

Elettronica In

www.elettronica.in.it

€ 6,90 - Anno XXVIII - n. 267 - SETTEMBRE 2022



Generatore di funzioni con AD9833

- Cubo LED con ARDUINO
- Breakout board ADXL372
- Blebricks a portata di app
- Monitor della qualità dell'aria
- Timer 555 Analyzer
- Smart Alarm con micro:bit

Demoboard ESP32



Sensore UV bluetooth

Blebricks



Contatore Geiger



€169,00

Cod. FR822

Misura l'intensità di raggi X, raggi gamma (γ) e raggi beta (β). Utilizza un microprocessore a 32 bit e dispone di ampio display LCD con retroilluminazione, tastiera a membrana, allarme acustico con soglia regolabile.



Air Quality

CONTROL

Purificatore d'aria da tavolo a raggi ultravioletti

MicroUVAIR è un sistema di sanificazione aria con ventilazione forzata a ciclo chiuso basato su led UVC. L'aria in ingresso viene purificata attraverso un filtro ad alta densità (30 PPI), quindi viene sterilizzata tramite luce ultravioletta emessa da LED UVC in grado di effettuare la massima azione germicida. Una volta purificata l'aria viene reimpressa nell'ambiente.

€199,00

Cod. MICROUVAIR-T



Prezzi IVA inclusa.

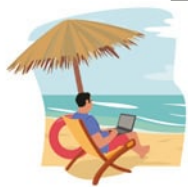
Misuratore di formaldeide e monitor della qualità dell'aria



Fornisce in tempo reale la quantità di formaldeide (HCHO) presente nell'ambiente circostante e monitora il livello di TVOC (Composti Organici Volatili Totali). È dotato di display per la lettura dei valori rilevati.

€21,90

Cod. FR832



Settembre: il mese della ripartenza

Dopo le meritate vacanze estive, eccoci alla ripartenza.

Ma parafrasando Sergio Marchionne, verrebbe da chiedersi: "ripartenza da cosa?".

Negli anni 80, agosto era il mese del: "tutto chiuso, ci rivediamo a settembre". Oggi, invece, e-commerce e servizi online sono inevitabilmente sempre operativi, costringendo così anche i negozi fisici a stare al passo: sempre più raramente le vetrine ospitano gli storici cartelli "Chiuso per Ferie" e sempre più spesso vengono rimodulate le ferie per garantire continuità

al servizio, nella speranza di poter fare due settimane di riposo e che gli impegni non ci seguano anche sotto l'ombrellone.

La vera ripartenza è sicuramente quella della scuola che, soprattutto per chi ha figli, segna la fine delle

La fine delle vacanze porta a confrontarsi con lo spettro del COVID19 che aleggia sugli istituti.

vacanze e che porta a confrontarsi con lo spettro del COVID19 che aleggia sugli istituti, dove ancora si parla di misure di contenimento soprattutto negli ambienti chiusi come le aule. Fortunatamente esistono soluzioni che ci avvertono quando è necessario aprire le finestre per limitare il rischio di contagio. Una ve la proponiamo noi ed è il misuratore di qualità dell'aria, che va a braccetto con un valido rilevatore di radiazioni solari UVA, utile per chi vuole avere una tintarella senza problemi e può godersi qualche giorno di vacanza anche a settembre. Purtroppo non abbiamo ancora inventato un "rilevatore di buona politica", altrimenti ve lo avremmo proposto prima del 25 settembre: andremo quindi alle urne sperando di non prenderci l'ennesima scottatura!

ON LINE

elettronica.in

SI RINGRAZIA

Futura Elettronica
Maker Faire
Microchip
Mouser



In copertina:
Una piccola breakout board basata sul chip AD9833 ci ha permesso di realizzare un generatore di funzioni veramente versatile.

Boris Landoni
 boris.landoni@elettronica.in

Elettronica In

Rivista mensile, anno XXVIII n. 267

SETTEMBRE 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://futanet.it/abbonamenti>

Distribuzione per l'Italia

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

via Bettola 18 - 20092 Cinisello Balsamo (MI)

Tel. 02-660301 Fax 02-66030320

Stampa

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

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 (per maggiori informazioni
scrivere a redazione@elettronica.in). Manoscritti, disegni e
foto non richiesti non verranno in alcun caso restituiti. L'utilizzo
dei progetti e dei programmi pubblicati non comporta alcuna
responsabilità da parte della Società Editrice.

© 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

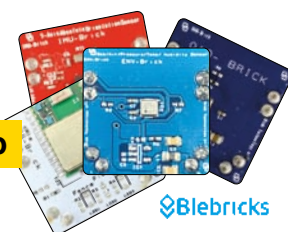
GENERATORE DI FUNZIONI CON AD9833

99

Networking

Blebricks,
a portata di app

91



Blebricks

19

ARTICOLI

GADGET

19 Cubo LED con Arduino

Gioco di luci tridimensionale capace di strabiliarvi con le combinazioni di cui è capace: economico e di facile realizzazione, grazie a un controller Arduino.

WIRELESS

27 Sensore UV Bluetooth

Realizziamo un rilevatore di radiazioni UV con allarme su smartphone tramite l'utilizzo di moduli Blebricks.

HOME PROJECT

33 Monitor qualità dell'aria

Visualizza la concentrazione di CO2 e TVOC nell'aria in un ambiente chiuso, come l'aula di una scuola o un ufficio e segnala tramite LED se la qualità dell'aria è buona o scadente.

Home Project Monitor qualità dell'aria



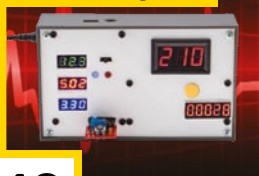
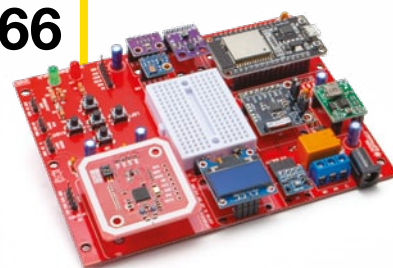
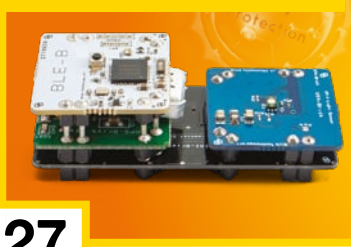
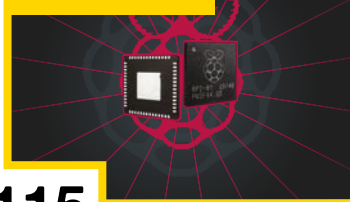
33

Gadget

Cubo LED con Arduino



19

Laboratorio**555 Analyzer****43****Top Project****Breakout board ADXL372****57****Didattica****Demoboard ESP32****66****Wireless****Sensore UV Bluetooth****27****Domotica****Smart Alarm con micro:bit****79****Didattica****Programmiamo con RP2040****115****LABORATORIO****43****555 Analyzer**

Usiamo l'Analizzatore del timer 555 per testare una serie di circuiti elettronici di uso comune. Seconda puntata.

TOP PROJECT**57****Breakout board ADXL372**

Scopriamo l'integrato ADXL372, un sistema completo di misurazione dell'accelerazione a 3 assi che opera a livelli di potenza molto bassi.

DIDATTICA**66****Demoboard ESP32**

Scopriamo i mille risvolti applicativi di uno dei più utilizzati moduli WiFi con una scheda di prototipazione che può ospitare e gestire sensori e attuatori sperimentabili con firmware di test da noi scritti appositamente. Prima puntata.

DOMOTICA**79****Smart Alarm con micro:bit**

Aggiungiamo funzionalità al progetto di domotica sviluppato con micro:bit aggiungendo la temporizzazione avanzata e le sveglie.

NETWORKING**91****Blebricks a portata di app**

Interagiamo tramite smartphone con i mattoncini per la prototipazione istantanea della serie Blebricks. Seconda e ultima puntata.

LABORATORIO**99****Generatore di funzioni con AD9833**

Sviluppa forme d'onda sinusoidale, triangolare e quadra a frequenza regolabile fra 1 Hz e 12 MHz.

DIDATTICA**115****Programmiamo con RP2040**

Continuiamo a sperimentare con semplici dispositivi didattici creati con la scheda FTR2040 e scriviamone il codice utilizzando l'ambiente MicroPython, C/C++ e quello Arduino.

Costruisci e programma un ROBOT!



cod. KSR21
€ 42,00

Robot RICCIO - in kit

Bellissimo robot riccio completo di tutte le parti meccaniche ed elettroniche. È ideale per avvicinare i ragazzi al mondo dell'elettronica e della meccanica. Il robot riccio è dotato di sensore sonoro che gli consente di interagire con il battito delle mani. In funzione del numero di battiti delle mani il robot riccio risponde con una sequenza di movimenti differente. Completo di manuale d'istruzioni in italiano per il montaggio organizzato in step e riccamente illustrato. Funziona con 4 batterie ministilo tipo AAA.



cod. KSR17
€ 38,00

Robot solare e idraulico 12 in 1 - in kit

Avvicinati al mondo dell'elettronica e della robotica imparando e divertendoti! Il Kit permette di realizzare fino a 12 differenti modelli con movimento idraulico e alimentati tramite energia solare. Completo di un manuale ricchissimo di illustrazioni per il montaggio passo passo di una scimmia, un cane e il suo padrone, uno pterosauro, un t-rex, un elefante, un coccodrillo, uno struzzo, uno scorpione, un carrello elevatore, una dune buggy, una scavatrice, un rover e per chi ha fantasia tantissimi altri modelli tutti da inventare.

Robot TOBBIE II con MICRO:BIT in kit

Robot a 6 gambe con corpo libero di ruotare, dotato di tutti i componenti elettronici necessari, tra cui un motore, un sensore a infrarossi e un cicalino per renderlo attivo e interattivo. Ideale in combinazione con Micro:bit. È possibile programmarlo in tutta semplicità con Javascript Blocks Editor o Python Editor sul tuo PC portatile o smartphone. Funziona con 4 batterie ministilo tipo AAA.



cod. KSR20
€ 89,00



Robot TOBBIE II in kit

Robot in kit a 6 gambe. Può funzionare in due differenti modalità: "Seguimi" e "Esplora". Nella modalità "Seguimi" Tobbie è in grado di seguire qualsiasi cosa gli si avvicini. Quando funziona in modalità "Esplora", grazie al rilevamento automatico tramite sensori infrarossi, è in grado di evitare gli ostacoli che trova davanti a sé trovando sempre un nuovo percorso e continuare a muoversi. Adatto dai 14 anni in su. Funziona con 4 batterie ministilo tipo AAA.

cod. KSR18
€ 37,00

Prezzi IVA inclusa

**FUTURA
ELETTRONICA®**

www.futurashop.it

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

Caratteristiche tecniche di questi prodotti
e acquisti on-line su www.futurashop.it

Quando il relé è bistabile

Più o meno tutti, per relé intendiamo il classico interruttore elettromeccanico azionato dal campo magnetico prodotto dalla sua bobina di eccitazione quando viene alimentata e che torna con lo scambio (il contatto) a riposo quando viene tolta alimentazione alla bobina stessa; in realtà sarebbe più corretto dire che questo è un tipo di relé, ossia quello più comune, ossia il relé monostabile, cosiddetto perché mantiene solo uno stato stabile: quello a riposo. Se è alimentato, passa nell'altra condizione, chiudendo il proprio scambio, però mantenendola solo finché riceve tensione sulla bobina. Oltre a questo esiste il relé bistabile, molto meno comune, la cui caratteristica è di riuscire a mantenere due stati stabili, ossia mantenere commutato lo scambio anche quando viene privato dell'alimentazione, grazie a una doppia bobina. Il suo funzionamento può essere così riassunto: quando si alimenta la sua bobina, lo scambio si sposta in una posizione e chiude il contatto comune sul normalmente aperto, e quando si toglie tensione lo scambio resta in questa posizione. Per farlo tornare nella condizione di riposo, ossia tornare a far chiudere il comune sul normalmente chiuso, bisogna applicare tensione alla bobina di rientro; dunque, una bobina produce il campo elettromagnetico che attiva lo scambio e la seconda produce il campo che lo riporta nella condizione iniziale. Le bobine si alimentano impulsivamente esattamente come quella di un relé monostabile e il loro estremo indipendente (hanno un polo in comune per semplificare il comando) per l'attivazione è spesso chiamato SET, mentre quello per il ritorno in posizione normale è contrassegnato RESET. Nel relé bistabile, un sistema di ritenuta di natura meccanica fa

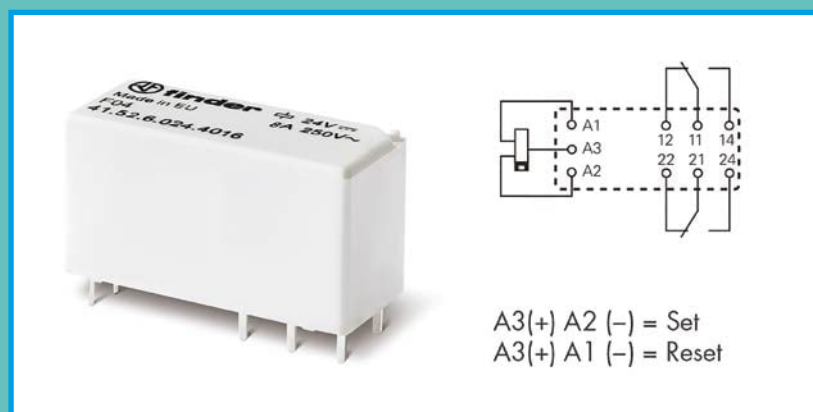
Tips & Tricks sono i nostri suggerimenti in pillole e i piccoli trucchi riguardanti sia l'hardware che il firmware, che sperimentatori, tecnici e progettisti elettronici vorrebbero conoscere per sviluppare meglio la propria passione: quella per l'Elettronica applicata. Ecco quelli che abbiamo preparato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

in modo che i contatti restino nella posizione di lavoro, vale a dire quella che implica il passaggio di corrente con il contatto chiuso, anche quando l'alimentazione viene tolta. Il relé bistabile trova applicazione in circuiti dove sia richiesta la funzione di invertire o deviare a permanenza, ma anche dove serva mantenere commutato un utilizzatore senza consumo di elettricità, se non nel momento in cui si deve fornire l'impulso di commutazione. Un tipico esempio sono i circuiti elettronici a bassissimo consumo e quelli alimentati da bus come nelle

unità a bus KNX, dove così è possibile tenerne commutati molti relé ma senza consumo, a parte negli istanti di commutazione, sul bus, il quale da protocollo può erogare 30 mA soltanto.

LIDAR e radar 4D

Il radar 4D è una tecnologia RF che utilizza l'ecolocalizzazione e un concetto chiamato "misurazione del tempo di volo" (ToF measurement) di raggi LASER per mappare oggetti in un ambiente tridimensionale. Attualmente tale tecnica



↑ Un relé bistabile Finder e la relativa piedinatura.

è in fase di test nel settore dei veicoli autonomi (per l'esattezza, nell'ADAS - Advanced Driver Assistance Systems) per mappare le posizioni degli elementi che vengono a trovarsi sul percorso di un veicolo e così riuscire a ricostruire l'ambiente in cui si muove, per evitare gli ostacoli e guidare la vettura. Non a caso viene anche chiamato 4D Imaging Radar.

Il radar 4D si distingue dalle tecnologie più datate come LIDAR, radar standard e telecamere, perché è in grado di determinare quando un veicolo è in movimento e quanto velocemente, in tutti i tipi di condizioni meteorologiche e ambientali.

Le fotocamere possono essere influenzate dalla luce solare intensa o dall'oscurità e possono avere problemi di profondità e contrasto. Il radar standard funziona bene in condizioni climatiche avverse, ma non è in grado di rilevare gli oggetti in modo molto dettagliato.

Il LIDAR è ideale per rilevare oggetti nello spazio ma è influenzato dalle condizioni meteorologiche avverse.

Il radar 4D risolve molti dei problemi di cui queste tecnologie soffrono. È una tecnologia relativamente nuova che utilizza un ampio array di canali a radiofrequenza (RF) per rilevare la velocità relativa, la distanza e l'azimut degli oggetti sulla carreggiata, nonché l'altezza degli oggetti sopra la strada. La quarta dimensione si riferisce al tempo, ma il radar 4D in realtà non mappa il tempo. Invece, usa il tempo nei suoi calcoli per determinare l'elevazione degli oggetti e la loro velocità relativa.

Per mappare l'ambiente intorno al veicolo ad alta risoluzione, un radar di imaging 4D utilizza un array di antenne

MIMO (Multiple Input Multiple Output).

Ciò può includere dozzine di antenne che trasmettono segnali a bersagli nell'ambiente circostante il dispositivo e quindi ricevono i segnali mentre rimbalzano sui bersagli. I dati ricevuti dalle antenne vengono quindi utilizzati per generare una nuvola di punti che rappresenta l'area circostante l'array. Un ampio array consente il rilevamento accurato di oggetti statici e dinamici contemporaneamente con dettagli elevati. Può anche catturare gli spostamenti doppler nell'ambiente e usarli per indicare la direzione in cui si sta muovendo un oggetto.

Poiché il radar 4D invia segnali fitti in tutte le direzioni, può funzionare all'interno dell'abitacolo di un veicolo classificando bambini e adulti, monitorando i segni vitali e rilevando la posizione degli occupanti. Questa funzione può essere utilizzata per ottimizzare l'attivazione degli airbag, ottimizzare il tenditore delle cinture di sicurezza, fornire avvisi anticipati sulle cinture di sicurezza e rilevare intrusi all'interno o intorno all'auto.

Il radar 4D funziona anche per applicazioni a corto, medio e lungo raggio perché ha una portata maggiore rispetto alle attuali tecnologie di visione artificiale. Un'altra caratteristica del radar 4D è la capacità di vedere attraverso gli oggetti perché non si basa su telecamere o ottiche.

Per riassumere, queste sono le peculiarità del sistema: i radar 4D utilizza un grande array di antenne MIMO (Multiple Input Multiple Output) per l'ecolocalizzazione. Invia segnali che rimbalzano sugli oggetti nell'ambiente e acquisisce i risultati per calcolare le dimensioni, la posizione, la direzione, la velocità e

l'elevazione degli oggetti nell'ambiente. Il radar 4D sostituirà, nell'ADAS, le vecchie tecnologie come telecamere, radar e LIDAR che vengono utilizzate per la visione artificiale e la guida autonoma nei veicoli.

Il radar 4D ha il vantaggio di poter funzionare con qualsiasi condizione atmosferica e qualsiasi livello di illuminazione, rilevare con precisione altitudine, velocità e direzione e rilevare bersagli dietro altri oggetti nell'ambiente.

Code formatter

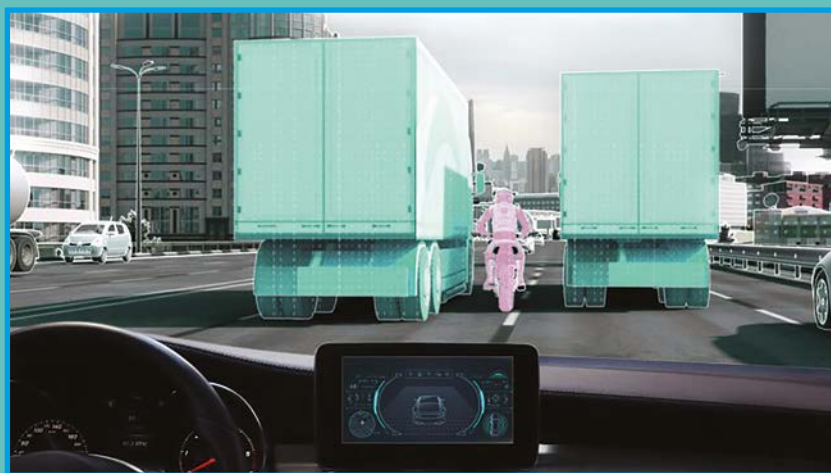
Scrivere il codice e farlo bene, è un lavoro difficile e ripetitivo, tant'è che talvolta sbagliamo indentazione, ripetiamo istruzioni o commettiamo quegli errori che si commettono quando programiamo con più linguaggi. Per aiutare il programmatore esistono tool specifici del sistema di sviluppo, ma anche servizi sul web tra i quali codebeautify.org ha attirato la nostra attenzione perché rende disponibili ottimi strumenti tra cui il Code Formatter, che vi permette di indentare correttamente del codice anche semplicemente copiato da un testo; il Code formatter di questa web app supporta C, C++, HTML, Java ecc. ma, cosa molto utile, Python, che è un linguaggio in cui è tutto basato sulle tabulazioni e indentazioni.

È anche possibile scaricare il codice riformattato (con il pulsante Download) ed anche caricare un file contenente del codice, ovvero importarlo specificando un URL dove si trova (pulsante URL). Lo strumento permette anche di ottimizzare e abbellire il codice scritto, per esempio colorando alcuni elementi come si fa negli editor i Python o nell'IDE Arduino, per renderlo di più semplice lettura.

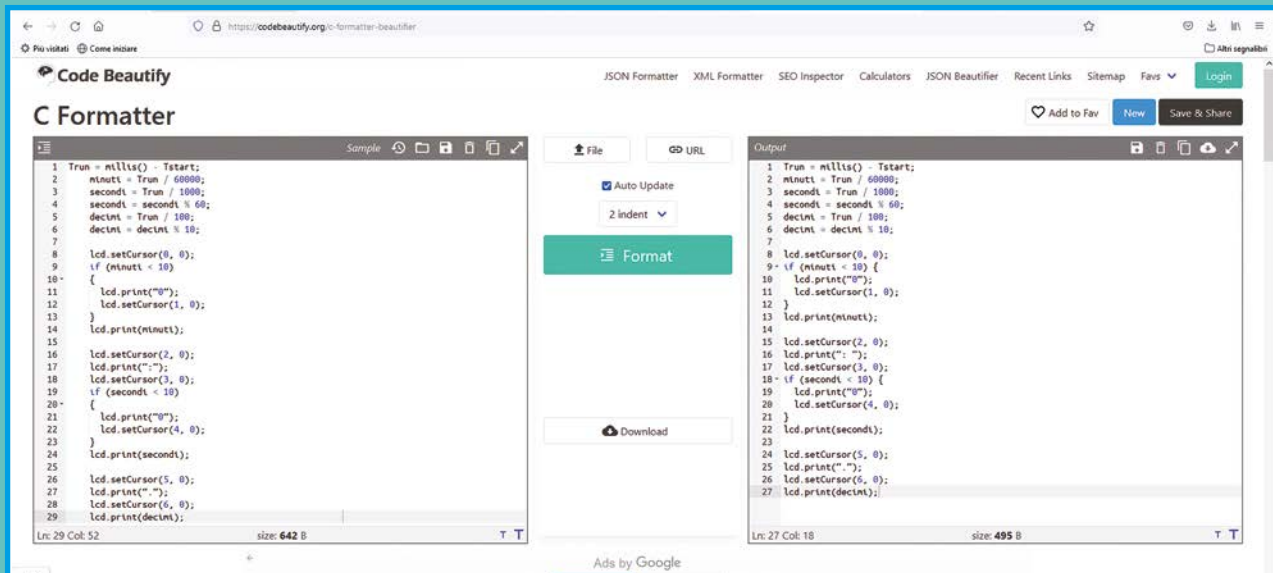
Dalla pagina web principale di Code Beautify si definisce innanzitutto il linguaggio cui si vogliono applicare le funzionalità della web app, poi, fatto questo, la pagina assume l'aspetto corrispondente e in alto a destra viene proposta la toolbar che raggruppa le funzioni disponibili.

IC regolatori buck con condensatori integrati

Creare alimentatori efficienti e compatti in grado di rispondere ai requisiti EMI



↑ Immagine della strada rilevabile da un radar 4D per ADAS.



↑ Funzione di formattazione del codice in C del sito codebeautify.org.

come quelli imposti dal CISPR, impone una serie di criticità; noti per l'elevata efficienza e le buone prestazioni termiche, i regolatori buck non sono generalmente considerati una buona soluzione per dispositivi con basso valore di EMI. Fortunatamente, esistono diverse opzioni per ridurre le emissioni elettromagnetiche generate da tali regolatori e per comprenderle consideriamo lo schema semplificato di un regolatore buck, dove troviamo un condensatore all'ingresso (Vin) ed uno di boot.

Oltre a selezionare i valori corretti dei componenti per garantire un design funzionale, il layout del PCB dovrebbe tenere conto di due regole generali:

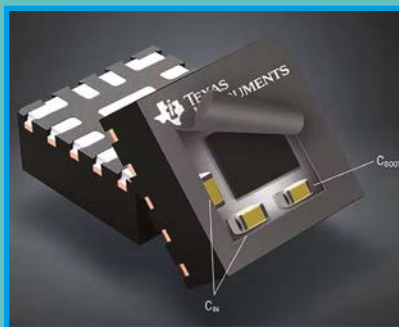
1. ridurre al minimo le aree di loop di corrente transitoria (di/dt) portando il condensatore di ingresso e il condensatore di avvio (boot capacitor) il più vicino possibile ai pin VIN e GND del circuito integrato;

2. ridurre al minimo la superficie dei nodi ad alta tensione transitoria (dv/dt) minimizzando l'area del nodo switch.

Nei casi in cui l'ottimizzazione del layout della scheda non sia possibile, sono disponibili altre opzioni, come quella proposta da Texas Instruments, che ha creato IC controller per regolatori buck con integrati i condensatori d'ingresso. Come accennato, la riduzione dell'area dei loop di corrente di/dt elevati è molto importante quando si progettano regolatori a commutazione per rimanere al di sotto dei limiti EMI. In un regolatore buck, è importante considerare il circuito della tensione di ingresso verso terra dal punto di vista EMI. Tale regolatore opera attivando e disattivando la connessione all'alimentazione, determinando la corrente del MOSFET sul lato superiore (high-side); la rapida commutazione crea correnti molto alte, quasi discontinue, erogate dal condensatore di ingresso. Alcuni dispositivi, come i regolatori buck 3-A LMQ66430-Q1 e 6-A LMQ61460-Q1 36-V di TI, integrano condensatori di ingresso per alta frequenza all'interno del contenitore, ottenendo così la più piccola area possibile di loop di corrente in ingresso. Riducendo l'area di questo loop di corrente in ingresso si ottiene una minore induttanza parassita, riducendo significativamente la quantità di energia elettromagnetica generata.

Un altro circuito con elevata corrente di/dt da considerare è il circuito del condensatore di bootstrap.

Quest'ultimo è responsabile della fornitura di carica al gate driver MOSFET high-side durante il periodo di attivazione. I circuiti interni scaricano questo condensatore durante il tempo di inattività. Il terminale di source del MOSFET high-side si collega al nodo switch anziché a GND. Il riferimento del condensatore di bootstrap al pin source del MOSFET garantisce che la tensione gate-source (VGS) sia sufficientemente alta da attivare il MOSFET. Con la maggior parte dei regolatori buck, bisognerà lasciare un'area del nodo switch disponibile sulla scheda per collegare il condensatore di bootstrap, sebbene ciò possa essere controproducente quando si tenta di ridurre al minimo l'area del nodo switch per ridurre le EMI. Integrando il condensatore di avvio all'interno del package, l'LMQ66430-Q1 della Texas Instruments ci fa rispettare le due regole menzionate in precedenza.



↑ Controller buck TI con condensatori di boot e ingresso integrati.

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



ROME

Maker Faire

THE EUROPEAN EDITION

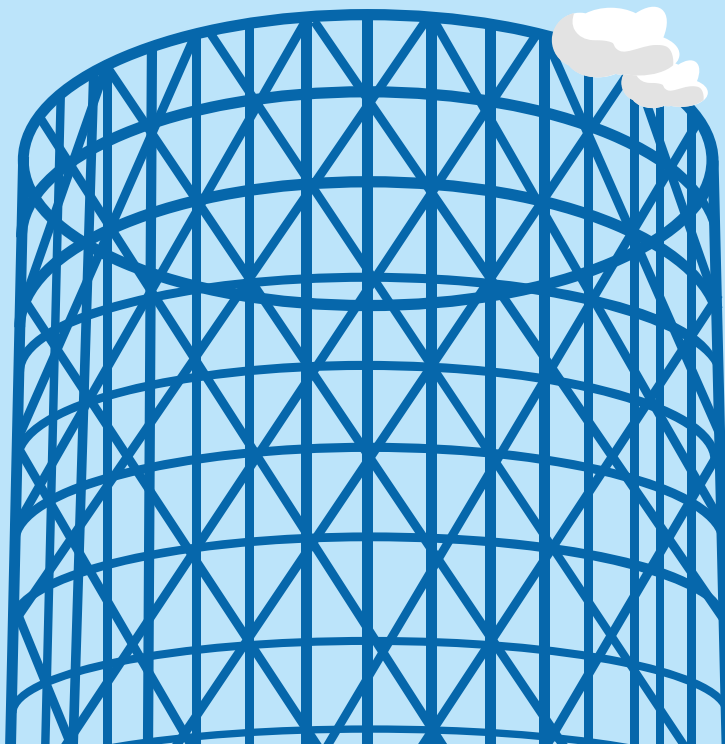
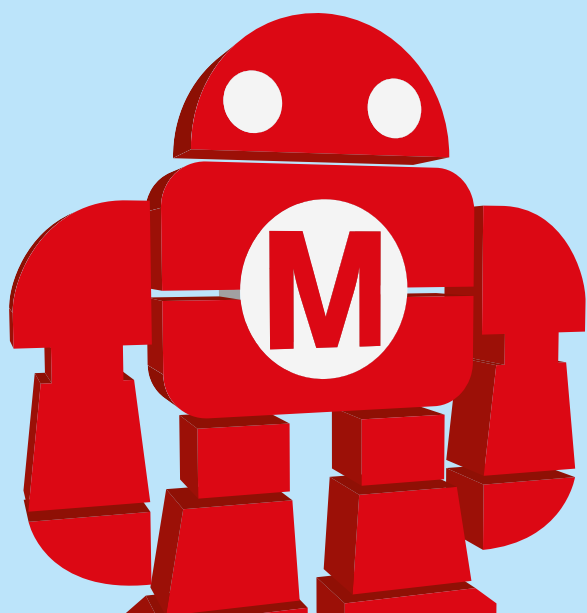
Powered by



Camera di Commercio
Roma



#MFR2022



7-9
OTTOBRE
2022

Scopri.INNOVA.Crea.
Gazometro Ostiense

Info e biglietti makerfairerome.eu

Promosso
e organizzato da



Con il supporto di



Main Partner



Gold Partner

sanofi



Five Years Out



Silver Partner



Technical Partner



I ricercatori dell'Indian Institute of Technology di Bombay hanno annunciato un nuovo chip AI a bassa potenza, che abilita quelle che vengono chiamate reti neurali spiking o SNN. Il cervello ha 100 miliardi di neuroni, ognuno dei quali è collegato ad altri 10.000 neuroni tramite sinapsi e che rappresentano informazioni attraverso schemi coordinati di picchi elettrici. Imitare questi neuroni utilizzando un dispositivo compatto e in modo efficiente dal punto di vista energetico, è stato ciò che i ricercatori hanno ottenuto. Proprio come i neuroni nel cervello raggiungono un picco a una determinata soglia di energia, gli SNN si basano su una rete di neuroni artificiali in cui una fonte di corrente carica un condensatore che si scarica finché non viene raggiunto un livello di soglia cui i neuroni artificiali si attivano e la carica

» Un chip AI low power per SNN



immagazzinata viene ripristinata a zero. Molte SNN si basano su transistor ad elevate correnti, il che porta a un elevato con-

sumo o all'attivazione troppo rapida dei neuroni. I ricercatori hanno risolto il problema con una nuova fonte di corrente per caricare i condensatori, chiamata corrente di tunneling da banda a banda (BTBT) dove la corrente può fluire attraverso il gap del silicio del neurone artificiale.

L'approccio limita anche la capacità dei condensatori, che possono essere integrati nei chip di silicio a 45 nm. La SNN è stata applicata a un modello di riconoscimento vocale ispirato alla corteccia uditiva del cervello: utilizzando 20 neuroni artificiali per la codifica dell'input iniziale e 36 neuroni aggiuntivi, il modello potrebbe riconoscere efficacemente le parole, dimostrando la fattibilità dell'approccio nel mondo reale.

www.iitb.ac.in/en/

» Siemens accelera la Digital Transformation

Nasce Siemens Xcelerator, la piattaforma open di business digitale, per accelerare la trasformazione digitale nell'industria, negli edifici, nelle reti e nella mobilità. Siemens Xcelerator comprende hardware, software e servizi digitali abilitati all'IoT. Tra questi, le soluzioni Brightly Software accelereranno la crescita degli Smart Building. Determinante è la partnership con NVIDIA per lo sviluppo immersivo del gemello digitale. Combinando il mondo reale e quello digitale, Xcelerator consentirà di aumentare la produttività, la competitività e il livello di innovazione. Con Xcelerator, Siemens trasformerà l'intero portafoglio di hardware e software per renderlo modulare, connesso al cloud e basato su API (Application Programming Interface) standard. Siemens ha annunciato anche la prima offerta SaaS, parte di Siemens Xcelerator. Building X è una nuova suite per Smart Building che elimina la complessità dalla digitalizzazione. È una suite di dati e analisi end-to-end che



abbatte i silos di dati in domini come la gestione dell'energia, la sicurezza e la manutenzione degli edifici. Building X è una suite software aperta modulare e completamente basata su cloud, con applicazioni abilitate all'intelligenza artificiale, connettività avanzata e sicurezza informatica integrata. Siemens prevede inoltre di integrare le sue soluzioni Industrial IoT come Industrial Operations X, che riunisce solu-

zioni e applicazioni dal sensore all'edge al cloud, IoT as-a-service e low cost capacità di sviluppo, un'ampia gamma di app pronte per l'uso, oltre a consentire la fusione di dati dal mondo reale dell'automazione con il mondo digitale della tecnologia dell'informazione, arricchito dalla conoscenza e dalle capacità di integrazione verticale IT/OT di Siemens.

<https://press.siemens.com>

› AMD potenzia l'elaborazione AI in real-time

La serie Versal AI Core, con tecnologia AMD AI Engine, è stata selezionata da Canon per il suo sistema video Free Viewpoint che dovrebbe rivoluzionare l'esperienza visiva per le trasmissioni sportive in diretta e i webcast. I versatili dispositivi AI Core offrono una potente elaborazione video basata su machine learning (ML) all'edge per i sistemi di fotocamere Canon. Il sistema video Canon Free Viewpoint è costituito da un anello di telecamere ad alta risoluzione che circondano uno stadio o un'arena per consentire agli spettatori della trasmissione di vedere l'azione sul campo da qualsiasi posizione o angolazione dello stadio. La tecnologia rivoluzionaria cambia radicalmente il modo in cui gli sport vengono visti, metten-

do gli spettatori al centro dell'azione. Il sistema Canon Free Viewpoint dovrebbe essere distribuito in più arene di basket professionistiche negli Stati Uniti e in altri importanti impianti sportivi in tutto il mondo.

L'elaborazione ad alte prestazioni e bassissima latenza abilitata dalla serie Versal AI Core riduce significativamente i tempi di elaborazione delle immagini, consentendo la produzione di replay dal vivo quasi in tempo reale, rispetto ai diversi minuti imposti dalle architetture tradizionali. La tecnologia fornirà agli spettatori di tutto il mondo immagini uniche che non sono mai state sperimentate prima. Le reti di trasmissione dal vivo possono anche monetizzare le metriche degli eventi

› Sensori intelligenti per IoT a connessione rapida



Renesas rende disponibili nuovi sensori e circuiti integrati per il condizionamento del segnale a complemento della linea di MCU, firmware avanzato, intelligenza artificiale integrata e un'esclusiva piattaforma di progettazione IoT a connessione rapida. Secondo un recente studio di Zion Market Research, si stima che il mercato globale dei sensori IoT cresca a un tasso di crescita annuale composto (CAGR) di circa il 27,9%, raggiungendo una cifra stimata di 27,9 miliardi di dollari entro il 2028. Ciò è un potente motore per lo sviluppo di soluzioni IoT.

Oltre alla nuova famiglia HS4XXX di

sensori di umidità relativa e temperatura, Renesas propone il circuito integrato di condizionamento del segnale del sensore (SSC) ZSSC3281 per l'amplificazione, la digitalizzazione e la correzione specifica del sensore dei segnali del sensore ad alta precisione. Questi nuovi prodotti completano l'ampio portafoglio MCU di Renesas e le soluzioni di intelligenza artificiale integrate.

Il processo di progettazione dei sensori è supportato dall'esclusiva piattaforma di progettazione dei sistemi di Renesas che semplifica notevolmente la prototipazione dei sistemi IoT. Il sistema Renesas Quick-Connect IoT è costituito da

schede e interfacce standardizzate, che consentono ai progettisti di collegare rapidamente e facilmente un'ampia gamma di sensori alle schede di sviluppo MCU/MPU. Il nuovo sistema fornisce anche blocchi di base del software portabili tra le schede, riducendo notevolmente i requisiti di codifica. Ora, invece di scrivere e testare centinaia di righe di codice del driver, i progettisti devono solo selezionare graficamente il loro sensore e scrivere poche righe di codice. La nuova famiglia HS4XXX di sensori di umidità relativa e temperatura offre elevata precisione, tempo di risposta di misurazione rapido e consumo energetico estremamente basso in un package di piccole dimensioni, rendendoli ideali per un'ampia gamma di applicazioni che vanno dai dispositivi portatili ai prodotti progettati per ambienti difficili. Il nuovo ZSSC3281 SSC offre amplificazione, digitalizzazione e correzione specifica del sensore dei segnali del sensore estremamente precisi. Include un front-end analogico a due canali e un MCU integrato basato su ARM con matematica incorporata per l'elaborazione del segnale del sensore. Si rivolge ad applicazioni di fascia alta, tra cui il rilevamento industriale, medico e dei consumatori.

www.renesas.com

dal vivo come le statistiche di gioco e giocatore. La tecnologia Canon sposterà l'intelligenza dell'elaborazione video basata sull'intelligenza artificiale all'edge, riducendo così il carico di lavoro nel data center. Versal AI Core alimenterà il sistema di elaborazione delle immagini che si trova dietro ciascuna delle telecamere Canon distribuite in ciascuna sede e implementa algoritmi proprietari Canon per il rilevamento delle immagini, la segmentazione e altro utilizzando il Machine Learning dal lato dell'edge computing.

L'implementazione dell'apprendimento automatico all'edge durante una trasmissione in diretta è un'entusiasmante novità nel settore, perché il sistema di fotocamere Canon calerà davvero lo spettatore nell'evento sportivo, come mai prima d'ora. Il cuore dei sistemi all'avanguardia di Canon è la serie Versal AI Core che offre l'elaborazione edge a bassa latenza che è la



chiave per annunciare questa nuova era nelle trasmissioni sportive in diretta.

www.amd.com

› NXP aiuta a standardizzare la sicurezza di nuova generazione con la crittografia post-quantistica

NXP Semiconductors annuncia che un algoritmo di sicurezza specializzato, di cui sono co-autori i suoi esperti di sicurezza, è stato selezionato dal NIST per entrare a far parte di uno standard globale del settore, progettato per contrastare le minacce informatiche nell'ambito del calcolo quantistico. Un secondo algoritmo di NXP passerà anche al quarto e ultimo round per ulteriori analisi e considerazioni per la standardizzazione. Man mano che i rischi cui i computer quantistici sono esposti diventano più chiari, questo

sforzo anticipa la necessità di proteggere i dati crittografati e i dispositivi connessi. Gli algoritmi di crittografia post-quantistica (PQC) selezionati verranno utilizzati per sviluppare un nuovo standard di crittografia a chiave pubblica sicuro contro i computer tradizionali e quantistici.

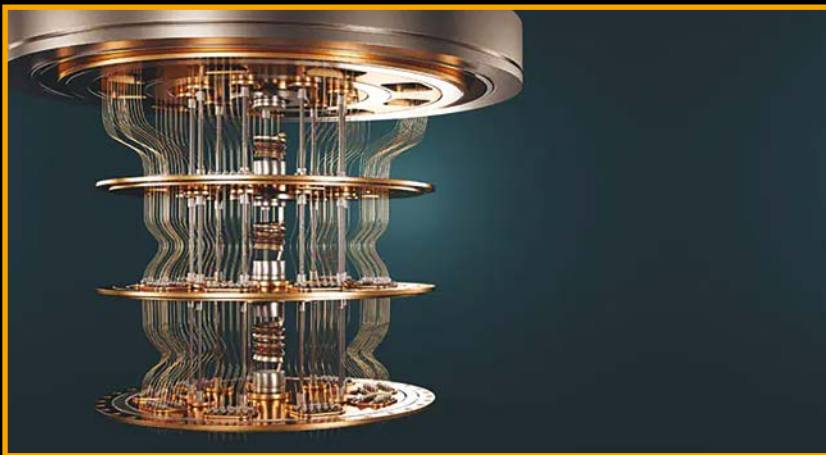
Molti esperti di sicurezza informatica ritengono che quando i computer quantistici saranno realizzati e opereranno su larga scala, l'assoluta potenza di calcolo di queste macchine sarà in grado di "risolvere" le sfide della

crittografia in una frazione di tempo, rompendo gli odierni sistemi di crittografia a chiave pubblica e lasciando i dati digitali, firme e dispositivi vulnerabili. Ma ciò determina sostanziali rischi per la sicurezza dei dispositivi e dei dati online, comprese le transazioni finanziarie, le infrastrutture critiche, i meccanismi di aggiornamento via etere (OTA) e altro ancora.

Per combattere questo, il NIST punta a standardizzare gli algoritmi PQC che consentirebbero all'industria di passare a nuovi sistemi sicuri prima della minaccia quantistica. L'algoritmo di crittografia basato su reticolo Crystals-Kyber, presentato da NXP con gli esperti di sicurezza di IBM, fungerà da base per questo nuovo standard. The Classic McEliece, un altro contributo del co-autore NXP che appartiene alla famiglia della crittografia basata su codice, avanza a un ulteriore ciclo di analisi e considerazione per la standardizzazione.

Il Kyber, non solo è più veloce degli standard attuali, ma fornisce una forte sicurezza per proteggere i sistemi e i dati anche nelle applicazioni di calcolo quantistico.

www.nxp.com



Costruisci un vero robot!

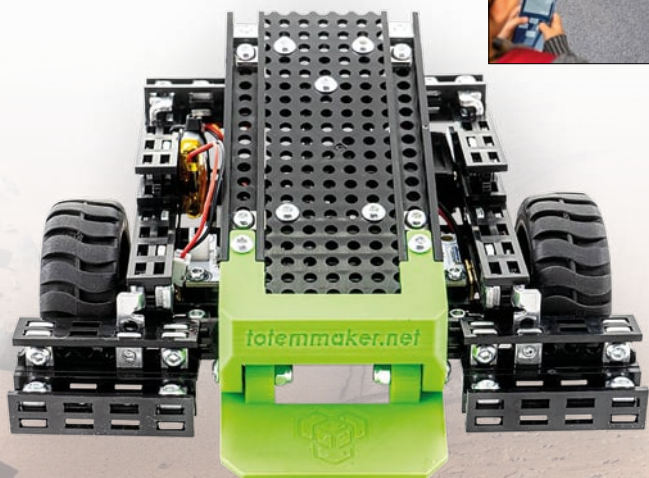


Mini robot da combattimento controllato da App

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®**

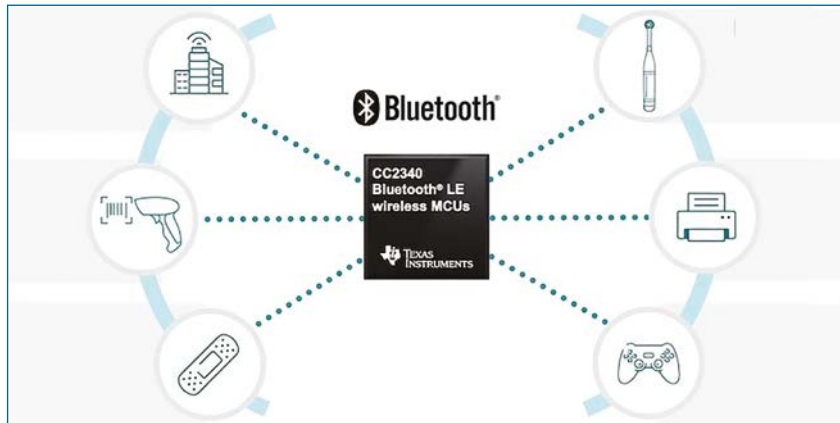
Futura Group srl
Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775

Caratteristiche tecniche
e vendita on-line su:

www.futurashop.it

MCU Bluetooth LE low-cost da Texas Instruments

Texas Instruments amplia il proprio portafoglio di connettività con una nuova famiglia di microcontrollori con Bluetooth Low Energy di alta qualità a basso costo, composta dagli MCU CC2340R2 e CC2340R5, che offrono una memoria flash rispettivamente di 256 kB e 512 kB, un'eccezionale flessibilità e ampio spazio per il codice dell'applicazione, oltre a 36 kB di RAM con supporto per il download over-the-air. Altra caratteristica è la corrente di standby leader del settore, inferiore a 830 nA, che porta l'autonomia fino a 10 anni con una pila a bottone in applicazioni wireless come tag per scaffali e sistemi di monitoraggio della pressione dei pneumatici. La famiglia CC2340 ha un intervallo di temperatura di esercizio da -40°C a 125°C per garantire una connessione stabile tra le applicazioni, dai sensori industriali e laboratori medici agli ambienti esterni come caricabatterie per



veicoli elettrici o contatori intelligenti. La potenza in trasmissione è +8 dBm e i dispositivi CC2340 sono dotati di un balun RF integrato per semplificare la progettazione e ridurre i costi. Gli MCU wireless CC2340R2 e CC2340R5 sono l'ultima aggiunta al portafoglio Simple-

Link, rivolto ai mercati dell'elettronica industriale, automobilistica e personale. Per iniziare con gli MCU wireless CC2340, Texas Instruments mette a disposizione il kit di sviluppo LP-EM-CC2340R5, al prezzo di 39 USD.

www.ti.com/wireless

Controller high-side switch automotive da STMicroelectronics

Il controller dell'interruttore high-side per autoveicoli VNF1048F di STM combina funzioni diagnostiche e di protezione del sistema avanzate con la tecnologia con fusibili al silicio I2-t di ST. Primo nella nuova famiglia STi2Fuse di ST con protezione I2-t integrata, il VNF1048F è compatibile con i sottosistemi di alimentazione per autoveicoli a 12V, 24V e 48V (reti di bordo). Il VNF1048F dispone di un'interfaccia SPI verso un microcontrollore host, per configurare e monitorare la protezione del sistema e le funzionalità diagnostiche, le quali includono la protezione da sottotensione (UVLO) e sovratemperatura, protezione termica per il MOSFET high-side esterno, spegnimento per desaturazione e impostazioni del fusibile I2-t. È previsto un rilevamento della corren-



te digitale ad alta precisione. Sono integrati anche convertitori analogico-digitali per il monitoraggio della tensione del termistore di rilevamento della temperatura, della tensione di uscita e della tensione drain-source del MOSFET. I fusibili al silicio aiutano a semplificare i cablaggi del veicolo e a ridurre il peso, eliminando le scatole dei fusibili accessibili all'utente e i relativi cablaggi. Il VNF1048FTR è incapsulato in package QFN 32L 5x5 mm a 32 contatti, ad un prezzo di 1,77\$ per 1.000 pezzi.

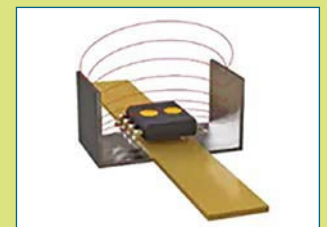
www.st.com

Melexis attiva un simulatore gratuito online per sensori di corrente

Nasce il Current Sensor Simulator di Melexis, che semplifica la progettazione e l'implementazione di soluzioni di rilevamento della corrente tramite la tecnologia IMC-Hall: è disponibile on-line e il suo utilizzo è gratuito. Il Current Sensor Simulator è un'applicazione web facile da usare per selezionare rapidamente i sensori IMC-Hall. Questo strumento interattivo accelera notevolmente il processo di progettazione, consentendo anche a chi ha limitata esperienza di rilevamento magnetico di trarre vantaggio da tale tecnologia. Compilando i parametri sull'interfaccia web e inserendo i loro requisiti di progettazione specifici, lo strumento raccomanda il codice di parte appropriato, nonché il relativo concentratore ferromagnetico (schermo). Lo strumento fornisce

anche un budget di errore totale, in modo che le prestazioni possano essere stimate istantaneamente nell'intero intervallo di rilevamento della temperatura e della corrente. I sensori di corrente in CI IMC-Hall forniscono misure di corrente accurate e senza contatto, ideali per applicazioni automobilistiche e industriali; non soffre dei limiti delle tecnologie coreless e shieldless, per loro natura sensibili alle tolleranze meccaniche e per le quali l'IC va posizionato molto vicino alla sbarra.

www.melexis.com



I progetti di riferimento per azionamenti motore di STM STSPIN32



STM ha rilasciato due progetti di riferimento che semplificano la realizzazione di azionamenti per compressori basati sul controller motore STSPIN32. Ciascuno integra il controller del motore con un inverter trifase per alimentare il motore e un convertitore offline e circuiti ausiliari, oltre a un

designer di PCB e un firmware di controllo motore. Il progetto STEVAL-CTM011V1 si rivolge a compressori industriali generici fino a 250 W, mentre lo STEVAL-CTM012V1 soddisfa le normative di progettazione ecocompatibile per gli elettrodomestici. Entrambi garantiscono un'efficienza media

superiore al 96,5%. Il controller STSPIN32F0601Q contiene un microcontrollore STM32F031 Arm Cortex-M0 e un gate-driver trifase da 600 V molto robusto per l'inverter. Il firmware fornito include il controllo ad orientamento di campo (FOC) sensorless ad alta efficienza. www.st.com

Nuovi controller touch maXTouch KoD da Microchip

Microchip annuncia i controller touchscreen maXTouch Knob-on-Display (KoD), la prima famiglia di livello automotive a supportare in modo nativo il rilevamento e il reporting di encoder rotativi capacitivi. Diversamente dai tradizionali encoder rotativi meccanici, questa tecnologia consente il montaggio della manopola direttamente sul display senza aperture nel pannello o personalizzazione del pattern tattile, aumentando la flessibilità progettuale e il risparmio sui costi. La tecnologia KoD non richiede sensori tattili personalizzati, offrendo ai progettisti la possibilità di implementare facilmente un numero, una forma e una posizione delle manopole diversi per adattarsi a una varietà di prodotti. Le configurazioni personalizzate sono regolabili senza modificare il firmware integrato del controller touch maXTouch KoD, portando a un ciclo di sviluppo più rapido e flessibile. Un ulteriore vantaggio è la facilità di creazione di un modulo HMI (Human-Machine In-



terface) sigillato, sempre più diffuso negli elettrodomestici e nelle applicazioni industriali. Microchip offre due varianti: la KD consente di scegliere il partner per la progettazione della manopola e creare la propria soluzione utilizzando un progetto di riferimento sviluppato in collaborazione con BNL Bearings. La variante MK aggiuntiva supporta la Magic

Knob sviluppata da Panasonic Industry Co., Ltd. Pur mantenendo la flessibilità per essere montata su modelli di sensori tattili standard.

Microchip fornisce una suite completa di strumenti di sviluppo hardware e software per accelerare lo sviluppo con l'encoder capacitivo.

www.microchip.com

Allegro MicroSystems annuncia rivoluzionari sensori di posizione per ADAS

Allegro MicroSystems propone i sensori magnetici di posizione A33110 e A33115, rivolti ad applicazioni ADAS che richiedono elevati livelli di precisione e ridondanza del segnale eterogenea, i nuovi sensori combinano la tecnologia Hall verticale (VHT) dell'azienda con la tecnologia all'avanguardia della magne-

toresistenza a tunnel (TMR) in un unico package. La tecnologia TMR su silicio offre risoluzione e precisione migliorate, fornendo una sensibilità fino a 8 volte maggiore. Questi sensori sono conformi ASIL D, con ridondanza eterogenea che riduce la probabilità di guasti. Gli algoritmi avanzati nell'A33110 e nell'A33115 of-

frono il tempo di risposta rapido, la correzione indipendente del guadagno/offset, il calcolo dell'angolo e le capacità di linearizzazione richieste dalle applicazioni ADAS. L'A33115 include anche un contagiri che tiene traccia del movimento con incrementi di 90° e una modalità a basso consumo.

www.allegromicro.com



Nordic: il futuro del suono wireless

LE Audio è la prossima generazione di streaming audio Bluetooth, il cui core è un codec chiamato Low Complexity Communication Codec (o LC3) che supporta streaming di qualità superiore e consente al contempo un consumo di energia (sempre rilevante, nello streaming wireless) inferiore rispetto alle soluzioni audio Bluetooth esistenti.

Per consentire ai progettisti di sviluppare applicazioni LE Audio, Nordic propone il suo kit di sviluppo audio nRF5340 per lo sviluppo rapido di prodotti LE Audio. L'Audio Development Kit (DK) della Nordic si basa sul SoC nRF5340 e contiene tutto il ne-

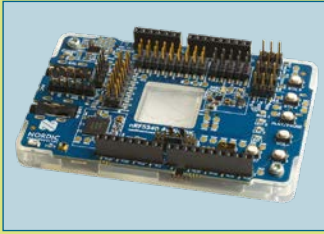
cessario per iniziare i progetti di sviluppo LE Audio.

Il SoC nRF5340 è uno dei dispositivi Bluetooth LE di Nordic più prestanti e su di esso si basa anche il Nordic Thingy:53, che incorpora anche il PMIC nPM1100, il modulo FEM (FEM

nRF21540, un amplificatore di potenza/amplificatore a basso rumore (PA/LNA) e sensori di movimento multipli (due

accelerometri, giroscopio e magnetometro) sensori ambientali (temperatura, umidità, pressione, gas), nonché sensori di luce e colore e un microfono digitale, tutti utilizzabili in applicazioni ML integrate.

www.nordicsemi.com



Controller 10BASE-T1S da ONSEMI

ONSEMI annuncia il controller Ethernet 10BASE-T1S NCN26010, progettato per fornire comunicazione multipoint affidabile in ambienti industriali, supportando fino a 40 nodi su un singolo doppino intrecciato: oltre il quintuplo dei nodi previsti dallo standard IEEE 802.3cg. Il nuovo controller implementa l'Ethernet 10BASE-T1S multi-drop, che sostituisce gli standard di comunicazione industriale punto-punto e multipunto legacy. Il T1S MAC/PHY può sostituire RS-485, CAN, RS-232, HART e altri. Di conseguenza, si ottiene una maggiore velocità di trasmissione dei dati rispetto al cablaggio esistente, eliminando la necessità di tirare nuovamente i cavi, che spesso è la spesa maggiore in un'installazione di rete.



L'NCN26010 ha una capacità inferiore sui pin di linea rispetto alle soluzioni T1S esistenti; è il primo controller PHY + MAC del settore per 10BASE-T1S e può connettersi a controller, sensori e altri dispositivi senza la necessità di un MAC esterno. L'NCN26010 e l'hardware di sviluppo di supporto sono disponibili tramite il supporto Onsemi.

www.onsemi.com

EarthCam: il primo sistema di telecamere 5G multinet

Sierra Wireless annuncia che EarthCam, fornitore di contenuti, tecnologia e servizi per webcam, ha scelto il suo router 5G AirLink XR80 per il proprio primo sistema di telecamere 5G multi-rete, StreamCam 5G.

AirLink XR80 5G fornisce a StreamCam 5G di EarthCam la connettività 5G veloce e affidabile, a bassa latenza, necessaria per inviare una quantità quasi illimitata di dati visivi da telecamere di streaming live ad alta risoluzione, consentendo agli utenti di tenersi aggiornati facilmente lo stato di avanzamento di un progetto o condividere la loro posizione con tempi di trasferimento file notevolmente migliorati. È possibile accedere rapidamente a grandi quantità di dati provenienti da sensori ambientali, di sicurezza e di protezione e la fotocamera dispone anche di funzionalità di intelligenza artificiale. Annunciato per la prima volta nell'ottobre del 2020, il sistema di telecamere multi-rete 5G di EarthCam offre streaming 4K, fotografia da 12 megapixel e funzionalità time-lapse in-

tegrate e più che una video-camera è un hub di comunicazione per telemetria, che consente l'invio di grandi quantità di dati ambientali, di sicurezza con bassa latenza.

I router AirLink di Sierra Wireless, inclusi XR80 5G e RV55 LTE-A Pro, sono essenziali per le telecamere per tutte le stagioni di EarthCam per la documentazione di progetti pluriennali, come i video time-lapse dello stadio più grande della NFL, Los Angeles SoFi, Denver Aeroporto Internazionale e la ricostruzione del sito del World Trade Center a Manhattan. La sua connettività globale senza interruzioni è parte integrante della soluzione EarthCam, perché fornisce la connessione di rete sicura e affidabile per lo streaming di video live dalle telecamere di EarthCam al cloud.

www.sierrawireless.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: futuranet.it/elettronica.in/abbonamenti

10 NUMERI
A SOLI

€39

ANZICHÉ € 69,00

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.it.

→ **e-mail**

scrivendo a
abbonati@elettronica.in.it

→ **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

10
NUMERI
A SOLI

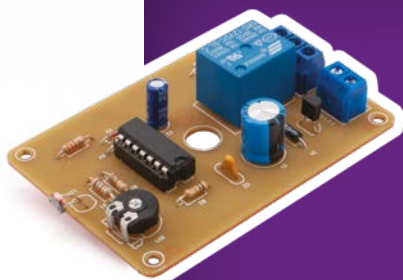
€49

ANZICHÉ € 69,00

A settembre riparti con il piede giusto.

Le scatole di montaggio KIT di Futura Elettronica sono semplici da montare e di facile utilizzo, ideali per avvicinarsi al fantastico mondo dell'elettronica.

Non utilizzano microcontrollori e sono fornite con schemi elettrici e istruzioni così da capire e personalizzare il funzionamento.



cod. FK001 | € 8,00

Interruttore Crepuscolare

Semplice circuito che permette di accendere e spegnere l'illuminazione elettrica, o altri utilizzatori, quando la luce ambientale è troppo bassa.

Modulo Intermittenza

Permette di attivare a intermittenza circuiti e apparati elettrici, come lampadine, proiettori o elementi per modellismo.



cod. FK002 | € 9,00



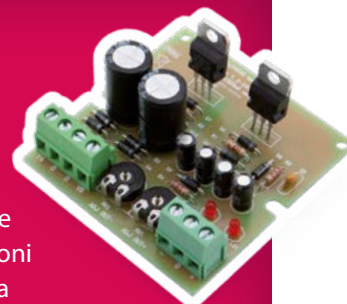
cod. FK003 | € 10,50

Alimentatore stabilizzato da 1,5 a 35 V

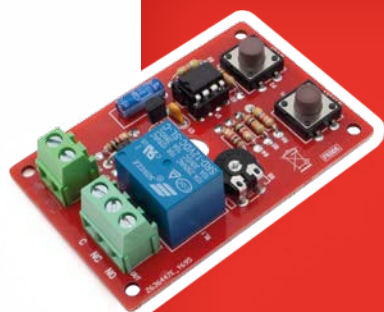
Piccolo e compatto alimentatore in grado di fornire una tensione di uscita regolabile tra 1,5 e 35 V con una corrente di 1 A.

