

Elettronica In

www.elettronica.in.it

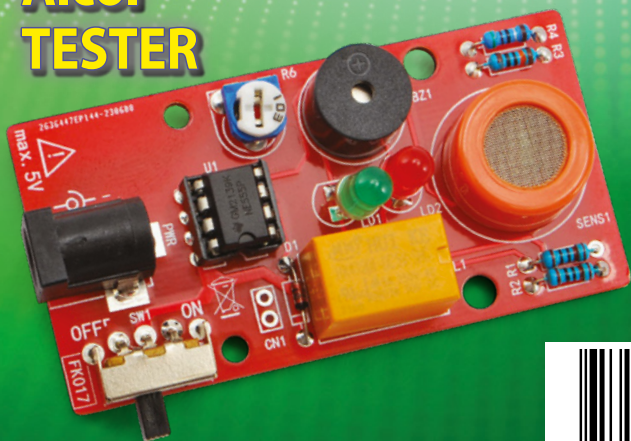
€ 7,50 - Bimestrale, anno XXIX - n. 275 - OTTOBRE / NOVEMBRE 2023

Internet Radio con Wi-Fi



160 PAGINE

Alcol TESTER



- ▶ LED sequenziali
- ▶ Apricancello con Ganimede
- ▶ Programmatore per ESP32-CAM
- ▶ Allarme batteria scarica
- ▶ Ricevitore 1 CH 434 MHz
- ▶ Misuratore di induttanze
- ▶ TOMBINGO il gioco elettronico per tutti
- ▶ Soluzioni IoT con Tuya
- ▶ Termostato ambiente sul Cloud
- ▶ Lumino a energia solare



E CORSO
EAGLE

ANCHE I DETTAGLI PIÙ PICCOLI DIVENTANO VISIBILI!

**MICROSCOPI DIGITALI IN GRADO DI CATTURARE IMMAGINI
CON UN'INGRANDIMENTO FINO A 1200X.**

MICROSCOPIO DIGITALE 1-1200X CON MONITOR LCD A COLORI DA 7"

Microscopio digitale con ampio monitor LCD a colori da 7 pollici e ingrandimento fino a 1200X. Dispone di una telecamera a colori che riprende l'oggetto da ingrandire e lo visualizza in alta risoluzione sullo schermo LCD.

Le foto e i video degli oggetti ingranditi possono essere salvati su una SD card inserita nell'apposito slot (max 128GB, acquistabile separatamente). Il microscopio dispone di una base stabile in alluminio e di doppia luce a LED con bracci flessibili. La distanza del punto di osservazione del campione può essere facilmente regolata tramite l'asta a cremagliera che consente di ottenere una distanza che va da 10 a 130 mm. Lo schermo è ruotabile di 360° e inclinabile. Può funzionare tramite batteria al litio integrata oppure tramite l'alimentatore USB in dotazione.



cod. LCDMICROS7

€154,00



Prezzi IVA inclusa.

MICROSCOPIO DIGITALE CON INGRANDIMENTO FINO A 600X CON MONITOR LCD DA 4,3"

Microscopio digitale portatile, dotato di telecamera a colori con sensore da 3,6 Megapixel, monitor LCD a colori da 4,3", illuminatore composto da 8 potenti LED a luce bianca, batteria al litio ricaricabile integrata, slot per micro SD card (max 64GB) e staffa a ventosa.



cod. LCDMICROS

€102,00

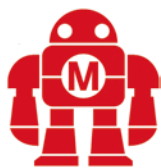
MICROSCOPIO USB DIGITALE CON INGRANDIMENTO FINO A 500X

Microscopio USB digitale a colori da 2 Megapixel dotato di stand con supporto regolabile in altezza. Si collega alla porta USB e consente di catturare e visualizzare immagini e video direttamente sullo schermo del computer. Dispone di 8 LED bianchi, con intensità luminosa regolabile.



cod. CP745

€28,00



Il Movimento Maker nell'Elettronica

Dal 20 al 22 ottobre tutti i Maker sono invitati a partecipare alla Maker Faire Rome. È l'occasione per noi per fare il punto sul fenomeno nato nei primi anni del XXI secolo.

Il Movimento Maker nell'elettronica ha rivoluzionato la creatività e l'innovazione, rendendo accessibili a tutti nuove tecnologie: strumenti come stampanti 3D e piattaforme come Arduino hanno reso la creazione di progetti elettronici più semplice e accessibile che mai. Questo fenomeno è basato sull'idea che "si può fare" e promuove la condivisione delle conoscenze

per migliorare la vita di tutti.

Il fare insieme coinvolge persone di diverse estrazioni sociali e abilità, permettendo loro di inventare con la testa, il cuore e le mani.

Il fare insieme coinvolge persone di diverse estrazioni sociali e abilità, permettendo loro di inventare con la testa, il cuore e le mani. Questo ambiente di collaborazione genera soluzioni creative e innovative, dimostrando il potenziale unico di ciascun individuo. Si può anche affermare che il Movimento Maker ha avuto un profondo

impatto sull'istruzione introducendo il concetto di STEAM (Scienza, Tecnologia, Ingegneria, Arte e Matematica) e quindi l'importanza della creatività nell'apprendimento. Questo approccio incoraggia gli studenti a diventare inventori e innovatori. I principi del Movimento Maker sono più attuali che mai e ci ricordano l'importanza di creare, esprimerci e contribuire alla società moderna. In un'era dominata dalla tecnologia pronta all'uso, abbracciare la mentalità dei maker è cruciale per stimolare l'innovazione e la creatività.

Vi aspettiamo a Roma!


Boris Landoni
@ boris.landoni@elettronica.in

ON LINE

@elettronica.in

SI RINGRAZIA

Futura Elettronica
Maker Faire
Mouser
Sicom



In copertina:
Un lettore musicale che sfrutta la potenza del modulo ESP32 per riprodurre brani da Internet e MP3, offrendo un'esperienza musicale versatile e accessibile.



Elettronica In

Anno XXIX n. 275

OTTOBRE / NOVEMBRE 2023

EDITORIALE

Direttore Responsabile

Boris Landoni

@ boris.landoni@elettronica.in

Redazione

Stefano Garavaglia, Paolo Gaspari,
Davide Scullino, Gabriele Daghetta

@ redazione@elettronica.in

Grafica

Alessia Sfulcini

@ alessia.sfulcini@elettronica.in

Ufficio Pubblicità

Monica Premoli (0331-752668)

@ monica.premoli@elettronica.in

Ufficio Abbonamenti

Tel. 0331-752668

@ abbonati@elettronica.in

Direzione, Redazione, Pubblicità
FUTURA GROUP srl - Divisione Editoriale
via Adige 11 - 21013 Gallarate (VA)

Tel. 0331-752668

Abbonamenti

Annuo 6 numeri Euro 39,00

Annuo 6 numeri digitale Euro 29,00

Le richieste di abbonamento vanno inviate a:

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

Tel. 0331-752668

oppure tramite il sito
<https://futuragroup.it/abbonamenti>

Distribuzione per l'Italia

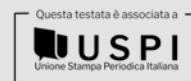
SO.DI.P. Angelo Patuzzi S.p.A.

via Bettola 18 - 20092 Cinisello Balsamo (MI)

Tel. 02-660301 Fax 02-66030320

Stampa

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



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

Prezzo di copertina Euro 7,50.

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

Poste Italiane Spa - Spedizione in abbonamento Postale - D.L. 353/2003 (conv. in L. 27/02/2004) art. 1 comma 1 - DCB Milano.

Futura Group srl è iscritta al Registro Operatori della Comunicazione n. 23650 del 02/07/2013.

Impaginazione ed immagini sono realizzati in DeskTop Publishing con programmi Adobe InDesign e Adobe Photoshop per Windows.

Tutti i contenuti della Rivista sono protetti da Copyright.

Ne è vietata la riproduzione, anche parziale, la traduzione e più in generale la diffusione con qualsiasi mezzo senza l'autorizzazione scritta da parte dell'Editore. I circuiti, il firmware ed il software descritti sulla Rivista possono essere realizzati solo per uso personale, ne è proibito lo sfruttamento a carattere commerciale e industriale. Tutti possono collaborare con Elettronica In. L'invio di articoli, materiale redazionale, programmi, traduzioni, ecc. implica da parte del Collaboratore l'accettazione dei compensi e delle condizioni stabilite dall'Editore (per maggiori informazioni scrivere a redazione@elettronica.in). Manoscritti, disegni e foto non richiesti non verranno in alcun caso restituiti. L'utilizzo dei progetti e dei programmi pubblicati non comporta alcuna responsabilità da parte della Società Editrice.

© 2023 Futura Group srl



Contenuti

01

RUBRICHE

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

Progetto di Copertina

ESP32 MUSIC PLAYER

59**11**

ARTICOLI

GADGET

11 LED sequenziali

Un punto luminoso si sposta su una barra da un lato all'altro, ciclicamente, a una velocità impostabile tramite un trimmer.

STRUMENTAZIONE

19 Alcol Tester

Consente di capire e avvisare acusticamente e visivamente se la concentrazione di alcol etilico nel fiato è oltre la soglia tollerabile, attivando eventualmente un relé.

AUTOMAZIONE

27 Apricancello IoT con Ganimede

Costruiamo un dispositivo in grado di controllare l'apertura e la chiusura di un cancello da remoto utilizzando la board Ganimede.E12, collegata ad un relé tramite uno dei suoi connettori di espansione.

LABORATORIO

45 Programmatore ESP32-CAM

Ospita e supporta la programmazione dei moduli ESP32 con telecamera, fornendo un'interfaccia USB.

IN AUTO

53 Allarme batteria scarica

Ci avvisa acusticamente quando rileva che la tensione dell'accumulatore di bordo si è abbassata sotto la soglia di carica minima, aiutandoci a non rimanere appiedati...

GADGET

LED sequenziali

**11**

STRUMENTAZIONE

Alcol Tester

**19**

AUTOMAZIONE

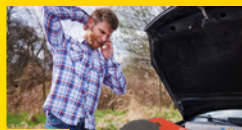
Apricancello IoT con Ganimede

**27**

LABORATORIO

Programmatore ESP32-CAM

**45**

IN AUTO**Allarme batteria scarica****53****WIRELESS****Ricevitore 1 CH 434 MHz****75****GADGET****TomBingo, gioco elettronico****83****DOMOTICA****Soluzioni IoT con Tuya****103****AUDIO****59 ESP32 music player**

Letto musicale con ESP32 in grado di riprodurre brani da Internet radio, file MP3 salvati localmente su una scheda SD e brani salvati su un server.

WIRELESS**75 Ricevitore 1 CH 434 MHz**

Abbinato a un trasmettitore commerciale, consente di controllare a distanza utilizzatori di vario genere mediante un relé che può operare in modalità sia impulsiva che a permanenza.

GADGET**83 TomBingo, il gioco elettronico per tutti**

Una rivisitazione elettronica della Tombola o del Bingo che unisce tradizione e tecnologia per un divertimento coinvolgente.

DOMOTICA**103 Soluzioni IoT con Tuya**

Scopriamo la tecnologia di Tuya per interconnettere i dispositivi in rete e realizzare un sistema di automazione casalingo in perfetto stile IoT.

DOMOTICA**115 Termostato ambiente sul Cloud**

Utilizziamo un sensore DHT22 per controllare la temperatura ambiente e l'umidità relativa e inviamo i dati ad Arduino IoT Cloud.

SOLARE**127 Lumino a energia solare**

Costruiamo un lumino elettronico alimentato da pannelli solari e batterie ricaricabili per una soluzione eterna, sostenibile, eco-friendly e senza costi energetici.

I PROGETTI DI eLektor**131 Un misuratore d'induttanze**

Hai bisogno di uno strumento per testare e misurare induttori? Questo strumento fai-da-te è un'economica alternativa rispetto alle costose soluzioni disponibili sul mercato.

DIDATTICA**137 Impariamo ad utilizzare Eagle**

Proseguiamo il nostro viaggio alla scoperta di EAGLE, un potente strumento di progettazione elettronica, che permette di coprire l'intero ciclo di sviluppo di una scheda elettronica. Quarta puntata.

DOMOTICA**115 Termostato ambiente sul Cloud****115****SOLARE****127 Lumino a energia solare****127****eLektor****131 Misuratore d'induttanze****131****DIDATTICA****137 Impariamo ad utilizzare Eagle****137**

Gli strumenti di sviluppo in un unico posto

Migliaia di strumenti da centinaia
di produttori fidati



Scegli tra i prodotti della
nostra vasta selezione su
mouser.it/dev-tools



MOUSER
ELECTRONICS



Il protocollo SENT

Fra i tanti protocolli di comunicazione dati esistenti e utilizzati prevalentemente nell'elettronica automobilistica, si colloca il SENT, meglio descritto dallo standard SAE J2716 SENT (Single Edge Nibble Transmission); nasce come alternativa ai classici LIN (Local Interconnection Network) e CAN (Controller Area Network) che sono di fatto vere e proprie reti dati. Il protocollo è stato pensato per realizzare interfacce di comunicazione molto semplici tipicamente dedicate ai sensori, quindi è un'alternativa al LIN e si colloca all'interno di quelli che riguardano connessioni fisiche a un solo filo come canale dati; il SENT è uno standard di comunicazione unidirezionale in cui i dati provenienti da un sensore vengono trasmessi in modo indipendente senza alcun intervento del dispositivo di ricezione dati (ad esempio l'MCU) che si limita a riceverli senza eseguire alcuno scambio.

Una tipica applicazione è quella nell'interfacciamento dei misuratori di massa d'aria digitali della Bosch o nei sensori di qualità e livello dell'olio prodotti dalla Hella, che supportano tale protocollo. Il protocollo a livello di dati supporta l'invio nella stessa stringa delle informazioni di due sensori affacciati sulla stessa interfaccia, ovvero di sensori combinati come ad esempio quelli di temperatura e pressione di un liquido, quelli di posizione angolare del piantone dello sterzo, di rilevamento della posizione dell'acceleratore, ed anche della misura della massa del flusso d'aria aspirata dal motore. Il SENT definisce la trasmissione di dati punto-punto da un sensore a un sistema di controllo che li acquisisce, quindi è tipicamente pensato per comunicazioni unidirezionali, dove un dispositivo invia dati continuamente a un

microcontrollore o microprocessore; il protocollo supporta la trasmissione di dati ad alta risoluzione con un costo complessivo del sistema inferiore a quello di altri link di comunicazione seriale già esistenti e maggiormente strutturati. Sul piano dell'interfaccia fisica (Physical Layer) il protocollo SENT prevede la trasmissione unidirezionale, seriale e asincrona, utilizzando un filo per il segnale digitale, con livelli logici basso corrispondente a una tensione $<0,5$ V e alto corrispondente a un potenziale superiore a $4,1$ V; l'interfaccia è a tre fili, comprendente una linea di alimentazione a 5 V e una massa comune.

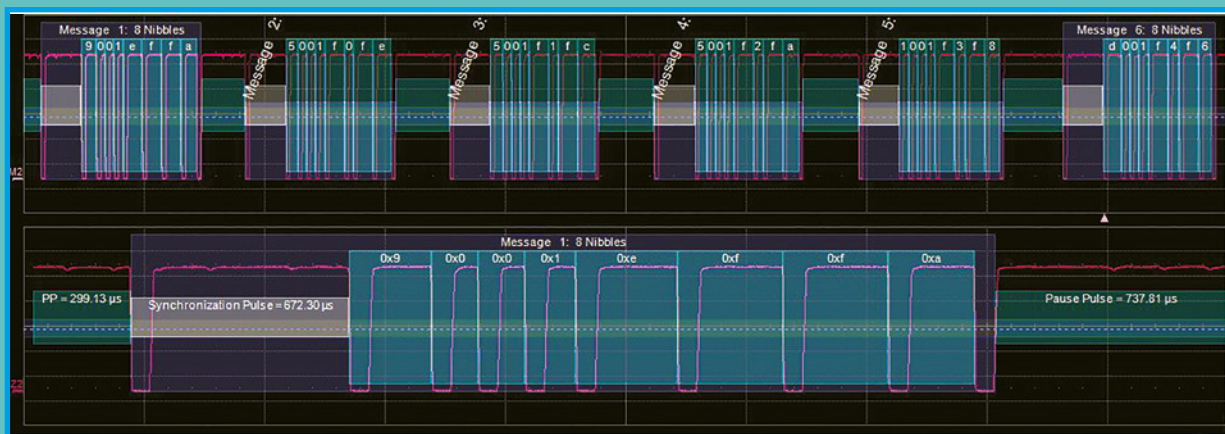
Sul piano dei dati, i dati vengono trasmessi dal dispositivo lungo il filo di segnale in nibble, ossia blocchi (parole) da 4 bit ciascuno, delimitati dall'intervallo tra due fronti di discesa. La distanza tra due fronti di discesa rappresenta il valore del nibble, che in formato decimale è compreso fra 0 e 15. Il tempo complessivo di trasmissione, ossia la durata totale della stringa, dipende dal valore dei dati trasmessi e dal valore della frequenza di clock del dispositivo che li trasmette.

Il tipico segnale trasmesso dal

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.

senore è costituito da una serie di impulsi, dove la distanza tra fronti di discesa consecutivi definisce la trasmissione di nibble di dati a 4 bit, il cui tempo di trasmissione totale dipende dai valori dei dati trasmessi e dalla frequenza di clock del trasmettitore. Nel protocollo SENT ogni stringa viene trasmessa consecutivamente alla precedente, dopo la fine della trasmissione di quest'ultima (il fronte di discesa del nibble del CRC fa anche da fronte di discesa iniziale della sincronizzazione/calibrazione consecutiva della trasmissione SENT successiva). Un'unità di tempo fondamentale della comunicazione SENT (UT, corrispondente al periodo nominale del clock del trasmettitore) può essere compresa tra 3 microsecondi e 10 microsecondi, secondo la specifica SAE J2716. Un messaggio o stringa secondo il protocollo SENT è lungo 32 bit (8 nibble) ed è composto dai seguenti elementi:

- impulso di calibrazione/sincronizzazione lungo 56 unità di tempo;
- fino a 6 nibble (24 bit di dati) che rappresentano il valore di 2 canali di misura da 3 nibble ciascuno (durata da 12 a 27 UT);
- 4 bit (1 nibble) per il codice di rilevamento di eventuali errori di comunicazione o CRC (da 12 a 27 UT);
- 4 bit (1 nibble) di informazioni di



↑ Struttura della comunicazione secondo il protocollo SENT, completa di dati, CRC e informazioni di stato.

stato sulla comunicazione (da 12 a 27 unità di tempo).

Opzionalmente, i dati possono essere trasferiti sotto forma di messaggi da 20 bit (5 nibble) composti da un singolo valore di misura a 12 bit (3 nibble), un codice di controllo degli errori CRC (1 nibble) e un campo a 4 bit (1 nibble) dedicato allo stato della comunicazione.

La struttura completa del protocollo SENT è esemplificata nell'oscillogramma proposto nella figura in alto in questa pagina; nel caso specifico è stato utilizzato un impulso di pausa (Pause Pulse). L'impulso di pausa serve a compensare la lunghezza variabile dei messaggi, ovvero mantiene una lunghezza di stringa costante. La variazione massima consentita del clock è del $\pm 20\%$ rispetto all'unità di tempo nominale, il che consente l'utilizzo di oscillatori RC di sistema a basso costo nel trasmettitore.

Editor di frame per LED Matrix

In questi anni abbiamo avuto modo di proporvi progetti di visualizzatori a matrici di LED RGB, dai più complessi basati su FPGA ai più maker-oriented gestiti da architetture compatibili con Arduino e programmabili con il relativo IDE. In tutti i casi abbiamo utilizzato degli editor di frame, che sono necessari per comporre le immagini da visualizzare sulle matrici e convertirle in un formato idoneo all'hardware utilizzato. Restando

in tema, qui vogliamo segnalarvi un interessante editor di frame per LED Matrix idoneo all'ambiente Arduino e ideale per sfruttare funzionalità e potenzialità della nuova nata di casa Arduino: la Uno R4 WiFi, dotata di interfaccia WiFi e di un potente core Arm® Cortex®-M4 a 32 bit, in grado di gestire immagini e animazioni.

L'editor si chiama ImageJ ed è un software gratuito e open source per l'elaborazione delle immagini sviluppato da Wayne Rasband. Il software è disponibile su un repository github raggiungibile all'indirizzo <https://github.com/mutterer/IJR4Matrix>.

Il programma è sostanzialmente uno script realizzato con ImageJ, uno strumento che consente l'elaborazione e l'analisi di immagine in Java (<https://imagej.nih.gov/ij/>).

Il progetto coinvolge anche la libreria Java JSSC per la comunicazione seriale con la porta destinata all'interfacciamento con la tipica matrice RGB (bus seriale).

Grazie al plugin Serial Macro Extensions, è possibile accedere alla comunicazione seriale tramite il linguaggio di programmazione macro di ImageJ. Inoltre, attraverso il plugin ActionBar, è possibile estendere l'interfaccia utente di ImageJ arricchendola con barre dei pulsanti personalizzabili. Il file di configurazione LED_Matrix.ijm è essenziale per rendere disponibili i pulsanti aggiuntivi nell'interfaccia. Nel repository trovate anche un esempio di file multipagina TIFF, Matrix.tif, utile per contenere i frame della matrice. Infine, con il firmware PreviewMatrix.ino, appositamente modificato dall'esempio "live preview" di

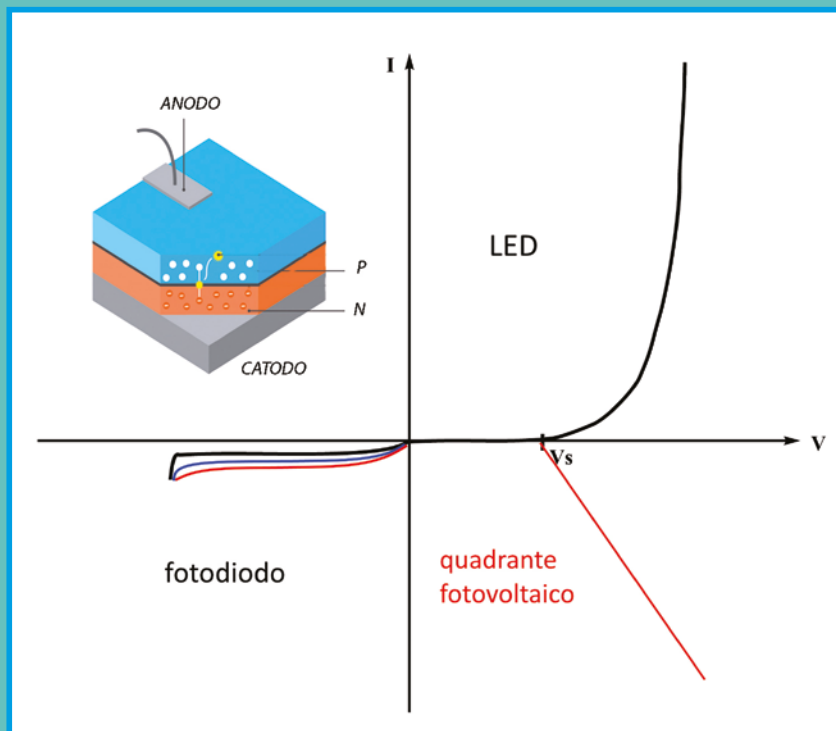
Arduino, è possibile sfruttare la board Arduino Uno R4 WiFi per aggiornare la matrice LED con dei set di byte.

Tutti i componenti software appena descritti sono disponibili sul repository github anzidetto e da esso possono essere scaricati liberamente.

LED: non solo luce

Contrariamente a quanto il suo nome potrebbe far pensare, il diodo luminoso non si limita a produrre luce per illuminare o segnalare una condizione quando viene alimentato; certo, se è polarizzato direttamente funziona da diodo e lavora nel primo quadrante della sua curva caratteristica, producendo una radiazione luminosa la cui colorazione dipende dagli elementi chimici che ne costituiscono la giunzione PN e che va dal rosso più scuro al violetto tanto più quanto maggiore è la tensione necessaria ad accenderlo.

Invece quando viene polarizzato inversamente ed ha il corpo trasparente, quasi sempre il LED può funzionare da fotodiode, ossia nel caso (quasi esclusivo) in cui il chip sia disposto con la regione P rivolta verso l'esterno, giacché tale regione è quella sensibile alla luce; in vero anche il LED girato al contrario rileva la luce, ma per riflessione sul cono ricavato nella parte interna dell'elettrodo inferiore. La risposta alle varie lunghezze d'onda dipende dalla luce che il LED emetterebbe polarizzato direttamente, giacché il lavoro di estrazione degli elettroni è tanto maggiore quanto più elevata è la lunghezza d'onda richiesta.



↑ Curva caratteristica del diodo che funziona da LED, fotodiodo e cella fotovoltaica.

Sempre se ha il corpo trasparente, il LED può anche funzionare da generatore elementare di elettricità, lavorando nel quadrante fotovoltaico e quindi funzionando da piccola cella solare, perché la luce viene emessa dal lato P che, similmente alle celle fotovoltaiche, è di modesto spessore e si fa attraversare da una discreta quantità di luce fino a raggiungere la giunzione PN e liberare elettroni. Ovviamente in questo caso non va alimentato, perché diventa esso stesso la fonte di elettricità.

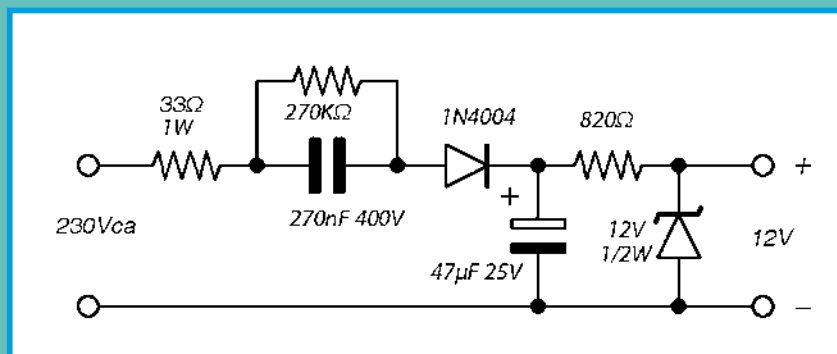
12 volt dalla rete senza trasformatore

Può capitare che per questioni di dimensioni del circuito o di costi, ma anche di interferenze, si voglia ottenere una tensione continua a partire da una alternata senza passare dal classico trasformatore con primario a 230V/50Hz. In casi del genere è possibile alimentare un circuito o un utilizzatore in continua che assorba pochi milliampere sfruttando la caduta di tensione su un'impedenza e raddrizzando poi la corrente a singola

semionda attraverso il classico diodo rettificatore a giunzione PN, quindi livellando la componente continua ottenuta mediante un condensatore elettrolitico. Questa tecnica viene utilizzata in molti dispositivi soprattutto laddove non serva l'isolamento galvanico dalla rete elettrica a 230 Vca e basti una modesta corrente di uscita.

Lo schema in basso a questa pagina propone un tipico esempio di alimentatore AC/DC senza trasformatore, dove l'impedenza che determina la caduta di tensione è principalmente quella dovuta alla reattanza capacitiva del condensatore da 270 nF; la resistenza che lo precede serve ad evitare un eccessivo picco di corrente nel diodo raddrizzatore che si verificherebbe se il circuito venisse alimentato (o collegato alla rete elettrica a 230Vca con un cordone di alimentazione) quando la sinusoide di rete è in corrispondenza del picco. Invece la resistenza da 270 kohm in parallelo al condensatore ha la funzione di scaricarlo quando il circuito viene sconnesso dalla rete, in modo da evitare che toccandop la spina o il circuito pur privo di alimentazione, si prenda la scossa. Il diodo 1N4004 raddrizza la corrente a singola semionda e gli impulsi costituiti dalle semionde positive della sinusoide di rete caricano l'elettrolitico a valle, ai capi del quale si trova un diodo Zener da 12V con la relativa resistenza zavorra (di caduta); tra anodo e catodo dello Zener si preleva la tensione stabilizzata di uscita. Volendo è possibile filtrare ulteriormente i 12 volt continui ottenuti ponendo in parallelo al diodo Zener un ulteriore condensatore elettrolitico da 100μF - 16V.

Ricordate che la massa (-) di questo circuito è collegata alla rete, quindi non va mai toccata, pena la folgorazione.



↑ Alimentatore AC/DC basato sulla caduta di tensione su un'impedenza .

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.elettronicaonline.it/webforum



ABBONAMENTO
ANNUALE



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



Visita: www.elettronica.in.it

6 NUMERI
A SOLI

€29

ANZICHÉ € 45,00

ABBONAMENTO ANNUALE PER RIVISTA DIGITALE

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

COME ABBONARSI

→ **on-line**

nella pagina "Riviste e Corsi Gratuiti"
del nostro sito www.elettronicaain.it

→ **e-mail**

scrivendo a
abbonati@elettronicaain.it

→ **telefono**

telefonando a
+39-0331-752668

NUOVO FORMATO 160 PAGINE CON PIÙ CONTENUTI

ABBONAMENTO ANNUALE PER RIVISTA **CARTACEA**

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

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

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

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

6
NUMERI
A SOLI

€39

ANZICHÉ € 45,00

Ad ogni batteria la sua ricarica!



PER BATTERIE NI-CD / NI-MH / LI-ION

CARICABATTERIE UNIVERSALE CON DISPLAY LCD PER BATTERIE NI-MH, NI-CD, LI-ION, LIFEPO4



Caricabatterie con display LCD in grado di caricare batterie Ni-MH, Ni-Cd, Li-Ion e LiFePO4 dei formati più diffusi in commercio. Sul display vengono visualizzate: tensione, corrente, stato di carica e tipo di batteria. Il caricabatterie è alimentato tramite una porta d'ingresso USB Type-C.

cod. UC2
€ **28,00**

PER BATTERIE NI-CD / NI-MH

CARICABATTERIE CON DISPLAY LCD PER BATTERIE NI-MH E DISPLAY

cod. TP203
€ **24,00**



Caricabatterie intelligente controllato da microcontrollore, dotato di display LCD retroilluminato e in grado di ricaricare 4 batterie Ni-MH tipo AA o AAA. Rileva le batterie danneggiate, non ricaricabili o inserite male. Dispone di timer di sicurezza che interrompe la ricarica dopo 6,5 ore. Sul display LCD viene visualizzato lo stato di carica di ogni singola batteria.

CARICABATTERIE A 4 POSTI PER BATTERIE AA/AAA/9V

Permette di caricare da 1 a 4 batterie (Ni-Cd e Ni-MH) formato AA, AAA e 9V. Dotato di LED operativi che forniscono indicazioni sullo stato di ricarica delle batterie.



cod. WT40902
€ **19,00**

PER BATTERIE AL PIOMBO

CARICABATTERIE INTELLIGENTE 6-12V CON CORRENTE DI CARICA MAX 4A

Per batterie tipo WET, AGM, GEL, VRLA, MF, Ca / Ca da 6V e 12V di tensione. Controllato da un microcontrollore esegue in totale autonomia: la diagnosi, la ricostruzione (rigenerazione), la carica, il mantenimento e verifica l'efficienza della batteria.



cod. TP274
€ **48,00**

CARICABATTERIE INTELLIGENTE 6-12V CON CORRENTE DI CARICA MAX 1A

Per batterie tipo WET, AGM, GEL, VRLA, MF, Ca / Ca da 6V e 12V di tensione. Controllato da un microcontrollore esegue in totale autonomia: la diagnosi, la ricostruzione (rigenerazione), la carica, il mantenimento e verifica l'efficienza della batteria.



cod. TP221
€ **19,50**

PER BATTERIE LI-ION

CARICABATTERIE A RICARICA RAPIDA CON CORRENTE DI CARICA MAX 2A

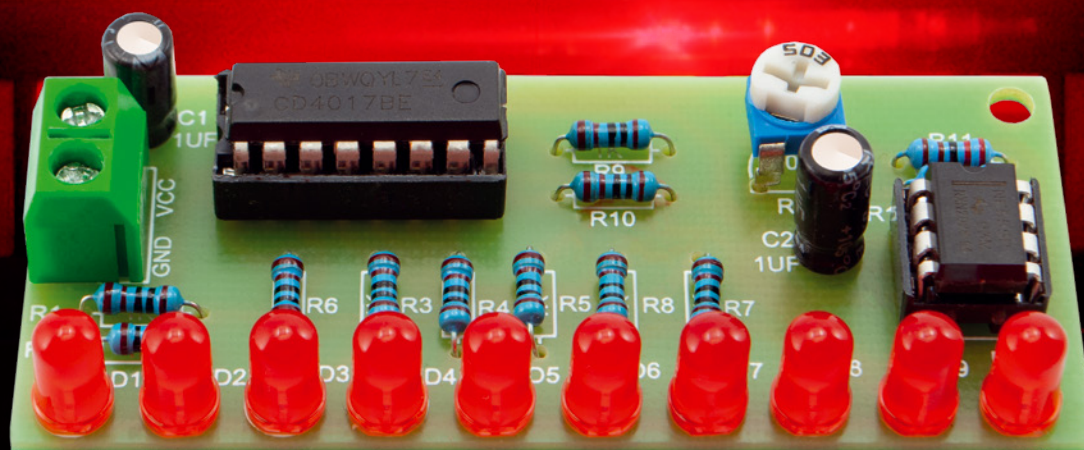
Compatibile con diverse tipologie di batterie come le 18650, 26650, 14500, 16340 e 18500. Fornisce una corrente di ricarica fino a 2A. Dispone di LED indicatori per monitorare lo stato di ricarica.



cod. LIONCH2
€ **12,00**

LED SEQUENZIALI

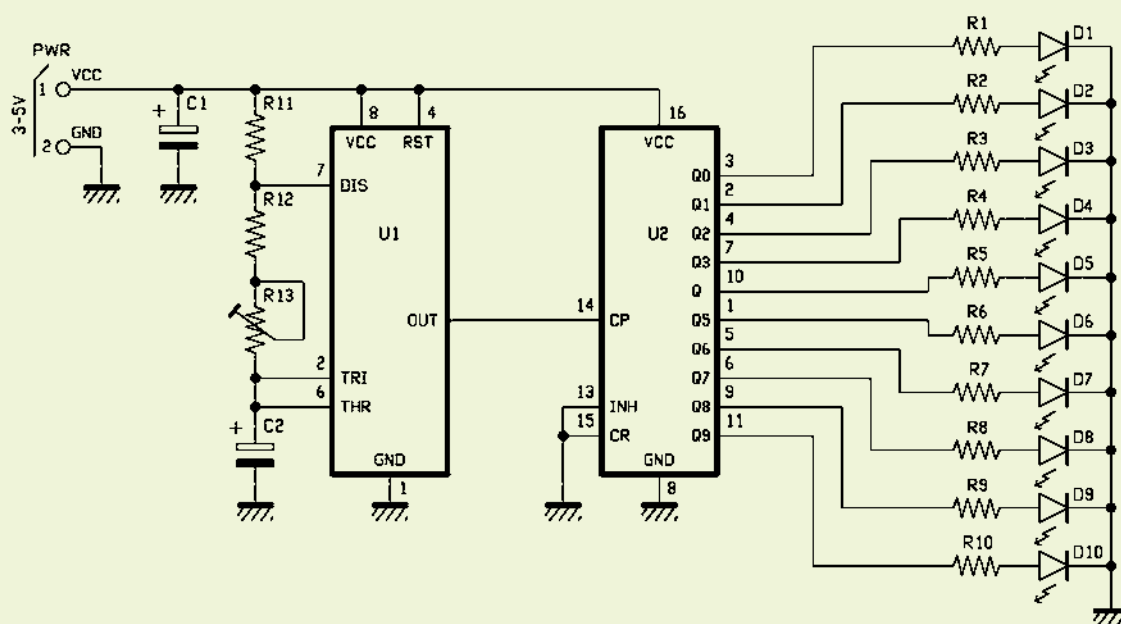
di DAVIDE SCULLINO



Un punto luminoso si sposta su una barra da un lato all'altro, ciclicamente, a una velocità impostabile tramite un trimmer.

