta esausta, all'interno dell'atmosfera e ricade al suolo: spesso in mare. Gli stadi booster esausti, ossia che hanno esaurito la loro spinta, diventano così "spazzatura" che se cade al suolo può costituire pericolo e richiede comunque il recupero, mentre se finisce negli specchi d'acqua e nel mare, forma comunque inquinamento e andrebbe recuperata. Il recupero è, peraltro, economicamente conveniente, perché renderebbe riutilizzabili quei booster per altre missioni, invece di condannarli a diventare "vuoti a perdere"; salvarli da questo destino e quindi risparmiare denaro, è stato un obiettivo degli ingegneri aerospaziali sin dalla

prima era spaziale. Lo SpaceX di Elon Musk ha notoriamente fatto atterrare i suoi booster Falcon 9 su navi drone al largo della costa della Florida: un'operazione strabiliante a guardarla, ma molto difficile da realizzare.

Rocket Lab, invece, suggerisce un metodo di recupero alternativo: inseguire lo stadio in caduta frenata da un paracadute, mediante un potente elicottero che lo agganci mediante una robusta fune e lo riporti a terra. Il "battesimo del fuoco" per questa tecnica è avvenuto con la missione "There and Back Again", il 26° lancio Electron di Rocket Lab, che è decollato dal Pad A al Launch Complex 1 nella penisola di Mahia in Nuova Zelanda lo scorso 2

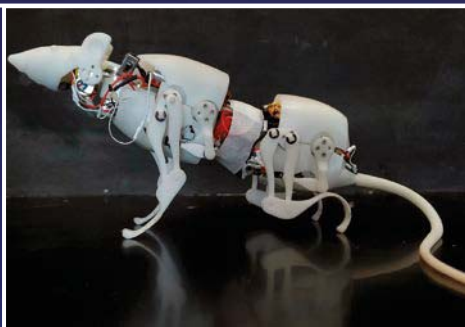
di maggio, dopo due rinvii causati dalle condizioni meteo avverse.

La missione del ventiseiesimo lancio del vettore Electron ha distribuito in orbita 34 satelliti degli operatori commerciali Alba Orbital, Astrix Astronautics, Aurora Propulsion Technologies, E-Space, Unseenlabs e Swarm Technologies tramite il fornitore di servizi di lancio globale Spaceflight Inc. Il lancio ha portato il numero totale di satelliti lanciati da Electron a 146.

Va detto che Rocket Lab è parzialmente riuscita nel test di cattura a mezz'aria il primo stadio del razzo vettore Electron mentre ritornava dallo spazio dopo il lancio: infatti la tecnica si è dimostrata praticabile, in quanto l'elicottero è riuscito ad agganciare lo stadio del razzo in caduta frenata dal paracadute, ma il pilota, riscontrando un'anomalia nella sollecitazione reale rispetto a quella prevista in fase di simulazione, ha preferito sganciarlo per rimanere entro un ampio margine di sicurezza. La cattura del "relitto" è avvenuta con un Sikorsky S-92 personalizzato, un grande elicottero birotore tipicamente utilizzato nel trasporto offshore di petrolio e gas e nelle operazioni di ricerca e salvataggio. Seppure parzialmente conclusa, la missione rappresenta il primo passo importante nel programma di sviluppo della Compagnia per rendere Electron un razzo riutilizzabile. Catturare lo stadio di un razzo che ritorna a mezz'aria mentre ritorna dallo spazio è un'operazione estremamente complessa che richiede estrema precisione. Invece di essere sganciato nel Pacifico, il primo stadio booster verrà catturato a mezz'aria mentre scende con il paracadute, e sarà riportato alla base, affumicato dal calore dell'atmosfera del rientro ma interiormente intatto, per un'eventuale ristrutturazione e riutilizzo.