Alimentatore stabilizzato duale e regolabile

Completo alimentatore che consente di ottenere tensioni stabilizzate simmetriche da 3 a 15 Vcc e correnti di 1 A per ogni ramo.



cod. FK004 | € 12,00



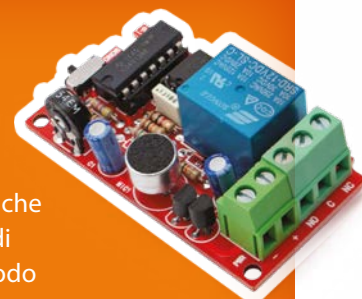
cod. FK005 | € 14,00

Timer Start e Stop da 1 a 240 secondi

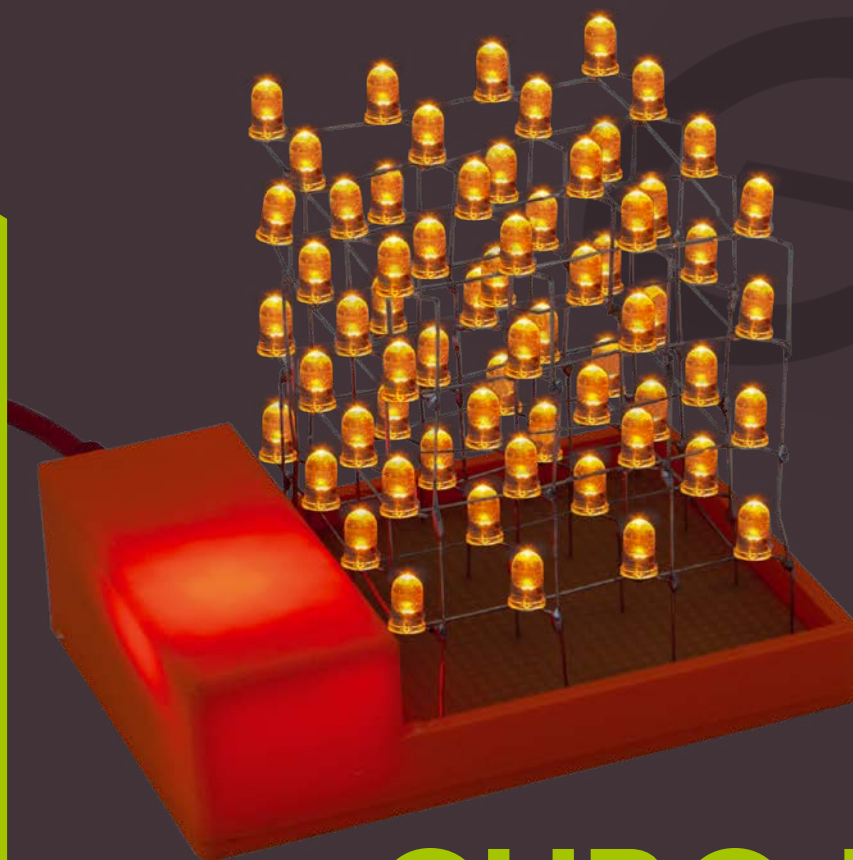
Permette di attivare per un predefinito tempo, tramite il relé di bordo, un utilizzatore ed anche di disattivarlo manualmente prima che l'intervallo stabilito sia trascorso.

Interruttore Battimani

Riconosce il rumore del battito delle mani (ed anche quelli simili per spettro di frequenza) e attiva in modo astabile (temporizzato) o bistabile un dispositivo alimentato elettricamente.



cod. FK006 | € 14,00



CUBO LED CON ARDUINO

di MATTEO PISONI

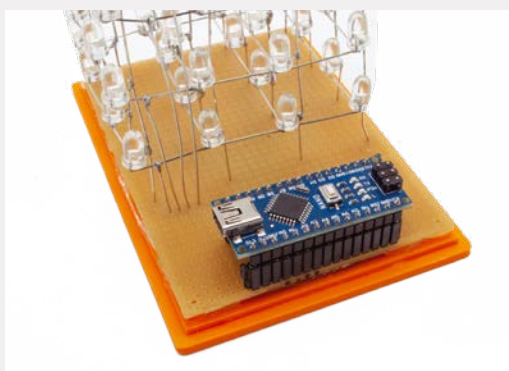
Gioco di luci tridimensionale capace di strabiliarvi con le combinazioni di cui è capace: economico e di facile realizzazione, grazie a un controller Arduino.

A

chi non piace assistere a spettacoli e giochi di luce, anche e soprattutto se creati con dei circuiti elettronici autocostruiti? Questo è possibile sempre più con le opportunità offerte da microcontrollori e da moduli reperibili in commercio, che hanno come protagonista uno dei componenti più semplici e conosciuti dell'elettronica, impiegato in molti settori, tra cui illuminazione delle abitazioni, display, semafori, fari per veicoli, ma anche per le classiche spie luminose. Questo componente è il LED, un particolare tipo di diodo a semiconduttore in grado di emettere luce quando attraversato da una corrente elettrica in polarizzazione diretta. In queste pagine vi proponiamo di realizzare un progetto che vi incanterà con i suoi giochi di luce, ancor più affascinanti al buio, basato proprio sui diodi luminosi; per l'esattezza, una matrice di LED monocromatici a tre dimensioni, disposta nello spazio a



La basetta millefori con applicata la Arduino Nano.



formare un cubo. Si tratta di qualcosa di suggestivo e al contempo semplice, alla portata anche dei più giovani e di coloro che si stanno affacciando all'elettronica per la prima volta; a ciò, aggiungete il fatto che per la realizzazione non è necessario costruire un circuito stampato, giacché le connessioni della struttura a cubo si realizzano saldando insieme i terminali dei LED e quelle terminali, verso l'unità controller, possono essere messe a punto attraverso una basetta millefori.

IL NOSTRO CUBO

Il progetto che vi proponiamo è composto da un cubo visualizzatore pilotato da una scheda elettronica, che non poteva non essere una delle mille "trasformazioni" della popolare Arduino; per l'esattezza una Arduino Nano, le cui dotazioni hardware, a livello I/O e di potenza computazionale sono più che sufficienti per implementare l'unità controller del

nostro circuito, la quale piloterà i LED per comporre i giochi di luce.

La struttura del cubo è costituita da quattro piani (strati) di diodi luminosi saldati tra loro dopo aver piegato opportunamente i reofori e disposti con l'opportuna polarità. Ogni piano è composto da 4 file, che a loro volta sono composte da 4 LED, con un totale di 16 LED per ogni piano; quindi il cubo è composto da 64 LED in totale. Il tutto è gestito da una scheda Arduino Nano attraverso i collegamenti fatti utilizzando cavi saldati a una basetta millefori, che diramano la corrente. Il firmware decide quali LED del cubo si accenderanno e quali no, creando effetti luminosi mediante il pilotaggio in multiplex. Il nostro progetto richiede dunque questi elementi:

- scheda Arduino nano;
- 64 LED monocromatici del colore che preferite;
- basetta millefori per sostenere la matrice.

Nella scheda Arduino caricheremo un firmware che abbiamo scritto appositamente e che vi rendiamo disponibile nella sezione Download del nostro sito www.elettronica.in.it, insieme ai file di questo numero della rivista.

Sul piano elettrico, lo schema di connessione è proposto dall'immagine in **Fig. 1**, la quale propone la corrispondenza tra le linee della Arduino Nano e le file di LED. Più esattamente, tra parentesi sono indicati fila e numero di LED: per esempio (1,2) significa che il pin di Arduino corrispondente va collegato al primo LED della fila 1; ecco perché nello schema tali coppie di numeri sono indicate come Y, X.

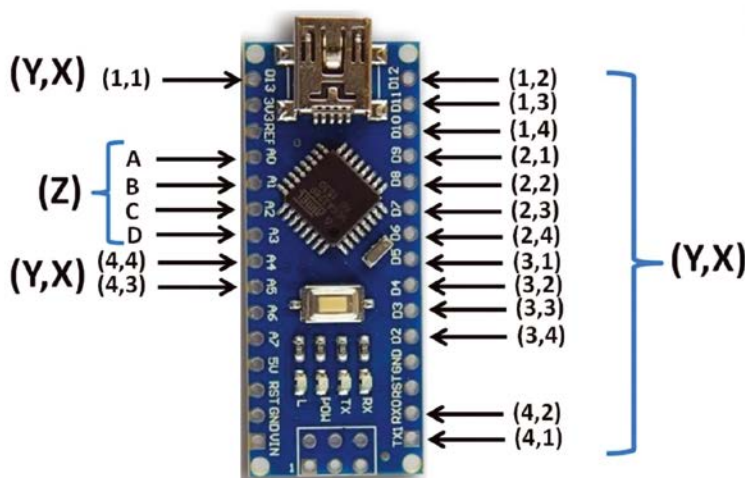


Fig. 1
Collegamenti tra gli I/O di Arduino e le file di LED.



Invece gli strati di 16 LED ciascuno corrispondono a Z e sono da connettere ai pin A0 (A), A1 (B), A2 (C) e A3 (D). L'indicazione Z è più che appropriata perché i piani sono disposti verticalmente, quindi, appunto, sull'asse Z, mentre X e Y sono la larghezza e profondità del cubo, definiti per colonna.

Queste connessioni risultano di più semplice comprensione guardando la **Fig. 2**, che propone la disposizione spaziale dei LED e chiarisce i collegamenti delle coppie di numeri riportate nella **Fig. 1**. Ogni connessione agli I/O di Arduino pilota un anodo dei LED, mentre i catodi sono uniti tra i diodi di ogni piano e vanno alle linee A, B, C, D, che verranno poste ciclicamente a livello logico basso.

Quindi a livello hardware ogni colonna del nostro cubo è collegata a un pad di I/O della scheda, così che ogni pin abbia collegati a sé 4 LED; ma siccome la nostra Arduino Nano ha solo 14 pin digitali dovremo convertire 2 pin analogici in pin digitali, così da ottenere 16 pin digitali (13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, A5, A4) che ci consentiranno di accendere o spegnere la colonna da noi desiderata.

Lo stesso procedimento lo abbiamo applicato ai 4 piani, a cui abbiamo collegato 4 pin (A3, A2, A1 e A0): ogni pin controlla un piano, così che incrociando l'accensione del piano con l'accensione della colonna potremo indicare ad Arduino quali led LED accendere e quali invece lasciare spenti.

Utilizzando questo stratagemma potremo realizzare questo progetto con solo 20 pin di I/O, cosa che sarebbe stata impossibile se avessimo collegato ad ogni pin un LED, dato che in quel caso avremmo necessitato di 64 pin e, chiaramente, non ci sarebbero bastati quelli di una piccola Arduino Nano.

Riassumendo, per poter comandare indipendentemente ogni singolo led dividiamo il cubo in livelli (per orizzontale) e in colonne (verticale).

Ogni LED posto sullo stesso livello (piano) avrà il catodo (-) in comune con gli altri led dello stesso livello mentre invece ogni LED posto sulla stessa colonna avrà l'anodo (+) in comune con gli altri LED della medesima colonna.

In tutto ci saranno da comandare 4 pin, che serviranno a selezionare il piano da alimentare e 16 anodi che daranno corrente alle singole colonne. Quando dovremo accendere un determinato LED dovremo fare in modo che il suo piano sia portato a zero logico e che la colonna cui appartiene sia attiva, ossia portata a uno logico.

REALIZZAZIONE PRATICA

Bene, giunti a questo punto possiamo spiegare come realizzare il nostro cubo luminoso a LED,

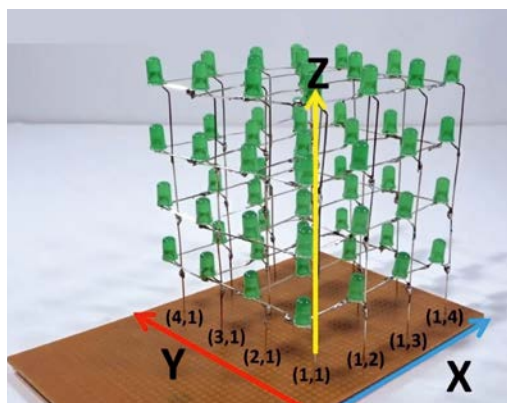


Fig. 2
Disposizione
spaziale dei
LED e identi-
ficazione dei
collegamenti.

operazione per la quale, oltre ai 64 diodi luminosi (nel nostro caso sono stati adottati LED tondi del diametro di 5 mm) e ad Arduino Nano, vi occorrerà del filo in rame rigido da 0,5x0,8 mmq e del cavo flessibile per i collegamenti verso il cubo e tra i LED che lo compongono.

Inoltre consigliamo di utilizzare un quadrato di cartone (oppure di aesite o masonite di piccolo spessore, come 3 mm) delle dimensioni di 13x13cm che fungerà da matrice o dima, che vi servirà per realizzare i piani del vostro cubo; praticamente segnerete questo cartone con quattro righe verticali e altrettante orizzontali, facendo poi quattro puntini equidistanti per ciascuna riga in modo che alla fine ci saranno 16 punti, ognuno equidistante da quello vicino.

Forerete quindi con la punta della penna il cartone, quanto basta a farci entrare un LED capovolto, ottenendo il risultato di **Fig. 3**; quindi per realizzare i piani che poi formeranno il cubo, ognuno dei quali è composto da 16 LED (in matrice di 4x4) che saranno disposti in modo equidistante l'uno dall'altro, forate il cartone negli incroci tra le righe così da mantenere la distanza corretta all'interno della struttura.

Nel nostro caso, la distanza (interasse) tra ciascun

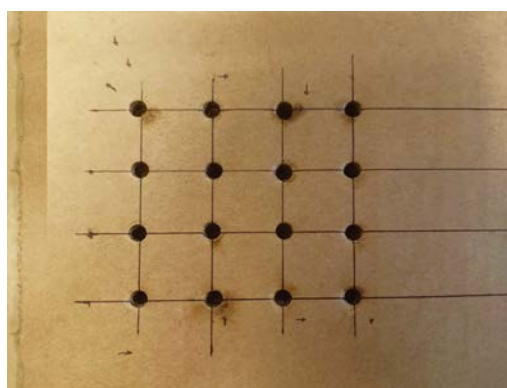
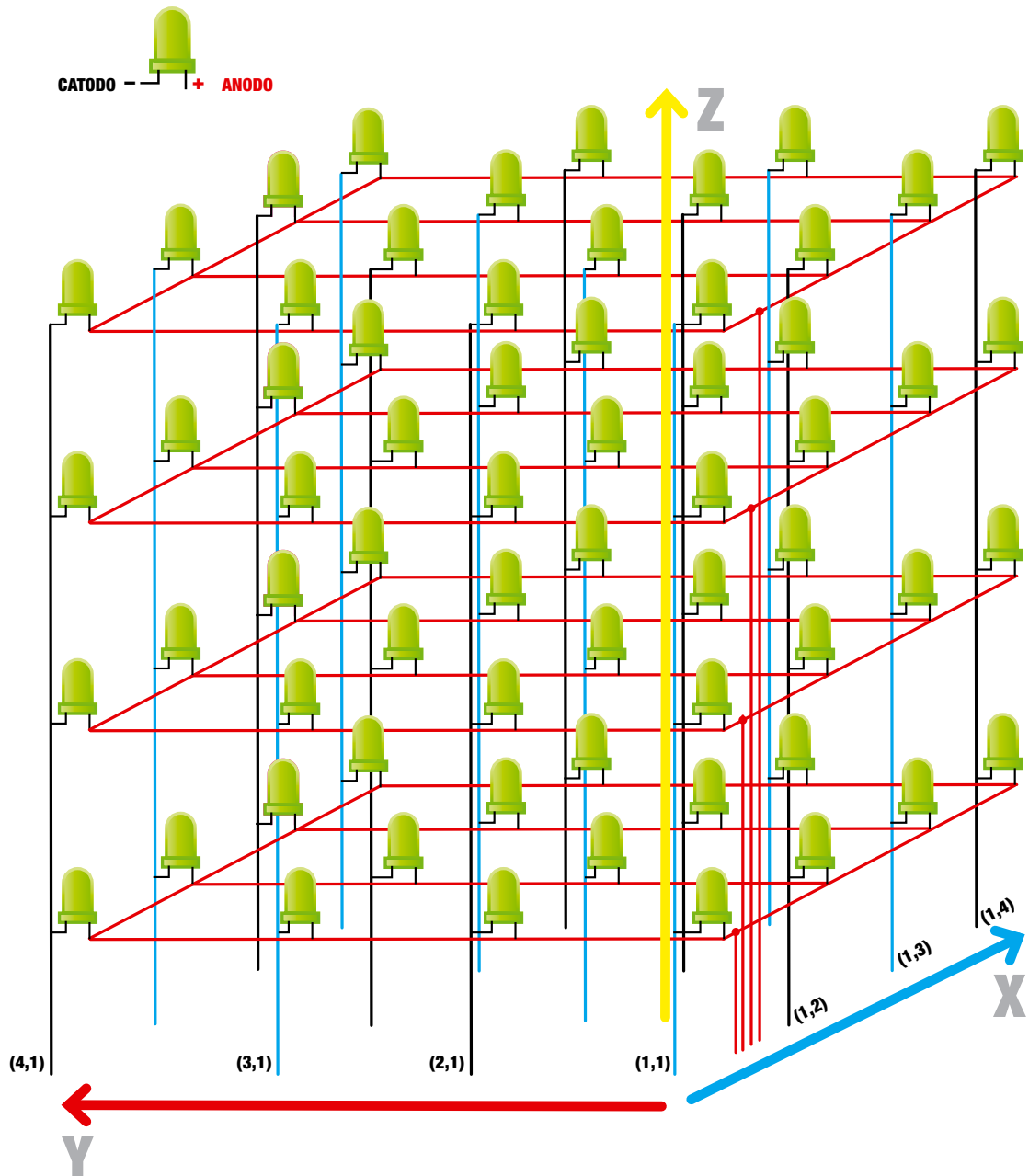


Fig. 3
La dima in
cartone per
distanziare i
LED.



LED e quello attiguo è di 3 cm: una distanza che ci sembra quella ottimale per ottenere una visualizzazione compatta e funzionale delle figure formate dal cubo al buio.

Prendete dunque il quadrato di cartone forato e inseritevi nei 16 fori altrettanti LED, rovesciati (ossia con la "testa" nei fori) e a quel punto potrete piegare i terminali e iniziare a saldare tra loro quelli componenti un piano; questa dritta non solo consente di stabilire una distanza precisa fra i LED, ma anche di tenere fermi i diodi luminosi per agevolare la saldatura.

Piegate quindi i catodi ad angolo retto e stagnateli

insieme: allo scopo, piegateli in modo che quello di un LED possa toccare quello del successivo ed esservi stagnato.

Se i terminali non sono lunghi abbastanza, unite i catodi di ogni linea con uno spezzone di filo in rame rigido e poi trasversalmente, unite con il solito filo le righe di ogni piano, fino a realizzare una griglia. Invece gli anodi vanno piegati a scalino, ossia una prima volta ad angolo retto e poi nuovamente a 90° a una distanza di circa 3 mm, quanto basta a fare in modo che possano passare a lato della testa dei LED sottostanti.

Poi devono essere uniti (se la loro lunghezza non è sufficiente, con del filo in rame rigido) a quelli dello strato successivo.

Completato un piano, estraetelo dalla dima di cartone e cablatene un altro in maniera analoga; una volta riprodotti questi piani nel medesimo modo, dovrete collegarli verticalmente, attraverso gli anodi, così da creare una sorta di gabbia che sosterrà la struttura, rendendola resistente e permettendole di acquisire una forma simile a un cubo.

Il collegamento verticale dovrà essere uno per ciascuna colonna di LED ed uno per ciascuno dei 16 LED di ogni piano; infatti le colonne sono 16 in tutto (il primo LED di ogni piano dovrà combaciare con quello del piano sottostante e via di seguito). Per fissare il tutto e collegarlo, è consigliabile utilizzare una basetta millefori delle dimensioni adatte: almeno 10x10 mm (la piastra P-972 delle dimensioni di 100x160 mm disponibile su www.futurashop.it va benissimo per lo scopo); nei suoi fori inserite gli estremi delle colonne del cubo, 16 in tutto, stagnandole nelle piazzole corrispondenti, quindi collegate queste ultime con spezzoni di filo in rame in guaina, ai pad della scheda Arduino Nano secondo quanto già spiegato e riferendovi alle indicazioni in **Fig. 1** e **Fig. 2**.

Una vista più dettagliata dell'insieme e delle connessioni è proposta nello schema di cablaggio nella pagina qui accanto.

A lavoro completato otterrete qualcosa come quello mostrato nelle foto del prototipo che trovate in queste pagine.

Prima di procedere analizzando gli aspetti salienti del firmware di governo del nostro gadget luminoso, è opportuno soffermarsi un istante sui LED da utilizzare: nel prototipo vedete impiegati diodi ad alta luminosità in package trasparente; ebbene, i LED ad alta luminosità hanno la caratteristica di concentrare la loro luce in un angolo più ristretto di quello di emissione dei LED convenzionali, ma questo implica che il cubo renderà ben visibile la luce solo a partire da una certa angolazione e preferibilmente da un punto di osservazione molto inclinato rispetto al piano orizzontale.

Se desiderate avere un'emissione luminosa più uniforme, seppure meno intensa, potete utilizzare LED convenzionali (che tipicamente emettono in un angolo di 120÷140 gradi) con l'involucro colorato: ad esempio quelli verdi proposti nella **Fig. 2**, sempre del diametro di 5 mm.

Ricordiamo che l'intero circuito funziona con l'alimentazione prelevata dalla connessione microUSB della scheda Arduino Nano, che conatterete a un computer oppure, una volta caricato lo sketch, ad un alimentatore dotato di connessione microUSB, capace di erogare almeno 500 milliampere.

IL FIRMWARE DEL CUBO

Il codice che abbiamo realizzato è basato sul principio di funzionamento già esposto, ossia considerando che Arduino deve comandare quattro pin che serviranno a selezionare il piano da alimentare e 16 anodi che daranno corrente alle singole colonne. Quando dovremo accendere un determinato LED della matrice tridimensionale che costituisce il cubo, dovremo fare in modo che il suo piano (ossia quello su cui si trova fisicamente ed elettricamente...) sia alimentato (ovvero i relativi catodi siano posti a massa dalla linea di Arduino Nano tra A, B, C, D) e che la sua colonna sia attiva; così facendo accenderemo un LED alla volta per ogni piano.

Questo è il principio, ma ora entreremo nel dettaglio di come è stato realizzato ed impostato passo-passo a livello firmware.

La prima cosa che abbiamo realizzato è stata la dichiarazione dei pin (in pratica nel listato abbiamo scritto i pin di I/O che ci saranno necessari in questo progetto). Dichiarare i pin all'interno del codice serve

↓ Listato 1

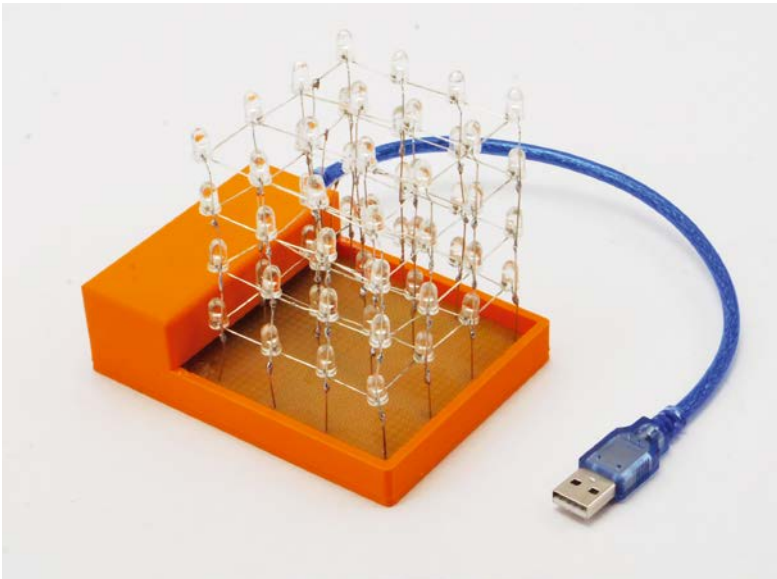
```
/*
  4x4x4 LED Cube
  Connection Setup:
  Columns
  [(x,y)-Pin]
  (1,1)-13
  (1,2)-12
  (1,3)-11
  (1,4)-10
  (2,1)-9
  (2,2)-8
  (2,3)-7
  (2,4)-6
  (3,1)-5
  (3,2)-4
  (3,3)-3
  (3,4)-2
  (4,1)-1
  (4,2)-0
  (4,3)-A5
  (4,4)-A4
  Layers
  [layer-Pin]
  a-A0
  b-A1
  c-A2
  d-A3
*/

//initializing and declaring led columns
int column[16]={13,12,11,10,9,8,7,6,5,4,3,2,1,0,A5,A4};

//initializing and declaring led layers
int layer[4]={A3,A2,A1,A0};

int time = 250;
```





Il prototipo
dotato di
base in
plastica che
contiene
la Arduino
Nano.

a configurare un determinato pin di Arduino e stabilisce se deve essere come linea di INPUT (entrata) o di OUTPUT (uscita).

In questo caso i pin dichiarati sono stati divisi in colonne (Column) e in piani (Layer), le quali sono uscite attive rispettivamente a livello alto e a zero logico (**Listato 1**).

Segue quindi il **setup**, nel quale vengono definite le impostazioni del firmware ed il ciclo **for**, che è alla base della visualizzazione dei giochi di luce, perché

Listato 2

```
void loop()
{
  Serial.println("sono nel loop");
  turnEverythingOff(); //turn all off
  flickerOn();
  turnEverythingOn(); //turn all on
  delay(time);
  turnOnAndOffAllByLayerUpAndDownNotTimed();
  layerstompUpAndDown();
  turnOnAndOffAllByColumnSideways();
  delay(time);
  aroundEdgeDown();
  turnEverythingOff();
  randomflicker();
  randomRain();
  diagonalRectangle();
  goThroughAllLedsOneAtATime();
  propeller();
  spiralInAndOut();
  flickerOff();
  turnEverythingOff();
  delay(2000);
}
```

Listato 3

```
////////////////////////////////////////propeller
void propeller()
{
  turnEverythingOff();
  int x = 90;
  for(int y = 4; y>0; y--)
  {
    for(int i = 0; i<6; i++)
    {
      //turn on layer
      digitalWrite(layer[y-1], 1);
      //a1
      turnColumnsOff();
      digitalWrite(column[0], 0);
      digitalWrite(column[5], 0);
      digitalWrite(column[10], 0);
      digitalWrite(column[15], 0);
      delay(x);
      //b1
      turnColumnsOff();
      digitalWrite(column[4], 0);
      digitalWrite(column[5], 0);
      digitalWrite(column[10], 0);
      digitalWrite(column[11], 0);
      delay(x);
      //c1
      turnColumnsOff();
      digitalWrite(column[6], 0);
      digitalWrite(column[7], 0);
      digitalWrite(column[8], 0);
      digitalWrite(column[9], 0);
      delay(x);
      //d1
      turnColumnsOff();
      digitalWrite(column[3], 0);
      digitalWrite(column[6], 0);
      digitalWrite(column[9], 0);
      digitalWrite(column[12], 0);
      delay(x);
      //d2
      turnColumnsOff();
      digitalWrite(column[2], 0);
      digitalWrite(column[6], 0);
      digitalWrite(column[9], 0);
      digitalWrite(column[13], 0);
      delay(x);
      //d3
      turnColumnsOff();
      digitalWrite(column[1], 0);
      digitalWrite(column[5], 0);
      digitalWrite(column[10], 0);
      digitalWrite(column[14], 0);
      delay(x);
    }
  }
  //d4
  turnColumnsOff();
  digitalWrite(column[0], 0);
  digitalWrite(column[5], 0);
  digitalWrite(column[10], 0);
  digitalWrite(column[15], 0);
  delay(x);
}
```

definisce l'avvicinarsi dei livelli logici sui pin della scheda Arduino, allo scopo di comandare i LED in multiplex.

Infine nello sketch si trova il loop, che contiene tutte le routine riguardanti i giochi di luce previsti (riferitevi al **Listato 2**).

All'interno di questo **loop** ci sono le singole funzioni corrispondenti ai giochi di luce, delle quali riportiamo nel **Listato 3** la composizione luminosa (animazione) Propeller, il cui effetto visivo consiste nell'accendere i LED dei piani in sequenza in modo da far apparire una luce rotante tipo il movimento di un'elica. Chiaramente le funzioni contenute nel loop sono una per ciascuno dei giochi di luce. L'elenco delle funzioni, ovvero delle animazioni previste dal cubo a LED, è il seguente:

- `turnEverythingOff();`
- `flickerOn();`
- `turnEverythingOn();`
- `turnOnAndOffAllByLayerUpAndDownNotTi;`
- `layerstompUpAndDown();`
- `turnOnAndOffAllByColumnSideways();`
- `aroundEdgeDown();`
- `randomflicker();`
- `randomRain();`
- `diagonalRectangle();`
- `goThroughAllLedsOneAtATime();`
- `propeller();`
- `spiralInAndOut();`
- `flickerOff();`

Al termine del loop, l'istruzione **delay(2000)**; impone un ritardo di 2 secondi prima dell'esecuzione di una nuova sequenza.

Naturalmente per installare lo sketch in Arduino dovete collegare la scheda Arduino Nano (montata o meno nel circuito del cubo, è irrilevante...) quindi avviare l'IDE, poi dal menù Strumenti > Scheda, scegliere la board Arduino Nano e poi aprire (File > Apri) lo sketch, per poi avviare il caricamento nella Arduino mediante un clic sul pulsante Carica. Bene, con questo abbiamo concluso la descrizione del nostro cubo a LED.

CONCLUSIONI

Il vantaggio di questo progetto è che potrete configurarlo a vostro piacimento, aggiungendo nuovi effetti di illuminazione, anche se per fare ciò dovete avere una certa conoscenza nel linguaggio di programmazione C applicato ad Arduino; comunque non vi preoccupate più di tanto, perché anche solo cercando qua e là in rete trovate molti progetti simili

Guarda il Cubo a LED in azione su YouTube!



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

a contare LED e anche oltre, con le relative complessità hardware

Per la configurazione dello sketch allo scopo di adattare il nostro al controllo di un maggior numero di diodi luminosi, basterà solamente copiare e incollare il codice sul programma della scheda (per il meno esperti), poi dovrete solamente modificare la dichiarazione dei pin all'interno del codice o cambiare i collegamenti dei piani e colonne su Arduino. Ciò detto, non vi resta che augurarvi buon lavoro e buon divertimento con il vostro cubo luminoso. ■



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. La scheda Arduino Nano (cod. ARDUINONANO) è disponibile a Euro 29,00, la piastra millefori 100x160 mm (cod. P-972) è in vendita a Euro 6,30, i led sono disponibili in confezioni da 100 pezzi in diverso colore (cod. 100LED5xxx, dove xxx indica il colore, ovvero BLU, YEL, RED, GRE o WHI) al costo di Euro 3,50 a confezione. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

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

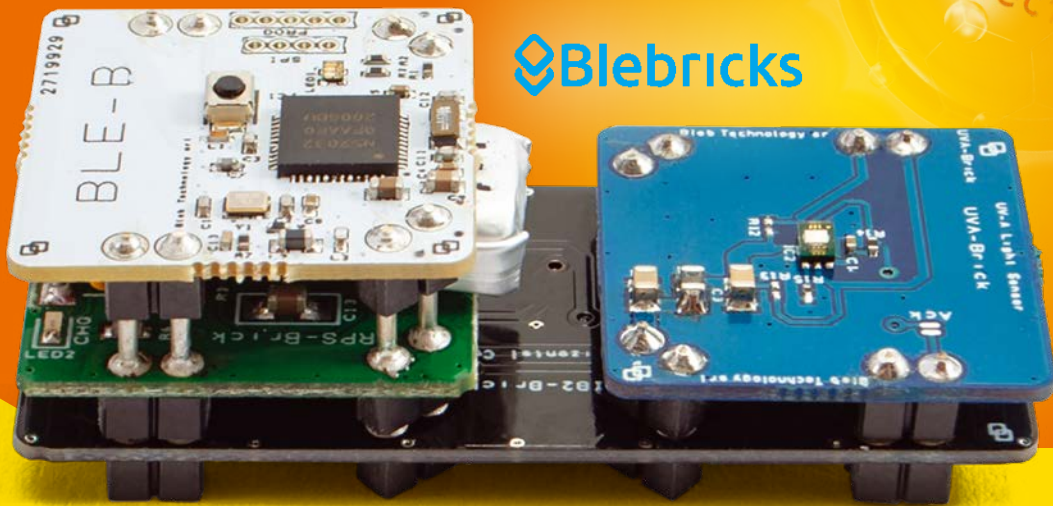
Prezzi IVA inclusa.

WIRELESS

SENSORE UV BLUETOOTH

dell' ing. MIRCO SEGATELLO

 Blebricks



Realizziamo un rilevatore di radiazioni UV con allarme su smartphone tramite l'utilizzo di moduli Blebricks.

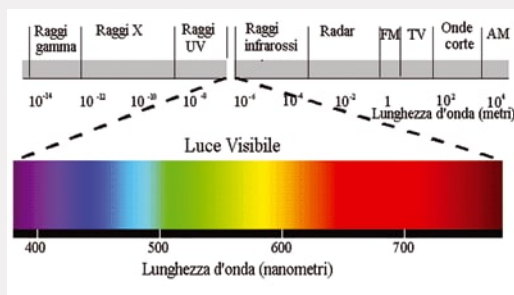
L

estate, come si sa, è il periodo dell'anno legato al divertimento, allo svago e soprattutto alle ferie ed inoltre la bella stagione favorisce le attività all'aria aperta. Occorre però porre molta attenzione al sole diretto che colpisce

la nostra pelle, perché un'esposizione eccessiva può avere effetti negativi, da cui l'esigenza di usare ombrelloni, creme solari e protezioni varie. Negli ultimi anni, a causa dell'assottigliamento dello strato di ozono, anche i mesi autunnali possono richiedere qualche accortezza aggiuntiva. Le radiazioni ultraviolette (UV) prodotte dal sole sono tra le più pericolose per la nostra pelle, soprattutto durante il periodo estivo quando raggiungono



Fig. 1
Spettro delle
radiazioni
elettromagnetiche.



il loro massimo. Esse interessano quella porzione di spettro elettromagnetico con lunghezza d'onda compresa tra 100 e 400 nanometri, ovvero le onde comprese tra la luce visibile e le radiazioni ionizzanti, come raggi X e raggi gamma. I livelli di UV aumentano al crescere dell'altitudine e dell'altezza del sole (specialmente verso mezzogiorno nei mesi estivi) e diminuiscono all'aumentare della latitudine e della nuvolosità.

I raggi ultravioletti (**Fig. 1**) si suddividono in tre categorie o, più tecnicamente, per gamma di lunghezza d'onda, secondo quanto esposto qui di seguito.

- Raggi UV-A: con lunghezza d'onda compresa tra 315 e 400 nm rappresentano il 96% dei raggi ultravioletti e penetrano in profondità nella pelle, pertanto sono responsabili dell'invecchiamento della pelle e favoriscono l'insorgere di dermatiti e tumori, ma hanno anche un ruolo fondamentale nella sintesi della vitamina D e nel fissaggio del calcio nelle ossa.
- Raggi UV-B: con lunghezza d'onda compresa tra 280 e 315 nm rappresentano il 4% dei raggi ultravioletti, che raggiungono la Terra. Sono assorbiti dalla pelle a livello superficiale pertanto responsabili dell'abbronzatura ma sono in gra-

do di alterare il DNA delle cellule aumentando il rischio di comparsa di tumori cutanei.

- Raggi UV-C: con lunghezza d'onda compresa tra 100 e 280 nm sono le radiazioni ultraviolette più dannose, che però sono bloccate dall'ozonofera.

Mentre una adeguata esposizione alla luce del sole è benefica perché stimola la sintesi organica di vitamina D ed è propedeutico alla "tintarella", un eccesso può avere effetti negativi; la lunghezza d'onda è così piccola che le radiazioni UV possono interagire direttamente a livello molecolare creando il più delle volte semplici scottature ma in alcuni casi anche modificazioni permanenti alla struttura cellulare come nel caso dei melanomi e carcinomi della pelle.

L'organizzazione mondiale della sanità, al fine di rendere consapevole la popolazione in merito ai danni da radiazioni UV, ha introdotto l'indice universale della radiazione UV solare (UVI) che descrive il livello di radiazione ultravioletta che colpisce la superficie terrestre, introducendo anche degli indicatori (come riportato in **Fig. 2**) che possano in modo facile ed immediato evidenziare il livello di pericolosità.

L'indice UV viene misurato da apposite centraline meteo i cui dati sono poi resi disponibili alla popolazione in appositi siti come ad esempio quello dell'ARPA (agenzia regionale per la protezione dell'ambiente). In commercio sono disponibili misuratori portatili con il vantaggio di poter fare misurazioni anche in condizioni particolari come sotto l'ombrellone, all'ombra di un albero o sotto una tettoia. In questo articolo vedremo come

Fig. 2
Indice
universale
della
radiazione UV
solare.

UV index	UV index	UV index	UV index	UV index	UV index	UV index	UV index	UV index	UV index	UV index
1	2	3	4	5	6	7	8	9	10	11+
Low		Moderate			High		Very high			Extreme
Non sono necessarie precauzioni		Protegersi: cercare ombra nelle ore attorno mezzogiorno, indossare cappello, maglietta ed occhiali da sole, usare creme protettive					Protegersi: rimanere all'ombra, indossare cappello, maglietta ed occhiali da sole, usare creme protettive			

costruire un piccolo ma efficace misuratore di raggi UV-A implementando anche un sistema di allarme con avviso direttamente su smartphone.

DESCRIZIONE DELL'HARDWARE

Per realizzare un misuratore di indice UV, il primo approccio è quello di utilizzare una scheda Arduino e cercare in commercio un sensore di raggi UV da abbinare. Il secondo passo sarebbe stato quello di studiare un sistema di alimentazione a batteria sufficientemente efficace da assicurare il funzionamento per giorni con la minima manutenzione. Questa volta, invece, cambiamo completamente approccio e ci affidiamo ai nuovissimi dispositivi denominati Blebricks.

Ma cosa sono i Blebricks? Innanzi tutto diciamo che si tratta di un prodotto tutto italiano progettato e distribuito dalla ditta Bleb Technology Srl che ha sede a Prato (PO) il cui sito di riferimento è bleb.it. Sulla pagina principale si legge la definizione di cos'è un Blebrick: "I Blebricks sono mattoncini intelligenti realizzati con una tecnologia brevettata grazie alla quale, con il loro assemblaggio, si possono creare in un attimo nuovi dispositivi elettronici, dotati di sensori e attuatori, capaci di comunicare con il tuo smartphone e con Internet". L'idea, se vogliamo, geniale, è quella di fornire tutta una serie di moduli facilmente assemblabili semplicemente incastrandoli gli uni sugli altri come avviene con i noti mattoncini utilizzati dai bambini per giocare, ed infatti è proprio questo l'intento rendere la tecnologia IoT facile e accessibile. I principali vantaggi dell'adozione dei Blebricks sono la possibilità di eseguire una prototipazione istantanea, la possibilità di creare prototipi senza una particolare competenza né attrezzature costose, nonché la drastica riduzione di tempi e costi di sviluppo del 90% dall'idea al prodotto finito.

Entriamo ora nei dettagli del progetto e tra gli innumerevoli "Bricks" a disposizione scegliamo quelli che servono al nostro progetto. Il primo modulo che useremo si chiama BLE-B (**Fig. 3**) ed è il modulo base cuore del sistema basato su un microcontrollore ad alte prestazioni che ha le dimensioni di una moneta da 2 euro (22,5 mm x 22,5 mm) ed un consumo estremamente ridotto, di soli pochi microampere, grazie all'utilizzo di un modulo Bluetooth Low Energy per la comunicazione.

Le principali caratteristiche di questo Blebrick sono:

- Bluetooth Low Energy (compatibile v4.2 o 5.x);
- NFC Tag Emulation;
- Pulsante;
- LED RGB;

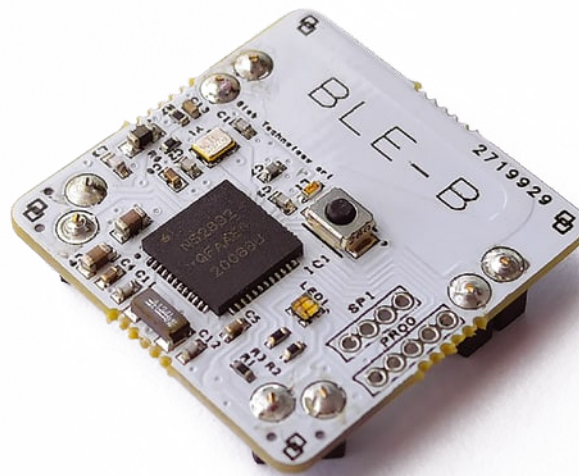


Fig. 3
Modulo
Blebricks
BLE-B.

- 4 GPIO configurabili come ingressi o uscite digitali, ingressi analogici, contatori di impulsi e contagiri;
- Interfacce I²C Bus, UART e SPI;
- sensore di temperatura;
- sensore tensione batteria (livello di carica).

Questo brick viene alimentato dal modulo di nome RPS (**Fig. 4**) che consiste in una piccola batteria Li-Po ed un circuito di ricarica tramite micro USB. Il modulo è fornito di due LED con le seguenti funzioni:

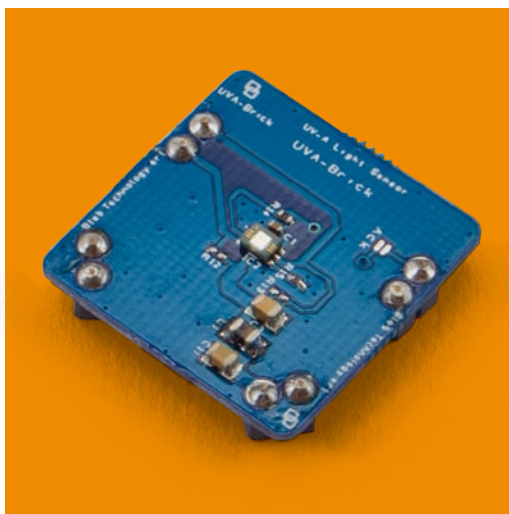
Il LED verde è spento quando la sola batteria è collegata al Blebrick RPS. Il LED verde si accende nel momento in cui il Blebrick RPS è collegato tramite cavetto micro-USB a una sorgente a 5VDC.

Quando sia la batteria che l'alimentazione esterna a 5VDC sono collegati, il LED verde è sempre acceso, mentre il LED rosso si accende soltanto



Fig. 4
Modulo
Blebricks
RPS.

→ **Fig. 5**
Modulo Blebricks
UVA.



↓ **Fig. 6**
Modulo
Blebricks IB2.

→ **Fig. 7**
Moduli
Blebricks
assemblati.

durante la carica della batteria. La cosa interessante è che il modulo BLE-B e RPS semplicemente si incastrano l'uno sull'altro realizzando tutte le connessioni elettriche necessarie, questa operazione rende immediatamente operativo il sistema Blebricks.

Per la misura dei raggi ultravioletti scegliamo (tra i tanti sensori disponibili) il modulo UVA (**Fig. 5**) che, con lo stesso form factor degli altri moduli, mette a disposizione un sensore in grado di rilevare radiazioni UV nello spettro da 320 a 410 nm (UVA) con una potenza massima di 328 mW/cm² e una sensibilità 5 µW/cm².

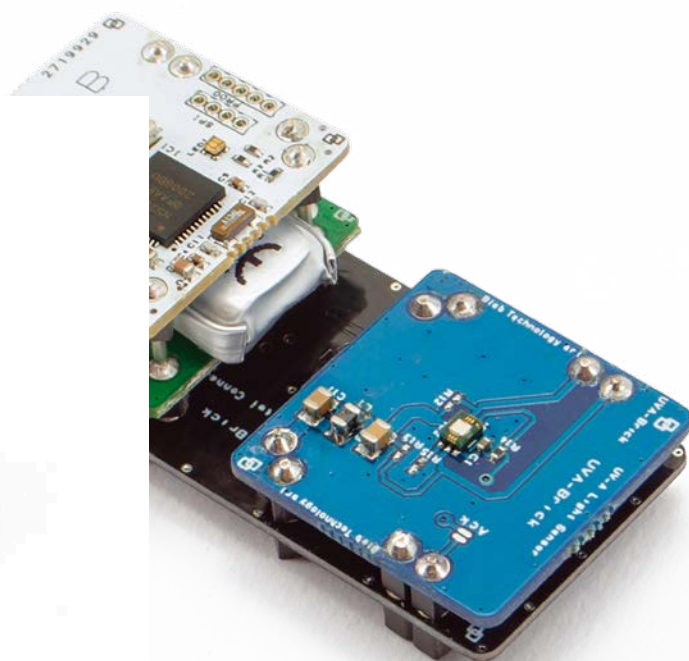
Anche questo modulo è facilmente interfacciabile agli altri utilizzando la base di montaggio di nome IB2 (**Fig. 6**), alla fine quando tutto è cablato ottenete il sistema proposto nella **Fig. 7**.

Possiamo affermare con sicurezza che implementare la parte hardware "è un gioco da ragazzi!".

IL FIRMWARE

È arrivato il momento di occuparci della parte di programmazione che, come vedremo, in perfetto stile Blebricks non sarà nulla di complicato. Con un dispositivo Android rechiamoci nel Play Store ed installiamo l'app di nome MakeApp for Blebricks, non appena avviata essa ricercherà i dispositivi BLE nelle vicinanze, non dovremo far altro che selezionare il modulo di interesse e si aprirà la relativa pagina di configurazione, come visibile in **Fig. 8**. Vengono mostrare subito le informazioni essenziali, come la distanza a cui si trova, il livello della batteria ed eventuali moduli connessi. Se è la prima volta che usate questi moduli vi consigliamo di prendere confidenza usando i comandi per la gestione del LED e del pulsante presenti sul modulo BLE-B. Cliccando sull'icona BLE-B potremmo entrare nella modalità di configurazione del modulo, come visibile in **Fig. 9**.

Tra i vari parametri possiamo impostare la potenza in trasmissione, maggiore sarà maggiore la distanza che riusciremo a coprire ma aumenteranno i consumi. Per impostazione predefinita il modulo BLE-B trasmettere pacchetti di advertising in



modo da essere rilevato anche da altri dispositivi permettendo di creare una rete di Blebricks, il parametro che ne determina la cadenza è chiamato *adv rate*. La modalità *sentry* significa che il modulo BLE-B entra in ascolto soltanto con un determinato BLE-B che gli viene specificato mentre la modalità Low Power Mode è utile quando è possibile leggere i sensori ad intervalli di minuti al fine di prolungare la durata della batteria. Nella modalità Data Logger avremo invece la possibilità di leggere e salvare in memoria i valori dei sensori ad intervalli regolari; i dati saranno poi leggibili tramite app. Sulla pagina principale in basso potremo vedere la lettura del valore dei raggi UVA forniti dal sensore che viene riconosciuto e configurato in automatico dal BLE-B (riferitevi alla Fig. 8).

A noi interessa inviare una notifica quando il livello di indice UV supera un determinato valore. Clicchiamo in alto sulla voce INTERACTIONS, si aprirà una schermata che con una semplice struttura del tipo IF-THEN permette di programmare facilmente l'applicazione per le nostre esigenze. Dovremo impostare i campi in modo che: se il valore di UVA supera la soglia impostata fai suonare lo smartphone. Nel primo menù a tendina selezioniamo il valore da monitorare UV Light from UVA, nel secondo menù la voce > e nel terzo menù la soglia che desideriamo utilizzare come riferimento: nel nostro caso, 4. Successivamente dobbiamo impostare l'azione che verrà effettuata quando il valore supera la soglia impostata, pertanto sceglieremo a quale dispositivo dovrà essere inviata la notifica: nel nostro caso "this smartphone". Poi dovremo definire che cosa dovrà essere eseguito: nel nostro caso, emettere un suono. Ogni qualvolta il mattoncino rileverà un superamento del valore UV impostato, lo smartphone emetterà un suono e in corrispondenza di ciò, l'app segnalerà l'evento. Le notifiche si susseguiranno fintanto che l'app sarà in esecuzione e finché il valore impostato sarà superato. Per quanto riguarda la durata della batteria, avendo attivato la modalità Low Power è possibile garantire diversi giorni di utilizzo ininterrotto senza la necessità di accendere e spegnere il circuito, anche perché non ha un interruttore ON/OFF.

CONCLUSIONI

Dall'idea iniziale al progetto completo in pochissimo tempo e con poche e semplici competenze tecniche: è questo il vantaggio offerto dai Blebricks e il progetto qui descritto lo dimostra senza dubbio: immediatezza hardware e software alla portata di tutti! Ma i vantaggi dei moduli Blebricks non si

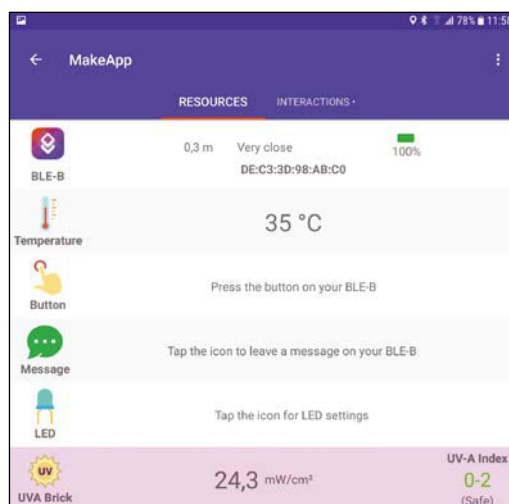


Fig. 8
Pagina di configurazione del modulo BLE-B.

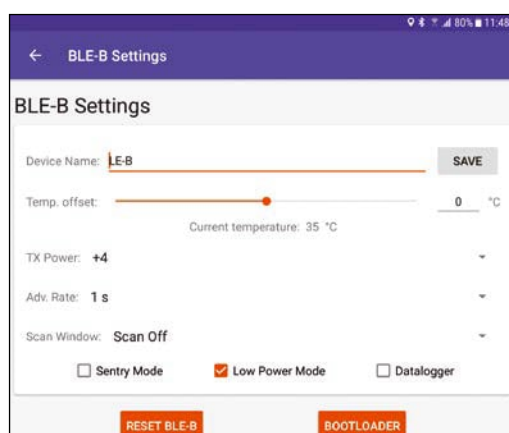


Fig. 9
Configurazione del modulo BLE-B.

esauriscono qui perché questi moduli sono pensati per funzionare in rete, permettendo funzionalità anche complesse come il posizionamento indoor e l'accesso ai dati nel cloud, ma, soprattutto, la facile prototipazione di applicazioni nell'ambito dell'Internet of Things. ■



Cosa occorre?

I moduli Blebricks sono disponibili presso Futura Elettronica. In particolare la board Blebrick Bluetooth Low Energy (cod. BLE-B) è in vendita a Euro 29,00, il modulo UVA (cod. BLEBUVA) è disponibile a Euro 18,00, la scheda per la gestione dell'alimentazione (cod. BLEBRPS) è in vendita a Euro 38,00, la base per il montaggio di più moduli (cod. BLEBIB2) 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.futurashop.it

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.

INDOOR AIR QUALITY METER

di FRANCESCO FICILI

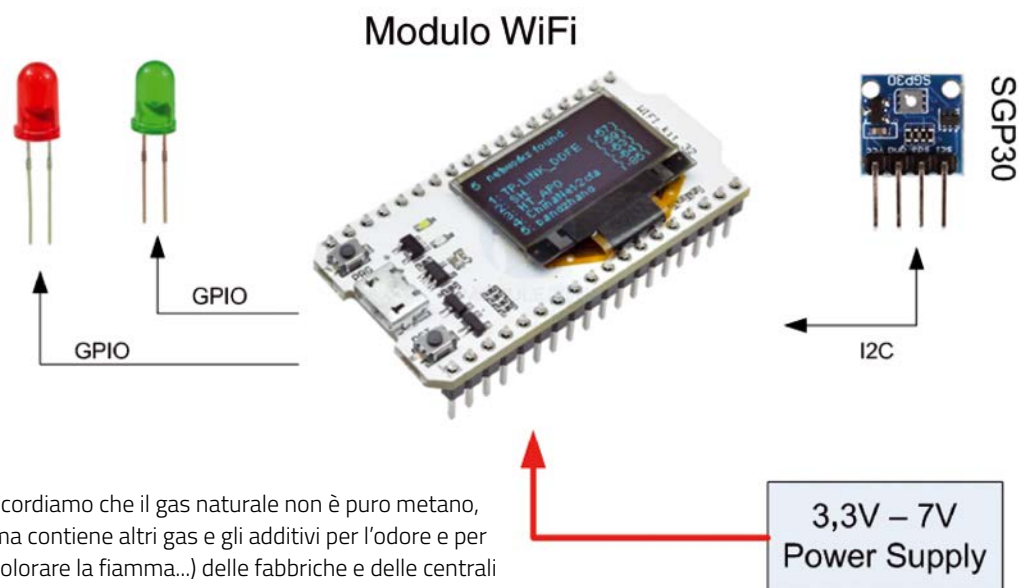


Visualizza la concentrazione di CO₂ e TVOC nell'aria in un ambiente chiuso, come l'aula di una scuola o un ufficio e segnala tramite LED se la qualità dell'aria è buona o scadente.

N

egli ultimi anni, sfortunatamente, la qualità dell'aria che respiriamo è diventata un problema non trascurabile. Lo è da tempo per gli ambienti indoor, specie in inverno, quando restiamo chiusi in tanti in un locale poco aerato, ma anche all'esterno, specie nelle città, la cui aria risulta sempre più contaminata dalle emissioni inquinanti degli autoveicoli, degli impianti di riscaldamento (non

Fig. 1
Concept del
sistema di misura
e segnalazione
qualità aria.



scordiamo che il gas naturale non è puro metano, ma contiene altri gas e gli additivi per l'odore e per colorare la fiamma...) delle fabbriche e delle centrali termoelettriche e via di seguito; e dove la scarsità di verde non favorisce l'ossigenazione dell'aria, che ha quindi un basso tenore di ossigeno e non favorisce i naturali processi dell'organismo umano e animale.

Negli ambienti indoor, la qualità dell'aria è pregiudicata da elementi come l'anidride carbonica che espiriamo, ma anche da polveri sottili ed eventuali agenti patogeni che possiamo emettere respirando o tossendo; si pensi ad esempio all'infezione da COVID 19, ormai da tempo un problema con il quale conviviamo, che ha nell'aria degli ambienti chiusi, uno dei principali veicoli di trasmissione. La diffusione di questo, come di altri virus, negli ambienti chiusi, sostanzialmente è un problema legato alla qualità dell'aria: più l'aria è "viziata", tanto maggiore è la concentrazione di droplet (goccioline di saliva e liquidi organici provenienti dalle mucose) potenzialmente cariche di agenti patogeni, capaci di diffondere malattie. La soluzione per mitigare il contagio è ovviamente quella di operare un ricambio d'aria, per ridurre la concentrazione di questi patogeni.

Ma quando farlo, dal momento che, sia in estate che in inverno, tendiamo a tenere le finestre chiuse

per ridurre i consumi di impianti di condizionamento e di riscaldamento?

Ebbene, in questo articolo vi proponiamo un semplice dispositivo che, leggendo un sensore di CO₂ e TVOC (ossia la concentrazione di particelle organiche in sospensione nell'aria, nota in Italia anche con l'acronimo COV – Composti Organici Volatili), ci consenta di avere un'idea della qualità dell'aria di un ambiente, e tramite dei LED, ci segnali quando è il momento di operare un ricambio, permettendoci di mantenere sotto controllo la qualità dell'aria e di minimizzare le perdite di energia.

L'HARDWARE DEL PROGETTO

Per realizzare il dispositivo che abbiamo sinora introdotto, abbiamo deciso di utilizzare un modulo Heltec WiFi kit 32 come unità di controllo, un sensore SGP30 della Sensirion come elemento rilevatore della concentrazione di anidride carbonica e TVOC e due comuni LED (uno rosso e l'altro verde) come elementi di segnalazione. Lo schema concettuale del sistema è proposto nella **Fig. 1**.

MODULO HELTEC WIFI KIT 32

Il modulo Heltec WiFi kit 32, che in definitiva è il cuore del nostro sistema, è una piccola scheda elettronica che integra un microcontrollore ESP32 e un display OLED da 0,96 pollici; la potete vedere nella **Fig. 2**.

Le caratteristiche hardware del modulo WiFi kit 32, con cui lavoreremo in questo progetto, sono:

- Display OLED blu da 0,96";
- Microcontroller Dual-Core Tensilica LX6 a 32 bit;

Fig. 2
Il modulo WiFi kit 32
della Heltec.



- Frequenza di clock fino a 240 MHz;
- SRAM interna da 520 kB;
- Memoria flash da 32 MB;
- Ricetrasmittitore Wi-Fi 802.11 b/g/n integrato;
- Bluetooth dual-mode integrato (classico e BLE);
- Alimentazione: da 3,3V a 7V;
- 28 GPIO (alcuni utilizzati per gestire il display OLED);
- Caricabatteria per batterie ai polimeri di litio integrato;
- Temperatura di lavoro: da -40°C a +90°C.

Questo modulo risulta perfetto per il nostro scopo, in quanto integra già un display OLED che utilizzeremo per la visualizzazione delle letture relative alla qualità dell'aria, e per i messaggi utente, oltre a diversi canali GPIO che ci servono per controllare i nostri dispositivi di segnalazione; in questa prima versione noi prevediamo di utilizzare solo 2 LED, ma il quantitativo di GPIO disponibili (28 in totale) ci permetterebbe agevolmente di aggiungere altri elementi, come ad esempio un buzzer per effettuare delle segnalazioni acustiche. Oltre a queste caratteristiche, il modulo ha

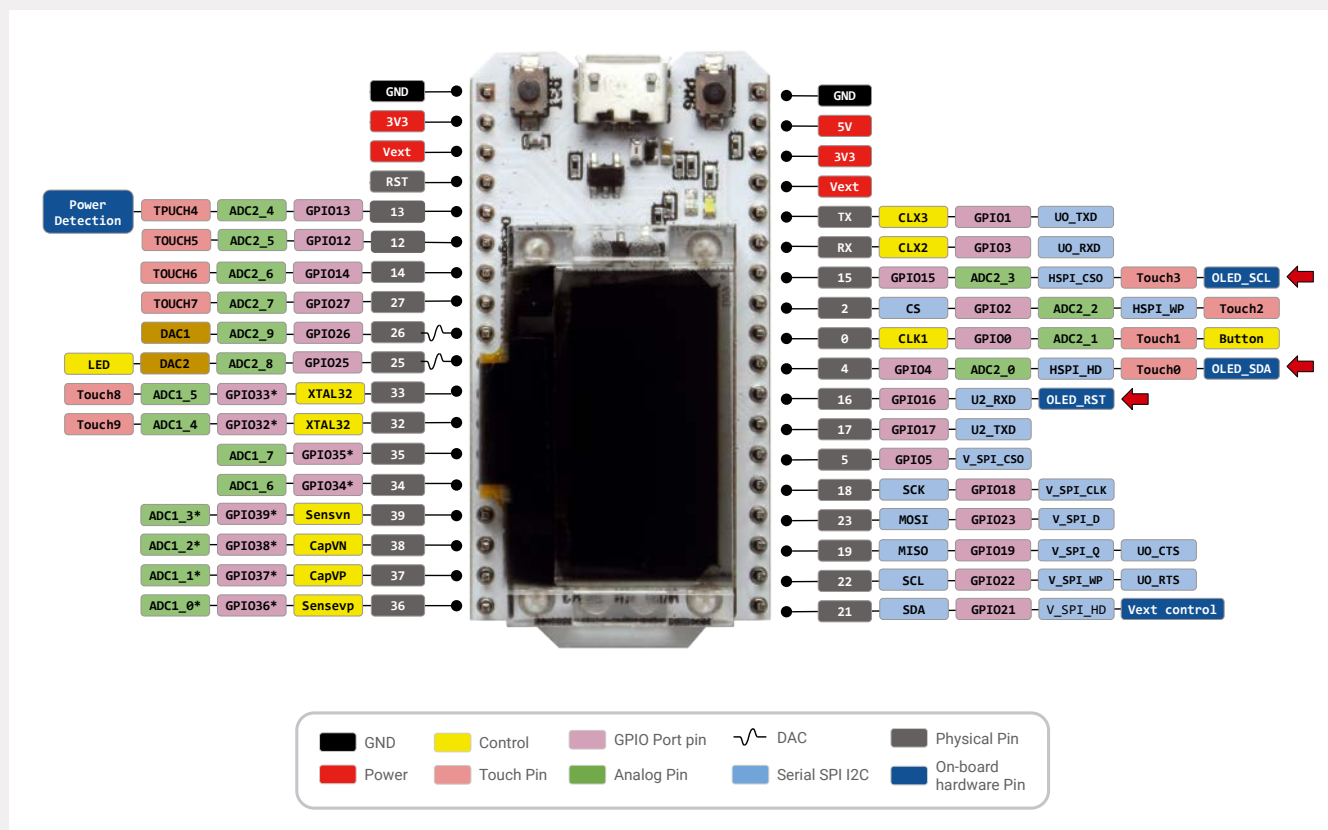
520kbyte di memoria RAM on-board, e ben 32Mbyte di Flash, caratteristiche che ci permettono di stare tranquilli in termini di risorse di memoria per il nostro software di controllo. Inoltre il modulo monta un transceiver WiFi standard 802.11, che potremo usare per future evoluzioni di questo dispositivo (ad esempio per pubblicare i dati relativi alle concentrazioni di CO₂ e TVOC direttamente su una pagina web, o per inviare eventuali allarmi via Telegram), oltre che una radio Bluetooth, che potremo utilizzare per interfacciarci con dispositivi esterni, come smartphone e PC, che potremo eventualmente utilizzare per configurare il dispositivo stesso.