S

ebbene viviamo nell'epoca dei microcontrollori e di Arduino utilizzato per fare qualsiasi cosa, anche con una manciata di tradizionali LED e due circuiti integrati tra i più datati della storia dell'elettronica allo stato solido, è possibile realizzare qualcosa di suggestivo; ve lo vogliamo dimostrare con il progetto proposto in queste pagine, che è in pratica un effetto luminoso consistente in un punto che si sposta ciclicamente dall'inizio alla fine di una fila di 10 LED, per poi ripartire dall'inizio, con una velocità impostabile attraverso un trimmer. Oltre che come puro e semplice gadget per attirare l'attenzione delle persone che guardano una vetrina o un banco, può servire da segnalatore per avvisare di un pericolo, indicare un accesso e via di seguito.



Il circuito è molto semplice e la sua semplicità, in particolare costruttiva, può anche essere l'occasione per esercitarsi nei montaggi elettronici a scopo didattico, per chi tra voi è "alle prime armi"; i componenti utilizzati sono di facile reperibilità, perché molto comuni: si tratta sostanzialmente di un timer 555 e di un contatore tipo CD4017. Entriamo dunque, senza indugio, nel vivo del progetto, analizzandone lo schema elettrico, che lo descrive meglio d'ogni parola e che ne palesa la grande semplicità.

SCHEMA ELETTRICO

Il nostro gadget è composto idealmente da tre blocchi, che sono un generatore di impulsi rettangolari, facente capo al timer 555 (che nello schema elettrico è siglato U1), un contatore decimale CMOS di tipo CD4017 e un visualizzatore composto da dieci LED, che sul circuito stampato sono stati disposti in linea a comporre una striscia. Per farvi comprendere come funziona il circuito dobbiamo per prima cosa spiegare come funziona il 555, che nel nostro gadget funziona da multivibratore astabile; l'integrato contiene due comparatori aventi in comune un partitore di tensione a scala di resistenze che polarizza, all'operazionale superiore l'ingresso invertente e a quello inferiore, il non-invertente; dei comparatori vengono resi accessibili dall'esterno gli ingressi non-invertenti di

quello superiore, corrispondente al piedino 6 (THR, ossia Threshold) e l'invertente di quello superiore, corrispondente al piedino 2 (TRI, ossia Trigger). Notate che per "superiore" intendiamo il comparatore che si trova in alto nello schema a blocchi interno proposto nella **Fig. 1** e per "inferiore" quello in basso; il comparatore superiore riceve dal partitore resistivo multiplo il potenziale di riferimento più alto e quello inferiore la tensione più bassa. Il nodo tra la prima e la seconda (considerate dall'alimentazione positiva) resistenza del partitore multiplo di riferimento dei due comparatori (alimentato, come il resto degli stadi interni al 555, dal piedino 8) viene portato all'esterno tramite il piedino 5 (CONT, che sta per Control Voltage) e ciò permette di alterare le tensioni di riferimento dei comparatori, così da controllare con una tensione esterna la frequenza di lavoro nella configurazione astabile (operando lo shift di frequenza come si fa nei VCO, ossia gli oscillatori controllati in tensione) oppure la durata degli impulsi in quella da timer (monostabile).

Più esattamente, nella configurazione astabile, abbassando la tensione sul piedino 5 si accorcia il periodo della commutazione dell'uscita e quindi si incrementa la frequenza generata, viceversa lasciando salire la tensione fino al livello caratteristico (a vuoto, ossia senza alcun circuito collegato al piedino 5) o incrementandola si allunga il periodo e

si riduce la frequenza prodotta. Nella configurazione monostabile, invece, la tensione altera la durata degli impulsi, il che permette, pilotando il trigger con impulsi a frequenza costante, di ottenere un modulatore PWM. Comunque nell'applicazione del timer 555 qui descritta, questa funzionalità non viene sfruttata e il piedino 5 (sebbene talvolta si preferisca collegarlo ad un condensatore di filtro) non è stato collegato (rimane fluttuante, ma la cosa non inficia il funzionamento dell'integrato). Le uscite dei comparatori entrano una nel reset (quella del comparatore superiore, che arriva su R) e nel SET (comparatore inferiore) di un flip-flop di tipo RS, che è un circuito logico la cui uscita diretta (Q) va a livello alto (corrispondente a circa il potenziale del piedino 8) quando il SET è a livello alto, ovvero assume lo zero logico se SET è posto a circa zero volt; l'uscita presenta un comportamento esattamente opposto quando ad essere stimolato è l'ingresso reset, il quale, posto a livello logico alto, resetta il flip-flop, intendendo con ciò che ne azzeri lo stato di Q. Il flip-flop ha anche un'uscita complementata ($\bar{Q}$), la cui condizione logica è sempre l'inverso di quella Q; nel 555 è collegata, tramite una resistenza, alla base di un transistor NPN configurato ad open collector il cui emettitore è connesso a massa e il collettore al piedino DIS (Discharge, 7). Il transistor può commutare una corrente di collettore di 200 mA, utilizzato in modo sink (ad assorbimento di corrente). L'uscita diretta, Q, è invece collegata al piedino OUT (3) dell'integrato

grato attraverso un buffer interno push-pull, capace di erogare un massimo di 200 mA. Il negativo di alimentazione, vale a dire la massa di riferimento del 555 corrisponde al piedino 1, mentre al 4 è connesso il reset generale (R1) a logica invertita del flip-flop: questo piedino permette di resettare forzatamente il circuito dall'esterno attraverso l'applicazione di un livello logico basso, ma nel nostro caso la relativa funzionalità è stata disattivata collegando il piedino 4 al positivo di alimentazione (piedino 8).

La massima frequenza di funzionamento del 555 configurato come stabile è 500 kHz, ma nel nostro caso, dovendo rendere visibile lo scorrimento del punto luminoso sui dei LED, operiamo a un valore nettamente più basso.

Ciò detto, vediamo che nel circuito il 555 è configurato come astabile, secondo il circuito canonico, che prevede una resistenza tra i piedini 4, 8 e 7, una tra 7 e 2,6 ed un condensatore tra questi ultimi e massa; nel nostro caso la resistenza tra i pin 4,8 e 7 è fissa ed è siglata R11 nello schema elettrico, mentre quella tra 7 e 2,6 è la serie di un resistore fisso (R12) e di un trimmer montato a reostato (R13) semifisso.

Il condensatore di temporizzazione collegato tra i piedini 2,6 e massa, è l'elettrolitico C2.

Per comprendere come funziona il multivibratore astabile ci appoggiamo allo schema a blocchi proposto nella **Fig. 1** e analizziamo il funzionamento del 555 immaginando di alimentare l'integrato

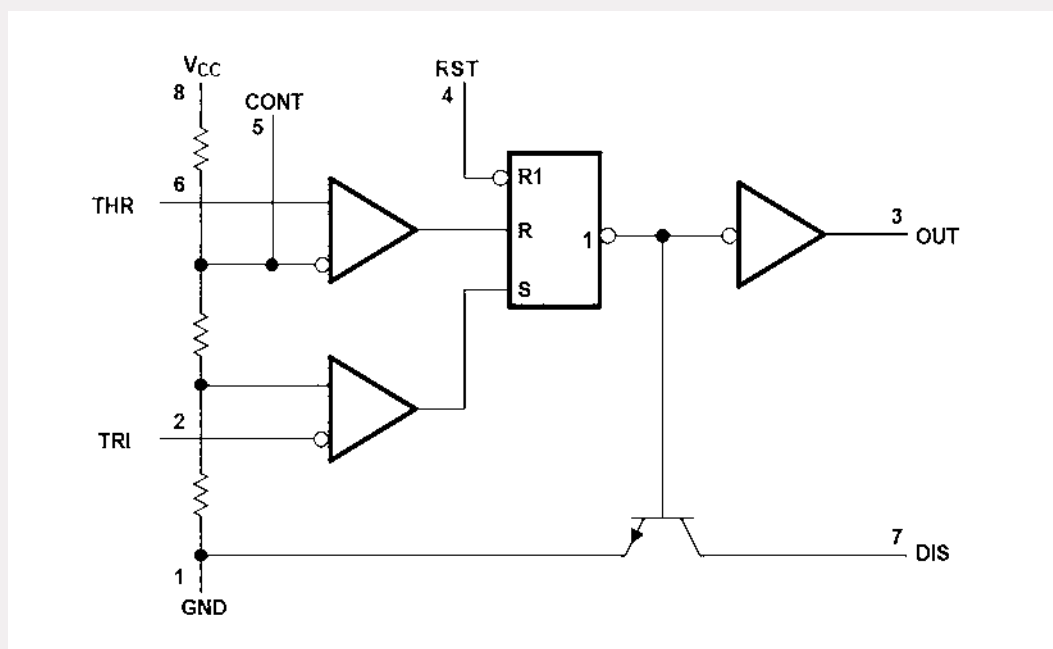
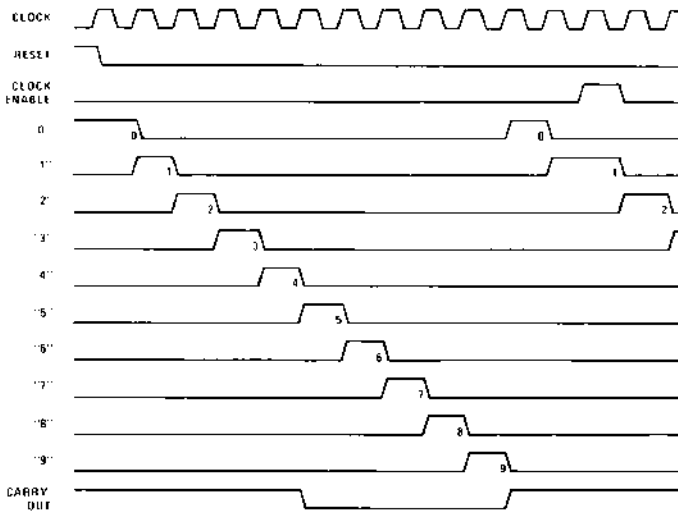


Fig. 1
Schema a
blocchi del
timer NE555.



lascia andare a livello alto) il SET del flip-flop, la cui uscita rimane comunque a livello alto fin quando la tensione ai capi di C2, continuando a crescere, non si porta a un valore superiore ai 2/3 della tensione di alimentazione del 555, ovvero alla soglia di commutazione del comparatore superiore. Quando ciò accade, l'uscita del comparatore superiore si porta a livello logico alto e resetta il flip-flop, la cui uscita diretta passa a zero logico mentre la /Q si porta a livello alto, mandando in saturazione il transistor interno collegato al piedino 7, il quale attraverso la serie R12-R13 scarica C2. La situazione rimane questa fin quando la tensione ai capi del C2 non scende al disotto della soglia di

commutazione del comparatore inferiore, allorché l'uscita di quest'ultimo torna a livello logico alto e setta il flip-flop, che riporta Q a 1 logico e /Q a zero. In questa condizione il condensatore riprende a caricarsi perché il piedino 7 torna nello stato aperto. La carica avviene nuovamente fino a 2/3 della tensione applicata tra i piedini 1 e 8 del 555, quindi il ciclo ricomincia.

Per effetto di ciò abbiamo l'uscita OUT dell'integrato che si mantiene a livello alto nel periodo che serve alla serie di condensatori affinché la tensione ai loro capi passi da 1/3 a 2/3 della tensione di alimentazione (siccome il 555 è alimentato a 9V, il passaggio è fra 3 e 6 volt, corrispondenti rispettivamente a 1/3 e 2/3 della tensione fra i pin 8 e 1) e a livello basso (zero volt circa) nel tempo che trascorre da quando i condensatori, in fase di scarica, vedono calare la tensione ai loro capi da 2/3 della tensione di alimentazione ad 1/3.

I tempi delle commutazioni dipendono dai valori delle resistenze R11, R12 ed R13 e sono tanto più simili quanto più la somma R12+R13 è piccola rispetto alla R11: infatti la carica avviene attraverso tutte e tre le resistenze mentre la scarica avviene solo attraverso R12 ed R13.

Ora va precisato che siccome R13 è un resistore variabile (dallo schema vedete un trimmer) il tempo di scarica del condensatore si può modificare entro certi limiti, giacché la resistenza complessiva vista in fase di scarica può variare da R12 (trimmer con il cursore cortocircuitato sui piedini 2-6) a R12+R13; questo permette di impostare la frequenza di lavoro dell'astabile tra un valore massimo (trimmer cortocircuitato) e uno minimo (trimmer completamente inserito). Variando la resistenza inserita dal trimmer si modifica anche il duty-cycle dell'onda rettangolare fornita dal piedino perché, la durata dell'impulso d'uscita a livello alto si determina come:

$$t_h = 0,693 (R_{11}+R_{12}+R_{13}) C_2$$

Tale durata coincide con la carica del condensatore. Invece la pausa dura un tempo pari a:

$$t_l = 0,693 \times (R_{12}+R_{13}) C_2$$

e coincide con la scarica dell'elettrolitico ad opera del piedino DIS.

La durata del periodo T è pari a $t_h + t_l$ e il duty-cycle vale:

$$dc = t_h/T$$

Se il cursore dell'R13 è spostato tutto verso R12, la durata dell'impulso è massima e pari a:

$$t_h = 0,693 (62,2 \text{ kohm}) 1 \mu F = 43 \text{ ms}$$

Invece la durata della pausa vale:

$$t_l = 0,693 \times 60 \text{ kohm} \times 1 \mu F = 41,58 \text{ ms}$$

In questo caso il periodo è di 84,58 ms, la frequenza (1/T) di 11 Hz e il duty-cycle vale:

$$dc = 43 \text{ ms}/84,58 \text{ ms} = 0,508$$

ossia circa il 51 %.

Se portiamo il cursore dell'R13 tutto verso il condensatore elettrolitico C2, la durata dell'impulso in uscita dal 555 vale:

$$t_h = 0,693 (22,2 \text{ kohm}) 1 \mu F = 8,45 \text{ ms}$$

mentre la pausa dura:

$$t_l = 0,693 \times 10 \text{ kohm} \times 1 \mu F = 6,93 \text{ ms}$$

In questo caso il periodo dura 15,38 ms e la frequenza sale a 65 Hz, mentre il duty-cycle vale:

$$dc = 8,45 \text{ ms} / 15,38 \text{ ms} = 0,549$$

quindi, in percentuale, il 54,9%.

Queste sono le caratteristiche del segnale rettangolare fornito dall'U1, che farà da clock al contatore U2, un CD4017 che utilizzeremo per trasformare la frequenza del 555 in un effetto luminoso. Vediamo dunque cosa accade nel contatore decimale, premettendo che tale integrato conta gli impulsi in ingresso a decadi, attivando di volta in volta l'uscita corrispondente al numero di impulsi di clock contati. Per comprendere bene come funziona potete fare riferimento allo schema logico proposto nella **Fig. 2** e nel grafico ad esso abbinato. Il 4017 dispone di 10 uscite, di peso crescente da 1 a 10, che sono nell'ordine Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q0. Partendo dall'istante del reset (che può essere forzato ponendo a 1 logico il piedino 15 oppure automatico all'accensione) ogni volta che viene rilevato uno stato logico alto (ovvero un passaggio 0/1 logico) da parte dell'ingresso di clock (CP) lo stato del contatore incrementa di 1, nel senso che si porta a livello alto l'uscita di peso decimale immediatamente successivo a quello dell'uscita attualmente attiva.

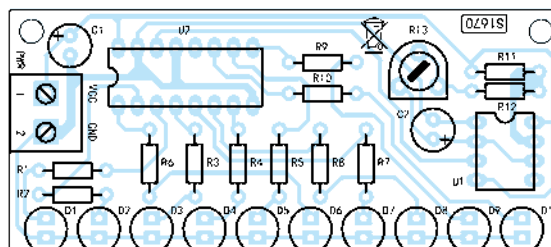
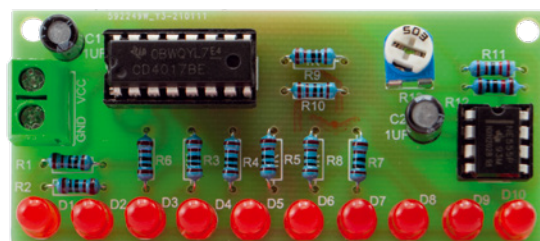


Elenco Componenti:

R1, R2, R3, R4, R5, R6, R7, R8, R9, R10: 1 kohm
 R11: 2,2 kohm 1%
 R12: 10 kohm 1%
 R13: Trimmer 50 kohm MO
 C1, C2: 1 μ F 50 VL elettrolitico
 U1: NE555
 U2: CD4017BE
 D1, D2, D3, D4, D5, D6, D7, D8, D9, D10: LED 5 mm rosso

Varie:

- Morsetto 2 vie 5 mm
- Zoccolo 4+4
- Zoccolo 8+8
- Circuito stampato S1670 (32x73 mm)



Quindi, partendo dalla condizione di contatore resettato, inizialmente è a livello alto l'uscita Q0; un primo impulso di clock porta ad 1 logico l'uscita Q1, un secondo pone a livello alto la Q2 e via di seguito, fin quando non si conta il decimo impulso, che attiva la Q0, oltre il quale il conteggio riprende da 1. Le uscite assumono lo stato alto una sola per volta. Il contatore dispone di un'ulteriore uscita, localizzata al piedino 12, chiamata Carry Out, la quale fornisce un impulso a livello logico alto quando il CD4017 riceve sul clock un'ulteriore impulso quando ad essere attiva è l'uscita decimale di peso 10 (Q0); tale uscita serve per triggerare un contatore in cascata, nei circuiti in cui serva contare impulsi secondo potenze di 10, quindi 100, 1.000 e via di seguito.

Per come abbiamo collegato il contatore, ossia con il piedino di reset (15) a massa (il che equivale a disattivare il reset esterno) il nostro CD4017 conta fino a dieci e viene azzerato automaticamente all'undicesimo impulso, per riprendere da capo il conteggio. Ad ogni impulso si porta a livello l'uscita di peso decimale superiore a quella attiva fino all'istante precedente, la quale torna a zero logico. Siccome ciascuna delle uscite Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8 e Q9 pilota un LED rosso attraverso una resistenza di limitazione della corrente (R1 per D1, R2 per D2 e così via fino ad R10) si accenderanno in sequenza ed uno solo alla volta, i 10 LED, dando la sensazione che un punto luminoso scorra su una barra da un lato all'altro, per poi riprendere una volta arrivato a fine percorso.

Concludiamo con l'alimentazione del circuito, che si alimenta tra i contatti PWR (Vcc è il positivo e GND il negativo) ai quali va applicata una tensione continua di valore compreso fra 3 e 5 volt, prelevata da una fonte in grado di erogare almeno 80 mA; il condensatore elettrolitico C1 filtra sia i disturbi eventualmente introdotti sulla linea di alimentazione sia l'eventuale ripple dell'alternata nel caso per far funzionare il gadget si utilizzi un alimentatore da rete, lineare (a trasformatore e diodi) o switching che sia.

REALIZZAZIONE PRATICA

Passiamo adesso dalla teoria alla pratica, con le note costruttive: per il gadget abbiamo previsto il montaggio su un circuito stampato, che abbiamo disegnato allo scopo e del quale rendiamo disponibile, nelle pagine di presentazione di questo numero sul sito www.elettronica.it, la traccia lato rame per ottenere la pellicola necessaria a procedere per fotoincisione. Data la natura prettamente didattica di questo progetto, abbiamo voluto disegnare uno stampato a singola faccia, perché è realizzabile da tutti con facilità, anche per tracciatura diretta con l'apposito pennarello.

Una volta inciso e forato il circuito stampato, dovete iniziare il montaggio dei componenti richiesti, i quali sono tutti a foro passante (THT=Through Hole Technology) e quindi di facile gestione da parte anche di chi ha poca esperienza ed un laboratorio con dotazione limitata al classico saldatore e al filo di stagno.

Una volta in possesso di tutto il necessario, iniziate il montaggio con le resistenze e i condensatori, prestando attenzione alla polarità dei due elettrolitici, quindi procedete con lo zoccolo dip 4+4 contatti per il 555 e con quello a 8+8 poli per il CD4017, cercando di allineare le tacche di riferimento come indicato nel piano di montaggio. Volendo è possibile saldare direttamente i due integrati, però bisogna fare attenzione a non surriscaldarli; invece se optate per il montaggio su zoccolo, inserite gli integrati una volta saldati tutti gli altri componenti.

Passate quindi ai 10 LED, ricordando che per il loro corretto orientamento, il catodo è l'elettrodo che si trova dal lato smussato del contenitore); completate il montaggio con la morsettiera bipolare a passo 5 mm per l'alimentazione (PWR). Per l'orientamento dei componenti polarizzati (quindi LED, elettrolitici e integrati) fate riferimento al piano di montaggio visibile nella pagina qui accanto.

Ora apriamo una parentesi sui LED, che nel progetto è previsto siano tondi da 5 mm di diametro: in realtà nulla vi vieta, in base all'applicazione desiderata, di sostituirli con LED triangolari o rettangolari (tanto elettricamente sono compatibili) ovvero con LED tondi ma da 3 mm; potete anche decidere di utilizzare dei diodi luminosi di altro colore, quindi non rossi ma arancioni, verdi, blu o come altro desiderate.

Comunque decidiate, una volta completato il montaggio e verificatane l'esattezza, il vostro circuito è pronto per l'uso, dato che non richiede alcuna operazione di taratura preliminare; è vero che dovete regolare il trimmer per variare a piacimento la velocità di scorrimento del punto luminoso, ma il gadget funziona qualunque sia la posizione del cursore del trimmer R13, al primo colpo, dopo l'accensione.

Per far funzionare il circuito fornitegli un'alimentazione continua a tensione di valore compreso fra 3 e 5 volt, quindi va benissimo utilizzare una pila a secco da 4,5V (o anche la serie di tre stilo montate in un apposito portatile formato AA) oppure un alimentatore di rete da 5Vcc. La corrente richiesta all'alimentatore è dell'ordine dei 50 milliampere, ma conviene che l'alimentatore stesso sia in grado di erogare qualcosa in più per far fronte ai picchi di assorbimento.

CONCLUSIONI

Bene, con questo abbiamo terminato la descrizione del nostro circuito a LED sequenziali; ricordiamo

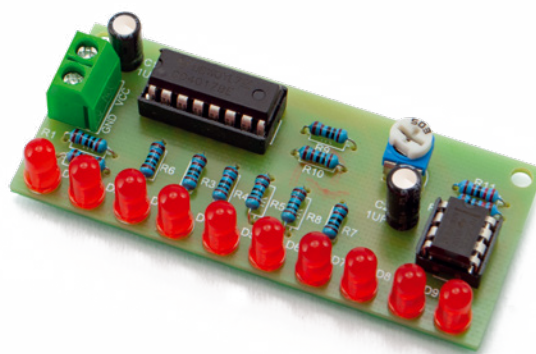


Fig. 3
Con il trimmer si regola la velocità di scorrimento del punto luminoso.

che oltre che come gadget per attirare l'attenzione su oggetti in vetrina o aree di un locale, può costituire una "spilla tecnologica" da portare sulla giacca o su un cappello ad una festa dove l'originalità è benvenuta. Il gadget può anche diventare parte di un travestimento.

Naturalmente il circuito qui proposto può trasformarsi da gadget in qualcosa di più funzionale, in quanto di fatto è un segnalatore ottico dinamico, che può prestarsi ad esempio a indicare una porta o un passaggio, orientandolo in modo che la direzione di "scorrimento" del punto luminoso sia tale da fornire l'indicazione. Può anche essere inglobato in una scatola trasparente da mettere a pavimento o in basso nelle pareti di un locale, sempre per indicare la direzione da seguire; ma è anche pensabile utilizzarlo come segnalatore di ingombro di un ponteggio o di strutture poste a lato della strada, ovvero la presenza di una curva o di un punto che richiede attenzione.

Insomma, per quanto semplice, il dispositivo che vi abbiamo proposto è utile ed utilizzabile davvero in moltissime applicazioni.

Buon lavoro e buon divertimento, dunque! ■



Cosa occorre?

Il kit dei LED sequenziali (cod. LEDSEQUENCE) è in vendita a Euro 5,90 IVA compresa.

Il materiale va richiesto a:

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

È disponibile il nuovo e-book Teoria e pratica del Timer555

L'e-book "Teoria e pratica del Timer555" approfondisce a 360 gradi un circuito integrato senza tempo che, a distanza di oltre 50 anni dalla sua uscita sul mercato, continua ad essere un elemento utilissimo per la progettazione di piccoli circuiti temporizzatori.

00:01

Acquistalo subito!

A SOLO
€ 7,90

cod. EBOOK236



Prezzo IVA inclusa.

ALCOL TESTER

a cura della REDAZIONE

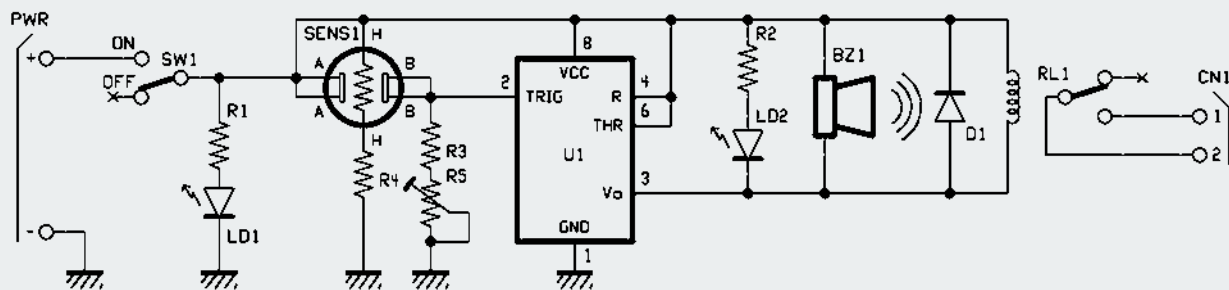
Consente di capire e avvisare acusticamente e visivamente se la concentrazione di alcol etilico nel fiato è oltre la soglia tollerabile, attivando eventualmente un relé.



Q

uando si dice "etilometro" il pensiero corre subito ai controlli stradali fatti dalle Forze dell'Ordine per identificare e sanzionare chi guida in stato di ebbrezza; in quelle occasioni viene utilizzato uno strumento dotato di un boccaglio intercambiabile (per ovvie ragioni di igiene) dove la persona sotto test soffia

e il fiato viene diretto contro la superficie sensibile di un sensore di alcol etilico. Qualcosa del genere vogliamo proporvelo in queste pagine, ma non perché andiate a ingrossare le fila dei controlli stradali né a giocare ai "poliziotti in erba" quanto piuttosto perché può sempre essere utile per verificare, dopo essersi alzati da tavola e prima di prendere l'automobile, di



non rischiare di essere in infrazione. Infatti, al pari degli alcol test disponibili in commercio per tale scopo, consente di misurare il tasso alcolemico (ossia la quantità specifica di alcol nel sangue) del soffio di fiato diretto contro la superficie del sensore utilizzato. Quest'ultimo è un dispositivo siglato MQ-3, capace di rilevare non solo i vapori degli alcoli ma anche vari tipi di gas e idrocarburi, seppure il composto in aerosol cui è più sensibile è certamente l'alcol, cui segue la benzina; lo utilizzeremo nel circuito descritto qui di seguito, analizzandone lo schema elettrico. Prima, però, facciamo il punto sul sensore, spiegando come funziona.

IL SENSORE MQ-3

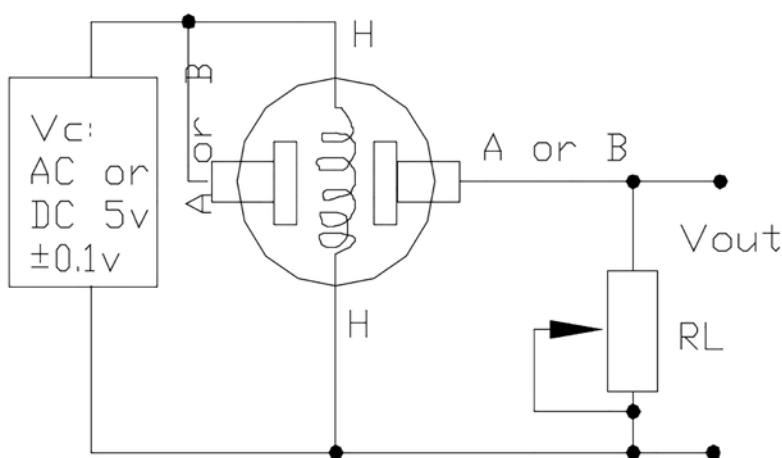
Questo dispositivo è prodotto dalla Hanwei Electronics (www.hwsensor.com) e si basa sulla collaudata struttura a film di ossido metallico, da tempo utilizzata con successo per rilevare i gas: l'elemento sensibile è uno strato di ossido che viene riscaldato da un filamento alimentato attraverso due piedini dedicati (siglati H come heater) così da portarsi alla temperatura ottimale per il suo funzionamento; il filamento ha una resistenza

a freddo (temperatura ambiente) di circa 33 ohm, è fatto di una lega nichel-cromo e non ha polarità, ma va preferibilmente alimentato in corrente continua, almeno per i nostri scopi. Il film metallico è applicato a un supporto ceramico di forma tubolare composto da allumina (Al₂O₃) che è un materiale elettricamente isolante; è fatto di biossido di stagno (SnO₂) e la sua caratteristica è che, quando viene riscaldato, diventa permeabile agli atomi dei gas nei confronti dei quali è sensibile.

Il contatto con l'alcol etilico e la reazione di ossidazione che ne deriva alterano la conducibilità del film di biossido di stagno, che in condizioni normali è bassissima: infatti la resistenza misurabile a vuoto è dell'ordine dei kohm. Per l'esattezza, siamo nell'ordine di qualche centinaio di ohm, fino a 20 kohm per concentrazioni di alcol etilico che variano fra 10 e 0,05 mg/L, corrispondenti al range misurabile dal sensore (più alta è la concentrazione minore è la resistenza e viceversa); la resistenza di riferimento fornita dal costruttore per una concentrazione di alcol etilico di 0,4 mg/L è definita R₀. La reazione di ossidazione innescata dall'alcol (tenendo conto del tenore di ossigeno nell'aria circostante) determina la cessione di elettroni che aumentano la conducibilità dell'elemento sensibile e quindi l'abbassamento della resistenza.

Per rilevare la variazione di resistenza del film si inserisce quest'ultimo in un partitore resistivo alimentato a 5V e avente come resistenza di carico (quella terminante alla massa del circuito di misura) tipicamente 20 kohm come mostrato nella **Fig. 1**, anche se nel nostro caso stiamo intorno agli 1,2 kohm perché il circuito lo richiede; all'aumentare della concentrazione dell'alcol nell'aria spinta oltre la griglia di protezione dell'MQ-3, la resistenza del filamento diminuisce e di conseguenza cresce la tensione rilevabile ai capi della resistenza di carico. Per avere un'idea del comportamento del sensore, ovvero del suo film sottile di biossido di stagno, nella **Fig. 2** proponiamo il grafico su scala

Fig. 1
Tipico schema applicativo del sensore.



logaritmica della risposta a vari tipi di gas e chiaramente all'alcol etilico.

Le curve si riferiscono a 20 °C di temperatura ambiente e 65% di umidità relativa, con una concentrazione di ossigeno dell'ordine del 21%.