Il razzo vettore Electron è alto 18 metri, di cui i 12 metri inferiori costituiscono il primo stadio; per questa missione decollerà dalla Nuova Zelanda verso un'orbita sincrona con il sole alta 520 chilometri. Il primo stadio si esaurisce dopo i primi 70 km e dopo due minuti e 32 secondi dall'inizio del volo, cade, seguendo un lungo arco che in passato l'avrebbe fatto precipitare nell'oceano, circa 280 km a valle. Ma Rocket Lab ha ora dotato il suo booster di schermatura termica, proteggendolo mentre ricade a terra a una velocità che può raggiungere 8.300 chilometri l'ora. Le temperature dell'involucro dovrebbero raggiungere i 2.400 °C poiché il booster viene rallentato dall'attrito con l'aria circostante, ma lo scudo termico appositamente progettato limiterà l'usura. A un'altitudine di 13 km, un piccolo paracadute drogue viene dispiegato dall'estremità superiore dello stadio del razzo, seguito da uno scivolo principale a circa 6 km, meno di un minuto dopo. Il paracadute rallenta notevolmente il razzo, tanto che la velocità di discesa scende a circa 36 km/h; a quel punto l'elicottero, con il gancio calato, andrà a catturare uno dei tiranti del paracadute per sostenere lo stadio e portarlo a destino. Per quanto appaia fantascientifico, il recupero a mezz'aria di razzi non è comunque una novità, perché già nel 1960, l'aeronautica militare americana si impadronì di una capsula di ritorno da una missione chiamata Discoverer 14. Ma ciò non aveva nulla a che fare con l'economia, perché i Discoverer erano in realtà satelliti da ricognizione del progetto CORONA e stavano inviando filmati dell'Unione Sovietica, inestimabile per l'intelligence della Guerra Fredda.

www.rocketlabusa.com



Il topo robot arriva dove gli altri si fermano!

I robot su gambe sono molto promettenti per l'uso in applicazioni del mondo reale, ma il loro funzionamento in spazi ristretti è ancora difficile. I robot quadrupedi di grandi dimensioni come BigDog e ANYmal hanno mostrato un'eccellente adattabilità e capacità di carico ma non possono entrare in spazi ristretti, mentre i robot micro quadrupedi come i robot sub-2g potrebbero entrare negli spazi ristretti ma non possono svolgere attività a causa della loro bassa capacità di carico. A questo problema ha dato risposta un team guidato dal Prof. Qing Shi del Beijing Institute of Technology, in Cina, sviluppando un ratto robotico chiamato SQuRo, capace di riprodurre i movimenti e comportamenti dei topi all'interno di spazi ristretti. In primo luogo ha ricavato le articolazioni di movimento chiave (KMJ) dei ratti e hanno completato la configurazione DOF (gradi di libertà). Nello specifico, hanno progettato 2 DOF in ciascun arto per riprodurre il movimento dell'arto, 2 DOF in vita e 2 DOF nella testa per replicare la flessibilità della colonna vertebrale. La gestione dei movimenti è affidata ad un controller gerarchico che opera su tre livelli: 1) pianificatore con quattro modalità di movimento di base per stabilire una relazione diretta tra la variabile di controllo e le forze di reazione al suolo (GRF); 2) ottimizzazione dei parametri con la considerazione dei limiti di stabilità e di attuazione; 3) generazione della traiettoria di ogni articolazione.

Nei test, SQuRo ha superato con successo uno stretto passaggio irregolare (larghezza interna di 90 mm), un ostacolo alto 30 mm e ha ottenuto una locomozione stabile su un pendio con un'inclinazione di 15°.

<https://smen.bit.edu.cn/english/>

Il robot pulisce gli specchi d'acqua

Oramai ci siamo abituati a vedere e sentire girare per casa e in ufficio i piccoli robot aspirapolvere e lavapavimenti, ma probabilmente non ci è mai venuto in mente che lo stesso lavoro potrebbe essere svolto nelle acque, per raccogliere rifiuti galleggianti che ogni giorno ci cadono dentro, magari solo perché fatti volare via dal vento o trasportati dalle piogge che li portano via...da terra. Insomma, si potrebbe immaginare di avere un "Room-ba" acquatico, che restituisca agli oceani, i fiumi, ai laghi e dei corsi d'acqua, un po' di pulizia? Come per molti dei nostri habitat, i corsi d'acqua del mondo sono diventati discariche per i nostri rifiuti. Raccogliere i rifiuti che sporcano queste aree è un lavoro impegnativo che non sempre è fattibile dall'uomo, ma ora la