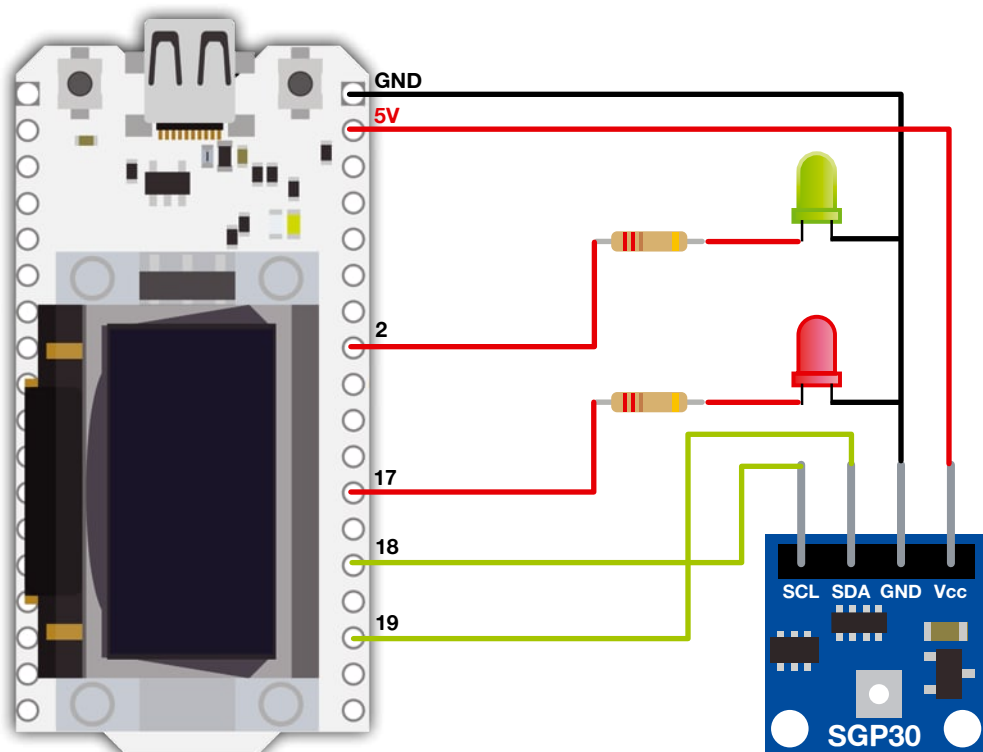
Infine, la scheda integra anche un caricabatterie LiPo on-board, che ci permetterebbe di associare una batteria a singola cella, utilissima per garantire alimentazione al dispositivo anche in assenza di tensione di rete.

In **Fig. 3** è riportata la piedinatura del nostro modulo.

Il modulo è disponibile a catalogo futura elettronica ed ulteriori informazioni possono essere reperite alla seguente pagina web: <https://futuranet.it/prodotto/modulo-esp32-con-display-oled-096/>.

Fig. 3
Pinout del modulo
WiFi kit 32.





IL SENSORE SGP30

Per ottenere la concentrazione di anidride carbonica e di TVOC nell'aria, abbiamo deciso di utilizzare una breakout board basata sul sensore SGP30 della Sensirion. Questo piccolissimo sensore permette di conoscere la concentrazione di anidride carbonica (così da indicarci quanto l'aria in un ambiente chiuso è viziata) e il TVOC (Total Volatile Organic Compounds) ossia la concentrazione in parti per

miliardo di particelle in sospensione. Un'immagine della breakout board è riportata in **Fig. 4**. Le caratteristiche hardware di dettaglio della breakout board sono queste:

- sensore: Sensirion SGP30 (TVOC and e CO₂);
- rilevamento TVOC (Total Volatile Organic Compounds): da 0 a 60.000 ppb (parti per miliardo);
- rilevamento CO₂: da 400 a 60.000 ppm (parti per milione);
- Frequenza di campionamento: 1 Hz;
- Interfaccia: I²C (indirizzo 0x58)
- Alimentazione: 3,3V o 5V.

La breakout board è disponibile, montata e collaudata, a catalogo futura elettronica e dettagli aggiuntivi possono essere reperiti alla seguente pagina web: <https://futura.net.it/prodotto/breakout-con-sensore-qualita-dellaria-sgp30/>.

CABLAGGIO DELL'HARDWARE

Vediamo adesso come realizzare in pratica il nostro apparato, partendo dalla costruzione fisica e quindi dal cablaggio dei componenti. Il collegamento tra la scheda sensore e i LED è piuttosto semplice, tanto da poter essere fatto mediante cavetti jumper maschio femmina tipo quelli per Arduino, ovvero appoggiandosi a una breadboard, come vedete in queste pagine.

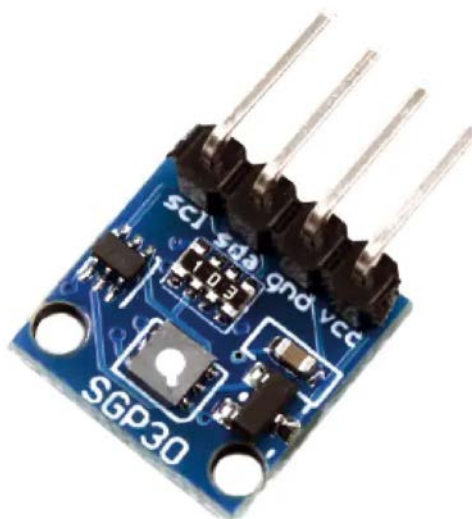


Fig. 4
La breakout board per il sensore SGP30.

FUNZIONE	WIFI KIT 32 PIN	SGP30 BREAKOUT PIN
SDA	19	sda
SCL	18	scl
Ground	GND	GND
3,3V	3V3	vcc

Tabella 1
Collegamenti
tra WiFi kit
32 e SGP30
breakout.

Il sensore SGP30 è una breakout board interfacciata tramite bus I²C, quindi necessita di un totale di quattro collegamenti, perché oltre alle linee di clock (SCL) e dati (SDA) del bus I²C, dobbiamo anche collegare le due linee di alimentazione (GND, ossia il negativo, oltre al positivo 3,3V).

I due LED, invece, sono interfacciati attraverso due GPIO standard del modulo WiFi kit 32; questi LED non possono essere pilotati direttamente dalle linee della scheda, onde evitare sovraccarichi dei GPIO, ma piuttosto bisogna porre loro in serie le opportune resistenze di limitazione della corrente, che abbiamo calcolato in 220 ohm ciascuna. Per agevolarvi nelle connessioni, nella pagina accanto proponiamo lo schema di cablaggio da utilizzare se volete interconnettere i vari elementi tramite cavetti saldati o jumper maschio-femmina. Inoltre nella **Tabella 1** sono riportati i collegamenti tra modulo WiFi kit 32 e SGP30 breakout board. Invece nella **Tabella 2** sono riportati i pin del modulo WiFi kit 32 utilizzati per i due LED di segnalazione.

REALIZZAZIONE SOFTWARE

Dopo aver visto i dettagli hardware del nostro sistema, passiamo al Software. Per lo sviluppo del software di controllo abbiamo deciso di utilizzare Micropython, e nello specifico il port per ESP32.

LED	WIFI KIT 32 PIN
Verde	2
Rosso	17

Tabella 2
Collegamenti
dei LED.

Micropython è un potente sistema per lo sviluppo software di applicazioni per microcontrollori moderni ed orientati all'IoT come l'ESP32, ed inoltre la grande quantità di librerie esistenti per la gestione di sensori, attuatori ed interfacce, permette di semplificare notevolmente questa fase dello sviluppo.

INSTALLAZIONE DI MICROPYTHON

Per utilizzare Micropython sulla WiFi Kit 32, la prima cosa da fare è installare l'interprete, dal momento che la WiFi kit 32 non lo presenta di default. Per farlo occorre prima di tutto procurarsi una versione del binario da flashare al seguente indirizzo web: <https://micropython.org/download/esp32/>. La versione utilizzata per lo sviluppo di questa applicazione è la v1.19 (2022-06-18). Scaricate il binario e salvatelo in una cartella del vostro PC. A questo punto occorre dotarsi anche di esptool, un tool utilizzato per flashare la memoria flash dell'ESP32 e distribuito gratuitamente da Expressif (l'azienda produttrice dell'ESP32 e dell'ESP8266). Potete scaricare esp-

```

C:\WINDOWS\system32\cmd.exe
.\esptool\esptool.py --chip esp32 --port COM3 --baud 460800 write_flash -z 0x1000 .\fw_upython\esp32.bin
esptool.py v4.2-dev
Serial port COM3
Connecting.....
Chip is ESP32-D0WQ6-V3 (revision 3)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 26MHz
MAC: 78:21:84:9f:3e:80
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Configuring flash size...
Flash will be erased from 0x00001000 to 0x00161fff...
Compressed 1445632 bytes to 925476...
Wrote 1445632 bytes (925476 compressed) at 0x00001000 in 23.4 seconds (effective 494.7 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

```

Fig. 5
Output della
console di
windows
dopo il
flashing di
micropython.

Fig. 6
Architettura software.

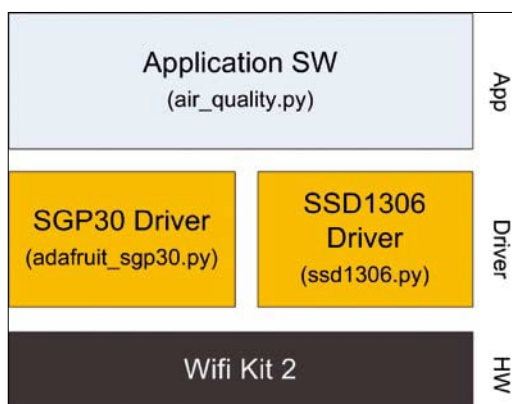


Fig. 7
Configurazione del port di micropython per ESP32 in Thonny.

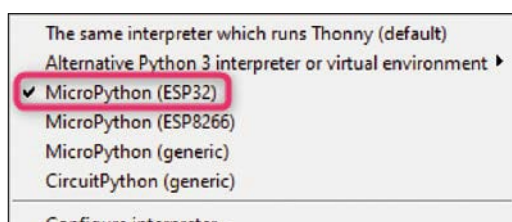


Fig. 8
Impostazione manuale dell'interprete e della porta COM in Thonny.

tool dal seguente indirizzo web: <https://github.com/espressif/esptool>. A questo punto dovete cancellare la memoria del dispositivo e poi "flashare" (caricare nella Flash Memory) il binario di Micropython.

CANCELLAZIONE

Per prima cosa collegate la WiFi kit 32 al vostro PC tramite un cavo USB micro. Al dispositivo dovrebbe

venire associata una porta COM virtuale (ad esempio COM3, ma il numero esatto dipende da quante porte avete già sul vostro sistema).

A questo punto mettete la scheda in modalità programmazione premendo e mantenendo premuto il pulsante PRG e premendo e rilasciando il pulsante RST. Ora non vi resta che usare lo script esptool.py per cancellare la memoria. Potete inserire il path dello script nel vostro system path, oppure operare direttamente da quella cartella e cancellare la memoria con il seguente comando (valido in ambiente Windows):

```
python esptool.py --port <port_num> erase_flash
```

Sostituite **<port_num>** con la porta COM virtuale assegnata al vostro dispositivo (per esempio la COM3).

FLASH DELL'INTERPRETE MICROPYTHON

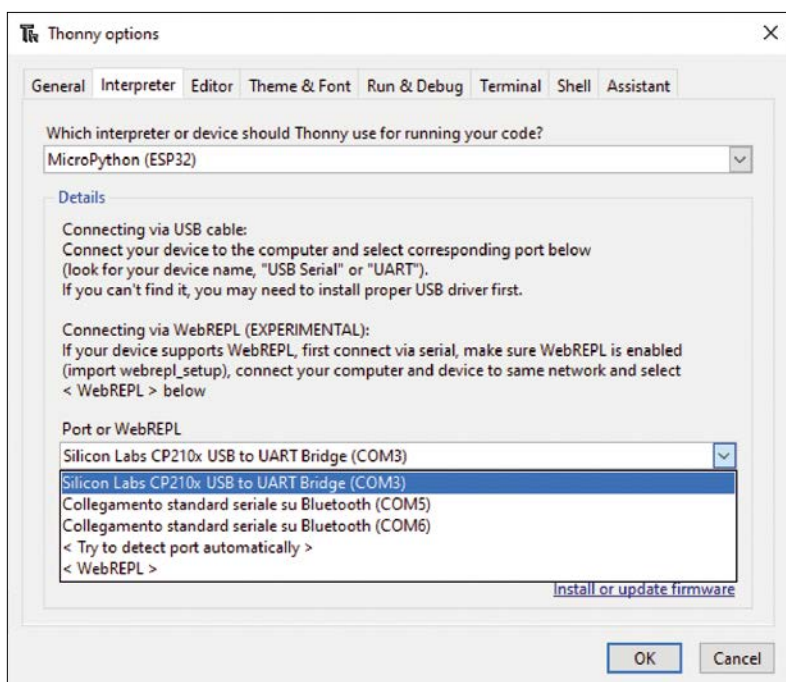
A questo punto potete flashare micropython (nell'ESP32 l'interprete va flashato all'indirizzo 0x1000). Potete usare il seguente comando di console, sempre sotto Windows:

```
python esptool.py --chip esp32 --port <port_num> --baud 460800 write_flash -z 0x1000 esp32.bin
```

Come prima sostituite **<port_num>** con la porta COM virtuale assegnata al vostro dispositivo e **esp32.bin** con il nome (ed eventualmente il path, se non è lo stesso da cui operate), del binario scaricato il precedenza. In **Fig. 5** è riportato il risultato di questa operazione.

SOFTWARE DI CONTROLLO

A questo punto possiamo passare alla gestione del software di controllo della nostra scheda. Come abbiamo accennato, ciò che vogliamo fare è acquisire i dati relativi ai valori di CO₂ e TVOC forniti dal sensore SGP30, visualizzare questi dati su display OLED; leggere da file system le soglie utente per i valori limite di CO₂ (con relativa isteresi). Gestire due LED di segnalazione (rosso e verde) per indicare se il livello di CO₂ è superiore alla soglia massima consentita. Per implementare questo software ci occorrono innanzitutto degli opportuni driver per la gestione del display OLED e del sensore SGP30. Fortunatamente entrambi i driver sono disponibili, sviluppati da adafruit. Potete procurarveli scaricandoli ai seguenti link web:



▼ Listato 1 - Inizializzazione del sistema

```
# -----  
# - SYSTEM INIT  
# -----  
# LEDs  
led_green = Pin(2, Pin.OUT)  
led_red = Pin(17, Pin.OUT)  
led_green.on()  
led_red.off()  
  
# Initialize I2C buses  
s_i2c = I2C(scl=Pin(18), sda=Pin(19), freq=100000)  
d_i2c = SoftI2C(scl=Pin(15), sda=Pin(4),  
freq=100000)  
  
# OLED enable pin  
oled_sts = Pin(16, Pin.OUT)  
oled_sts.on()  
  
# OLED initialization  
oled_width = 128  
oled_height = 64  
oled = ssd1306.SSD1306_I2C(oled_width, oled_height,  
d_i2c)  
  
# Air quality sensor init  
sgp30 = adafruit_sgp30.Adafruit_SGP30(s_i2c)  
  
# Initialize SGP-30 internal drift compensation  
algorithm.  
sgp30.iaq_init()  
  
# Wait 15 seconds for the SGP30 to properly  
initialize  
user_message('Sensor Init...')  
time.sleep(SENS_INIT_DELAY_S)  
  
# User threshold for LEDs indication  
get_user_threshold()  
  
# Sensor Calibration  
get_sensor_baseline()  
  
# Store the time at which last baseline has been  
saved  
baseline_time = time.time()
```

maggiormente utilizzati, che permette di sviluppare sia applicazioni Python standard, che micropython. Thonny può essere scaricato liberamente dal seguente link: <https://thonny.org/>.

L'interfacciamento con micropython è molto semplice: dopo aver collegato la scheda WiFi kit 32 alla porta USB del PC, aprite Thonny e cliccate sul bordo in basso a destra: nel menù a tendina che si apre dopo aver fatto click, tra i vari interpreti disponibili selezionate "ESP32", come illustrato in **Fig. 7**. Se l'interprete non dovesse essere presente, selezionate "Configure Interpreter..." e indicate manualmente l'interprete e la porta COM da utilizzare, come illustrato in **Fig. 8**.

Passiamo adesso al dettaglio del software applicativo, contenuto interamente all'interno del file `air_quality.py`, che si presenta ripartito in due fasi distinte:

- System Setup
- Main Loop

La prima fase, come si può facilmente intuire dal nome, costituisce quella di inizializzazione del sistema, mentre la seconda è il loop principale nel quale avviene la misurazione, la stampa dei valori su display OLED e la segnalazione dello stato tramite LED.

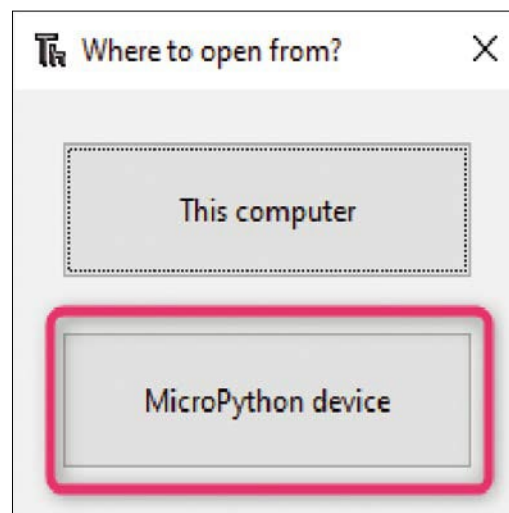
La prima fase, il cui codice è riportato nel **Listato 1**, ha il compito di inizializzare tutte le periferiche del sistema, come i LED, i due bus I²C (uno utilizzato per l'interfacciamento del display OLED, l'altro per la gestione del sensore SGP30), e di recuperare da file system i valori della baseline del sensore SGP30 e della soglia di CO₂ impostata dall'utente. Questi file sono dei file di testo che il software

- per il driver dei display OLED: <https://github.com/adafruit/micropython-adafruit-ssd1306/blob/master/ssd1306.py>;
- per il driver SGP30: <https://github.com/alexmrqt/micropython-sgp30>.

Lo schema architetturale del software che ci proponiamo di realizzare è rappresentato nella **Fig. 6**. Il progetto completo risulta quindi composto da questi tre file:

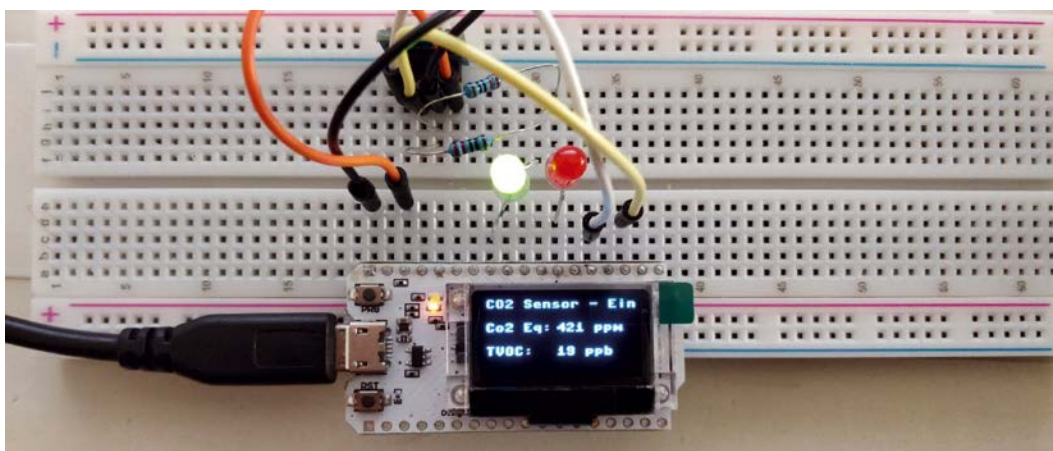
- `Air_quality.py` (file principale dell'applicazione);
- `Ssd1306.py` (driver display OLED);
- `Adafruit_sgp30.py` (driver del sensore sgp30).

Come IDE di sviluppo per questa applicazione utilizzeremo Thonny, uno degli IDE per Python



◀ **Fig. 9**
Accesso alla
memoria del
dispositivo da
Thonny.

➔ **Fig. 10**
Il prototipo cablato
su breadboard per
le prove.



si aspetta di trovare nel file system. I valori della baseline del sensore sono valori di calibrazione che migliorano la precisione della lettura e vengono automaticamente creati dal software se non sono presenti. I file contenenti la soglia di allarme per il valore di CO₂ invece devono essere creati dall'utente e se non presenti vengono caricati i valori predefiniti. Per impostare questa soglia vanno creati i seguenti due file:

- `co2eq_th.txt` = contiene il valore di allarme della lettura di CO₂, in ppm (parti per milione);
- `hist_th.txt` = contiene l'isteresi della soglia precedente, sempre in ppm.

La creazione di questi file è molto semplice: in ambiente Thonny, con la board collegata via USB, selezionate l'icona a forma di file oppure File > New.

Ripetete questa operazione due volte, per creare due file e scrivete i valori all'interno (ad esempio 450 nel file `co2eq_th.txt` e 10 nel file `hist_th.txt`, in modo da avere una soglia di 450ppm di CO₂ e una isteresi di 10ppm).

Adesso cliccate sull'icona a forma di floppy disk (o selezionate File > Save) e nel pop-up che compare, selezionate "Micropython Device", come illustrato in **Fig. 9**.

Questa operazione permette di accedere al file system del dispositivo e salvare i file creati in precedenza. In questo modo è possibile impostare il valore della soglia di CO₂ del caso.

Una volta completata la fase di inizializzazione, il software passa al loop principale, rappresentato nel **Listato 2**.

Qui vengono eseguite le seguenti operazioni in sequenza. Tramite la funzione `iaq_measure()` della libreria `adafruit_sgp30` (`sgp30.iaq_measure()`), vengono acquisiti i valori di CO₂ e TVOC, salvati rispettivamente nelle due variabili `co2eq` e `tvoc`. A questo punto, nella riga successiva, i valori vengono stampati sul display OLED, tramite la funzione `oled_data()`. Successivamente, tramite una serie di istruzioni `if...elif`, vengono gestiti i LED

⬇ **Listato 2 - Main Loop**

```
# -----
# - MAIN LOOP
# -----
while True:
    # Get CO2 and TVOC data
    co2eq, tvoc = sgp30.iaq_measure()

    # Print values on Display
    oled_data(co2eq, tvoc)

    # Manage LEDs
    if (co2eq > co2eq_th):
        led_green.off()
        led_red.on()
    elif (co2eq < (co2eq_th - hysteresis)):
        led_green.on()
        led_red.off()

    # Update sensor baseline
    update_sensor_baseline()

    # Wait for the new sampling time
    time.sleep(SAMPLING_TIME_S)
```

⬇ **Listato 3 - Parametri di configurazione**

```
-----
# - USER TH DEFAULT VALUES
-----
co2eq_th = 450
hysteresis = 10
-----
# - TIMING CONST
# -----
SAMPLING_TIME_S = 60
SENS_INIT_DELAY_S = 15
```

di segnalazione. In particolare, se il valore immagazzinato nella variabile **co2eq** risulta superiore a quello della soglia **co2eq_th**, impostata dall'utente, si accende il LED rosso e si spegne il LED verde, altrimenti, se il valore risulta inferiore a quello di soglia meno l'isteresi (**co2eq_th - hysteresis**), allora viene spento il LED rosso ed acceso quello verde. Terminata questa fase viene invocata la funzione **update_sensor_baseline()**, per aggiornare la baseline del sensore.

Per garantire un funzionamento ottimale del sensore, il software aggiorna la baseline la prima volta dopo 12 ore di funzionamento e poi ogni ora, come indicato nel datasheet del costruttore dell'SGP30. A questo punto il loop va in pausa per 60 secondi, prima di eseguire un nuovo ciclo di acquisizione dati.

Il progetto software completo di questa applicazione, corredato di descrizione e di tutte le librerie micropython necessarie può essere scaricato dal seguente repository github:
<https://github.com/ffich/IndoorAirQuality>.

Per testare il software dovrete salvare i due file contenenti i driver nella memoria della scheda WiFi kit 32, usando la stessa procedura illustrata per salvare i file utente, avendo l'accortezza di mantenere gli stessi nomi dei file, e di non dimenticare l'estensione .py. Il file principale può essere fatto girare direttamente da Thonny, oppure salvato sulla memoria del dispositivo rinominandolo **main.py**, accortezza necessaria se si desidera far funzionare il dispositivo in modalità "standalone", ossia non collegato al PC (infatti il file **main.py** è il file che viene automaticamente cercato dall'interprete micropython all'avvio).

In **Fig. 10** è proposta una foto del dispositivo cablato su breadboard mentre acquisisce i dati di qualità dell'aria ed attiva i LED.

CONFIGURAZIONE

Il software presenta anche alcune opzioni di configurazione che possono essere modificate dall'utente, tutte presenti nel file **air_quality.py**. In particolare è possibile:

- modificare i valori predefiniti della soglia di allarme CO₂ e relativa isteresi, modificando i due parametri **co2eq_th** e **hysteresis**, impostati per default a 450 e 10, come rappresentato nel **Listato 3** (agendo sui valori predefiniti non è necessario salvare su file system i valori di questi due parametri);



 Il prototipo in funzione.

- modificare il tempo di attesa iniziale per la stabilizzazione del sensore ed il tempo di campionamento (**SENS_INIT_DELAY_S** e **SAMPLING_TIME_S**), impostati per default a 15 e 60 secondi, come rappresentato sempre nel **Listato 3** (noi suggeriamo comunque di non modificare queste impostazioni).

CONCLUSIONI

In questo articolo abbiamo descritto in dettaglio la realizzazione di un dispositivo per il monitoraggio della qualità dell'aria. Il dispositivo, leggendo un sensore di qualità dell'aria (SGP30), è in grado di mostrare su display le concentrazioni di CO₂ e TVOC. Inoltre, leggendo un file utente che contiene la soglia di tolleranza per la concentrazione di CO₂ è in grado di segnalare visivamente tramite un LED, di operare un ricambio d'aria nell'ambiente. Data la ricchezza di periferiche aggiuntive della piattaforma utilizzata, è possibile espandere questo dispositivo con funzionalità IoT, come ad esempio una pagina web che permetta di consultare i dati in remoto e segnalazioni via e-mail o notifica telegram. □



Cosa occorre?

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

La scheda con ESP32 e OLED (cod. ESP32OLED) è in vendita a Euro 29,50, il sensore qualità dell'aria (cod. SGP30CO2) è disponibile a Euro 19,50

I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

La più ampia selezione di componenti elettronici™

A magazzino e pronti per la spedizione



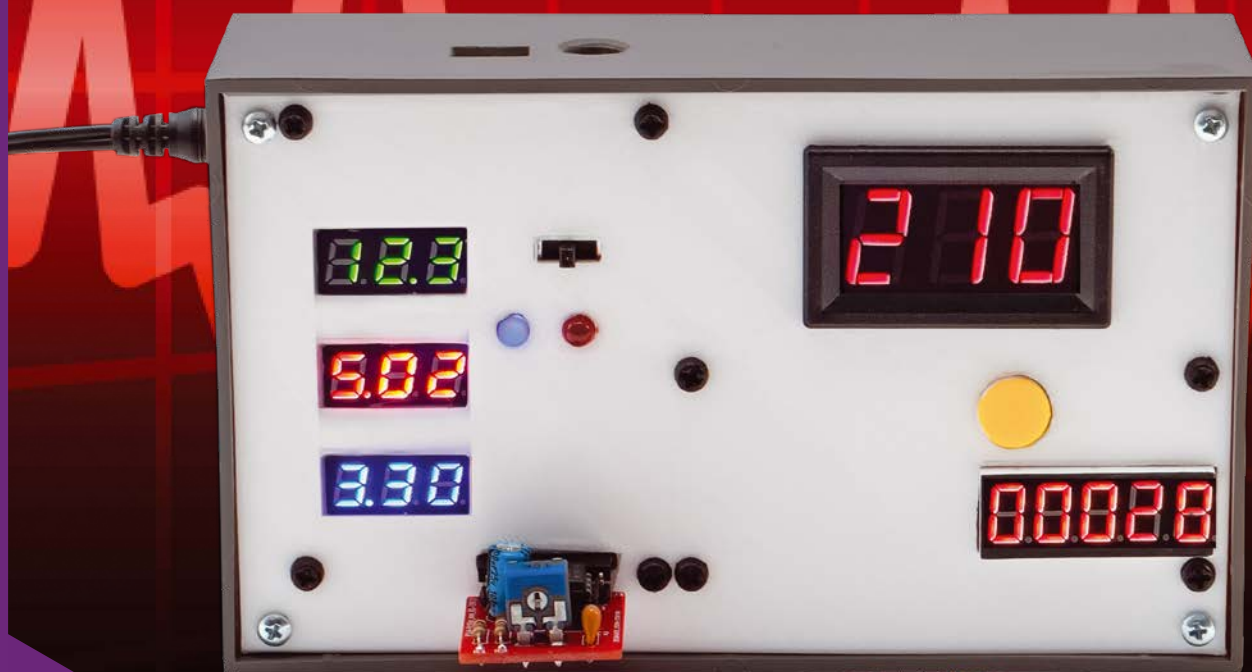
mouser.it



**MOUSER
ELECTRONICS**

555 ANALYZER

del Prof. MICHELE MENNITI



Usiamo
l'Analizzatore del
timer 555 per testare
una prima serie di
circuiti elettronici di
uso comune.
Seconda puntata.

Dopo aver presentato, nello scorso numero della nostra rivista, l'analizzatore di circuiti basati sul timer 555, in questa seconda puntata lo useremo per testare il funzionamento di cinque circuiti, basati sul componente 555 operante in configurazione astabile; più esattamente si tratta dei seguenti circuiti, che elenchiamo antepo-
nendo la sigla identificativa:

- P1A, generatore di onda quadra a frequenza variabile alimentato a 5V;
- P2A, generatore di onda quadra con Duty-Cycle al 50% alimentato a 5V;
- P3A, generatore di onda quadra con Duty-Cycle variabile tra 1 e 99% alimentato a 5V;

 **Tabella 1**
Tabella funzionale del 555.

TRIGGER 2	SOGLIA 6	S	R	SCARICA 7	USCITA 3
$< 1/3 V_{CC}$	Irrilevante	H	X	OFF	H
$> 1/3 V_{CC}$	$> 2/3 V_{CC}$	L	H	ON	L
$> 1/3 V_{CC}$	$< 2/3 V_{CC}$	L	L	non cambia	non cambia

- P4A, sirena bitonale manuale alimentata a 12V;
- P5A, astabile quarzato con alimentazione a 12V.

Nella prossima puntata vedremo altri sette circuiti basati sulle configurazioni monostabile e bistabile del 555. Per facilitarci il compito, abbiamo assegnato sigle che identificano il progetto e la tipologia di configurazione, quindi P1A significa circuito 1 Astabile e via di seguito.

Come potete vedere nei nostri circuiti non abbiamo usato la tensione di 3,3V, ma nell'Analizzatore l'abbiamo prevista, nel caso vogliate testare uno schema con un 555 in versione CMOS, a basso consumo.

Prima di passare alla descrizione ed al test dei circuiti, facciamo un rapido riepilogo delle tre configurazioni del timer 555. Nella **Tabella 1** abbiamo riportato la tabella funzionale dei pin TRIGGER e SOGLIA del comparatore del 555, dei suoi effetti sul SET/RESET del flip-flop interno e, di conseguenza, sui pin SCARICA e USCITA.

MULTIVIBRATORE ASTABILE

La **Fig. 1** mostra la tipologia base del multivibratore astabile, in cui i tre componenti fondamentali sono i due resistori RA e RB ed il condensatore CX; il C1 (10 nF) funge da filtro antidisturbo.

I due ingressi del comparatore, TRIGGER e SOGLIA, sono collegati assieme, e ciò comporta che siano costantemente allo stesso livello di tensione. Al momento dell'accensione del circuito CX è scarico, quindi il TRIGGER si trova ad un livello inferiore ad $1/3 V_{CC}$ e l'USCITA è alta (H). Poiché contemporaneamente il pin SCARICA è scollegato dalla massa (OFF), inizia la carica del condensatore CX, mediante la corrente che passa attraverso RA ed RB, e la tensione sugli ingressi inizia gradualmente a salire. Nel momento in cui la tensione supera $1/3 V_{CC}$ si verifica la condizione $\text{TRIGGER} > 1/3 V_{CC}$ e $\text{SOGLIA} < 2/3 V_{CC}$, quindi non ci sono variazioni. Quando invece essa supera $2/3 V_{CC}$ si verifica la condizione $\text{TRIGGER} > 1/3 V_{CC}$ e $\text{SOGLIA} > 2/3 V_{CC}$, il comparatore attiva il RESET del flip-flop interno e l'USCITA

diventa bassa (L); inoltre ora il pin SCARICA viene collegato alla massa (ON), CX inizia a scaricarsi attraverso la sola resistenza RB, e la tensione sugli ingressi ricomincia a diminuire. Nell'intervallo di tempo in cui la tensione scende sotto $2/3 V_{CC}$ ma resta sopra $1/3 V_{CC}$, si verifica la condizione $\text{TRIGGER} > 1/3 V_{CC}$ e $\text{SOGLIA} < 2/3 V_{CC}$, quindi non ci sono variazioni.

Quando invece la tensione scende nuovamente sotto $1/3 V_{CC}$, si verifica la condizione $\text{TRIGGER} < 1/3 V_{CC}$, l'USCITA ritorna alta (H), il pin SCARICA viene scollegato dalla massa (OFF) ed il condensatore CX inizia a caricarsi nuovamente. È quindi evidente che si ha un funzionamento a ciclo continuo, con alternanza degli stati alto e basso dell'USCITA, le due condizioni sono instabili, ed è questa la ragione per la quale questo tipo di configurazione viene definita ASTABILE.

Questo significa che ci troviamo in presenza di un circuito oscillatore con due stadi H-L ben definiti, e quindi parliamo di un'onda quadra.

MULTIVIBRATORE MONOSTABILE

La **Fig. 2** mostra la tipologia base del multivibratore monostabile, in cui i due componenti fondamentali sono il resistore RX, il condensatore CX e lo switch S1; il C1 (10 nF) è il solito filtro antidisturbo, il resistore R1 ha la funzione di pull-up. Da notare che in questo caso l'ingresso SOGLIA è scollegato dal TRIGGER ed è invece collegato al pin SCARICA.

Al momento dell'accensione il TRIGGER si trova ad un livello maggiore di $1/3 V_{CC}$ a causa del resistore di pullup e l'USCITA è bassa (L).

Contestualmente il pin SCARICA è collegato a massa, mantiene a livello L anche il pin SOGLIA ed impedisce al condensatore CX di caricarsi, in quanto la corrente che attraversa R1 finisce direttamente a massa. Nel momento in cui si preme il pulsante S1, il TRIGGER viene cortocircuitato a massa, quindi va ad un livello inferiore a $1/3 V_{CC}$, l'USCITA diventa alta (H); il pin SCARICA viene scollegato dalla massa e la corrente che attraversa RX inizia a caricare il condensatore CX e la tensione sulla SOGLIA inizia gradualmente

a salire. Rilasciando il pulsante il TRIGGER ritorna a livello maggiore di $1/3 V_{cc}$. Durante tutta la fase di carica l'USCITA si mantiene alta, in quanto la combinazione $TRIGGER > 1/3 V_{cc}$ e $SOGLIA < 2/3 V_{cc}$ non provoca modifiche di stato del LATCH.

Nel momento in cui il condensatore è abbastanza carico da avere ai suoi capi (e quindi sulla SOGLIA) una tensione maggiore di $2/3 V_{cc}$, avviene la nuova commutazione dell'USCITA a livello basso (L) ed il condensatore si scarica istantaneamente attraverso il pin SCARICA, ora collegato nuovamente a massa; tale condizione resterà stabile fino ad una nuova pressione del pulsante. Riepilogando, mentre lo stato di USCITA bassa è stabile, quello di USCITA alta è instabile, in quanto dipendente dalla costante di tempo risultante dai valori di RX e CX. Ecco la ragione per la quale questo tipo di configurazione viene definita MONOSTABILE.

MULTIVIBRATORE BISTABILE

La Fig. 3 mostra la tipologia base del multivibratore bistabile, in cui i due componenti fondamentali sono i microswitch S1 e S2; il C1 (10 nF) è il solito elemento di filtro antisturbo, mentre i resistori R1 e R2 hanno la funzione di pull-up.

Anche in questo caso l'ingresso SOGLIA è scollegato dal TRIGGER, ma questa volta il pin SCARICA resta inutilizzato, poiché il circuito non prevede l'uso di condensatori, infatti

questa configurazione non prevede alcuna temporizzazione, ma solo due stati logici stabili. Al momento dell'accensione il TRIGGER si trova ad un livello maggiore di $1/3 V_{cc}$, quindi l'USCITA si trova inizialmente al livello L, nel quale resta in modo stabile, in mancanza di eventi; la SOGLIA si trova ad un livello minore di $2/3 V_{cc}$, e questa combinazione, in base alla terza riga della Tabella 1, non provoca commutazioni.

Se si preme il pulsante S2 la SOGLIA viene portata ad un livello maggiore di $2/3 V_{cc}$, e ci si trova nella condizione prevista dalla riga 2 della tabella,

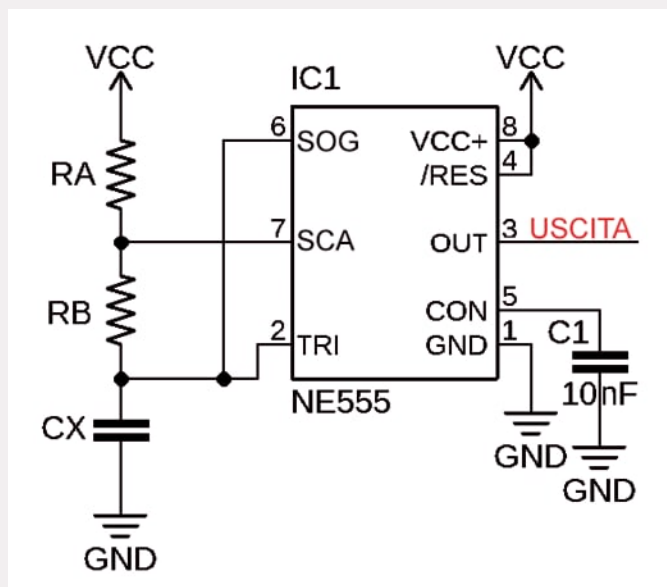


Fig. 1
Multivibratore
astabile.

Fig. 2
Multivibratore
monostabile.

Fig. 3
Multivibratore
bistabile.

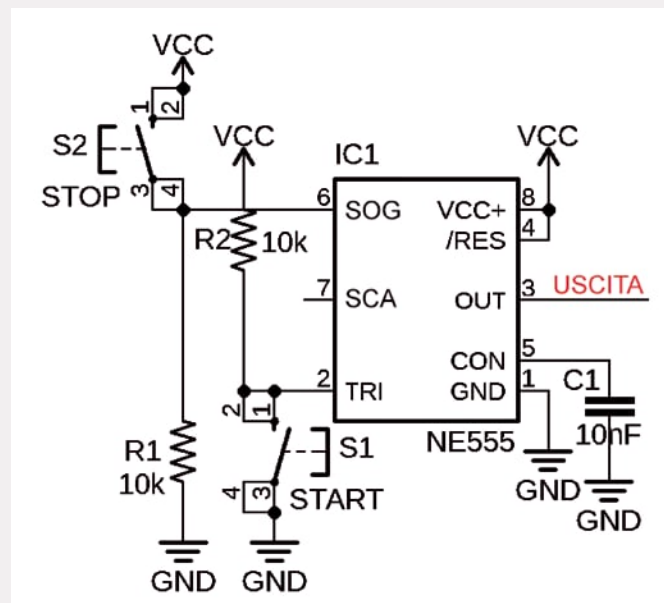
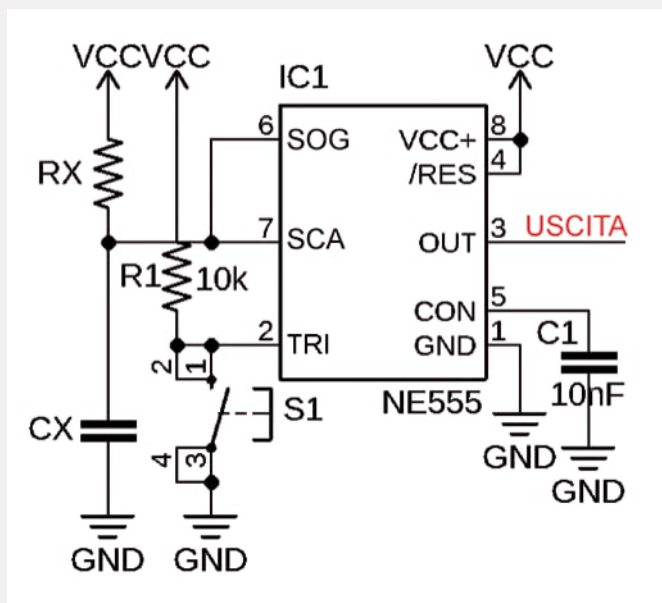


Tabella 2
Impostazioni del MINI DSO 138 e del deviatore S1.

Circuito	Impostazioni MINI DSO 138						Posizione Deviatore S1		
	CPL	SEN1	SEN2	Timebase	Trigger	Misure	1 (TRIG)	2(SOG)	3(SCA)
P1A	DC	1V	x2	0.2s	NORM	sì	TRIG/SOG		SCA
P2A	DC	1V	x2	0.5s	NORM	sì	TRIG/SOG		SCA
P3A	DC	1V	x2	0.5ms	AUTO	sì	=		=
P4A	DC	1V	x5	0.5ms	AUTO	sì	=		=
P5A	DC	1V	x5	50µs	AUTO	sì	=		=

quindi lo stato dell'USCITA non cambia. Se invece si preme il pulsante S1 (START) il TRIGGER viene portato ad un livello minore di $1/3 V_{cc}$, e ci si trova nella condizione della riga 1 della tabella, quindi l'USCITA diventa alta (H). Rilasciando S1 la combinazione degli ingressi ridiventa quella della riga 3, e quindi anche questo stato è stabile. Questa è la ragione per la quale questo tipo di configurazione viene definita bistabile. A questo punto possiamo passare alla descrizione dei circuiti test; per ognuno di essi riportiamo lo schema elettrico, il piano di montaggio e l'elenco dei componenti, quindi ne descriveremo il funzionamento, dopo averlo inserito nel connettore del nostro Analizzatore. Abbiamo approntato la **Tabella 2** per suggerirvi come impostare il mini DSO (cod. DSO138MINI) ed il deviatore per il voltmetro esterno (posizioni della levetta numerate da sinistra a destra). Per quanto riguarda il mini DSO, nei casi in cui è prevista la visualizzazione delle misure, conviene portare il cursore sinistro (giallo) nella parte bassa del display per evitare sovrapposizioni. Riteniamo utile fornirvi alcune indicazioni utili per l'uso del trigger: la modalità in genere va su AUTO; se il segnale scorre sempre si può provare a portarlo su NORM. Invece la posizione SING è utile se si verificano eventi periodici molto distanziati nel tempo, in questo caso, appena ne viene catturato uno il DSO va in STOP, il ripristino avviene con la pressione del pulsante OK; nelle modalità AUTO e NORM il segnale può comunque essere bloccato con la pressione diretta del pulsante OK. Il segnale fermo

permette di leggere in modo stabile le misure, se sono attive.

La sorgente del Trigger deve essere sempre su "Int"; il fronte in genere può essere indifferentemente impostato su falling o rising. La freccetta del livello in genere va posizionata a metà dell'ampiezza segnale. Naturalmente nulla vi vieta di provare impostazioni alternative a quelle da noi indicate. Per quanto riguarda il deviatore, la tabella indica quale tensione viene monitorata in base alla posizione della levetta; naturalmente nei casi in cui due pin del 555 sono collegati tra loro, le rispettive posizioni mostreranno sempre gli stessi valori; la presenza di un doppio trattino indica che in quella particolare configurazione il pin non è usato oppure la frequenza di variazione è troppo elevata, quindi eventuali valori mostrati non sono da tenere in considerazione.

Una nota finale riguarda il voltmetro esterno, le cui letture non sono istantanee; nei circuiti astabili, per ottenere una visione chiara dell'andamento delle variazioni di tensione bisognerà far sì che la frequenza di lavoro sia molto bassa, agendo sul valore del condensatore. Il problema non si porrà nel caso dei circuiti monostabili o bistabili. In ogni caso, come spiegato nella prima puntata, si può sempre ricorrere ad uno strumento analogico, nel quale il movimento della lancetta rende sicuramente molto di più.

MONTAGGIO DEI CIRCUITI TEST

Il montaggio di questi piccoli circuiti non richiede

Fig. 4
I circuiti di test montati.



particolari attenzioni, ci limitiamo qui a darvi alcune indicazioni generali.

Come sempre conviene iniziare dai componenti più bassi, come resistori e diodi, poi lo zoccolo ed i condensatori, ed infine gli altri componenti. Tutti i circuiti prevedono una barretta di pin strip maschio a 8 pin, che permetterà l'inserimento del circuito nella presa header della base dell'Analizzatore; per i circuiti che prevedono un connettore per il collegamento di un componente esterno, daremo indicazioni specifiche. Infine, nei circuiti che prevedono un condensatore elettrolitico per la temporizzazione, abbiamo previsto due pad aggiuntivi: potete usare il valore da noi indicato, saldando direttamente, oppure potete saldare due pin header nei pad e poi inserire diversi valori di capacità, per testare le diverse temporizzazioni. Per quanto riguarda i calcoli delle temporizzazioni e dei componenti di riferimento, nella sezione download del sito www.elettronica.it troverete un file Excel con tutte le necessarie formule. In **Fig. 4** vedete una panoramica dei circuiti utilizzati.

PROGETTO 1

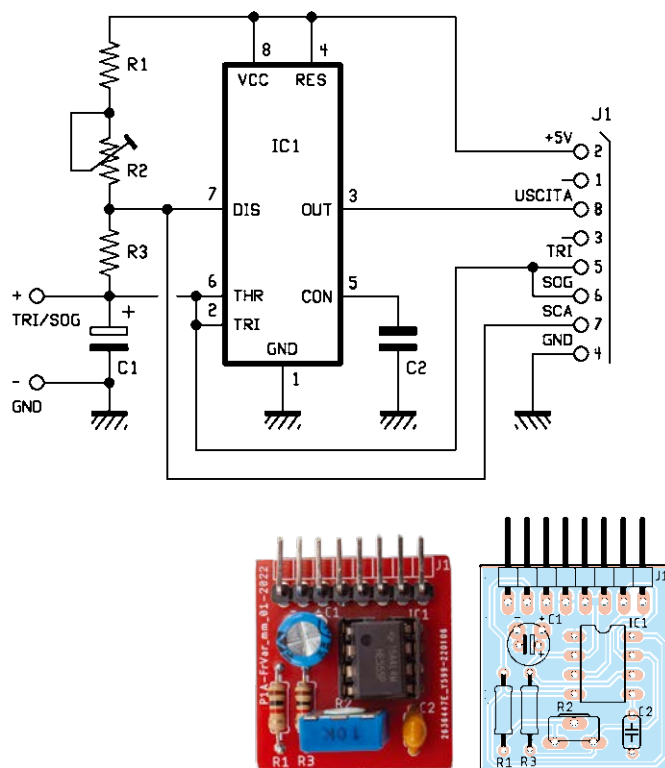
Generatore di onda quadra a frequenza variabile alimentato a 5V

Questo circuito serve per generare un segnale ad onda quadra a frequenza variabile, il range di frequenza dipende fondamentalmente dal condensatore C1, ma anche dai due resistori R1 e R3 e dal trimmer R2. La variazione della frequenza avviene ruotando il cursore del trimmer R2. Usando il calcolatore che abbiamo approntato espressamente (ma se ne trovano diversi anche sulla rete) è possibile progettare un circuito che operi nel range di frequenza desiderata.

All'accensione il condensatore C1 (CX) è scarico, il pin TRIGGER (collegato assieme a SOGLIA) si trova quindi a livello minore di $1/3 V_{cc}$ e, come rappresentato nella tabella 1, l'USCITA è alta (LED blu acceso). Poiché il transistor collegato al pin SCARICA è interdetto, C1 inizia a caricarsi mediante il resistore R1 ed il trimmer R2 (RA) ed R3 (RB), la tensione sul pin TRIGGER aumenta, nel momento in cui il pin TRIGGER supera i $3,34 V$ ($2/3 V_{cc}$) si verifica la combinazione TRIGGER > $1/3 V_{cc}$ e SOGLIA > $2/3 V_{cc}$, quindi il comparatore del 555 attiva il RESET del flip-flop interno e avviene il cambio di stato, l'USCITA diventa bassa (LED rosso acceso), il transistor interno va in saturazione, il pin SCARICA viene collegato a massa e C1 inizia a scaricarsi, tramite R3. Quando la tensione sul

PROGETTO 1

GENERATORE DI ONDA QUADRA A FREQUENZA VARIABILE



Elenco Componenti:

R1, R3: 1 kohm
 R2: 10 kohm trimmer M.V.
 C1: 100 µF 16 V elettrolitico
 C2: 10 nF poliestere o ceramico
 IC1: NE555
 J1: strip maschio 8 vie 90°

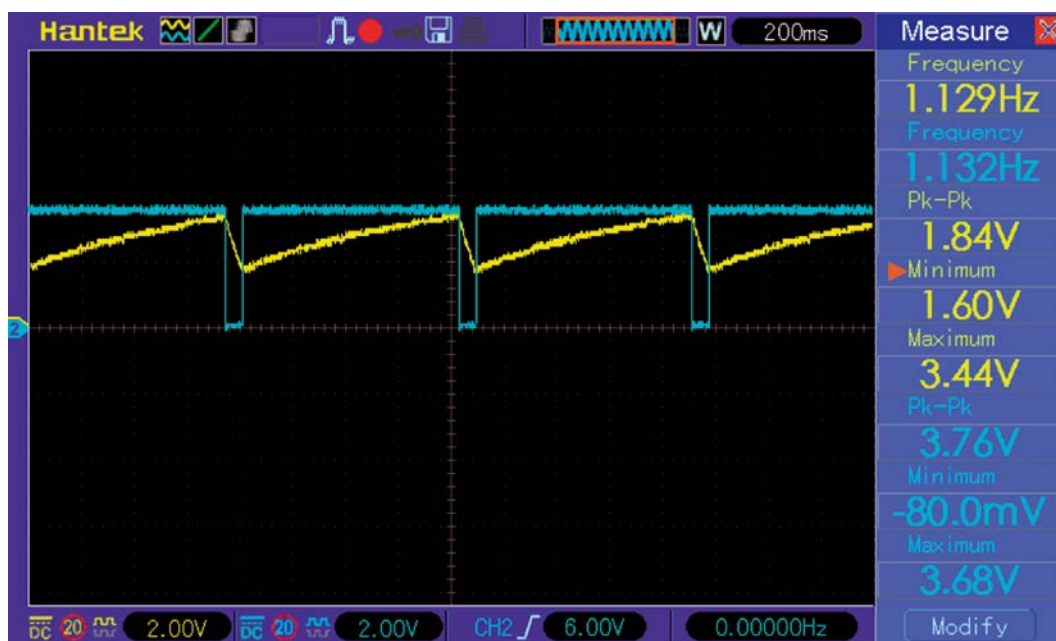
Varie:
 - zoccolo 4+4 pin
 - Circuito stampato S1635_P1A (27,5 x 25,5 mm)

pin TRIGGER scende sotto $1,67 V$ ($1/3 V_{cc}$), il comparatore attiva il SET del flip-flop interno e l'USCITA torna nuovamente alta (LED blu acceso). Ruotando il cursore del trimmer si modifica il valore della sua resistenza, permettendo di ottenere una variazione della coppia R1-R2 tra 1.000 ohm (trimmer al minimo, 0 ohm) e 11.000 ohm (trimmer al massimo, 10.000 ohm). Applicando la formula

$$f = \frac{1,44}{(R1+R2+2R3) \times C1}$$



Fig. 5
Funzionamento
del circuito astabile
P1A.



possiamo stabilire che il range di frequenza in uscita dal 555 va da

$$f = \frac{1,44}{(1000+0+2000) \times 0,0001} = 4,8 \text{ Hz}$$

a

$$f = \frac{1,44}{(1000+10000+2000) \times 0,0001} = 1,1 \text{ Hz}$$

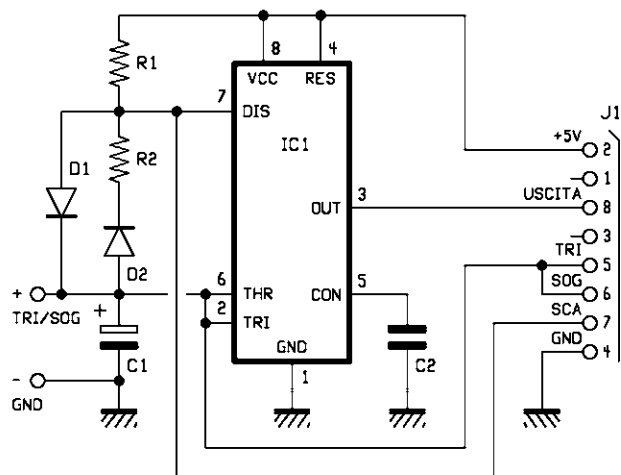
Ricordiamo che la resistenza va espressa in Ohm

(10 kohm = 10.000 ohm) e la capacità in Farad (100 μ F = 0,0001 F).

Sostituendo C1 con altri valori naturalmente cambiano sia la frequenza che le tempistiche; in particolare, se C1 = 10 μ F il range di frequenza varia tra 11 Hz e 48 Hz circa, invece con C1 = 1000 μ F si ottiene un range tra 0,1 Hz e 0,5 Hz circa. Il Duty Cycle, ossia il rapporto temporale tra la fase HIGH e la fase LOW dell'onda quadra, varia dall'85-90% circa (frequenza minima) al 64-68%

PROGETTO 2

GENERATORE DI ONDA QUADRA CON DUTY CYCLE AL 50%

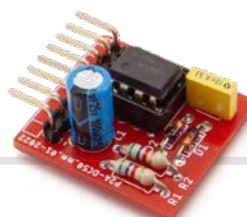
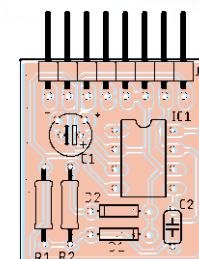
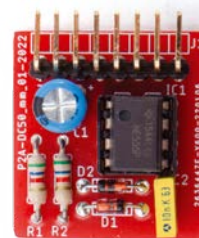


Elenco Componenti:

R1, R2: 5,6 kohm
C1: 100 μ F 16 V elettrolitico
C2: 10 nF poliestere o ceramico
D1, D2: 1N4148
IC1: NE555
J1: strip maschio 8 vie 90°

Varie:

- zoccolo 4+4 pin
- circuito stampato S1635_P2A (27,5 x 25,5 mm)



(frequenza massima); ciò comporta una disparità dei tempi di accensione dei due LED, in ogni caso il LED blu resterà acceso per un tempo maggiore rispetto a quello rosso, anche se alla frequenza minima questa cosa sarà molto più evidente. Per la corretta regolazione della frequenza ci si potrà basare sulla lettura frequenzimetro.

Per poter meglio comprendere cosa avviene in questo circuito, abbiamo impostato una frequenza di 1,1 Hz e rilevato il segnale in USCITA (celeste) e quello sui pin TRIGGER/SOGLIA (giallo) con un DSO professionale (Fig. 5).

Il segnale giallo quindi mostra l'andamento di carica (segnale in salita) e scarica (segnale in discesa) del condensatore, mentre quello celeste il corrispondente stato dell'USCITA. Poiché la fase di carica del condensatore C1 è molto più lunga di quella di scarica, il Duty Cycle è molto elevato. Il cambio di stato avviene quando la tensione sui pin TRIGGER/SOGLIA raggiunge i 3,34 V (2/3 Vcc), l'USCITA diventa bassa, il condensatore inizia a scaricarsi, quando la tensione scende sotto 1,67 V (1/3 Vcc) avviene il nuovo cambio di stato, con l'USCITA che diventa alta ed il condensatore inizia nuovamente a caricarsi.

PROGETTO 2

Generatore di onda quadra con Duty Cycle al 50% alimentato a 5V

Come abbiamo visto nello schema precedente, la tipologia classica di configurazione astabile

difficilmente ci fa ottenere un Duty Cycle al 50%; il motivo è da ricercare nella differenza dei tempi di carica e scarica del condensatore collegato ai pin TRIGGER/SOGLIA; infatti, mentre la carica avviene mediante il resistore collegato tra positivo di alimentazione e pin SCARICA ($R_A = R_1$), e quello collegato tra pin SCARICA e pin TRIGGER/SOGLIA ($R_B = R_2$), la scarica avviene solo attraverso quest'ultimo, quindi in un tempo tanto più inferiore quanto maggiore è la differenza di valore tra R_1 ed R_2 . Se usiamo due resistori identici da 10 kohm, la corrente di carica troverà una resistenza totale di 20 kohm, mentre quella di scarica troverà solo 10 kohm. Diventa chiaro che, compatibilmente con la frequenza desiderata, un modo per avere un Duty Cycle molto prossimo al 50% consiste nell'adottare per R_1 un valore molto più piccolo di R_2 , in rapporto circa 1:30; per esempio, con $R_1 = 1,5 \text{ kohm}$ e $R_2 = 47 \text{ kohm}$ e $C_1 = 10 \mu\text{F}$ otteniamo un'onda quadra di 1,5 Hz e Duty Cycle al 50%, naturalmente sono calcoli teorici, poi, come sempre, bisognerà tener conto delle tolleranze dei componenti in gioco.

Ma possiamo anche ricorrere ad un semplice trucco, visibile in questo schema. Sappiamo che i diodi hanno la caratteristica di far passare la corrente in un solo verso, quello che va dall'anodo verso il catodo; usandone due, possiamo fare in modo, ponendo $R_1 = R_2$, che sia la corrente di carica che quella di scarica attraversino solo uno dei due resistori. Infatti, nel nostro caso, il



Fig. 6
Funzionamento del circuito astabile P2A.

condensatore si caricherà attraverso R1 e D1, in quando D2 impedirà il passaggio della corrente attraverso R2. Viceversa, il condensatore si scaricherà solo attraverso D2 e R2, in quanto D1 impedirà il passaggio della corrente attraverso R1. Nella progettazione bisognerà tener conto del fatto che i due diodi hanno una propria resistenza, che è influente sul Duty Cycle, ma non sulla frequenza di oscillazione del circuito, visto che tale resistenza si andrà a sommare a quelle dei resistori adottati. Con i valori da noi adottati, in assenza dei diodi, le formule note ci danno come risultato:

$$f = \frac{1,44}{(R1+2R2)*C1} = \frac{1,44}{(5600+11200)*0,0001} = 0,86\text{Hz}$$

e

$$D.C. \% = (1 - \frac{R_B}{R_A+2R_B}) * 100 = 1 - \frac{5600}{(5600+11200)} * 100 = 66,7\%$$

In realtà si otterrà una frequenza di circa 1 Hz (quindi maggiore rispetto a quella teorica) con un Duty Cycle al 50%; si deduce che per ottenere la frequenza calcolata, nella pratica bisognerà diminuire la capacità del condensatore.

Anche in questo caso, sostituendo C1 con altri valori cambia la frequenza, con C1 = 10 µF avremo circa 10 Hz mentre con C1 = 1000 µF avremo circa 0,1 Hz.

Vediamo il comportamento del circuito.

All'accensione il condensatore C1 è scarico, il pin TRIGGER (collegato assieme a SOGLIA) si trova quindi a livello minore di 1,67 V (1/3 Vcc) e, in base alla tabella 1, l'USCITA è alta (LED blu acceso). Poiché il transistor collegato al pin SCARICA è interdetto, C1 inizia a caricarsi mediante R1 e D1, la tensione sul pin TRIGGER aumenta, quando supera i 3,34 V (2/3 Vcc), si verifica la combinazione TRIGGER > 1/3 Vcc e SOGLIA > 2/3 Vcc, il comparatore del 555 attiva il RESET del flip-flop interno e avviene il cambio di stato, l'USCITA diventa bassa (LED rosso acceso), il transistor interno va in saturazione, il pin SCARICA viene collegato a massa e C1 inizia a scaricarsi, mediante D2 e R2. Quando la tensione sul pin TRIGGER scende sotto 1,67 V (1/3 Vcc), il comparatore attiva il SET del flip-flop interno e l'USCITA torna nuovamente alta (LED blu acceso). Poiché la durata degli stati HIGH e LOW dell'USCITA è identica, i due LED lampeggeranno in modo regolare, restando accesi/spenti alternativamente per lo stesso periodo di tempo. Il frequenzimetro misurerà correttamente la frequenza di 1 Hz, anche se ogni tanto potrebbe

mostrare il valore 0, cosa normale visto che la risoluzione del frequenzimetro è proprio 1 Hz, con tolleranza di ± 1 digit.

Anche in questo caso abbiamo effettuato una rilevazione dei segnali TRIGGER/SOGLIA e USCITA con un DSO professionale (**Fig. 6**).