Nel grafico, R_0 è la resistenza del sensore a una concentrazione di alcol etilico pari a 0,4mg/L in aria pulita e secca; invece R_s è la resistenza assunta dall'elemento sensore in corrispondenza di una certa concentrazione di composto nell'aria, che nel caso specifico è l'aria. Vedete che in corrispondenza di 0,4 mg/L di concentrazione R_s vale quanto R_0 e quindi il rapporto è pari ad 1; a 10 mg/L R_s diventa praticamente 1/10 della R_0 . Il rapporto R_s/R_0 determina la sensibilità dell'MQ-3.

Dal grafico risulta evidente che l'elemento sensibile del sensore MQ-3 risente anche della presenza di altri aerosol e gas, perché il film di biossido di stagno ha tale comportamento, sebbene nel caso del sensore da noi utilizzato sia rilevante il comportamento nei confronti dell'alcol etilico (il rapporto varia di circa 20 volte nel range di misura).

SCHEMA ELETTRICO

Bene, stabilito che cos'è e come funziona il sensore, possiamo vedere come l'MQ-3 è stato impiegato nel circuito dell'alcol test proposto in queste pagine: guardando lo schema elettrico vedete che adottiamo la configurazione suaccennata dove la resistenza dell'elemento sensibile è posta in serie a una resistenza con la quale fa partitore; per aiutarvi a seguire lo schema proponiamo nella **Fig. 3** la piedinatura e le connessioni all'interno dell'MQ-3.

Siccome il sensore è analogico, tra la serie di resistenze R_3 ed R_5 otterremo una tensione che varia secondo il grafico di **Fig. 2** in funzione della concentrazione di alcol etilico; essendo R_5 un trimmer, è possibile variare l'ampiezza della tensione ottenuta, ovvero il rapporto tra la differenza di potenziale presentata al piedino 2 dell'integrato U1 e massa e la tensione di alimentazione del filamento, che è di 5 volt in continua. Conformemente alle prescrizioni del produttore, in serie al filamento riscaldatore abbiamo posto un resistore che ne limiti la corrente e con essa la potenza dissipata, restando in valori di sicurezza e tali da garantire sia la giusta risposta all'alcol, sia la durata nel tempo. Il segnale fornito dall'elemento sensibile e quindi la tensione ai capi della serie R_3 - R_5 lo utilizziamo in maniera abbastanza inconsueta: niente lettura da parte del DAC di un microcontrollore o roba simile, ma molto più banalmente lo inviamo all'ingresso di

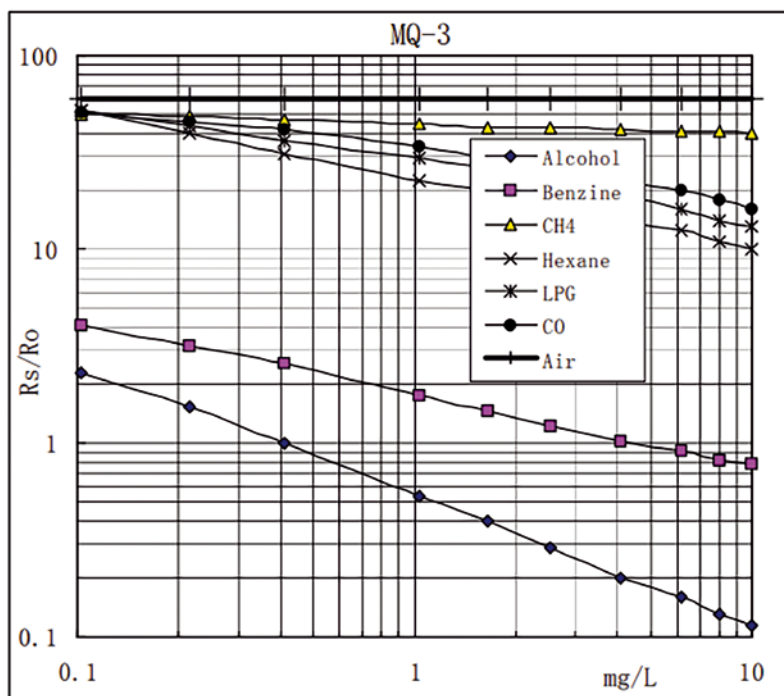


Fig. 2
Risposta del sensore MQ-3 a vari tipi di aerosol e gas

Fig. 3
Piedinatura e connessioni interne del sensore MQ-3.

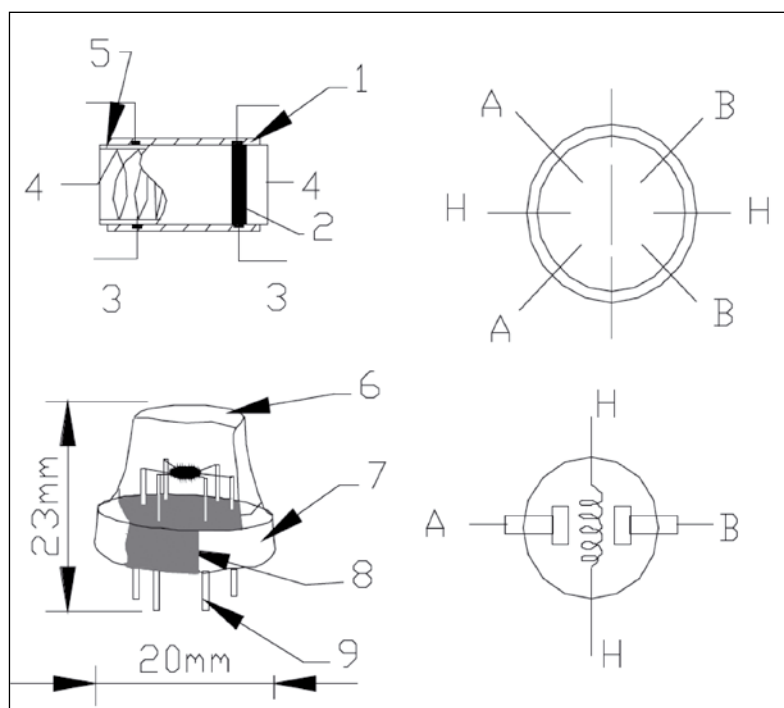
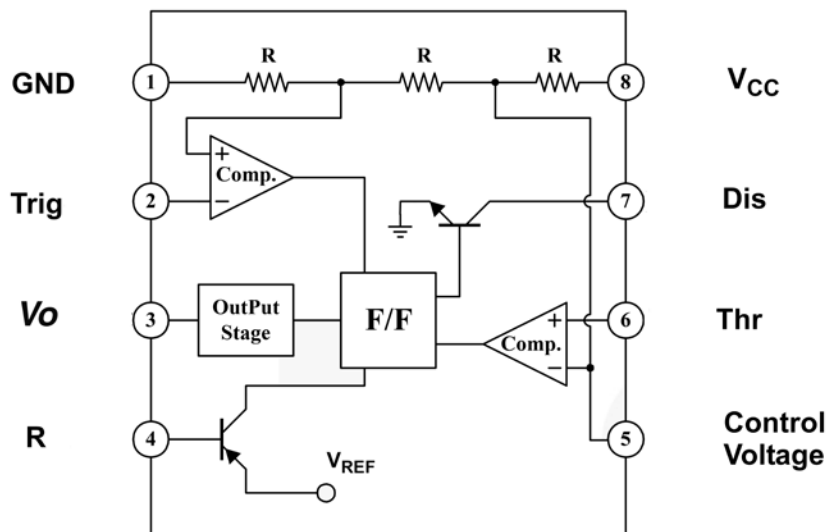


Fig. 4
Schema interno
del timer 555.



trigger di un comune timer 555 del quale sfruttiamo i flip-flop interni.

Per capire quale ruolo svolge il 555 dobbiamo vedere cosa c'è al suo interno, aiutandoci con lo schema a blocchi proposto nella **Fig. 4**: nel 555 troviamo due comparatori aventi in comune un partitore di tensione a scala di resistenze che polarizza per l'operazionale superiore l'ingresso invertente e per quello inferiore, il non-invertente; dei comparatori vengono resi accessibili dall'esterno gli ingressi non-invertente di quello superiore (i termini superiore e inferiore si riferiscono al potenziale di riferimento che ricevono dal partitore multiplo, quindi il comparatore superiore riceve la tensione più elevata), corrispondente al piedino 6 (THR, ossia Threshold) e l'invertente di quello superiore, corrispondente al piedino 2 (TRI, ossia Trigger). Il nodo tra la prima e la seconda resistenza (a partire dal positivo di alimentazione, ovvero dal piedino 8) del partitore multiplo di riferimento dei due comparatori viene portato all'esterno tramite il piedino 5 (CV, Control Voltage) e ciò permette di alterare le tensioni di riferimento dei comparatori, così da controllare con una tensione esterna la frequenza di lavoro nella configurazione astabile (operando lo shift di frequenza come si fa nei VCO, ossia gli oscillatori controllati in tensione) oppure la durata degli impulsi in quella da timer (monostabile). Ma questa funzionalità a noi non interessa per il circuito che stiamo descrivendo.

Quel che ci interessa è quanto viene dopo: le uscite dei comparatori pilotano una il RESET (quella del

comparatore superiore) e l'altra l'ingresso di SET (comparatore inferiore) di un flip-flop di tipo RS, che è un circuito logico la cui uscita diretta (Q) va a livello alto (corrispondente a circa il potenziale del piedino 8) quando il SET è a livello alto, ovvero assume lo zero logico se SET è posto a circa zero volt; l'uscita presenta un comportamento esattamente opposto quando ad essere attivato è l'ingresso RESET, il quale, quando è posto a livello logico alto, resetta il flip-flop, portandone a zero lo stato logico dell'uscita diretta Q.

Il flip-flop ha un'uscita complementata (/Q), la cui condizione logica è sempre l'inverso di quella Q; nel 555 pilota la base di un transistor NPN configurato in modo open collector, il cui emettitore è connesso a massa e il collettore al piedino DIS (Discharge, 7) che in questa applicazione non utilizziamo. L'uscita diretta, Q, è collegata al piedino OUT (3) dell'integrato attraverso un buffer interno push-pull, capace di erogare un massimo di 200 mA. Il negativo di alimentazione, vale a dire la massa di riferimento del 555 corrisponde al piedino 1, mentre al 4 è connesso il reset a logica invertita del flip-flop: questo piedino permette di resettare forzatamente il circuito dall'esterno applicando un livello logico basso, ma lo abbiamo posto fisso a livello alto perché non ci interessa.

Nella nostra applicazione abbiamo connesso al positivo di alimentazione il piedino 6, cosicché il comparatore superiore non potrà commutare e la sua uscita rimarrà costantemente a livello alto; con essa il reset del flip-flop. Quindi utilizzeremo il

solo comparatore inferiore, che fa capo al piedino di trigger; ogni volta che il piedino 2 viene portato a livello logico basso, l'uscita (pin 3) del 555 si porta a livello logico alto; quindi fin quando la concentrazione di alcol è bassa e la tensione ai capi della serie di resistenze è al di sotto del valore di 1/3 di quella di alimentazione del 555, l'uscita di quest'ultimo rimane a livello alto, mentre se la concentrazione sale abbastanza da portare la tensione tra il pin 2 e massa oltre 1/3 della tensione d'alimentazione, il comparatore inferiore porta la propria uscita a livello basso e il set del flip-flop torna a livello basso. In questa condizione prevale lo stato del reset e l'uscita diretta del flip-flop si porta a livello basso; tale condizione permette l'alimentazione del cicalino piezo e del LED di positività al test siglato LD2, polarizzato attraverso la resistenza di limitazione della corrente R2.

Ma non solo: il piedino 3 del 555 alimenta anche la bobina del relé RL1, che abbiamo volutamente inserito nel circuito per consentire a chi tra voi lo volesse, di utilizzare il circuito anche in abbinamento ad una centralina di allarme, per rilevare la presenza di gas come GPL e altri GAS rilevabili dal sensore, ma anche aerosol di benzina e alcoli. Per proteggere il transistor d'uscita del 555 dalle extratensioni inverse che si verificano ai capi della bobina del relé quando viene privata dell'alimentazione, abbiamo previsto il diodo D1, collegato in antiparallelo ad essa. Il relé è utilissimo se si intende utilizzare il circuito come rivelatore e allarme gas, tuttavia bisogna considerare che occorre utilizzare un relé a tenuta stagna, giacché la pur piccola scintilla scaturita dalla commutazione sul carico potrebbe innescare un'esplosione. Del RL1 rendiamo disponibile lo scambio limitatamente ai contatti C ed NO, in modo da poterlo utilizzare come interruttore.

Concludiamo la descrizione dello schema elettrico con l'alimentazione, che deve essere a tensione continua massima di 5V stabilizzati e va applicata al plug di alimentazione rispettando la polarità indicata; per accendere e spegnere il circuito abbiamo previsto un deviatore utilizzato come interruttore e siglato SW1, sul cui cursore si prende l'alimentazione per il sensore MQ-3, per il 555 e per il resto del circuito, bobina del relé (a 5V) compresa.

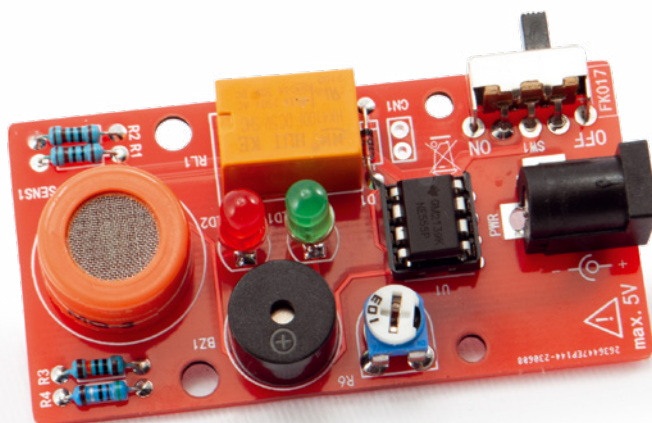
Tra la linea positiva e massa, all'uscita del deviatore, si trova il LED LD1, alimentato attraverso la resistenza R1, che funziona da spia di funzionamento del circuito.

REALIZZAZIONE E UTILIZZO

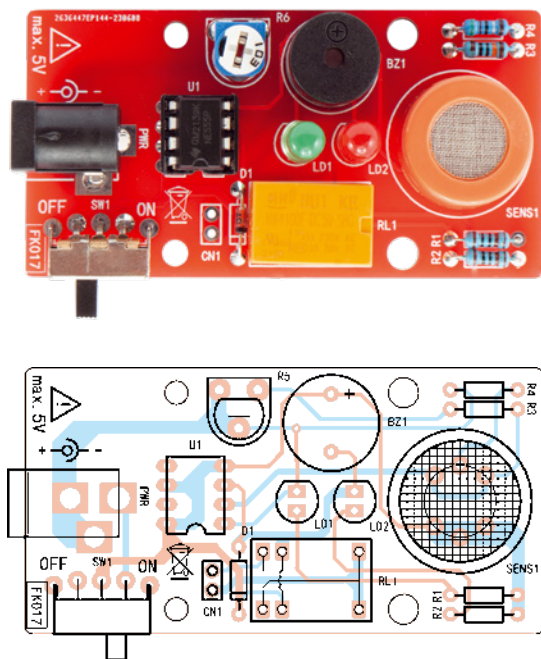
Bene, ora che abbiamo descritto il funzionamento del circuito possiamo vedere come realizzarlo, partendo dal circuito stampato da noi previsto per ospitare tutti i componenti occorrenti, sensore compreso; il PCB è a doppia faccia e allo scopo di realizzarlo per fotoincisione pubblichiamo, sul nostro sito www.elettronica.in.it, le due tracce lato rame da stampare su carta da lucido o acetato per ottenere le pellicole del caso. Inciso e forato il circuito stampato, potete montarvi i componenti a partire da quelli a più basso profilo come resistente e diodo al silicio (da orientare come indicato nel piano di montaggio visibile in queste pagine) per poi proseguire con lo zoccolo per il 555, il trimmer orizzontale e l'interruttore a slitta con piedini ad angolo retto. Si procede con il jack DC di alimentazione, il sensore MQ-3, il relé miniatura a 5V, il cicalino e i due LED. Prestate attenzione alla polarità dei due LED, i cui catodi (corrispondenti allo smusso sul contenitore) devono essere rivolti al cicalino piezo BZ1; quest'ultimo ha una polarità da rispettare, quindi nell'inserirlo nei rispettivi fori

CARATTERISTICHE TECNICHE

- ◆ Tensione di alimentazione: 5Vcc
- ◆ Corrente max assorbita: 250 mA
- ◆ Sensibilità all'alcol: 0,05÷10 mg/L
- ◆ Avviso acustico e visivo
- ◆ Sensibilità regolabile tramite trimmer



Il prototipo montato e pronto per l'uso.



Elenco Componenti:

R1, R2: 1 kohm	SENS1: MQ-3
R3: 130 ohm	SW1: Deviatore a slitta
R4: 5,1 ohm	RL1: Relè 5V
R5: Trimmer 10 kohm MO	
LD1: LED 5 mm verde	Varie:
LD2: LED 5 mm rosso	- Zoccolo 4+4
D1: 1N4148	- Plug DC
U1: NE555	- Circuito stampato FK017 (69x35 mm)
BZ1: Buzzer 5V	

dello stampato tenetelo con il + rivolto al lato della basetta lungo cui va posizionato. Le foto del prototipo e il piano di montaggio che trovate in queste pagine sono comunque d'aiuto nelle varie fasi del montaggio.

Per quanto riguarda l'utilizzo, serve un alimentatore da rete con uscita stabilizzata a 5Vcc, capace di erogare 250 milliampere di corrente; è anche possibile l'alimentazione a batteria utilizzando un pacco batteria con regolatore a 5V o meglio ancora un power bank in grado di fornire una tensione di 5V e sufficiente corrente per avere una buona autonomia, considerando l'assorbimento del circuito, che anche a riposo si attesta sui 170 mA a causa dell'assorbimento del filamento riscaldatore del sensore MQ-3.

Per utilizzare il prodotto, si deve come prima cosa portare l'interruttore su ON, dopo di che sarà probabile sentire il buzzer suonare perché verosimilmente occorrerà regolare il trimmer così

da impostare la soglia di allarme, vale a dire la sensibilità del nostro alcol test. A questo punto, se nell'aria non è presente alcol, ruotate il trimmer in senso antiorario per poi rilasciarlo nel momento in cui non si sentirà più suonare il buzzer. Per una corretta calibrazione sarebbe buona cosa lasciare il dispositivo alimentato per qualche minuto (almeno 5 minuti) con il trimmer al minimo e procedere solo dopo alla calibrazione, perché a quel punto il sensore, ovvero il suo film sensibile, si sarà portato alla temperatura di regime e il suo funzionamento risulterà stabile.

Una volta effettuata la calibrazione si potrà usare il dispositivo. In caso di presenza di alcol, il LED rosso si accenderà, il buzzer suonerà e il relè si ecciterà comportando la chiusura del contatto sul connettore CN1.

È opportuno precisare che non si tratta di un etilometro perché il nostro circuito non compie misure: è un alcol tester, vale a dire un qualcosa che avvisa se la concentrazione di alcol è sopra la soglia impostata; per renderlo allineato ai dispositivi utilizzati prima di mettersi alla guida, sarebbe opportuno compararlo con uno di questi e tarare il trimmer dopo aver alitato in esso, verificando quando segnala l'eccesso di alcol.

In alternativa bisogna riferirsi al grafico della risposta proposto nella **Fig. 2** e registrare il trimmer in modo che il LED LD2 si accenda in corrispondenza della concentrazione di alcol etilico desiderata; per esempio, potete tararlo deponendo un po' d'alcol etilico denaturato sul piano di un tavolo e affacciandogli a qualche centimetro di distanza il sensore con il circuito alimentato: regolate il trimmer affinché il cicalino inizi a suonare e il LED LD2 si accenda, verificando che una volta evaporato l'alcol tutto torni a riposo.

Nel deporre l'alcol state attenti a non spruzzarlo su corpi caldi o circuiti ad alta tensione, perché si incendierebbe. □



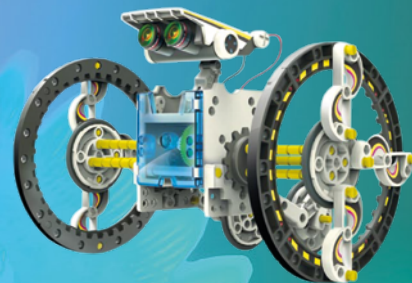
Cosa occorre?

L'alcol tester con segnalazione a LED e buzzer sonoro (cod. FK017) è in vendita a Euro 9,00. Il sensore MQ-3 (cod. MQ3) è disponibile anche separatamente a Euro 4,50. Per alimentare l'alcol tester in auto è possibile utilizzare il cavo USB con spina DC 5,5x2,1 da 1 mt (cod.TOL08639) disponibile a Euro 3,00.

Il materiale va richiesto a:

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

SERVOMOTORI, INDISPENSABILI PER GLI APPASSIONATI DI ROBOTICA!



INGRANAGGI IN METALLO

cod. MC946R

€ 8,50



SERVO ANALOGICO AD ALTE PRESTAZIONI CON ROTAZIONE DI 180°

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

cod. SERVO213

€ 11,00



SERVO ANALOGICO CON ROTAZIONE DI 180°

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

cod. MC996R

€ 9,00



SERVO ANALOGICO CON ROTAZIONE DI 360°

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

cod. SERVO860M9C

€ 9,00



MINI SERVO ANALOGICO CON ROTAZIONE CONTINUA DI 360°

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

cod. SERVO09METAL

€ 9,00



MINI SERVO ANALOGICO CON ROTAZIONE DI 180°

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

INGRANAGGI IN PLASTICA

cod. SERVO214

€ 8,50



SERVO ANALOGICO CON ROTAZIONE DI 120°

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

cod. SERVO860LC

€ 9,90



SERVO ANALOGICO CON ROTAZIONE CONTINUA DI 360°

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

cod. SERVO209

€ 8,00



SERVO ANALOGICO CON ROTAZIONE DI 180°

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

cod. SERVO207

€ 9,90



MINI SERVO ANALOGICO CON ROTAZIONE DI 180°

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

cod. SERVO206

€ 7,50



MINI SERVO ANALOGICO CON ROTAZIONE DI 180°

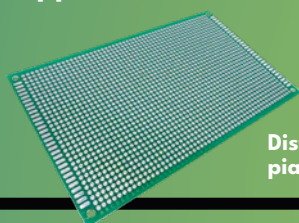
- Coppia di torsione: 1,2kg/cm (4,8V) - 1,5kg/cm (6V)
- Velocità a 4,8V: 0,12sec/60°, velocità a 6V: 0,11sec/60°
- Alimentazione: da 4,8 a 6V
- Dimensioni: 22,5x12x29mm
- Peso: 9g

Piastre sperimentali

Ideali per creare velocemente un circuito sperimentale per effettuare test, senza dover necessariamente realizzare un PCB.

MILLEFORI

Piastra Millefori 150×90 mm
doppia faccia



cod. YK653

€ 4,50

Distanza tra piazzole: 2,54 mm

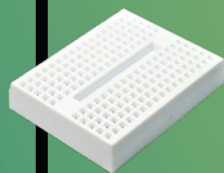
CODICE	MISURA (mm)	FACCIA	DISTANZA PIAZZOLE (mm)	PREZZO (€)
YK604	150×90	singola	2,54	3,80
YK647	100×50	singola	2,54	2,60
YK009	70×30	doppia	2,54	1,50
HR2570	120×80	doppia	-	4,50

BREADBOARD

Breadbord 170 contatti
colore bianco
45×35 mm

cod. BBMINIW

€ 2,50



● ROSSO

cod. BBMINIR

● NERO

cod. BBMININ

● BLU

cod. BBMINIB

● VERDE

cod. BBMINIV

Breadbord con base
autoadesiva 400 contatti
82×55 mm

cod. BB400

€ 4,00



Breadboard 1660 contatti
con base in metallo
e boccole

Misura piastra:
164×109×9,5 mm

Misura base:
215×131×1 mm

cod. YJ319

€ 14,00



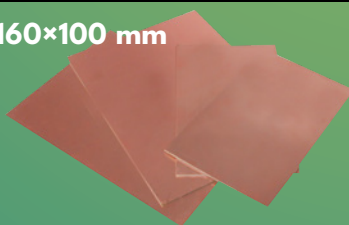
COD.	MISURA (mm)	N. CONT.	PREZZO (€)	COD.	MISURA PIASTRA (mm)	MISURA BASE (mm)	N. CONT.	PREZZO (€)
BB830	165×55	830	5,50	BB3220	188×182×9,5	240×210	3220	21,00

RAMATE

Piastra Ramata 160×100 mm
singola faccia

cod. P-951

€ 2,50



CODICE	MISURA (mm)	FACCIA	PREZZO (€)
P-952	200×150	singola	4,20
P-956	160×100	doppia	5,40
P-957	200×150	doppia	6,20

PRESENSIBILIZZATE

Piastra Presensibilizzata
160×100 mm singola faccia

cod. 4062

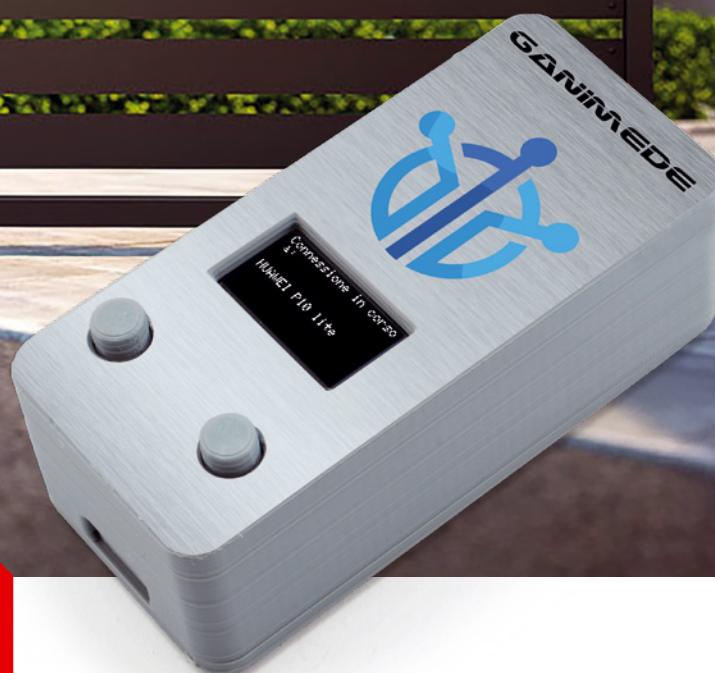
€ 6,00



CODICE	MISURA (mm)	FACCIA	PREZZO (€)
4072	160×100	doppia	7,80
P-995	233×160	singola	13,50
P-997	233×160	doppia	17,00

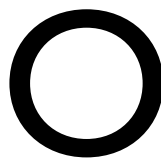
Prezzi IVA inclusa.

APRICANCELLO IOT



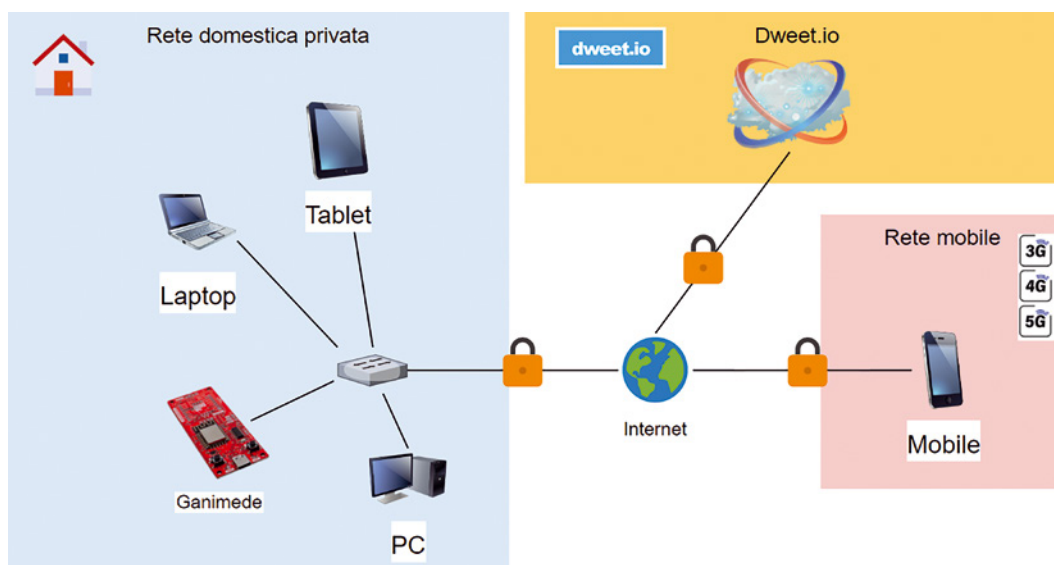
di ANDREA MISURACA

Tramite la board Ganimede.E12, gestiamo l'apertura e la chiusura di un cancello, o di altri apparati elettronici, utilizzando un'applicazione dedicata su smartphone.



gggiorno l'attenzione verso la domotica e i dispositivi IoT (Internet of Things) è così ampia che siamo spesso alla ricerca di elettrodomestici o altri dispositivi elettrici dotati di software che ci consentano di gestirli al meglio. Tuttavia, controllare il condizionatore o le tapparelle da remoto ha spesso un costo più elevato rispetto ad apparecchi puramente meccanici, per non parlare del fatto che per controllarli dai nostri cellulari dovremmo sostituirli con apparecchiature più moderne, nonostante le nostre vecchie funzionino ancora più che bene.

Fig. 1
Schema
di principio.



In aggiunta a ciò, un aspetto che chi ha meno familiarità con la materia non considera è quello della cybersecurity. È vero che non è necessario implementare misure di sicurezza informatiche simili a quelle di una banca o di un'automobile per gestire dei semplici elettrodomestici, ma è anche vero che sono in tanti coloro che non si fanno problemi ad acquistare una telecamera che consente un live streaming dal proprio telefono, senza preoccuparsi del fatto che quelle immagini passino attraverso server che si trovano dall'altra parte del mondo, dove il GDPR (la normativa per la tutela della privacy in Europa) non è sempre rispettato alla lettera. I più avvezzi opteranno per soluzioni fatte in casa, come l'uso di un proprio server o di un servizio su cloud, ma tutto ciò ha un costo.

In questo articolo proponiamo di realizzare un dispositivo adatto a controllare l'apertura e la chiusura di un cancello da remoto, costituito dalla scheda Ganimede.E12 (già presentata sul numero 272 di ElettronicaIn) e da una applicazione mobile appositamente sviluppata. Data la versatilità del progetto (si utilizza un relè per il controllo del carico), sarà possibile estenderne l'utilizzo anche ad altre applicazioni, come il controllo di boiler o caldaie domestiche. L'applicazione mobile non raccoglie alcun tipo di dato personale, e non trasmette alcuna informazione rilevante da un punto di vista privacy ad alcun server, come si può facilmente verificare con un software che analizza il traffico dati del proprio cellulare.

Per concludere, la comunicazione tra l'app e l'apri-cancello avviene tramite il servizio cloud gratuito offerto da *dweet.io*, in maniera totalmente sicura.

SCHEMA A BLOCCHI DEL SISTEMA

In **Fig. 1**, è il riportato lo schema a blocchi del sistema. Ciò che balza subito all'occhio è il tipo di architettura: serverless, cioè senza alcun server, né domestico, né fornito come servizio da un provider. La comunicazione è diretta, tra il dispositivo apri-cancello connesso alla rete WiFi di casa e l'applicazione mobile, per mezzo del servizio cloud fornito da *dweet.io*, che consente lo scambio di messaggi in modo sicuro ed efficiente. Un vantaggio sia economico sia per l'ambiente, visto che non ci sono server e dispositivi di rete da gestire.

HARDWARE

Come abbiamo anticipato, per l'implementazione HW ci serviamo della board Ganimede.E12, dotata di un microcontrollore ESP-12F e degli slot di espansione necessari per il collegamento dell'hardware aggiuntivo. In particolare, collegheremo al connettore Neopixel (CN4) il relè 5 Vdc 10 A, in modo da controllarlo utilizzando un PIN atto alla gestione di un output digitale, ed il Display OLED miniatura 0,96" connesso tramite I2C al connettore CN7 predisposto sulla board. L'architettura hardware, riportata in **Fig 2**, è molto semplice e consentirà di sviluppare un firmware altrettanto lineare e leggibile.

PROTOCOLLO

Più volte è stato menzionato *dweet.io* come fornitore di servizi, ma prima di entrare nel dettaglio dell'architettura software che è stata utilizzata, è necessario capire il funzionamento di questo provider.

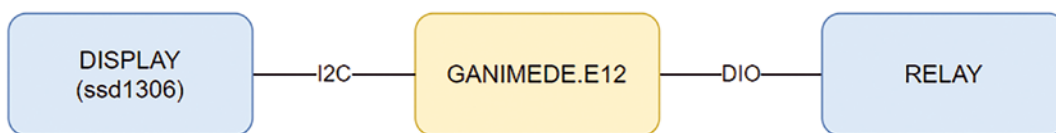


Fig.2
Schema a
blocchi HW.

Il nome stesso richiama un famoso social network e piattaforma di microblogging, ed è proprio dal noto Twitter che prende spunto. Qui però non sono persone fisiche a scambiarsi messaggi, ma dispositivi IoT. Una volta definito un nome per il proprio dispositivo, il cosiddetto *endpoint*, tutti potranno scrivere e leggere dei messaggi riferiti a quell'endpoint, come se uno o più utenti stessero twittando riferendosi ad uno specifico argomento usando un hashtag. Molti altri servizi permettono la comunicazione di dispositivi IoT, ma i due fattori che rendono dweet.io particolarmente appetibile sono il protocollo usato e la gratuità del servizio. Non vengono usati i tipici protocolli del mondo embedded per comunicare (o "dweettare"), ma delle web API, in particolare delle semplici HAPI (Humanized Application Programming Interface). L'idea dietro le HAPI è creare uno standard per interfacce in modo che siano auto documentate, e quindi facili da essere interpretate, sia dalle macchine che le utilizzano, ma soprattutto dagli sviluppatori. I cardini principali dell'HAPI sono infatti:

- l'accessibilità attraverso un URL, anche con un semplice web browser;
- la facilità di lettura di input e output sotto forma di normali frasi;
- la facilità di lettura, in modo che sia comprensibile anche a persone non tecniche.