società francese di robotica Interactive Autonomous Dynamic Systems (IADYS) ha introdotto Jellyfishbot, una macchina galleggiante che può funzionare autonomamente o sotto la direzione di un operatore remoto, va in giro raccogliendo spazzatura e spazzatura (come bottiglie di plastica, fuoriuscite di olio e alghe) che galleggiano sull'acqua, nonché detriti situati fino a 10 metri sotto la superficie. Jellyfishbot è un robot dotato di sensori che non solo gli consentono di navigare in autonomia, ma misurano anche la qualità dell'acqua in termini di salinità, temperatura, torbidità e proliferazione di organismi, inclusi cianobatteri e fitoplancton.

www.jellyfishbot.io

Droni per gestire l'eolico offshore

La maggiore continuità di produzione dell'eolico offshore viene parzialmente inficiata dai tempi e costi di installazione e manutenzione, più elevati. Una soluzione può essere rappresentata dal trasporto dei tecnici in aerotaxi e degli strumenti e materiali tramite un drone cargo; si tratta di una possibilità reale, anche perché le traiettorie di volo sono quasi esclusivamente su aree disabitate, quindi soggette a meno restrizioni rispetto alla mobilità aerea urbana, come nel caso della logistica su drone. L'utilizzo di droni cargo per trasportare strumenti e materiali direttamente in cima a una turbina eolica, che arriva ad altezze di 100 metri, eliminerebbe la necessità

di gru. Se i tecnici dell'assistenza dovessero viaggiare in aerotaxi, come vengono chiamati i droni passeggeri, non ci sarebbero bisogno di trasferimenti alle turbine e turni di due settimane con pernottamenti in mare, secondo EnBW, azienda energetica che dispone di impianti offshore. Il progetto potrà prendere piede una volta superati i vincoli normativi tedeschi e ottenuto il parere del DLR - Centro aerospaziale tedesco.