Come nella situazione precedente, il segnale giallo quindi mostra l'andamento di carica (segnale in salita) e scarica (segnale in discesa) del condensatore C1, mentre quello celeste il corrispondente stato dell'USCITA. Si vede chiaramente che la durata della fase di carica (ascendente) è praticamente identica a quella di scarica (discendente) del condensatore, e questo genera un Duty Cycle al 50%; il cambio di stato avviene quando la tensione sui pin TRIGGER/SOGLIA raggiunge i 3,34 V (2/3 Vcc), l'USCITA diventa bassa, il condensatore inizia a scaricarsi, quando la tensione scende sotto 1,67 V (1/3 Vcc) avviene il nuovo cambio di stato, con l'USCITA che diventa alta ed il condensatore inizia nuovamente a caricarsi.

PROGETTO 3

Generatore di onda quadra con Duty Cycle tra 1 e 99% alimentato a 5V

Questo circuito può essere usato come generatore di PWM (Pulse Width Modulation), nelle applicazioni che richiedono un Duty Cycle variabile per il loro funzionamento. Questa configurazione permette di modificare i tempi di carica e scarica del condensatore C1, che viene caricato tramite R1, D2 e R2, mentre si scarica tramite R2 e D1. Il potenziometro è utilizzato come partitore resistivo, in base alla posizione del cursore la sua resistenza totale si suddivide in due parti; chiaramente aumentando il valore di uno dei due rami diminuisce l'altro. Abbiamo analizzato il funzionamento di questo circuito con un DSO professionale, in modo da ottenere le immagini riportate in **Fig. 7**; in questo caso però abbiamo invertito i canali, quindi il segnale celeste è quello che rileva l'andamento di carica (segnale in salita) e scarica (segnale in discesa) del condensatore C1 (pin TRIGGER/SOGLIA), mentre quello giallo il corrispondente stato dell'USCITA.

Poniamo che il cursore inizialmente sia esattamente a metà, ognuno dei due rami avrà una resistenza di 25 kohm, quindi i tempi di carica e scarica saranno identici e si otterrà un Duty Cycle al 50% (**Fig. 7a**).

Se spostiamo il cursore verso D1 la resistenza di questo ramo diminuirà e quella dell'altro ramo



aumenterà; per esempio se ruotiamo il cursore fino ad ottenere 47,5 kohm sul ramo di D2 e 2,5 kohm sul ramo D1, il rapporto tra i due rami sarà di circa 19:1, quindi la fase di carica sarà 19 volte maggiore di quella di scarica, in questo caso otterremo un Duty Cycle al 95% (**Fig. 7b**). Viceversa, se ruotiamo il cursore fino ad ottenere 2,5 kohm sul ramo D2 e 47,5 kohm sul ramo D1, il rapporto tra i due rami sarà di circa 1:19, quindi la fase di carica sarà 19 volte minore di quella di scarica, in questo caso otterremo un Duty Cycle al 5% (**Fig. 7c**).

In realtà bisogna considerare che R1, pur avendo un valore piuttosto basso, ha una sua incidenza sulla carica, specialmente quando il valore del partitore R2 sul ramo di D2 è molto basso, quindi, per chi vuole avere dati precisi, di seguito riportiamo le formule di calcolo dei valori reali dei tempi delle fasi H e L, nelle quali $R2_2$ è la resistenza del partitore sul ramo D2.

$$t_H = 0,693 * (R1 + R2_2) * C1$$

$$t_L = 0,693 * (R2_2) * C1$$

Il calcolo della frequenza è piuttosto complicato, se si considerano le moltissime combinazioni del partitore; se ci basiamo sulla condizione in cui i due rami sono identici, quindi 25 kohm, risulta essere:

$$f = \frac{1,44}{(R1 + 2R2) * C1} = \frac{1,44}{(470 + 50000) * 0,000000047} = 608 \text{ Hz}$$

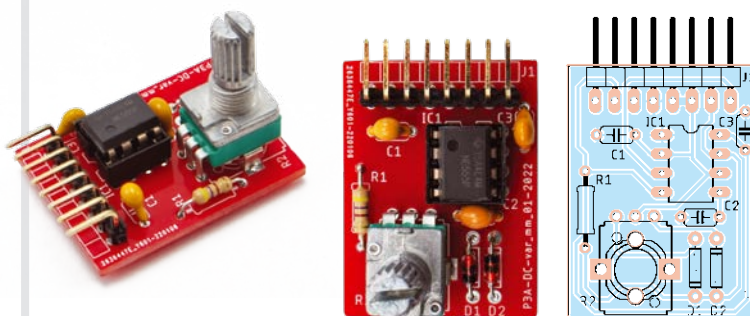
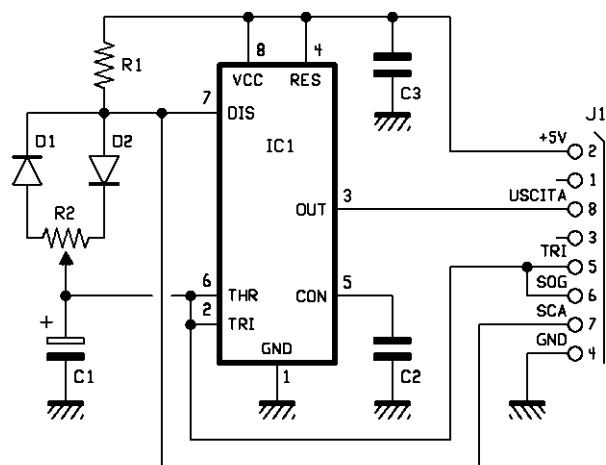
circa. Di fatto la presenza dei diodi, che hanno una soglia di conduzione ed un proprio valore resistivo interno, modifica in modo consistente il risultato della formula; inoltre la modifica stessa del valore resistivo sui due rami ha un'incidenza sulla frequenza. Nel nostro prototipo, col potenziometro posizionato per un Duty Cycle del 50%, abbiamo rilevato una frequenza di circa 510 Hz, con una variazione di ± 10 Hz agli estremi, naturalmente potreste ottenere risultati ancora differenti a seconda dei margini di tolleranza della precisione dei componenti usati.

Il funzionamento del circuito è quello tipico della configurazione astabile. All'accensione il condensatore C1 è scarico, il pin TRIGGER (collegato assieme a SOGLIA) si trova quindi a livello minore di 1,67 V ($1/3 V_{CC}$) e l'USCITA è alta (LED blu acceso). Poiché il transistor collegato al pin SCARICA è interdetto, C1 inizia a caricarsi mediante R1, D2 e R2, la tensione sul pin TRIGGER

aumenta, quando supera i 3,34 V ($2/3 V_{CC}$) il comparatore del 555 attiva il RESET del flip-flop interno e avviene il cambio di stato, l'USCITA diventa bassa (LED rosso acceso), il transistor interno va in saturazione, il pin SCARICA viene collegato a massa e C1 inizia a scaricarsi, mediante R2 e D1. Quando la tensione sul pin TRIGGER scende sotto 1,67 V ($1/3 V_{CC}$), il comparatore attiva il SET del flip-flop interno e l'USCITA torna nuovamente alta (LED blu acceso). L'effetto della variazione del Duty Cycle è palesemente

PROGETTO 3

GENERATORE DI ONDA QUADRA CON DUTY CYCLE 1~99%



Elenco Componenti:

R1: 470 ohm

R2: 50 kohm potenziometro c.s.
RV9

C1: 47 nF poliestere o ceramico

C2: 10 nF poliestere o ceramico

C3: 100 nF poliestere o ceramico

D1, D2: 1N4148

IC1: NE555

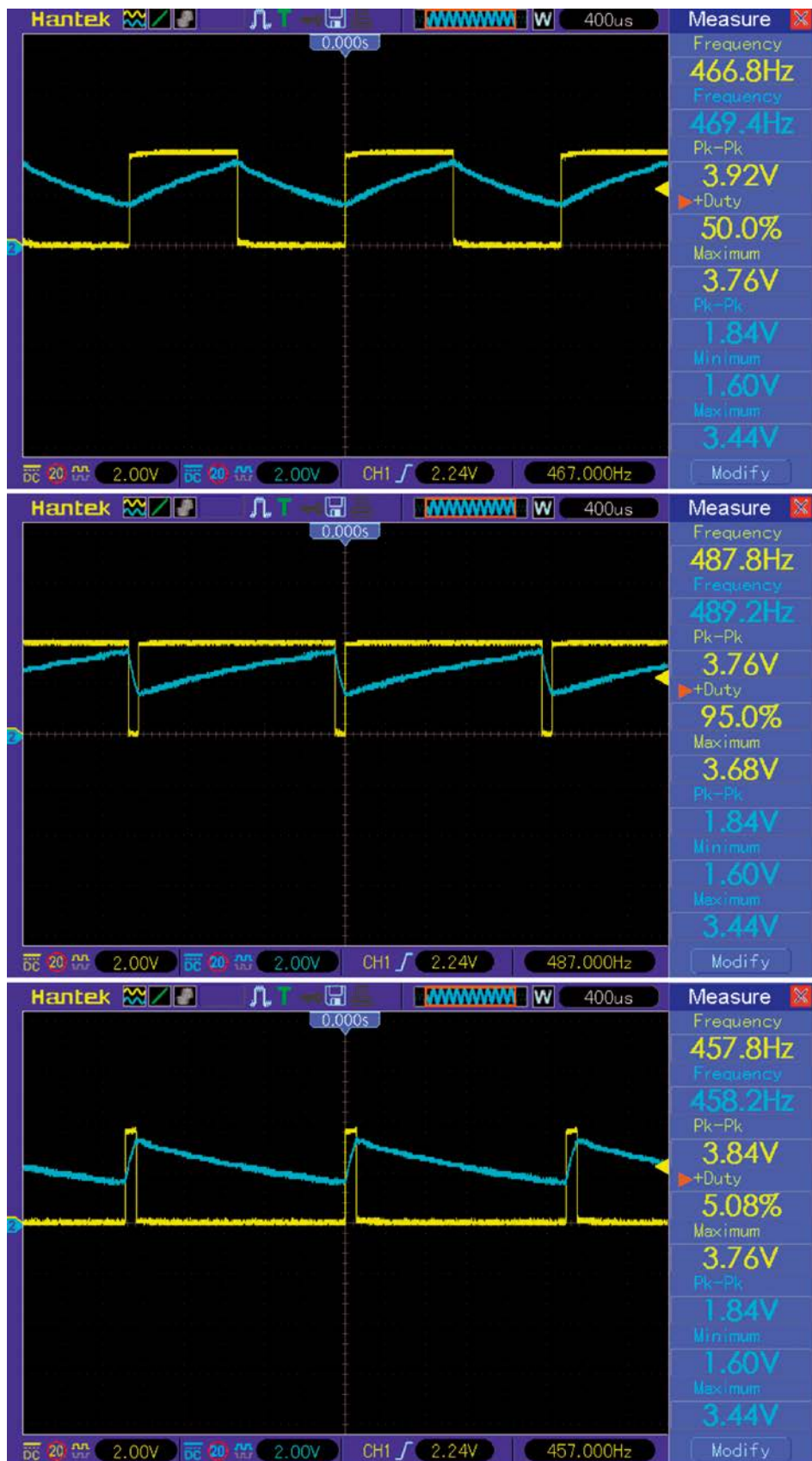
J1: strip maschio 8 vie 90°

Varie:

- zoccolo 4+4 pin

- circuito stampato S1635_P3A
(33,5 x 25,5 mm)

Fig. 7
Funzionamento
del circuito astabile
P3A.



visibile sui due LED; infatti quando esso è all'1% la durata dello stato alto è talmente breve che il LED blu si vede appena mentre l'intensità del LED rosso è massima. Man mano che il Duty Cycle sale l'intensità del LED blu aumenta mentre quella del LED rosso diminuisce. Quando il Duty Cycle è al 50% l'intensità luminosa dei due LED è praticamente identica; quando il Duty Cycle arriva al massimo si inverte la situazione, con il LED blu alla massima luminosità mentre il LED rosso è appena visibile. Una nota finale riguarda il condensatore C3 sul pin di alimentazione del 555, serve per evitare disturbi sul segnale, ed è importante per il tipo di applicazioni alle quali si presta questo circuito.

PROGETTO 4

Sirena bitonale manuale alimentata a 12V

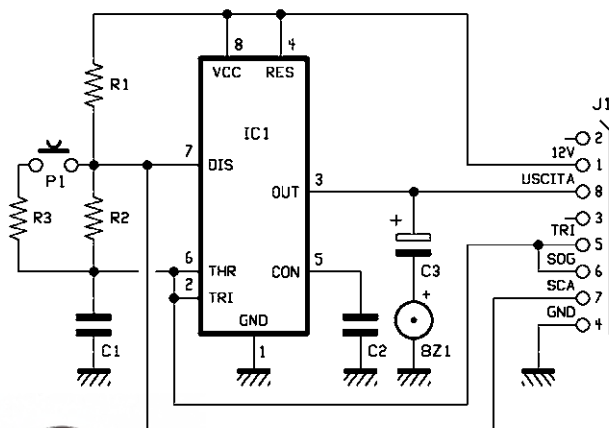
Questo circuito genera un segnale sonoro a frequenza fissa, e prevede un tastino che, per tutto il tempo che viene mantenuto premuto cambia la frequenza del segnale, quindi premendolo ad intervalli regolari si ottiene la simulazione di una sirena bitonale.

Come per tutte le configurazioni astabili l'USCITA oscilla costantemente in funzione del circuito RC applicato ai pin TRIGGER/SOGLIA. All'accensione,

il condensatore C1 è scarico e si comporta come un cortocircuito a massa, per cui, trovandosi il TRIGGER ad una tensione minore di $1/3 V_{cc}$ e, in base alla tabella 1, l'USCITA è alta; la corrente che scorre attraverso i resistori R1 ed R2 inizia a caricare C1, la tensione sui pin TRIGGER/SOGLIA comincia ad aumentare; nel momento in cui il pin TRIGGER supera $1/3 V_{cc}$ (4 V) non succede nulla, in quanto il pin SOGLIA è allo stesso livello, quindi inferiore a $2/3 V_{cc}$ (8 V). Solo quando la tensione sui due pin supera i $2/3 V_{cc}$, si verifica la combinazione TRIGGER > $1/3 V_{cc}$ e SOGLIA > $2/3 V_{cc}$, il comparatore del 555 attiva il RESET del flip-flop interno e porta l'USCITA a livello basso. A questo punto il condensatore inizia a scaricarsi, tramite R2, attraverso il pin SCARICA (il transistor interno del 555 è in saturazione); quando la tensione sul pin SOGLIA scende sotto i $2/3 V_{cc}$ non succede nulla, in quanto il pin TRIGGER si trova nelle stesse condizioni, quindi ad un livello maggiore di $1/3 V_{cc}$. Solo nel momento in cui la tensione sui due pin scende al di sotto di $1/3 V_{cc}$, il comparatore attiva il SET del flip-flop interno ed avviene il nuovo cambio di stato, e quindi ricomincia il ciclo. Questo ciclo avviene ad una frequenza stabilita dalla combinazione R1-R2-C1 che, come vedremo, rientra nella gamma

PROGETTO 4

SIRENA BITONALE MANUALE

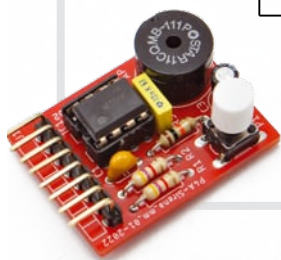
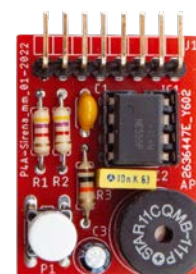
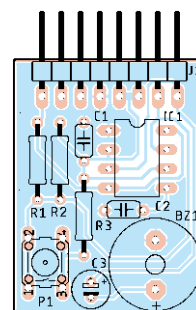


Elenco Componenti:

- R1, R2: 4,7 kohm
- R3: 10 kohm
- C1: 100 nF ceramico
- C2: 10 nF poliestere
- C3: 10 μ F 16 V elettrolitico
- BZ1: buzzer passivo da cs
- IC1: NE555
- J1: strip maschio 8 vie 90°
- P1: pulsante da c.s.

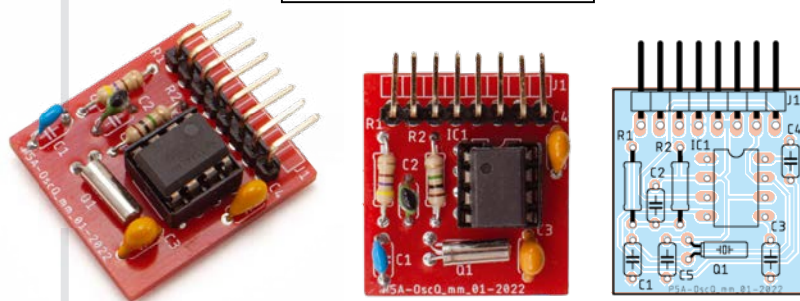
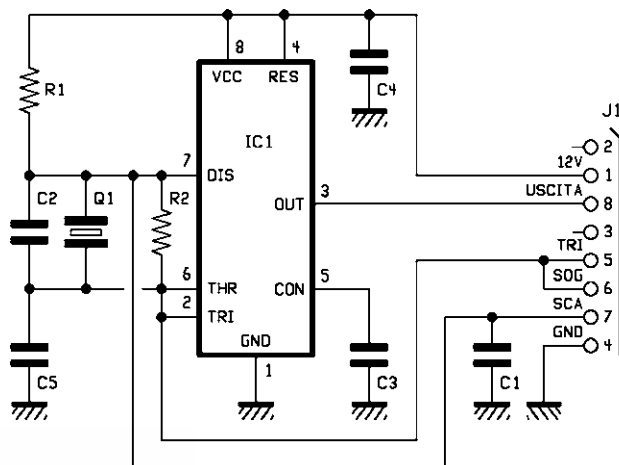
Varie:

- cappuccio pulsante
- zoccolo 4+4 pin
- circuito stampato S1635_P4A (33,5 x 25,5 mm)



PROGETTO 5

OSCILLATORE QUARZATO



Elenco Componenti:

- | | |
|-----------------------------|--|
| R1: 470 ohm | Q1: quarzo 32,768 kHz TC26H |
| R2: 1 Mohm | |
| C1: 10 nF ceramico | |
| C2, C5: 10 pF ceramico | |
| C3: 10 nF ceramico | |
| C4: 100 nF ceramico | |
| J1: strip maschio 8 vie 90° | |
| | Varie: |
| | - zoccolo 4+4 pin |
| | - circuito stampato S1635_P5A (27,5 x 25,5 mm) |

audio, per cui l'USCITA invierà un segnale al buzzer passivo BZ1, mediante il condensatore C3, che serve per bloccare la tensione continua del segnale. Tale frequenza rappresenta il primo dei due toni della nostra sirena, mentre il secondo si ottiene premendo il pulsante P1, che applica il resistore R3 in parallelo a R2, riducendone di fatto il valore a:

$$R_x = \frac{R2 * R3}{R2 + R3} = \frac{10.000 * 4.700}{14.700} = 3.200 \text{ ohm}$$

circa. La prima frequenza è data dalla formula:

$$f = \frac{1,44}{(R1 + 2R2) * C1} = \frac{1,44}{(4700 + 9400) * 0,0000001} = 1020 \text{ Hz}$$

circa; per il calcolo della seconda dobbiamo considerare, come abbiamo visto, $R2 = 3.200 \text{ ohm}$, quindi la formula diventa:

$$f = \frac{1,44}{(R1 + 2R2) * C1} = \frac{1,44}{(4700 + 6400) * 0,0000001} = 1300 \text{ Hz}$$

circa. Quindi, riepilogando, alimentando il circuito sentiremo nel buzzer passivo un suono con frequenza pari a circa 1.020 Hz, ogni volta che premeremo il pulsante P1 la frequenza cambierà in 1300 Hz, provocando l'effetto bitonale. Come sempre ricordiamo che questi sono calcoli teorici, che non tengono conto delle tolleranze tipiche dei componenti passivi utilizzati, di conseguenza, nella realizzazione pratica, potreste ottenere frequenze differenti da quelle teoriche, che saranno comunque misurate con precisione dal frequenzimetro.

PROGETTO 5

Oscillatore quarzato alimentato a 12V

Questa particolare applicazione è utile per realizzare una frequenza molto stabile e precisa, in quanto basata su un quarzo. La frequenza massima raggiungibile dipende ovviamente dal quarzo utilizzato ma anche dalle caratteristiche del modello di 555 utilizzato, per cui è inutile prevedere un quarzo che oscilli ad una frequenza superiore a quella massima del 555. Il nostro circuito funzionerà con qualsiasi modello di 555, anche obsoleto, in quanto abbiamo usato un quarzo che oscilla a 32,768 kHz (utilizzato in genere come clock per orologi), ed è questa la frequenza del segnale, estremamente stabile, che otterremo in uscita.

Questo schema è un po' particolare rispetto al tipico astabile, R1 e C1 rappresentano l'oscillatore di riferimento per il 555, mentre Q1-C2-R2-C5 costituiscono il gruppo oscillatore quarzato, che controlla con precisione l'andamento del flip-flop interno. Mentre la carica di C1 avviene normalmente tramite R1, la scarica non avviene tramite un resistore ma direttamente sul pin SCARICA, quindi è quasi istantanea. Il condensatore C4 sull'alimentazione serve per evitare disturbi sul segnale.

Per comprendere il funzionamento di questo circuito abbiamo fatto ancora una volta ricorso ad un DSO professionale, ed abbiamo rilevato il segnale sui pin TRIGGER/SOGLIA (fucsia), sull'USCITA (celeste) e su SCARICA, per osservare l'andamento della carica (ascendente) e scarica



(discendente) del condensatore C1; abbiamo inoltre posizionato due cursori di riferimento (linee tratteggiate bianche) esattamente su 4 V e 8 V (**Fig. 8**). All'accensione, dopo qualche istante di stabilizzazione del circuito, l'oscillatore quarzato si assesta sulla frequenza di 32,768 kHz (sinusoide fucsia); nell'istante in cui il TRIGGER (segnale fucsia) scende ad un livello inferiore ad $1/3 V_{cc}$ (circa 4 V), si vede che l'USCITA diventa alta (onda quadra celeste) come confermato dalla prima riga della **Tabella 1**, vista in precedenza. Contestualmente il condensatore C1 inizia a caricarsi (segnale giallo) in quanto il transistor interno è interdetto. Quando il TRIGGER supera $1/3 V_{cc}$ (4 V) non succede nulla in quanto il pin SOGLIA (collegato assieme al TRIGGER) è al disotto di $2/3 V_{cc}$. Solo quando la tensione sui due pin supera i $2/3 V_{cc}$ (circa 8 V, è il punto in cui la sinusoide subisce una lieve distorsione) si verifica la combinazione $\text{TRIGGER} > 1/3 V_{cc}$ e $\text{SOGLIA} > 2/3 V_{cc}$, ed il comparatore del 555 attiva il RESET del flip-flop interno, portando l'USCITA a livello basso; nello stesso istante il condensatore C1 inizia a scaricarsi attraverso il pin SCARICA, in quanto il transistor interno del 555 è in saturazione; in questo caso la scarica è quasi istantanea, in quanto C1 è collegato al pin direttamente e non tramite un resistore. Da qui ricomincia il ciclo appena descritto. Nel caso del nostro prototipo, sia il frequenzimetro

che il DSO hanno misurato una frequenza di 32,766 kHz, quindi 2 Hz meno del valore atteso; questo dipende dalla tolleranza del quarzo, espressa sui datasheet in ppm, e per la cui compensazione occorrerebbe agire sulla capacità di C2 che, nel nostro caso, potrebbe essere sostituito da un compensatore, ossia un condensatore variabile, per esempio un 4 ± 20 pF, in modo da trovare la capacità precisa per ottenere l'oscillazione nominale.

CONCLUSIONI

Con la serie dei circuiti astabili termina qui la seconda puntata dedicata all'Analizzatore di circuiti basati sul timer 555. Nella prossima puntata presenteremo una serie di circuiti del tipo monostabile e bistabile. ■



Cosa occorre?

Il componente principale di questo progetto, il timer 555 (cod. 555), è disponibile a Euro 0,60. Il Mini DSO (cod. DSO138MINI) è in vendita a Euro 68,00. I circuiti stampati possono essere realizzati utilizzando il servizio di prototipazione dei PCB (cod. PCBPRODUCTION).

Il materiale va richiesto a:

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

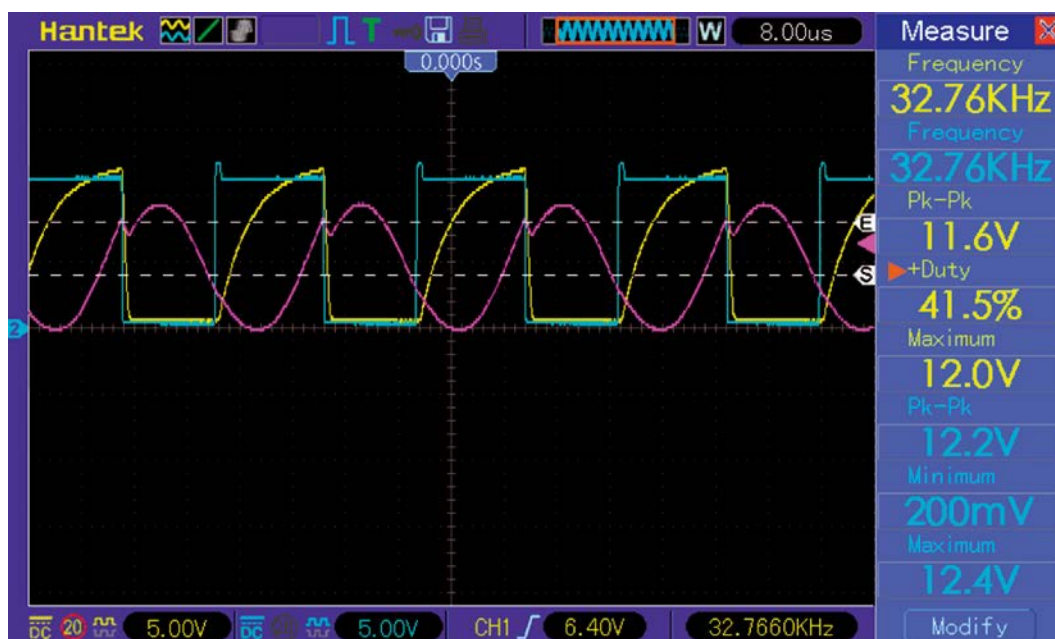


Fig. 8
Funzionamento del circuito astabile P5A.



cod. FR830

€ 38,00

**PENNA CON TELECAMERA
AD ALTA RISOLUZIONE**

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



cod. FR825

€ 80,00

**TELECAMERA A BOTTONE
CON TELECOMANDO**

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

Non farti sfuggire nulla. Riprendi ogni attimo con le mini telecamere.



cod. FR806

€ 79,00

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

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



cod. CP756

€ 19,90

**MINI TELECAMERA
23 x 23 x 23 mm**

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

Prezzi IVA inclusa

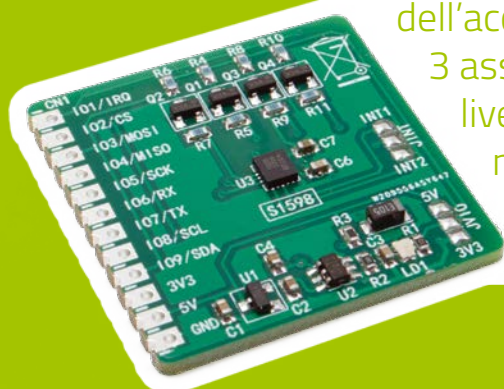


dell'Ing. **VINCENZO GERMANO**

BREAKOUT BOARD

ADXL372

Scopriamo l'integrato ADXL372, un sistema completo di misurazione dell'accelerazione a 3 assi che opera a livelli di potenza molto bassi.



Q

Quando sorge la necessità di avere un'interfaccia per interagire con l'ambiente e per trasformare una grandezza fisica in un segnale elettrico digitale, la scelta della sensoristica oggi è piuttosto semplice, data la vastità dell'offerta in un vasto campo di applicazioni. Alcuni dei trasduttori

comunemente utilizzati sono sensori di corrente, sensori di tensione, tachimetri, sensori di luce, fino ad arrivare a sensoristica più complessa come accelerometri, giroscopi o anche IMU (*Inertial Measurement Unit*, ovvero unità di misura inerziale), che ormai si trova dappertutto. Questi non sono altro che piccoli sensori incredibilmente utili, che vengono integrati sempre di più nei dispositivi elettronici che ci circondano: li troviamo nei telefoni cellulari, nei droni, robot di auto-bilanciamento, nei giocattoli, nei telecomandi delle

console di gioco, tute "motion capture" e molto altro. Come suggerisce il nome, il sensore più completo in questo ambito è l'IMU, che in realtà è un sistema composto in grado di misurare i dati di orientamento e fornire una comprensione completa dell'accelerazione, velocità, posizione e orientamento di un dispositivo; tutto ciò, grazie alla combinazione di tre tipologie di sensori, ovvero accelerometro, giroscopio e magnetometro. Gli accelerometri vengono utilizzati principalmente per misurare l'accelerazione e l'inclinazione, la loro scelta e il loro utilizzo non è per niente banale, infatti è importante considerare tutta una serie di dettagli che vedremo a breve analizzando la breakout board *ADXL372*, reperibile sul sito web www.futuranet.it.

GLI ACCELEROMETRI

Come ci suggerisce il termine, un accelerometro non è altro che un dispositivo meccanico o, più recentemente, microelettromeccanico (MEMS=Micro Electro Mechanical System) dedicato alla misura di un'accelerazione, ovvero la quantità di variazione della velocità di un oggetto. Le forze in gioco possono essere sia statiche che dinamiche, basti pensare nel primo caso alla forza di gravità che costantemente ci tiene saldi al terreno; andandone a misurare la quantità di accelerazione statica si può scoprire il particolare angolo di inclinazione del dispositivo rispetto alla terra; oppure nel caso di forze dinamiche come spostamenti o vibrazioni di qualsiasi tipo e misurandone la quantità di accelerazione dinamica, si può analizzare il modo in cui un particolare dispositivo si muove nello spazio.

Quindi in generale possiamo affermare che possono essere utilizzati per rilevare le vibrazioni nei sistemi o per applicazioni di orientamento. Grazie agli accelerometri, i dispositivi su cui sono montati riescono a comprendere meglio l'ambiente circostante, un accelerometro può aiutare ad analizzare i problemi di un motore di un'auto andandone a rilevare le sue vibrazioni, oppure sono presenti in varie tipologie di computer portatili in modo da poter proteggere i dischi rigidi dai danni accidentali causati da cadute (l'accelerometro rileva l'improvvisa "caduta libera" del computer e spegne il disco rigido in modo che le testine non rovinino i dischi magnetici). Un altro esempio può essere la loro applicazione nel mondo degli incidenti automobilistici, dove vengono utilizzati per rilevare un evento di "schianto" e attivare gli airbag al momento giusto; oppure nel più comune

mondo dei droni, per permettergli una vasta gamma di funzionalità aggiuntive che altrimenti non avrebbero.

Gli accelerometri possono misurare l'accelerazione su uno, due o tre assi; quelli triassiali si sono diffusi maggiormente da quando il costo di sviluppo e industrializzazione di questi prodotti è diminuito drasticamente. Semplificando di molto il loro funzionamento e considerando una linea generale, questi dispositivi possono contenere internamente piastre capacitive e quando le forze di accelerazione agiscono sul sensore, queste piastre si muovono l'una rispetto all'altra facendo cambiare la capacità tra esse permettendo di rilevarne l'accelerazione. Esistono anche altre tipologie di accelerometri realizzati con materiali piezoelettrici e grazie a queste minuscole strutture cristalline vengono emesse delle cariche elettriche quando sono poste sotto uno stress meccanico, ad esempio l'accelerazione; quindi ne esistono di varie tecnologie, pur assolvendo alla stessa funzionalità. Passando oltre e considerando sempre in linea generale le diverse tipologie di accelerometri da un punto di vista funzionale, per potersi interfacciare con loro, oltre la gestione dell'alimentazione, si hanno tre possibili letture/comunicazioni, ovvero analogica, digitale e PWM (Pulse-Width Modulation - modulata a larghezza di impulso). Nel primo caso questi dispositivi mostrano le accelerazioni attraverso livelli di tensione variabili (tra zero volt e il livello della tensione di alimentazione), in modo tale che un microcontrollore mediante il suo convertitore analogico/digitale (ADC) possa "comprenderne" i dati. Nel caso di PWM, gli accelerometri trasmettono delle onde quadre con un periodo noto ma con un "duty cycle" (ciclo di lavoro) che varia al variare dell'accelerazione. Infine, quelli digitali come vedremo a breve, hanno un'interfaccia digitale e mediante un protocollo di comunicazione possono trasmettere un numero di informazioni maggiori verso i microcontrollori, con un conseguente aumento di funzionalità rispetto le due precedenti tipologie e meno suscettibilità al rumore.

Riguardo altre considerazioni generiche sugli accelerometri possiamo dire che sono dei dispositivi a bassissima potenza, con correnti richieste dell'ordine dei milli/microampere o meno e con tensioni di alimentazione di 5 V o inferiore (rendendoli adatti per applicazioni a batteria); questi consumi variano a seconda delle impostazioni del particolare dispositivo e le modalità di risparmio energetico come vedremo nel prossimo paragrafo.



Infine si possono differenziare anche in merito al range di misura, riuscendo a misurare accelerazioni da pochi "g" fino anche ad arrivare a centinaia di "g" e dipendentemente dalla particolare applicazione bisognerà selezionare il più adatto tenendo presente che minore è il range del sensore, più sensibili saranno le letture e viceversa. In base a quanto appena visto, a questo punto analizziamo i dettagli dell'accelerometro ADXL372, presente come cuore della breakout board acquistabile, qui proposta.

L'ACCELEROMETRO ADXL372

L'ADXL372 (www.analog.com) è un accelerometro MEMS (*Micro Electro Mechanical Systems* che indica un dispositivo costituito da strutture meccaniche e circuiti elettronici integrati sullo stesso chip di silicio) a bassissima potenza, a 3 assi, con un range di ± 200 g che consuma 22 μ A ad una velocità di dati di uscita (*Output Data rate - ODR*) dell'ordine dei 3.200Hz; fornisce all'utente la possibilità di accedere alla sua configurazione e ai registri dei dati tramite periferica seriale (*SPI - Serial Peripheral Interface*) o il protocollo I²C (*Inter-Integrated Circuit*). Opera su un'ampia gamma di tensioni di alimentazione ed è disponibile in un package LGA (16 pin Terminal Land Grid Array) da 3 mm $\times$ 3,25 mm $\times$ 1,06 mm, con una configurazione di pin come riportato in **Fig. 1**.

Come potete notare dei 16 pin disponibili, 3 sono riservati (*Reserved*), 2 non collegati (*NIC*), 2 sono pin di interrupt (*INT1* e *2*), tolti i 3 pin di GND e i 2 di alimentazione, i restanti pin riguardano i protocolli di comunicazione appena citati.

Oltre al suo consumo energetico estremamente basso, l'ADXL372 ha molte funzioni tra cui quella per il rilevamento dell'impatto fornendo al contempo una riduzione della potenza a livello di sistema. Sono disponibili varie modalità di funzionamento, descritte qui di seguito.

- **Measurement Mode:** (modalità di misura) ovvero la modalità operativa predefinita, che permette una rilevazione dei dati di accelerazione continua e il consumo è dell'ordine dei 22 μ A (tipico), con un ODR di 3.200 Hz, utilizzando un'alimentazione di 2,5 V. Se il parametro ODR (*Output Data rate - velocità dei dati in uscita*) viene cambiato, anche il relativo consumo in corrente subirà una variazione.
- **Instant On Mode:** indica la modalità di accensione istantanea, consente il rilevamento dell'impatto a potenza estremamente bassa; in questa modalità, l'accelerometro monitora

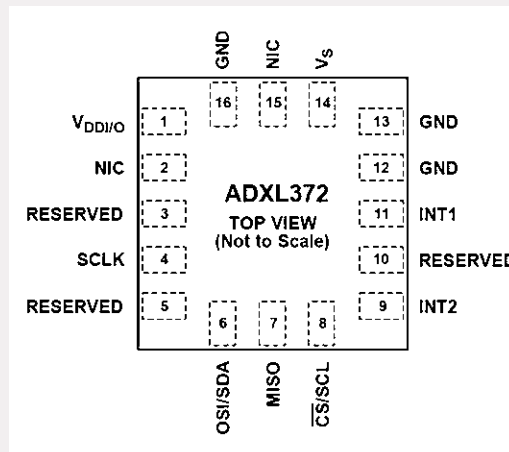
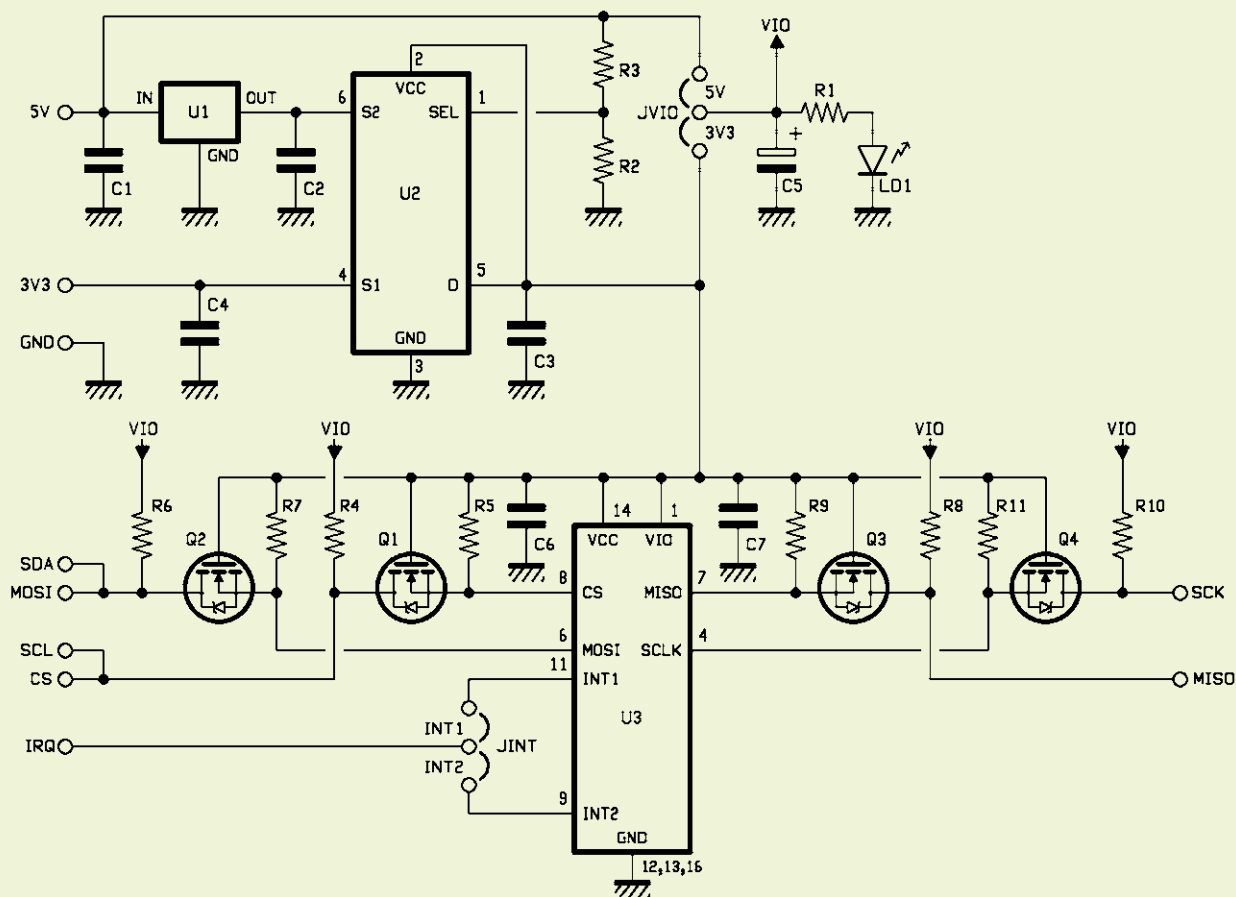


Fig. 1
Configurazione
PIN.

costantemente l'ambiente consumando una corrente dell'ordine degli 1,4 μ A (tipica). Quando viene rilevato un evento che supera una soglia interna definita, il dispositivo passa alla modalità di misurazione per registrare l'evento. Di default la soglia è compresa tra 10 g e 15 g, tuttavia si può impostare un valore differente tramite il registro corrispondente, fino a 30 g - 40 g. Essendo una modalità per risparmio energetico, non vengono resi disponibili nuovi dati di accelerazione digitale fino a quando l'accelerometro non passa al funzionamento normale, pur rimanendo accessibili tutti i registri, trovandosi in normale funzionamento di lettura/scrittura.

- **Wake-Up Mode:** ovvero la modalità "sveglia" è ideale per il rilevamento semplice della presenza o assenza di movimento con un consumo energetico estremamente basso. La modalità di riattivazione è particolarmente utile per l'implementazione di un interruttore on/off attivato a basso movimento, che consente di spegnere il resto del sistema fino a quando non si ha un'attività prolungata. Se viene rilevato un movimento, l'accelerometro può passare autonomamente in altre modalità, a seconda della configurazione del dispositivo.
- **Standby:** infine questa modalità permette al dispositivo la messa in standby che sospende la misurazione e riduce il consumo di corrente a meno di 100 nA (tutte le funzioni del sensore sono disattivate).

Abbiamo visto come siano presenti differenti configurazioni, tra cui un metodo per acquisire solo l'accelerazione di picco degli eventi oltre la soglia (che incorpora anche un filtro per eliminare gli errori iniziali e di variazione lenta, come la deriva



della temperatura ambiente), vediamo ora come sia possibile implementare un funzionamento di questo tipo andando però prima ad analizzare le caratteristiche della breakout board e le sue funzionalità.

SCHEMA ELETTRICO

Diamo dunque uno sguardo allo schema che descrive la breakout board qui proposta; in esso si può notare come il circuito è stato realizzato per portare all'esterno tutte le connessioni dell'integrato (mediante un comodo *header* con passo 2,54mm) ed avere tutti e soli i componenti per il suo pilotaggio. Considerando in prima battuta l'accelerometro, esso è identificato come "U3" e rispetto all'analisi fatta sin qui, possiamo notare come sia il protocollo di comunicazione SPI che I²C sono accessibili tramite piedinatura, perciò si ha la possibilità di scegliere il protocollo d'interesse per poterci comunicare. L'accortezza che è stata tenuta riguarda le alimentazioni. Dato che l'ADXL372 ha una tensione massima d'alimentazione pari a 3,5V

e volendolo rendere compatibile sia con i dispositivi a 3,3V che a 5V (come Arduino, Raspberry Pi e quindi con tutti i circuiti a microprocessore funzionanti con livelli logici TTL), è stato previsto un adattatore di livelli, che per fornire la massima flessibilità viene applicato ad entrambi i protocolli di comunicazione. Al contrario le linee di interrupt (INT1 e INT2) che possono essere utilizzate mediante il jumper "JINT", pur portate all'esterno del componente tramite il contatto "IRQ" non hanno alcun adattamento di segnale e devono quindi essere gestite con dispositivi operanti con livelli logici compatibili.

Per avere una corretta alimentazione dell'accelerometro e contemporaneamente anche la possibilità di alimentare la scheda con 5V o 3,3V sono stati aggiunti i componenti "U1" e "U2". Il primo dispositivo è un regolatore di tensione (MCP1700T-3302E) della Microchip, utilizzato per generare una tensione costante di 3,3V partendo da una tensione di 5V; come si può notare quando sono presenti i 5V viene anche alimentato un

partitore di tensione dato da R2-R3 che fornisce sul pin 1 (SEL) del dispositivo "U2" una tensione diversa da zero. Il dispositivo "U2" altro non è che uno switch bidirezionale della Texas Instrument (*TMUX1219DBVR*) e attraverso il pin 1, esso permette di selezionare sulla sua uscita "D" (pin 5) una delle due tensioni in ingresso S1 o S2 (pin 4 e 6 rispettivamente).

Perciò con un valore diverso da zero sul pin 1 (SEL) in uscita sul pin 5 (D) si avrà la tensione presente sul pin 6 (S2), mentre con un valore pari a zero sul pin 1 (SEL) si avrà in uscita sul pin 5 (D) la tensione d'ingresso del pin 4 (S1). Questo permette di avere automaticamente in uscita al pin 5 di "U2" sempre la corretta alimentazione di 3,3V, indipendentemente dalla tensione d'ingresso scelta. Ultimo dettaglio importante dello schema elettrico riguarda il jumper "JVIO", che oltre ad alimentare un LED (Light Emitting Diode - diodo ad emissione luminosa) permette di definire i livelli di tensione (VIO) dei protocolli di comunicazione con cui comunicare verso il "mondo esterno", perciò può essere impostato a piacimento e dipendentemente dalla tensione utilizzata.

RILEVIAMO GLI IMPATTI

Passiamo a questo punto a un'applicazione pratica della breakout board, con la quale vi spieghiamo come sia possibile rilevare un impatto utilizzando semplicemente una scheda Arduino UNO (Arduino) e le sue risorse di calcolo.

Facciamo riferimento alla **Fig. 2**, nella quale viene riportata la disposizione dei pin dell'header (connettore) e in aggiunta, alla **Tabella 1**, dalla quale è possibile vedere quelle che sono le connessioni tra le due schede; notate immediatamente che è stata utilizzata l'alimentazione a 5V, perciò il jumper JVIO va messo in corto verso i 5V, mentre riguardo la gestione IRQ il jumper JINT va cortocircuitato verso INT1. Il protocollo d'interesse è SPI.

A questo punto per poter utilizzare l'Arduino si ha la necessità di scaricare ed installare le librerie presenti sul sito ufficiale Github della Analog Device che riporta un progetto d'esempio su questo dispositivo: https://github.com/analogdevicesinc/arduino/tree/master/Arduino%20Uno%20R3/examples/ADXL372_example. Per l'installazione di questo progetto nell'IDE di Arduino si rimanda ad una delle varie guide presenti su internet. Dal precedente link si potranno scaricare i cinque file presenti nello screenshot proposto nella **Fig. 3**, che ci serviranno a gestire le informazioni fornite dalla breakout e ad impostare correttamente

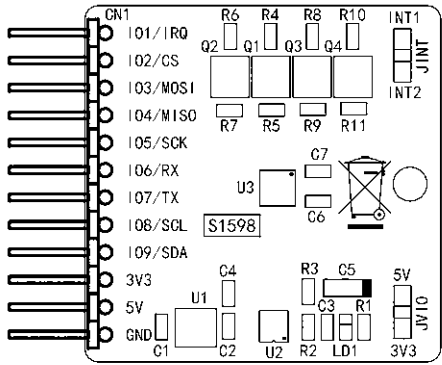


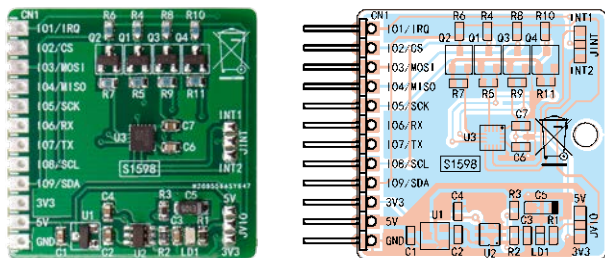
Fig. 2 Serigrafia e pinout della breakout board.

Tabella 1 Connessioni tra Arduino UNO e breakout board ADXL372.

ARDUINO UNO	BREAKOUT BOARD ADXL372
PIN 7	IO1 (IRQ)
PIN 10	IO2 (CS)
PIN 11	IO3 (MOSI)
PIN 12	IO4 (MISO)
PIN 13	IO5 (SCK)
5V	5V
GND	GND

l'integrato ADXL372. Nell'elenco, i due file *adxl372.cpp* e *adxl372.h*, sono le librerie vere e proprie dell'accelerometro, al loro interno vengono definiti tutti i dettagli implementativi per poter gestire le funzionalità dell'ADXL372; mentre nei due file *Communication.cpp* e *Communication.h* sono definite alcune strutture e funzioni aggiuntive per la lettura SPI, per maggiori dettagli si rimanda ai file specifici e relative guide on-line.

Analizziamo brevemente il funzionamento di questo progetto facendo riferimento esclusivamente a quanto presente nel file *ADXL372_example.ino*. Al suo interno, dopo l'import delle varie librerie e la definizione di alcune variabili/strutture, è presente la funzione *Set_Impact_Detection(void)* riportata nea **Listato 1**. Come si può notare, in esso viene impostato il setup del dispositivo prima della sua messa in funzione partendo dalla riga 19 in cui viene messo in standby per poi configurarlo. Tra le varie impostazioni, in prima analisi, troviamo la disattivazione dell'auto-sleep (riga 21), la definizione della larghezza di



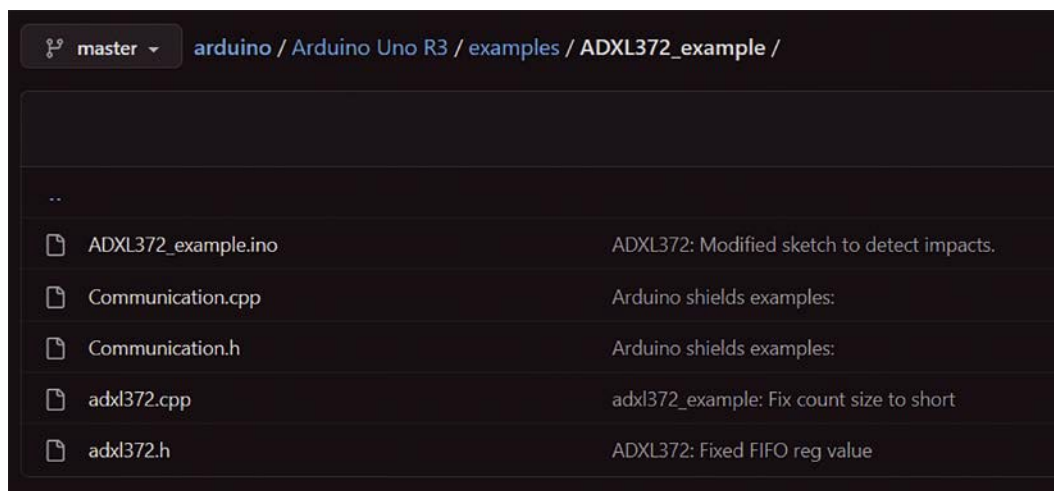
Elenco Componenti:

R1: 470 ohm (0603)
R4, R5, R6, R7, R8, R9, R10, R11: 10 kohm (0603)
R2: 6,8 kohm (0603)
R3: 3,3 kohm (0603)
C1, C2, C3, C4, C6, C7: 1 μ F 6,3V ceramico (0603)
C5: 10 μ F 6,3V tantalio (1206)
U1: MCP1700T-3302E/TT
U2: TMUX1219DBVR
U3: ADXL372BCCZ-RL7
Q1, Q2, Q3, Q4: BSS138W-7-F
LD1: LED red (0603)
Varie:
- Circuito stampato S1598 (33x33mm)

banda (riga 23) e la definizione dell'ODR (riga 25). Si tenga presente che le applicazioni di rilevamento dell'impatto spesso richiedono misurazioni di accelerazione con la necessità di una larghezza di banda e "g" elevati, questo perché un impatto è un evento transitorio che produce un impulso di accelerazione con contenuto di frequenza su un ampio intervallo e con una conseguente necessità di larghezza di banda sufficientemente ampia per catturarlo (la riduzione della larghezza di banda ha l'effetto di ridurre l'ampiezza del segnale registrato con conseguente imprecisione della misurazione). Viene poi impostato (sulla riga 32) il range da rilevare, di cui si è parlato ampiamente in precedenza, per poi definire due parametri

fondamentali in questa tipologia di progetto: le soglie di attività/inattività (righe 38 e 39) e il tempo di attività/inattività (righe 42 e 43), oltre che la gestione interrupt (riferitevi alla riga 46). Con riferimento alla **Fig. 4**, la modalità di attivazione istantanea (*Instant On Mode*) è una modalità a bassissima potenza che monitora continuamente l'ambiente per eventi di impatto che superano una soglia definita, quando viene rilevato un impatto, il dispositivo passa alla modalità di misurazione completa e acquisisce il profilo dell'impatto (perciò tali soglie sono fondamentali a seconda delle particolari necessità di progetto). In aggiunta con riferimento alla **Fig. 5**, in alcune applicazioni potrebbe sorgere la necessità di disporre di un singolo campione di accelerazione (3 assi) al culmine di un evento di impatto (esso contiene informazioni sufficienti sull'evento) e non è richiesta la cronologia completa dell'accelerazione; in questi casi l'ADXL372 offre la capacità di memorizzare solo l'accelerazione di picco di ogni evento oltre la soglia e del picco più alto in assoluto avendo dei registri dedicati. Sempre con riferimento alla **Fig. 5**, non appena viene attivato l'interrupt di "attività", il dispositivo registra i valori dei 3 assi dell'evento di picco di accelerazione che si verifica tra il trigger di interruzione di attività e il successivo trigger di interruzione di inattività. Tutto ciò avviene ciclicamente per ogni periodo di attività compreso tra l'attivazione dell'interruzione di attività e la conseguente attivazione dell'interruzione di inattività, memorizzando i dati nella FIFO (First In, First Out) attivata con l'istruzione di cui alla riga 35. Finché è selezionata la modalità di rilevamento del picco, il dispositivo memorizza anche il picco

Fig. 3
Progetto d'esempio per rilevare gli impatti.



▼ Listato 1 - Funzione "Set_Impact _ Setection(void)".

```

17 void Set_Impact_Detection(void)
18 {
19     adxl372_Set_Op_mode(&adxl372, STAND_BY);
20
21     adxl372_Set_Autosleep(&adxl372, false);
22
23     adxl372_Set_BandWidth(&adxl372, BW_3200Hz);
24
25     adxl372_Set_ODR(&adxl372, ODR_6400Hz);
26
27     adxl372_Set_WakeUp_Rate(&adxl372, WUR_52ms);
28
29     adxl372_Set_Act_Proc_Mode(&adxl372, LOOPED);
30
31     /* Set Instant On threshold */
32     adxl372_Set_InstaOn_Thresh(&adxl372, ADXL_INSTAON_LOW_THRESH); //Low threshold 10-15 G
33
34     /*Put fifo in Peak Detect and Stream Mode */
35     adxl372_Configure_FIFO(&adxl372, 512, STREAMED, XYZ_PEAK_FIFO);
36
37     /* Set activity/inactivity threshold */
38     adxl372_Set_Activity_Threshold(&adxl372, ACT_VALUE, true, true);
39     adxl372_Set_Inactivity_Threshold(&adxl372, INACT_VALUE, true, true);
40
41     /* Set activity/inactivity time settings */
42     adxl372_Set_Activity_Time(&adxl372, ACT_TIMER);
43     adxl372_Set_Inactivity_Time(&adxl372, INACT_TIMER);
44
45     /* Set instant-on interrupts and activity interrupts */
46     adxl372_Set_Interrupts(&adxl372);
47
48     /* Set filter settle time */
49     adxl372_Set_Filter_Settle(&adxl372, FILTER_SETTLE_16);
50
51     /* Set operation mode to Instant-On */
52     adxl372_Set_Op_mode(&adxl372, INSTANT_ON);
53 }

```

complessivo più alto e quando questi valori vengono letti dai registri, i dati dal registro vengono cancellati e il dispositivo inizia a cercare il nuovo picco più alto. Infine, sempre tornando al **Listato 1**, viene attivata la modalità "Instant On" (nella riga 52) per mettere in funzione definitivamente l'accelerometro. Passando oltre e considerando **Listato 2**, vengono riportate le altre due funzioni definite nel file *ADXL372_example.ino*, ovvero *setup()* e *loop()*, nella prima viene configurato il monitor seriale (riga 56) e il protocollo SPI (righe 57 e 58) oltre che i pin di *Chip Select* (riga 61) e gestione interrupt (riga 62), tuttavia è interessante notare che viene chiamata la funzione di configurazione "Set_Impact_Detection()", che imposta l'accelerometro nella corretta modalità di utilizzo. Mentre nella funzione *loop()*, oltre ad essere riscaldati i valori letti (righe 80, 81 e 82), vengono visualizzati mediante monitor seriale formattati come da codice. Come visto sin qui, mediante le librerie dedicate dell'Analog Device si riesce ad utilizzare

rapidamente l'accelerometro per i propri progetti, tuttavia ne abbiamo vista un'implementazione mediante SPI in configurazione "Instant On Mode". A riguardo vi facciamo notare che chi volesse approfondire il funzionamento con il protocollo I²C-Bus in "Measurement Mode", può fare riferimento all'esempio applicativo disponibile su Github nel repository https://github.com/Seeed-Studio/Accelerometer_ADXL372.

▼ **Fig. 4**
Soglie di attività/
inattività in "Instant
On Mode".

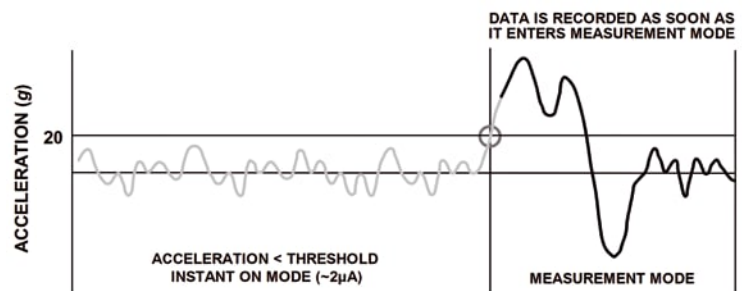
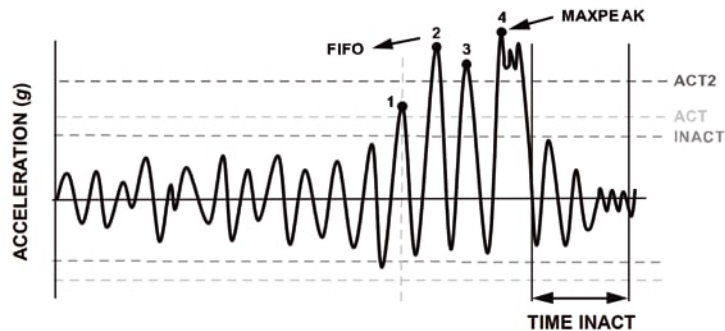


Fig. 5
Cattura degli eventi
d'impatto.



Listato 2 - Funzioni setup e loop per la rilevazione impatti.

```

55 void setup() {
56     Serial.begin(9600);
57     SPI.begin();
58     SPI.setDataMode(SPI_MODE0); //CPHA = CPOL = 0    MODE = 0
59     delay(1000);
60
61     pinMode(CS_PIN, OUTPUT);
62     pinMode(ADXL_INT1_PIN, INPUT);
63
64     adxl372_Get_DevID(&adxl372, &devId);
65
66     Serial.print("Device id: ");
67     Serial.println(devId, HEX);
68
69     adxl372_Set_Op_mode(&adxl372, FULL_BW_MEASUREMENT);
70     Set_Impact_Detection();
71 }
72
73 void loop() {
74
75     if (digitalRead(ADXL_INT1_PIN)) {
76         delay(500);
77         adxl372_Get_Highest_Peak_Accel_data(&adxl372, &accel_data);
78
79         /*Transform in G values*/
80         data_G.x = (float)accel_data.x * 100 / 1000;
81         data_G.y = (float)accel_data.y * 100 / 1000;
82         data_G.z = (float)accel_data.z * 100 / 1000;
83
84         Serial.print("X accel = "); Serial.print(data_G.x); Serial.println(" G");
85         Serial.print("Y accel = "); Serial.print(data_G.y); Serial.println(" G");
86         Serial.print("Z accel = "); Serial.print(data_G.z); Serial.println(" G");
87         Serial.println("");
88
89     }

```

CONCLUSIONI

Arriviamo così al termine di questo articolo, dove abbiamo analizzato i dettagli di funzionamento dell'accelerometro integrato ADXL372 di Analog Devices, le sue caratteristiche e vi abbiamo esposto come sia facile e immediato realizzare dei progetti mediante la breakout board dedicata, qui descritta ed acquistabile già pronta sul sito www.futurashop.it. 



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. La breakout board con l'accelerometro ADXL372 (cod. ADXL372BOUT) è in vendita a Euro 49,00.
I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

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



€32,90
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

DEMOBOARD ESP32



```
// Declaration for an SSD1306 display
// The pins for I2C are defined
// On an arduino UNO:      A4(A5)
// On an arduino MEGA 2560: 20(21)
// On an arduino LEONARDO:  2(3)
#define OLED_RESET      -1 // Reset pin (if any), for controlling the display (1 = no pin)
#define SCREEN_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32
Adafruit_SSD1306 display(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS);

#define NUMFLAKES 10 // Number of flakes to display
#define LOGO_HEIGHT 16
#define LOGO_WIDTH 16

Adafruit_BMP280 bmp; // I2C

void setup() {
  Serial.begin(9600);

  // Initialize the display buffer
  display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS);
  display.println(F("SSD1306 all"));
  for(;;) { // Don't proceed, loop forever
  }
}
```



```

display height, in pixels

display connected to I2C (SDA, SCL pins)
d by the Wire-library.
(SDA), A5(SCL)
(SDA), 21(SCL)
(SDA), 3(SCL), ...
set pin # (or -1 if sharing Arduino reset pin)
< See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32
_WIDTH, SCREEN_HEIGHT, sWire, OLED_RESET);

per of snowflakes in the animation example

```

```

ate display voltage from 3.3V internally
HCAPVCC, SCREEN_ADDRESS)) {
ocation failed");
op forever

```

```

contents on the screen --
with an Adafruit splash screen.