Le HAPI sfruttano il protocollo applicativo HTTP, che usiamo quotidianamente per navigare su un qualsiasi sito internet, ma ne semplificano l'uso,

poiché vengono utilizzati solo due dei diversi metodi a disposizione: GET e POST. Il primo per leggere un'informazione, o dweet, il secondo per crearla.

Con le HAPI vengono standardizzati anche gli URL, così richieste e risposte sono facilmente interpretabili per qualunque tipo di comunicazione. L'idea di limitare la personalizzazione degli URL da parte di colui che si occupa del design delle API nasce dall'esigenza di semplificare le operazioni CRUD (Create, Read, Update, Delete) per la gestione dei dati, senza la necessità di consultare ripetutamente la documentazione.

Definendo un nome per il nostro device, ad esempio "ElnApricancello", il metodo GET che ci consentirà di leggere l'ultimo dweet sarà chiamato dal seguente URL:

<https://dweet.io/get/latest/dweet/for/ElnApricancello>

Allo stesso modo sarà possibile creare un dweet con una richiesta POST all'URL seguente, inserendo un JSON nel corpo della richiesta stessa:

<https://dweet.io/dweet/for/ElnApricancello>

Infine, è utile dare uno sguardo al contenuto completo di una richiesta, ovvero il JSON, che viene inviato quando postiamo un dweet. Supponendo che il corpo della richiesta sia una stringa di test, il dweet completo sarà:

```
{
  "this": "succeeded",
  "by": "getting",
  "the": "dweets",
  "with": [
```

Il display e il relè possono essere fissati con della semplice colla a caldo.

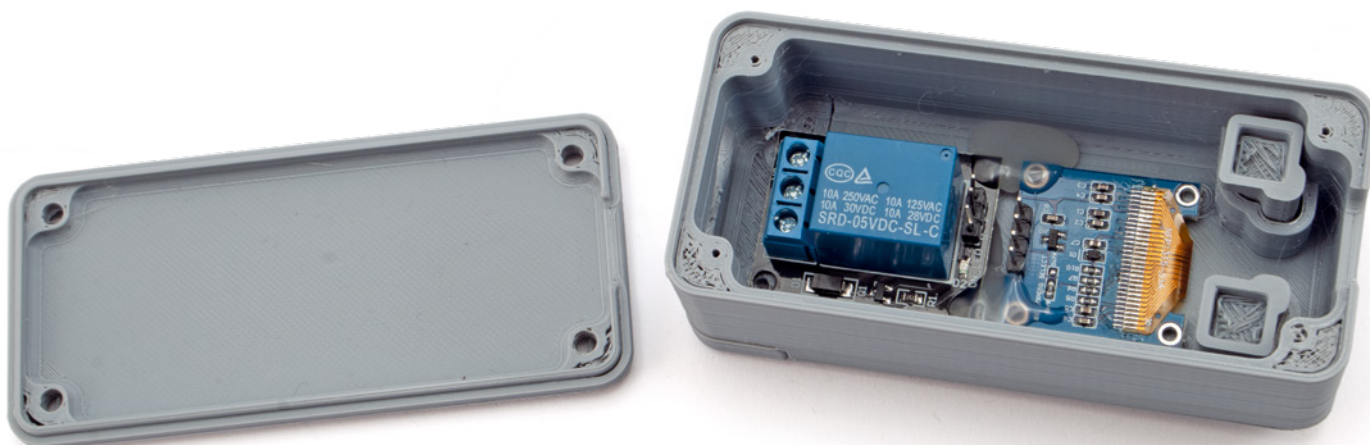
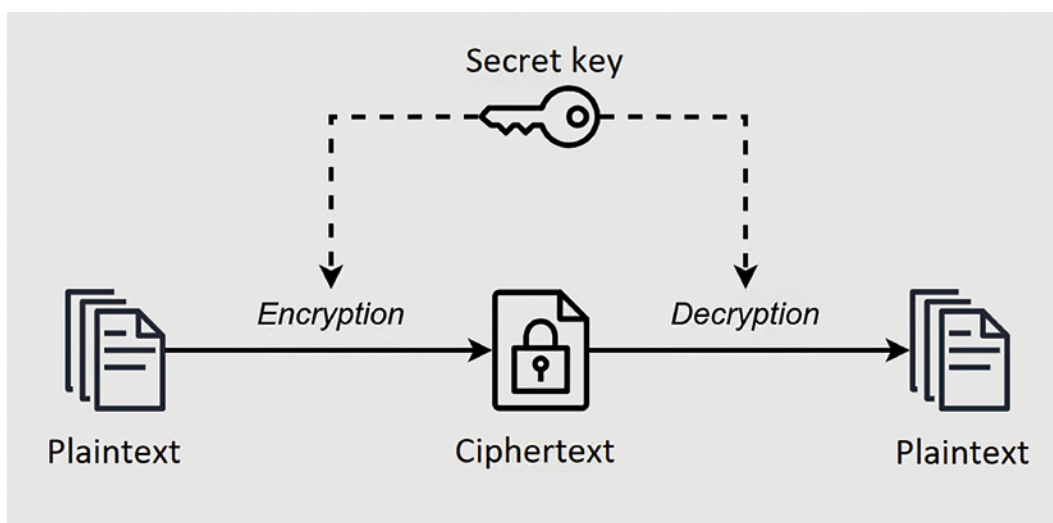


Fig. 3
Schema di cifratura.



```
{
  "thing": "EInApricancello",
  "created": "2023-05-15T18:41:17.166Z",
  "content": {
    "test": "stringa di test!"
  }
}
```

Analizzando in dettaglio il JSON del dweet è possibile notare come la lettura sia immediata grazie alla struttura delle HAPI. I campi dell'oggetto sono creati e popolati automaticamente dal servizio dweet.io, che oltre ai valori predefiniti inserisce il nome dell'endpoint, il timestamp, ed il contenuto della nostra richiesta nel campo "content". Ci siamo soffermati sulla semplicità di utilizzo del provider e abbiamo evidenziato la gratuità del servizio. Ma proprio quest'ultimo fattore implica delle restrizioni che sono sormontabili utilizzando la versione a pagamento del provider. Le maggiori limitazioni offerte dalla versione gratuita sono il limite temporale di richieste (non è possibile 'dweettare' con una frequenza maggiore di 1 Hz) e non sono previsti metodi di autenticazione e cifratura. Il fatto di non poter inviare più di una richiesta al secondo potrebbe essere un problema per applicazioni *strict real-time*, cioè che hanno bisogno di scambiarsi segnali con frequenze molto alte e tempi di risposta inferiori al millisecondo. Per applicazioni IoT, ed in particolare per un apricancello, non è necessario avere un requisito del genere. Ovviamente ciò potrebbe essere una limitazione per sistemi embedded in cui le misure

di safety sono indispensabili, come ad esempio le ECU utilizzate nel mondo automotive. Potrebbe invece essere un problema la mancanza di autenticazione, che ci garantirebbe di conoscere chi è la persona che sta interagendo con il dispositivo, e la mancanza di cifratura, per nascondere i dettagli dei messaggi inviati. La versione gratuita di dweet.io non solo non ha queste feature, ma al seguente link <https://dweet.io/see> è possibile vedere i messaggi scambiati dai dispositivi di tutto il mondo. Ciò potrebbe essere un problema per i dispositivi domotici, perché chiunque potrebbe, ad esempio, aprire il cancello di casa nostra, o vedere le informazioni scambiate dai nostri dispositivi. La soluzione più semplice al problema sarebbe acquistare la versione a pagamento del servizio, ma la più interessante è implementare un sistema per criptare il contenuto dei nostri dweet per fare in modo che nessun altro possa interagire con la nostra board, aprendo ad esempio il cancello al posto nostro.

CYBERSECURITY

Per proteggere la comunicazione tra la board e l'applicazione mobile da malintenzionati è necessario implementare un meccanismo in grado di cifrare i messaggi. Come illustrato nel paragrafo precedente, possiamo 'dweettare' un messaggio, che non è altro che un JSON contenente i dati che vogliamo inviare. Il corpo di questo messaggio finisce nel campo *content* del dweet, mentre tutti gli altri parametri sono automaticamente impostati dal provider con dei metadati. Per rendere sicuri i nostri segnali abbiamo bisogno quindi di criptare

il campo *content*, mentre possiamo ignorare il fatto che i metadati vengano inviati in chiaro.

Il primo passo da compiere è scegliere una tipologia di algoritmo che faccia al caso nostro.

Per l'applicazione si è pensato di utilizzare un algoritmo a **chiave simmetrica**, vale a dire in grado di criptare e decifrare un messaggio utilizzando la stessa chiave. Questa tecnica è molto diffusa ed abbastanza semplice, ma ha anche alcuni svantaggi, ad esempio entrambi gli estremi coinvolti nella comunicazione devono avere la stessa chiave, e spesso proprio lo scambio della chiave rappresenta il nodo cruciale di tutta la comunicazione.

Si può quindi intuire che sia la board, che l'applicazione mobile, dovrebbero avere in memoria una chiave necessaria per far funzionare l'algoritmo di cifratura.

L'idea è di fare in modo che l'utente imposti la chiave (sostanzialmente una password) all'interno dell'applicazione sul proprio cellulare, ed al primo avvio della board la trasmetta al dispositivo, insieme ad una serie di dati: il nome dell'endpoint dweet, il nome della rete wifi a cui la board si dovrà connettere, e la password di questa rete.

Una volta pensato a come la chiave possa essere usata dai nostri due punti di comunicazione possiamo passare alla descrizione di un generico algoritmo a chiave simmetrica:

- Dato un messaggio P (*PlainText* o testo in chiaro) ed una chiave k (una password), il mittente utilizzando un algoritmo di crittografia simmetrica S otterrà un nuovo messaggio C (*Ciphertext* o messaggio cifrato): $S(P, k) = C$
- Il destinatario, ottenuto il messaggio cifrato C , utilizzando la stessa chiave k ed un algoritmo di decrittazione D otterrà il messaggio iniziale in chiaro P : $D(C, k) = P$

Nel nostro caso sarà sempre l'app a fare da mittente ed a dover criptare il messaggio, mentre il dispositivo embedded dovrà decrittarlo.

Sono molteplici le implementazioni di algoritmi simili, ma per questa applicazione abbiamo scelto l'*Advanced Encryption Standard* (**AES**, conosciuto anche come Rijndael).

L'AES è uno dei più famosi algoritmi a chiave simmetrica, usato in tutto il mondo per le sue specifiche di sicurezza, perfino come standard dal governo americano.

Non entreremo nel merito dei dettagli tecnici e matematici dell'algoritmo perché grazie alla sua notorietà oggi sono tantissime le librerie che offrono robuste implementazioni facili da usare in diversi linguaggi di programmazione.

Una volta implementato il nostro algoritmo i nostri dweet avranno il campo *content* cifrato:

```
{
  "this": "succeeded",
  "by": "getting",
  "the": "dweets",
  "with": [
    {
      "thing": "EInApricancello",
      "created": "2023-05-15T18:41:17.166Z",
      "content": {
        "ciphertext": "1kdjf98sdaf1kj4lkajsdf!asd1kfj9"
      }
    }
  ]
}
```

La stringa contenuta dentro il *ciphertext*, apparentemente incomprensibile, una volta decifrata dalla nostra board conterrà il messaggio originale, ad esempio il comando per aprire il cancello.

Ricordiamo che nessuno senza la password potrà decifrare il messaggio, ma il ciphertext sarà sempre visibile a tutti su dweet.io. Ciò potrebbe esporre il nostro sistema ad un attacco di tipo replay: se il nostro ciphertext contiene l'equivalente di un messaggio "Apri il cancello", un potenziale malintenzionato potrebbe intercettare il messaggio ed inviarlo nuovamente in un secondo momento, riuscendo nell'intento. Ciò potrebbe essere risolto aggiungendo un layer di autenticazione, ma c'è un'alternativa ben più semplice per ovviare al problema. È sufficiente inserire nel contenuto del nostro messaggio il timestamp dell'applicativo mobile, così facendo la board potrà decifrare il messaggio contenente il timestamp, confrontarlo con il proprio, e se la differenza è superiore al secondo potrà scartare il messaggio. In questo modo un attacco replay non funzionerebbe, perché, anche se un messaggio venisse intercettato e ritrasmesso in un secondo momento, il ciphertext conterrebbe un timestamp così vecchio che verrà sempre ignorato dal dispositivo (si noti anche che il timestamp è a sua volta crittografato, quindi non sarebbe possibile alterarlo per farlo apparire come recente).

ARCHITETTURA SOFTWARE

Come abbiamo anticipato nell'introduzione, ci serviremo della scheda Ganimede.E12 come piattaforma HW per questo progetto. Come già ampiamente riportato nel numero 272 di ElettroIn, uno dei maggiori vantaggi di Ganimede è la



Fig. 4
Diagramma UML
delle classi.

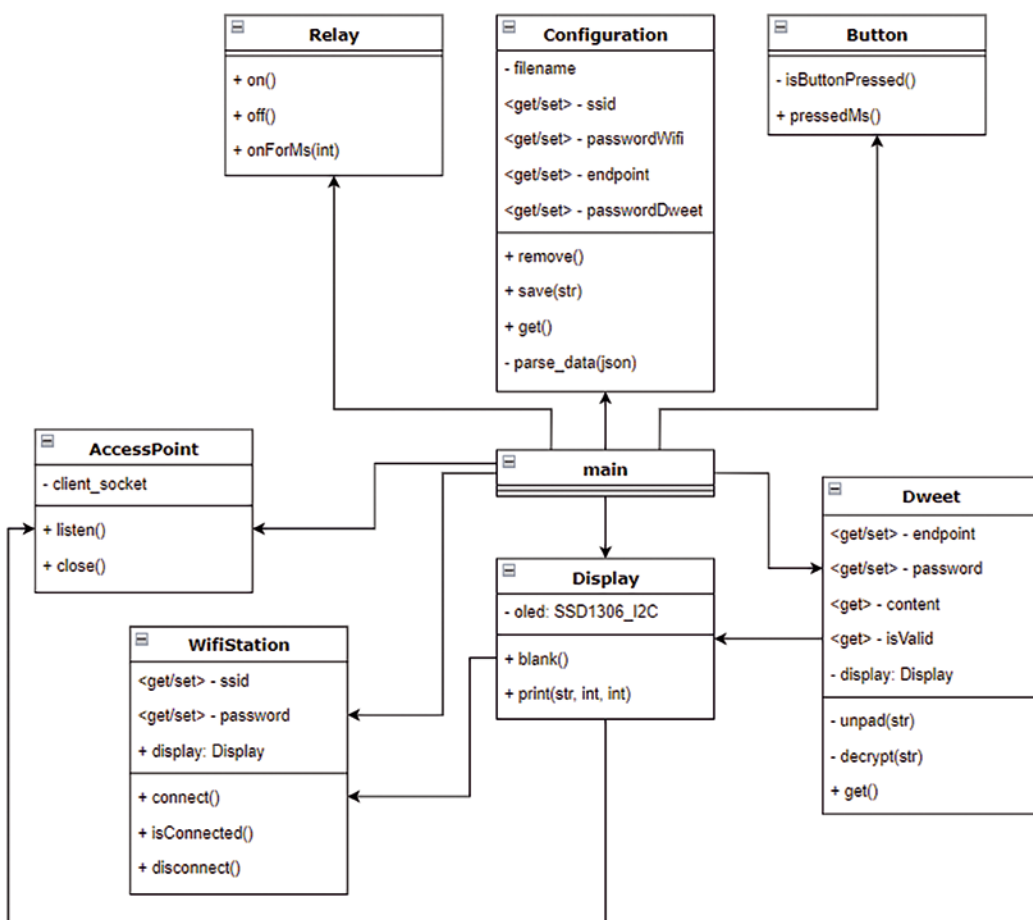
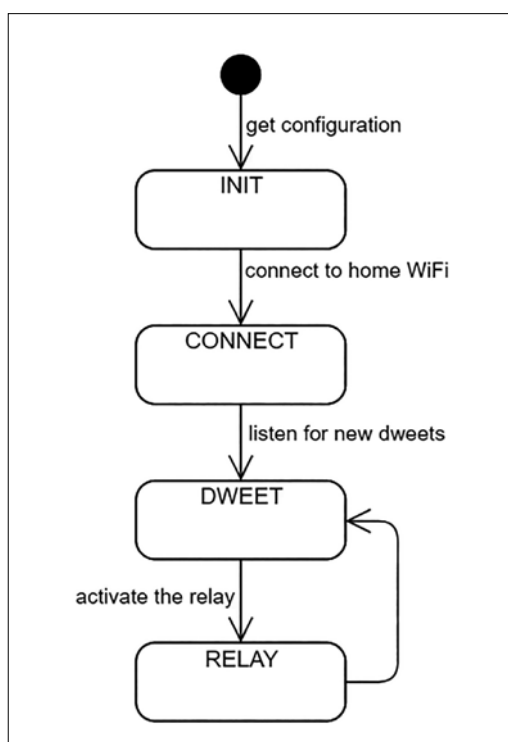


Fig. 5
Diagramma di flusso
so main.py



possibilità di programmarla utilizzando **micropython**. L'uso di questo famoso linguaggio interpretato nel mondo embedded, ed in particolare per dispositivi domotici, che non hanno bisogno di performance elevatissime, ci consente di creare un'architettura software pulita, semplice, e con dei tempi di implementazioni bassissimi. Prima di addentrarci nel codice e nei diagrammi dobbiamo precisare che per quest'articolo si è deciso di utilizzare una programmazione orientata agli oggetti (**OOP**) come paradigma di programmazione. Potrebbe essere considerata una scelta inusuale, ma probabilmente conveniente in questo caso. La programmazione procedurale è spesso preferita nei dispositivi embedded perché i linguaggi di basso livello, come il C, sono la scelta più usata per controllare la memoria byte per byte e ottimizzare quindi le performance. Sono veloci, danno al programmatore il più alto grado di libertà possibile, ma sono anche complicati, e prони agli errori. Ed ovviamente non prevedono l'uso di classi e oggetti.

È per questo che sono nati negli anni framework, linee guida, architetture e stili di implementazioni da seguire per raggiungere quella flessibilità nativa nei linguaggi di programmazione ad oggetti che nei linguaggi a basso livello sono difficili da riprodurre. Python non ci darà la possibilità di controllare ogni singolo byte che stiamo allocando, ma per un dispositivo IoT non è neanche troppo necessario. Per il nostro specifico caso, l'uso dell'OOP associato ad un linguaggio così flessibile e potente ci consentirà di scrivere del codice semplice, leggibile, pulito e di rispettare i principi *SOLID* (Single responsibility, Open-closed, Liskov substitution, Interface segregation, Dependency inversion), che consentono di avere un'implementazioni scalabile e mantenibile. Ricordiamo che una classe non è altro che un modello, una sorta di descrizione di una parte del nostro programma. La classe contiene degli attributi, cioè delle variabili, e dei metodi, cioè funzioni. Quando una classe viene istanziata, allora avremmo creato un oggetto.

Le classi che costituiscono il software sono:

- **AccessPoint:** per l'implementazione della rete wifi Ad-Hoc utilizzata per configurare la board al primo avvio o dopo un reset;
- **WifiStation:** che gestisce la connessione alla propria rete WiFi domestica;
- **Configuration:** con un wrapper di metodi per semplificare il salvataggio in NvM delle configurazioni impostabili tramite app, in modo che siano disponibili anche quando effettuiamo un'hard reset (ad esempio spegnimento e riaccensione);
- **Dweet:** che si limita a leggere in polling i messaggi su dweet.io, decriptarli ed effettuare il parser delle richieste.

Infine ci sono tre classi che possiamo definire *hardware dependent*, poiché la loro implementazione è strettamente correlata all'hardware ed ai collegamenti di Ganimede:

- **Display:** in cui sono specificati i pin e i metodi per interagire con il display OLED, utile per stampare messaggi di stato del device;
- **Button:** che gestisce la pressione del bottone utente, in modo da gestire le funzionalità di reset dopo una lunga pressione;
- **Relay:** che attiva e disattiva il relè connesso al connettore neopixel.

In **Figura 4** è riportato un diagramma delle classi semplificato (*UML Class Diagram*), dove vengono riportati i metodi pubblici e privati della nostra implementazione, così come gli attributi con eventuali *getter* e *setter*.

Micropython automaticamente esegue il file `main.py` all'avvio della board, per questo lo abbiamo scelto come entry point del nostro software, il posto migliore per istanziare le nostre classi e per implementare la macchina a stati del nostro apricancello. `Main.py` contiene una sola funzione che esegue all'infinito la macchina a stati riportata in **Figura 5**. Lo stato *INIT* verifica se è presente una configurazione in memoria come rappresentato in **Figura 6**. Se non vi è alcuna configurazione, Ganimede attiva la rete WiFi Ad-Hoc per riceverne una, altrimenti passa allo stato *CONNECT*.

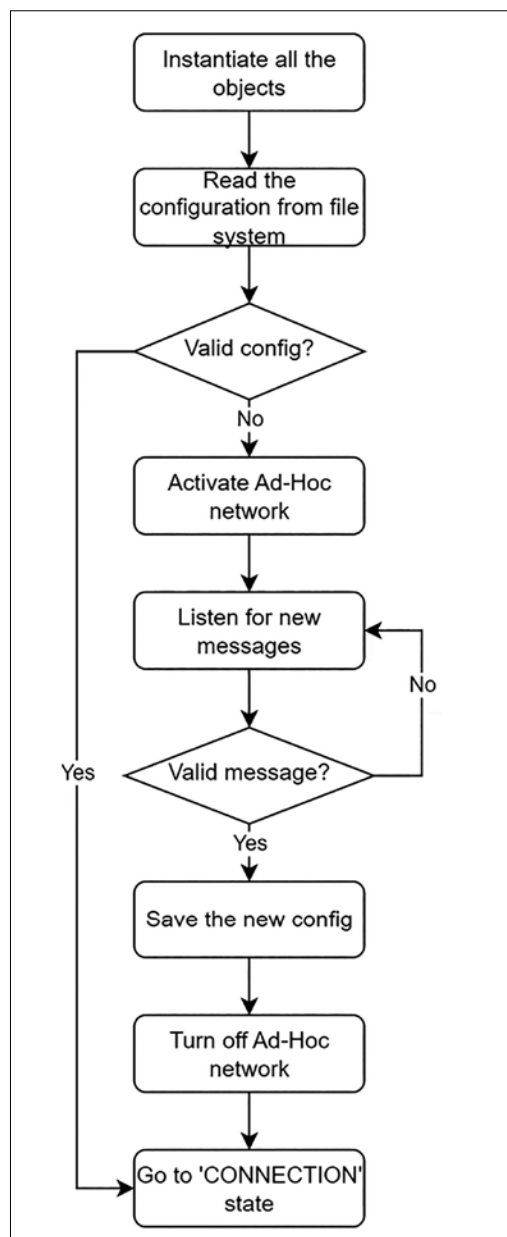
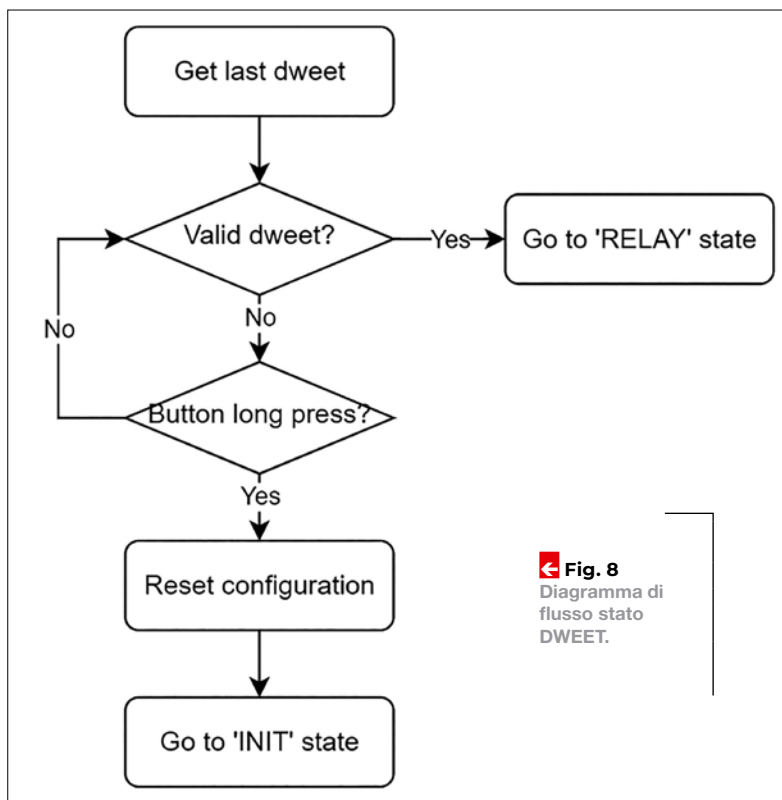
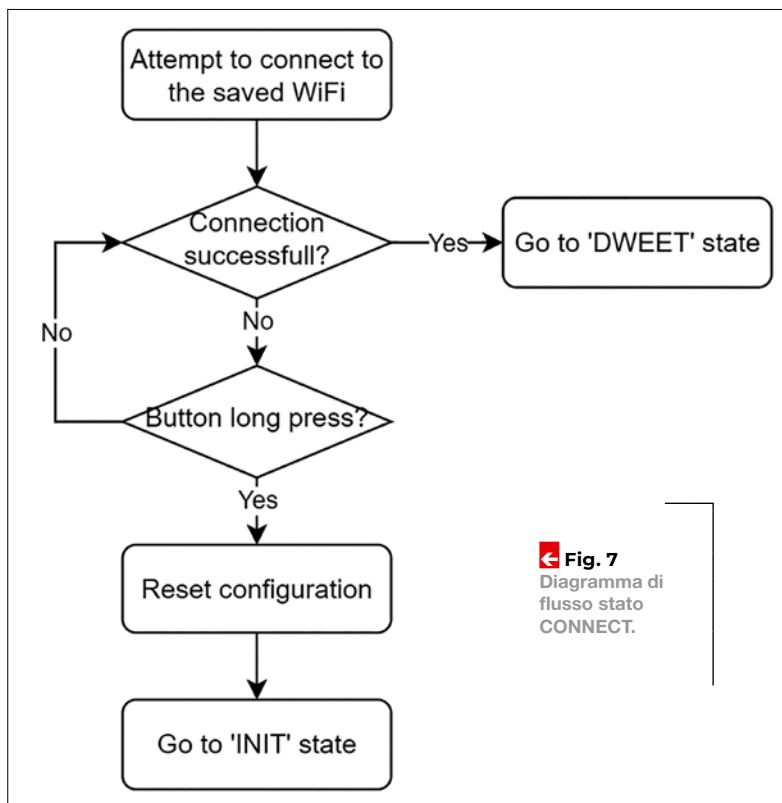


Fig.6
Diagramma
di flusso
stato INIT.



In questa fase il dispositivo cercherà di collegarsi alla nostra rete WiFi. Una volta che la connessione effettua la *FSM* (*Finite State Machine*) transita allo stato *DWEET*, come riportato in **Figura 7**.

Se la connessione dovesse cadere perché ad esempio vogliamo riavviare il router di casa, la board si riconnetterà automaticamente non appena il segnale ritorna disponibile. Lo stato *DWEET* è quello in cui il nostro software si troverà per quasi tutto il tempo di esecuzione, perché è proprio in queste poche righe di codice, in particolare con la funzione *dweet.get()* che andremo a leggere se abbiamo ricevuto un nuovo comando. Ogni qualvolta un nuovo messaggio valido è ricevuto, la FSM passerà a *RELAY*, ma soltanto per il tempo necessario ad attivare il relè. Il diagramma di flusso dello stato *DWEET* è riportato in **Figura 8**, mentre lo stato *RELAY* in **Figura 9**.

Si noti come la FSM sia corredata del codice necessario (**Listato 1**) per leggere la pressione del bottone ed eventualmente effettuare un reset. Inoltre il SW è corredata di una serie di messaggi di debug (inviati sulla console micropython) e utente (stampati sul display OLED) per fornire all'utente finale maggiori dettagli sullo stato del device.

La classe *AccessPoint*, riportata nel **Listato 2**, è estremamente semplice poiché consta solo di due metodi: *listen* e *close*. Grazie ai moduli già inclusi in micropython, l'implementazione di questa funzionalità è realizzabile con una manciata di righe di codice, a differenza delle centinaia necessarie se il linguaggio scelto fosse ad esempio il C.

L'idea è di creare una rete WiFi con SSID (il nome della rete) ed una password predefinite e non modificabili, poiché l'interazione tra l'app e la rete Ad-Hoc avviene solo in fase di configurazione, una tantum, e la disponibilità della rete decade una volta che le impostazioni sono state salvate.

Per lo stesso motivo i dati scambiati in questa fase avverranno con l'utilizzo del protocollo HTTP (senza 'S'). Il protocollo è per definizione non sicuro, ma l'implementazione di misure di sicurezza sofisticate che prevedono la gestione di chiavi e certificati non sembrano per niente adatte in questo caso. Rendere particolarmente sicura la comunicazione tra app e board, che avviene una sola volta nel ciclo di vita del prodotto, e che può durare non più di un paio di minuti, non giustificerebbe l'overhead che si creerebbe in termini di performance (salvataggio e recupero dalla memoria di chiavi e certificati) e in termini di tempi di sviluppo.

Il metodo in questione, come richiama la parola stessa, non fa altro che attivare la rete WiFi e

Listato 1

```
#!/usr/bin/env python3
# -*- coding: UTF-8 -*-
from configuration import Configuration
from button import Button
from access_point import AccessPoint
from dweet import Dweet
from wifi_station import WifiStation
from relay import Relay
from display import Display
import ntptime

def loop():
    configuration = Configuration()
    button = Button()
    ap = AccessPoint()
    station = WifiStation()
    dweet = Dweet()
    relay = Relay()
    display = Display()
    state = 'INIT'
    while True:
        if state == 'INIT':
            display.print('BENVENUTO', 26, 0)
            display.print('ElettronicaIn', 12, 35)
            display.print('Apricancello', 15, 45)
            configuration.get()
            if configuration.passwordWifi is None:
                data = ap.listen()
                configuration.save(data)
                ap.close()
            else:
                state = 'CONNECT'
        elif state == 'CONNECT':
            station.password = configuration.passwordWifi
            station.ssid = configuration.ssid
            station.connect()
            while not station.isConnected():
                if button.pressedMs() > 5000:
                    configuration.remove()
                    print('perform reset')
                    display.blank()
                    display.print('RESET...', 0, 0)
            if station.isConnected() is True:
                ntptime.settime()
                display.blank()
                state = 'DWEET'
        elif state == 'DWEET':
            display.print('ElettronicaIn', 12, 0)
            display.print('Apricancello', 15, 10)
            display.print('In attesa di un', 0, 35)
            display.print('comando...', 0, 45)
            if button.pressedMs() > 5000:
                configuration.remove()
                print('perform reset')
                display.blank()
                display.print('RESET...', 0, 0)
            dweet.password = configuration.passwordDweet
            dweet.endpoint = configuration.endpoint
            dweet.get()
            if dweet.content is not None:
                state = 'RELAY'
        elif state == 'RELAY':
            display.blank()
            display.print('Comando ricevuto!', 0, 0)
            activation_ms = int(dweet.content['time'])
                * 1000
            relay.onForMs(activation_ms)
            display.blank()
            state = 'DWEET'
    loop()
```

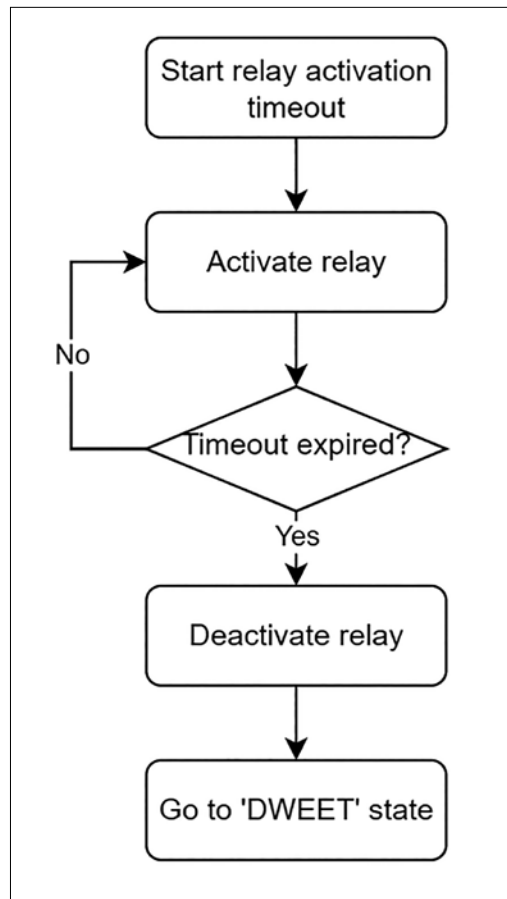


Fig.9
Diagramma
di flusso sta-
to RELAY.

ascoltare l'arrivo di un nuovo messaggio. Una volta che il messaggio è ricevuto, viene salvato in memoria non volatile ed il metodo `close` viene invocato. Quest'ultimo si limita ad interrompere la rete creata, in modo che nessun client vi si possa connettere, inviando per esempio altri messaggi malevoli.