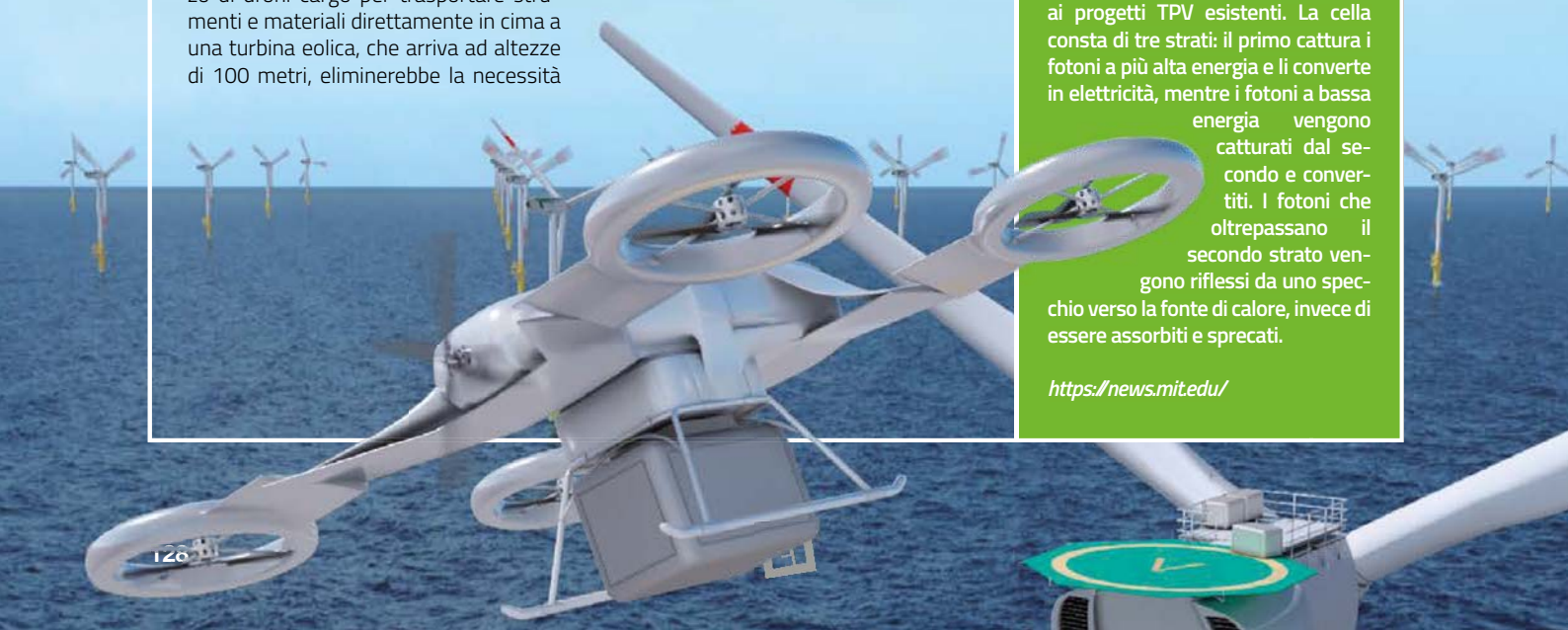
www.enbw.com



Record di elettricità dal calore allo stato solido

Gli ingegneri del MIT e del National Renewable Energy Laboratory hanno creato un generatore termoelettrico con un'efficienza superiore al 40%. Si tratta di una cella termofotovoltaica (TPV) che può generare elettricità da una fonte di calore compresa tra 1.900 e 2.400 gradi Celsius, o fino a circa 4.300 gradi Fahrenheit. I ricercatori hanno in programma di incorporare la cella TPV in una batteria termica che assorbirebbe l'energia in eccesso da fonti rinnovabili come il sole e immagazzinerebbe quell'energia in banchi fortemente isolati di grafite calda. Quando l'energia è necessaria, le celle TPV convertono il calore in elettricità e inviano l'energia a una rete elettrica. Ad oggi, la maggior parte delle celle TPV ha raggiunto efficienze solo di circa il 20 per cento, con il record del 32 per cento, poiché sono state realizzate con materiali a banda proibita relativamente bassa che convertono fotoni a bassa temperatura e bassa energia e quindi convertono l'energia in modo meno efficiente. La nuova cella utilizza materiali a banda proibita più elevata e giunzioni multiple, o strati di materiale, rispetto ai progetti TPV esistenti. La cella consta di tre strati: il primo cattura i fotoni a più alta energia e li converte in elettricità, mentre i fotoni a bassa energia vengono catturati dal secondo e convertiti. I fotoni che oltrepassano il secondo strato vengono riflessi da uno specchio verso la fonte di calore, invece di essere assorbiti e sprecati.

<https://news.mit.edu/>



Costruisci un vero robot!

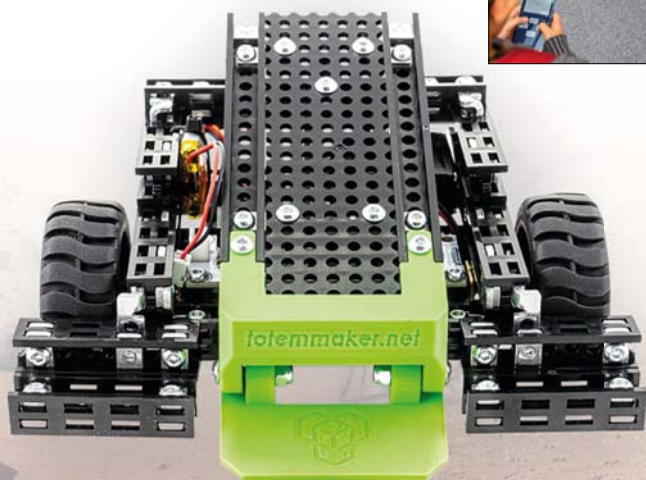


Mini robot da combattimento controllato da App

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

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.