```

```

nds

```

Scopriamo i mille risvolti applicativi di uno dei più utilizzati moduli WiFi con una scheda di prototipazione che può ospitare e gestire sensori e attuatori sperimentabili con firmware di test da noi scritti appositamente. Prima puntata.

di **DAVIDE SCULLINO**

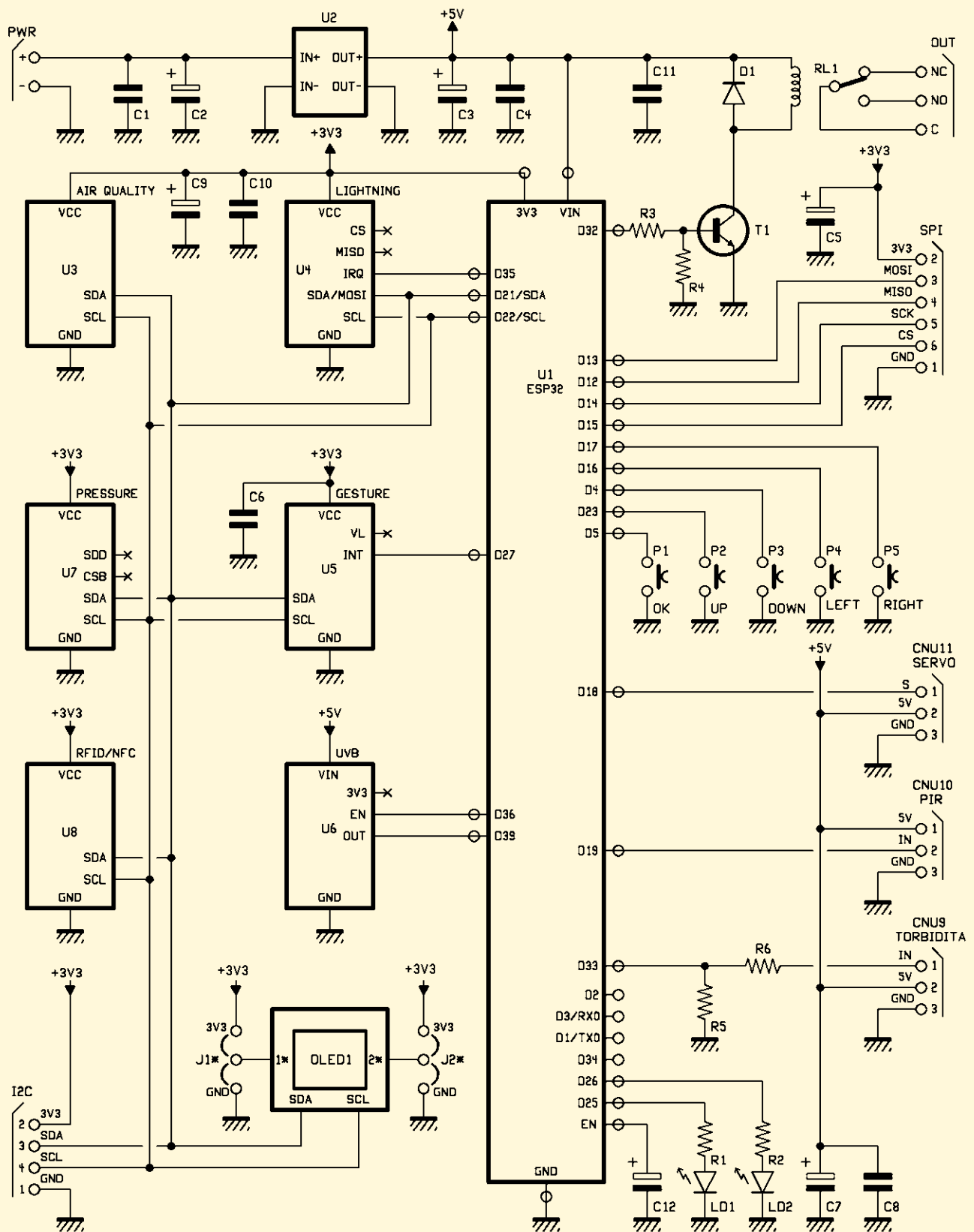
I moduli basati sul SoC ESP32 della Espressif sono molto apprezzati dai progettisti di applicazioni connesse tramite interfaccia wireless WiFi e per questo da anni li vediamo utilizzati e li utilizziamo anche noi in vari progetti. Proprio per la loro importanza e per la grande richiesta da parte degli sviluppatori e degli sperimentatori, abbiamo pensato di realizzare una demoboard che consenta di creare firmware sia per testare una vasta gamma di sensori, sia di imparare a programmare per l'ESP32 e creare applicazioni connesse, prime fra tutte quelle nell'ambito dell'Internet of Things. La scheda che utilizzeremo ha come "core" un modulo basato wireless su ESP32 e dispone poi di tutta una serie di header (connettori) per innestarvi svariate breakout board e moduli che ne costituiscono le periferiche, ovvero sensori di pressione, ultravioletti, qualità dell'aria, fulmini, gesture ed un lettore RFID compatibile NFC. A completare la scheda c'è un'interfaccia utente formata da un display OLED, una serie di pulsanti e dei LED, tutti configurabili per ottenere una specifica modalità di interfacciamento. Vediamo subito di cosa si tratta, analizzando nel dettaglio la scheda, prima di passare alla descrizione dello schema elettrico corrispondente.

LA SCHEDA

La nostra demoboard è basata sulla Board con ESP32 (cod. YB555 commercializzata da www.futurashop.it), dotata di connettore MicroUSB che viene utilizzato per la programmazione, ma allo stesso tempo può fornire alimentazione a tutta la demoboard, a patto che la corrente richiesta non sia superiore a un certo valore, erogabile dalla board ESP32. In alternativa, può essere fornita alimentazione 12V al plug di alimentazione PWR, quindi sarà il regolatore U2 della demoboard (cod. MP1584ENDCDC) ad ottenere i 5V per tutto il resto del circuito. Nella **Fig. 1** trovate la pin-out del modulo ESP32 in questione.

Sulla nostra demoboard possono essere alloggiati i seguenti moduli:

- Sensore qualità aria cod. SGP30CO2 (U3)
- Sensore fulmini cod. FT1324M (U4)
- Gesture cod. APDS9960BOUT (U5)
- Sensore UVB cod. GYLM8511 (U6)
- Sensore pressione cod. BMP280 (U7)
- Lettore NFC/RFID cod. PN532 (U8)



- Display OLED 0,96" cod. OLED096C oppure cod. OLEDTVSCSD (OLED1)
- Relé 5V (RL1)
- LED rosso (LD1)
- Led verde (LD2)
- 5 pulsanti (P1, P2, P3, P4, P5)
- BreadBoard
- Possibilità di collegamento dispositivi SPI 3.3V (SPI)
- Possibilità di collegamento di altri dispositivi I²C 3,3V (I²C).

Oltre a quanto già previsto sopra, è possibile connettere altri dispositivi sfruttando i pin liberi della board ESP32. I pin senza alcuna connessione sono D1/TX0, D2, D3/RX0, D34 oltre ai pin SPI che non sono utilizzati per il collegamento ad alcuna breakout board.

Per quanto riguarda i connettori pensati per la connessione a breakout board e sensori esterni alla scheda, abbiamo previsto questi elementi:

- Sensore torbidità (trasparenza) dei liquidi cod. TORBISENS (CNU9)
- Sensore PIR cod. PIRMOD / cod. MINIPIRMOD (CNU10)
- Servo motore, ad esempio cod. SERVO206 (CNU11).

Tutti i codici riportati tra parentesi si riferiscono ai prodotti presenti sul sito www.futurashop.it, per una più rapida identificazione.

Diamo comunque uno sguardo allo schema elettrico della board, riportato al completo nella pagina qui accanto.

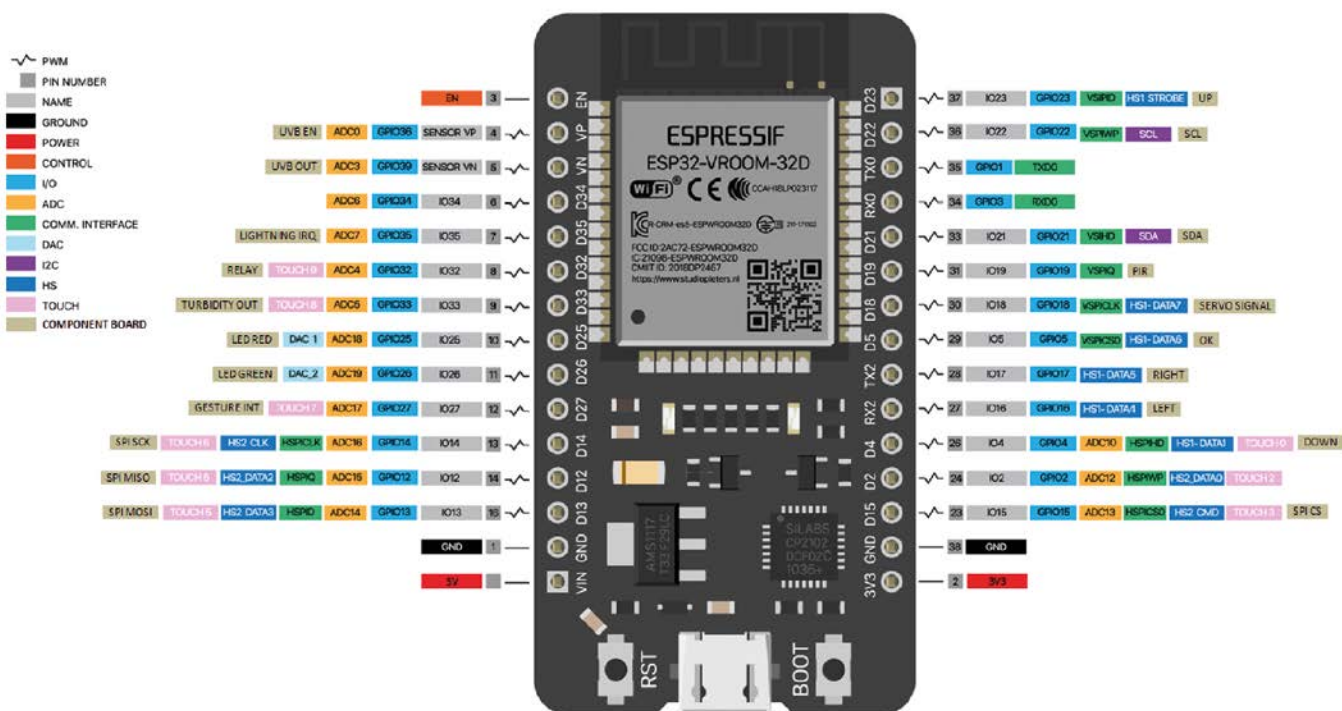
SCHEMA ELETTRICO

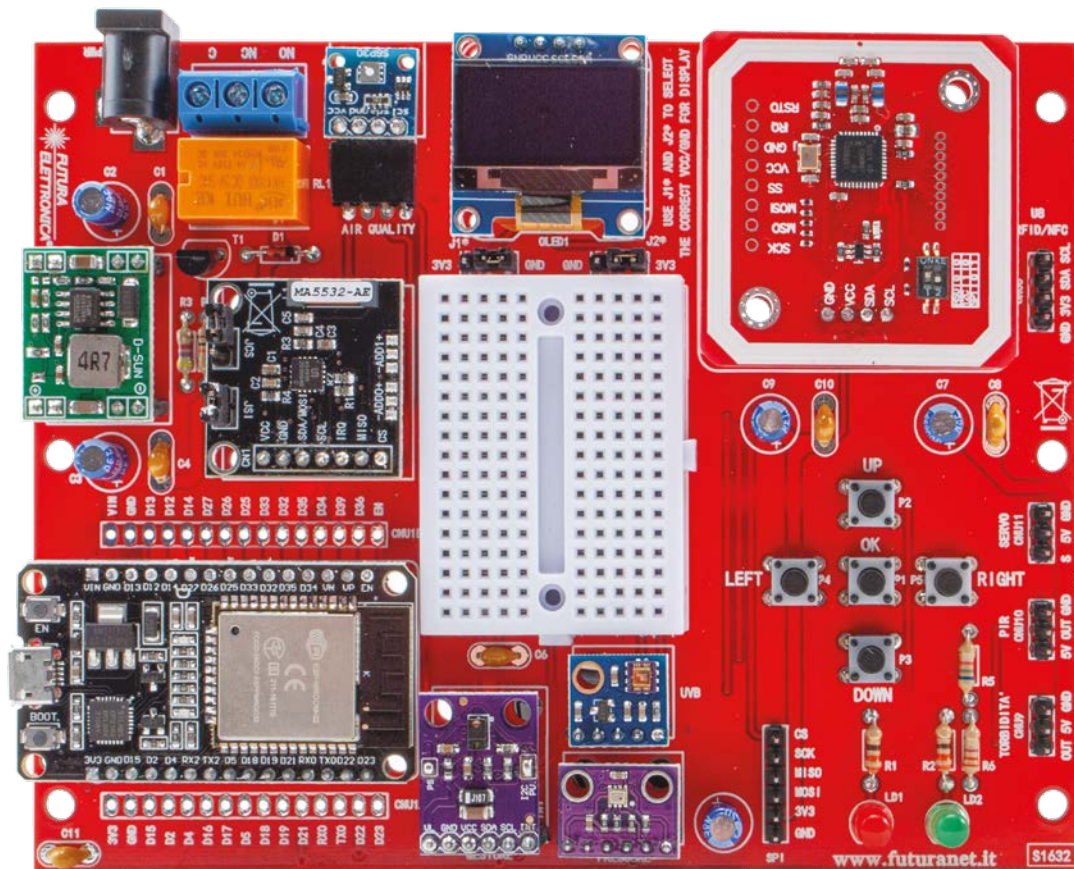
Il circuito, come vedete, è centrato sulla board ESP32 che è siglata U1 e che riceve l'alimentazione principale attraverso il piedino Vin; il suo regolatore interno fornisce 3,3V stabilizzati che alimentano le schede di interfaccia applicabili alla demoboard alimentabili con tale tensione, ossia a tutti quelli ospitabili on-board eccetto U6).

Il contatto Vin dell'U1 è alimentato con i 5V che l'alimentatore principale ricava dalla tensione d'ingresso applicata al jack DC di alimentazione PWR; la tensione d'ingresso viene dapprima filtrata dai condensatori C1 e C2 e poi ridotta a 5V dal regolatore U2, che è un modulo DC/DC step-down MP1584ENDCDC, capace di erogare 3A di corrente su 5V stabilizzati: abbastanza per far funzionare tutti i moduli breakout board supportati dalla nostra scheda demo.

I 5V di uscita (filtrati adeguatamente da C3, C4 e C11 per abbattere i residui di commutazione del modulo switching) vanno ad alimentare i moduli

Fig. 1
Piedinatura del
modulo ESP32.





Elenco Componenti:

C1, C4, C6, C8, C10, C11:
100 nF ceramico

C2, C3, C5, C7, C9: 100 μ
16 VL elettrolitico

D1: 1N4148

LD1: LED 5 mm rosso

LD2: LED 5 mm verde

P1, P2, P3, P4, P5: Microswitch

R1, R2: 330 ohm

R5: 6,8 kohm

R6: 3,3 kohm

R3: 4,7 kohm

R4: 10 kohm

RL1: Relè 5V 1 scambio

T1: BC547

U1: Breakout ESP32-WROOM-32

U2: Modulo Step-Down 5V/3A

U3: Modulo qualità aria SGP30

U4: Sensore fulmini AS3935

U5: Modulo Gesture PDS-9960

U6: Modulo sensore UVB

U7: Modulo sensore pressione

U8: Modulo NFC/RFID

OLED1: Display OLED 0.96"

Varie:

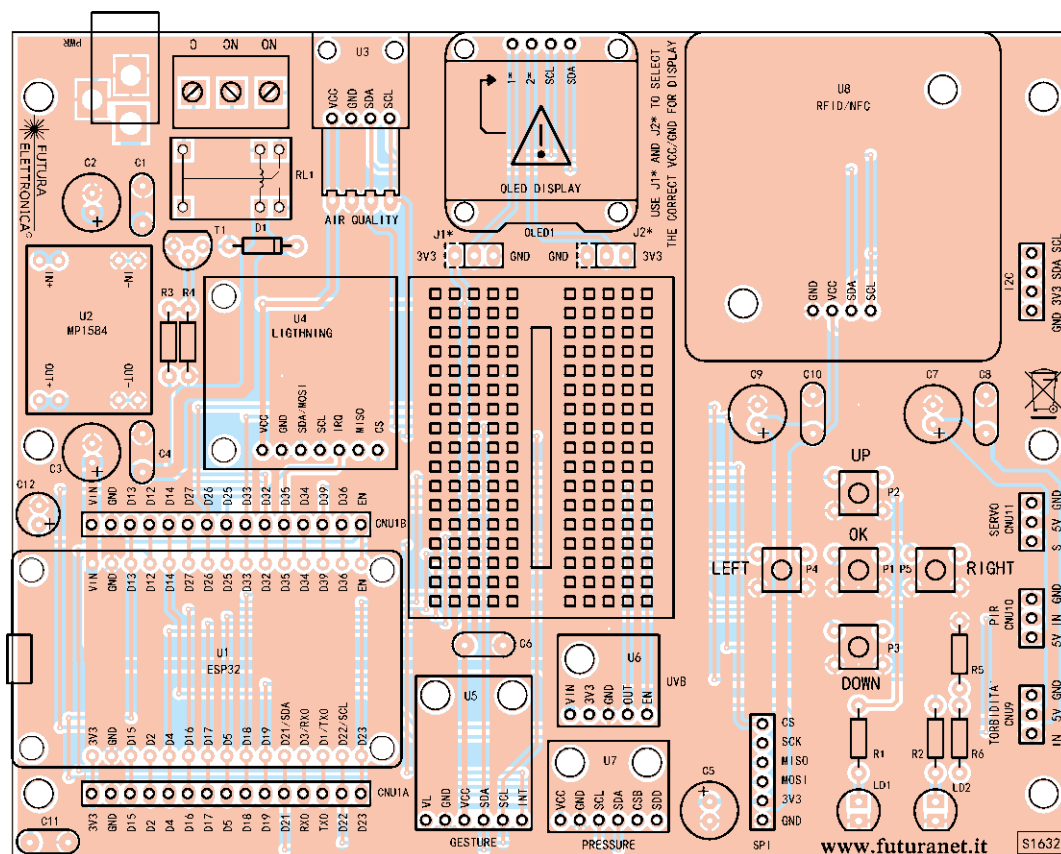
- Pin strip maschio 2 vie passo
2.54mm (4 pz.)

- Pin strip maschio 3 vie passo
2.54mm (5 pz.)

funzionanti a tale tensione (modulo UVB) e i connettori per servo (CNU11) PIR (CNU10) e sensore di torbidità esterno (CNU9), nonché la bobina del relè, che viene commutata dal transistor NPN T1, dietro comando della linea di I/O D32 del modulo ESP32. Del RL1 (la cui bobina è stata dotata di diodo in antiparallelo per proteggere la giunzione di collettore del T1 all'interdizione) rendiamo disponibili i tre contatti dello scambio, in modo da gestire vari tipi di utilizzatore e varie condizioni operative; il relè è una periferica attuatrice associabile ad esempio al superamento di certe soglie di rilevamento da

parte dei sensori, ovvero alle funzionalità di lettura del modulo RFID/NFC.

Il display a bordo scheda è un piccolo OLED interfacciabile tramite I²C-Bus più volte utilizzato nei nostri progetti per la sua versatilità, per il basso costo e per la disponibilità di librerie Arduino dedicate alla sua gestione. Si tratta dell'OLED096C commercializzato dalla Futura Elettronica Modulo con display OLED (Organic Light Emitting Diode) 128x64 pixel di colore blu, interfaccia I²C, controller grafico SSD1306, angolo di visione circa 180°, compatibile Arduino.



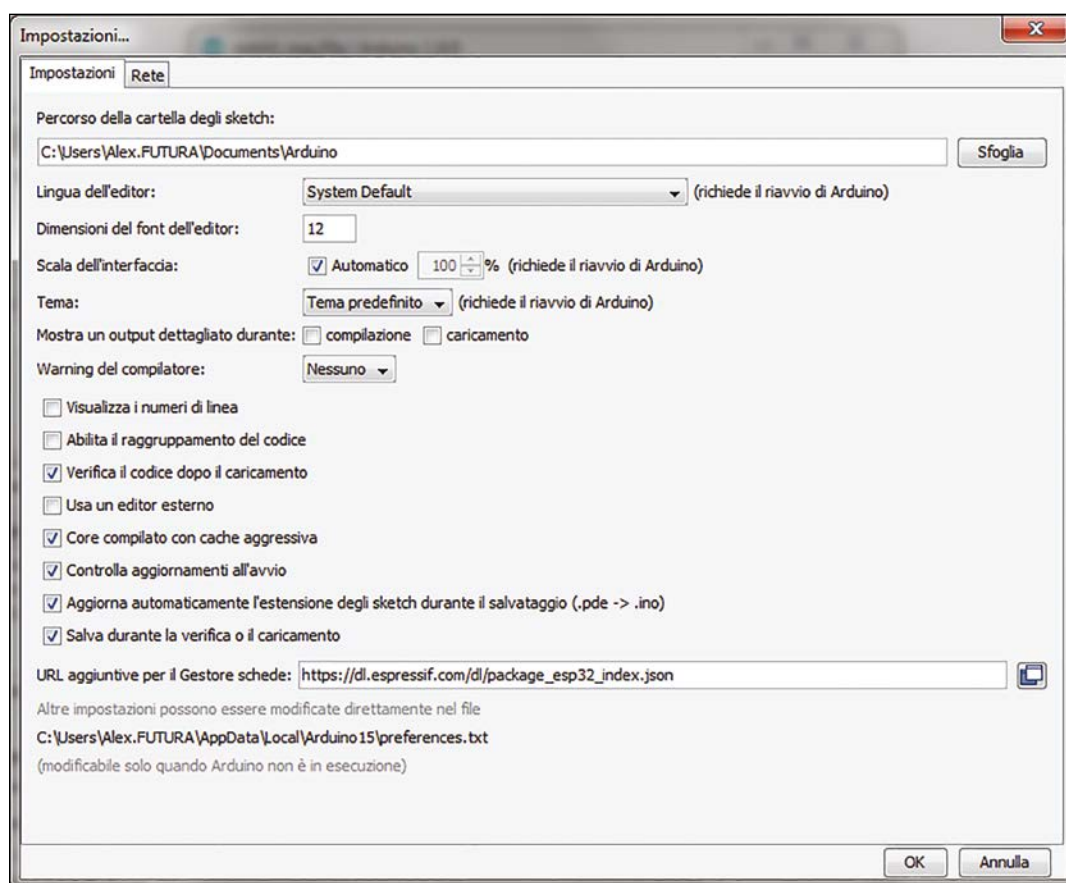
- Pin strip maschio 4 vie passo 2.54mm (1 pz.)
- Pin strip maschio 6 vie passo 2.54mm (3 pz.)
- Pin strip maschio 7 vie passo 2.54mm (1 pz.)
- Pin strip maschio 15 vie passo 2.54mm (2 pz.)
- Pin strip femmina 2 vie passo 2.54mm (4 pz.)
- Pin strip femmina 4 vie passo 2.54mm (2 pz.)
- Pin strip femmina 5 vie passo 2.54mm (1 pz.)
- Pin strip femmina 6 vie passo 2.54mm (2 pz.)
- Pin strip femmina 7 vie passo 2.54mm (1 pz.)
- Pin strip femmina 15 vie passo 2.54mm (2 pz.)
- Pin strip femmina 4 vie passo 2.54mm 90° (1 pz.)
- Jumper (2 pz.)
- Morsetto 3 vie passo 5 mm
- Breadboard 170 contatti
- Plug alimentazione
- Circuito stampato S1632 (140x110 mm)

La libreria prevista per il controller grafico ha una struttura modulare e alcuni moduli possono essere esclusi dalla compilazione per ridurre l'utilizzo di spazio nella Flash memory. Richiede pochissima RAM e supporta, oltre alla comunicazione I²C-Bus, anche la SPI: nel primo caso si esegue un'implementazione software (libreria Wire, AVR Twi, Linux i2c-dev) e nel secondo un'implementazione SPI a 4 fili (libreria Arduino SPI, AVR Spi, modulo AVR USI). Il controller permette di realizzare funzioni grafiche primitive (linee, rettangoli, pixel, bitmap) e stampa del testo (utilizzando caratteri di dimensioni di-

verse, è possibile utilizzare GLCD Font Creator per creare nuovi caratteri).

La gestione del display avviene da parte del modulo U1, mediante il bus I²C comune ad altre unità di bordo come i moduli U3 (sensore di qualità dell'aria) U4 (rilevatore di fulmini) U5 (rilevatore di gesture) U7 (sensore di pressione) U8 (modulo RFID/NFC) e riportato all'esterno tramite il connettore siglato I2C, per interfacciare la demoboard con unità esterne. Nello schema elettrico potete notare che l'alimentazione dell'OLED1 avviene tramite due jumper a tre vie che vanno disposti l'uno com-

Fig. 2
Aggiunta del
percorso di ricerca
del driver di terze
parti.



plementarmente all'altro; lo scopo è consentire di montare le varie versioni esistenti in commercio, che possono avere i pin di alimentazione + e - (1 e 2) invertiti; quindi se il pin 1 è il positivo, chiuderete il jumper J1 su +3V3 e il J2 su GND, mentre se è il negativo, farete il contrario (J1 su GND e J2 su +3V3).

Il nostro modulo ESP32 implementa anche un bus SPI a 4 fili (MISO, MOSI, SCK e CS) utile per interfacciare la demoboard a dispositivi SPI (in questo caso si utilizza il connettore omonimo...) e sensori vari; a riguardo va detto che malgrado il sensore on-board U4 supporti la comunicazione bivalente SPI/I²C-Bus, abbiamo preferito interfacciarlo all'U1 tramite I²C-Bus già descritto, che fa capo alle linee D21/SDA e D22/SCL dell'ESP32. Invece il bus SPI completo fa capo alle linee di I/O D13 (MOSI) D12 (MISO) D14 (SCK) e D15 (CS-Chip Select) dell'ESP32 e sul relativo connettore riportiamo anche GND e alimentazione +3,3 volt.

Oltre ai connettori e ai moduli sinora descritti, la demoboard riporta due LED multifunzione pilotati dalle linee D25 e D26 del modulo U1 (nella comunicazione USB servono a indicare la commutazione

del canale dati...) gestibili da firmware in base all'applicazione desiderata; a completare il circuito abbiamo previsto cinque pulsanti unipolari normalmente aperti utilizzabili ad esempio per impostare dei valori, delle soglie, ovvero per gestire menu a display, non a caso i tasti hanno associate queste funzioni:

- P1 = OK;
- P2 = UP;
- P3 = DOWN;
- P4 = LEFT;
- P5 = RIGHT.

E la loro disposizione sul circuito stampato ricalca un po' quella di un classico joystick. Se intendete utilizzare i pulsanti per altre funzioni, potete farlo tranquillamente (modificando l'esempio firmware che verrà proposto di volta in volta tra gli esempi) e rinominarli in base alle vostre esigenze.

A completare la demoboard abbiamo inserito una piccola breadboard a 170 contatti posizionata centralmente, utilizzabile sia per cablare breakout board utilizzando jumper Arduino e i contatti o

connettori SIL dei moduli on-board, sia per collegare componenti attivi e passivi a specifiche parti del circuito, con cui fare piccoli circuiti.

REALIZZAZIONE PRATICA

Bene, a questo punto abbiamo descritto la scheda demo, possiamo passare alla parte costruttiva: la realizzazione della demoboard inizierà dal circuito stampato a doppia faccia sul quale prenderanno posto tutti i componenti necessari, i quali, per rendere il progetto più accessibile ai maker, sono a montaggio THT ossia a foro passante o tradizionale, se preferite. Quindi è possibile, dopo aver ottenuto il circuito stampato, realizzare la scheda con un minimo di attrezzatura, che si limita a un saldatore per elettronica, del filo di lega saldante e un tronchesino per tagliare gli avanzi dei terminali dei componenti.

Una volta preparata la basetta per fotoincisione (ovvero dopo averla ottenuta dal servizio di realizzazione PCB offerto da Futura Elettronica, alla pagina web del sito <https://futura.net.it/prodotto/servizio-realizzazione-pcb/> partendo dalle due tracce lato rame scaricabili dal nostro sito web www.elettronica.in.it, potete passare al montaggio vero e proprio.

Iniziate posizionando e saldando i componenti a più basso profilo, quindi resistenze, LED, condensatori non polarizzati, pin-strip e strip femmina per le connessioni e per i moduli che saranno ospitati a bordo, poi i LED, i pulsanti e i condensatori elettrolitici, infine il jack di alimentazione. Montate poi la breadboard, che va semplicemente incollata, nella posizione indicata dal piano di montaggio, con del biadesivo perché non ha piedini o reofori. Per i componenti polarizzati (semiconduttori ed elettrolitici) dovete rispettare scrupolosamente il verso d'inserimento indicato nel piano di montaggio che trovate in queste pagine.

Non dimenticate di prestare attenzione alla polarità dei contatti di alimentazione del display OLED

e di impostare di conseguenza (come già spiegato qualche paragrafo indietro...) i relativi jumper J1 e J2 per applicare l'alimentazione con la polarità corretta.

Per alimentare la scheda è necessario fare uso di una fonte (non necessariamente stabilizzata) di alimentazione che eroghi una tensione di 8 ± 12 volt in continua ed una corrente di almeno 1A (in realtà la corrente dipende da quanti e quali moduli monterete).

Ricordiamo che sebbene nelle foto in queste pagine vedete la demoboard completa, in realtà non serve montarvi tutti i moduli ma, di volta in volta, quelli con cui dovrete lavorare; resta inteso che due devono sempre essere presenti: il DC/DC (U2) e chiaramente l'U1, vale a dire il modulo contenente il SoC ESP32.

PREPARAZIONE IDE ARDUINO

Quando avete completato la vostra demoboard a livello hardware, potete concentrarvi sugli aspetti software, che riguardano la creazione delle applicazioni e la programmazione del modulo ESP32. Come prima cosa sarà necessario installare la board ESP32 in modo che possa poi essere programmata dall'IDE. Per farlo dovete aprire l'IDE, quindi dal menu "File > Impostazioni" inserire la stringa https://dl.espressif.com/dl/package_esp32_index.json nella casella URL aggiuntive per il Gestore schede (Fig. 2).

Fatto ciò, dal menu "Strumenti > Scheda" dovete impartire il comando "Gestore Schede" e selezionare la vostra scheda ESP32, ovvero digitare ESP32 nella casella di testo in alto e procedere con l'installazione del pacchetto board ESP32 by Espressif Systems (Fig. 3).

Ora, sempre dal menu "Strumenti > Scheda > Gestore Schede" scegliere la board "DOIT ESP32 DEVKIT V1" come mostrato in Fig. 4.

Tutto è pronto per la programmazione, pertanto da questo punto in poi si potrà scrivere il codice

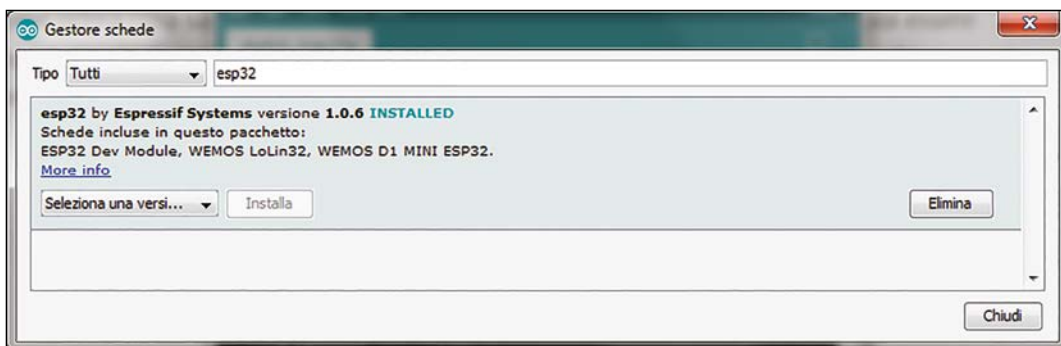


Fig. 3
Installazione
del pacchetto
per l'ESP32.

↓ Listato 1

```

/*****
This is an example for our Monochrome OLEDs based on SSD1306 drivers

This example is for a 128x64 pixel display using I2C to communicate
3 pins are required to interface (two I2C and one reset).

Adafruit invests time and resources providing this open
source code, please support Adafruit and open-source
hardware by purchasing products from Adafruit!

Written by Limor Fried/Ladyada for Adafruit Industries,
with contributions from the open source community.
BSD license, check license.txt for more information
All text above, and the splash screen below must be
included in any redistribution.
*****/

#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
// The pins for I2C are defined by the Wire-library.
// On an arduino UNO:      A4(SDA), A5(SCL)
// On an arduino MEGA 2560: 20(SDA), 21(SCL)
// On an arduino LEONARDO:  2(SDA),  3(SCL), ...
#define OLED_RESET    -1 // Reset pin # (or -1 if sharing Arduino reset pin)
#define SCREEN_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

#define NUMFLAKES 10 // Number of snowflakes in the animation example

#define LOGO_HEIGHT 16
#define LOGO_WIDTH 16

const int Sens = 33;

void setup()
{
  Serial.begin(9600);

  // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
  if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }

  // Show initial display buffer contents on the screen --
  // the library initializes this with an Adafruit splash screen.
  display.display();
  delay(2000); // Pause for 2 seconds

  // Clear the buffer
  display.clearDisplay();
  display.setTextSize(2); // Draw 2X-scale text
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(5, 0);
  display.print("** Test **");
  display.setCursor(20, 30);
  display.print(" Analog ");

  // Show the display buffer on the screen. You MUST call display() after
  // drawing commands to make them visible on screen!
  display.display();
  delay(2000);
  // display.display() is NOT necessary after every single drawing command,
  // unless that's what you want...rather, you can batch up a bunch of

```



```

// drawing operations and then update the screen all at once by calling
// display.display(). These examples demonstrate both approaches...

/*
// Invert and restore display, pausing in-between
display.invertDisplay(true);
delay(1000);
display.invertDisplay(false);
delay(1000);
*/
pinMode(Sens, INPUT);
}

void loop()
{
  int val_adc = 0;
  float val_tens;
  String txt_msg;
  val_adc=analogRead(Sens);
  val_tens = float(500)/float(4096);
  val_tens = float(val_tens) * float(val_adc);
  val_tens = float(val_tens) / float(100);
  txt_msg="";
  txt_msg = String(val_tens,4) + "V";
  Serial.println(txt_msg);
  OLED_Message("5V max for CNU9-OUT",txt_msg);
  delay(500);
}

void OLED_Message(String text_display1, String text_display2)
{
  int lunghezza = 0;
  display.clearDisplay();
  display.setTextSize(1); // Draw 2X-scale text
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 20);
  lunghezza = text_display1.length();
  lunghezza=21-lunghezza;
  lunghezza=lunghezza/2;
  for (int i=0;i<lunghezza;i=i+1)
  {
    display.print(" ");
  }
  display.print(text_display1);
  display.setCursor(0, 40);
  lunghezza = text_display2.length();
  lunghezza=21-lunghezza;
  lunghezza=lunghezza/2;
  for (int i=0;i<lunghezza;i=i+1)
  {
    display.print(" ");
  }
  display.print(text_display2);
  display.display();
}

```

necessario per la gestione della breakout board d'interesse. A tal proposito abbiamo già reso disponibili una serie di Sketch di esempio per ogni breakout, pertanto avrete sempre disponibile una base di partenza che semplifica senz'altro lo sviluppo ad hoc del codice. Le cartelle con gli sketch scaricabili dal nostro sito www.elettronica.it sono quelle proposte nello screenshot di **Fig. 5**. Nella cartelle di ogni singolo esempio si

troverà disponibile la libreria ad esso abbinata, qualora il codice lo renda necessario. Inoltre, sempre in ogni cartella sarà disponibile il file "info.txt" che spiega in breve lo scopo dell'esempio.

Quasi per tutti gli esempi sarà necessario avere montato il modulo display OLED da 0,96"; a tal proposito ricordiamo che a bordo scheda è possibile utilizzare sia i display OLED con la pinout

Fig. 4
Selezione della scheda nell'IDE Arduino.

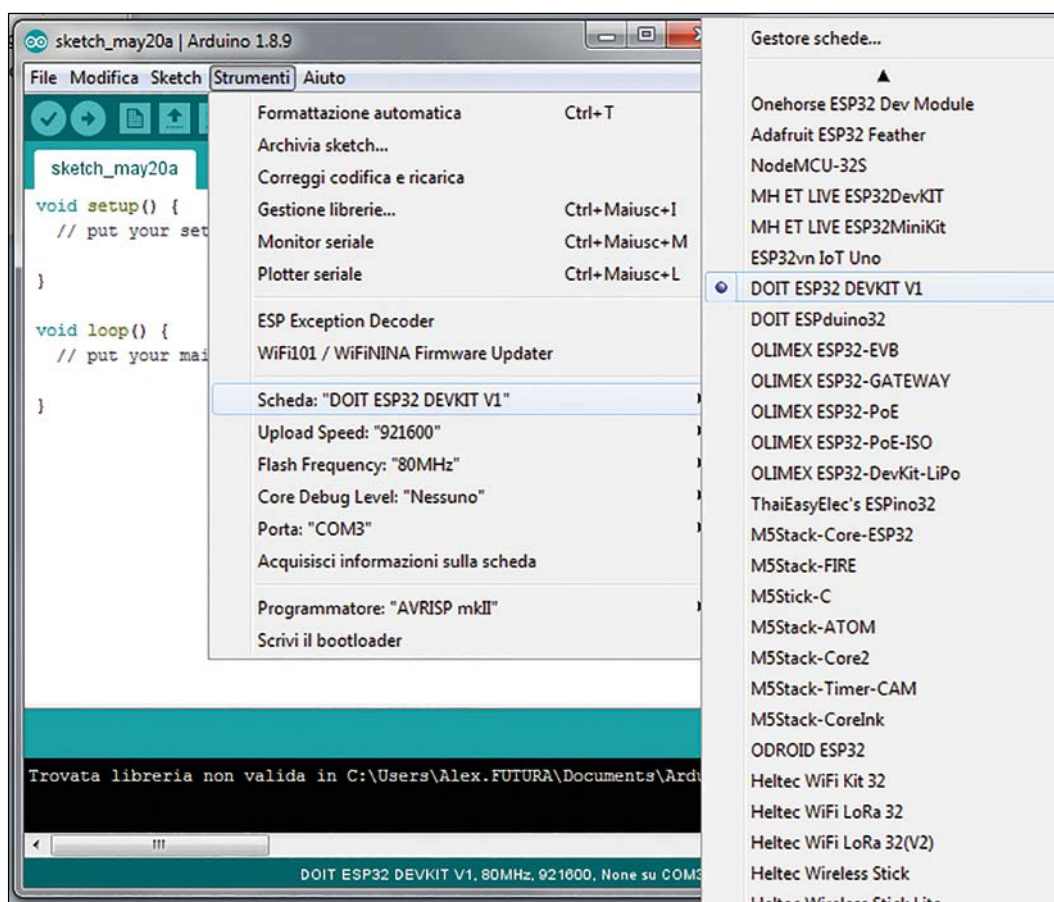
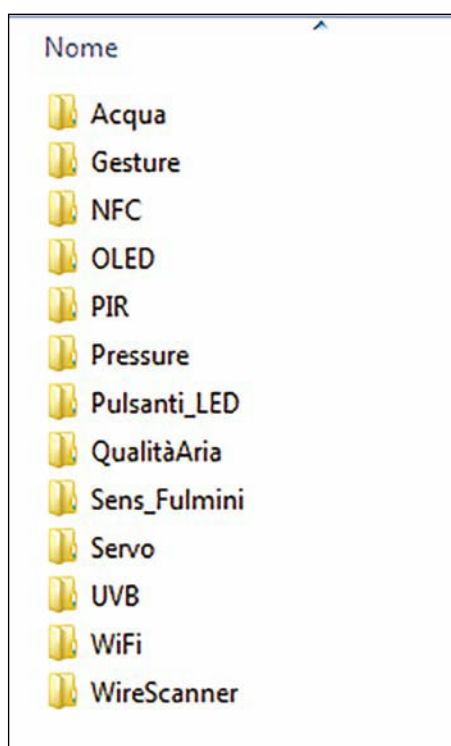


Fig. 5
Le cartelle contenenti gli esempi di sketch proposti.



GND/VCC/SCL/SDA, come pure quelli aventi una piedinatura del tipo VCC/GND/SCL/SDA, semplicemente andando a configurare i due jumper J1 (per il pin 1) e J2 (per il pin 2) in modo da portare sul pin 1 o 2 del display la GND piuttosto che VCC a seconda del display installato.

PRIMA APPLICAZIONE: SENSORE ACQUA

Questo esempio applicativo consiste nella lettura del valore di "torbidità" dell'acqua mediante lo specifico sensore collegabile allo strip CN9. Questo connettore può anche essere utilizzato per leggere una tensione massima di 5V, in quanto vi è un partitore all'ingresso che ne abbassa la tensione ad un massimo di 3,3V; tale riduzione diventa necessaria perché l'ingresso corrispondente dell'ESP32 sopporta una tensione ordinaria di 3,3V ed una massima di 5V, oltre cui il modulo ESP32 viene danneggiato.

Il sensore utilizzato è il 9145-TORBISENS reperibile sul sito www.futuranet.it (Fig. 6) e consta di un elemento sensore vero e proprio, da immergere parzialmente nell'acqua di cui si vuole rilevare la

l'impurezza o torbidità, ed una scheda di condizionamento del segnale e interfaccia, che fornirà un'informazione digitale o analogica e che dovrete collegare al connettore CNU9.

Il sensore è in grado di rilevare le particelle sospese nell'acqua (o altro liquido, purché non aggressivo chimico o solvente) misurando la trasmissione della luce e la velocità di dispersione che cambia con la quantità totale di solidi sospesi (TSS) in acqua. All'aumentare del TTS (Total Suspended Solids), il livello di torbidità liquida aumenta. Dispone di uscita analogica e digitale.

Per l'uso di questo sensore è bene ricordare che va immersa la sola parte in plastica trasparente con le due "punte" perché la parte superiore della sonda (quella dove si collega il cavetto) non è a tenuta d'acqua. La temperatura dei liquidi in cui il sensore può essere immerso deve essere compresa fra +5°C e +90°C: tenetene conto nell'utilizzo. Una volta introdotta nell'acqua la parte sensibile, il sensore fornisce il relativo valore in un tempo



Fig. 6
Il sensore di qualità dell'acqua (in alto a destra) con i relativi cavetti e la scheda d'interfaccia (in basso a destra).

dell'ordine dei 500 ms. Se l'uscita è impostata come analogica, fornisce una tensione da 0 a 4,5 V inversamente proporzionale alla torbidità del liquido; se invece è impostata come digitale, la soglia di torbidità si può regolare tramite l'apposito trimmer

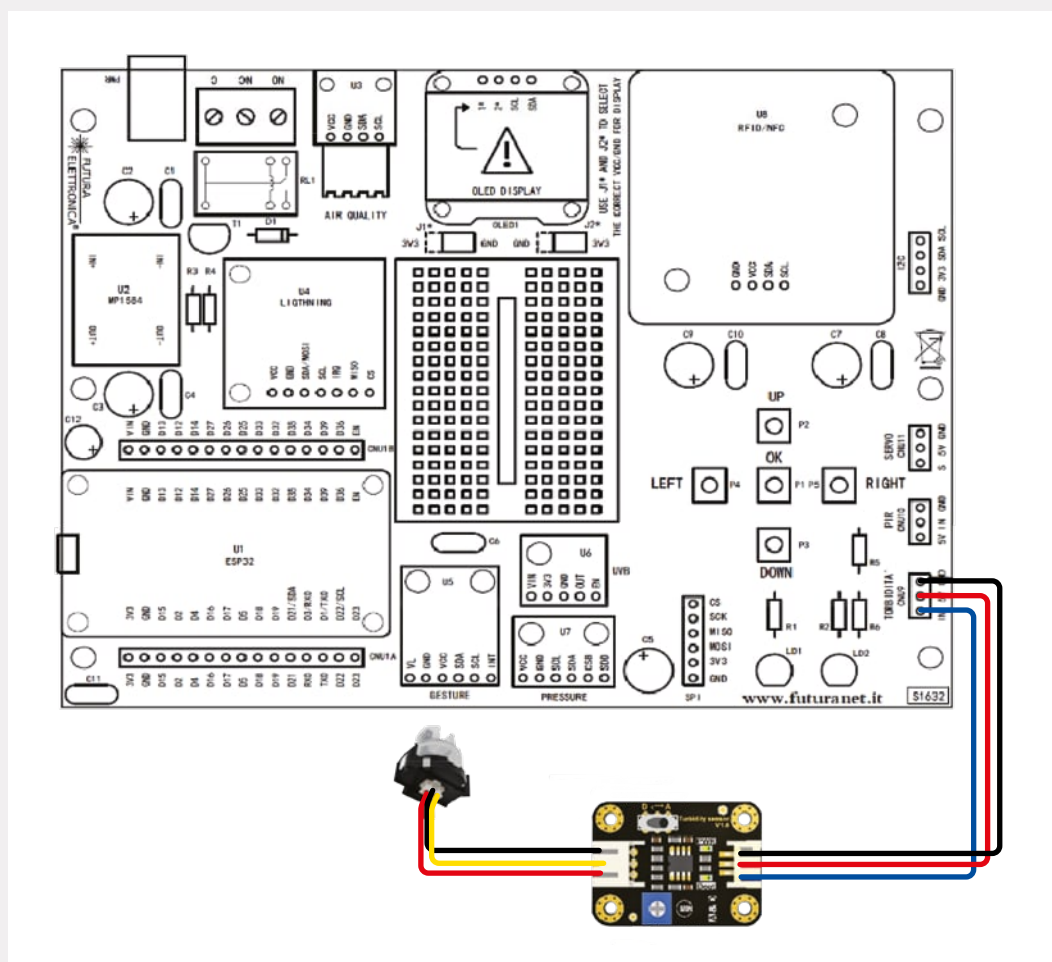


Fig. 7
Cablaggio del sensore di torbidità dei liquidi alla demoboard.

Dai alle tue piante il giusto nutrimento!

Misuratore di luce-umidità-PH



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

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

Il misuratore non richiede nessuna fonte di alimentazione.

**FUTURA
ELETTRONICA®**

www.futurashop.it

sulla scheda d'interfaccia (0=torbidità; 1=limpido). Lo sketch che vi proponiamo di caricare per utilizzare il sensore di torbidità dell'acqua è quello che vedete proposto nel **Listato 1**.

Quanto al collegamento tra sensore, scheda d'interfaccia in dotazione e demoboard, vale lo schema di cablaggio proposto nella **Fig. 7**; nell'immagine, i colori dei fili sono rosso per il +5V, nero per la massa (GND) e blu per il segnale.

CONCLUSIONI

Bene, con questo primo esempio di utilizzo terminiamo la panoramica sulla demoboard ESP32; nella prossima puntata vi proporremo nuovi sketch e nuove applicazioni che coinvolgeranno sia i sensori e i moduli on-board, sia moduli collegabili esternamente mediante i connettori laterali. La demoboard proposta in queste pagine nasce per sviluppare e prototipare applicazioni "connesse" che sfruttano l'interfaccia Wireless WiFi per pubblicare o trasferire dati relativi ai sensori e lettori presenti a bordo o collegati esternamente tramite le connessioni di cui dispone; gli stessi parametri possono essere parzialmente gestiti e visualizzati, in funzione di monitor, dal display OLED di bordo. La ricca dotazione di sensori rende la scheda candidata ideale allo sviluppo e la prototipazione di applicazioni nel campo dell'IoT proprio in virtù della funzione di bridge tra dati dei sensori e Internet o connessione wireless locale.

Con qualche riga di codice in più è anche possibile eseguire un minimo di Edge Computing in modo da aggregare i dati di più sensori per risparmiare banda nella trasmissione verso la Rete. □

Cosa occorre?

La demoboard per ESP32 (cod. FT1632) è disponibile in scatola di montaggio a Euro 25,00 e comprende tutti i componenti ad eccezione dei sensori, display e moduli. Questi sono disponibili separatamente. In particolare il modulo con ESP32 (cod. YB555) è in vendita a Euro 13,50, il display OLED (cod. OLED096C) costa Euro 11,00, il sensore qualità dell'aria (cod. SGP30C02) è disponibile a Euro 19,50, il sensore fulmini (cod. FT1324M) è in vendita a Euro 25,00, il modulo gesture (cod. APDS9960BOUT) costa Euro 7,50, il sensore UVB (cod. GYML8511) è in vendita a Euro 8,00, il sensore di pressione (cod. BMP280) costa Euro 4,50 e il modulo RFID (cod. PN532) è disponibile 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.futurashop.it

SMART ALARM

di VINCENZO GERMANO e MARCONATALE PARISE

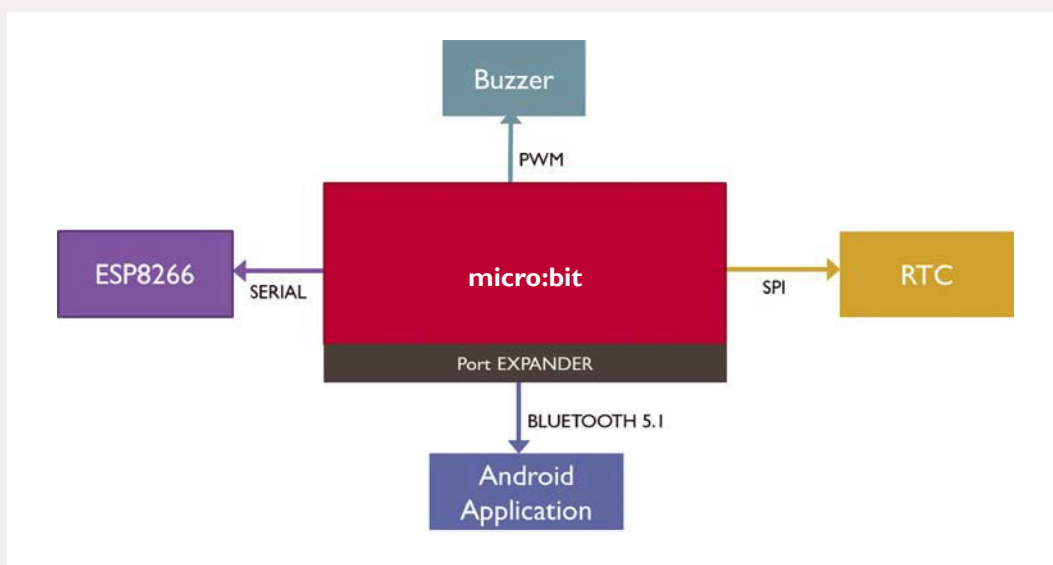


Aggiungiamo funzionalità al progetto di domotica sviluppato con micro:bit inserendo la temporizzazione avanzata e le sveglie.

C

ome analizzato negli articoli precedenti, la domotica è quell'insieme di automazioni e tecnologie domestiche che puntano a migliorare l'abitabilità e la funzionalità delle nostre case, sfruttando lo studio e l'applicazione di tecnologie basate sull'ingegneria informatica ed elettronica, che mirano alla realizzazione di dispositivi integrati in grado di automatizzare ma anche facilitare operazioni solitamente svolte in un alloggio. Questo con il fine di ottimizzare i consumi e/o le funzionalità di un appartamento, l'automatizzazione di alcune attività

Fig. 1
Diagramma a
blocchi.



o anche la gestione ad alto livello di sistemi che rendono l'utente finale capace di controllare diversi strumenti attraverso tecnologie dedicate. Proprio da questo punto di vista la domotica tende a rendere intelligenti i dispositivi, gli impianti elettrici e i sistemi degli appartamenti e con questa rassegna di articoli poniamo al centro la scheda micro:bit e la sua tecnologia evoluta per guidarvi nel controllare funzionalità domestiche attraverso la gestione di differenti tecnologie, dando la possibilità all'utente di un controllo semplice ed efficace del proprio sistema "home-made".

Nei precedenti articoli abbiamo visto come sia possibile gestire in modo automatico o manuale l'impianto di illuminazione casalingo e per semplicità di trattazione ne abbiamo fatto un'equivalenza con l'accensione di un singolo LED (Light Emitting Diode), tuttavia chiarendo che al posto di esso è possibile inserire per esempio la gestione di un relè o una board elettronica che pilota una o più strisce LED, lampade, neon oppure un intero impianto d'illuminazione domestico. Successivamente abbiamo realizzato e sviluppato un'applicazione dedicata mediante Android Studio, analizzato come modificarla ed evolverla a proprio piacimento per centralizzare la gestione del sistema di domotica. Non solo, abbiamo continuato ad aggiungere funzionalità al sistema mettendo in pratica un'altra equivalenza come fatto per la gestione LED, ovvero aggiungendo la gestione di una ventola per il ricircolo d'aria forzato nell'appartamento che può essere gestita manualmente o in automatico a seconda delle necessità dell'utente o di parametri ambientali come la temperatura e/o l'umidità.

Infine si è esteso il sistema implementando la gestione di tapparelle/persiane utilizzando un motore stepper, per poi visualizzare le informazioni di tutto il sistema su un LCD.

Tutte le funzionalità implementate nel sistema sono man mano state aggiunte nell'applicazione presentata per essere in linea con la filosofia domotica e poterle gestire in modo centralizzato. A questo punto introduciamo nel sistema la funzionalità della sveglia andando a complicare l'architettura e considerando due moduli aggiuntivi, ovvero il modulo RTC (Real Time Clock – orologio in tempo reale) e l'ESP8266. Ovviamente ci sono possibilità differenti (e anche semplificate) per implementare una funzionalità di questo tipo ma il tutto viene illustrato per scopo didattico e per mostrare come i vari moduli possano comunicare tra loro.

Prima di passare all'implementazione software su micro:bit e all'aggiornamento dell'applicazione Android, analizziamo l'hardware necessario illustrandone le caratteristiche di ogni componente di progetto per meglio comprendere le fasi successive.

SETUP DI SISTEMA

Si riporta in **Fig. 1** un diagramma a blocchi nel quale è immediatamente visibile il micro:bit al centro di tutto il sistema, a esso vengono connessi il buzzer (per la gestione delle segnalazioni acustiche), il modulo RTC (per permettere al sistema di avere una gestione temporale accurata) ed infine il modulo ESP8266 collegato mediante seriale; infine, grazie al Bluetooth il micro:bit comunica con il proprio smartphone e tramite Matrice LED

del micro:bit vengono visualizzati i messaggi di sistema. Nello schema a blocchi vengono messi in risalto anche le tipologie di comunicazione o i particolari protocolli utilizzati per lo scambio di dati tra i vari blocchi.

Per un agevole reperimento del materiale di questo progetto e per evitare la realizzazione di board dedicate, mostreremo alcune soluzioni di schede elettroniche integrate "pronte all'uso" a costi contenuti che permettono la realizzazione delle funzioni d'interesse, reperibili facilmente sul sito futuranet.it.

LA SCHEDA D'ESPANSIONE

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

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

Tra le varie soluzioni disponibili, quella più compatta e flessibile è sicuramente un "Port Expander" (acquistabile sullo store futurashop.it con il codice MICROBITPROTO) in grado di rendere di facile utilizzo tutte le porte di input/output della board. In **Fig. 2** è visibile il TOP della scheda e si comprende rapidamente come il connettore del micro:bit possa essere reso usabile/accessibile mediante pin header colorati in vario modo, equivalentemente a quanto accade con Arduino, Raspberry Pi altro. Oltre l'immediata possibilità di riportare la connessione in un ambito "standard", ci sono vari vantaggi aggiuntivi, come la possibilità di essere alimentata tramite i due connettori rossi, con tensioni variabili fino a 24V o fino a 6V e poter sfruttare uno dei due regolatori stabilizzati di tensione da 5V o 3.3V; oppure la possibilità della scheda di essere alimentata direttamente tramite la porta USB di micro:bit. Riguardo i pin header con passo 2,54mm, si hanno le seguenti funzionalità:

- colore giallo e blu per le porte di I/O e protocolli di comunicazione (I2C, SPI e Seriale)

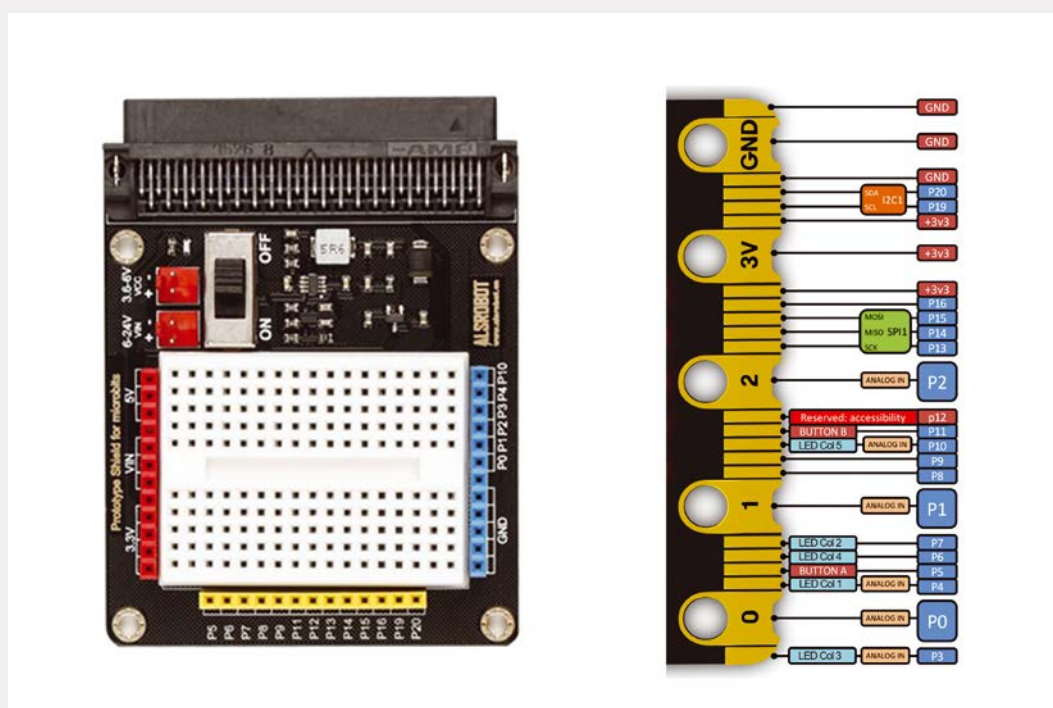


Fig. 2
Port
Expander e
pinout.

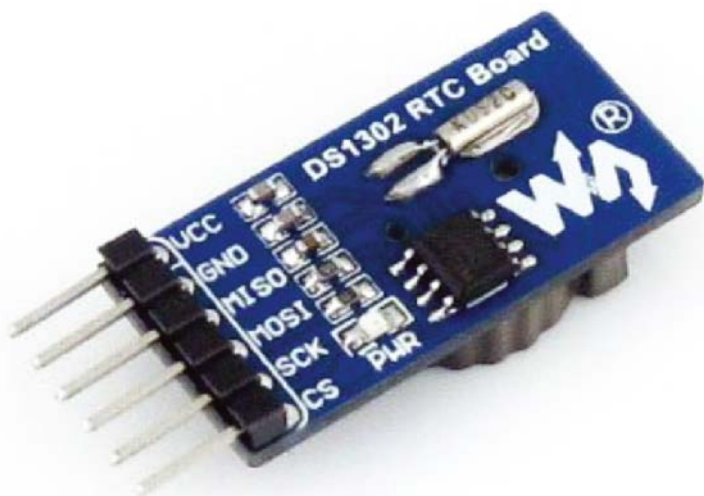


Fig. 3
Modulo RTC.