Abbiamo più volte menzionato il salvataggio in memoria non volatile. Chi ha esperienza con la scrittura di firmware sa che è spesso un tasto dolente. Avere uno stack software in grado di fornire delle interfacce a chi scrive il software applicativo è indispensabile in un progetto medio/ grande, altrimenti gli sviluppatori passeranno il tempo non a scrivere codice, ma ad analizzare gli indirizzi della memoria flash e verificare tutte le possibili eccezioni che possono nascere con i propri dati. Ganimede non solo ci aiuta, ma va ben oltre. Possiamo salvare i nostri dati in semplici file di testo, utilizzando le normali funzioni di libreria di Python, e vi possiamo accedere come se stessimo leggendo il file system del nostro PC.

Più che darci una mano Micropython compie una vera e propria rivoluzione sotto questo aspetto,

Listato 2

```
#!/usr/bin/env python3
# -*- coding: UTF-8 -*-
import network
import socket
import time
from display import Display
class AccessPoint():
    """
    Handle the WiFi AP state.
    At the first startup, or after a reset, the board needs to be in AP mode
    in order to receive the configuration from the mobile app via an
    Ah-Hoc network.
    """
    def listen(self):
        # Set up the network
        _display = Display()
        self._ap = network.WLAN(network.AP_IF)
        self._ap.active(True)
        self._ap.config(essid='EIn Apricancello', password='ElettronicaIn')
        self._ap.ifconfig(('192.168.0.1',
                           '255.255.255.0',
                           '192.168.0.1',
                           '8.8.8.8'))

        # Print the IP address of the access point
        print('Access point IP address:', self._ap.ifconfig()[0])
        _display.blank()
        _display.print('Access point ip:', 0, 0)
        _display.print(self._ap.ifconfig()[0], 0, 10)
        # Define the TCP server socket
        server_socket = socket.socket()
        server_socket.bind(('0.0.0.0', 80))
        server_socket.listen(5)
        print('Server socket is listening...')
        # Wait (blocking) for an incoming connection
        self._client_socket, client_address = server_socket.accept()
        print('Received connection from', client_address)
        _display.blank()
        _display.print('Connessione da:', 0, 0)
        _display.print(str(client_address), 0, 10)
        # Read the request data
        request_data = self._client_socket.recv(1024).decode('utf-8')
        print('Received request data:', request_data)
        # Parse the request to get the POST data
        char = '{'
        index = request_data.find(char)
        parsed_data = request_data[index:]
        print('Received POST data: ', parsed_data)
        return parsed_data
    def close(self):
        # Send the response
        response_data = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n \
Hello, world!"
        self._client_socket.send(response_data.encode())
        self._client_socket.close()
        # Wait for the response to be sent to the app and disable the AP
        time.sleep(1)
        self._ap.active(False)
```

consentendoci di focalizzarci sulla logica applicativa facendoci dimenticare indirizzi, dimensioni e allineamenti vari.

La classe Configuration (**Listato 3**) non fa quindi altro che fornire dei metodi per le operazioni di scrittura, lettura e cancellazioni più comuni e necessarie per salvare i nostri parametri. Il metodo

privato `parse_data` viene utilizzato per leggere il JSON proveniente dall'app mobile. Il formato JSON è stato scelto per lo scambio dei dati perché è il più diffuso quando si hanno delle web API, è immediato da leggere e semplice da gestire, visto che Python possiede delle funzioni di libreria per la gestione del formato. I metodi contenuti in questa



Listato 3

```
#!/usr/bin/env python3
# -*- coding: UTF-8 -*-
import ujson
import os

class Configuration:
    """
    Configuration handler for the gate opener module.
    Save, retrieve and delete the configuration of the Ganimede board.
    """
    def __init__(self) -> None:
        self._filename = 'config.txt'
        self._ssid = None
        self._passwordWifi = None
        self._endpoint = None
        self._passwordDweet = None
    @property
    def ssid(self):
        return self._ssid
    @property
    def passwordWifi(self):
        return self._passwordWifi
    @property
    def endpoint(self):
        return 'https://dweet.io:443/get/latest/dweet/for/' + self._endpoint
    @property
    def passwordDweet(self):
        return self._passwordDweet
    def remove(self):
        """
        Delete the configuration from the memory, e.g.: after a reset
        """
        # If there is a config file, remove it
        try:
            os.remove(self._filename)
        except OSError:
            print('There is no config file to remove')
    def save(self, configuration: str):
        """
        Save the configuration into the memory
        :param configuration: The actual configuration json
        :type configuration: str
        """
        with open(self._filename, 'w') as config_file:
            config_file.seek(0)
            config_file.write(configuration)
        # Parse the JSON string
        self._parse_data(configuration)
    def get(self):
        """
        Read the configuration and returns the actual str parameters
        """
        try:
            with open(self._filename, 'r') as config_file:
                data = config_file.read()
                # Parse the JSON string
                self._parse_data(data)
        except Exception as e:
            print('Cannot read config file: ', e)
    def _parse_data(self, data):
        # Parse the JSON string
        data = ujson.loads(data)
        self._ssid = data['ssid']
        self._passwordWifi = data['passwordWifi']
        self._endpoint = data['endpoint']
        self._passwordDweet = data['passwordDweet']
```

classe non fanno altro che leggere il contenuto di un file di testo, denominato configuration.txt, scrivere il contenuto del file, ed eventualmente cancellarlo, ad esempio in seguito ad un reset. Un altro importante componente software del nostro sistema è dato dalla classe WifiStation. La connessione ad una rete WiFi è ormai una

procedura integrata nelle librerie standard di molti linguaggi di programmazione. Python non è da meno, basta fornire SSID (il nome della rete) e la password, che prenderemo dalla configurazione precedentemente salvata, alle API fornite dalla libreria e grazie al modulo network importato sarà possibile connettersi. Ovviamente per passare al

Listato 4

```
#!/usr/bin/env python3
# -*- coding: UTF-8 -*-
import network
from display import Display
class WifiStation():
    """
    Handle the WiFi station.
    """
    def __init__(self) -> None:
        self._ssid = None
        self._password = None
        self._display = Display()
    @property
    def ssid(self):
        return self._ssid
    @ssid.setter
    def ssid(self, ssid):
        """
        Set password
        :param passwordDweet: The key (password) used to decrypt
                               the content
        :type passwordDweet: str
        """
        self._ssid = ssid
    @property
    def password(self):
        return self._password
    @password.setter
    def password(self, password):
        """
        Set password
        :param passwordDweet: The key (password) used to decrypt
                               the content
        :type passwordDweet: str
        """
        self._password = password
    def connect(self):
        self._sta_if = network.WLAN(network.STA_IF)
        if not self._sta_if.isconnected():
            print('connecting to network... ', self._ssid)
            self._display.blank()
            self._display.print('Connessione in', 0, 0)
            self._display.print('corso a:', 0, 10)
            self._display.print(self._ssid, 0, 30)
            self._sta_if.active(True)
            self._sta_if.connect(self._ssid, self._password)
    def isConnected(self):
        isConnected = False
        if self._sta_if.isconnected():
            print('network config:', self._sta_if.ifconfig())
            isConnected = True
        return isConnected
    def disconnect(self):
        self._sta_if = network.WLAN(network.STA_IF)
        if self._sta_if.isconnected():
            print('disconnecting to network... ', self._ssid)
            self._display.blank()
            self._display.print('Disconnessione in corso', 0, 0)
            self._sta_if.active(False)
```

prossimo stato della FSM dobbiamo assicurarci di essere connessi alla rete, da qui l'esposizione del metodo `isConnected` che effettua questa verifica. La maggior parte della logica applicativa è gestita all'interno della classe *Dweet* (**Listato 5**). Importando il modulo *request* abbiamo accesso ad

una serie di funzioni in grado di chiamare in polling il metodo *GET* del protocollo *HTTP* con l'API definita da *dweet.io*. Il metodo viene ciclicamente chiamato in attesa di un nuovo comando. Quando il servizio invia una risposta valida, la board deve processare la richiesta. Il primo passo



Listato 5

```
#!/usr/bin/env python3
# -*- coding: UTF-8 -*-
import ujson
import os
class Configuration:
    """
    Configuration handler for the gate opener module.
    Save, retrieve and delete the configuration of the Ganimede board.
    """
    def __init__(self) -> None:
        self._filename = 'config.txt'
        self._ssid = None
        self._passwordWifi = None
        self._endpoint = None
        self._passwordDweet = None
    @property
    def ssid(self):
        return self._ssid
    @property
    def passwordWifi(self):
        return self._passwordWifi
    @property
    def endpoint(self):
        return 'https://dweet.io:443/get/latest/dweet/for/' + self._endpoint
    @property
    def passwordDweet(self):
        return self._passwordDweet
    def remove(self):
        """
        Delete the configuration from the memory, e.g.: after a reset
        """
        # If there is a config file, remove it
        try:
            os.remove(self._filename)
        except OSError:
            print('There is no config file to remove')
    def save(self, configuration: str):
        """
        Save the configuration into the memory
        :param configuration: The actual configuration json
        :type configuration: str
        """
        with open(self._filename, 'w') as config_file:
            config_file.seek(0)
            config_file.write(configuration)
        # Parse the JSON string
        self._parse_data(configuration)
    def get(self):
        """
        Read the configuration and returns the actual str parameters
        """
        try:
            with open(self._filename, 'r') as config_file:
                data = config_file.read()
            # Parse the JSON string
            self._parse_data(data)
        except Exception as e:
            print('Cannot read config file: ', e)
    def _parse_data(self, data):
        # Parse the JSON string
        data = ujson.loads(data)
        self._ssid = data['ssid']
        self._passwordWifi = data['passwordWifi']
        self._endpoint = data['endpoint']
        self._passwordDweet = data['passwordDweet']
```

è decriptare i dati. Il metodo privato *decrypt* che prende in input il messaggio criptato (*ciphertext*) e la password contenuta nel file *config.txt*, utilizzando

la *cryptolib* di Python chiama le funzioni di libreria necessarie per decifrare il messaggio utilizzando l'AES spiegato precedentemente.

Listato 6

```
#!/usr/bin/env python3
# -*- coding: UTF-8 -*-
from machine import Pin
import time

class Button():
    """
    Handle the Ganimede button.
    """
    def __init__(self) -> None:
        self._button = Pin(0, Pin.IN, Pin.PULL_UP)
    def _isButtonPressed(self):
        return self._button.value() == 0
    def pressedMs(self):
        # Record the start time
        start_time = time.ticks_ms()
        while self._isButtonPressed():
            pass
        # Record the end time
        end_time = time.ticks_ms()
        # Calculate the elapsed time in milliseconds
        elapsed_time = time.ticks_diff(end_time, start_time)
        # Print the elapsed time
        if elapsed_time > 50:
            print("Button pressed for: {} ms".format(elapsed_time))
        return elapsed_time
```

Nel paragrafo dedicato alla cybersecurity abbiamo menzionato come la validità del messaggio, per evitare un attacco di tipo replay, viene implementata non tramite autenticazione, ma paragonando il timestamp della board con quello inviato dall'applicazione mobile. Se la differenza è sufficientemente bassa il messaggio viene considerato valido. Alla semplice logica che compara gli istanti di tempo, sono state aggiunte le funzioni per con-

vertire i timestamp in modo che siano facilmente confrontabili.

È noto che solitamente per timestamp si intendono i secondi, o in alcuni casi i millisecondi, trascorsi dal 1 gennaio 1970 UTC. Questo è il valore inviato dall'app, mentre il timestamp restituito dal componente RTC (*Real Time Clock*) della board restituisce i secondi dal 1 gennaio 2000. Le cosiddette epoch sono differenti, è necessario quindi effettuare una conversione. Il lettore più attento noterà che, mentre l'applicazione sul cellulare invierà sempre un orario valido vista la presenza quasi permanente di una connessione ad internet, il dispositivo embedded ha bisogno di precise istruzioni per effettuare questo sync. La seguente linea di codice contenuta in main.py permette la sincronizzazione dell'orario attraverso il protocollo NTP (*Network Time Protocol*):

```
ntptime.settime()
```

Le classi descritte fin qui interagiscono con l'hardware, in modo particolare con i componenti interni al System-on-Chip, ossia l'ESP12F. Passiamo ora all'interazioni con l'hardware esterno, cioè il relè, il display ed il bottone utente presente sulla Ganimede.

La classe Button, riportata nel **Listato 6**, contiene una funzione di inizializzazione in cui sono spe-

Listato 7

```
#!/usr/bin/env python3
# -*- coding: UTF-8 -*-
from machine import Pin
import time

class Relay():
    """ Handle the Ganimede relay via the
    Neopixel connector. """
    def __init__(self) -> None:
        self._relay = Pin(2, Pin.OUT, value=1)
    def on(self):
        self._relay.value(0)
    def off(self):
        self._relay.value(1)
    def onForMs(self, ms: int):
        deadline = time.ticks_add(time.ticks_ms(), ms)
        self.on()
        while time.ticks_diff(deadline, time.ticks_ms()) > 0:
            pass
        self.off()
```

Listato 8

```
from machine import Pin, SoftI2C
import ssd1306
def singleton(cls):
    instance = None
    def getinstance(*args, **kwargs):
        nonlocal instance
        if instance is None:
            instance = cls(*args, **kwargs)
        return instance
    return getinstance
@singleton
class Display():
    """Print the input text into the display."""
    def __init__(self, scl_pin: int = 5, sda_pin: int = 4) -> None:
        print('Initializing the display')
        _oled_width = 128
        _oled_height = 64
        _i2c = SoftI2C(scl=Pin(scl_pin), sda=Pin(sda_pin))
        self._oled = ssd1306.SSD1306_I2C(_oled_width, _oled_height, _i2c)
        self.blank()
    """Print the input text into the display."""
    def print(self, text: str, x: int, y: int) -> None:
        self._oled.text(text, x, y, 1) # draw some text at x, y, colour=1
        self._oled.show()
    """Color the display total black."""
    def blank(self) -> None:
        self._oled.fill(0) # fill entire screen with colour=0
        self._oled.show()
```

IL LABORATORIO VICINO AI SUOI CLIENTI

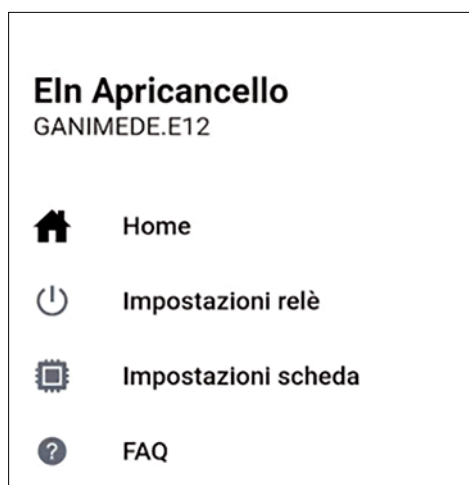
Il laboratorio di **collaudo**
e **certificazione** che rende
l'immissione dei prodotti
sul mercato **semplice**,
veloce e **sicura**.

Per tutti i settori si eseguono:

- prove di conformità durante il processo di certificazione
- sessioni di pre-testing durante le fasi di progettazione del prodotto
- prove per il controllo di produzione durante tutta la vita del prodotto



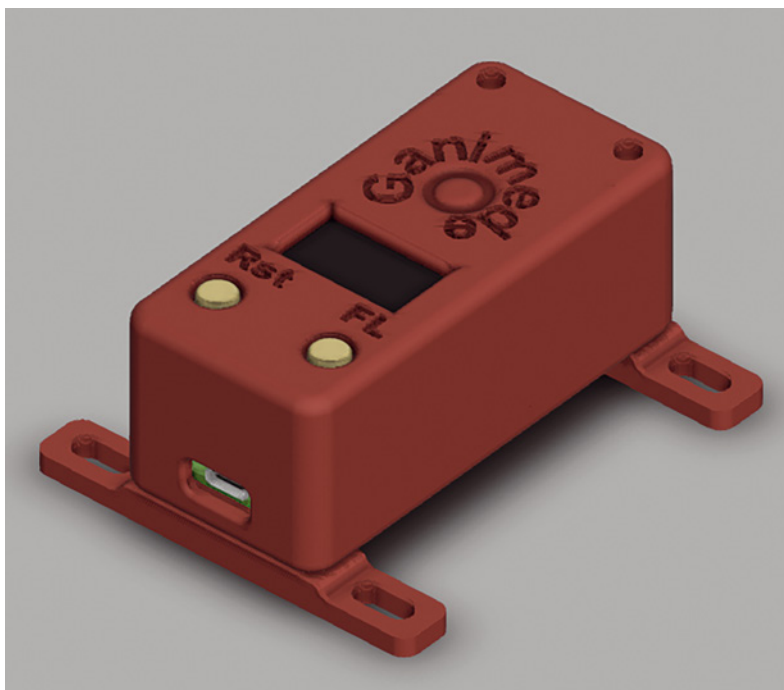
Fig. 10
L'app per Android ed iOS per il controllo dell'apricancello.



cificati i pin del pulsante, un metodo privato che rileva se il pulsante è stato premuto, ed un metodo pubblico che restituisce il numero di millisecondi per cui l'utente l'ha tenuto pigiato. Questa funzionalità è utile per effettuare un reset solo dopo una pressione piuttosto lunga, ad esempio 5 secondi.

Relay.py, descritta dal codice del **Listato 7**, ha un'analogia funzione di init, un metodo per attivare il relè ed uno per disattivarlo, ed un terzo per chiudere il contatto per un certo numero di millisecondi. Quest'ultimo è particolarmente utile nel caso dell'attivazione di un motore per il cancello,

Fig. 11
Immagine 3d del case contenente Ganimede, relè e display.



perché vogliamo che si attivi per un certo numero di secondi necessari all'apertura, nel caso il nostro relè vada fisicamente a pilotare un motore. Nel caso in cui sia sufficiente solo un impulso, possiamo comunque ridurre questa attivazione a piacimento (ad esempio impostando 1 o 2 secondi di attivazione).

In Display.py, **Listato 8**, infine abbiamo le funzioni che permettono di stampare un testo sul display OLED da 0.96". Il modulo *ssd1306* già contiene tutte le funzioni necessarie, quindi la classe si limita a fare da wrapper per adeguarle meglio al nostro caso d'uso. In questo file però, a differenza dei precedenti, abbiamo usato l'implementazione del *singleton* pattern, creando l'apposito decorator. Così facendo, ogni qualvolta si tenterà di creare un nuovo oggetto, viene effettuato un controllo: se un'istanza della classe esiste già, verrà restituito il riferimento, altrimenti verrà creata. Il motivo di questa scelta viene dal fatto che mentre Button e Relay vengono istanziate ed usate sono all'interno di main.py, Display viene usato all'interno di tutte le classi, poiché la stampa di un messaggio di stato può essere effettuata da qualunque componente. Poiché non vogliamo che la funzione di init venga richiamata ogni volta, visto che è sempre la stessa, così come non vogliamo allocazioni multiple di variabili che contengono uguale valore, abbiamo deciso di utilizzare questo paradigma.

ARCHITETTURA APP

Come più volte annunciato la semplicità di utilizzo dell'intero progetto è data anche dall'applicazione mobile che permette una facile interazione con l'hardware. L'applicazione è stata realizzata con un focus particolare alla facilità di utilizzo e alle performance in termini di velocità di esecuzione, con una grafica basata su un'interfaccia Material Design, e la possibilità di avere un prodotto multipiattaforma, Android e iOS. In **Fig. 10** è visibile un'immagine dell'interfaccia.

L'applicazione è autoesplicativa e molto intuitiva: da un menù laterale si può selezionare la schermata principale per inviare un comando alla baord, una schermata con le impostazioni relative alla configurazione di dweet.io ed un'altra con SSID e password della propria rete WiFi. Per maggiori informazioni relative ai settaggi ed al dispositivo in generale è presente una maschera dedicata alle FAQ, le domande più frequenti.

L'applicazione è pronta per essere usata dall'utente finale, ma per i più curiosi vogliamo fornire maggiori dettagli relativi all'implementazione.

Per la realizzazione si è scelto di usare il framework *ionic*. Così facendo si può creare un'applicazione per Android e iOS partendo dallo stesso codice. Il framework offre la possibilità di sviluppare utilizzando il linguaggio Javascript associato ad un framework di front-end, nel nostro caso React. Così facendo il tempo di sviluppo è notevolmente ridotto, e allo stesso tempo è possibile utilizzare note librerie per React, come *Material UI* per lo sviluppo di un'interfaccia grafica moderna e accattivante, piuttosto semplice nel nostro caso. Per concludere sottolineiamo ancora una volta come il comando inviato dall'applicazione non è altro che una richiesta HTTP POST criptata a dweet.io, quindi non è presente alcun intermediario e la nostra privacy è ben tutelata.

VARIANTI

Il progetto descritto fa uso del seguente hardware esterno alla board: un relè ed un display OLED da 0.96". Ricordiamo che grazie all'implementazione modulare descritta al paragrafo precedente, per sostituire uno dei componenti non si dovrà fare altro che sostituire la relativa classe. Per utilizzare un altro display basterà modificare `Display.py`, o basterà rimuovere le chiamate alle funzioni se non vogliamo alcuno schermo. Cambiando il file `Relay.py` potremmo utilizzare un hardware diverso, come un modulo click-relè o un relè-groove, sfruttando i connettori presenti su Ganimede. Potenzialmente potremmo anche utilizzare un altro pulsante al posto di quello integrato sulla board, tutto grazie alla versatilità ed ai numerosi bus e connettori presenti su Ganimede e già descritti nel numero 272 di *ElettronicaIn*.

CASE

Grazie al contributo di un appassionato della comunità Ganimede (Bruno Luziatelli, che ringraziamo), è stata sviluppata anche una scatola stampabile in 3d, in grado di contenere Ganimede, il relè (codice Futura Elettronica RELAY1CH) ed il display OLED (sempre disponibile sul sito di Futura Elettronica, OLEDTVSCSD). Il risultato (visibile in **Fig. 13**) è una scatola dotata di staffe per fissaggio e opportuni fori per i cablaggi, che risulta molto utile per l'utilizzo in installazioni reali. Data la presenza di fori per display e cablaggi vari, raccomandiamo comunque l'utilizzo di questa scatola per installazioni in interni, o protette da opportuni contenitori con grado di protezione adeguato all'applicazione. I file Step per la stampa del case possono essere scaricati dal sito della rivista.

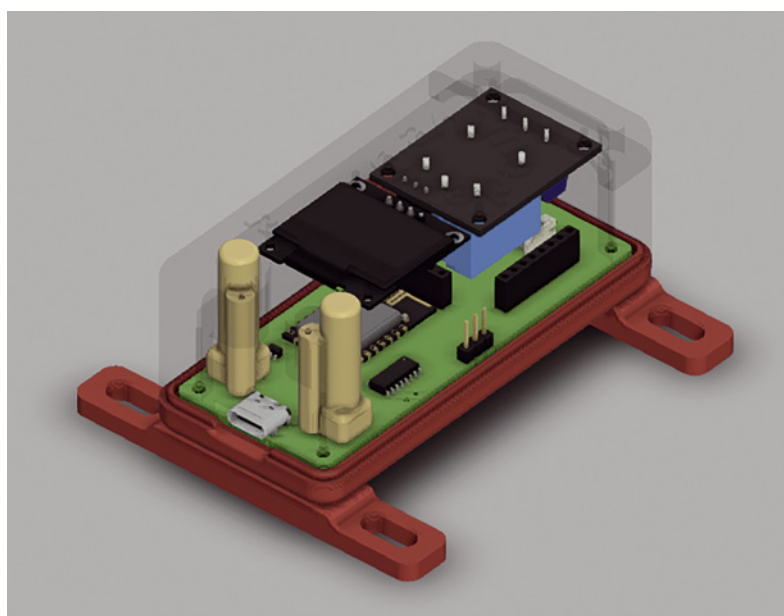


Fig. 12
Spaccato 3d del case contenente Ganimede, relè e display.

CONCLUSIONI

Il progetto presentato in questo numero dimostra come, tramite l'utilizzo di Ganimede.E12 e del framework micropython correlato, sia possibile realizzare in pochissimo tempo un dispositivo affidabile e sicuro, anche per chi è meno esperto. Troviamo a bassissimo prezzo migliaia di dispositivi in grado di controllare relè a distanza, ma pochissimi offrono un'architettura serverless come quella descritta, e quasi nessuno è realizzato sfruttando servizi **gratuiti, open source e sicuri**.

Il sistema esposto consente inoltre al lettore di trarre spunto per creare altri infiniti dispositivi che hanno bisogno di dialogare con sistemi remoti, sfruttando sia dweet.io, sia le potenzialità di Ganimede e MicroPython. □



Cosa occorre?

La board Ganimede.E12 (cod. GANIMEDE-E12) è in vendita a Euro 13,00, il display OLED (cod. OLEDTVSCSD) a Euro 11,00, il relè (cod. RELAY1CH) a Euro 5,00. 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

NON RISCHIARE DI FORARE UN CAVO SOTTO TENSIONE!

UTILE PER TANTISSIMI UTILIZZI IN CAMPO
PROFESSIONALE E HOBBISTICO



€ 74,00
cod. FR850



RILEVATORE PROFESSIONALE DA PARETE

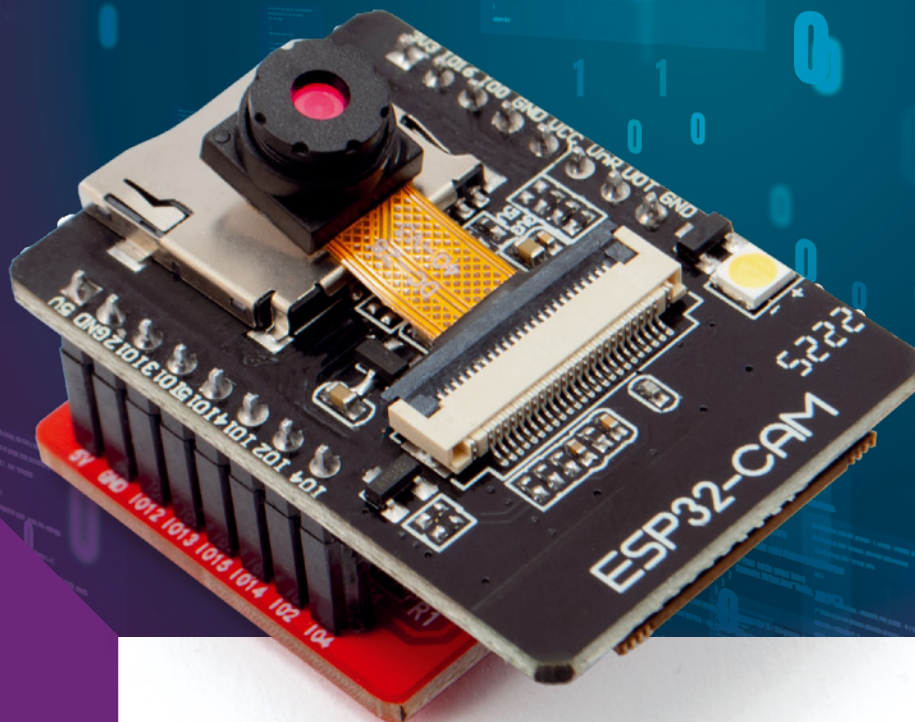
Particolarmente utile (ad esempio prima di praticare un foro su una parete) per rilevare la presenza di cavi elettrici sotto tensione, strutture in legno (telai, travetti, barre, ecc), oggetti metallici (travi, telai, tubi, tondini, perni, barre in metallo).



Prezzo IVA inclusa.

PROGRAMMATORE ESP32-CAM

di BORIS LANDONI



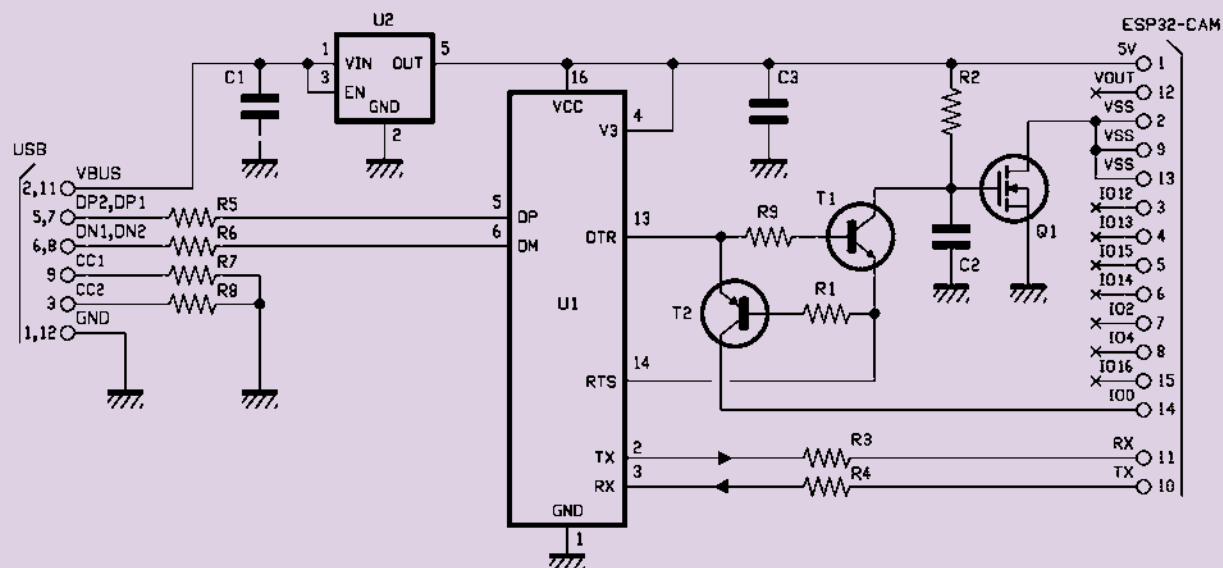
Ospita e supporta la programmazione dei moduli ESP32 con telecamera, fornendo un'interfaccia USB.

E

SP32-CAM Development Board è una speciale versione della popolare ESP-32, quindi una scheda di sviluppo e prototipazione basata su un SoC (System On Chip) ESP32, equipaggiata con una videocamera da 2MPixel (OV2640) di cui è possibile impostare il frame-rate, nonché provvista di uno slot per una

memoria miniSD fino a 4GByte.

Sulla scheda è presente un LED ad alta luminosità che può essere impiegato come flash nelle fotografie o come illuminatore per la scena da riprendere, ma vi sono anche alcune linee GPIO configurabili da firmware per interfacciarsi con il mondo esterno, oltre alle connessioni per le più comuni interfacce di comunicazione.



Già in passato abbiamo ospitato nelle pagine della nostra rivista diverse applicazioni con questo modulo ed ora torniamo sull'argomento proponendo un progetto che interesserà a molti di voi: un programmatore dedicato, interfacciabile via USB, che permette di sfruttare al meglio le librerie che la Espressif ha rilasciato per la programmazione attraverso l'IDE di Arduino, rendendo così più accessibile il modulo a chi un tempo si fermava di fronte alla necessità di procurarsi un sistema di sviluppo dedicato.

Il programmatore è sostanzialmente una scheda progettata per ospitare il modulo ESP32-CAM e interfacciarla ad un Personal Computer tramite USB per caricare gli sketch ed eventualmente un bootloader; questa board, che possiamo considerare un adattatore sia fisico che elettronico, permette di adattare l'interfaccia UART nativa del modulo a quella più comune e versatile che è la USB, gestendo alcuni segnali di controllo come RTS e DTR. Praticamente implementa quello che altrimenti andrebbe realizzato interfacciando la porta seriale (TXD/RXD) dell'ESP32-CAM con uno dei classici moduli convertitori USB to serial, risolvendo però a bordo anche il problema dell'adattamento dei livelli di tensione, che in alcuni moduli (per esempio l'FT782M della Futura Elettronica) non è possibile (cioè sul lato seriale lavorano a 0/5V, quindi su standard TTL) mentre in altri richiede l'impostazione di jumper (UC00A della Futura Elettronica).

SCHEMA ELETTRICO

Vedremo meglio questi dettagli analizzando lo schema elettrico del circuito, proposto in cima a questa pagina e centrato sull'integrato U1, che è un convertitore seriale/USB di tipo CH340C, molto diffuso perché utilizzato anche in alcune schede Arduino non originali. Il CH340 è un chip di conversione dal bus USB alla classica porta seriale bidirezionale con linee TX ed RX.

In modalità interfaccia seriale, CH340 fornisce il set di segnali per i tradizionali modem da linea commutata, utilizzato per estendere l'interfaccia seriale asincrona del computer o aggiornare direttamente il dispositivo affacciato con una seriale comune, mediante USB; per questo fa al caso nostro.

Il CH340 supporta la modalità USB full-speed conforme alle specifiche USB versione 2.0 ed è molto semplice da utilizzare e configurare, in quanto necessita solo di un quarzo e di due capacità esterni per il clock, ma può anche funzionare con l'oscillatore interno.

L'interfaccia seriale hardware è di tipo full-duplex, imposta il buffer del ricetrasmittitore e supporta vari valori di baud-rate, da 50 bps a ben 2 Mbps. Estrae e supporta il segnale di controllo della RS232 e quindi funziona da DTE (Data Terminal Equipment) disponendo dei segnali per il controllo dei comuni modem (DCE=Data Communication Equipment) che sono:

- RTS (Request To send), uscita utilizzata dal DTE per richiedere al dispositivo collegato (tipicamen-

- te un DCE) l'invio di dati; la linea RTS si connette alla CTS del dispositivo collegato;
- DTR (Data Terminal Ready) utilizzato per indicare al DCE che il terminale è pronto alla comunicazione;
 - DCD (Data Carrier Detect) che indica quando è stata rilevata la portante che i dati modulano per la comunicazione;
 - RI (Ring Indicator) che i modem utilizzano per avvisare il computer (o comunque il DTE dell'applicazione) dell'arrivo di una chiamata su linea commutata (PSTN);
 - DSR (Data Set Ready) impiegato per indicare al DTE che i dati sono pronti;
 - CTS (Clear To Send) che serve al DCE per comunicare al terminale (DTE) di essere pronto per l'invio dei dati.