- colore blu parzialmente per il riferimento GND
- colore rosso per le tensioni d'alimentazione

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

IL MODULO RTC

A questo punto per avere un orario di sistema preciso nel tempo, tra i vari moduli disponibili sul mercato la scelta è ricaduta sulla scheda, visibile in **Fig. 3**, acquistabile sullo store *futurashop.it* (cod. DS1302RTC). Questo modulo Real Time Clock (RTC) si basa sull'integrato della Maxim DS1302, al suo interno ha a disposizione una RAM (Random Access Memory – memoria ad accesso casuale) statica da 31 byte e un orologio in tempo reale che permette di conteggiare i secondi, minuti, ore, giorno, data, mese e anno, con correzione per gli anni bisestili fino al 2100 ed un formato ora 12/24 (con indicazione AM/PM). La data di fine mese viene regolata automaticamente per i mesi con meno di 31 giorni, comprese le correzioni per l'anno bisestile. I dati trasferiti possono essere sia letti che inviati verso il modulo e grazie ad un sistema di backup interno, può mantenerli in memoria anche in assenza di corrente grazie a un alloggiamento per una batteria a bottone, che mantiene aggiornato l'orologio anche in assenza della alimentazione principale; perciò se si verifica una interruzione dell'alimentazione principale, il dispositivo può

continuare a fornire tempi precisi (in modo tale che le interruzioni non influenzano le prestazioni). Progettato per funzionare con una potenza molto bassa e per conservare dati e informazioni di clock con meno di 1µW, è molto indicato in tutte le applicazioni dove è necessario avere un preciso riferimento temporale.

Questo modulo utilizza un cristallo esterno a 32,768 kHz (direttamente montato su scheda) e utilizzando un cristallo con caratteristiche specificate, il tempo di avvio è generalmente inferiore a un secondo. L'oscillatore integrato migliora l'accuratezza a lungo termine del dispositivo e riduce il numero di componenti esterni, infatti il circuito dell'oscillatore non richiede resistori o condensatori esterni per funzionare. Tuttavia, la frequenza dell'oscillatore cambia con la temperatura ambiente e l'errore di frequenza dell'oscillatore esterno può causare un errore di clock di sistema. Possiamo riassumere le caratteristiche principali come segue:

- Tensione applicabile: 2.0V - 5.5V
- Tensione batteria di backup: 3.0V - 5.0V
- Temperatura di lavoro: 0°C - +70°C
- Supporto batteria on board
- Interfaccia SPI, supporta 3 fili o 4 fili di comunicazione seriale
- Consumo: <1mW (mantenendo dati/informazioni di clock)
- Dimensioni: 31x15 mm

Come si nota dalla **Fig. 3**, il modulo comunica con un microprocessore tramite una semplice interfaccia seriale SPI (Serial Peripheral Interface), perciò sono necessarie solo quattro linee per comunicare con il modulo: Chip select (CS), Master Input Slave Output (MISO – ingresso per il master e uscita per lo slave), Master Output Slave Input (MOSI – uscita dal master) e SCK (orologio seriale emesso dal master), che possono essere riportate direttamente nel micro:bit per la comunicazione.

IL BUZZER

Passando oltre e riconsiderando il diagramma a blocchi di **Fig. 1**, un altro componente gestito dal micro:bit è il buzzer (o noto anche come cicalino), ovvero un dispositivo di segnalazione audio che generalmente troviamo in molti apparecchi di uso comune, come possono essere sveglie, forni a microonde, lavatrici, lavastoviglie, frigoriferi e molto altro. Ne esistono di vari tipi ma possiamo fare un breve confronto tra le due varianti maggiormente utilizzate e comuni rispetto alle altre, i cicalini

piezoelettrici e quelli magnetici.

Come anticipato sono dispositivi "acustici elettrici" che generano dei suoni, generalmente vengono alimentati da correnti alternate e sono disponibili in diversi package in base anche alla tipologia di suono differente emesso. È facile immaginare come un cicalino piezoelettrico abbia un principio di funzionamento e una struttura interna differenti da uno magnetico, questo perché come ci suggerisce il nome il suo nucleo è realizzato da un elemento piezoelettrico, quindi realizzato in ceramica piezoelettrica e "incollata" ad una piastra metallica. Quando questa struttura viene attraversata da una corrente alternata, essa si restringe e si espande, generando una vibrazione che produce di conseguenza onde sonore.

Per quanto riguarda i buzzer magnetici, sono più tradizionali nella loro composizione rispetto ai cicalini piezoelettrici, infatti sono realizzati da un magnete, una bobina e dal disco vibrante ferromagnetico. La corrente viene azionata attraverso una bobina mobile per produrre un campo magnetico; quando viene applicata una tensione, la bobina genera un campo magnetico e quindi consente al diaframma di vibrare e produrre suono. In generale questo tipo di cicalino ha un basso livello di pressione sonora, una tensione operativa compresa tra 1,5 e 16 V e un elevato assorbimento di corrente, comunemente maggiore di 20 mA e minore di 100mA. Per l'applicazione presentata viene utilizzato un cicalino appartenente alla seconda categoria ovvero magnetico; come visibile in **Fig. 4**, esso ha un package cilindrico (diametro 12 mm) da circuito stampato, una frequenza sonora di 2 kHz con una potenza di 95 dBA, con un'alimentazione compresa tra 3,5 e 5 Vdc e un conseguente consumo di 35 mA.

IL MODULO ESP8266

Per quanto non ha bisogno di nessuna presentazione, vediamo brevemente le caratteristiche principali del modulo ESP8266, ovvero è un System on a Chip (SoC) prodotto dalla società Espressif, costituito principalmente da un'unità di controllo (microcontrollore) e un ricetrasmittente Wi-Fi. Dato il suo basso costo, le dimensioni ridotte e l'adattabilità, è ampiamente utilizzato in applicazioni IoT (Internet of Things), infatti permette una connessione Wi-Fi a 2,4 GHz utilizzando lo standard IEEE 802.11 (mediante l'intero stack TCP/IP) quindi si può utilizzare per connettersi ad una rete Wi-Fi, a Internet, ospitare un server Web con pagine Web o altri servizi, consentire al



Fig. 4
Il buzzer.

tuo smartphone di connettersi ad esso, e molto altro; in più è programmabile sia con un linguaggio dedicato che mediante l'IDE di Arduino, perciò non c'è da stupirsi che sia diventato uno dei dispositivi IoT più popolare disponibile.

Ne esistono molti moduli diversi o schede di sviluppo complete che possono avere pin diversi, antenne Wi-Fi diverse o una diversa quantità di memoria flash a bordo. Bisogna però fare una distinzione, i formati con i quali è possibile acquistare l'ESP8266 sono sostanzialmente tre, il primo è la versione in chip, ovvero direttamente il componente da saldare su circuito stampato ma presuppone di realizzare un progetto dedicato su cui applicarlo; il secondo è mediante modulo come riportato in **Fig. 5**, con il chip montato su una scheda e l'elettronica minima per mettere in funzione il dispositivo; infine mediante schede di sviluppo dedicate che hanno preinstallati i moduli e si differenziano principalmente a seconda della tipologia di antenna, pin disponibili analogici/digitali, schermatura, led a disposizione e memoria.

Nel nostro caso abbiamo scelto l'ESP-01 acquistabile direttamente dal sito futuranet.it, perché

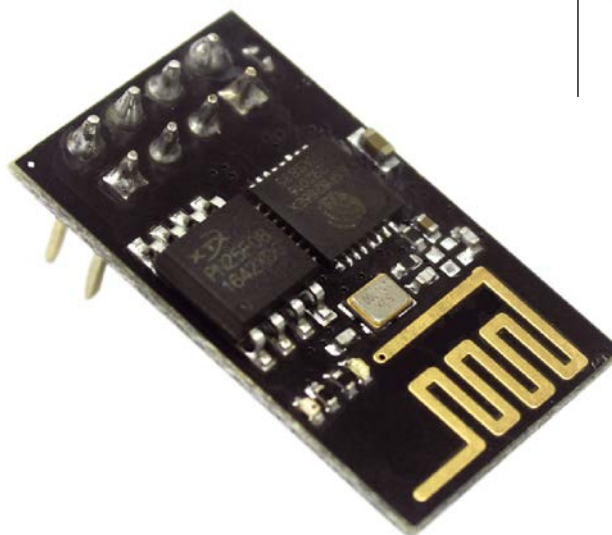


Fig. 5
Il modulo ESP8266.

Tabella 1
Connessioni fisiche tra i pin header
delle varie schede

PIN	FUNZIONI
P0	ESP8266 (Tx)
P1	ESP8266 (Rx)
P8	PWM Buzzer
P13	SCLK RTC
P14	MISO RTC
P15	CS RTC

ampiamente utilizzato in progetti di smart home e networking, grazie al firmware predefinito ne consente l'utilizzo in combinazione con un Arduino ed è pronto all'uso con tutto il necessario per la nostra applicazione.

DESCRIZIONE CONNESSIONI

A questo punto vediamo brevemente le interconnessioni fisiche presentate a livello logico nel diagramma a blocchi di **Fig. 1**, il setup di sistema tra il micro:bit e le varie periferiche. Ripartiamo dal considerare il port expander, con riferimento all'immagine di **Fig. 2**, abbiamo scelto di utilizzare la tensione di alimentazione V1, il riferimento G ed i connettori gialli per le periferiche. Per quanto

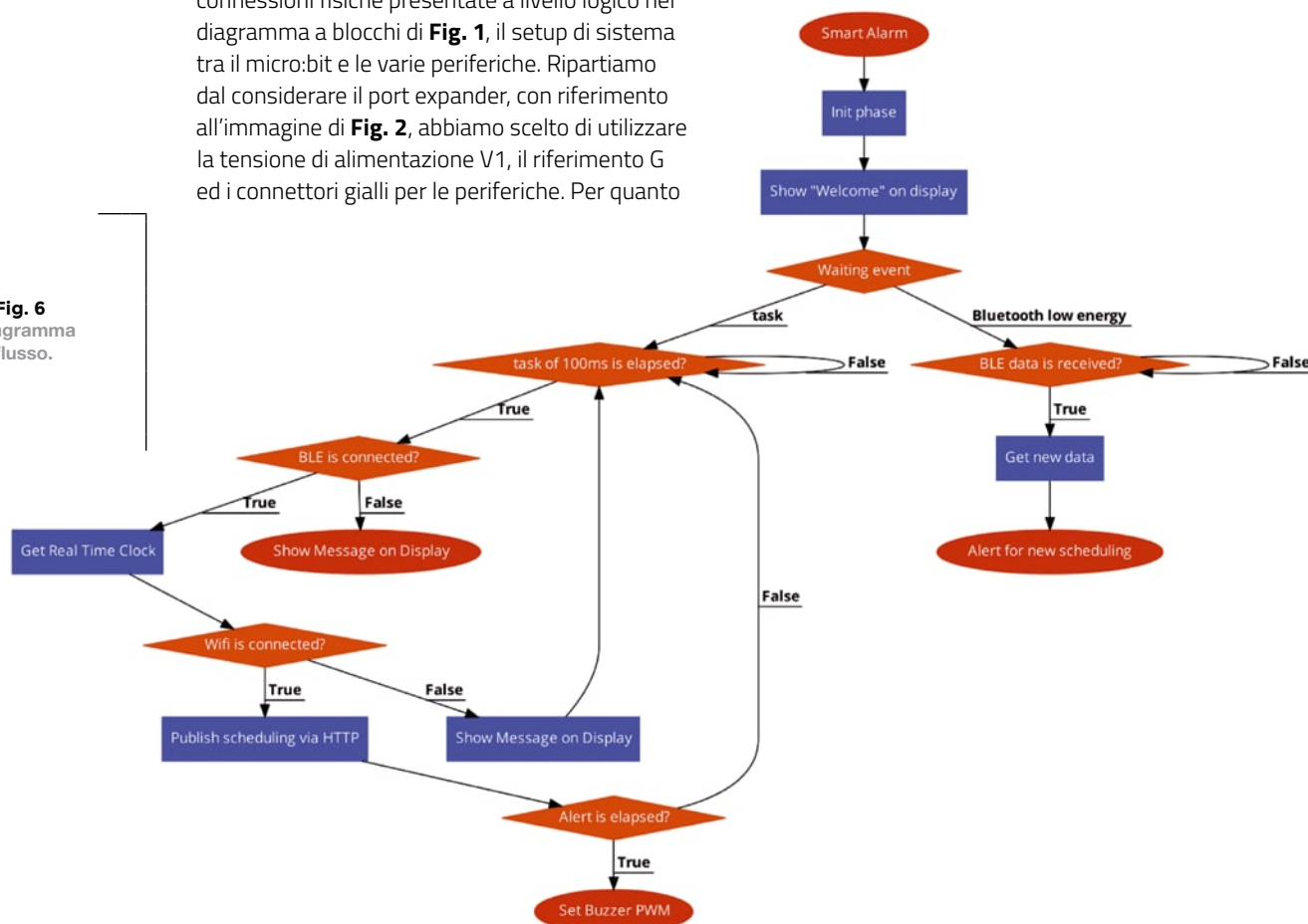
riguarda le tensioni di alimentazione è importante impostare la linea V1 su 5V, collegando i due jumper verdi come mostrato in figura. Per semplicità i pin d'interesse destinati alle periferiche vengono riportati in **Tabella 1** ed esplicitate le varie connessioni tra schede attraverso i loro pin-header, ma vediamo insieme.

Le connessioni principali sono quelle che vanno da micro:bit al buzzer, al modulo RTC ed infine all'ESP8266. Con riferimento alla **Tabella 1** vengono riportate tutte le connessioni appena elencate. Per i meno esperti, come vedremo successivamente, le librerie software utilizzate ci astrarranno da tutti questi dettagli e impostazioni, permettendoci di semplificare enormemente lo sviluppo software e rendendo la sua configurazione immediata, illustrando le funzioni più importanti per implementare la lettura dei dati d'interesse.

DESCRIZIONE LOGICHE DI FUNZIONAMENTO

La descrizione dell'implementazione software del

Fig. 6
Diagramma di flusso.



progetto richiede ulteriori dettagli sulle logiche che, per una migliore comprensione, verranno descritte brevemente attraverso il diagramma di flusso di **Fig. 6**. Il diagramma di flusso è uno strumento molto importante per il programmatore in quanto permette di avere una panoramica sulle possibili situazioni (case) che dovrà gestire e implementare. La prima fase dell'implementazione riguarda l'inizializzazione della periferica bluetooth, del protocollo SPI per la gestione RTC, del modulo ESP8266 e delle variabili di "appoggio" per una corretta implementazione. La logica di funzionamento si basa su due eventi:

1. Ricezione dati bluetooth e successiva decodifica dei dati.
2. Attesa della scadenza della sveglia e gestione delle periferiche.

La decodifica dei dati bluetooth (punto 1) prevede l'estrazione dei dati relativi alla programmazione sveglia gestita attraverso applicazione Android, che verrà descritta successivamente, il quale compito è fornire l'orario iniziale di condivisione dati e l'imminente orario della sveglia pianificata. Le informazioni che il micro:bit riceverà avranno il seguente formato: #lhhmmssEhhmmss; dove:

'l' = Initial Timer
'E' = End Timer
'hh' = hour
'mm' = minute
'ss' = second

I dati ricevuti via Bluetooth vengono codificati e valutata la scadenza della prossima sveglia. Quando il timer (ora, minuti e secondi) calcolato dal modulo RTC eguaglierà la pianificazione ricevuta via Bluetooth allora il buzzer verrà abilitato per avvisare l'utente. Successivamente, il micro:bit rimane in attesa di ricevere la successiva pianificazione che se considerata valida porterà ad una nuova inizializzazione delle periferiche e successiva attesa della scadenza timer, altrimenti verrà avvisato l'utente che non saranno previste nuove pianificazioni tramite la matrice led. Infine, una "pianificazione" è considerata valida se all'interno dell'applicazione Android ci sono schede di pianificazione attive, ma addentriamoci maggiormente nella trattazione per ulteriori dettagli.

DESCRIZIONE LIBRERIE

Sottolineiamo che per un'implementazione in linea con quanto presentato, lo strumento utilizzato è il

Fig. 7
Thingspeak
creazione
canale.

programma "Make Code Block" disponibile online sul sito <https://makecode.microbit.org/>. Quindi creiamo il nostro progetto assegnando il nome desiderato (ad esempio: SmartAlarm) e prima di procedere con la scrittura importiamo le librerie necessarie. Dalla sezione "Estensioni", aggiungiamo la libreria Bluetooth, successivamente cerchiamo ed aggiungiamo tramite la parola chiave ESP8266 la libreria ThingSpeak ESP8266 ed infine selezioniamo anche la libreria DS1302 per il modulo RTC.

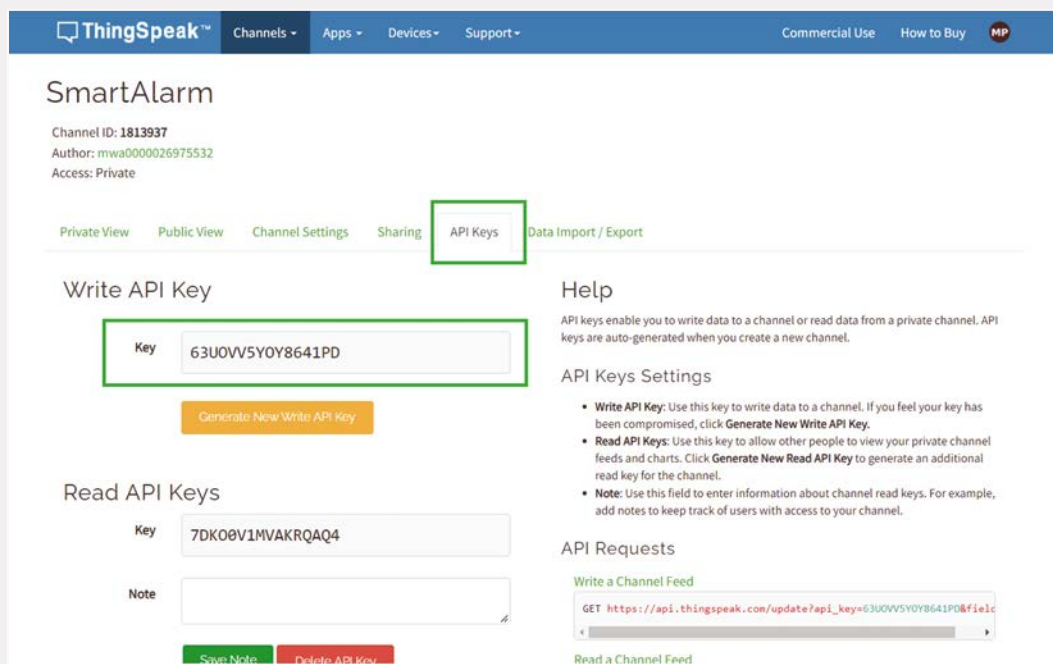
Tra le varie funzioni messe a disposizione dalla libreria ESP8266, troviamo principalmente:

- Initialize ESP8266: che inizializza il modulo attraverso la definizione dei pin da utilizzare per la comunicazione, il baud rate e le informazioni sulla rete di Wifi;
- Upload data to ThingSpeak: che consente di caricare via HTTP 8 byte utilizzando la piattaforma "ThingSpeak" fornendo le informazioni su URL/IP e Write API Key che vedremo in seguito.

Le altre funzioni come "Last data upload successful ?", "ThingSpeak connected ?", "Wifi connected ?", "Wait 5000ms" sono di supporto e permettono di verificare se la connessione Wi-Fi è andata a buon fine, se l'ultimo pacchetto dati è stato trasferito correttamente oppure aggiungere un ritardo prima di effettuare il successivo upload dati. Per garantire una corretta comunicazione o meglio per garantire la corretta scrittura dei dati su piattaforma ThingSpeak (un servizio offerto dalla MathWorks che consente di aggregare, visualizzare e analizzare flussi di dati live nel cloud prove-



Fig. 8
ThingSpeak chiave di scrittura.



nienti da dispositivi IoT) è necessario procedere con la registrazione dell'account utilizzando il seguente link <https://thingspeak.com>, procedere con la registrazione di un account e successivamente creare un canale di comunicazione come mostrato in **Fig. 7**. Dalla sezione Channel è necessario procedere con la creazione di un nuovo canale aggiun-

gendo il nome (ad esempio smart alarm) e abilitare sei degli otto canali ai quali andremo ad assegnare un'etichetta. I primi tre campi verranno utilizzati per le informazioni correnti su ore minuti secondi mentre gli ultimi tre campi forniranno le informazioni sul prossimo allarme pianificato, infine basta cliccare su tasto Save.

Dalla sezione API/Keys (vedi **Fig. 8**) è possibile risalire alle informazioni su chiavi di scrittura e lettura che potranno essere utilizzati per la condivisione dati attraverso strumenti esterni come MakeCodeBlock, copiate o scrivete il vostro Write API Key perché tornerà utile nella fase di scrittura codice per micro:bit.

Per quello che riguarda invece la libreria RTC, sono disponibili le seguenti funzioni:

- CLK P13 DIO P14 CS P15: per l'inizializzazione del modulo RTC con la definizione della mappatura pin da connettere al modulo (il riferimento è la **Tabella 1** descritta nel precedente paragrafo).
- ds set hour "var": che consente di impostare un'ora di partenza dal quale l'RTC inizierà il conteggio.
- ds get hour: consente di estrarre l'ora calcolata fino a quel momento dal modulo RTC.
- ds start RTC: per avviare il modulo.
- ds pause RTC: per interrompere il conteggio.

Viste le macro funzioni di libreria, vediamo come utilizzarle per realizzare la sveglia domotica.

Fig. 9
Inizializzazioni Variabili.

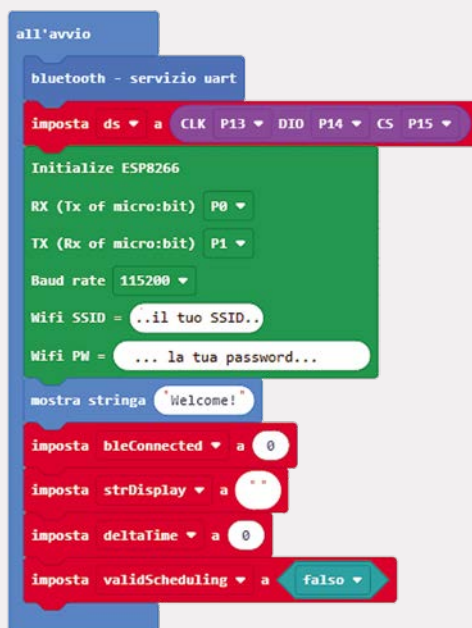


Fig. 10
BLE e modul
RTC.



IMPLEMENTAZIONE

La prima fase dell'implementazione consiste nel definire le variabili necessarie per lo sviluppo delle logiche (Fig. 9), in particolare:

- la funzione bluetooth - servizio uart per inizializzare la periferica bluetooth;
- la creazione di un'istanza "ds" per la funzione di Real Time clock;
- inizializzazione del modulo ESP8266 aggiungendo le informazioni sulla rete Wi-Fi d'interesse (come SSID e password);
- la variabile blueConnected per verificare se il dispositivo è connesso via Bluetooth;
- la variabile strDisplay per mostrare le informazioni relative su timer sulla matrice led del micro:bit;
- deltaTime che è il tempo compreso tra l'ora pianificata ed il tempo corrente rilevato dal modulo RTC espresso in secondi;
- La variabile validScheduling che consente di verificare se il dato ricevuto sull'ora di pianificazione è valido.

In linea con il diagramma di flusso definito in Fig. 7, segue l'implementazione delle logiche riguardanti la decodifica dei dati ricevuti via Bluetooth che richiede l'estrazione dei dati compresi tra il '#' e il ';' e l'estrazione delle informazioni relative a ora, minuti e secondi per quanto riguarda l'ora iniziale di trasmissione

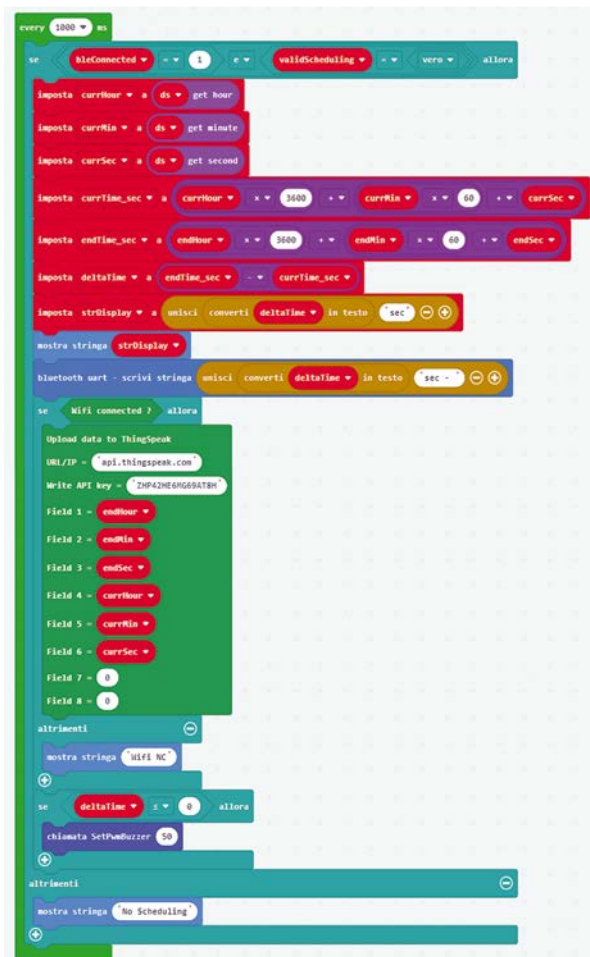


Fig. 11
Logiche.

dati e la prossima pianificazione. Il buzzer verrà disattivato impostando un duty cycle a 0%, mentre l'ora iniziale verrà utilizzata per inizializzare il conteggio del modulo RTC e se la successiva pianificazione sarà valida, il modulo RTC verrà avviato come mostrato in **Fig.10**.

Il ramo di sinistra del diagramma di flusso verifica che il task periodico di un secondo sia effettivamente

scaduto e nel caso di dispositivo connesso via Bluetooth e corretta pianificazione, tramite il flag `validScheduling`, verranno avviate le logiche per la verifica del timer. Calcolato il tempo residuo tra l'ora corrente estratta dal modulo RTC e la pianificazione in termini di secondi, viene valutata la corrispondenza delle due date e conseguente avviso utente tramite buzzer gestito attraverso il segnale PWM

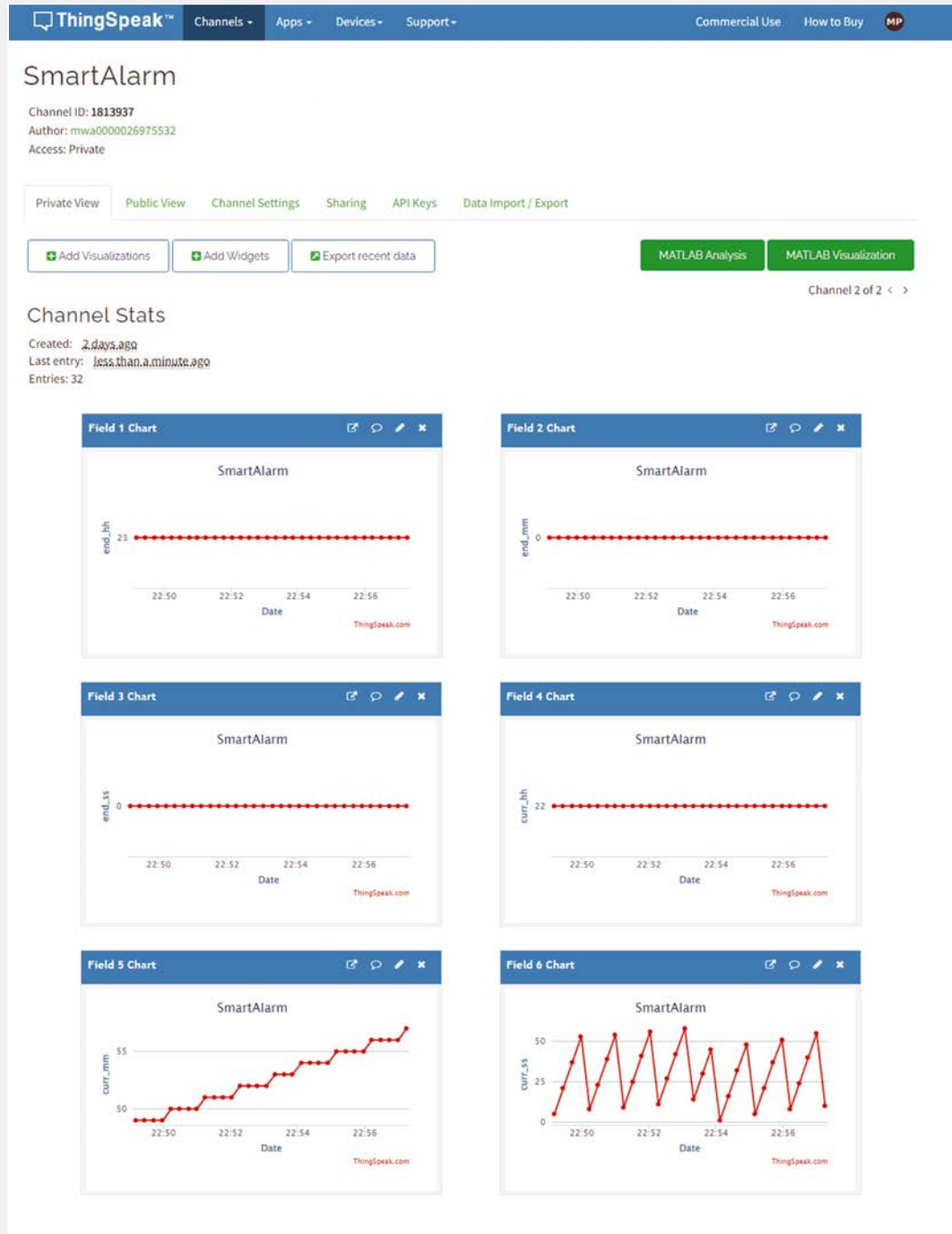
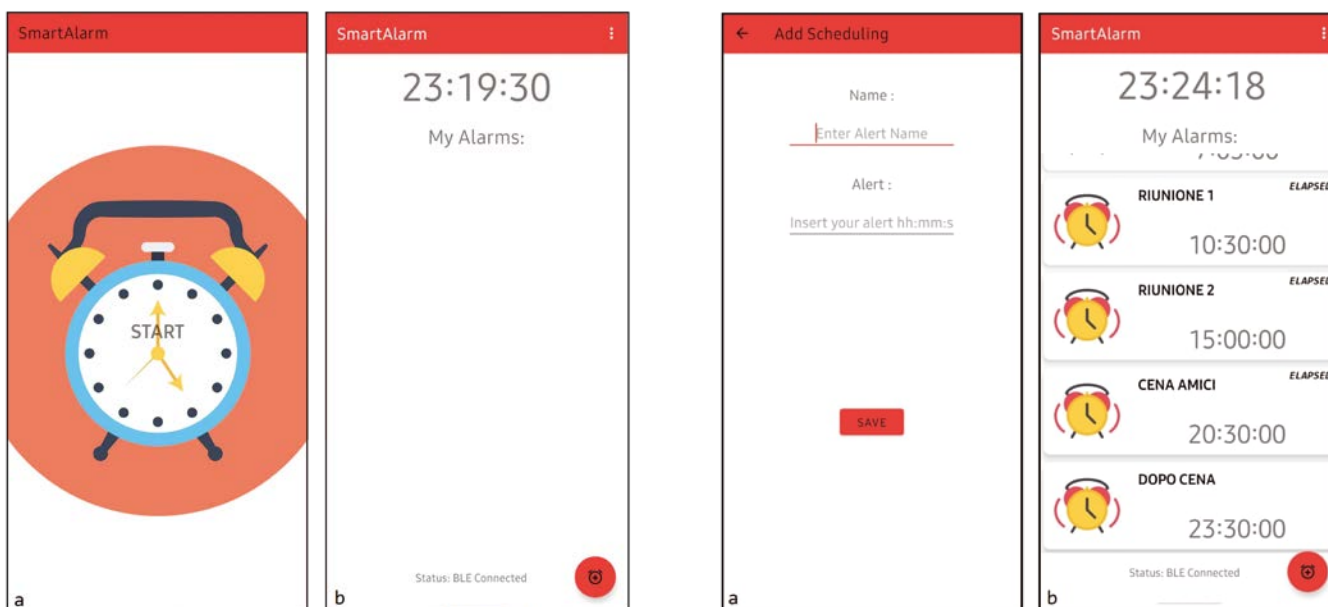


Fig. 12
Condivisione dati
ThingSpeak.



che emetterà un segnale acustico finché il micro:bit non riceverà il successivo dato BLE. Scopo di questo progetto è anche quello di spiegare come sia possibile trasmettere le informazioni via HTTP utilizzando la piattaforma ThingSpeak per pubblicare le informazioni sui due timer. Con un blocco if-else è possibile verificare che il modulo ESP8266 sia connesso alla rete domestica mentre la funzione Upload data to ThingSpeak consente di condividere le informazioni sull'ora iniziale e pianificata digitando come URL/IP l'indirizzo 'api.thingSpeak.com' e come Write API Keys il codice precedentemente salvato, il risultato è mostrato in **Fig. 12**.

APPLICAZIONE ANDROID

Passando infine all'applicazione, essa si presenta con l'immagine mostrata in **Fig. 13a**, dove attraverso un semplice tap-in (pressione a schermo) è possibile raggiungere la pagina principale **Fig. 13b**. Da quest'ultima pagina, l'applicazione ricerca il dispositivo micro:bit mostrando lo stato della connessione tramite l'etichetta "Status:" mentre attraverso il tasto di floating action l'utente ha la possibilità di aggiungere una nuova scheda di pianificazione indicando un nome e l'ora di allarme come mostrato in **Fig. 14a**. A questo punto l'applicazione invierà l'informazione sulla successiva pianificazione attraverso il formato illustrato nel paragrafo precedente ed attenderà la scadenza, aggiornando la rispettiva scheda di pianificazione attraverso l'etichetta

"ELAPSED" come mostrato in **Fig. 14b**.

Ad ogni scadenza, l'applicazione processerà la successiva pianificazione così da comunicarla via Bluetooth e ripercorrere il processo appena descritto.

CONCLUSIONI

Arriviamo al termine di questo progetto di espansione della gestione domotica con il quale abbiamo visto come si possa facilmente aggiungere la gestione di una sveglia utilizzando anche sistemi aggiuntivi e "triangolando" con servizi presenti in rete. Torniamo a ribadire che questo sistema poteva essere implementato in modo semplificato e senza l'utilizzo di dispositivi così evoluti, tuttavia è stato implementato per scopo didattico e per mostrare come tecnologie differenti possano comunicare facilmente tra loro mediante l'utilizzo del micro:bit. Alla prossima puntata! □

Fig. 13
Applicazione
Android.

Fig. 14
Schede di
pianificazione.



Cosa occorre?

La board micro:bit V2 è disponibile con lo starterkit (cod. MICROBITQSKIT) a Euro 45,00, la protoboard (cod. MICROBITPROTO) è in vendita a Euro 15,00, la breakout board con DS1302 (cod DS1302RTC) costa Euro 4,80. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

Ora i componenti non hanno più segreti!

DISPOSITIVI PROFESSIONALI PER MISURE E ANALISI PRECISE

MINI OSCILLOSCOPIO A 1 CANALE CON DISPLAY TFT LCD DA 2,4 POLLICI



COD. DSO138MINI
€ 68,00

Dispone di display da 2,4" TFT LCD - 320x240, larghezza di banda da 200 kHz e frequenza di campionamento 1 MSa/s con risoluzione 12 bit. I circuiti di alimentazione sono stati semplificati rispetto al modello precedente, così come la portabilità grazie alle dimensioni del case notevolmente ridotte. L'ingresso di trigger esterno accetta segnali TTL e LVTTTL come sorgente e permette infine di esportare (in formato testo) la forma d'onda in seriale. Alimentazione attraverso connettore micro USB. Questo oscilloscopio è particolarmente indicato per gli hobbisti interessati a misurare e visualizzare la maggior parte dei segnali elettronici.

TESTER PER COMPONENTI ELETTRONICI



COD. JT-LCRT7
€ 34,00

Dotato di display LCD da 3,5" a colori, ricevitore a infrarossi e batteria ricaricabile integrata da 3,7 volt 350 mAh. Permette di misurare diodi, diodi zener, resistori, condensatori, l'ESR (Resistenza Equivalente Serie) di un condensatore, induttori, tiristori, triac, FET, transistor bipolari (indicando se è NPN o PNP) e batterie (max. 4,5 V). Funziona anche con condensatori elettrolitici polarizzati e non polarizzati in entrambi gli orientamenti. Grazie al ricevitore a infrarossi integrato possono essere decodificati anche i segnali a infrarossi e visualizzati sul display, in forma d'onda e codice esadecimale. Le misurazioni vengono avviate automaticamente premendo semplicemente un pulsante.

Prezzi IVA inclusa.

Analizzatore di componenti DCA55



€ 73,00
cod. DCA55

Consente di analizzare una vasta gamma di componenti attivi, identificando la pinout e fornendo informazioni relative ai principali parametri.

Analizzatore di semiconduttori DCA75



€ 148,00
cod. DCA75

Analizza svariati tipi di semiconduttori. Può essere interfacciato al PC e, grazie al software in dotazione, fornisce un'analisi dettagliata del componente.

Misuratore di induttanza, capacità, resistenza e impedenza LCR45



€ 129,00
cod. LCR45

Strumento multifunzione. Misura induttanza, resistenza e impedenza arbitraria. Riconosce automaticamente il tipo di componente, e fornisce informazioni relative ai principali parametri.

Blebricks

A PORTATA DI APP

di **PAOLO TEALDI**

Interagiamo tramite smartphone con i mattoncini per la prototipazione istantanea della serie Blebricks. Seconda e ultima puntata.

N

ella prima puntata di questo articolo vi abbiamo presentato il sistema Blebricks della Bleb Technology, basato su dispositivi hardware che permettono la prototipazione istantanea, ovvero la creazione immediata di dispositivi basati su sensori/attuatori, in ambito IoT (Internet of Things).

I Blebricks sono mattoncini intelligenti realizzati con una tecnologia brevettata italiana che permette di costruire sistemi semplicemente assemblando ad incastro i moduli richiesti, senza bisogno di cavi o saldature. I Blebricks sono una soluzione rapida, efficace, economica, di facile implementazione (in termini hardware e software), impiego e utilizzo, per progetti in cui sia necessaria l'interazione fra una serie di oggetti fisici e lo smartphone degli utenti; una soluzione rivolta sia a maker e studenti, sia a professionisti del settore.

Fig. 1
La schermata principale della app MakeApp: in questo caso è riferita al modulo BLE-B.



PIÙ FACILE, CON L'APP

Bleb Technology ha pensato a come semplificare ulteriormente il lavoro di prototipazione, rendendo disponibile un'App specifica per interagire con i progetti basati su Blebricks.

La App si chiama MakeApp ed è disponibile gratuitamente sul Play Store per smartphone e tablet con sistema operativo Android; una volta installata, ci permette di interagire con i mattoncini, ovvero con il nostro prototipo appena assemblato.

Una volta scaricata, installata e lanciata dal nostro dispositivo mobile (ricordatevi prima di attivare su di esso il Bluetooth e il Sensore di Posizione), potete visualizzare subito i dati dei sensori, interagire con i Blebricks e configurare il loro funzionamento con semplice logica IFTTT (IFTTT è un acronimo per If This Then That, ovvero se questo accade allora fai accadere quello).

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

Accertiamoci di aver installata sempre l'ultima versione della MakeApp (controlliamo eventualmente nel Play Store la presenza di eventuali aggiornamenti da installare) nonché l'ultima versione del firmware dei Blebricks (firmware e tutorial della procedura di aggiornamento OTA sono disponibili nella sezione Download del sito web della Bleb Technology – www.bleb.it). Questo per evitare problemi di funzionamento non corretto della

Fig. 2
I settaggi del BLE-B all'interno della MakeApp.



MakeApp oppure riconoscimento parziale o errato dei Blebricks).

Una volta collegato il BLE-B al modulo di alimentazione (ad esempio Blebrick RPS o Blebrick SPS) e aperta la MakeApp, sulla prima pagina troveremo le informazioni relative al nostro dispositivo: nome, distanza stimata, RSSI, livello della batteria e indirizzo Bluetooth (**Fig. 1**). Ricordiamo che la configurazione minima di un qualsiasi sistema Blebricks prevede un BLE-B e un modulo di alimentazione. Il Blebrick RPS è un modulo di alimentazione che include un caricatore per batterie Li-Po / Li-Ion micro-USB (Type B). Per ottenere funzionalità aggiuntive si possono collegare ad incastro altri Blebricks contenenti altri sensori o attuatori. Se clicchiamo sull'icona del BLE-B possiamo vedere le impostazioni e cambiare ad esempio il nome o la potenza di trasmissione, ma anche gestire impostazioni più avanzate come gli intervalli di advertising e scan BLE. Viene inoltre mostrata nella parte inferiore della schermata la versione del firmware installato (**Fig. 2**) nel modulo.

Un pratico pulsante di reset (RESET BLE-B) permette altresì di riportare il mattoncino BLE-B e i Blebricks ad esso eventualmente collegati, alle condizioni iniziali e alle impostazioni predefinite (factory default). Inoltre, potete gestire la **Demo-Mode**, rammentando che se intendete limitare i consumi energetici la dovete disattivare. Se volete, potete modificare il nome del dispositivo proposto per impostazione predefinita: vi basta toccare accanto a Device Name e scrivere il nome desiderato. Potete anche aumentare o diminuire la potenza di trasmissione operando dalla casella accanto a TX power.

Ritornando indietro alla schermata principale, invece, vediamo che in essa sono presenti ulteriori informazioni (**Fig. 3**) come la temperatura rilevata dal sensore del BLE-B, il feedback del pulsante fisico che si trova sul medesimo (ad esempio provando a premerlo, nella MakeApp verrà indicato tale stato, mentre nel momento in cui lo ripremeremo oppure lo premeremo a lungo apparirà la nuova condizione) nonché il messaggio di testo memorizzato (cliccando sulla relativa icona è possibile modificarlo o inserirne uno nuovo).

Cliccando invece sull'icona del LED si aprirà la pagina delle relative impostazioni; in essa, con gli slider è possibile modificare l'intervallo di lampeggio del LED (slider blinking) e l'intensità di ogni colore (cursori Red, Green, Blue).

Da notare che in alto i tre numeri separati da una virgola indicano il colore creato scorrendo gli slider



secondo lo standard RGB. Premendo sul pulsante "set blinking" attribuiamo le impostazioni scelte al LED lampeggiante, mentre cliccando "set steady color" attribuiamo le impostazioni al LED fisso. È sempre possibile tornare alle impostazioni iniziali per uno o entrambi i pulsanti (**Fig. 4**).

Il modulo BLE-B dispone di quattro GPIO completamente configurabili. Nella schermata principale della MakeApp è mostrato il feedback di ogni canale. Inoltre, anche qui, cliccando sulla relativa icona, è possibile accedere alle relative impostazioni (**Fig. 5**). Nella schermata delle impostazioni dei GPIO è presente una foto con chiare indicazioni che vi aiutano a individuare ciascuno degli I/O in modo da poterli configurare facilmente e correttamente (**Fig. 5**). Configurando un ingresso digitale, ne viene visualizzato lo stato, mentre impostando il duty-cycle delle uscite, si aggiusta il valore della tensione. Gli ultimi due GPIO possono inoltre essere configurati come ingressi analogici.

IL BLE-B NEL DETTAGLIO

Prima di proporvi un paio di applicazioni dimostrative che si basano sulla MakeApp, approfondiamo la conoscenza del BLE-B, perché è il mattoncino di base, il cuore del sistema ed è delegato alla gestione operativa del sistema e delle comunicazioni tramite Bluetooth Low Energy BLE 4.2 con distanza di comunicazione in aria fino a 20-30 m. Oltre alla gestione del sistema e dei vari mattoncini Blebricks, il BLE-B presenta una serie di funzio-

Fig. 3
Dalla schermata principale della MakeApp è possibile ottenere le principali informazioni sul BLE-B e accedere alle configurazioni.

Fig. 4
I settaggi del
led del BLE-B
all'interno della
MakeApp.



nalità aggiuntive di seguito descritte.

- Datalogger: attivandola vengono salvati fino a 600 Datapoint in memoria interna. Se è presente un Brick di memoria (es MEM) tale memoria può essere aumentata a 256.000 Datapoint.
- Modalità low power, utile per risparmiare energia in caso si utilizzino sensori a lunga latenza (ad esempio decine di minuti).
- Localizzazione indoor e RTLS (Real Time Locating System), sia tramite i gateway dedicati che tramite tecnologia proprietaria QUUPPA. Programmazione semplice tramite metodologia IF This Than do That diretta.

- Aggiornamento del firmware OTA (over the air).
- Possibilità di interfacciamento via WiFi con l'assistente vocale Alexa e/o IFTTT per la gestione e comunicazione di dispositivi prodotti da terze parti compatibili con tali tecnologie.

Le principali caratteristiche tecniche del mattoncino BLE-B sono:

- interfaccia Bluetooth Low Energy (v4.2);
- transceiver a 2,4 Ghz;
- sensibilità in ricezione di -96 dBm;
- potenza di trasmissione impostabile da -20 to +4 dBm in sted 4 dB;

Fig. 5
I settaggi dei
GPIO del BLE-B
all'interno della
MakeApp.



- RSSI (risoluzione di 1 dB);
- interfacce I²C, UART e SPI;
- Data rate 1 Mbps o 2 Mbps;
- emulazione NFC Tag;
- pulsante (short e long press);
- LED RGB con colori e luminosità impostabili da utente;
- 4 GPIO configurabili come ingressi o uscite digitali, ingressi analogici, contatori di impulsi e contagiri;
- sensore di temperatura incorporato;
- sensore tensione batteria (livello di carica) integrato;
- tensione di alimentazione: 3V +/- 10%
- consumo medio di corrente senza carico esterno @25°C, adv rate: 1m5 sec, Potenza: 0 dbm: 6uA;
- temperatura operativa: - 40 + 85°C.

Il BLE-B può funzionare in tre modalità:

- “advertising” è la modalità operativa predefinita con consumo energetico minimo; in essa i principali dati “advertising” sono codificati e trasmessi utilizzando un ID produttore registrato Bluetooth (0x0668) e riservato a Bleb Technology;
- “observing” che consiste nella scansione di pacchetti advertising emessi da altri dispositivi BLE;
- “connection” per trasferire i dati da/verso i dispositivi BLE central.

Il mattoncino BLE-B può essere configurato (anche attraverso la MakeApp), in modalità di programmazione, nella quale i comandi vengono ricevuti in modalità “connection”, per essere poi eseguiti e salvati nella memoria Flash del BLE-B per la successiva esecuzione.

Concludiamo questa parentesi sul modulo BLE-B dicendo che il firmware del BLE-B può essere aggiornato in modalità wireless (OTA-DFU: Over The Air Device Firmware Update).

LE APPLICAZIONI

Il mattoncino BLE-B (codice BLE-B su www.futurashop.it) può essere utilizzato con tutti i dispositivi, sensori, attuatori, alimentatori e dispositivi di comunicazione aggiuntivi della famiglia Blebrick, ma anche da solo è in grado di implementare funzionalità essenziali.

Pertanto, si presta a molteplici impieghi che richiedono il monitoraggio dei più vari parametri ambientali, sollecitazioni meccaniche e la trasmissione locale o remota di dati, nonché l’attivazione di attuatori al verificarsi di particolari eventi.

ESEMPIO:

CAMPANELLO WIRELESS

Con Blebricks, ed in particolare con il modulo BLE-B, possiamo ad esempio realizzare un semplice campanello senza fili.

Ponendo ad esempio il BLE-B all’esterno dell’abitazione (nel caso di campanello guasto o non presente), quando viene premuto il pulsante presente sul BLE-B, il nostro smartphone suona avvisandoci che c’è qualcuno alla porta che ci cerca (**Fig. 6**).

I moduli Blebricks necessari a realizzare questa semplice ed utile applicazione dimostrativa sono BLE-B e RPS (o altro modulo di alimentazione della serie Blebricks).

ESEMPIO:

ANTIFURTO PER BICI E MOTORINO

Oppure possiamo realizzare un antifurto che faccia partire una suoneria sullo smartphone su cui è installata la MakeApp quando la nostra bicicletta o il nostro motorino vengono mossi oppure semplicemente urtati.

In questo caso è necessario utilizzare un sensore di vibrazione (accelerometro) e impostare i relativi parametri facendo in modo che se l’accelerazio-

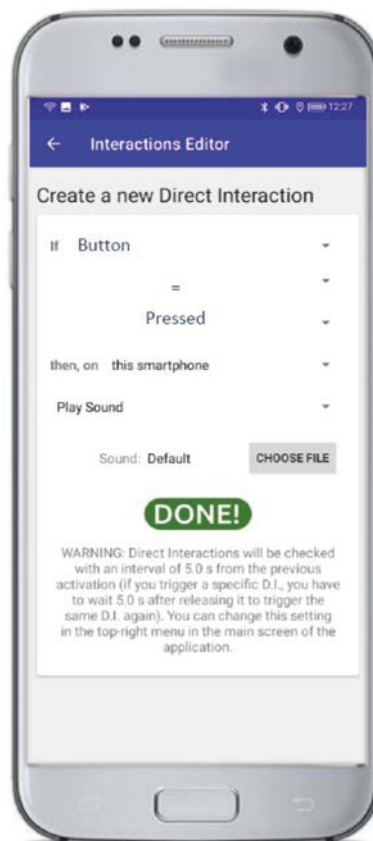
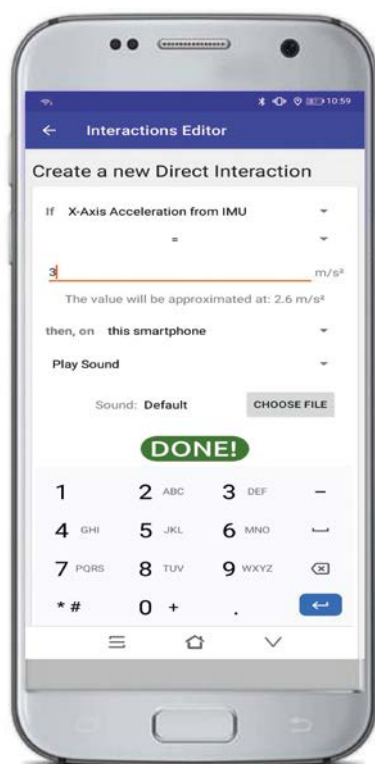


Fig. 6
Applicazione
con il
campanello
wireless.

→ **Fig. 7**
Impostazione
del valore di
accelerazione per la
soglia di allarme.



ne sull'asse X è maggiore o uguale a 3 m/s^2 lo smartphone emetta un suono (**Fig. 7**).
I moduli Blebricks necessari a realizzare questa applicazione sono un BLE-B (codice BLE-B) ed un RPS (codice BLEBRPS) oltre al modulo IMU (codice BLEBACC), che in questo caso ci serve per rilevare il movimento della bici o della moto e indicarcene la direzione.
Notate che per l'accelerazione è disponibile soltanto un numero limitato di valori impostabili, pertanto l'app approssima e mostra qual è il primo

valore più vicino disponibile in base al numero che le abbiamo fornito come soglia di accelerazione desiderata.

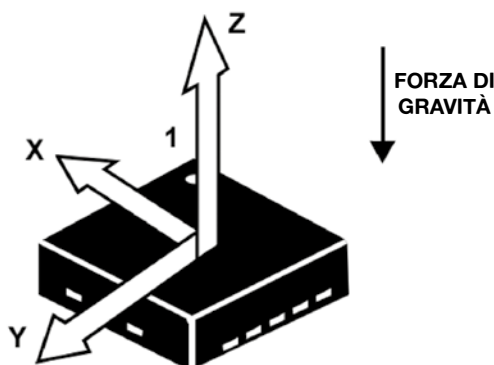
Notate che in realtà il modulo IMU (**Fig. 8**) contiene un accelerometro triassiale, quindi misura l'accelerazione lungo le tre direzioni dello spazio; notate altresì che l'accelerazione misurata a riposo su uno dei tre assi avrà sempre un valore di circa $\pm 10 \text{ m/s}^2$, perché significa che il mattoncino IMU sta misurando l'accelerazione dovuta alla forza di gravità terrestre cui tutti noi siamo soggetti; quella è sempre presente e se il modulo è posto orizzontalmente risulta sull'asse Z.

Abbiamo inserito soltanto il valore di un asse (X), ma è possibile rendere più sensibile il sensore (e quindi più accurato) inserendo tutti e tre i valori nella interaction (X, Y, Z).

CONCLUSIONI

Con questo articolo abbiamo voluto continuare il viaggio, iniziato nel precedente numero, per rendervi partecipi della nostra personale esperienza nel mondo della prototipazione istantanea grazie alla tecnologia italiana Blebricks, mattoncini intelligenti IoT prodotti dalla Bleb Technology.
Ci siamo addentrati nel funzionamento della MakeApp, cuore e motore di interazione con i prototipi realizzati.
Vi abbiamo poi mostrato due semplici esempi su come creare un campanello wireless e un antifurto, entrambi con notifica su smartphone.
Vi sarete accorti di come sia semplice ed intuitivo far prendere forma in pochi attimi alle vostre idee. Questo, naturalmente, è solo un piccolo assaggio: quanto basta a stuzzicare la vostra curiosità e fantasia creativa, alle quali darete libero sfogo una volta che avrete preso dimestichezza con l'ambiente Blebricks. □

→ **Fig. 8**
I tre assi su cui
l'accelerometro
rileva
l'accelerazione.



Cosa occorre?

I moduli Blebricks sono disponibili in comodi starter-kit o anche singolarmente. In particolare lo starter-kit base (cod. BLEBSTARTERKIT) è in vendita a Euro 41,50. Tutti i prodotti Blebricks sono disponibili alla pagina <https://bit.ly/blebricks>. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

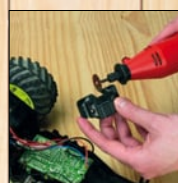
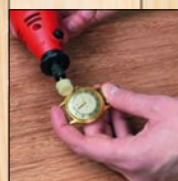
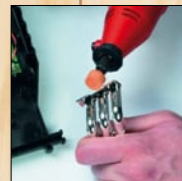
MINI TRAPANI per ogni esigenza

Mini Trapano con set per incisione - 162 Pezzi

cod. **VTHD0051**
€ 52,⁰⁰



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**
€ 32,⁰⁰

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



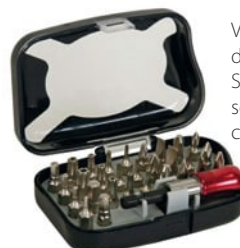
Pratica confezione comprendente 188 accessori per incidere, forare e smerigliare. Da utilizzare con i trapani della serie VTHD.

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



Libreria Self-Test Certificata SIL 2/3 per Sicurezza Industriale

Semplifica lo sviluppo e la certificazione del tuo sistema

La Sicurezza Industriale è una questione fondamentale nel campo dei controlli industriali, robot, sensori, rilevatori di gas e rilevatori di fumo, rendendo l'Industrial Safety Standard IEC 61508 un prerequisito fondamentale per queste applicazioni.



Il nostro ampio portfolio di microcontroller PIC32 e SAM 32 bit (MCU) e Digital Signal Controllers (DSC) dsPIC33C offre librerie diagnostiche certificate TÜV Rheinland (librerie di self-test) per progetti mirati a SIL 3, IEC 61508 FMEDA e/o manuali di sicurezza, disponibili in un package completo.

Utilizza la completa documentazione e le librerie software certificate per semplificare e accelerare lo sviluppo del tuo sistema, risparmiando tempo e costi di certificazione.

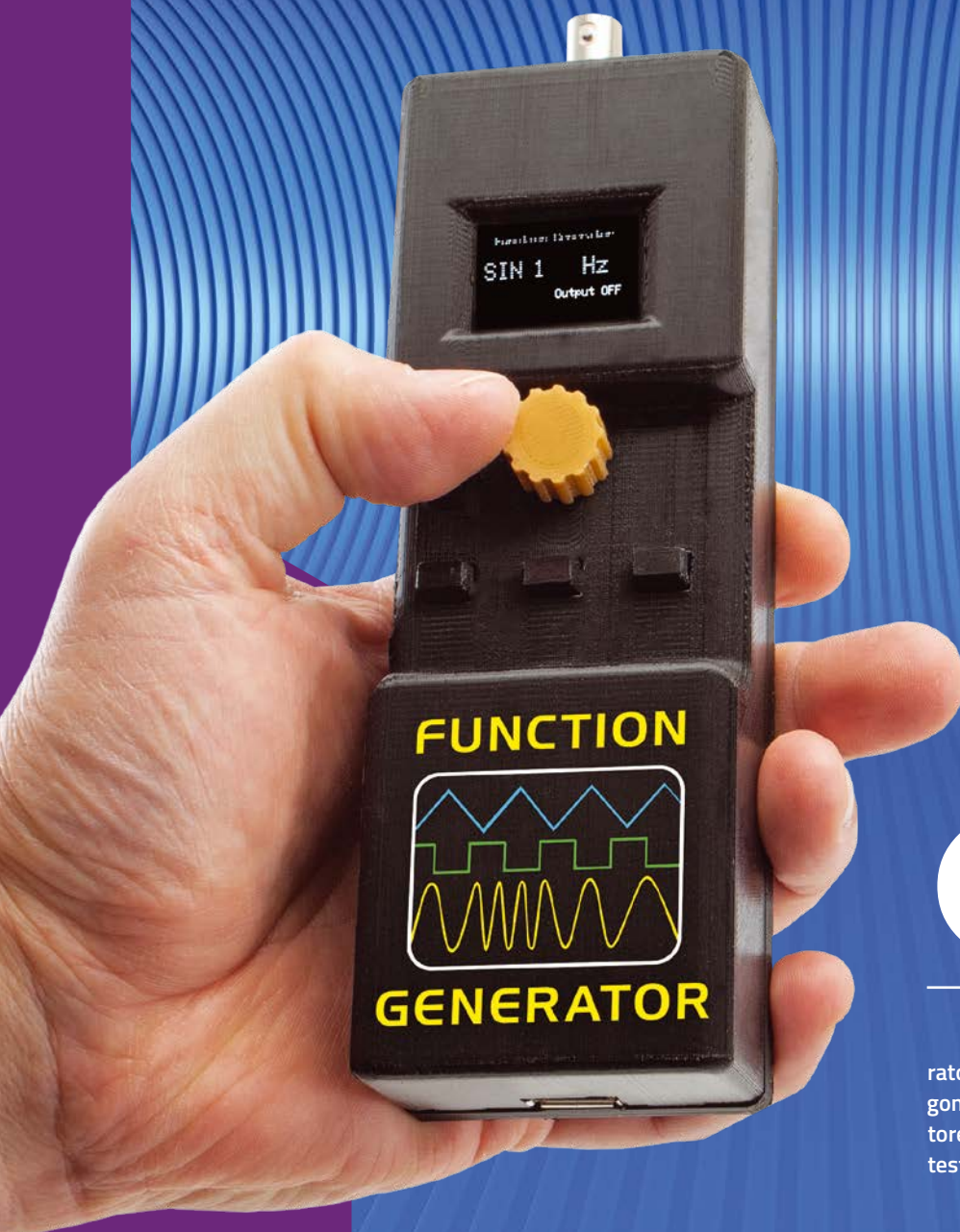
Vantaggi delle Librerie Diagnostiche IEC 61508 di Microchip:

- Le librerie diagnostiche certificate TÜV Rheinland possono essere utilizzate per implementare un livello di sicurezza SIL 2 nelle applicazioni a canale singolo e un livello di sicurezza SIL 3 nelle applicazioni a doppio canale
- Rileva guasti hardware random nel core, nella memoria Flash, nella SRAM e in altre periferiche
- Le librerie diagnostiche SIL 2/3 fanno parte di un pacchetto di sicurezza generale che include un manuale di sicurezza software e una lista di controllo di sicurezza per i progetti di sicurezza industriale IEC 61508
- Codice sorgente completo per MCU PIC® e AVR® e dsPIC33C DSC, e codici binari per MCU PIC32C e SAM a 32-bit



GENERATORE DI FUNZIONI CON AD9833

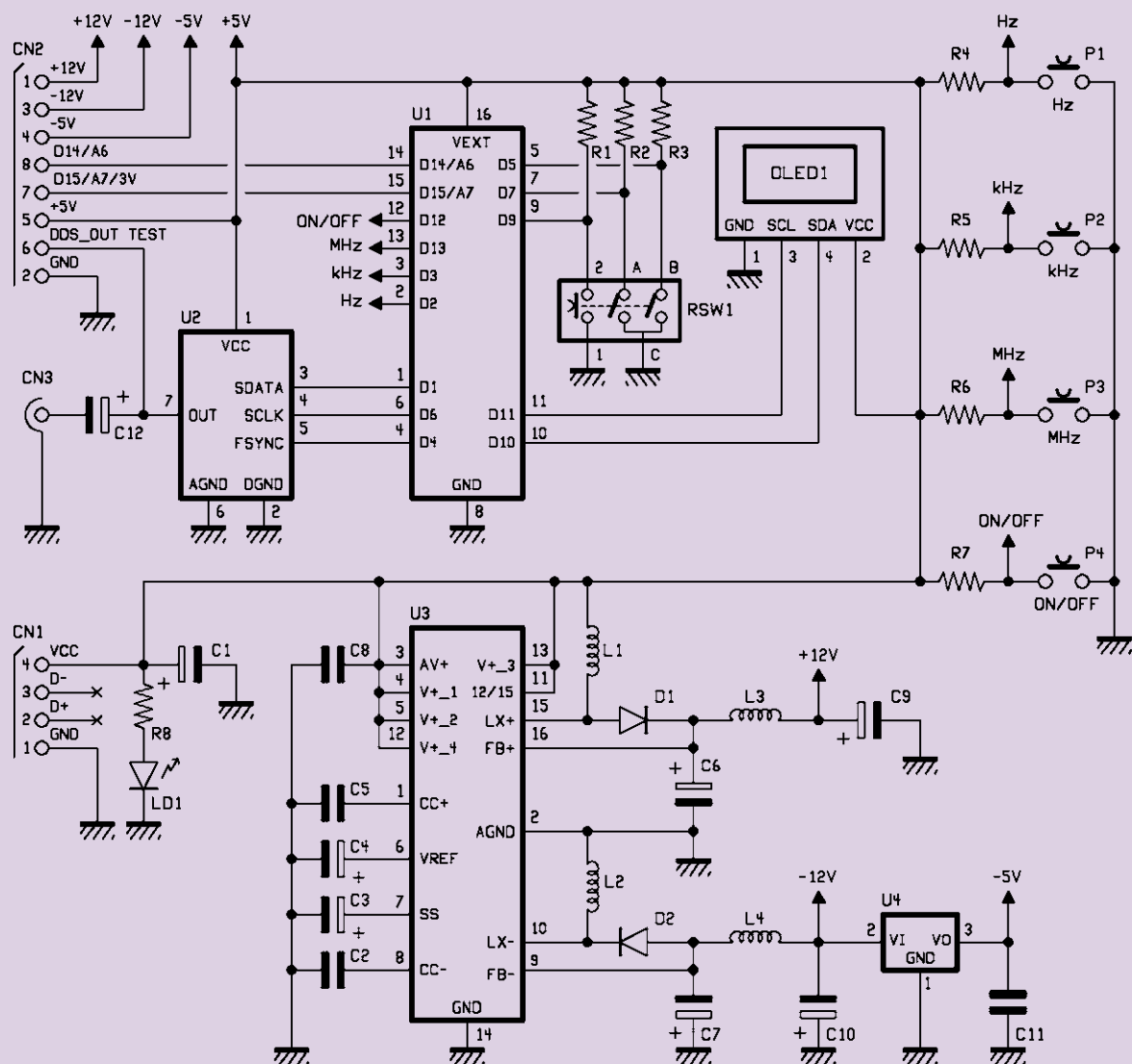
di FULVIO DE SANTIS



Svilupa forme
d'onda sinusoidale,
triangolare e quadra a
frequenza regolabile
fra 1 Hz e 12 MHz.