Di queste linee utilizziamo RTS e DTR, come vedete dallo schema elettrico, sebbene la loro gestione non sia affidata in toto all'ESP32-CAM; naturalmente della porta UART del modulo usiamo il canale dati bidirezionale TX-RX.

Il lato USB dell'U1 è connesso al connettore USB Type-C (maschio da circuito stampato) per trasportare i segnali DP e DN del bus da e verso il convertitore; notate che avendo scelto un connettore USB tipo C e disponendo di un integrato che supporta l'USB 2.0, abbiamo configurato la connessione per l'utilizzo secondo questo protocollo e quindi vedete che i contatti DP1 e DP2 sono uniti e così pure DN1 e DN2. L'unione dei pin è dovuta al fatto che sono sdoppiati perché per come sono disposti, qualunque sia il verso di inserimento del connettore femmina volante, la polarità rimane invariata. Inoltre abbiamo portato a massa i contatti CC1 e CC2 del connettore, sia pure attraverso due resistenze di limitazione della corrente (rispettivamente R7 ed R8); ricordiamo che nello standard USB-C i pin CC1 e CC2 consentono la configurazione del canale dati (si chiamano, infatti, Configuration Channel) e, nello specifico, l'impostazione della modalità di trasporto dei dati, che nel nostro caso è quella conforme all'USB 2.0.

L'integrato CH340 si alimenta con una tensione continua di valore compreso fra 3,3 V e 5V; nel nostro circuito viene fatto funzionare con il potenziale fornito dal pin 5 (OUT) del regolatore integrato U2, un LDO siglato MIC5317-3.3, che partendo dalla tensione d'ingresso prelevata dalla linea VBUS del connettore USB-C (quindi 5 Vcc) ricava 3,3V ben stabilizzati.

Notate che, sebbene l'integrato preveda la possibi-

tà di porlo in standby o accenderlo tramite l'apposita linea EN, in questo caso lo manteniamo sempre attivo e ciò lo facciamo connettendo fisso il suo pin 3 al VIN (1).

La linea di alimentazione a 3,3V come del resto quella a 5V VBUS, viene adeguatamente filtrata dai disturbi impulsivi mediante i condensatori ceramici C1 e C3.

Andiamo ora sul lato seriale RS232 del CH341 e vediamo che il chip rende disponibili il canale dati formato dalle linee TX (uscita dati verso l'ESP32-CAM) ed RX (ingresso dati in arrivo dall'ESP32-CAM) oltre ai criteri di controllo della comunicazione che servono a sincronizzare il flusso dei dati; nello specifico il nostro circuito fa uso dei soli RTS e DTR, interconnessi attraverso una rete formata da due transistor bipolari e disposti in modo da interagire con il modulo ESP32-CAM che prenderà posto nel connettore omonimo (lo vedete a destra nello schema elettrico).

Il circuito formato dai transistor ha il duplice scopo di accendere il modulo ESP32-CAM mediante il MOSFET a canale N siglato Q1; essendo quest'ultimo di tipo enhancement-mode, il suo terminale di drain entrerà praticamente in contatto con il source quando tra gate e quest'ultimo verrà a trovarsi una tensione di almeno 3 volt, accendendo il modulo perché in tale condizione le linee Vss verranno portate a massa.

Questa condizione si verificherà ogniqualvolta la linea DTR si troverà a livello basso e la RTS ad uno logico, quindi durante l'invio dei dati, giacché l'RTS viene attivato quando è necessario inviare una richiesta di invio dati all'ESP32-CAM e il DTR è invece a zero in tale condizione; per comprendere meglio la cosa considerate che i segnali di controllo sono negati, quindi attivi a zero.

Analizziamo dunque le varie combinazioni possibili:



La scheda ospita l'ESP32-CAM grazie a due strip femmina.

L'ESP32-CAM Development Board è un modulo su PCB che integra un'interfaccia WiFi 802.11 b/g/n, una Bluetooth dual-mode (classica e Low Energy), una piccola telecamera da 2 Megapixel (OV2640), uno slot per una memoria miniSD fino a 4 GByte e 9 pin I/O; integra inoltre un LED ad alta luminosità collocato frontalmente, che può essere impiegato sia come flash che come illuminatore per la scena da riprendere. Programmabile direttamente tramite l'IDE Arduino, può essere utilizzato in varie applicazioni come in dispositivi domotici, controllo wireless industriale, monitoraggio wireless, identificazione wireless tramite QR code e molte altre applicazioni IoT (Fig. A).

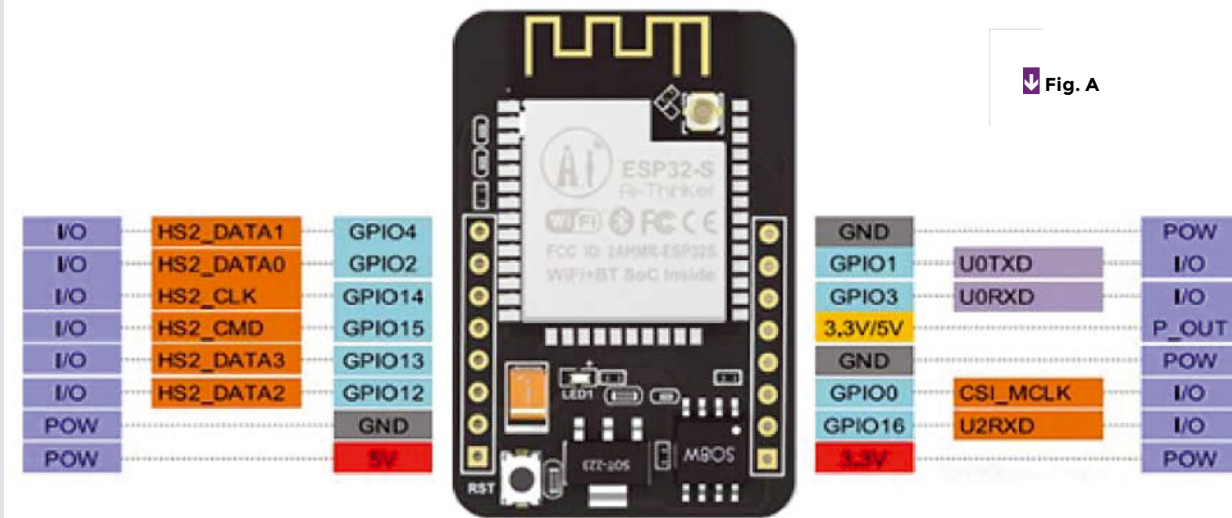
Il modulo integra un transceiver WiFi 802.11 b/g/n ed uno Bluetooth 4.2 LE con antenna su PCB capace di un guadagno di 2dBi (è comunque previsto un connettore u.FL); la potenza del trasmettitore WiFi è circa 17 dBm per lo standard 802.11b

(@11 Mbps) 14 dBm per l'802.11g (@54 Mbps) e 13 dBm per l'802.11n, mentre la sensibilità in ricezione è di -90 dBm in CCK a 1 Mbps e scende fino a -67 dBm in MCS7 (65 Mbps, 72.2 Mbps).

A livello di memoria, il SoC dispone di una flash SPI da 32Mbit, una PSRAM da 4MBit ed uno slot per mini SDcard che supporta fino a 4 GB. Il SoC può comunicare con l'esterno attraverso interfacce cablate UART (fino a 115.200 bps), SPI, I²C e rende disponibili 9 GPIO; supporta i protocolli di sicurezza WPA/WPA2/WPA2-Enterprise/WPS.

La telecamera di bordo (si tratta di una OV2640) è connessa tramite connettore FPC e restituisce immagini in formato JPEG (solo sulla OV2640), BMP e in scala di grigi; il modulo supporta anche la OV7670.

Il modulo è prodotto dalla A.I. Thinker, ma esistono in commercio soluzioni equivalenti di altre aziende.



quando DTR è a zero logico e RTS a livello alto, i transistor T1 e T2 sono interdetti entrambi, perché il primo (che è un NPN) ha l'emettitore positivo rispetto alla base ed il secondo (PNP) ha la base positiva rispetto all'emettitore; in queste condizioni il MOSFET è in stato di ON perché il suo gate viene portato a 3,3V (livello alto) dalla resistenza R2, quindi il suo canale è acceso.

Se il pin RTS passa a zero anch'esso, i transistor BJT sono ancora interdetti perché entrambi hanno una V_{be} nulla, essendo i loro terminali di base ed emettitore equipotenziali; lo stesso accade se RTS e DTR sono tutti e due a livello logico alto.

Invece se RTS assume lo stato zero quando DTR è a livello alto, la base del T1 diviene positiva rispetto all'emettitore ed il transistor conduce, trascinando a poche centinaia di millivolt il gate del MOSFET, il

quale andrà quindi in interdizione lasciando spegnere il modulo ESP32-CAM; anche T2 conduce, perché la sua base diviene negativa rispetto all'emettitore. Dunque, il modulo viene acceso quando almeno DTR è a livello basso, ovvero quando da USB il computer comunica all'ESP32-CAM di essere pronto a comunicare dati, a prescindere da quale sia lo stato del Request To Send.

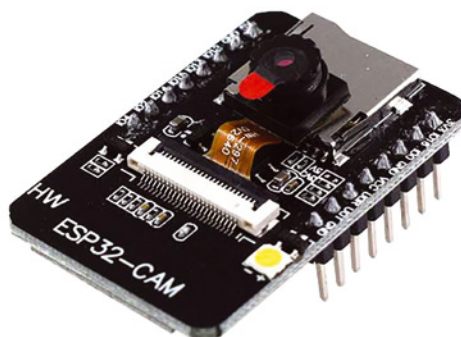
La funzione dei due transistor è quindi questa: T1 serve a controllare l'accensione del modulo ESP32-CAM intervenendo sulla polarizzazione di gate del MOSFET, quindi ogni volta che rimane interdetto (RTS a livello alto o DTR basso) il modulo viene alimentato, mentre se va in conduzione (RTS a livello basso e DTR nello stato logico alto) lo stesso viene privato della connessione di massa e si spegne. Quanto al T2, funziona da interruttore statico per

controllare la linea IO0, facente capo al pin 14 del connettore header della ESP32-CAM, la quale durante la programmazione deve essere tenuta a livello basso; quindi durante il periodo di caricamento del firmware il T2 deve rimanere interdetto, mentre negli altri momenti dev'essere in conduzione (condizione che si verifica quando il RTS è a zero logico e il DTR ad 1) per portare ad 1 logico (+3,3V) la predetta linea IO0.

I due pin U0TXD (TX) e U0RXD (RX) che permettono di caricare il nostro codice sul modulo ESP32-CAM sono esposti sul connettore esterno, sul quale è presente anche una seconda seriale U2RXD (RX) utilizzabile nel caso in cui serva ricevere dati tramite seriale; tali pin sono anche utilizzabili come GPIO, rispettivamente GPIO1 (U0TXD) e GPIO3 (U0RXD) nelle applicazioni in cui non occorra utilizzare l'UART. Per quanto riguarda la gestione da parte del Personal Computer, il chip CH340 utilizza dei driver software compatibili con il chip CH341, quindi la nostra scheda verrà riconosciuta dal sistema operativo Windows utilizzando direttamente il driver del CH341.

REALIZZAZIONE PRATICA

Bene, completata la descrizione dello schema elettrico andiamo a vedere come costruire questo programmatore, che si basa su un circuito stampato, molto piccolo, addirittura più di quello della ESP32-CAM, almeno per quel che riguarda la lunghezza; la larghezza è invece la stessa, perché sui lati vi troviamo le piazzole per ospitare gli strip femmina in cui verrà introdotto il modulo ESP. Il PCB è del tipo a doppia faccia e potete realizzarlo per fotoincisione



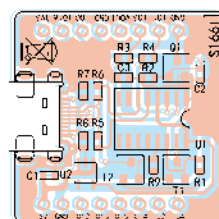
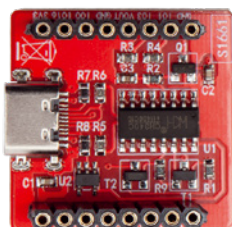
Il modulo ESP32-CAM dotato di pin-strip, visto dall'alto.

partendo dalle tracce lato rame che trovate pronte per il download nella pagina di presentazione di questa rivista sul nostro sito www.elettronica.in e che vanno stampate su carta da lucido o acetato per ottenere le relative pellicole. Una volta inciso e forato lo stampato, dovrete provvedere al montaggio dei pochi componenti, tutti in SMD, ragion per cui dovrete avere un po' di manualità e un minimo di attrezzatura, che consiste in una pinzetta per posizionare i vari elementi sulle piazzole, un saldatore a punta molto fine (o una stazione di saldatura ad aria calda, meglio nota come hot-air station) una lente d'ingrandimento, del filo di lega saldante del diametro massimo di 0,5 mm, nonché della pasta flussante da spalmare sulle piazzole prima di posizionare i componenti. Iniziate quindi centrando l'integrato CH340 nelle rispettive piazzole, dopo averlo orientato come mostra il piano di montaggio pubblicato in queste pagine, quindi stagnando un pin di un angolo e poi, accertato che il chip sia nella posizione corretta, uno dell'angolo opposto, così

piano di MONTAGGIO

Elenco Componenti:

R3, R4, R5, R6: 22 ohm
 R7, R8: 5,1 kohm
 R1, R9: 2,2 kohm
 R2: 4,7 kohm
 C1, C2, C3: 100 nF ceramico
 T1: BC817
 T2: BC807
 Q1: BSS138
 U1: CH340C
 U2: MIC5317-3.3YM5-TR
 USB: Molex USB connector 2169900002

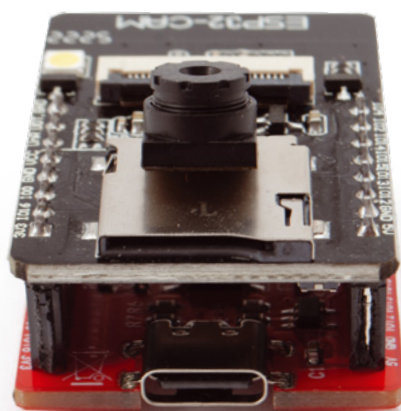


Varie:

- Pin strip female 8 way step 2,54mm
- PCB S1661 (60x56 mm)



L'ESP32-CAM va inserita in modo che il socket per SD e il connettore USB siano dallo stesso lato



da fermarlo; poi procedete stagnando i vari piedini, intervallando un lato con l'altro. Il CH340 è incapsulato in un package SSOP-20, ragion per cui pur essendo un SMD è abbastanza facile da saldare. Con la lente d'ingrandimento verificate di non aver unito accidentalmente piedini adiacenti e nel caso spalmate con abbondante pasta flussante gli stessi e poi sciogliete lo stagno della saldatura per rimuovere l'eccesso. Fatto ciò montate i componenti passivi, vale a dire resistenze e condensatori, quindi il connettore USB-C (con mano ferma e precisione perché ha i pin molto vicini) e l'integrato MIC5317-3.3, da orientare come mostrato nel piano di montaggio; notate che in ogni caso realizzando il circuito stampato con la nostra traccia lato rame, il regolatore U2 ha un verso di montaggio obbligato, giacché è incapsulato in package SOT-32 a tre terminali ed è quindi asimmetrico.

Ponete cura nello stagnare abbondantemente le alette di ancoraggio del connettore USB-C, per fare in modo che resista alle inserzioni e disinserzioni della spina del cavetto nel normale utilizzo.

Completate il montaggio della scheda inserendo e saldando nelle piazzole laterali gli strip femmina a passo 2,54 mm che vi permetteranno di inserire (rispettando il verso indicato nel piano di montaggio) la scheda ESP32-CAM, che dovete già aver provvisto degli opportuni strip maschi da saldare nelle piazzole laterali. Ricordiamo che la ESP32-CAM va orientata in modo che il lato dello zoccolo per SD-Card sia rivolto dalla parte del connettore USB Type C della demoboard, come mostrato chiaramente dalla **Fig. 1** in questa pagina.

Una volta completate le saldature e verificato che sia tutto a posto, la scheda del programmatore è pronta per assolvere alla propria funzione: potete

quindi inserirvi la vostra ESP32-CAM, badando all'orientamento (state attenti che gli strip femmina sono in ugual numero e disposti parallelamente, quindi il modulo potrebbe anche entrare girato) connetterla ad un Personal Computer mediante un cavo USB/A-microUSB e iniziare a gestire i firmware, ovvero a programmare il modulo.

Rammentiamo che sulla linea di alimentazione, quindi dal connettore USB-C, l'insieme della scheda del programmatore e dell'ESP32-CAM assorbe un massimo di 360 milliampere: questa corrente può essere erogata da qualsiasi porta USB, ragion per cui non fatevi problemi a collegare il programmatore a un PC dotati di comuni porte USB mediante un adattatore, giacché non è obbligatorio utilizzare porte di tipo USB-C.

I driver per il converter USB/seriale si possono scaricare da Internet alla pagina www.wch-ic.com/downloads/CH341SER_EXE.html

CONCLUSIONI

Vi abbiamo presentato una scheda molto utile a chi vuole programmare il popolare modulo ESP32-CAM, con telecamera e interfaccia WiFi, senza difficoltà e sfruttando un circuito in grado di gestire l'accensione e lo spegnimento mediante i segnali di controllo della porta seriale; una scheda che peraltro incorpora il convertitore USB/seriale con segnali, lato UART, compatibili con i livelli del SoC ESP32, ossia 0/3,3V. Ricordiamo a chi vuole lavorare con l'ESP32-CAM che sul web, alla pagina <https://www.futurashop.it/ESP32-CAM-development-board-2846-ESP32-CAM>, è disponibile una nutrita documentazione corredata da firmware di prova per WiFi e telecamera di bordo e una serie di altre utili risorse. □



Cosa occorre?

Il programmatore per ESP32-CAM (cod. FT1661M) è disponibile a Euro 16,00 mentre la telecamera (cod. ESP32-CAM) è disponibile a Euro 16,50. I prezzi si intendono IVA compresa.

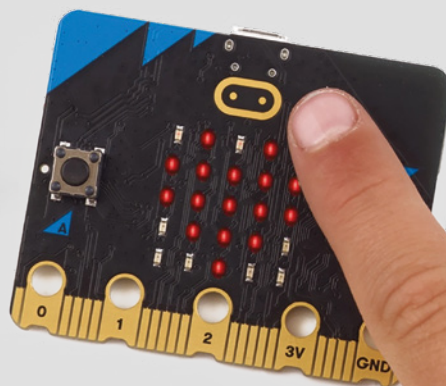
Il materiale va richiesto a:

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

Easy to Program

MICRO:BIT È UNA PICCOLA SCHEDA DI SVILUPPO CHE INTEGRA UN MICROCOMPUTER A 32 BIT; PUÒ ESSERE UTILIZZATA PER IMPARARE LE BASI DELLA PROGRAMMAZIONE ED È ADATTA A TUTTE LE ETÀ.

Prezzi IVA inclusa.



cod. MICROBITGOV2

€37,00

UTILIZZALA SUBITO CON IL NUOVO SET!

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

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

ROBOT MAQUEEN LITE

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



Blu



Bianco



Rosa



Verde

Saldare diventa più facile!

Supporto per saldatura con 6 bracci flessibili e accessori



cod. **HR2120**
€ 39,00

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

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

Postazione multifunzione



cod. **TIP019**
€ 48,00

Supporto per saldatura a 4 bracci flessibili con clip a coccodrillo



cod. **SHH4ARMS**
€ 23,90

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

Terza mano con lente, luce e portasaldatore



cod. **TP232**
€ 17,00

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

Terza mano con lente d'ingrandimento



cod. **VTHH**
€ 7,90

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

Supporto per serraggio scheda

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



cod. **VTHH6**
€ 12,00

Prezzi IVA inclusa

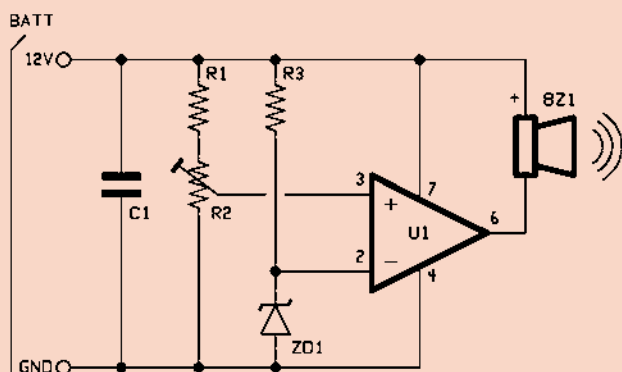
ALLARME BATTERIA SCARICA

di DAVIDE SCULLINO

Ci avvisa acusticamente quando rileva che la tensione dell'accumulatore di bordo si è abbassata sotto la soglia di carica minima, aiutandoci a non rimanere appiedati...

Sebbene oggi nelle automobili ci sia tanta ed anche "troppa" elettronica, succede ancora di rimanere a piedi perché la batteria non viene caricata correttamente o perché ha perso gradualmente buona parte della propria capacità e non si carica più come dovrebbe, pur alimentata correttamente dal generatore di bordo.

Se è vero che nella gestione elettronica degli autoveicoli moderni siano implementati accorgimenti per monitorare lo stato della batteria, è vero altrettanto che l'analisi sia della quantità di energia entrata (in carica) e uscita (in scarica) dall'accumulatore sia della tensione che si può ritenere quella più corretta per il generatore, sono affidate alla logica e quantificate numericamente e pertanto in modo discreto. Questo porta a non ricevere alcuna segnalazione fin quando



l'accumulatore non è quasi scarico o il generatore non è guasto.

Una valida e semplice soluzione per conoscere, senza doversi affidare ciecamente al display multifunzione di bordo e agli strumenti del quadro, che dipendono dal firmware della ECU dell'auto, lo stato della batteria, è montare un voltmetro collegandolo direttamente alla batteria dell'auto o all'ingresso della scatola fusibili.

Un'altra è realizzare un circuito come quello proposto in queste pagine, che rileva la tensione della batteria ed emette un avviso acustico quando, in base alla taratura che vien fatta, la ritiene insufficiente a garantire il buon funzionamento dell'impianto elettrico e soprattutto all'avviamento del motore una volta che verrà arrestato.

Rilevare la tensione della batteria è utile, soprattutto a motore fermo, perché consente di fare una valutazione qualitativa sulle condizioni dell'accumulatore, perché infatti in base alla tensione che essa presenta a riposo è possibile comprenderne lo stato e predire eventuali guasti.

SCHEMA ELETTRICO

Andiamo dunque a spiegare in cosa consiste la nostra soluzione, analizzando insieme lo schema elettrico del piccolo circuito che vi proponiamo di realizzare e di alimentare con due fili collegati a positivo e negativo dell'impianto elettrico, preferibilmente sulla batteria.

I fili serviranno sia ad alimentare il circuito, sia ad effettuare la lettura della tensione dell'accumulatore dell'auto per comprendere se attivare o meno la segnalazione acustica.

Infatti il circuito funziona proprio per effetto della polarizzazione di un comparatore ad opera di una rete a partitore resistivo derivata dalla linea di alimentazione.

Premettiamo che il circuito qui proposto è stato progettato per impianti a 12V, quindi è da utilizzare su auto e camper o furgoni con batteria a 12V: non va bene per i camion e per gli autobus, il cui impianto elettrico prevede una batteria da 24V (tipicamente due da 12 volt poste in serie).

Come vedete dallo schema elettrico qui a fianco, quel che vi proponiamo è l'essenza della semplicità, perché consta di un comparatore basato su un amplificatore operazionale in tecnologia bipolare (un comune e collaudatissimo LM741, un vero e proprio baluardo dell'elettronica integrata) la cui tensione di riferimento è stabilizzata tramite il diodo Zener ZD1, mentre la tensione di ingresso (filtrata dal condensatore C1 per spegnere eventuali disturbi impulsivi che potrebbero causare involontarie commutazioni del comparatore) viene fornita dal cursore di un trimmer (si tratta dell'R2 che vedete nello schema elettrico in cima a questa pagina), a sua volta alimentato dalla linea di ingresso (BATT - 12V e GND).

Il comparatore opera nella configurazione non-invertente (quindi la sua uscita assume il livello alto quando l'ingresso raggiunge un potenziale che supera quello di riferimento) e il suo riferimento è stabilito in 5,1V dal bipolo R3-ZD1: non deve sembrarvi strano, perché per poter commutare con tensioni di ingresso inferiori ai 12V è opportuno che il riferimento sia ben al disotto di tale valore. Per rendere più agevole ed anche molto precisa la regolazione della soglia, abbiamo previsto un trimmer multigiri (R2); quest'ultimo fa partitore con R1 in modo da presentare al piedino 3 del comparatore U1 un potenziale che può variare tra un minimo di zero (quando il cursore è tutto a massa) e un massimo di poco superiore ai 10,7 V con 12V in ingresso.

Quando la tensione presentata dal cursore del trimmer R2 all'ingresso non-invertente dell'U1 è tale da superare i 5,1V che fanno da riferimento al piedino 2, l'uscita del comparatore si porta a livello alto e il cicalino tace perché non viene alimentato (la differenza di potenziale ai suoi capi è praticamente nulla perché l'uscita dell'integrato è più o meno al potenziale del piedino 7); non appena la tensione della batteria scende abbastanza da far presentare tra il cursore dell'R2 e massa una tensione minore dei 5,1V stabilizzati dal diodo Zener, il piedino 6 dell'LM741 si porta a circa 1V determinando una differenza di potenziale tale da determinare l'alimentazione del cicalino con elettronica BZ1, il quale inizia a suonare avvisandoci del calo della tensione della batteria e comunque dell'ab-

bassamento della tensione dell'impianto elettrico del veicolo al disotto del valore che ritenete critico. Il trimmer multigiri va tarato per fare in modo che in corrispondenza della tensione considerata normale della batteria il cicalino taccia; spiegheremo tra breve come fare, ma in linea di massima considerate che una batteria al piombo quale quella dell'automobile, non alimentata, inizialmente presenta una tensione di 2,1V per elemento (mediamente 12,5V) che scendono a circa 12V dopo qualche minuto dall'interruzione della carica. Sotto gli 11,5V può considerarsi quasi scarica. Per essere più esatti, tipicamente, con batteria alla temperatura ambiente di 25°C e a riposo da un paio d'ore, la tensione rilevabile è 12,5V a piena carica e in piena efficienza (accumulatore nuovo o quasi), di 12,2 V a metà della carica e di meno di 11,7 V in condizione di batteria prossima alla scarica. Facciamo notare che l'intero circuito assorbe pochissima corrente anche quando l'allarme acustico è attivo, quindi non va a condizionare in alcun modo lo stato della batteria dell'auto, che anche quando è praticamente scarica ha energia sufficiente per far suonare il cicalino per qualche ora; non c'è quindi il rischio che l'allarme smetta di svolgere il suo ruolo perché la batteria è scarica e quindi dovrebbe dare l'avviso acustico per impedirvi di tentare l'avviamento e scaricare quel poco di energia rimasta. Lasciare un po' di energia residua nella batteria è importante perché permette di avviare il veicolo a traino o a spinta, mentre se l'accumulatore è completamente scarico, considerato che l'intero motore è gestito da centraline elettroniche, queste non avrebbero quel minimo di energia per consentire l'avviamento; e non ci sarebbe neppure quel poco che serve a fornire all'alternatore la corrente di eccitazione che, durante l'avviamento a spinta o a traino, permetterebbe al generatore stesso di produrre elettricità e bastare al bisogno dell'elettronica di gestione del motore.

REALIZZAZIONE PRATICA

Bene, esaurita la spiegazione dello schema elettrico passiamo ora alla costruzione e all'utilizzo del circuito avvisatore, la cui semplicità è tale da poterlo realizzare persino su un pezzetto avanzato di basetta millefori; comunque per chi desiderasse il circuito stampato e fosse in grado di realizzarlo, abbiamo disegnato uno specifico PCB (lo vedete montato nelle foto proposte in queste pagine) la cui traccia lato rame è scaricabile dalla sezione Download delle pagine di presentazione di questo

numero.

Si tratta di uno stampato a singola ramatura composto da poche piste e un numero esiguo di piazzole, quindi davvero alla portata di tutti, anche di chi ha da poco preso la strada dell'autocostruzione anche a livello hobbistico.

Per realizzarlo potete anche semplicemente tracciare direttamente su una basetta ramata monofaccia le piste e le piazzole con l'apposito pennarello indelebile per circuiti stampati; se invece volete ricorrere alla fotoincisione, stampate su carta da lucido o acetato il file del "bottom copper" (cioè la traccia lato rame) e ottenete così la pellicola per impressionare la basetta ramata presensibilizzata. Vedete un po' voi cosa preferite e ciò che avete a disposizione.

Comunque abbiate deciso di procedere, una volta inciso e forato il PCB potete saldarvi i pochi componenti occorrenti a iniziare dalle resistenze fisse e dal diodo Zener, che orienterete come mostrato nel piano di montaggio visibile in queste pagine; inserite e saldate quindi lo zoccolo per l'integrato LM741, da orientare preferibilmente come proposto nel piano di montaggio, poi passate alla morsettiera bipolare di alimentazione e al cicalino piezo, da collocare secondo la polarità indicata perché essendo del tipo con elettronica integrata ha una sua polarità da rispettare. Terminate le saldature innestate l'integrato nel suo zoccolo, allineando la tacca di riferimento come indicato dal piano di montaggio visibile in queste pagine e verificando che durante l'inserzione non vi sia alcuno degli otto piedini piegato sotto il corpo del 741.

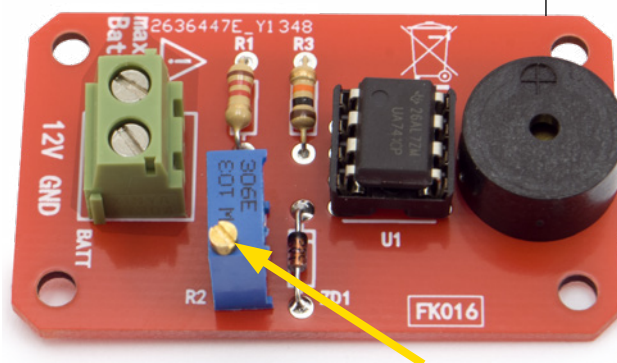
TARATURA E INSTALLAZIONE

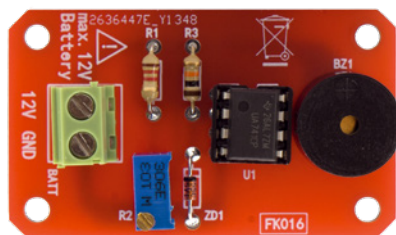
Ora il vostro circuito è pronto, però va tarato affinché fornisca delle indicazioni veritiere, ovvero

CARATTERISTICHE TECNICHE

- ◆ **Tensione di alimentazione:**
10÷15V
- ◆ **Corrente max assorbita:**
15 mA
- ◆ **Tensione di soglia allarme**
impostabile tra 0 e 10,7V
- ◆ **Avviso acustico tramite**
cicalino piezo

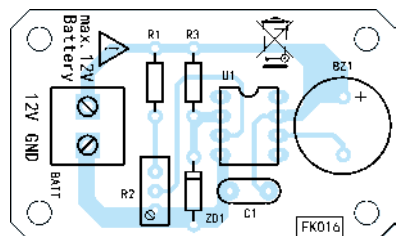
Il trimmer permette di regolare la soglia di allarme; a fine regolazione bloccate la vite con smalto o colla.





Elenco Componenti:

R1: 1,2 kohm
R3: 4,7 kohm
R2: Trimmer multigiri 10 kohm
C1: 100 nF ceramico
ZD1: Zener 5,1V 400mW
U1: LM741
BZ1: Buzzer 12V con elettronica



Varie

- Zoccolo 4+4
- Morsetto 2 vie passo 5.08mm
- Circuito stampato FK016 (51x31 mm)

fornisca l'allarme solo quando la batteria è da ritenersi veramente scarica.

Per tarare il dispositivo collegate la morsettiera d'ingresso (alimentazione) con due fili (rispettando la polarità) all'uscita di un alimentatore stabilizzato capace di erogare almeno 12V (la corrente erogabile è irrilevante), ponendo in parallelo i puntali di un multimetro disposto sulla misura di tensione continua con fondo-scala di almeno 20V, quindi regolate l'uscita dell'alimentatore su quello che volete sia il valore della tensione che ritenete sia quella di batteria scarica.

Fatto ciò, con un piccolo cacciavite a lama regolate il cursore del trimmer fino a quando non sentirete il cicalino suonare. A quel punto fermatevi e scollegate l'alimentatore (spegnetelo perché non vi serve più); idem per il tester: il vostro circuito ora è pronto per essere installato.

Nell'eseguire la taratura prestate attenzione al fatto che le auto moderne hanno l'alternatore che può erogare da 11,8 a 14,8V, quindi il trimmer non va mai regolato sopra gli 11,8V perché tale

tensione può essere considerata normale, almeno in determinate condizioni operative.

Per l'installazione, identificate un luogo della plancia che fa al caso vostro perché consente di sentire bene l'avviso acustico: può anche essere un vano esistente inutilizzato (purché elettricamente isolato) ma nulla vieta di inserire il circuito in un contenitore di plastica forato per far entrare il cavo bipolare di alimentazione (suggeriamo di utilizzare una piattina rosso-nera da 2x0,5 mmq) e per lasciar uscire il suono del cicalino.