Q

uella della sintesi digitale diretta (DDS) è una tecnica ormai collaudata per produrre segnali, tant'è che la abbiamo già utilizzata in passato in diversi progetti per implementare un generatore di segnali. Vogliamo ora tornare sull'argomento proponendo il progetto di un generatore di funzioni, da utilizzare in laboratorio per testare i circuiti elettronici, capace di generare



forme d'onda sinusoidali, triangolari e onda quadra, selezionabili come spiegheremo in questo articolo. La scheda che vi presentiamo in questo articolo (a cui abbiamo dato il codice *GenDDS_AD9833*) permette di visualizzare la frequenza delle forme d'onda su un display OLED. Ma a completare il nostro generatore di funzioni vi è un'ulteriore board (dal nome *GenerDDS_Amplif&Filter*) utilizzabile come filtro passa-basso e/o per amplificare, con regolazione del livello di uscita, i segnali nella banda del filtro. Quest'ultima però verrà descritta nel prossimo numero della rivista, ma ve la anticipiamo nello schema a blocchi funzionale del generatore di funzioni visibile nella **Fig. 1**.

LA SCHEDA DEL GENERATORE

Per ora ci occupiamo del generatore puro e semplice, che è costituito da tre sezioni:

- generazione del segnale a sintesi digitale diretta di frequenza (DDS);
- sezione di controllo;
- stadio di alimentazione.

La sezione di generazione del segnale DDS viene realizzata mediante il circuito integrato AD9833 dell'Analog Devices, montato a bordo di un modulo breakout acquistabile, ad esempio, da Futura Elettronica con il codice *AD9833BREAKOUT* su www.futurashop.it (**Fig. 2**).

La sezione di controllo è costituita dal micro-

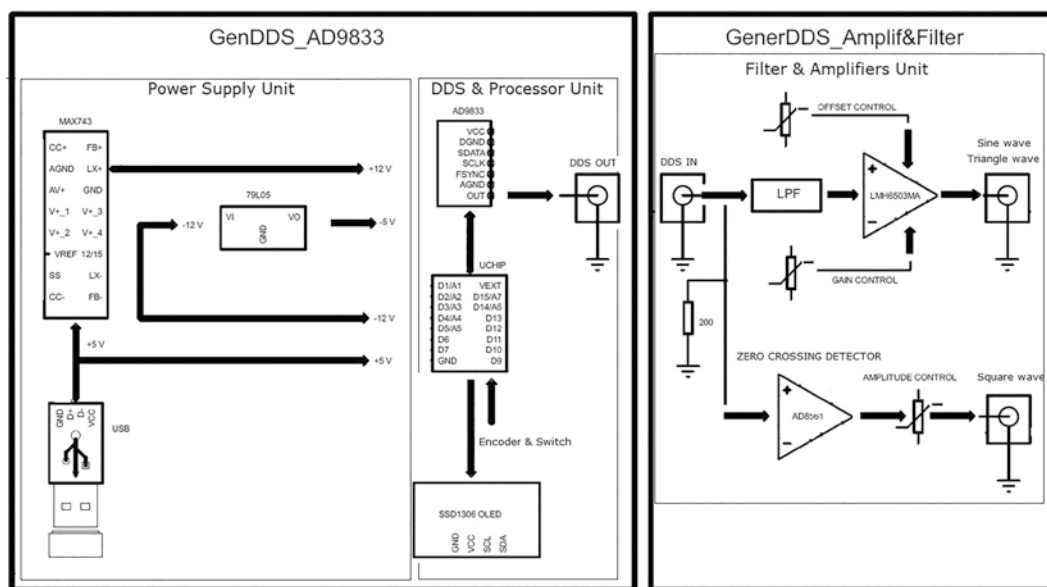


Fig. 1
Schema a blocchi funzionale del generatore di funzioni con le due schede: quella qui descritta è la parte a sinistra.

controllore uChip di Itaca Innovation, anch'esso acquistabile con il codice *UCHIP* (www.futurashop.it), programmato con il firmware che scriveremo e caricheremo utilizzando l'IDE di Arduino. Completano il circuito dei pulsanti di selezione della gamma di frequenze, un Encoder con il quale viene effettuata la regolazione della frequenza e la selezione della forma d'onda, e infine un display OLED SSD1306 per la visualizzazione dello stato ON/OFF del generatore, del valore di frequenza e del tipo di forma d'onda selezionato.

La sezione di alimentazione è costituita dal regolatore di tensione switching MAX743 della MAXIM, dal regolatore di tensione lineare 79L05 e da un connettore USB. La sezione di alimentazione alimenta anche la connessione USB.

Il tutto è meglio descritto dallo schema elettrico che vedete nella pagina accanto e che a breve descriveremo dettagliatamente.

Il circuito rende disponibili le forme d'onda attraverso un connettore BNC, ossia segnali ad onda sinusoidale, triangolare e quadra alla frequenza da 1 Hz fino ad un massimo di 12 MHz.

SCHEMA ELETTRICO

La scheda uChip (siglata U1) è il gestore del generatore di funzioni, controlla il modulo AD9833 (U2) mediante comunicazione seriale SPI, impostandone le funzioni DDS (che vedremo in dettaglio nella trattazione dell'AD9833) per il valore di frequenza e la forma d'onda impostati dall'encoder RSW1 e dai pulsanti P1, P2 e P3 di selezione della gamma di frequenze, tutti controllati mediante le porte

digitali della uChip. I segnali di uscita sinusoidale e triangolare dell'AD9833 sul connettore BNC CN3 hanno un'ampiezza di circa 0,6 Vpp da 1 Hz fino alla frequenza di 1 MHz, dopodiché si riduce con l'aumentare della frequenza fino a 0,3 Vpp a 10 MHz, anche per la presenza del condensatore C1 da 22 pF in parallelo all'uscita dell'AD9833. Il segnale di uscita ad onda quadra ha un'ampiezza di 5 Vpp per tutta la banda di frequenze. Il pulsante P4 ON/OFF attiva o disattiva l'uscita DDS. I dati relativi al valore di frequenza e della forma d'onda selezionata vengono inviati da uChip al display OLED SSD1306 mediante comunicazione I²C.

Il circuito viene alimentato a +5V collegando al connettore USB CN1, o al connettore microUSB dell'uChip, l'uscita USB di un computer o di un power bank. Il connettore USB CN1 sembrerebbe

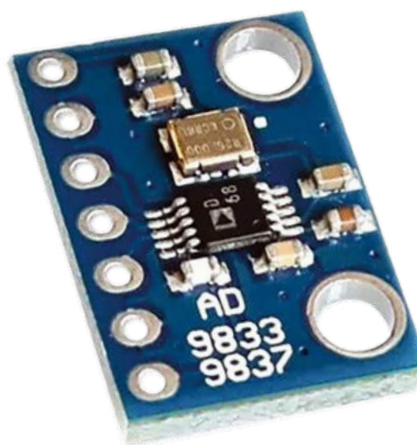


Fig. 2
La breakout board con l'AD9833.

disalimentazione selettiva di sezioni non utilizzate del chip, minimizzando così il consumo di corrente. Con apposite impostazioni di programmazione del chip, l'AD9833 è in grado di generare segnali DDS la cui frequenza e fase del segnale di uscita può essere impostata via software in tempo reale. La comunicazione con l'AD9833 può essere effettuata mediante un'interfaccia seriale SPI a 3 fili con frequenze di clock fino a 40 MHz, semplificando notevolmente l'invio dei comandi di programmazione da parte di un microcontrollore esterno. Oltre alla generazione di segnali, l'AD9833 è in grado di fornire un'ampia gamma di schemi di modulazione anche complessi implementati nel dominio digitale, consentendo la realizzazione di algoritmi di modulazione utilizzando tecniche di processo digitali (DSP).

Seguendo lo schema a blocchi funzionale di

Fig. 3, possiamo rilevare che il circuito interno dell'AD9833 è costituito dai seguenti elementi:

- un oscillatore a controllo numerico (NCO): consiste di due registri di selezione della frequenza (FREQ0 REG, FREQ1 REG), un accumulatore di fase a 28 bit (PHASE ACCUMULATOR), due registri di offset di fase (PHASE0 REG, PHASE1 REG), un sommatore dei registri di selezione della frequenza e dell'offset di fase, modulatori di frequenza e fase;
- una SIN ROM: è una memoria ROM che contiene una tabella di valori della fase che vengono convertiti in ampiezza della sinusoide da generare;
- un convertitore digitale-analogico DAC con risoluzione a 10 bit. Il DAC riceve i valori digitali dell'ampiezza dalla SIN ROM e li converte nelle corrispondenti tensioni analogiche. Il DAC genera una tensione di uscita tipicamente di 0,6 V pp. Non necessita di carico esterno in quanto un resistore da 200 Ω è collegato internamente fra l'uscita del DAC e massa;
- un regolatore di tensione. Una tensione di alimentazione da 2,3V a 5,5V applicata fra il pin VDD e massa alimenta la sezione digitale (DVDD) e la sezione analogica (AVDD) dell'AD9833. La sezione digitale (DVDD) è alimentata a 2,5 V mediante un regolatore di tensione interno; se la tensione applicata al pin VDD è inferiore o uguale a 2,7 V, i pin CAP/2,5 V e VDD devono essere collegati insieme bypassando così il regolatore;
- una SERIAL INTERFACE AND CONTROL LOGIC. L'AD9833 ha un'interfaccia seriale standard a 3 fili compatibile con le interfacce standard SPI,

QSPI, MICROWIRE e DSP.

I dati a 16 bit vengono caricati nei registri dell'AD9833 sotto il controllo del clock seriale applicato all'ingresso SCLK. L'ingresso FSYNC funge da frame di sincronizzazione e abilitazione del chip. I dati possono essere trasferiti solo quando FSYNC è basso rispettando il minimo tempo di setup del fronte di discesa dell'impulso di clock SCLK. Dopo che FSYNC è settato a livello basso, i dati vengono trasferiti in corrispondenza dei fronti di discesa di SCLK per 16 impulsi di clock.

Esponiamo ora il principio di funzionamento del generatore premettendo che, in generale, le onde sinusoidali sono espresse matematicamente dalla formula:

$$a(t) = \sin(\omega t)$$

Queste forme d'onda sono lineari e non facili da generare, se non attraverso la costruzione a tratti. D'altra parte, l'informazione angolare è di natura lineare, ovvero, l'angolo di fase ruota di un angolo fisso per unità di tempo. La velocità angolare ω dipende dalla frequenza del segnale (del segnale di uscita dal DAC) come definito dalla formula $\omega = 2\pi f$. Stabilito che la fase di un'onda sinusoidale è lineare e dato un intervallo temporale di riferimento (periodo del clock = Δt), può essere determinata l'entità di rotazione della fase per questo periodo:

$$\Delta \text{Phase} = \omega \Delta t$$

Risolvendo rispetto a ω :

$$\omega = \Delta \text{Phase} / \Delta t = 2\pi f$$

Risolvendo poi rispetto alla frequenza f e sostituendo la frequenza di clock (f_{MCLK}) di riferimento per il periodo di riferimento $1/f_{\text{MCLK}} = \Delta t$:

$$f = \Delta \text{Phase} \times f_{\text{MCLK}} / 2\pi$$

L'AD9833 costruisce il segnale di uscita in base a questa semplice equazione. Noto il valore digitale ΔPhase , ad esso viene associato il rispettivo valore di ampiezza rilevato dalla tabella SIN ROM, per ogni step di ΔPhase fino alla completa costruzione della sinusoide. Un semplice chip DDS può implementare questa equazione con tre sottocircuiti principali: oscillatore a controllo numerico (NCO) e modulatore di fase, tabella SIN ROM e convertitore digitale-analogico (DAC).



Fig. 5
La scheda di
sviluppo uChip.



IL MICROCONTROLLORE UCHIP

uChip è una piccola scheda di sviluppo open source a basso costo compatibile con Arduino. Il suo layout è compatibile con la disposizione dei pin di un circuito integrato a 16 pin a profilo stretto. In **Fig. 5** è riportato il layout della scheda di sviluppo uChip. La scheda uChip integra il chip microcontrollore Cortex M0+ ATSAMD21E18 della Microchip; implementa la maggior parte delle funzioni di Arduino Zero e dispositivi basati su SAMD21E18, incluso un host USB OTG completamente funzionale. Come Arduino Zero, ha 256 kB di flash e 32 kB di RAM. UChip include anche convertitori DC-DC buck e boost integrati, che consentono di generare una tensione di 5V-500mA verso l'USB Host, quando la tensione esterna è compresa tra 3,3V e 5V. Inoltre può fornire esternamente una tensione di 3,3 V-100mA o la tensione USB con una corrente fino a 1 A quando uChip è alimentato tramite USB. Un circuito di supporto impedisce anche la corrente inversa tra la tensione esterna e l'USB quando

uChip non funziona come host USB. Inoltre, la circuiteria di supporto implementa anche la selezione automatica dell'alimentazione tra l'USB e la tensione esterna. Quando uChip è alimentato tramite USB, la tensione di uscita (VEXT) applicata al pin 16 può essere selezionata via software tra 3,3V e la tensione USB (nominale 5V).

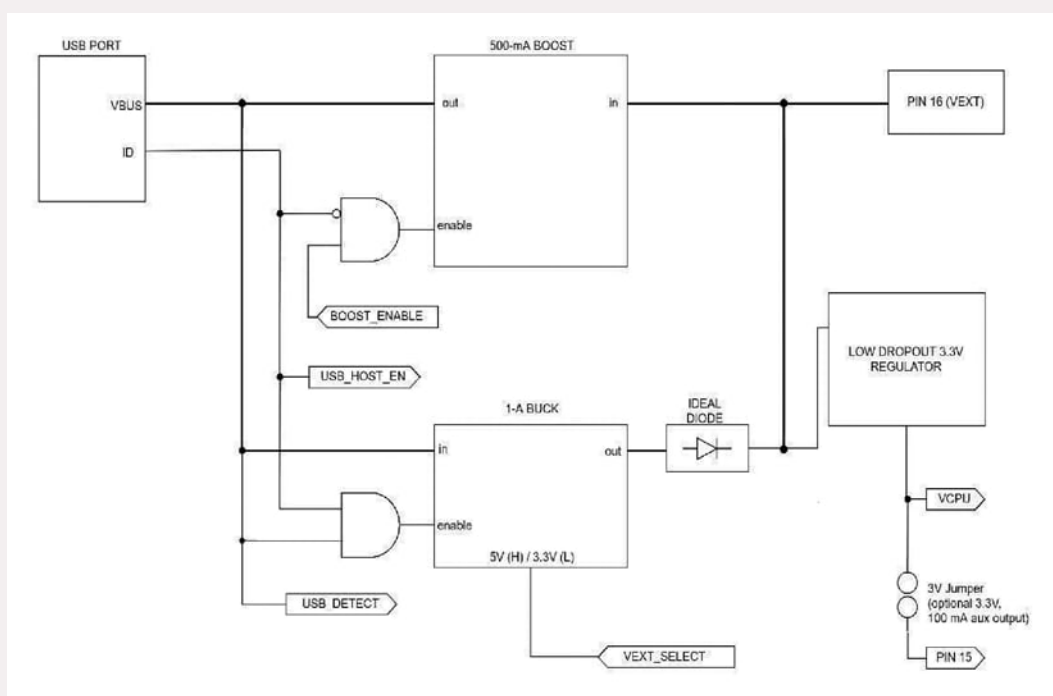
IL SISTEMA DI GESTIONE DELL'ALIMENTAZIONE DI UCHIP

La scheda di sviluppo uChip non è costituita funzionalmente solo da un microcontrollore, ma include anche un circuito di gestione dell'alimentazione che consente di fornire la tensione USB o 3,3 V regolati sul pin 16 (VEXT) quando uChip è alimentato tramite USB.

Può fornire una tensione di 5 V dal suo connettore micro-USB quando è alimentato esternamente il pin 16 VEXT con una tensione da 3,3 V a 5 V ed è collegato un cavo USB-OTG alla porta micro-USB di uChip. Questa 5 V può essere abilitata/disabilitata via software. La **Fig. 6** mostra lo schema a blocchi funzionale del sistema di gestione delle alimentazioni di uChip.

Sebbene uChip accetti e possa fornire una tensione esterna fino a 5 V al pin 16, tutti gli altri pin I/O NON sono compatibili con livelli logici e analogici a 5 V ma accettano solo livelli digitali o analogici di

Fig. 6
Schema a blocchi
funzionale del
sistema di gestione
delle alimentazioni
di uChip.



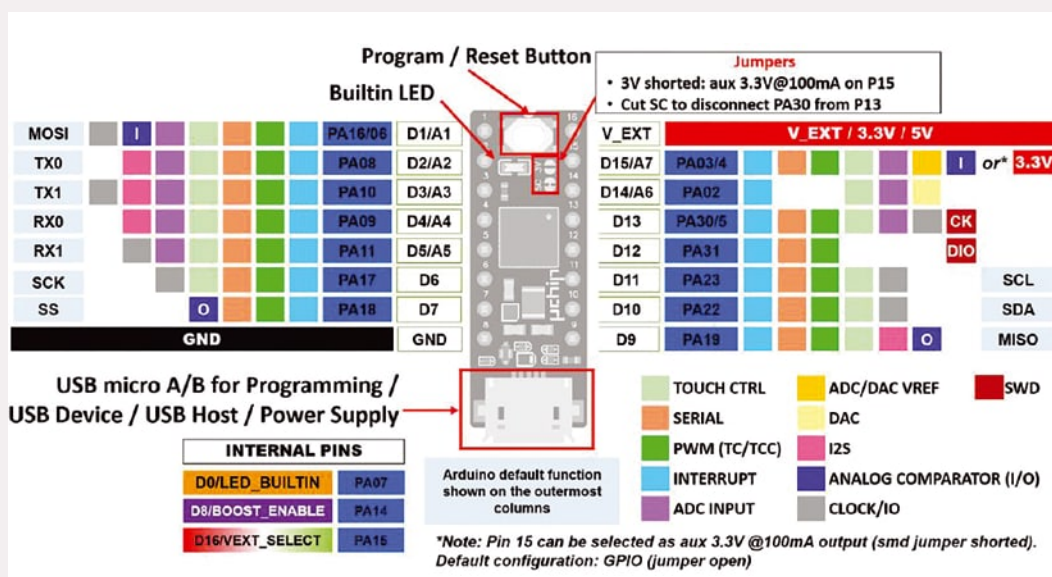


Fig. 7
Disposizione
dei pin della
uChip.

3,3 V. Forzare una tensione superiore a 3,3 V sui pin I/O danneggerà irreversibilmente uChip. Inoltre, non si deve applicare una tensione superiore a 5,5 V al pin 16, poiché, anche in questo caso, ciò danneggerebbe irreversibilmente uChip. Se è necessario collegare un carico induttivo (es. servomotori) al pin 16 VEXT di uChip, occorre prestare particolare attenzione predisponendo una circuiteria di protezione per ridurre le sovratensioni induttive che potrebbero far salire il livello di tensione sul pin VEXT oltre il valore massimo consentito di 5,5V e distruggere uChip. Nella **Fig. 7** viene mostrata la disposizione dei pin di uChip, dove, dalle colonne più interne verso quelle più esterne, viene indicato:

- il numero del pin digitale/analogico (colonna più interna);
- il numero della porta effettivo corrispondente al processore SAMD21 (questo è utile soprattutto se non si utilizza Arduino Framework).
- le funzionalità del particolare pin identificabile dal colore delle varie caselle.
- la funzione predefinita del pin di Arduino (colonne più esterne).

PIN, JUMPER, LED E PULSANTE DI UCHIP

La scheda uChip dispone di due jumper, SC e 3V evidenziati in **Fig. 8**. Il jumper SC è in cortocircuito per default e collega la porta PA30 al Pin 13 collegato anche alla porta PA05. La porta PA30 è per default il segnale di clock del sistema di debug SWD (Serial Wire Debug), un'interfaccia di pro-

grammazione e debug di Atmel ARM Cortex-M. Se SC viene interrotto, il Pin 13 non sarà più collegato alla porta PA30 ma sarà ancora collegato alla PA05. PA30 viene riferita a livello alto per default e deve restarci durante il reset per permettere al SAMD21 di funzionare, altrimenti rimarrà nello stato di reset. Ciò è dovuto all'implementazione SWD sul microcontrollore SAMD21. Se l'applicazione non necessita di debug SWD e se il pin 13 resta basso anche durante il reset, si può tagliare la piccola traccia che collega i pad del jumper SC. Nel caso di necessità di utilizzare la porta PA30 o la funzione SWD, basta cortocircuitare nuovamente il jumper SC con una goccia di stagno. Quanto al jumper 3V, è normalmente aperto, quindi il pin 15 riferito a questo jumper è un normale pin GPIO. Una volta cortocircuitato, il jumper 3V

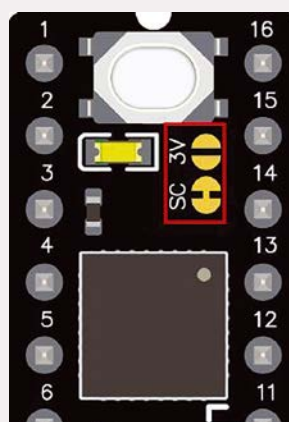
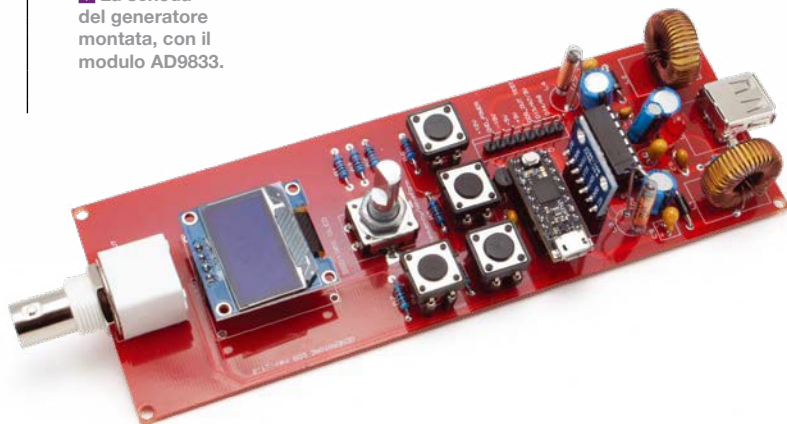


Fig. 8
Jumper SC
e 3V della
uChip.

↓ La scheda del generatore montata, con il modulo AD9833.



consente a uChip di fornire un'uscita ausiliaria a 3,3 V-100 mA sul pin 15. In questa modalità uChip può fornire sia 5V (sul pin 16-VEXT) che 3,3V (sul pin 15). Occorre tenere ben presente che questa uscita a 3,3V è fornita da un regolatore lineare da 3,3 V, collegato a VEXT (pin 16). Questo regolatore fornisce l'alimentazione anche al SAMD21, quindi, anche se il regolatore può erogare fino a 300 mA, non è consigliabile assorbire più di 100 mA per evitare un dannoso surriscaldamento.

La scheda uChip è dotata di un LED integrato collegato internamente al pin DO, che può essere facilmente controllato con una singola istruzione. La uChip include anche un pulsante a doppia funzione: una singola pressione attiva la funzione di reset di uChip; una doppia pressione attiva la modalità bootloader. Quest'ultima modalità è utile

quando si effettua la programmazione senza usare Arduino Framework (ad esempio se si utilizza Atmel Studio), oppure se lo sketch di Arduino non risponde (in genere causato da overflow del buffer, da interrupts disabilitati, e da altre cause).

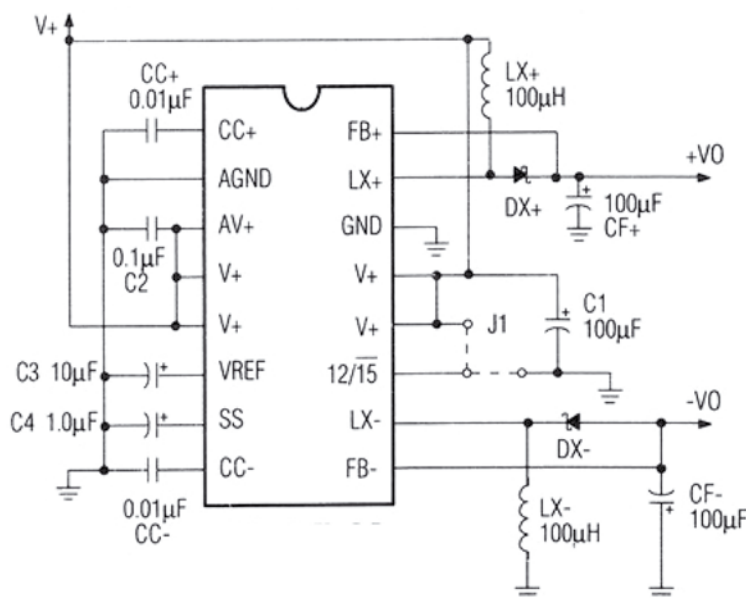
IL DC-DC CONVERTER DUALE MAX743

Riteniamo utile soffermarci su questo interessante integrato, per la sua peculiarità di produrre da una tensione di 5V una tensione duale di $\pm 12V$ o $\pm 15V$ con una corrente erogabile fino a $\pm 100mA$, come dichiarato nel datasheet del MAX743 di cui di seguito riportiamo alcune parti.

Applicando in ingresso una tensione di +5V e basandosi su due induttori piuttosto che trasformatori, il circuito integrato MAX743 è in grado di fornire una doppia uscita di $\pm 12V$ o $\pm 15V$ selezionabile sia staticamente che dinamicamente mediante un controllo logico. Il MAX743 regola entrambe le tensioni di uscita in modo indipendente con una tolleranza di $\pm 4\%$ in tutte le condizioni di carico e di temperatura. Il MAX743 in genere garantisce un'efficienza dal 75% all'82% sulla maggior parte delle variazioni di carico.

Il suo funzionamento switching si basa sul feedback in corrente a 200 kHz. Inoltre, l'ondulazione e il rumore è facilmente filtrabile.

Lo shutdown termico previene il surriscaldamento e il sensore della corrente protegge il transistor interruttore interno dell'alimentazione (funzione power switch). Altre caratteristiche includono il blocco sotto tensione e l'avvio graduale program-



➔ Fig. 9
Una tipica applicazione del MAX743.

mabile. Una tipica applicazione del MAX743 è mostrata in **Fig. 9**.

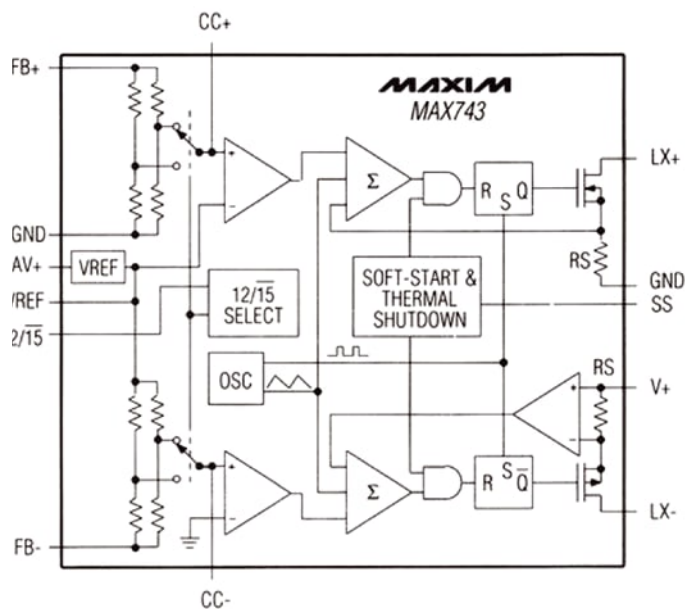
I componenti fondamentali in questo esempio di applicazione sono i due induttori di potenza da 100uH e i due diodi Schottky 1N5817.

Una coppia di induttori, diodi e alcuni condensatori sono sufficienti per realizzare un convertitore DC-DC con uscita duale a basso costo.

La **Fig. 10** propone lo schema a blocchi del MAX743. Il circuito interno del MAX743 è costituito da due amplificatori operazionali (amplificatori di errore) e da due amplificatori operazionali sommatore. Gli operazionali sommatore sommano tre segnali in ingresso: il segnale proveniente dal FET di potenza, il segnale di errore VOUT (pin FB)-VREF, il segnale a rampa generato dall'oscillatore. L'uscita del sommatore è applicata a un flip-flop che pilota alternativamente il FET switch di potenza. Entrambi gli switch sono sincronizzati dall'oscillatore e si accendono simultaneamente quando sono settati i flip-flop. I due FET switch si spengono quando la corrente di commutazione raggiunge una soglia determinata dal segnale di errore VOT-VREF. Questo funzionamento genera un treno di impulsi a 200 kHz modulati in duty-cycle, ovvero da impulsi PWM in cui la durata del fronte ON dell'impulso è proporzionale al valore di tensione del segnale di errore e dalla corrente di picco nell'induttore. Basse correnti di picco o alti valori di tensione del segnale di errore producono un duty-cycle fino ad un massimo del 90%.

PROGRAMMAZIONE DELLA UCHIP

Per poter utilizzare la uChip, innanzitutto occorre installare l'IDE di Arduino dal sito ufficiale www.arduino.cc (in questo articolo si fa riferimento alla versione software Arduino 1.8.15). Avviate l'IDE di Arduino e installate la scheda di sviluppo



uChip da File -> Impostazioni, alla voce "URL aggiuntive per il Gestore schede" nella barra di ricerca in basso inserite il seguente link:

http://github.com/ITACAIInnovation/uChip/raw/master/package_uChip_index.json.

Poi, con Strumenti > Scheda"...> Gestore Schede, digitate "uchip" nella barra di ricerca; selezionate e installate il firmware di gestione "uChip SAMD Boards" (**Fig. 11**). Eseguite l'installazione del package di uChip, tranne che per Windows 10, linux, Mac e OS X, per i sistemi operativi Windows 8, 8.1, 7, Vista e XP bisogna installare il driver per la comunicazione seriale USB di uChip con il computer. I driver si trovano nella cartella drivers di questo percorso: %USERPROFILE%\AppData\Local\Arduino15\packages\uChip\hardware\samd\1.0.1\drivers\, oppure, se avete una versione portatile di Arduino, troverete la cartella "drivers" in questo percorso: D:\...\\arduino-1.8.9\\portable\

Fig. 10
Schema a blocchi
del MAX743

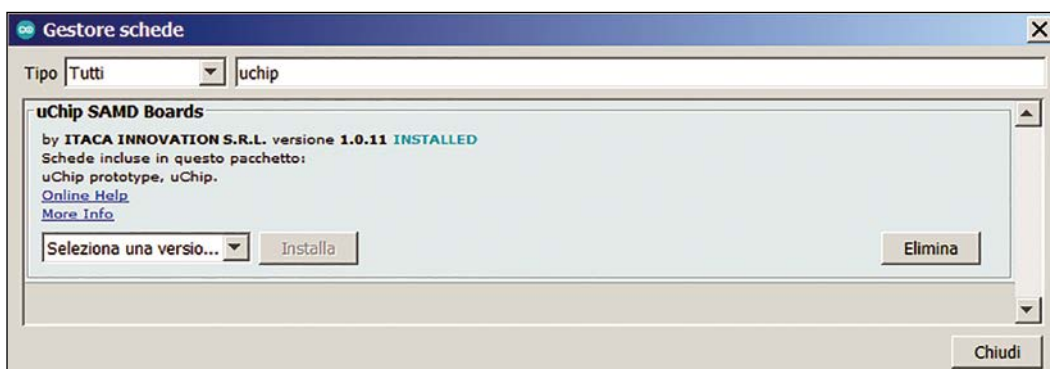
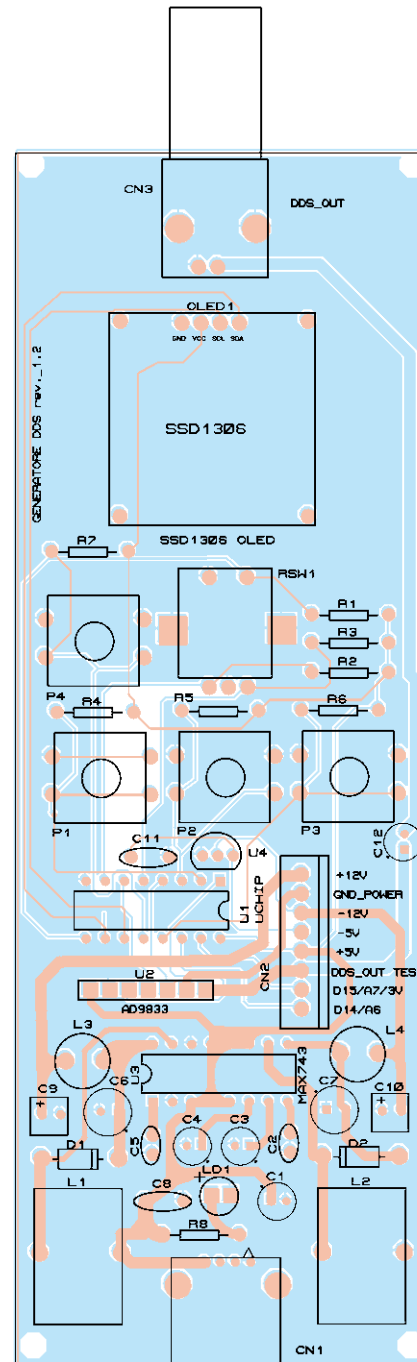


Fig. 11
Installazione
del firmware
di gestione di
uChip.



Elenco Componenti:

R1-R7: 10 kohm
 R8: 470 ohm
 C1: 100 μ F 25 V elettrolitico
 C6, C7: 100 μ F 35 V elettrolitico
 C2, C5: 10 nF ceramico
 C3: 1 μ F 100 V elettrolitico
 C4: 10 μ F 16 V Tantalio

C8, C11: 100 nF ceramico
 C9, C10: 2,2 μ F 35 V Tantalio
 C12: 47 μ F 25 V elettrolitico
 LD1: LED 5 mm rosso
 D1, D2: 1N5817
 OLED1: SSD1306
 L1, L2: 100 μ H toroidale
 L3, L4: 22 μ H assiale

U1: UCHIP
 U2: AD9833 breakout
 U3: MAX743
 U4: 79L05
 P1, P2, P3, P4: PULSANTE 12x12 mm da c.s.
 RSW1: Encoder rotativo
 CN1: USB-A FEMMINA da c.s.

CN2: pin strip 8 vie maschio
 CN3: BNC 90° da c.s.
 Varie:
 - pin strip 4 vie femmina
 - zoccolo 8+8 pin
 - Circuito stampato S1622 (159x55 mm)

packages\uChip\hardware\samd\1.0.11\drivers. Aprite la cartella "drivers" e fate doppio click sul file eseguibile "uChip_driver_installer.exe".

Windows chiederà l'autorizzazione per installare i driver, quindi cliccate su continua. La procedura di installazione dovrebbe richiedere alcuni secondi. Se non utilizzate l'IDE di Arduino, in alternativa potete scaricare i driver dal sito:

<https://github.com/ITACAIInnovation/uChip/raw/master/uChip1.0.1.zip>.

Una volta decompresso questo archivio, troverete i driver "uChip_driver_installer.exe" nella directory: uChip1.0.1\drivers\ . Poi, come sopra, con doppio clic sul file "uChip_driver_installer.exe" installate i driver seguendo la procedura guidata.

Procediamo ora con la creazione dello sketch dell'applicazione. Aprite l'IDE di Arduino e in un nuovo sketch copiate il codice riportato nella rivista. Salvate il codice con un nome, ad es. "GenerFunzioni". Chiudete lo sketch.

Prima di caricare il codice nella scheda di sviluppo uChip, occorre scaricare e installare i file delle librerie necessarie per la gestione del display OLED SSD1306, del generatore DDS AD9833 e della gestione delle alimentazioni della scheda di sviluppo uChip.

Librerie per la gestione del display OLED SSD1306

Dal sito https://github.com/adafruit/Adafruit_SSD1306 cliccate sul pulsante verde "code", poi su "Download ZIP". Copiate il file ZIP scaricato "Adafruit_SSD1306-master.zip" in una cartella. Dal sito <https://github.com/adafruit/Adafruit-GFX-Library>, con la stessa procedura, scaricate e copiate nella stessa cartella il file ZIP "Adafruit-GFX-Library-master.zip".

Librerie per la gestione del generatore DDS

Dal sito <https://github.com/BillWilliams1952/AD9833-Library-Arduino> scaricate e copiate il file ZIP "AD9833-Library-Arduino-master.zip". Aprite l'IDE Arduino, cliccate su Sketch -> #include libreria -> Aggiungi libreria da file .ZIP e, per ogni file ZIP, selezionate la cartella in cui avete copiato i tre file ZIP e aggiungeteli semplicemente cliccando sul nome del file e poi su "Apri", e la libreria verrà caricata nell'IDE di Arduino.

Libreria per la gestione delle alimentazioni di uChip

Il pacchetto di supporto della scheda di sviluppo uChip che avete installato precedentemente

include la libreria uChipPowerManagement che consente di configurare il sistema di alimentazione di uChip. Per utilizzare questa libreria si devono aggiungere allo sketch del codice che andremo a creare, i file uChipPowerManagement.h e uChipPowerManagement.cpp selezionando nell'IDE Sketch -> Aggiungi file -> uChipPowerManagement.h e poi ancora Sketch -> Aggiungi file -> uChipPowerManagement.cpp. Questi file si trovano nella cartella d'installazione di uChip: C:\Arduino...\packages\uChip\hardware\samd\1.0.11\libraries\uChipPowerManagement\src. Dopo aver aggiunto i file della libreria "uChipPowerManagement" chiudete e riaprite lo sketch.

DESCRIZIONE DEL CODICE

Il codice inizia includendo tutte le librerie necessarie alla gestione del modulo DDS AD9833, dell'OLEDD, dell'alimentazione della uChip, infine, viene inclusa una libreria di matematica necessaria poiché il codice esegue dei calcoli.

```
include <AD9833.h> // Library for AD9833 Module
#include <Wire.h> // Wire Library for OLED
#include <Adafruit_GFX.h> // Support Library for OLED
#include <Adafruit_SSD1306.h> // OLED library
#include "uChipPowerManagement.h"
#include <math.h> // Math Library
```

Poi vengono definiti tutti i pin di input e output di uChip relativi ai pulsanti, interruttore ON/OFF, l'encoder e dell'OLEDD:

```
#define SCREEN_WIDATA_PINH 128 // OLED display Width
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
#define SET_FREQUENCY_HZ A2
// Pushbutton To Set Frequency In Hz
#define SET_FREQUENCY_KHZ A3
// Pushbutton To Set Frequency In Khz
#define SET_FREQUENCY_MHZ 13
// Pushbutton To Set Frequency In Mhz
#define ENABLE_DISABLE_OUTPUT_PIN 12
// Pushbutton To Enable / Disable the Output
#define FNC_PIN 4
// Fsync Required by the AD9833 Module
#define CLK_PIN 5 // Clock Pin of the Encoder
#define DATA_PIN 7 // Data Pin of the Encoder
#define BTN_PIN 9
// Internal Push Button on the Encoder
```

Vengono definite anche le variabili:

- la variabile intera counter memorizza il valore dell'encoder;



- le due variabili successive *clockPin* e la variabile *clockPinState* che memorizza lo stato del pin di direzione di rotazione dell'encoder;
- la variabile *time*, che contiene i valori correnti del contatore del timer, utilizzata per l'antirimbombo dei pulsanti;
- la variabile *unsigned long moduleFrequency* che contiene la frequenza calcolata del segnale DDS che verrà applicata;
- la variabile *debounce* contiene il valore del ritardo di antirimbombo che può essere modificato se necessario;
- la variabile stringa *waveSelect* che memorizza lo stato della forma d'onda di uscita selezionata che per default è un'onda sinusoidale;
- la variabile *encoder_btn_count* che contiene il conteggio del pulsante dell'encoder utilizzato per selezionare la forma d'onda in uscita.

Le tre variabili booleane *set_frequency_hz*, *set_frequency_khz* e *set_frequency_mhz* vengono utilizzate per determinare l'impostazione corrente della frequenza del modulo AD9833, come vedremo dettagliatamente più avanti nell'articolo. Infine, sono riportati due oggetti, uno per il display OLED e l'altro per il modulo AD9833:

```
int counter = 1; // This Counter value will increase
// or decrease if when the rotary encoder is turned
int clockPin; // Placeholder for pin status used by
```

```
//the rotary encoder
int clockPinState; // Placeholder for pin status
//used by the rotary encoder
unsigned long time = 0; // Used for debouncing
unsigned long moduleFrequency;
//used to set output frequency
long debounce = 220; // Debounce delay
bool btn_state; // used to enable/disable output of
//the AD9833 Module
bool set_frequency_hz = 1;
// Default frequency of the AD9833 Module
bool set_frequency_khz;
bool set_frequency_mhz;
String waveSelect = "SIN";
// Startup waveform of the module
int encoder_btn_count = 0;
// used to check encoder button press
```

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_
HEIGHT, &Wire, -1);
AD9833 gen(FNC_PIN);
```

La funzione di configurazione *setup()* riportata di seguito, con la funzione *uChipSetVextValue(VEXT_USB)* imposta la uChip per ricevere alimentazione esterna applicata al suo connettore microUSB, inizializza il Monitor Seriale per il debug del codice; si noti l'istruzione *SerialUSB.begin()* che rispetto all'istruzione *Serial.begin()* normalmente utilizzata nei codici per dispositivi Arduino Uno e simili, viene

↓ Listato 1

```
void setup() {
    uChipSetVextValue(VEXT_USB);
    SerialUSB.begin(9600);
    gen.Begin(); // This MUST be the first command after declaring the AD9833 object
    pinMode(CLK_PIN, INPUT);
    pinMode(DATA_PIN, INPUT);
    pinMode(BTN_PIN, INPUT_PULLUP);
    clockPinState = digitalRead(CLK_PIN);
    pinMode(SET_FREQUENCY_HZ, INPUT);
    pinMode(SET_FREQUENCY_KHZ, INPUT);
    pinMode(SET_FREQUENCY_MHZ, INPUT);
    pinMode(ENABLE_DISABLE_OUTPUT_PIN, INPUT);
    if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64
        SerialUSB.println(F("SSD1306 allocation failed"));
        for (;;);
    }
    display.clearDisplay(); // Clear the Screen
    .....
    display.display(); // Update the Display

    delay(2000); // Delay of 2 SEC
    update_display(); // Call update_display Function
}
```



Listato 2

```
void loop()
{
  clockPin = digitalRead(CLK_PIN);
  if (clockPin != clockPinState && clockPin == 1) {
    if (digitalRead(DATA_PIN) != clockPin) {
      counter --;
    }
    else {
      counter ++; // Encoder is rotating CW so increment
    }
    if (counter < 1 ) counter = 1;
    SerialUSB.println(counter);
    update_display();
  }
  clockPinState = clockPin; // Remember last CLK_PIN state
}
```

impiegata per dispositivi uChip. Successivamente viene inizializzato il modulo AD9833 con il metodo *begin()*. Poi vengono assegnati come Input i pin dell'encoder rotativo e memorizzato il valore del pin di clock dell'encoder rotativo nella variabile *clockPinState*.

Successivamente sono impostati come INPUT tutti i pin dei pulsanti, e abilitato il display OLED con il metodo *display.begin()* verificandone la presenza di eventuali errori con un'istruzione if e, se tutto va bene, viene eseguito il clear del display e imposta una schermata iniziale di avvio con un ritardo di 2 secondi. Infine (**Listato 1**) viene chiamata la funzione *update_display()* che cancella lo schermo e aggiorna la visualizzazione ancora una volta (i dettagli di questo metodo verranno descritti più avanti nell'articolo).

Nelle seguenti sezioni della funzione *loop()* sono codificate tutte le principali funzionalità. Nella prima parte della funzione *loop()* viene letto lo stato del pin Clock dell'encoder e memorizzato nella variabile *clockPin*.

Poi, con l'istruzione if viene verificato se il valore precedente del pin e il valore corrente sono simili o meno. Se il risultato è vero, significa che l'encoder sta ruotando in senso antiorario e viene decrementato il valore del contatore con l'istruzione *counter--*, altrimenti il contatore viene incrementato con *counter ++*. Infine, viene utilizzata un'altra istruzione if per impostare il valore minimo del contatore a 1. Infine, viene aggiornato *clockPinState* con il valore *clockPin* corrente per un successivo utilizzo (**Listato 2**).

Nella successiva sezione della funzione *loop()* viene controllato il pulsante interno dell'encoder di selezione delle forme d'onda con alcune istruzioni if annidate.

La if (*digitalRead(BTN_PIN) == LOW && millis() - time > denounce*), controlla prima lo stato del pin associato al pulsante e se è LOW vuol dire che è stato premuto e, previa conferma dello stato del pulsante mediante la funzione *millis()* utilizzata per l'antirimbato del pulsante, si incrementa il valore della variabile *encoder_btn_count*.

Successivamente, viene utilizzata un'altra istruzione if per impostare il valore massimo del contatore su 2, necessaria per impostare la forma d'onda di uscita. Le tre istruzioni if consecutive impostano la forma d'onda, ovvero, se il valore del contatore è 0 viene selezionata la forma d'onda sinusoidale, se è 1 l'onda quadra e se è 2 l'onda triangolare. In tutte e tre queste istruzioni if, viene aggiornato il display con la funzione *update_display()*. Infine, si aggiorna la variabile *time* con il valore corrente del contatore del timer (**Listato 3**).

In quest'altra sezione di codice della funzione *loop()* vengono effettuate le letture di tutti i pulsanti rilevandone la corretta pressione con un ritardo antirimbato di 200 ms. Un valore LOW dei pin associato ad ogni pulsante imposta la gamma di frequenze in Hz, Khz e Mhz della forma d'onda in uscita del generatore di funzioni. Dopo il controllo di ogni pulsante viene aggiornato il display e la variabile *time* (**Listato 4**).

La seguente funzione *void update_display()* interna alla funzione *loop()* attua più di un semplice aggiornamento del display OLED in quanto una certa parte del display non può essere aggiornata. Per aggiornarlo completamente occorre rigenerare la schermata con nuovi valori rendendo più complesso il processo di codifica.

Quindi, si inizia con la cancellazione del display. Successivamente, si imposta la dimensione richiesta del testo. Poi si imposta il cursore e il

↓ Listato 3

```
//If we detect LOW signal, button is pressed
if ( digitalRead(BTN_PIN) == LOW && millis() - time > debounce) {
  encoder_btn_count++; // Increment the values
  if (encoder_btn_count > 2) // if value is grater that 2 reset it to 0
  {
    encoder_btn_count = 0;
  }
  if (encoder_btn_count == 0) { // if the value is 0 sine wave is selected
    waveSelect = "SIN"; // update the string variable with sin value
    update_display(); // update the display
  }
  if (encoder_btn_count == 1) { // if the value is 1 square wave is selected
    waveSelect = "SQR"; // update the string variable with SQR value
    update_display(); // update the display
  }
  if (encoder_btn_count == 2) { // if the value is 1 Triangular wave is selected
    waveSelect = "TRI"; // update the string variable with TRI value
    update_display();// update the display
  }
  time = millis(); // update the time variable
}
```

↓ Listato 4

```
/* Check buttton press action with analogread and digitalRead methods. Put in a slight delay to help de-
bounce the reading*/

// check analog pin with debounce delay.
if (analogRead(SET_FREQUENCY_HZ) < 30 && millis() - time > debounce) { //update boolean values
  set_frequency_hz = 1;
  set_frequency_khz = 0;
  set_frequency_mhz = 0;
  update_display();// update the display
  time = millis();// update the time variable
}
if (analogRead(SET_FREQUENCY_KHZ) < 30 && millis() - time > debounce) { // check analog pin with debounce delay
  //update boolean values
  set_frequency_hz = 0;
  set_frequency_khz = 1;
  set_frequency_mhz = 0;
  moduleFrequency = counter * 1000;
  update_display();// update the display
  time = millis();// update the time variable
}

// check digital pin with debounce delay
if (digitalRead(SET_FREQUENCY_MHZ) == LOW && millis() - time > debounce ) { // check digital pin with
debounce delay
  //update boolean values
  set_frequency_hz = 0;
  set_frequency_khz = 0;
  set_frequency_mhz = 1;
  moduleFrequency = counter * 1000000;
  update_display();// update the display
  time = millis();// update the time variable
}
if (digitalRead(ENABLE_DISABLE_OUTPUT_PIN) == LOW && millis() - time > debounce ) { // check digital pin
with debounce delay
  btn_state = ! btn_state; // Invert the button state
  gen.EnableOutput(btn_state); // Enable / Disable output of the function generator depending on button state
  update_display();// update the display

  time = millis();// update the time variable
}
```



testo "Function Generator" con l'istruzione *display.println("Function Function")*.

Si imposta nuovamente la dimensione del testo su 2 e il cursore su (0,20) con la funzione *display.setCursor(0, 20)*. Da questo punto si realizza il testo relativo all'informazione della forma d'onda selezionata (**Listato 5**). Nelle seguenti righe di codice, proseguendo con la funzione *void update_display()*, sono definiti il tipo di forma d'onda e il valore della frequenza. In sostanza, vengono controllati i pulsanti di selezione della gamma di frequenze in Hz, kHz e MHz, e aggiornato il relativo valore nella variabile *moduleFrequency*. Successivamente, dal valore della variabile *waveSelect* viene identificata la forma d'onda correntemente selezionata (**Listato 6**). Ora viene impostato nuovamente il cursore e aggiornato il valori del contatore. Occorre nuovamente controllare i valori della gamma di frequenze per aggiornarla sul display, per garantire una sicura acquisizione dei dati da parte dell'OLED (**Listato 7**). Infine, a completamento della funzione *void update_display()*, della funzione *loop()* e quindi

▼ Listato 5

```
void update_display()
{
    display.clearDisplay(); // First clear the display
    display.setTextSize(1); //set text Size
    display.setCursor(10, 0); // Set cursor position
    display.println("Function Generator"); //print the text
    display.setTextSize(2); //set text Size
    display.setCursor(0, 20); //Set cursor position
}
```

di tutto il codice, viene controllata la variabile *btn_state* associata allo stato del pulsante ON/OFF di abilitazione del segnale in uscita, per visualizzare sull'OLED l'informazione di output on/output off (**Listato 8**).

COLLAUDO

Per alimentare la scheda *GenDDS_AD9833* potete collegare con un cavo USB-USB la scheda ad un computer o ad un power bank con uscita a +5V, o,

▼ Listato 6

```
if (set_frequency_hz == 1 && set_frequency_khz == 0 && set_frequency_mhz == 0) { // check if button for
setting the frequency in Hz is pressed
    moduleFrequency = counter; //update the moduleFrequency variable with current counter value
}
if (set_frequency_hz == 0 && set_frequency_khz == 1 && set_frequency_mhz == 0) { // check if button for
setting the frequency in KHz is pressed
    moduleFrequency = counter * 1000; //update the moduleFrequency variable with current counter value but
we multiply 1000 to set it on KHz
}
if (set_frequency_hz == 0 && set_frequency_khz == 0 && set_frequency_mhz == 1) { // check if button for
setting the frequency in MHz is pressed
    moduleFrequency = counter * 1000000;
    if (moduleFrequency > 12000000)
    {
        moduleFrequency = 12000000; // do not let the frequency to be greater than 12Mhz
        counter = 12;
    }
}
if (waveSelect == "SIN") { // Sine wave is selected
    display.println("SIN");
    gen.ApplySignal(SINE_WAVE, REG0, moduleFrequency);
    SerialUSB.println(moduleFrequency);
}
if (waveSelect == "SQR") { // Sqr wave is selected
    display.println("SQR");
    gen.ApplySignal(SQUARE_WAVE, REG0, moduleFrequency);
    SerialUSB.println(moduleFrequency);
}
if (waveSelect == "TRI") { // Tri wave is selected
    display.println("TRI");
    gen.ApplySignal(TRIANGLE_WAVE, REG0, moduleFrequency); // update the AD9833 module.
    SerialUSB.println(moduleFrequency);
}
```

▼ Listato 7

```
display.setCursor(45, 20);
display.println(counter); // print the counter information on the display.

if (set_frequency_hz == 1 && set_frequency_khz == 0 && set_frequency_mhz == 0 ) {
    display.setCursor(90, 20);
    display.println("Hz"); // print Hz on the display
    display.display(); // when all set update the display
}

if (set_frequency_hz == 0 && set_frequency_khz == 1 && set_frequency_mhz == 0 ) {
    display.setCursor(90, 20);
    display.println("Khz");
    display.display(); // when all set update the display
}

if (set_frequency_hz == 0 && set_frequency_khz == 0 && set_frequency_mhz == 1) {
    display.setCursor(90, 20);
    display.println("Mhz");
    display.display(); // when all set update the display
}
```

in alternativa, collegate questi dispositivi con un cavo microUSB-USB alla scheda di sviluppo uChip. Eseguite questi collegamenti premete il pulsante P4 (ON/OFF). Si accenderà il LED indicatore della presenza della tensione di alimentazione +5V. Con un voltmetro digitale misurate sul connettore CN2 le tensioni di alimentazione +12V, -12V, -5V e +5V. Per testare le funzioni di generazione del segnale DDS per le tre forme d'onda occorre un oscilloscopio con una banda di almeno 20 MHz collegandolo con un cavo BNC-BNC al connettore BNC della scheda. Selezionando con il pulsante dell'encoder prima la forma d'onda sinusoidale e poi la triangolare, e agendo sui pulsanti P1 (Hz), P2 (kHz), P3 (MHz) selettori di banda e sulla manopola dell'en-

coder di regolazione della frequenza, verificate che la forma d'onda selezionata e il valore della frequenza del segnale di uscita corrispondano a quanto visualizzato sull'OLED.

Verificate che il livello della tensione di uscita delle forme d'onda sinusoidali e triangolari sia di circa 0,6 Vpp (+0,3V/-0,3V) da 1 Hz a 1 MHz e che questo valore si riduca gradualmente all'aumentare della frequenza fino a 0,3 Vpp a 10 MHz. Infine, con il pulsante dell'encoder selezionate l'onda quadra e verificate che l'ampiezza del segnale sia di circa 5 Vpp per tutta la banda da 1 Hz a 10 MHz. In un successivo articolo presenteremo il progetto della scheda di filtraggio e amplificazione *GenerDDS_Amplif&Filter* di cui vi abbiamo accennato nell'introduzione di questo articolo. Questa scheda consentirà di filtrare e amplificare il segnale generato dalla scheda DDS *GenDDS_AD9833*. □

▼ Listato 8

```
if (btn_state) {
    display.setTextSize(1);
    display.setCursor(65, 45);
    display.print("Output ON");
    // print output on to the display
    display.display();
    display.setTextSize(2);
}

else {
    display.setTextSize(1);
    display.setCursor(65, 45);
    display.print("Output OFF");
    // print output off to the display
    display.display();
    display.setTextSize(2);
}
}
```



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. La breakout board DDS (cod. AD9833BREAKOUT) è in vendita a Euro 6,90, la board uChip (cod. UCHIP) è disponibile a Euro 19,00, il display oled da 0,96" (cod. OLED096C) costa Euro 11,00. Il PCB può essere realizzato con il servizio PCBPRODUCTION disponibile dal sito www.futurashop.it I prezzi si intendono IVA compresa.

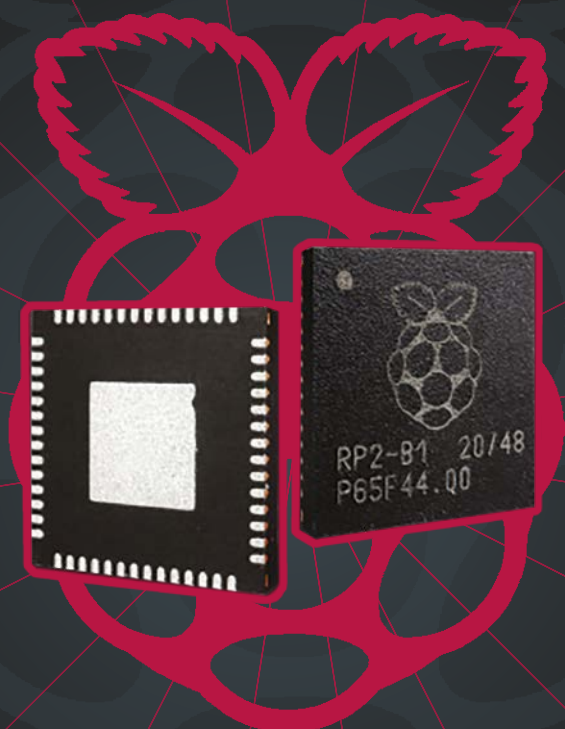
Il materiale va richiesto a:

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



Programmare con RP2040

di PIER CALDERAN



PUNTATA N. 2

Continuiamo a sperimentare con semplici dispositivi didattici creati con la scheda FTR2040 e scriviamone il codice utilizzando l'ambiente MicroPython, C/C++ e quello Arduino.

Nella puntata introduttiva di questo corso dedicato alla programmazione di applicazioni basate sulla scheda RP2040 vi abbiamo spiegato come utilizzare i pin di I/O della nostra scheda FTR2040 in uscita, ovvero come output, per applicazioni come ad esempio accendere e spegnere un LED. Ricordiamo che per tutti gli esempi proposti e che proporremo, faremo riferimento a una specifica versione della board, che è quella realizzata e commercializzata da Futura Elettronica: la FTR2040.

In questa seconda puntata andremo a realizzare qualche circuito applicativo nel quale faremo interagire la scheda con altri componenti, per implementare vere e proprie applicazioni di complessità crescente.

LED E PULSANTI

Oltre a collegare un LED a un qualsiasi contatto (pin) di I/O della scheda, potete cablare un semplice circuito che accende e spegne il LED stesso a seguito della pressione di un pulsante, connesso ad un pin che imposterete come ingresso. In pratica, al circuito che avete assemblato la volta scorsa, dovete solo aggiungere un pulsante nella breadboard, come indicato nella **Fig. 1**.

Eseguite queste connessioni:

1. collegate nella breadboard l'anodo del LED al pin GP15 con in serie una resistenza da 330 Ω ;
2. collegate nella breadboard il catodo del LED al negativo della breadboard;
3. collegate un terminale del pulsante al pin GP9 e l'altro terminale al positivo della breadboard.

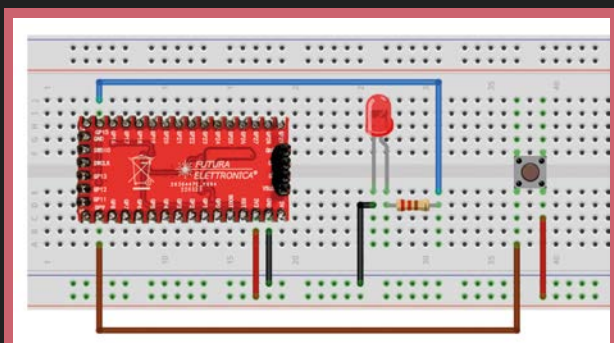


Fig. 1 - Collegamento di un LED e di un pulsante.

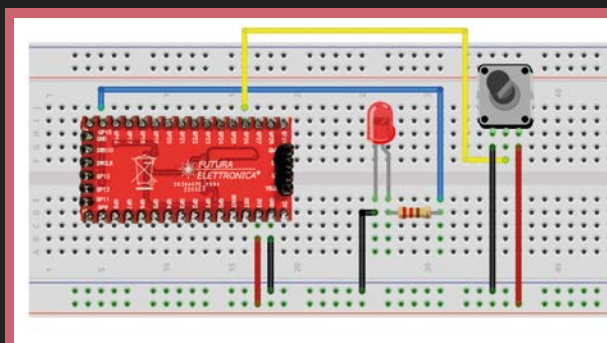


Fig. 2 - Lo schema di collegamento di un LED e di un potenziometro.

CODICE PER MICROPYTHON

Seguendo la stessa procedura spiegata la volta scorsa, collegate la scheda FTR2040 alla vostra board Raspberry Pi, quindi lanciate l'ambiente di sviluppo Thonny e scrivete queste poche righe di codice, stando attenti a rispettare l'indentatura:

```
import machine
import utime

led_external = machine.Pin(15, machine.Pin.OUT)
button = machine.Pin(9, machine.Pin.IN, machine.Pin.PULL_UP)

while True:
    if button.value() == 1:
        led_external.value(0)
    if button.value() == 0:
        led_external.value(1)
```

Come potete vedere:

- con l'istruzione `machine.Pin(15, machine.Pin.OUT)` si imposta il pin GP15 come uscita;
- con l'istruzione `machine.Pin(9, machine.Pin.IN, machine.Pin.PULL_UP)` si imposta il pin GP9 come ingresso in pull up, cioè sempre alto.

Una volta avviato il programma, tenendo premuto il pulsante, il LED si accenderà e, viceversa, si spegnerà. Questo perché l'ingresso è stato impostato come pull-up, cioè sempre alto (high o 1), e quando si porta la massa al pin tramite il pulsante, l'ingresso viene portato basso (low o 0). La lettura dei due stati del pulsante consente di gestire l'accensione e lo spegnimento del LED.

USARE LA PWM

Il convertitore analogico-digitale (ADC, Analog Digital Converter) di Pico prende un segnale analogico e lo converte in un segnale digitale così che il microcontrollore lo possa elaborare.

Al contrario, se il microcontrollore digitale dovesse creare

un'uscita analogica dovrebbe essere dotato di un convertitore digitale-analogico (DAC, Digital Analog Converter). Dal momento che il Pico non possiede un DAC ma solo uscite digitali, ovvero con uno stato alto o basso, possiamo usare la tecnologia PWM (Pulse Width Modulation o modulazione di larghezza di impulso) per "simulare" un segnale analogico.

In questo modo, potrete collegare un potenziometro a un ingresso ADC e un LED a un'uscita digitale e trattare questa come fosse PWM.

Vi ricordiamo che gli ingressi del convertitore analogico-digitale della scheda FTR2040 sono disponibili nei seguenti pin:

- GP26: ADC0
- GP27: ADC1
- GP28: ADC2

Noi useremo l'ingresso analogico ADC0 e quindi il pin GP26. Come vedremo, ci sarebbe anche un quarto ingresso ADC, chiamato ADC_VREF, che però viene utilizzato dal sensore di temperatura incorporato nella scheda.

Lo schema di collegamento del potenziometro e del LED è illustrato nella Fig. 2.

CODICE PER MICROPYTHON

Aprite Thonny e scrivete queste poche righe, rispettando l'indentazione:

```
import machine
import utime

potentiometer = machine.ADC(26)
led = machine.PWM(machine.Pin(15))
led.freq(1000)

while True:

    reading = potentiometer.read_u16()

    print(reading)
    led.duty_u16(reading)
    utime.sleep(.5)
```



Come potete vedere, con l'istruzione `machine.PWM(machine.Pin(15))` il LED viene attivato con la PWM sull'uscita GP15. L'istruzione `led.freq(1000)` imposta il duty cycle della PWM a 1000 cicli al microsecondo. L'ingresso analogico ADC0 è collegato al terminale centrale del potenziometro e viene impostato dall'istruzione `machine.ADC(26)`. In questo modo, abilita uno dei quattro ingressi analogici (GP28, GP25, GP26 e GP35). Una volta avviato il programma, ruotando il potenziometro, vedrete cambiare la luminosità del LED. Notare che la risoluzione di lettura dell'ADC è a 16 bit per cui si vedranno stampati nella shell i valori da 0 a 65535.

SENSORE DI TEMPERATURA

Come abbiamo detto, il sensore di temperatura incorporato nel chip RP2040 è collegato all'ingresso 4 dell'ADC. Il pin ADC supporta un intervallo di valori, determinato dalla tensione di ingresso applicata al pin.

Nel chip RP2040, i pin ADC supportano 12 bit, il che significa che il valore può andare da 0 a 4095. Ma il codice MicroPython può ridimensionare i valori ADC in un intervallo di 16 bit. Quindi otteniamo effettivamente l'intervallo da 0 a 65535.

Il microcontrollore funziona a 3,3 V, il che significa che un pin ADC restituirà un valore di 65535 livelli di tensione quando gli vengono applicati 3,3 V o 0 quando non c'è tensione. Quindi, possiamo ottenere tutti i valori intermedi quando la tensione applicata al pin è compresa tra 0 e 3,3 V. I pin ADC nella scheda Pico utilizzano il proprio schema di numerazione invece di utilizzare il numero pin GPIO. Come avete visto, i pin sopra sono etichettati ADC0, ADC1, ADC2 e ADC_VREF (che tecnicamente sarebbe ADC3), e sono i quattro pin ADC accessibili dall'esterno. Il sensore di temperatura non ha un pin fisico nella scheda ma è accessibile da MicroPython come ADC4.

LETTURA DEL VALORE DELLA TEMPERATURA IN MICROPYTHON

Ora vediamo il codice del sensore di temperatura in MicroPython. Scrivete il seguente codice:

```
import machine
import utime

sensor_temp = machine.ADC(4)
conversion_factor = 3.3 / (65535)

while True:
    reading = sensor_temp.read_u16() * conversion_factor
    temperature = 27 - (reading - 0.706) / 0.001721
    print(temperature)
    utime.sleep(2)
```

Avviate il programma e vedrete che nella shell verrà visua-

lizzato il valore della temperatura in gradi Celsius.

Il modulo `machine` fornisce la classe `ADC()` per lavorare con i pin ADC. Quindi, con l'istruzione `machine.ADC(4)`, si imposta la lettura diretta del sensore di temperatura, che abbiamo chiamato `sensor_temp`.

Alla variabile `conversion_factor` viene assegnato il fattore di conversione di 3.3 volt su 65535 livelli di lettura.

Con la seguente istruzione...

```
reading = sensor_temp.read_u16() * conversion_factor
```

... la variabile `reading` legge il valore del sensore di temperatura e lo moltiplica per il fattore di conversione.

Se stampate il valore del valore della temperatura, otterrete un numero intero compreso tra 0 e 65535. Per ottenere i gradi centigradi, dovete eseguire una seconda conversione:

```
temperature = 27 - (reading - 0.706) / 0.001721
```

Questa è un'operazione matematica specifica per il sensore di temperatura del chip RP2040. I valori sono presi da un documento tecnico disponibile al seguente link: <https://datasheets.raspberrypi.com>.

Questo succede perché il sensore di temperatura funziona fornendo una tensione al pin ADC4 che è proporzionale alla temperatura. Dal foglio dati, una temperatura di 27 gradi Celsius fornisce una tensione di 0,706 V.

Con ogni grado aggiuntivo la tensione si riduce di 1,721 mV o 0,001721 V.

Il primo passaggio nella conversione della temperatura a 16 bit è riconvertirla in volt, che viene fatto in base alla tensione massima di 3,3 V utilizzata dalla scheda.

Con questa conversione, il valore della temperatura viene stabilito con questa operazione:

`27 - (valore da 0 a 65535 - 0.706) / 0.001721`

LA TEMPERATURA SUL DISPLAY LCD I²C

Nello Starter Kit di Futura Elettronica (cod. RPICOKIT) potete trovare anche un display LCD compatibile con l'interfaccia I²C del chip RP2040 (codice LCD16X2AI2C). Fra le varie interfacce disponibili, questo display consente un collegamento davvero molto semplice.

Il chip del microcontrollore RP2040 ha due controller I²C. Potete accedere a entrambi i controller I²C tramite questi pin GPIO della scheda FTR2040:

I ² C CONTROLLER	GPIO
I2C0 – SDA	GP0/GP4/GP8/GP12/GP16/GP20
I2C0 – SCL	GP1/GP5/GP9/GP13/GP17/GP21
2C1 – SDA	GP2/GP6/GP10/GP14/GP18/GP26
2C1 – SCL	GP3/GP7/GP11/GP15/GP19/GP27

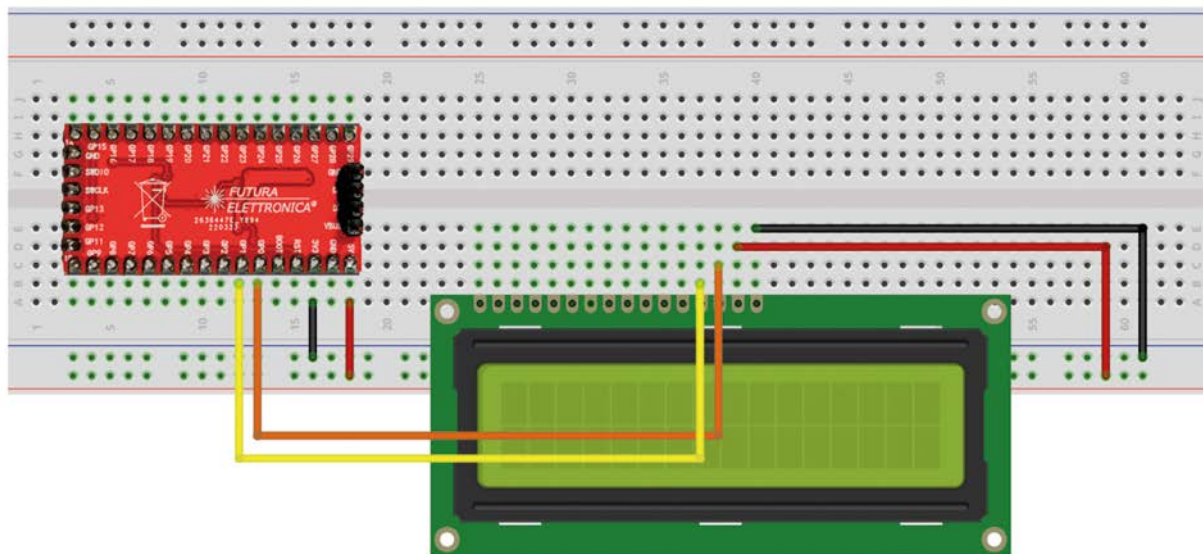


Fig. 3 - Il display LCD collegato a una delle porte I²C.

Ogni connessione del controller può essere configurata tramite più pin GPIO, ma prima di utilizzare un controller I²C, è necessario configurare nel software quali pin GPIO si desiderano utilizzare.

CARATTERISTICHE DEI PIN I²C

Il controller I²C del chip RP4020 supporta le seguenti funzionalità:

- modalità Master o Slave (indirizzo Slave predefinito = 0x055)
- tre modalità di velocità: Standard (da 0 a 100 Kb/s), Fast (minore o uguale a 400 Kb/s) e Fast Plus (minore o uguale a 1.000 Kb/s)
- buffer di trasmissione e ricezione
- può essere utilizzato in modalità di interruzione e DMA

Collegando il display a una delle porte I²C e l'alimentazione (Fig. 3), potete aggiungere poche righe allo script precedente per vedere l'output della temperatura sul display:

```
import machine
import utime

sda = machine.Pin(0)
scl = machine.Pin(1)

i2c = machine.I2C(0,sda=sda, scl=scl, freq=400000)
adc = machine.ADC(4)
conversion_factor = 3.3 / (65535)

while True:
    reading = adc.read_u16() * conversion_factor
```

```
temperature = 27 - (reading - 0.706)/0.001721
i2c.writeto(114, '\x7C')
i2c.writeto(114, '\x2D')

out_string = "Temp: " + str(temperature)
i2c.writeto(114, out_string)
utime.sleep(2)
```

L'istruzione machine *I2C(0,sda=sda, scl=scl, freq=400000)* imposta i pin GP0 e GP1 come interfaccia I²C in modalità Fast, ovvero alla velocità di 400 Kb/s.

Nel ciclo while sono state aggiunte le istruzioni *i2c.writeto()* per la scrittura nei registri 7C e 2D del display. La variabile *out_string* imposta la stringa di output con la scritta "Temp: " più il valore di temperatura *str(temperature)*, convertito in stringa.

Una volta avviato il programma, potrete vedere la temperatura sul display.

RASPBERRY PI PICO SDK

L'SDK Raspberry Pi Pico (d'ora in poi solo SDK) fornisce le librerie e il sistema di build necessari per scrivere programmi per i dispositivi basati su RP2040 in C, C++ o linguaggio assembly.

L'SDK è progettato per fornire un'API e un ambiente di programmazione agli sviluppatori C.

Un singolo programma viene eseguito su un dispositivo alla volta e viene avviato con un metodo convenzionale *main()*, che praticamente contiene tutte le funzioni principali. Le librerie C/C++ standard sono supportate insieme alle librerie/API di livello C per l'accesso a tutto l'hardware



dell'RP2040, incluso la modalità PIO (Programmed Input Output). Inoltre, l'SDK fornisce librerie di livello superiore per gestire timer, sincronizzazione, USB (TinyUSB) e programmazione multi-core insieme a varie utilità. L'SDK può essere utilizzato per creare qualsiasi cosa, da semplici applicazioni create con MicroPython a software di basso livello in C/C++.

AVVIARE RAPIDAMENTE UN PROGETTO IN C

Senza poter entrare nella specifica trattazione del linguaggio C e cercando di facilitare le cose, soprattutto ai principianti, queste istruzioni sono estremamente concise e basate solo su Raspberry Pi.

Per altre piattaforme come Windows o MacOS, è disponibile il documento Raspberry Pi Pico C/C++ SDK a questo indirizzo: <https://rptl.io/pico-c-sdk>

Una volta collegata la scheda FTR2040 a un Raspberry Pi, bisogna verificare che sia installata una versione CMake (almeno la versione 3.12) e il compilatore GCC.

Per fare questo, digitate il comando seguente su un terminale:

```
sudo apt install cmake gcc-arm-none-eabi libnewlib-arm-none-eabi
libstdc++-arm-none-eabi-newlib
```

Se il sistema operativo di Raspberry Pi è recente, dovrete avere già tutto installato e comparirà un messaggio simile a questo:

```
gcc-arm-none-eabi is already the newest version (15:8-2019-q3-1+b1).
libnewlib-arm-none-eabi is already the newest version (3.3.0-1).
libnewlib-arm-none-eabi set to manually installed.
libstdc++-arm-none-eabi-newlib is already the newest version
(15:8-2019-q3-1+b13).
libstdc++-arm-none-eabi-newlib set to manually installed.
cmake is already the newest version (3.18.4-2+rpt1+rp1).
0 upgraded, 0 newly installed, 0 to remove and 231 not upgraded.
```

A questo punto, dovete clonare l'SDK localmente, per esempio, in `/home/pi` (consigliato):

```
git clone https://github.com/raspberrypi/pico-sdk
```

Verrà creata una cartella `pico-sdk`. Dopodiché bisognerà configurare il percorso per utilizzare l'SDK.

Per fare questo, basta impostare il percorso `PICO_SDK_PATH` in modo che punti alla cartella dell'SDK.

Se avete clonato l'SDK nella directory di default, dovrete digitare nel terminale il comando seguente:

```
export PICO_SDK_PATH=/home/pi/pico-sdk
```

ESEMPI DELL'SDK

Per vedere se tutto funziona, dovrete clonare tutti gli

esempi dell'SDK dal repository seguente:

```
git clone https://github.com/raspberrypi/pico-examples
```

Per compilare tutti gli esempi, seguite questa procedura.

1. Entrate nella cartella `pico-examples`
2. Create una cartella di compilazione `cmake`:
`mkdir build`
3. Entrate nella cartella `build`:
`cd build`
4. Compilate tutti gli esempi (fate attenzione a immettere due punti dopo il comando `cmake`):
`cmake ..`
5. Create il target per tutti gli esempi, ovvero gli eseguibili veri e propri:
`make`

Abbiate pazienza, perché la compilazione di tutti gli esempi può impiegare diverso tempo. Tutti gli eseguibili saranno in formato `uf2` e si troveranno nelle rispettive cartelle all'interno della cartella `build`, cioè, per esempio:

```
/home/pi/pico-examples/build/blink/blink.uf2
```

Per provare l'esempio `blink`, collegate un LED al pin 25 della scheda (corrispondente al LED di default della scheda di Raspberry Pi Pico) seguite questa procedura.

1. Cortocircuitate il pin `BOOT` a massa e alimentate la scheda collegandola via USB a Raspberry Pi.
2. Togliete il corto.
3. Apparirà il prompt che vi chiederà di aprire la cartella del supporto rimovibile (Fig. 4).
4. Si aprirà una cartella simile alla Fig. 5.

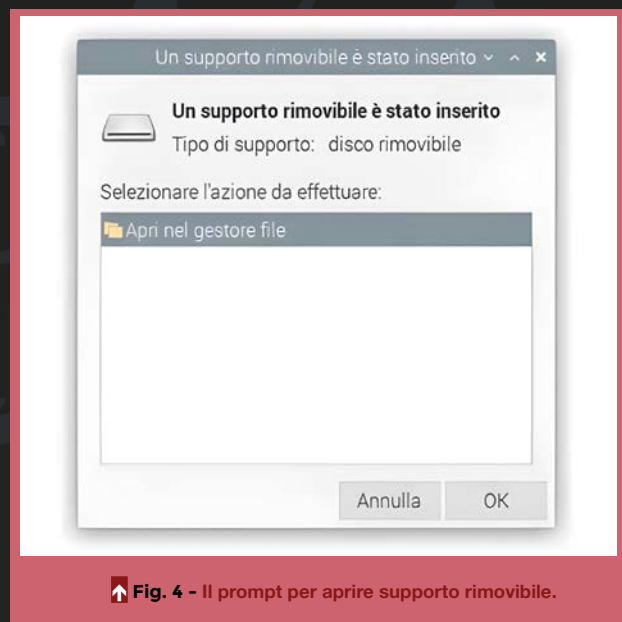


Fig. 4 - Il prompt per aprire supporto rimovibile.

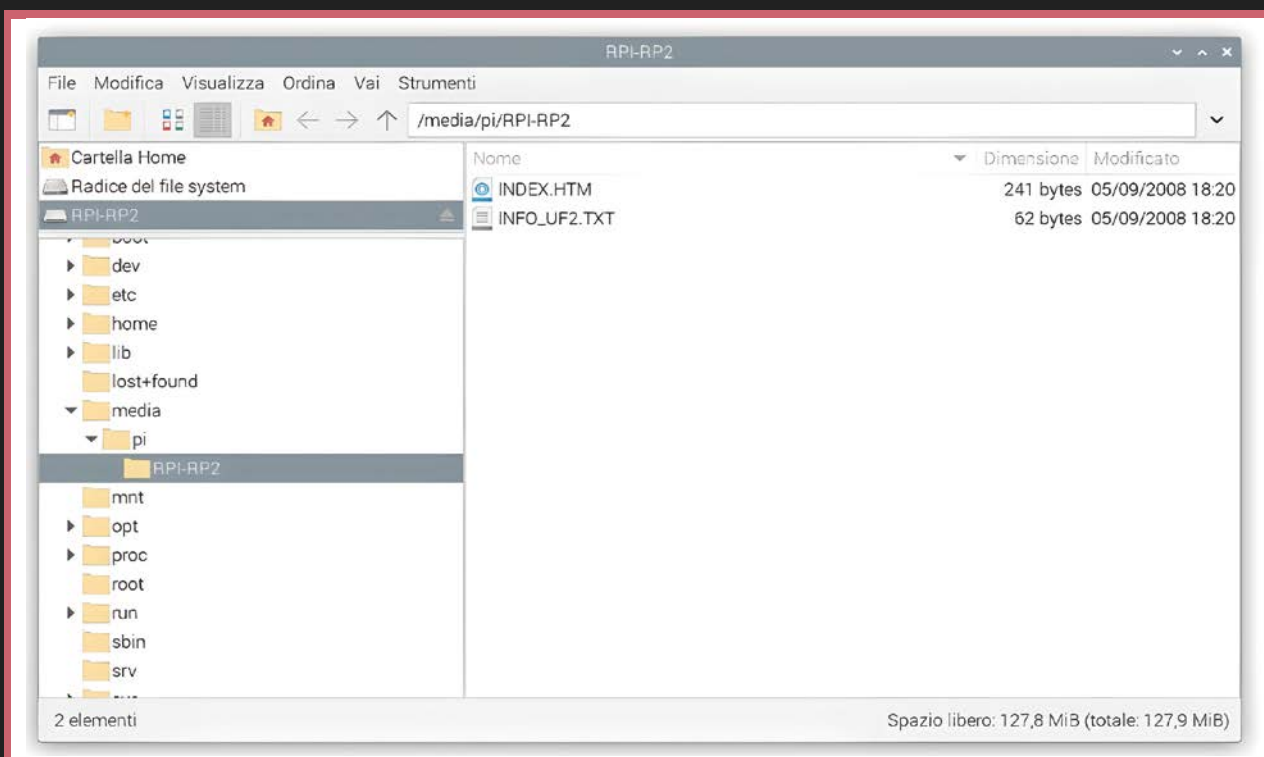


Fig. 5 - Il contenuto del supporto rimovibile.

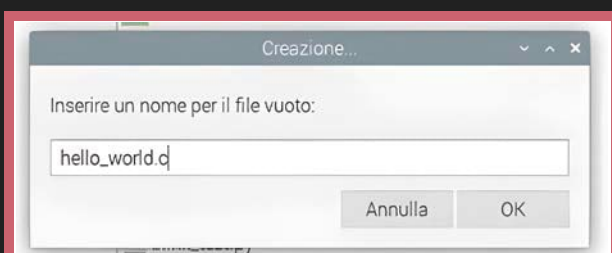


Fig. 6 - Il file di testo con estensione .c.

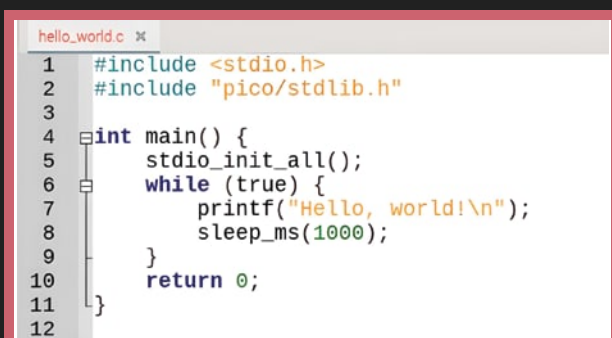


Fig. 7 - Il codice scritto all'interno dell'editor Geany.

5. Copiate al suo interno il file *blink.uf2*.

6. La cartella si chiuderà e vedrete il LED lampeggiare.

IL PRIMO PROGRAMMA IN C

Se non avete mai programmato in C, questo è il momento di iniziare, approfittando di questo corso e dell'SDK Raspberry Pi Pico. La cosa più semplice che potete fare all'inizio è creare un singolo file di testo, che chiamerete per esempio, "hello_world.c" (Fig. 6). Questo sarà il file sorgente

Lo potete creare ovunque, ma per dare un po' di ordine, vi consigliamo di creare prima una cartella "test" nella directory */home/pi* e poi una sottocartella "hello_world" all'interno della cartella test, ovvero, da terminale digitate i comandi seguenti:

```

mkdir test
cd test
mkdir hello_world

```

Aprirete il file *hello_world.c* con un buon editor per programmatori, per esempio, Geany che è già installato nel sistema, e scrivete queste righe di codice (Fig. 7):

```
#include <stdio.h>
```

```
#include "pico/stdlib.h"
int main() {
    stdio_init_all();
    while (true) {
        printf("Hello world!\n");
        sleep_ms(1000);
    }
    return 0;
}
```

Dopo aver incluso le librerie standard *stdio* e *stdlib*, il programma inizia all'interno della funzione *main()*.

Si noti che questo esempio utilizza l'istruzione *stdio_init_all()* che inizializza le porte seriali e USB del sistema. Questo per poter leggere i messaggi dagli ingressi seriali e visualizzarne il contenuto sul terminale di output (il terminale).

Nel nostro caso, dovremmo vedere stampata sul terminale la stringa *Hello, world!* mandata dalla scheda collegata via USB. Tramite l'istruzione *printf* infatti possiamo accedere alla periferica UART o USB.

Tecnicamente, *printf* scrive la stringa C a cui punta il formato della stringa nello standard output (*stdout*). Se il formato include identificatori, gli argomenti aggiuntivi che seguono il formato vengono formattati e inseriti nella stringa risultante.

Nel nostro caso, abbiamo questo:

```
printf("Hello world!\n");
```

Il formato della stringa è "Hello world!" più il carattere `\n` di a capo. Quello che apparirà sarà il messaggio Hello world! e un a capo.

Gli intervalli sono impostati a 1000 ms (un secondo) dalla funzione *sleep_ms*.

Tenete presente che il loop viene creato dall'istruzione *while (true)*, esattamente come in MicroPython e tanti altri linguaggi.

CMAKE

Per la compilazione dei file scritti C di solito si usa gcc (GNU Compiler Collection), sito ufficiale <https://gcc.gnu.org>. Si tratta di un compilatore C creato inizialmente per GNU/Linux e poi diffuso su tutte le piattaforme, Windows compreso.

La compilazione avviene a riga di comando e può essere poco agevole, soprattutto per i principianti.

Per semplificare le cose, possiamo usare CMake, un software libero multiplatforma per l'automazione della compilazione in C.

CMake permette con poche istruzioni di generare Makefile che possono venire eseguiti dall'utility make. CMake ha una particolare sintassi con moltissime macro che vengono immesse in un apposito file chiamato *CMakeLists.txt*.

Per la compilazione del programma è necessario verificare che siano incluse tutte le librerie e verificare i loro percorsi. Proprio per questo motivo, è impossibile avere un Makefile predefinito, per cui bisognerà sempre avere un file *CMakeLists.txt* e creare una cartella di compilazione build. Per la generazione del Makefile e la successiva compilazione del progetto, è necessario eseguire i seguenti comandi:

```
mkdir build
cd build
cmake ..
make
```

I due punti dopo il comando *cmake* stanno a significare che il file *CMakeLists.txt* si trova al livello di cartella esterno a build.

Con CMake è possibile annidare anche diversi file *CMakeLists.txt* in modo da rendere tutto estremamente facile. Vediamo come fare.

Il primo file CMakeLists.txt

Create nella cartella *hello_world* un nuovo file, chiamatelo *CMakeLists.txt*, apritelo con Geany e scrivete al suo interno queste righe (notare i commenti con anteposto il simbolo #):

```
# se il target è inizializzato come usb..
if (TARGET tinyusb_device)

# ... aggiunge l'eseguibile
add_executable (hello_world hello_world.c)

# estrae le dipendenze comuni
target_link_libraries (hello_world pico_stdlib)

# abilita l'uscita USB e disabilita l'uscita uart
pico_enable_stdio_usb (hello_world 1)
pico_enable_stdio_uart (hello_world 0)

# crea file map/bin/hex/uf2 ecc.
pico_add_extra_outputs (hello_world)

# aggiunge l'URL tramite pico_set_program_url
example_auto_set_url (hello_world)

# altrimenti se il target non è inizializzato come USB, appare
il messaggio seguente
elseif (PICO_ON_DEVICE)
    message (" hello_world non viene compilato perché TinyUSB
non è inizializzato nell'SDK")
endif ()
```

Il secondo file CMakeLists.txt

Tornate nella cartella test e copiate al suo interno i

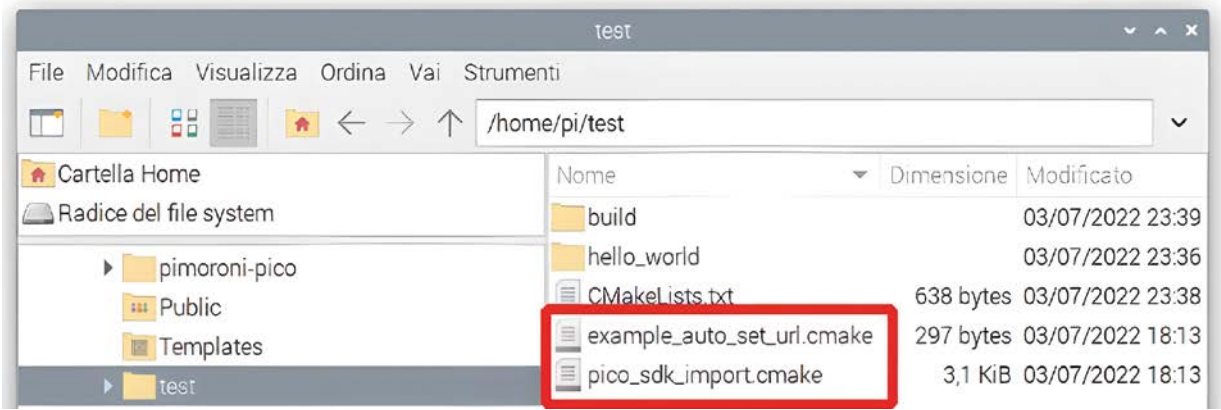


Fig. 8 - Il contenuto della cartella test.

seguenti file che trovate nella cartella *pico-examples* che avete clonato in precedenza (Fig. 8):

```
example_auto_set_url.cmake
pico_sdk_import.cmake
```

Create un nuovo file *CMakeLists.txt* e scrivete al suo interno (notate i commenti con anteposto il simbolo #):

```
cmake_minimum_required (VERSION 3.12)
```

```
# Inserisce l'SDK (deve essere prima del progetto)
include(pico_sdk_import.cmake)
```

```
project(pico_examples C CXX ASM)
set(CMAKE_C_STANDARD 11)
set(CMAKE_CXX_STANDARD 17)
```

```
set(PICO_EXAMPLES_PATH ${PROJECT_SOURCE_DIR})
```

```
# Inizializza l'SDK
pico_sdk_init()
```

```
include(example_auto_set_url.cmake)
```

```
# Aggiunge l'esempio hello_world ovvero la cartella sua cartella
che contiene il CMakeLists.txt creato in precedenza
```

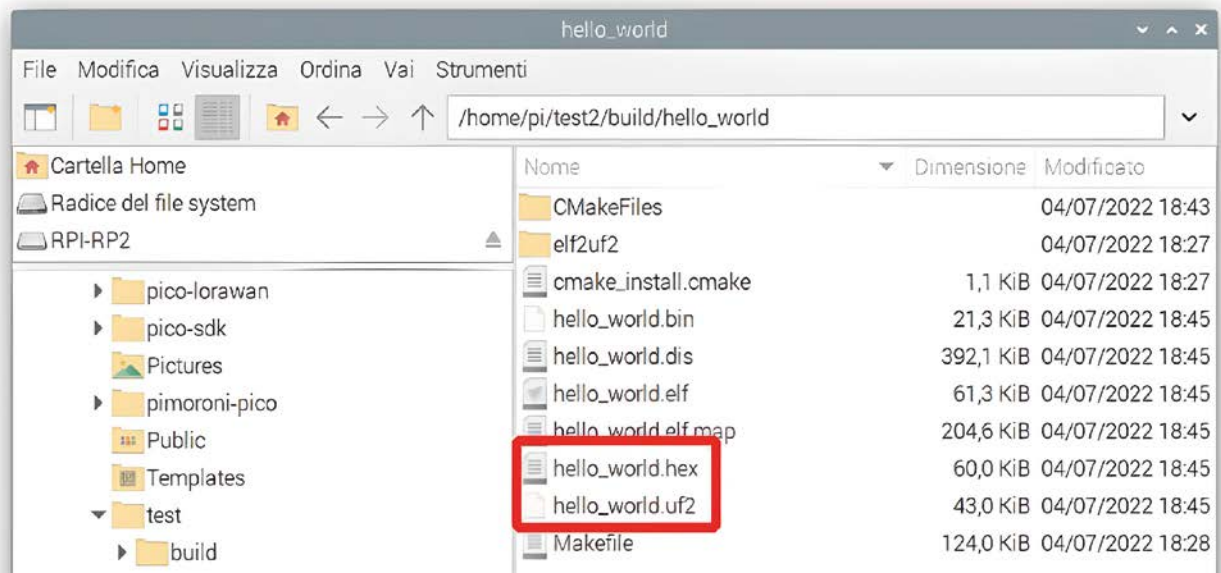
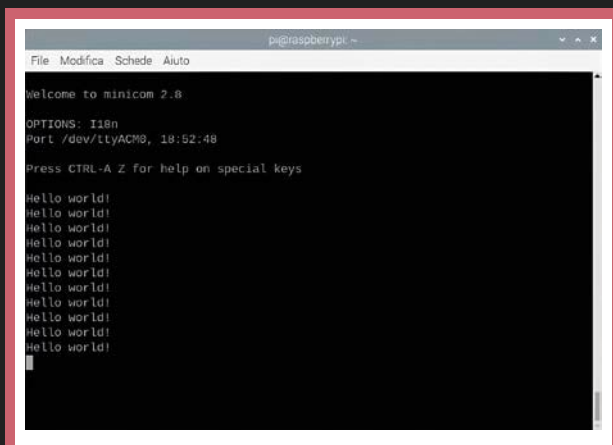


Fig. 9 - Il contenuto della cartella build /hello_world.



↑ Fig. 10 - L'output del programma nel terminale.

```
add_sottodirectory (hello_world)

# Aggiunge le opzioni di compilazione
add_compile_options(-Wall -Wno-format -Wno-unused-function
-Wno-maybe-uninitialized)
```

COMPILARE CON CMAKE

A questo punto avete un *CMakeLists.txt* nella cartella *test* che fornisce tutte le istruzioni per includere librerie cmake e per aggiungere la sottodirectory *hello_world*. Portatevi nella cartella *test* e create una directory build di compilazione CMake.

```
mkdir build
cd build
cmake ..
```

Se non avete fatto errori, dovreste vedere nell'output del terminale i seguenti messaggi:

```
-- Configuring done
-- Generating done
-- Build files have been written to: /home/pi/test/build
```

A questo punto create il target dalla directory della build creata con il comando seguente:

```
make
```

Se non ci sono errori di compilazione, dovreste vedere alla fine dei messaggi qualcosa di simile a questo:

```
[ 17%] Linking CXX executable hello_world.elf
[100%] Built target hello_world
```

Significa che nella cartella *build/hello_world* troverete l'es-

guibile *hello_world.elf* assieme al file *hello_world.uf2* (Fig. 9). Basta solo copiare il file *hello_world.uf2* nella scheda FTR2040, seguendo la consueta procedura per il bootsel. Quando si aprirà l'unità removibile, trascinate il file *hello_world.uf2* al suo interno.

Per vedere l'output sul terminale digitate il comando seguente che apre l'applicazione minicom, che dovrebbe già essere installata nel sistema:

```
minicom -b 115200 -o -D /dev/ttyACM0
```

Nel terminale vedrete apparire il messaggio *Hello World!* a intervalli di un secondo (Fig. 10).

CODICE IN C DEL PULSANTE PIÙ UN LED

Adesso che tutto sembra più chiaro, dovreste essere in grado di rifare i programmi scritti in MicroPython con l'SDK in C. All'interno della cartella *test*, create una cartella, per esempio *button*. Quindi create un file, per esempio, *button.c*, apritelo con Geany e scrivere le righe di codice riportate nel **Listato 1** al suo interno.

Come potete vedere, oltre all'output su terminale, abbiamo aggiunto la libreria *gpio* che gestisce l'hardware della scheda. Una volta definito un pulsante con *BUTTON_PIN*, bastano solo le istruzioni della libreria *gpio* per impostare il pulsante

↓ Listato 1

```
#include <stdio.h>
#include "pico/stdlib.h"
#include "hardware/gpio.h"

#define BUTTON_PIN 9
#define LED_PIN 15

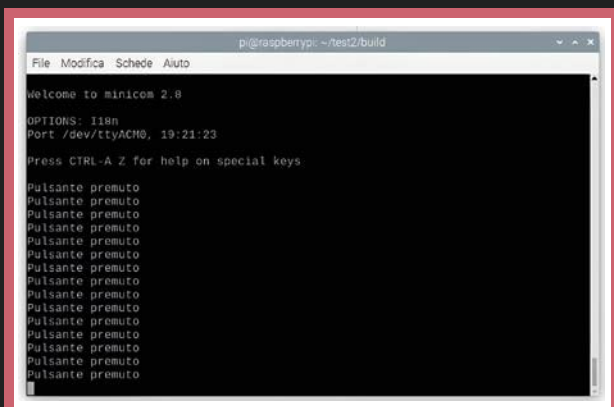
int main()
{
    stdio_init_all();

    gpio_init(LED_PIN);
    gpio_set_dir(LED_PIN, GPIO_OUT);

    gpio_init (BUTTON_PIN);
    gpio_set_dir (BUTTON_PIN, GPIO_IN);
    gpio_pull_up(BUTTON_PIN);

    while (true)
    {
        if (!gpio_get (BUTTON_PIN))
        {
            printf ("Pulsante premuto\n");
            gpio_put(LED_PIN, 1);

            sleep_ms(250);
        }
        else
        {
            gpio_put(LED_PIN, 0);
        }
    }
}
```

 **Fig. 11** - L'output del programma sul terminale quando viene premuto il pulsante.

come ingresso (pin 9) e il led come uscita (15). Nel ciclo *while*, se (*!gpio_get(BUTTON_PIN)*), ovvero se il pulsante viene portato a massa apparirà il messaggio "Pulsante premuto" e si accenderà il led. Altrimenti il led rimarrà spento. Create il file *CMakeLists.txt* copiando il precedente scritto per *hello_world* e sostituite tutte le occorrenze *hello_world* con *button*.

Listato 2

```
// le costanti sono usate qui per impostare i pin:
const int buttonPin = 9; // pin del pulsante
const int ledPin = 15; // pin del LED

// le variabili di stato cambieranno in base alla pressione
del pulsante:
int buttonState = 0; // variabile per la lettura dello stato
del pulsante

void setup() {
    // inizializza il pin del LED come output:
    pinMode(ledPin, OUTPUT);

    // inizializza il pin del pulsante come input pullup (sempre
    alto):
    pinMode(buttonPin, INPUT_PULLUP);
}

void loop () {
    // legge lo stato del valore del pulsante:
    buttonState = digitalRead(buttonPin);

    // controlla se il pulsante è premuto.
    if (buttonState == HIGH) {

        // tiene spento il LED
        digitalWrite(ledPin, LOW);

    // altrimenti, se lo è, lo stato del pulsante è LOW..
    } else {

        // accende il LED:
        digitalWrite(ledPin, HIGH);
    }
}
```

```
if (TARGET tinyusb_device)
    add_executable (button    button.c)
    target_link_libraries (button pico_stdlib)
    pico_enable_stdio_usb (button 1)
    pico_enable_stdio_uart (button 0)
    pico_add_extra_outputs (button)
    example_auto_set_url (button)

elseif (PICO_ON_DEVICE)
    message(WARNING "not building button because TinyUSB submo-
    dule is not initialized in the SDK")
endif ()
```

Entrate nella cartella *test* e modificate il file *CMakeLists.txt* aggiungendo solo questa riga:

```
add_subdirectory (hello_world)
add_subdirectory (button)
```

In pratica, verrà aggiunta la cartella *button* alla compilazione. Se volete compilare i due programmi *hello_world* e *button* digitate il comando seguente:

```
make
```

Se volete compilare solo *button* o solo *hello_world*, digitate uno dei seguenti comandi:

```
make button
make hello_world
```

Una volta compilato senza errori il file *button*, troverete il file eseguibile *button.uf2* nella cartella *build/button*. Copiatelo nella scheda come al solito, avviate minicom e premete il pulsante. Vedrete nel terminale qualcosa di simile alla **Fig. 11**, mentre il led si accenderà e si spegnerà alla pressione del pulsante.

CODICE PER ARDUINO IDE

Abbiamo spiegato nella scorsa puntata come usare l'IDE di Arduino per programmare la scheda FTR2040. Lasciando inalterati i circuiti assemblati finora, basta collegare la scheda a una porta USB di un PC e avviare l'IDE di Arduino.

Selezionate la scheda Raspberry Pico dal menu Strumenti > Schede > Raspberry Pi RP2040 Boards (2.0.1) e la porta seriale a cui è collegata la scheda.

CODICE PER ARDUINO

Sketch pulsante/LED

Fate riferimento al codice riportato nel **Listato 2** e caricate lo sketch nella scheda per vedere accendersi il LED ad ogni pressione del pulsante.



Listato 3

```
// seleziona il pin di ingresso per il potenziometro
int sensorPin = A0;

// seleziona il pin per il LED
int ledPin = 15;

// variabile per memorizzare il valore proveniente dal sensore
int sensorValue = 0;

void setup() {
  // dichiara il ledPin come OUTPUT:
  pinMode(ledPin, OUTPUT);

  // apre la seriale
  Serial.begin(57600);
}

void loop() {
  // legge il valore dal sensore:
  sensorValue = analogRead(sensorPin);

  // scrive il valore analogico su ledPin
  analogWrite(ledPin, sensorValue);

  // stampa il valore analogico sul monitor seriale
  Serial.println(sensorValue);
  delay(10);
}
```

Sketch potenziometro/LED

Fate riferimento al codice riportato nel **Listato 3** e caricate lo sketch nella scheda per mandare il valore analogico letto dal potenziometro all'uscita PWM del LED. Facciamo notare che il numero del pin A0, quello che di solito si usa con le schede Arduino, viene convertito automaticamente come numero del pin ADC0 del chip RP2040. Una volta caricato lo sketch, vedrete variare la luminosità del LED in base alla rotazione del potenziometro e i valori da 0 a 1023 stampati sul monitor seriale.

Sketch temperatura

Fate riferimento al codice riportato nel **Listato 4** e caricate

Listato 4

```
void setup() {
  Serial.begin(57600);
  delay(1000);
}

void loop() {
  Serial.printf("Ambient temperature: %2.1fC\n",
    analogReadTemp());
  delay(1000);
}
```

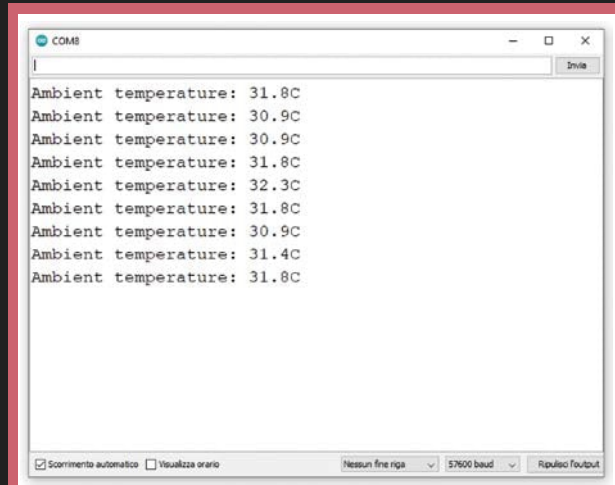


Fig. 12 - L'output del programma sul terminale quando viene premuto il pulsante.

lo sketch nella scheda per leggere e stampare l'output del sensore di temperatura interno al chip RP2040. Vi facciamo notare che viene usata solamente la funzione `analogReadTemp()`, già inclusa nella libreria Raspberry Pico e che fa tutto. Una volta caricato lo sketch, vedrete sul monitor seriale i valori di temperatura in gradi Celsius (**Fig. 12**).

CONCLUSIONI

Con la speranza di essere stati sufficientemente chiari, vi invitiamo a sperimentare i progetti presentati in queste pagine e vi diamo appuntamento alla prossima puntata. Per vostra comodità, ricordiamo che tutto il codice degli esempi qui proposti è disponibile nell'area di download del nostro sito Internet www.elettronica.in, insieme ai file di questo numero di Elettronica In. ■

**Cosa occorre?**

La board di Futura Elettronica con RP2040 (cod. FTR2040) è disponibile a Euro 13,90 e include i pin strip da saldare. Dal sito futura.net è possibile acquistare lo starter kit con Raspberry Pi Pico (cod. RPICOKIT) a Euro 49,90 o il solo Raspberry Pi Pico (cod. RPI-PICO) a Euro 6,90. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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



Droni e robot per le rinnovabili

I parchi eolici e solari e gli impianti idroelettrici di Enel Green Power si avvalgono di droni, robot e altri dispositivi tecnologici per svolgere le attività di esercizio e manutenzione. I parchi eolici implicano, sotto l'aspetto delle attività di esercizio e manutenzione (O&M, Operation and Maintenance) sfide non indifferenti: rilevare potenziali guasti delle apparecchiature, pulire le pale eoliche in seguito a fenomeni atmosferici elettrici e proteggere gli impianti dai fulmini; a quest'ultima esigenza risponde il sistema di protezione dai fulmini LPS (Lightning Protection System) in grado di attrarre le scariche elettriche verso punti di impatto controllato, allo scopo di evitare avarie alle pale delle turbine. I droni vengono usati per ispezionare la continuità elettrica e all'isolamento dei recettori, per esaminare le condizioni delle pale ed evitare malfunzionamenti. Di solito queste ispezioni vengono svolte manualmente da squadre composte da tre tecnici addestrati e qualificati, che, a seconda delle condizioni atmosferiche, possono impiegare circa due giorni di lavoro con turni di otto ore; il drone permette di eseguire l'attività con un grado di precisione maggiore in meno di tre ore per aerogeneratore.

Nel caso degli impianti idroelettrici, un robot in miniatura è capace di eseguire attività di ispezione in zone pericolose. Il robot è dotato di una sorta di "naso elettrico" per rilevare la presenza di gas. Nell'impianto del Vajont, in Veneto, si sta utilizzando Guardian S, una specie di serpente che può operare in tubature fino a 18 cm di diametro. Tra le altre soluzioni tecnologiche spicca l'utilizzo di fibra ottica per il rilevamento del ghiaccio sulle pale al parco eolico di Zoodochos Pigi, in Grecia.

Per le ispezioni negli impianti solari si stanno impiegando droni equipaggiati con termocamere, il drone Sparrow 1.3 e l'ARP (Autonomous Robot Platform) che si serve di droni e di un rover terrestre.

www.enelgreenpower.com

Vega C completa con successo il volo inaugurale

Dopo una lunga attesa, il 13 luglio scorso il lanciatore Vega C ha completato con successo il suo volo inaugurale, portando nell'orbita pianificata il satellite scientifico LARES 2 (che costituiva il carico principale) e sei CubeSats per la ricerca (come carico secondario). LARES 2 è stato lanciato nell'ambito di una missione scientifica dell'Agenzia Spaziale Italiana ASI. I sei CubeSat per la ricerca scientifica da Francia, Italia e Slovenia, costituivano il carico secondario. Il lancio è stato il coronamento

dell'impegno pluriennale dell'ESA, del prime contractor Avio e dei partner industriali di 13 Paesi membri dell'ESA di portare avanti l'eredità del predecessore, Vega.

Vega C ha un notevole incremento di capacità di carico rispetto al vettore Vega, che vola fin dal 2012. Con i primi due stadi nuovi e uno stadio superiore potenziato, Vega C aumenta la sua performance dalle 1,5 tonnellate del Vega alle 2,3 per l'orbita polare a 700 km di altitudine.

Vega C è dotato di un nuovo e più po-



tente primo stadio, il P120C, derivato dal P80 del Vega. Subito sopra si trova il nuovo secondo stadio, Zefiro 40, poi come terzo stadio lo Zefiro 9, già usato dal Vega.

Lo stadio superiore, in grado di riaccendersi più volte, è anch'esso potenziato. L'AVUM + ha un'aumentata capacità di carico di propellenti liquidi, per portare i carichi su orbite multiple in base ai parametri di missione e per permettere un maggiore tempo operativo nello spazio, consentendo missioni di durata estesa.

Il motore P120C sarà usato anche come booster laterale per l'Ariane 6 in configurazione doppia o quadrupla. L'utilizzo di questo componente comune aumenta l'efficienza industriale e migliora la competitività di entrambi i lanciatori.

Questo successo è il frutto del lavoro congiunto di Avio e di ESA, ASI e CNES, e conferma, ancora una volta, l'eccellenza tecnologica italiana e di Avio nello spazio e in Europa.

La presenza al volo del Ministro per la Digitalizzazione e l'Innovazione Tecnologica Vittorio Colao, testimonia inoltre l'attenzione e l'impegno del Governo Italiano, che con il PNRR ha deciso di dare un ulteriore impulso alla ricerca e all'innovazione in un

settore chiave per il Sistema Paese e per l'accesso indipendente allo Spazio. Il volo inaugurale di Vega C è il primo di una serie di 7 lanci già contrattualizzati e programmati.

Avio è un gruppo internazionale leader nella realizzazione e nello sviluppo di lanciatori spaziali e sistemi di propulsione solida e liquida per il trasporto spaziale.

L'esperienza e il know-how acquisiti nel corso degli oltre 50 anni di attività consentono ad Avio di primeggiare nel campo dei lanciatori spaziali, della propulsione spaziale a solido, a liquido e criogenica, nonché nella propulsione tattica. Avio è presente in Italia, Francia e Guyana Francese con 5 stabilimenti ed impiega circa 1.100 persone altamente qualificate, di cui il 30% circa impegnate in attività di ricerca e sviluppo. Avio opera in qualità di prime contractor per i programmi Vega e Vega C, e di subcontractor per il programma Ariane, entrambi finanziati dalla European Space Agency (ESA), permettendo all'Italia di essere oggi uno tra i pochi Paesi al mondo in grado di produrre un vettore spaziale completo, insieme a Francia, Stati Uniti, Russia e Cina.

www.avio.it



Avanti tutta verso il litio geotermico

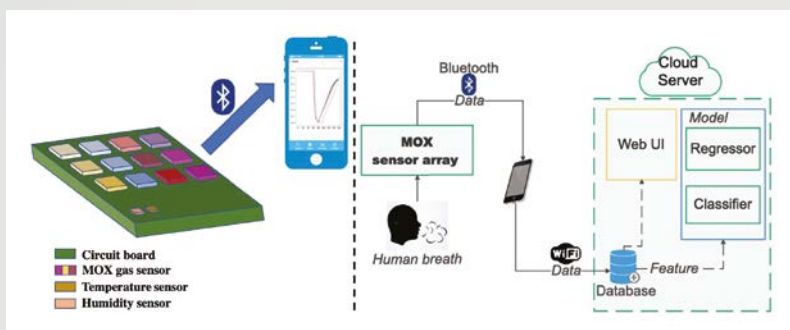
Uno dei principali ostacoli che rallentano la diffusione delle auto elettriche è il costo dell'elemento chiave delle loro batterie: il litio; per questo si cercano costantemente soluzioni per produrlo in gran quantità e perciò a basso costo. In questa direzione va un'intesa siglata fra Enel Green Power e Vulcan Energy, che mette al centro la ricerca sul litio geotermico valorizzando le opportunità di sviluppo offerte dalla licenza "Cesano", alle porte di Roma. Enel Green Power metterà le proprie competenze nella geotermia, mentre Vulcan porterà il proprio know-how sull'estrazione del litio.

La collaborazione prevede un approccio graduale, che in una prima fase di studio esplorativo per valutare se dalla licenza "Cesano" di Vulcan, sito che si estende su un'area di 11,5 kmq a pochi chilometri da Roma, sia possibile applicare la tecnologia di estrazione del litio geotermico, ossia di litio idrossido da salamoie ad alta temperatura combinando ciò con lo sfruttamento dell'energia geotermica.

Vulcan ambisce a diventare il primo produttore di litio al mondo in grado di azzerare le emissioni nette di gas serra, grazie alla sua tecnologia Zero Carbon Lithium, che punta a produrre idrossido di litio per batterie, combinando una fonte di energia geotermica e un giacimento di litio in Germania, il più grande esistente in Europa. Con l'esclusivo progetto Zero Carbon Lithium, Vulcan intende produrre sia l'energia geotermica rinnovabile che l'idrossido di litio dal medesimo giacimento di litio nel sottosuolo profondo. In tal modo, Vulcan intende soddisfare le richieste di litio del mercato UE, principalmente di quello automotive, riducendo l'elevata impronta idrica e di carbonio della produzione e la totale dipendenza dalle importazioni.

<https://v-er.eu/zero-carbon-lithium/>

Il naso elettronico rileva la glicemia



Per gestire il diabete, le persone malate devono testare i livelli di glucosio nel sangue più volte al giorno, il che spesso comporta punture dolorose. Negli ultimi anni sono entrati nel mercato alcuni dispositivi indossabili non invasivi per la misurazione del glucosio, ma sono in genere costosi e dipendono ancora dal campionamento diretto e dall'interazione con il sangue. Un naso elettronico in grado di misurare i livelli di glucosio in base al respiro potrebbe essere una

soluzione non invasiva e a basso costo; è questo che ha ispirato Qiliang Li, professore presso la George Mason University, che ha creato un e-Nose capace di determinare la glicemia attraverso la concentrazione di acetone e altri chetoni nel respiro, associata a condizioni metaboliche umane come il diabete. L'e-nose contiene 12 diversi sensori chimici e un microprocessore.

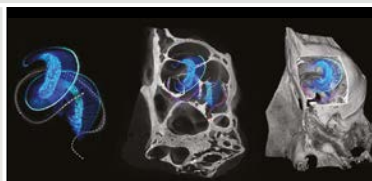
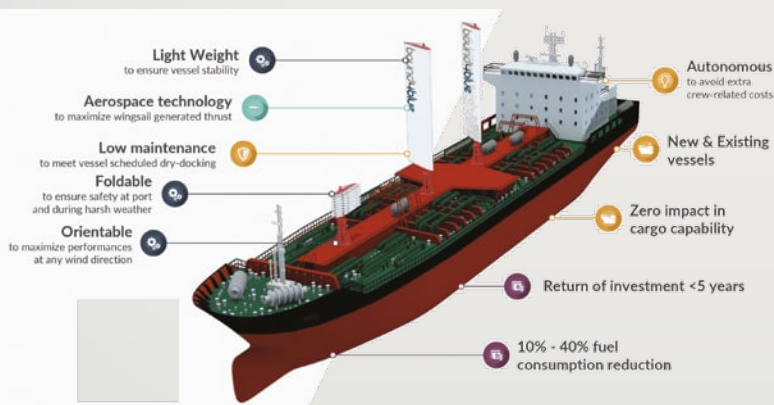
<https://ece.gmu.edu/>

Come le petroliere diventano velieri

La start-up spagnola Bound4blue ha sviluppato una propulsione basata su vele automatizzate mosse tramite cuscinetti a strisciamento e radiali a sfere esenti da lubrificazione e manutenzione. L'invenzione consiste in due sistemi di propulsione complementari: Wingsails ed E-Sails. Si basano su strutture alte circa 20 m, montate sul ponte, che si allineano automaticamente al vento, come una classica vela. La Wingsail ricorda l'ala di un aereo e si basa sugli stessi principi aerodinamici. Il vantaggio: è un sistema

pieghevole con poco spazio richiesto sul ponte. L'E-sail a colonna ha un meccanismo che aspira il vento sul dorso della vela durante la navigazione. Ciò riduce significativamente la turbolenza, perché il flusso d'aria racchiude la vela, determinando alta portanza con poca resistenza e un'efficienza fino a sette volte superiore a quella di una vela convenzionale. Wingsails ed E-Sails riducono fino al 40% il consumo di carburante.

<https://bound4blue.com/en/>



Ripristinare l'udito con fasci di luce

Il nostro udito dipende dalla coclea, una sorta di chiocciola posta nell'orecchio interno, che da anni può essere sostituita con dispositivi chiamati "impianti cocleari". Purtroppo tali impianti non ripristinano l'udito naturale, ma una sensazione che il paziente deve imparare a utilizzare. L'udito naturale nasce dalle vibrazioni che colpiscono minuscole strutture chiamate cellule ciliate all'interno della coclea nell'orecchio interno. Un impianto cocleare bypassa le parti danneggiate o disfunzionali dell'orecchio e utilizza elettrodi per stimolare direttamente il nervo cocleare, che invia segnali al cervello. Malgrado le molte sessioni con i tecnici per regolare le impostazioni degli impianti per rendere i suoni più piacevoli e reali, c'è un limite a ciò che può essere ottenuto con la tecnologia odierna. Per ovviare a questo inconveniente è stato sviluppato, da ricercatori delle Università di Göttingen e Friburgo, in Germania, un impianto cocleare basato su LED e cellule geneticamente modificate per rispondere alla luce. Utilizzando la luce per stimolare il nervo cocleare, tali impianti riprodurranno meglio l'intera natura spettrale dei suoni e simuleranno l'udito naturale. La sorgente di luce, un array di micro-LED flessibile o fibre ottiche, si avvolgerebbe a spirale attraverso la coclea e l'impianto potrebbe avere molti più siti di stimolazione, perché la luce è più facilmente confinata nello spazio rispetto alla corrente elettrica. Il paziente deve sottoporsi a un trattamento di terapia genica per rendere le cellule del nervo cocleare reattive alla luce, il che attiverebbe segnali che viaggiano lungo il nervo fino al cervello.

www.auditory-neuroscience.uni-goettingen.de/

Hantek®

Nuovo oscilloscopio **WiFi** per PC e tablet 2 canali - 70 MHz

ALIMENTAZIONE
AUTONOMA
CON BATTERIA
RICARICABILE
AL LITIO!



Prezzo IVA inclusa.

COD. DSOPC1070A

€ 265,00

CARATTERISTICHE TECNICHE PRINCIPALI

- Canali: 2
- Banda passante: 70 MHz
- Frequenza di campionamento (in tempo reale): 250 MS/s usando un solo canale - 125 MS/s usando due canali
- Impostazione del coefficiente di attenuazione della sonda: x1, x10, x100, x1000
- Tensione di ingresso massima: 35 Vpk (CC + picco)
- Sensibilità di ingresso: 10 mV/div ~ 5 V/div
- Funzione FFT: (Fast Fourier Transform) integrata

Può essere collegato contemporaneamente con iPad e iPhone.

ANTENNA WiFi

PRESA USB

CH 1

CH 2

EXT



ROME

Maker Faire

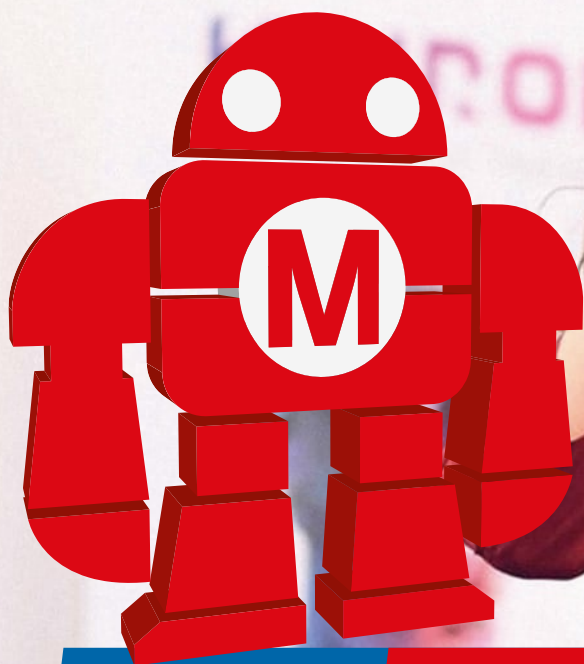
THE EUROPEAN EDITION

Powered by



Camera di Commercio
Roma

#MFR2022



7-9
OTTOBRE
2022

Scopri.INNOVA.Crea.
Gazometro Ostiense

Info e biglietti **makerfairerome.eu**

Promosso
e organizzato da



Con il supporto di



Main Partner



Gold Partner

sanofi



Silver Partner



Technical
Partner