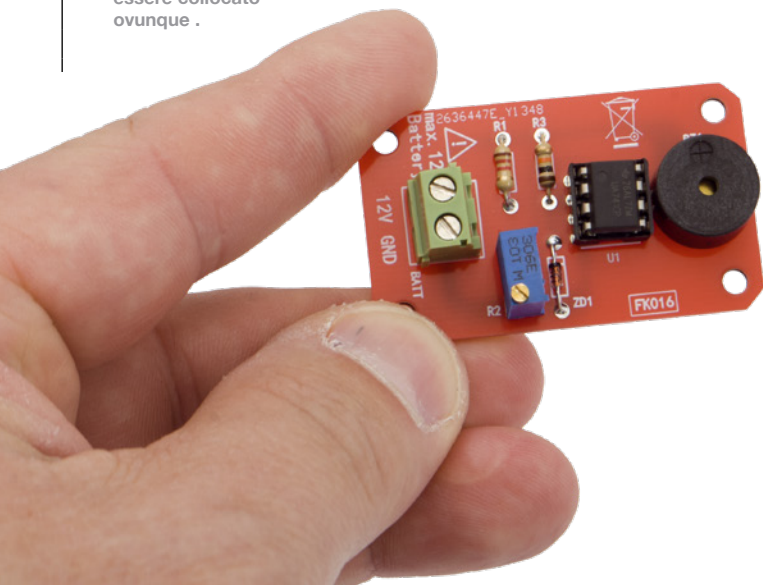
Se montate il circuito stampato senza contenitore, assicuratevi che il vano dove lo mettete non sia in metallo né esponga contatti dell'impianto elettrico dell'automobile, altrimenti isolatelo con del nastro isolante o della spugna sintetica.

Come potete notare dalle foto del prototipo, il nostro circuito è talmente piccolo e compatto che potete posizionarlo praticamente ovunque all'interno dell'abitacolo, senza difficoltà e senza dovervi scervellare su dove collocarlo.

Prima di chiuderlo in un contenitore e comunque di installarlo in auto, bloccate la vite di regolazione del trimmer con una goccia di colla o dello smalto per le unghie, in modo che non si sposti a seguito delle vibrazioni presenti in auto durante la marcia, modificando indesideratamente la soglia di intervento dell'allarme.

Il cavo di alimentazione del circuito deve essere connesso preferibilmente ai capi della batteria, ovvero all'ingresso della scatola fusibili principale; non è una cattiva idea porre in serie al filo positivo (il rosso della piattina rosso-nera) un portafusibile 5x20 volante con all'interno un fusibile cilindrico 5x20 da 500 mA ritardato: questo accorgimento proteggerà l'impianto elettrico dell'auto in caso facciate cortocircuito inavvertitamente sul PCB o tra il PCB e una parte metallica del veicolo.

Il PCB è così piccolo da poter essere collocato ovunque.



Cosa occorre?

Il circuito di allarme per batteria scarica (cod. FK016) è disponibile in scatola di montaggio presso Futura Elettronica a Euro 8,00 IVA compresa.

Il materiale va richiesto a:

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

CONCLUSIONI

Bene, con questo abbiamo concluso la descrizione di un utile accessorio per ogni tipo di autoveicolo dotato di impianto elettrico a 12 volt; ricordiamo che il dispositivo può comunque essere utilizzato

come generico segnalatore di "undervoltage" ossia per avvisare se su una linea elettrica sulla quale viene collegato la tensione va al disotto di un determinato valore.

Insomma, le sue applicazioni sono molteplici e limitate solamente dalla tensione nominale dell'impianto, che deve essere compresa fra 9 e 15V. ■

LO STATO DELLA BATTERIA LO DICE LA ECU

Nelle auto moderne la condizione della batteria viene valutata dalla ECU motore (electronic control unit), che si avvale di un sensore posto di norma tra il morsetto del polo negativo e il cavo che porta alla messa a terra sulla scocca; tale sensore può funzionare ad effetto di Hall oppure mediante uno shunt di piccolissimo valore resistivo: nel primo caso il rilevamento avviene misurando il campo elettromagnetico prodotto dal conduttore costituito dalla lamina in rame che dal morsetto negativo porta al filo di collegamento alla scocca (quindi senza accoppiamento galvanico con la batteria) mentre nel secondo si realizza misurando la caduta di tensione sullo shunt. Il sensore può anche essere formato da una bobina toroidale dentro la quale passa il cavo negativo della batteria.

In tutti i casi, è integrato nel sensore un circuito elettronico in grado di convertire il valore di corrente in una serie di dati caratterizzati da un segno, indicante la polarità, nonché da un modulo; i dati vengono trasmessi alla ECU attraverso, ancora una volta, una connessione LIN.

La Fig. A propone il tipico sensore (Hella IBS - Intelligent Battery Sensor).

Il sensore comunica alla ECU se la batteria sta erogando o assorbendo corrente; calcola e comunica anche l'intensità della corrente stessa, in modo che la ECU calcoli l'energia assorbita in carica ed erogata, determinando altresì lo stato di carica (SOC).

Il calcolo viene eseguito considerando che ogni valore positivo indica che la batteria sta assorbendo corrente dall'alternatore, mentre un valore negativo è indice che sta cedendo elettricità, ad esempio se l'alternatore non è sufficiente o il motore è in avviamento; il calcolo del valore assoluto permette invece di conoscere l'intensità. Questo permette di capire se la batteria ha bisogno di essere caricata o meno: se dal momento in cui inizia l'acqui-

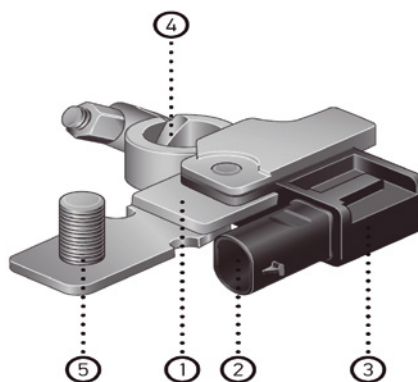
sizione dei dati l'energia erogata è uguale a quella assorbita in carica la ECU riterrà la batteria carica e quindi manterrà la tensione dell'alternatore anche al disotto dei 12,5V perché non sarà necessario ricaricarla. Se invece l'energia erogata sarà maggiore di quella incamerata, la ECU forzerà il regolatore di carica ad aumentare la tensione fornita dall'alternatore, tanto più quanto maggiore risulterà la differenza, arrivando anche a 14,6-14,8 volt.

Dalla tensione assunta dalla batteria sia in carica che a motore spento, la ECU, tramite appositi algoritmi, valuta lo stato e quindi l'idoneità o meno dell'accumulatore, predicendo guasti imminenti; per esempio analizza la curva di discesa della tensione nel periodo dopo lo spegnimento del motore o l'abbassamento della tensione del generatore.

▼ Fig. A

Fig. 1 - Sensore batteria IBS:

- 1) shunt;
- 2) connettore LIN;
- 3) elettronica;
- 4) morsetto batteria;
- 5) attacco cavo negativo.



Fai il pieno di energia!

BATTERIA RICARICABILE ULTRACELL AL PIOMBO SENZA MANUTENZIONE

Batterie al piombo senza manutenzione, ad alta densità d'energia e a bassissima autoscarica, prodotte dalla ULTRACELL. Per uso generico, la loro robustezza e affidabilità le consentono un ampio campo di applicazioni: solare, antifurto, backup, UPS, illuminazione d'emergenza ecc. La modalità di ricarica consigliata è quella a corrente costante.



BATTERIE RICARICABILI AL NI-MH VARTA "READY TO USE"



€ 14,50
Cod. HR03AAA800



€ 15,80
Cod. HR6AA2100



€ 13,80
Cod. HR14C3000

Prezzi IVA inclusa.

**FUTURA
ELETTRONICA®**

www.futurashop.it

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

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

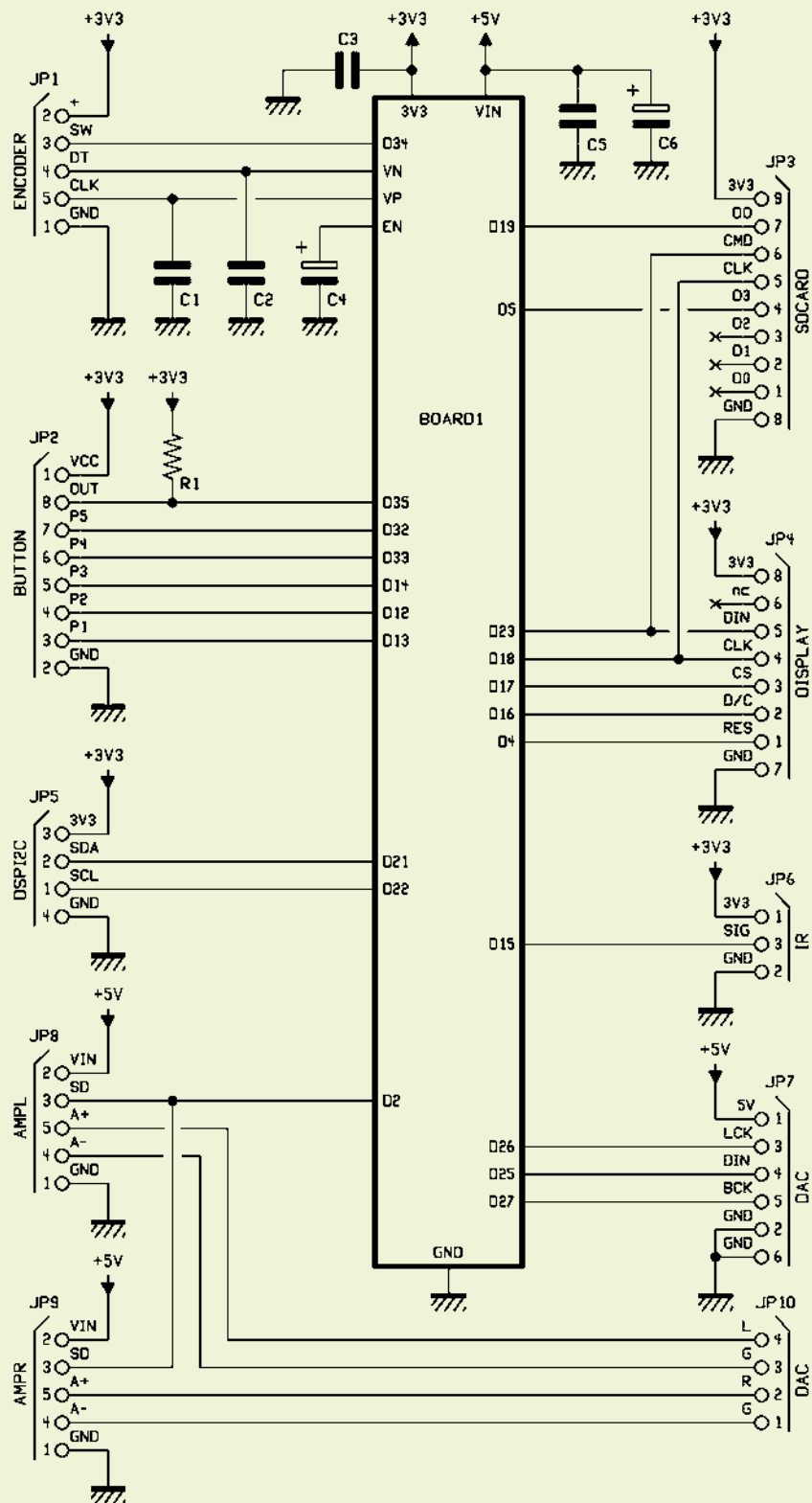
ESP32 MUSIC PLAYER

dell' ING. MIRCO SEGATELLO



Lettore musicale con ESP32 in grado di riprodurre brani da Internet radio, file MP3 salvati localmente su una scheda SD e brani salvati su un server.

L'ascolto di musica è da sempre una passione universale, che ci accompagna a casa, in viaggio o durante il lavoro, senza mai mancare. Anche se la musica è una costante nella vita di tutti i giorni quello che è cambiato è il modo di ascoltarla, dai dischi in vinile si è passati ai CD per finire ultimamente con la musica liquida, che giunge a casa nostra direttamente dalla rete internet. Con la diffusione delle tecnologie digitali, complice la comodità di fruizione, la musica liquida sta soppiantando tutte le altre tecnologie, da quelle più obsolete come le musicassette a quelle più recenti come il CD, anche se rimane ancora un mercato di nicchia per le tecnologie vintage come gli LP ed i giradischi.



Ma cosa intendiamo con il termine “musica liquida”? Con musica liquida si definiscono essenzialmente i file musicali che non dispongono di un supporto di memorizzazione fisico dedicato (come CD e musicassette), bensì sono disponibili in rete ovvero nel cloud.

Il successo di questa nuova tecnologia è subito evidente, la musica può essere riprodotta in qualsiasi momento e luogo, anche in mobilità disponendo di un comune smartphone, inoltre è facilissimo riprodurre un particolare brano di uno specifico cantante semplicemente digitando alcuni tasti. La mancanza di un supporto fisico ha permesso la nascita di servizi di streaming come ad esempio Spotify, Tidal, Qobuz, tramite i quali è possibile disporre di un catalogo musicale praticamente infinito. Con questa tecnologia ciascun brano viene codificato in un file, che deve presentare specifiche caratteristiche, i cui parametri fondamentali sono: il numero dei bit usati per la codifica che può andare da 16 a 32bit, la frequenza di campionamento che può andare da 44.1 a 384KHz ed il bit rate, ovvero il flusso in bit al secondo. Il riferimento per capire la qualità di riproduzione è il CD, in esso il brano è codificato a 16bit con frequenza di 44.1KHz senza alcuna compressione, ed è rimasto a lungo la massima qualità audio disponibile. Su di un supporto da 700MB (classico CD) si possono salvare circa 70minuti di musica, l'equivalente potrebbe essere un file con estensione WAV (standard di formato digitale per file audio sviluppato da Microsoft e IBM) che, essendo non compresso, presenta il problema dell'ingombro.

Sin da quando sono stati commercializzati i riproduttori musicali portatili il problema è stato quello di ridurre lo spazio occupato dai file, ci pensarono gli esperti della MPEG con l'algoritmo di compressione MP3 a risolvere il problema permettendo una compressione da 4 a 10 volte rispetto l'originale. Si tratta di una codifica lossy perché è presente una perdita di dati e quindi una qualità inferiore, ma il beneficio nel ridurre il numero di byte necessari è tale da decretarne un successo planetario; a breve comunque andrà in pensione e sostituito dalla nuova codifica AAC (Advanced Audio Coding). La qualità dei file MP3 viene definita dal relativo bit-rate che parte da 64Kbit/sec per i brani di qualità inferiore sino a 382Kbit/sec per i brani di qualità superiore, comunque ben lontani dai 1440Kbit/sec della qualità CD. Con l'affinamento degli algoritmi di compressione e la necessità di garantire un'elevata qualità audio ultimamente si parla sempre più spesso di audio HiRes, nel quale

il numero di bit è elevato a 24 o 32 e la frequenza di campionamento può essere portata sino a 192KHz. Per ottenere un file di dimensioni accettabili è necessario applicare una compressione che però in questo caso è di tipo lossless, ovvero senza perdita di informazione, garantendo una qualità audio superiore al CD. Attualmente il file più comune che dispone di questa codifica è il FLAC che in circa 20MB comprime un brano in qualità CD, contro i 3-5MB di un file MP3. Anche la radio si è evoluta e alle normali trasmissioni via etere in AM o FM quasi tutte le stazioni radio trasmettono in streaming sul web il loro palinsesto con un vantaggio non da poco: non è più necessario essere vicini all'antenna trasmittente, ma basta un accesso alla rete internet e si possono ascoltare in digitale praticamente tutte le stazioni radio del globo, in modo gratuito. Il lettore audio che andiamo a descrivere servirà per l'appunto a riprodurre la vostra musica liquida, sia essa disponibile in internet, in rete locale e o su supporto SD. Non saremo in grado di riprodurre musica in qualità HiRes ma la qualità sarà comunque più che soddisfacente.

HARDWARE

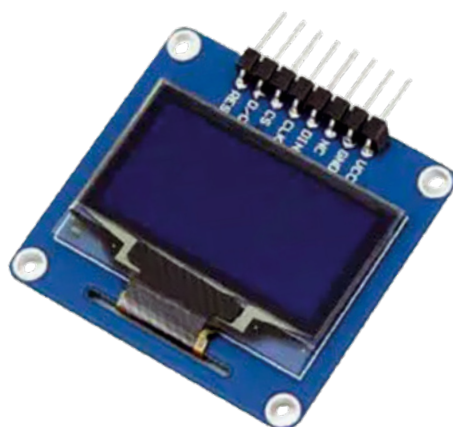
Dopo questa disamina sui formati audio siamo pronti ad entrare nel concreto di questo progetto e lo facciamo ricordando quando nel lontano 2017 su questa stessa rivista (per la precisione nel numero 214) veniva presentato un progetto di una radio FM chiamato Open Radio. Ovviamente non potevamo esimerci dal presentare un progetto più aggiornato e sofisticato che potesse sfruttare la disponibilità della musica liquida, che per sua natura è fruibile da diverse fonti sia in locale che in rete, pertanto dobbiamo disporre di un hardware connesso. Dopo aver preso in considerazione diverse soluzioni, valutando costi e reperibilità dei componenti la scelta è caduta sull'utilizzo di una scheda basata su microcontrollore ESP32, che già include un modulo Wi-Fi interno ed ha capacità elaborativa per gestire i vari codec audio, inoltre è ben supportata da diverse librerie che ci permetteranno di ridurre il carico di lavoro nello sviluppo del software. Vogliamo la possibilità di poter accedere alle infinite internet radio, ma vogliamo anche poter accedere ai nostri brani preferiti in locale, memorizzati all'interno di una SD Card, ed anche accedere anche ai brani memorizzati in un server di rete sfruttando il protocollo UPnP. Il protocollo Universal Plug and Play (UPnP) rende automatico il rilevamento dei dispositivi su una rete locale, consentendo a diversi dispositivi sulla stessa rete



→ **Fig. 1**
Board ESP32.



→ **Fig. 2**
Display OLED.



di individuarsi a vicenda e di comunicare automaticamente tra loro, pertanto se nella vostra rete domestica è presente un server multimediale compatibile con il protocollo UPnP il nostro lettore musicale potrà accedere ai brani in esso memorizzati. La comodità è data dal fatto che qualsiasi altro dispositivo della rete potrà fare la stessa cosa con il vantaggio che tutto il vostro catalogo musicale è concentrato in un solo luogo e potrà essere aggiornato più facilmente.

La scheda ESP32 non è sufficiente per realizzare un completo lettore musicale pertanto dobbiamo completare l'hardware con diversi altri componenti il cui elenco è visibile nella **Tabella 1** e la cui reperibilità è affidata alla ditta Futura Elettronica (www.futurashop.it). La board ESP32 (**Fig. 1**) è il cuore del progetto, è disponibile con il nome ESP32-DOIT-DEV-KIT-v1 e viene fornita già assemblata e completa di convertitore USB/seriale per la comunicazione tra PC e MCU e dispone già dello stabilizzatore di tensione per alimentare il modulo ESP alla tensione di 3,3V.

Ci servirà poi un display per la visualizzazione delle informazioni sul funzionamento del lettore, abbiamo scelto un economico display OLED perché abbiamo previsto la funzionalità di stand-by, durante la quale il lettore funzionerà da orologio e come si sa i display OLED sono quelli a più basso consumo, perfetti per questa funzione (**Fig. 2**).

Una pulsantiera è d'obbligo se vogliamo interagire con il lettore ed attivare le classiche funzioni di skip brano, menù, selezione sorgente e spegnimento. La scelta è caduta su una scheda già assemblata che necessita di soli 3 pin di collegamento. Con

→ **Tabella 1**
Elenco componenti.

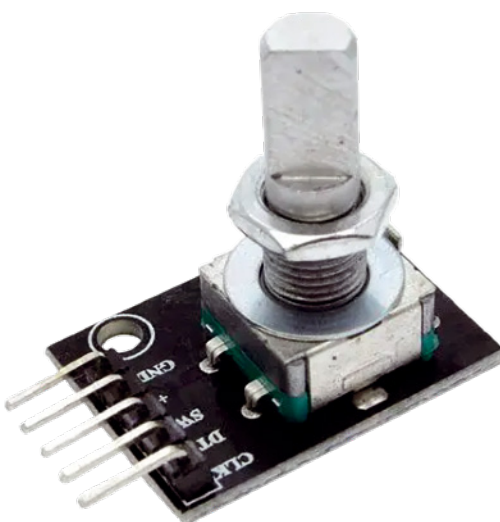
COMPONENTE	CODICE
ESP32 – Scheda di sviluppo WIFI + bluetooth	8300-YB555
Micro SD Storage Board	2850-MICROSDSTORAGE
Breakout con Encoder rotativo 12 mm	8300-YC093
Display Oled 1.3" pollici SPI	2850-OLEDA
Scheda a pulsanti	3085-BUTTONBOARD
Modulo Amplificatore Classe D 2,5 watt	2846-PAM8302A
Telecomando e ricevitore IR	1606-HX1838
PCM5102A I2S DAC Decoder GY-PCM5102 I2S	2846-I2SPCM5102A
Altoparlante 3 W / 4 ohm – Ø 52mm	2846-AP5243W
Coppia altoparlanti da pannello – 5W / 8 ohm	2850-SPEAKER8

questa scheda potremmo gestire 5 pulsanti indipendenti tramite una sola linea del microcontrollore, vediamo in dettaglio come questo si possa fare. Normalmente ciascun pulsante viene connesso ad una linea digitale del microcontrollore e ad ogni pressione il relativo livello logico cambia di stato, invece in questa board i pulsanti sono connessi ad un partitore resistivo in modo che alla pressione di ciascun di essi venga generata una distinta tensione. È pertanto sufficiente usare una linea analogica e discriminare il livello di tensione letto per valutare quale pulsante sia stato premuto. La scheda è già assemblata permettendo di semplificare la realizzazione pratica del progetto (**Fig. 3**).



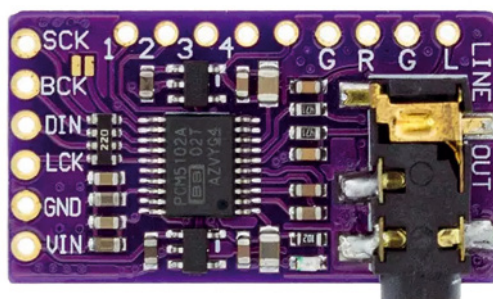
← **Fig. 3**
Pulsantiera.

A supporto dei pulsanti abbiamo previsto un encoder rotativo che risulta più comodo per l'impostazione del volume e la selezione di alcune funzioni come la scelta del brano e l'accesso al menù. Anche in questo caso abbiamo scelto una versione già assemblata in una piccola schedina con incluse le resistenze di pull-up, inoltre l'encoder è fornito di un piccolo contatto, attivato premendo la sua manopola (**Fig. 4**).



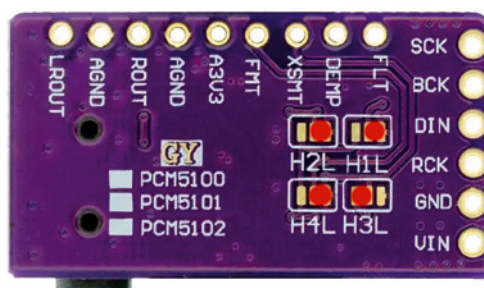
← **Fig. 4**
Encoder rotativo

Un altro componente fondamentale per il funzionamento del lettore è il DAC (digital to analog converter) in quanto la musica liquida è in formato digitale e per poterla ascoltare è necessario convertirla in un segnale analogico (che è la natura con cui si presentano le onde sonore). Il modulo ESP32 dispone di due DAC ma sono a bassa risoluzione, mentre per assicurare la massima qualità audio è necessario un componente più performante, nel nostro caso di sigla PCM5102 fornito già assemblato in una piccola breakout board; questo componente è di così elevata qualità che lo troviamo come DAC all'interno del lettore CD ROTEL CD11 Tribute (**Fig. 5** e **Fig. 6**). Questa è la minima configurazione ammessa per far funzionare il lettore musicale, per l'ascolto è sufficiente utilizzare un paio di cuffie che saranno connesse alla presa jack disponibile nella scheda del DAC, oppure potete usare delle casse amplificate.



← **Fig. 5**
Convertitore DAC.

Sul lato inferiore di questo componente sono presenti delle piazzole di saldatura che impostano alcune funzioni interne del convertitore, munitevi pertanto di saldatore ed aggiungete una piccola goccia di stagno sulle piazzole come evidenziato nella **Fig. 6**.



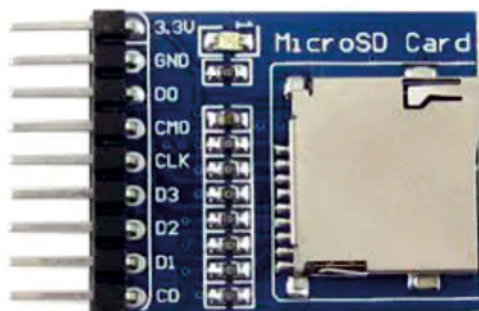
← **Fig. 6**
Saldatura piazzole lato inferiore convertitore DAC.

Fate molta attenzione perché le piazzole sono molto piccole ed è necessaria una certa precisione per evitare di fare corto con la piazzola vicina. Per completare l'hardware possiamo aggiungere

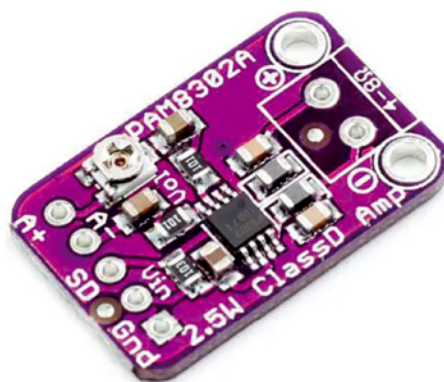
→ Fig. 7
Ricevitore
infrarossi.



→ Fig. 8
SD card reader.



→ Fig. 9
Modulo
amplificatore.



→ Fig. 10
Altoparlante full
range.



un ricevitore ad infrarossi con telecomando che permetterà di gestire le principali funzioni del lettore, quando ad esempio siamo comodamente sdraiati nel divano. (Fig. 7).

Se aggiungiamo un lettore di schede SD Card possiamo anche salvare i brani in locale ma potremmo anche salvarci delle playlist personalizzate delle internet radio preferite. Si tratta di un modulo (che accetta micro SD) con interfaccia SPI, completo di tutto il necessario per essere interfacciato con un microcontrollore (Fig. 8).

Per rendere il sistema completamente autonomo abbiamo previsto due piccoli amplificatori in classe D (uno per ciascun canale) in grado di fornire sino a 2,5W ciascuno (Fig. 9). Sono basati sull'integrato PAM8302, dispongono di un trimmer per la regolazione del volume, bassa distorsione armonica totale e basso rumore (THD+N) per una maggiore fedeltà audio, inoltre dispongono di protezione contro corto circuito e protezione termica, inoltre sono a basso consumo e basse emissioni EMI.

La cosa importante per il nostro progetto è che questi moduli dispongono del pin di abilitazione di nome SD, pertanto sarà possibile portare l'integrato PAM8302 in modalità a basso consumo, quando vorremmo mettere l'intero sistema in standby; la corrente assorbita scende al valore irrisorio di 1µA. Per concludere aggiungiamo due altoparlanti full range con diametro di 5mm e potenza di 3W (Fig. 10), oppure in alternativa una coppia altoparlanti da pannello (5W / 8 ohm) come quelli di codice SPEAKER8.

Per l'alimentazione nulla di più semplice, un comune alimentatore per smartphone con spina micro USB da 1A andrà più che bene. Questo ci garantisce un elevato rendimento del sistema ed un consumo quasi irrisorio in modalità stand by.

REALIZZAZIONE PRATICA

Terminata la descrizione del circuito, descriviamo ora la realizzazione pratica. Per connettere tutti i componenti sopra elencati abbiamo previsto uno stampato con dei connettori, uno per ciascuna periferica. Per la realizzazione dello stampato consigliamo di scaricare i file gerber dalle pagine di presentazione di questo numero, i quali possono essere usati usufruendo del servizio di realizzazione PCB di Futura Elettronica all'indirizzo web.ftlink.it/pcb.

Quando avrete il PCB pronto, sarà sufficiente saldare solo pochi componenti ed alcuni connettori strip come visibile in Fig. 11. Saldare i condensatori poliestere e a seguire i connettori, per ultimi

i condensatori elettrolitici che di norma sono a più alto profilo. I componenti che non fanno parte dell'interfaccia utente (board ESP, SD card, convertitore DAC e amplificatori), possono essere inseriti direttamente nei relativi strip come visibile in **Fig. 12**. I rimanenti componenti vengono connessi con semplici jumper già terminati femmina-femmina come quelli di codice JUMPERFF40P distribuiti da Futura Elettronica, la disposizione delle connessioni rispettano il connettore presente nelle rispettive schede, fate riferimento alla **Fig. 13** per gli esatti collegamenti.

Nella **Fig. 14** potete vedere il lavoro ultimato con un esempio di come saranno posizionati i componenti dell'interfaccia utente fissati sul pannello frontale del contenitore che racchiude il tutto. In questo prototipo, abbiamo utilizzato un piccolo pannello stampato in 3D, ma è solo un esempio di quello che si potrebbe fare, lasciamo al lettore massima libertà nella fase realizzativa. Il lettore di SD card è stato inserito nel rispettivo connettore nella scheda principale, con alcuni jumper è però possibile svincolarlo dal PCB e posizionarlo in una zona più comoda da raggiungere da parte dell'utente.

Verificate con scrupolo i corretti collegamenti e che tutti i jumper siano ben inseriti nei rispettivi connettori, questo semplificherà il collaudo finale. Gli altoparlanti sono fissati sul pannello posteriore, ma nulla vieta di inserirli lateralmente oppure di

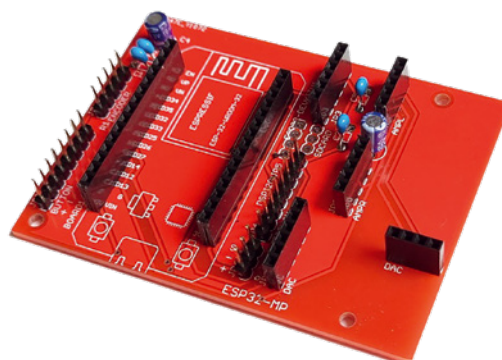


Fig. 11
Circuito stampato senza i moduli.

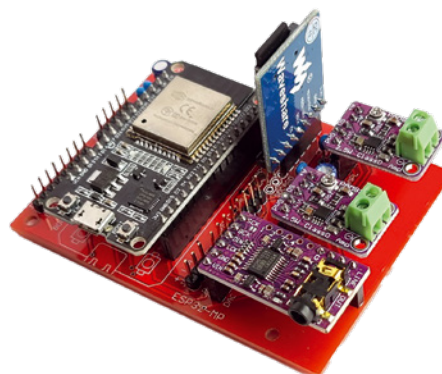


Fig. 12
Circuito stampato con inseriti i componenti non facenti parte dell'interfaccia utente.

Fig. 14
Circuito completato con tutte le periferiche.

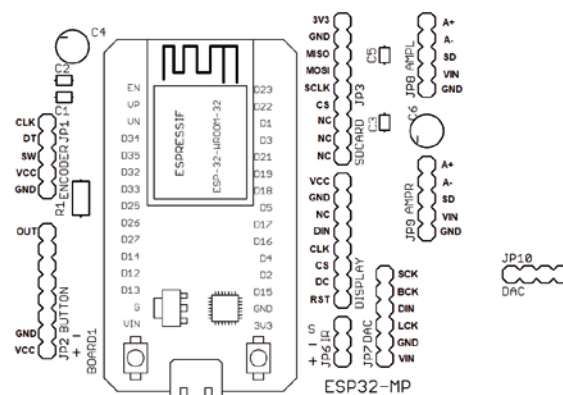
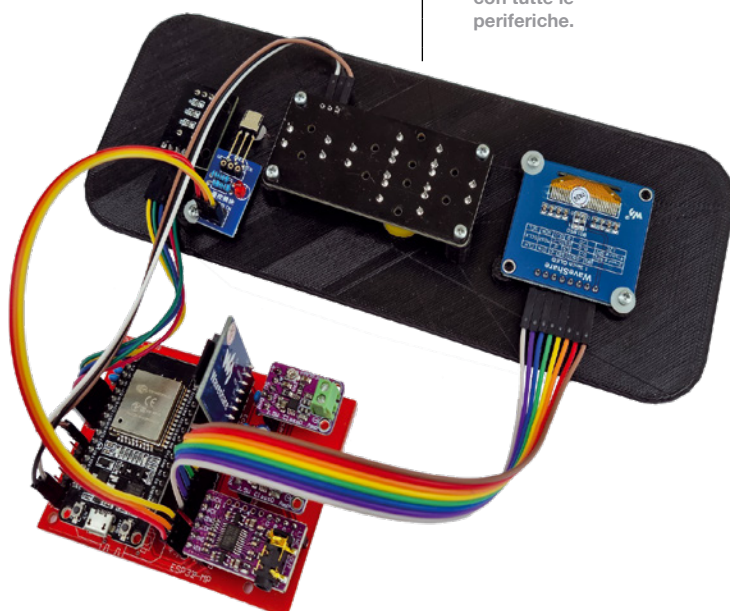


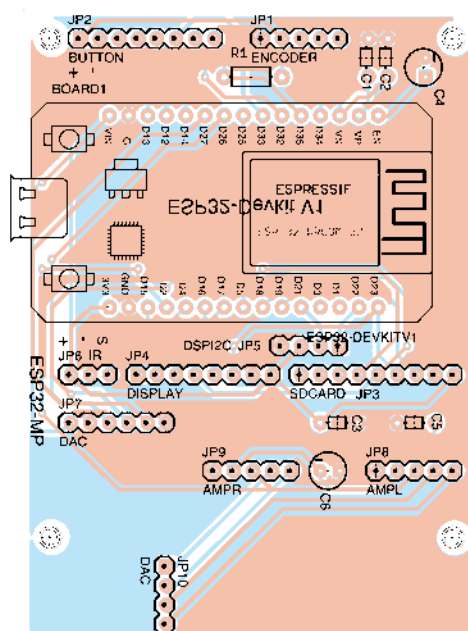
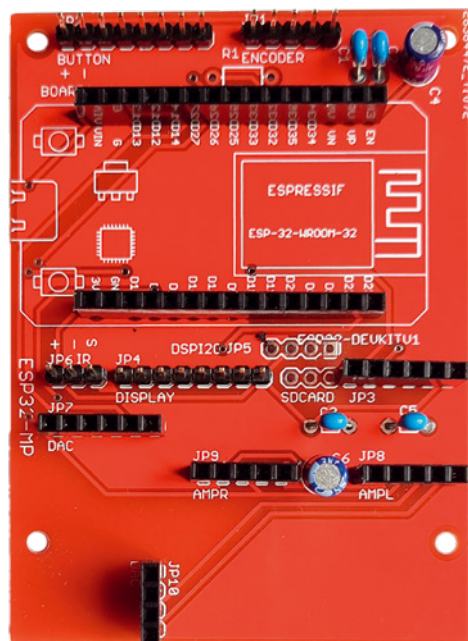
Fig. 13
Disposizioni delle connessioni sul circuito stampato.

Elenco Componenti:

- R1: 10 kohm
- C1, C2, C3, C5: 100nF
- 63 VL poliestere
- C4, C6: 10µF 16 VL elettrolitico
- JP6: Strip maschio 3 vie
- JP1: Strip maschio 5 vie
- JP2, JP4: Strip maschio 8 vie (2 pz.)
- JP10: Strip femmina 4 vie
- JP8, JP9: Strip femmina 5 vie (2 pz.)
- JP3, JP7: Strip femmina 6 vie (2 pz.)
- BOARD1: Strip femmina 15 vie (2 pz.)

Varie:

- Scheda ESP32 (cod. YB555)
- Micro SD Storage Board (cod. MICROSDSTORAGE)
- Breakout Encoder (cod. YC093)
- Display Oled 1.3" pollici (cod. OLEDA)
- Scheda a pulsanti (cod. BUTTONBOARD)
- Modulo Amplificatore (cod. PAM8302A)
- Telecomando e ricevitore IR (cod. HX1838)
- DAC Decoder (cod. I2SPCM5102A)
- Altoparlante 3 W / 4 ohm (AP5243W)
- Altoparlanti da pannello 5W/8ohm (cod. SPEAKER8)
- Circuito stampato S1698



creare un pannello frontale più ampio e fissarli anteriormente in bella vista. La soluzione da noi proposta è assai compatta, tanto che la parte di amplificazione è risultata il punto debole della qualità audio finale, già l'ascolto con le cuffie permette di ottenere un suono di maggiore qualità. Nulla vieta di utilizzare un amplificatore audio di qualità superiore e soprattutto con una maggiore potenza, in questo caso sarà necessario ricorrere ad una alimentazione con una tensione (e corrente) più alta. Quest'ultima variante non prevede l'integrazione con la funzione di stand-by che invece avevamo deciso di implementare e che, come vedremo nella descrizione del software,

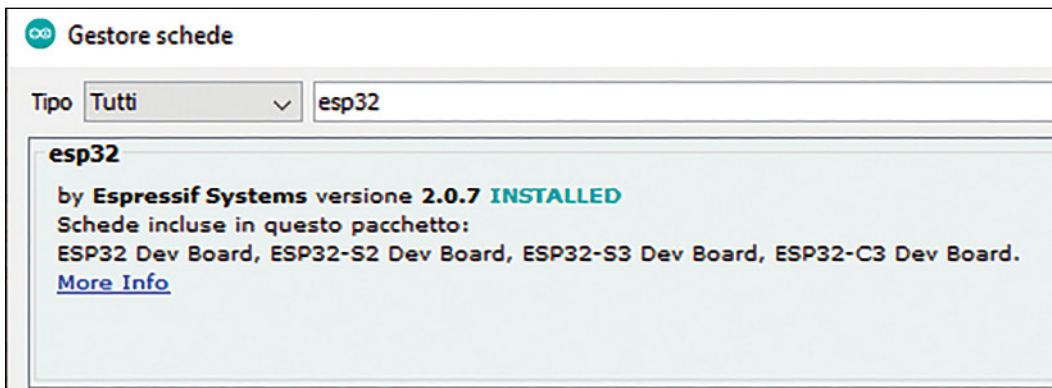
prevede di porre in modalità di basso consumo tutte le periferiche non essenziali. In ogni caso è necessario prevedere un apposito cabinet che non sia di tipo metallico perché andrebbe ad interferire con la trasmissione Wi-Fi del modulo ESP, infatti la scheda da noi utilizzata non prevede un connettore per l'utilizzo di un'antenna esterna ma è presente solo l'antenna stampata sul PCB. Per il contenitore consigliamo materiali tipo legno o plastica con una dimensione tale da accogliere tutti i componenti ed i piccoli altoparlanti.

IL FIRMWARE

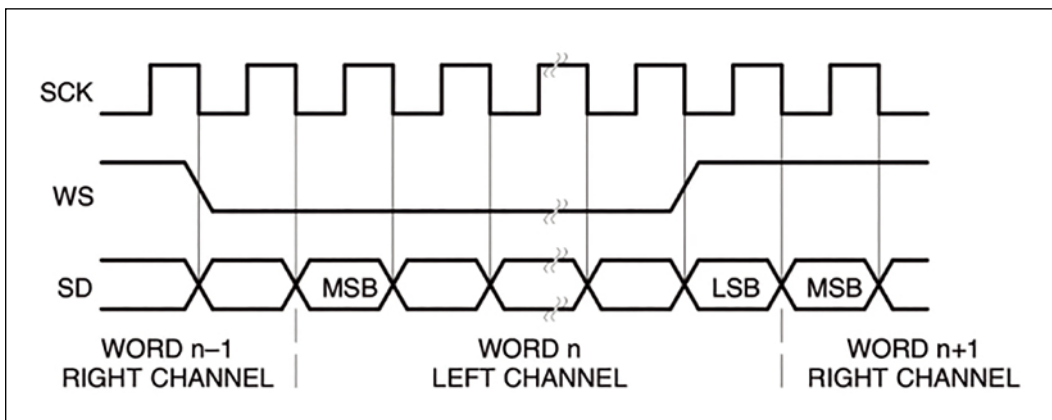
Occupiamoci ora del codice dell'ESP32, delle

Fig. 15
Aspetto esterno
del lettore
multimediale.





← **Fig. 16**
Installazione
pacchetto per
ESP32.



← **Fig. 17**
Segnali del
BUS I2S.

diverse funzioni implementate e del suo utilizzo pratico. Il firmware è stato sviluppato su piattaforma Arduino IDE sfruttando SDK messo a disposizione dalla ditta Espressif Systems produttrice del chip ESP, pertanto la prima operazione è quella di installare proprio il supporto per i moduli ESP32, nell'ambiente di lavoro di Arduino. Accediamo alle impostazioni di Arduino IDE e nel campo URL aggiuntive per il gestore schede inseriamo il link https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json, successivamente recatevi nel gestore schede ed installate il pacchetto per ESP32 come visibile in **Fig. 16**.

Installazione librerie

Il passo successivo è l'installazione di alcune librerie che sono servite da supporto per implementare alcune funzioni complesse, infatti una peculiarità del software di Arduino è quello di poter integrare il programma principale con altri listati sviluppati da terze parti e specificatamente scritti per svolgere specifiche funzioni. La prima libreria di cui abbiamo bisogno si chiama *ESP32-audioI2S* ed è la più importante perché si occupa del flusso audio e della sua decodifica, può essere scaricata dal sito

<https://github.com/schreibfaul1/ESP32-audioI2S>.

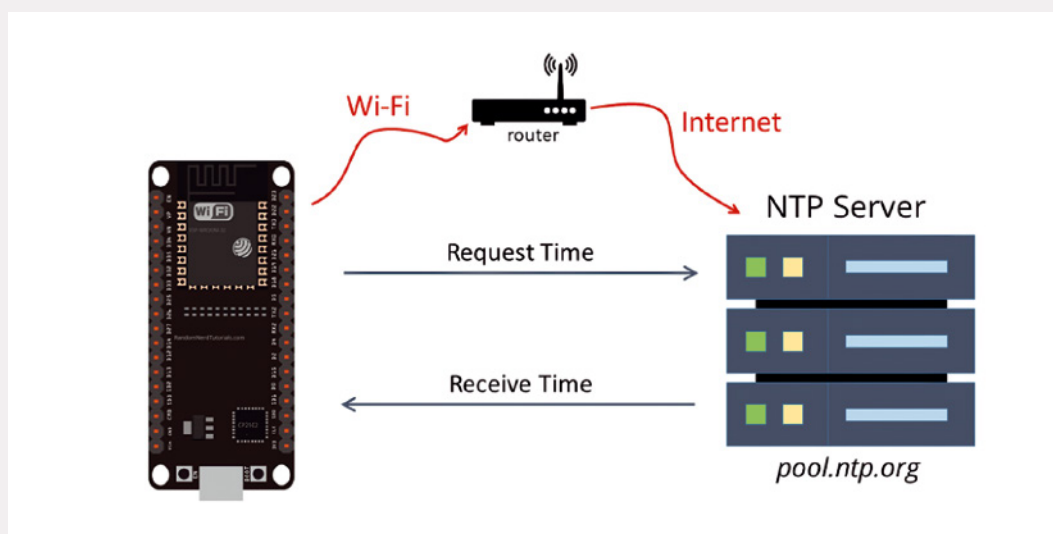
Questa libreria si occupa di leggere il flusso dei dati digitali del brano in ascolto dalla sorgente specificata tramite un percorso di rete (internet o server locale) ma può gestire anche un flusso di dati locali da una SD Card.

Le potenzialità di questa libreria sono veramente notevoli, tanto che sarebbe possibile salvare i brani in un server cloud e riprodurli direttamente da lì, ma può anche gestire playlist nel formato m3u, asx e pls. Questa libreria può gestire file WAV, AAC, MP3 e FLAC (quest'ultimo formato solo se il modulo ESP dispone di memoria PSRAM). Il flusso digitale viene reso disponibile sulla porta I2S del modulo ESP32 e sfrutta il protocollo I2S (Inter-IC Sound), introdotto per la prima volta da Philips Semiconductor nel 1986, per trasferire dati audio digitali PCM tra dispositivi elettronici.

L'interfaccia hardware I2S è costituita da:

1. La linea Serial Clock (SCK o SCLK) di solito scritto come "bit clock (BCLK)" rappresenta il clock principale che fornisce informazioni sulla cadenza di invio di ogni singolo bit.
2. La linea LRCLK o Frame Sync (FS), fornisce informazione in relazione a quale canale (1 o 2)

→ **Fig. 18**
Funzioni dei pulsanti del telecomando.



- fa riferimento il dato in transito.
3. La linea dati SDATA, SDIN, SDOUT che rappresenta il dato vero e proprio in transito.

Il segnale I2S (**Fig. 17**) viene poi convertito in segnale audio analogico dal convertitore DAC ad alta risoluzione connesso esternamente al modulo ESP proprio sulla porta I2S.

A supporto della libreria audio includiamo nel progetto le librerie FS.h e SD.h che non necessitano di essere installate perché librerie di sistema e già incluse nell'SDK dell'EPS32. Queste librerie servono per poter accedere ai brani in locale contenuti della SD card e forniscono funzionalità per leggere cartelle e nomi dei file ivi salvate.

Altre importanti librerie sono utilizzate per la gestione del display, si tratta delle librerie Adafruit_GFX.h, Adafruit_SSD1306.h e Adafruit_SH1106.h. Siccome in commercio si trovano diversi modelli di display OLED potrebbe capitare di disporre già di un display a casa anche se non necessariamente

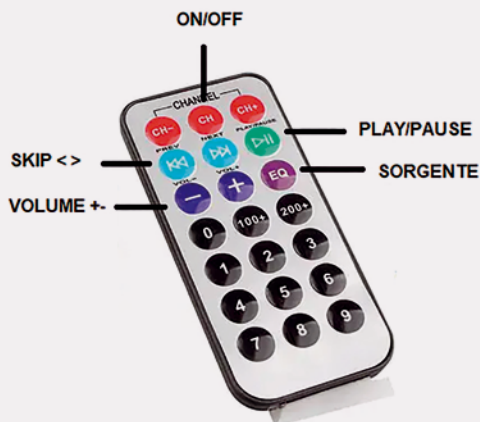
proprio quello da noi indicato (si veda l'articolo precedente), abbiamo previsto pertanto la possibilità di selezionare tra 4 diversi display con differenti chipset e diversa porta di comunicazione; nella sezione iniziale del software lasciate de-commentata la riga relativa al vostro display.

```

/***** Select display type *****/
// #define OLED_SSD1306I2C      //0,96" HW I2C
// #define OLED_SH1106I2C      //1,30" HW I2C
#define OLED_SH1106SPI         //1,30" HW SPI
// #define OLED_SSD1309SPI      //2,42" HW SPI

```

In tutti i casi viene usata un'interfaccia hardware sia essa I2C che SPI anche se quest'ultima è decisamente la più veloce e questo ci garantisce di non interferire con il processo di decodifica audio, assai energivoro in fatto di elaborazione matematica ed impiego della CPU. Non dimentichiamo infatti, che la CPU del modulo ESP32 deve svolgere tutte le funzioni, lettura del flusso dati, decodifica e gestione dell'interfaccia utente. Tutti i display indicati sono di tipo grafico, lasciandoci la massima libertà di scelta dei font e delle icone da rappresentare, permettendoci di ottenere un'interfaccia di una certa complessità. Sempre in tema di interfaccia utente utilizziamo la libreria ESP32Encoder.h reperibile all'indirizzo <https://github.com/madhephaestus/ESP32Encoder> la quale ci fornirà tutte le funzioni per la gestione dell'encoder rotativo, che utilizzeremo per impostare rapidamente il volume ma anche per muoverci più agevolmente all'interno del menù, questo grazie anche al suo pulsante integrato per il quale dobbiamo utilizzare la libreria Button2.h reperibile



→ **Fig. 19**
Sincronizzazione data e ora tramite server NTP.

all'indirizzo <https://github.com/LennartHennigs/Button2>. Una pulsantiera è d'obbligo per le funzioni classiche di un lettore audio come lo skip del brano, la scelta della sorgente e l'accesso al menù. Per queste funzionalità utilizziamo una pulsantiera analogica composta da 5 pulsanti che gestiremo con la libreria `AnalogMultiButton.h` scaricabile dal sito <https://github.com/dxinteractive/AnalogMultiButton>. Questa libreria implementa anche un valido sistema per impedire che i rimbalzi elettrici dei pulsanti possano fornire falsi segnali, inoltre non impegna in modo particolare la CPU e la memoria RAM.

Per completare l'interfaccia utente usiamo la libreria `IRremote.h` (reperibile dal sito <https://github.com/Arduino-IRremote/Arduino-IRremote>) che ci aiuterà nella gestione dei segnali ad infrarossi forniti dal modulo IR e relativo telecomando, con le funzionalità riportate in **Fig. 18**. L'abbinamento tra pulsanti e funzioni svolte la trovate nella sezione di programma iniziale e che, per comodità, riportiamo di seguito:

```

/***** IR pin and code constant *****/
const byte IR_PAUSE   = 67;
const byte IR_NEXT    = 64;
const byte IR_PREV    = 68;
const byte IR_INCVOL  = 21;
const byte IR_DECVOL  = 7;
const byte IR_ONOFF   = 70;
const byte IR_SOURCE  = 9;

```

Abilitando il debug avanzato ad ogni pressione dei tasti del telecomando sarà visualizzato su Serial Monitor il corrispondente codice, sarà sufficiente annotarsi il codice del pulsante di interesse e sostituirlo nell'elenco sopra riportato. La portata del telecomando è essenzialmente influenzata da come lo avrete installato nel case del lettore, per la massima funzionalità è importante che il sensore sia ben visibile dall'esterno e libero da ogni impedimento.

Come anticipato nella prima parte dell'articolo, non è previsto un interruttore di accensione bensì un pulsante per la gestione della modalità a basso consumo quando non usiamo il lettore, in questo caso verrà visualizzata a video la data e l'ora, per questa funzionalità ci appoggiamo alla libreria `time.h` già disponibile nel package della board ESP32. Spendiamo alcune parole sulle funzionalità implementate da questa libreria che sono davvero molto interessanti. Sappiamo che il nostro lettore è connesso alla rete e questo è tutto quello che

ci serve per implementare un validissimo e molto preciso orologio con datario senza la necessità di aggiungere alcun hardware come ad esempio un RTC (real time clock) clock. La libreria `time` sfrutta l'hardware del modulo ESP, in particolare il clock di sistema, per mantenere aggiornata data e ora di un orologio, ma la particolarità di quella di implementare un client NTP per la sincronizzazione automatica dell'orologio. I server NTP (Network Time Protocol) sono usati dai diversi dispositivi della rete per sincronizzarsi in modo preciso, inviando semplicemente a chi ne faccia richiesta la data e l'ora attuali; nell'istante in cui leggete questo articolo è molto probabile che il vostro PC si stia proprio sincronizzando usando un server NTP (**Fig. 19**). Per il nostro programma usiamo il server di nome `pool.ntp.org`, ma per ottenere data e ora occorrono alcuni piccoli calcoli che tengano conto del fuso orario e dell'eventuale ora legale. La libreria si occupa anche di tutti questi passaggi e riesce a mantenere aggiornata data e ora anche per parecchio tempo senza la necessità di alcuna sincronizzazione.

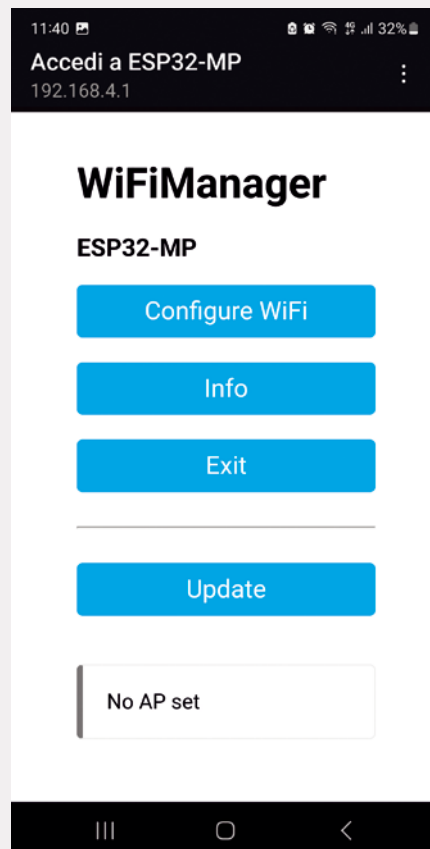
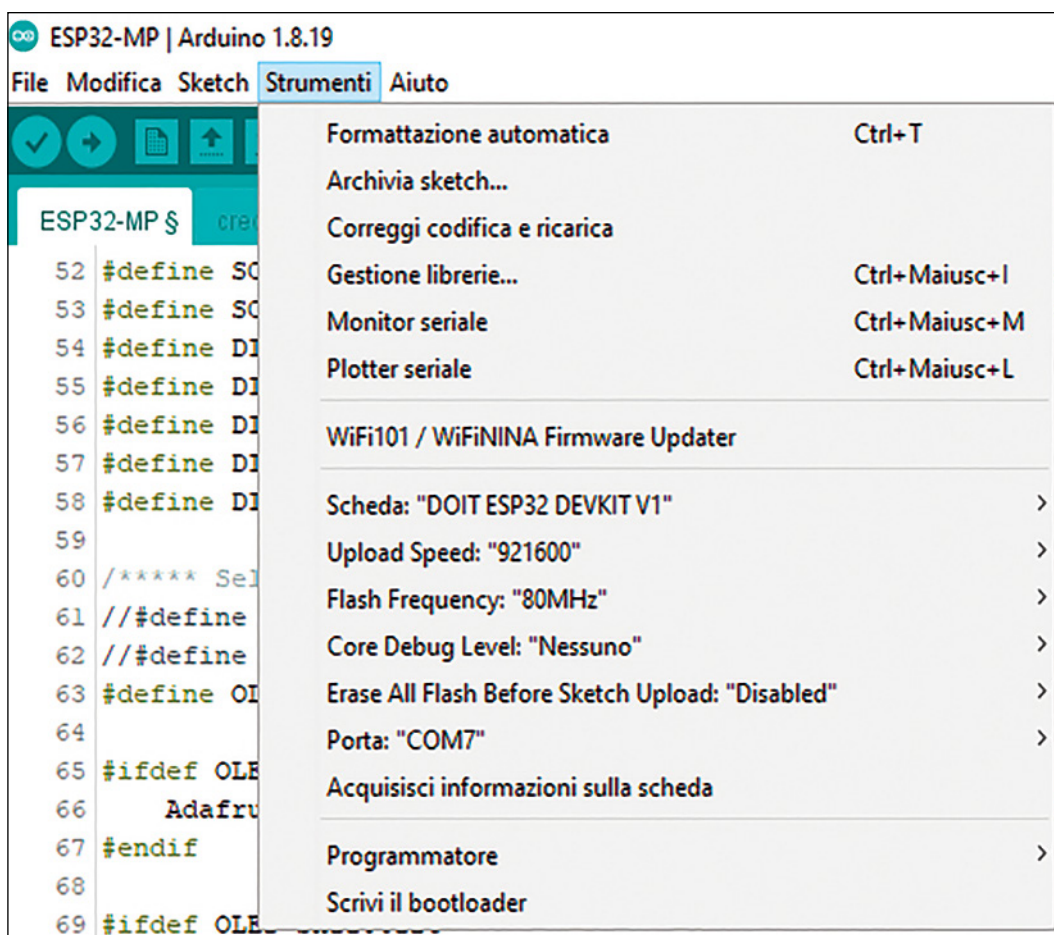


Fig. 20
WiFiManager
per la
gestione delle
credenziali di
accesso.

Fig. 21
Impostazioni di
programmazione.



Altra libreria che dovremmo installare si chiama WiFiManager.h (<https://github.com/zhzhouhan0126/WIFIMANAGER-ESP32>) che ci permetterà di implementare facilmente una metodologia per impostare l'accesso alla propria rete. Questa libreria consente di gestire in automatico l'accesso alla rete WiFi domestica qualora nella memoria dell'ESP vi siano salvate delle credenziali valide. In mancanza di queste credenziali, ad esempio alla prima accensione o ogniqualvolta lo riteniamo opportuno, il modulo ESP viene impostato in modalità access point e sarà lui stesso a creare una rete, di nome ESP32-MP, alla quale accederemo con un dispositivo mobile. L'accesso a questa rete avviene senza alcun login e ci servirà solo per impostare le credenziali di accesso alla nostra rete le quali saranno salvate permanentemente nella memoria del modulo ESP ed usate ad ogni accesso, come visibile in **Fig. 20**.

Non appena connessi a questa rete, si verrà reindirizzati verso un piccolo webserver fatto girare all'interno del modulo ESP, dal quale potremmo

selezionare la rete alla quale vorremmo connetterci ed inserire anche la relativa password.

Per concludere installeremo la libreria SoapESP32.h reperibile all'indirizzo <https://github.com/yellobyte/SoapESP32>.

Utilizzeremo questa libreria per scansionare la rete locale in cerca di server con funzionalità UPnP, sfogliandone il contenuto ed accedendo ad ogni singolo file musicale. Diciamo subito che questa funzionalità è ancora in modalità basilare e non abbiamo previsto particolari funzioni avanzate, inoltre abbiamo testato il software nel caso di una semplice rete domestica in cui è presente un solo dispositivo NAS con funzionalità di server DLNA. Per le nostre prove ci siamo limitati ad utilizzare le funzionalità di server di rete di un router FritzBox che, come molti router di nuova generazione, possono essere configurati per gestire un media server di rete condividendo i file salvati in una pen drive, connessa alla porta USB. In questo caso il router diventa un dispositivo DLNA in una rete domestica e permette la condivisione di file audio

e video in modo automatico. L'accesso a questi file può avvenire da un qualsiasi client presente nella stessa rete come ad esempio una smart TV o uno smartphone. Nel nostro caso la libreria SoapESP32 provvederà a trasformare il modulo ESP in un client DLNA che utilizza il protocollo di comunicazione UPnP, per eccedere ai vari file in rete. Sfrutteremo la libreria per indicizzare tutti i file presenti nella nostra rete, funzionalità più avanzate avrebbero permesso la selezione di brani audio per genere, per tipo di file o altro.

UTILIZZO PRATICO

Dopo aver installato le librerie ed aver aperto il file sorgente che potete scaricare assieme ai file di questo numero della rivista, non vi resta che avviare la programmazione del modulo ESP, fate riferimento alla **Fig. 21** per impostare correttamente la scheda.

Ad ogni accensione (da spento) viene avviata una ben precisa procedura che prevede:

- inizializzazione display
- inizializzazione WiFi e accesso alla rete
- inizializzazione pulsanti
- inizializzazione encoder
- inizializzazione telecomando IR
- inizializzazione libreria audio
- sincronizzazione ora e data
- scansione scheda SD

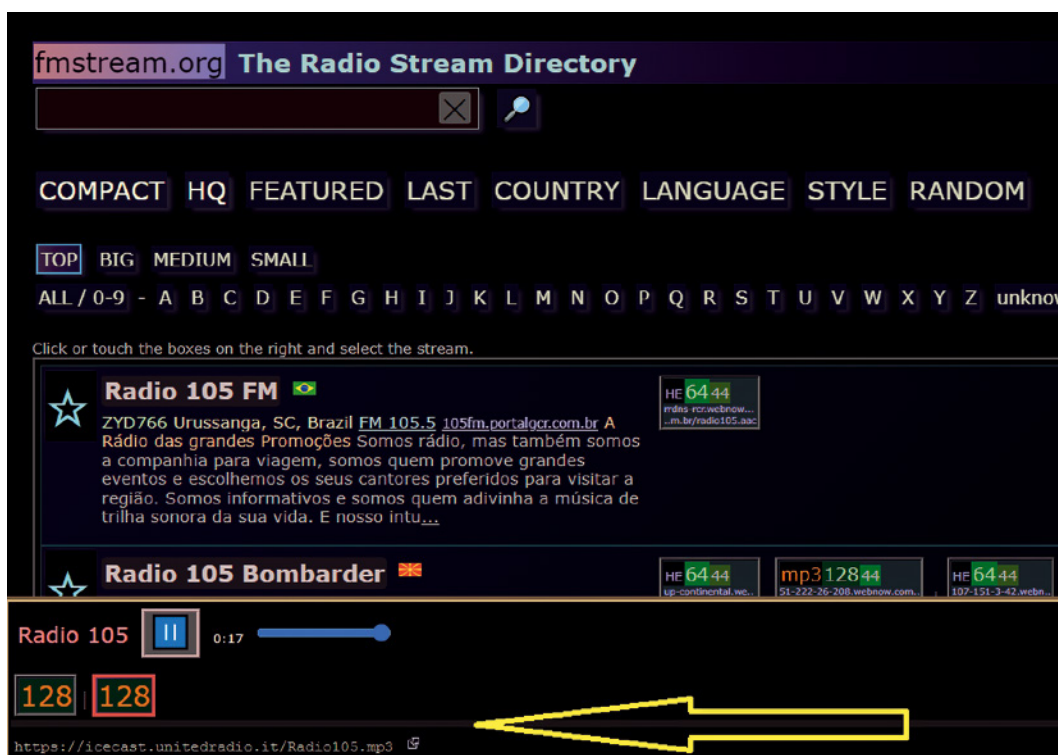
Se la scheda SD non è presente potremmo ancora usare il lettore audio ma non potremmo ascoltare brani salvati in locale, ovviamente. In questo caso potremmo ascoltare solo le stazioni radio di default elencate nella parte iniziale del programma, come riportato il **Listato 1**.

Ciascuna radio è specificata tramite un nome e il corrispondente link, per evitare di dover editare il software ogni qualvolta vogliamo aggiornare la lista delle stazioni radio abbiamo previsto di usare lo spazio in memoria della SD Card. All'interno della SD Card è necessario creare una cartella di nome RadioPlaylist ed all'interno di questa dei file divisi per categorie contenenti le informazioni delle radio. Ad esempio potremmo creare un file di testo di nome Italiana.txt contenente tutte le stazioni nazionali, il file va popolato con tante righe di testo tante sono le stazioni ciascuna specificata con il nome ed il carattere ";", seguito dal link della radio. Potremmo creare a piacere quanti file vogliamo ognuno popolato con i link di interesse. A livello software abbiamo previsto una procedura che legge in automatico l'elenco di questi file, permette di selezionare quello di interesse e di aggiornare la lista della radio station con i vari link in esso contenuti. Tra i file di questo numero della rivista trovate un esempio di contenuto da inserire all'interno della SD Card compresi alcuni brani musicali.

▼ Listato 1 - Elenco default radio station

```
/****** VARIABLE FOR DEFAULT RADIO STATION *****/
struct radioStation listStation[]=
{
    "0n-70s",                "http://0n-80s.radionetz.de:8000/0n-70s.mp3",
    "Radio105",              "https://icecast.unitedradio.it/Radio105.mp3",
    "KissKiss",              "http://ice07.fluidstream.net:8080/KissKiss.mp3",
    "Virgin",                "http://icecast.unitedradio.it/Virgin.mp3",
    "Rai1",                  "http://icestreaming.rai.it/1.mp3",
    "Rai2",                  "http://icestreaming.rai.it/2.mp3",
    "Rai3",                  "http://icestreaming.rai.it/3.mp3",
    "Radio Bella e Monella", "https://ice02.fluidstream.net/bella.mp3",
    "Radio Monte Carlo",     "http://edge.radiomontecarlo.net/RMC.mp3",
    "80s Super Dancefloor",  "https://s41.myradiostream.com:33604/;",
    "Radio Freccia",         "https://streamingv2.shoutcast.com/radiofreccia",
    "RTL 102.5",             "https://streamingv2.shoutcast.com/rtl-1025",
    "Radio Company",         "http://str01.fluidstream.net:7010/stream.mp3",
    "Radio Piterpan 1",      "http://wma02.fluidstream.net:2230/stream.mp3",
    "Radio Dancefloor 90",   "http://streaming.radiodancefloor.it:443/live-hq",
    "Radio Caffè",           "http://onair7.xdevel.com:8744/stream.nsv?1486762607955.mp3",
    "Radio country",         "https://stream.laut.fm/country",
    "Smooth Jazz",           "http://smoothjazz.cdnstream1.com/2585_320.mp3"
};
```

Fig. 22
Uso del portale
fmstream per
ricavare i link della
radio station.



Come facciamo a rintracciare il link della radio che ci interessa ascoltare? Tra le infinite possibilità consigliamo di accedere al sito <http://fmstream.org/> e cercare la radio di nostro interesse (**Fig. 22**), a fine pagina, in basso, troverete il link che dovrete copiare.

Potete copiare il link ed inserirlo nella barra degli indirizzi di qualsiasi browser per verificare se funziona correttamente.

Appena avviato lo sketch, il software chiede all'utente di selezionare il modo di ascolto tra internet radio, SD Card, lettore di rete, per ciascuna

Tabella 2
Menù di sistema.

VOCE	FUNZIONE
Scan audio in SD	Scansiona la SD Card e aggiorna la lista dei brani MP3 in riproduzione
Scan UPnP Server	Scansiona la rete locale e aggiorna la lista dei brani MP3 in riproduzione
Read radio playlist	Legge l'elenco delle internet radio disponibili nella cartella RadioPlaylist
Configure WiFi	Configura l'accesso alla rete WiFi
No repeat	Disabilita la ripetizione dei brani
Repeat single	Abilita la ripetizione del solo brano in corso
Repeat all	Abilita la ripetizione di tutti i brani in sequenza
Repeat random	Abilita la ripetizione di tutti i brani in sequenza casuale
Disable Amp	Disabilita l'amplificatore e lascia attiva solo l'uscita cuffie
Enable Amp	Abilita l'amplificatore

modalità il display mostra le dovute informazioni. Il display è diviso in tre sezioni, quella più in alto mostra il modo di funzionamento e l'ora, la zona centrale le informazioni inerenti alla stazione radio o il brano in riproduzione ed in basso la barra delle informazioni relative al volume selezionato, il bit rate del flusso di dati e delle icone a rappresentare lo stato di funzionamento.

Nel caso di internet radio il display si presenta come in **Fig. 23** e nella zona centrale sarà visualizzato il nome della stazione radio e nella seconda riga il brano attualmente riprodotto. Nel caso di riproduzione da SD Card o UPnP la schermata si presenta come in **Fig. 24**, in zona centrale è mostrato il nome dell'autore ed il titolo del brano oltre al tempo di riproduzione attuale. Solo in queste ultime due modalità è possibile selezionare tra riproduzione del singolo brano, riproduzione in sequenza di tutti i brani e riproduzione casuale. Per accedere al menù occorre premere l'apposito pulsante, successivamente con l'encoder possiamo sfogliare le varie voci e premendo il pulsante dell'encoder confermare la scelta.

In **Tabella 2** riassumiamo le possibili selezioni. Premendo il pulsante on/off è possibile abilitare la funzione di standby, nella quale il microprocessore della scheda ESP32 non viene spento ma usato per mantenere aggiornata l'ora di sistema. Per ottimizzare i consumi entrambi i moduli di amplificazione vengono posti in modalità a basso consumo azionando l'apposito pin, il DAC si pone automaticamente in modalità di risparmio energetico quando non riceve alcun flusso di dati. Dal menù è anche possibile disabilitare i moduli di amplificazione qualora si desideri lasciare attiva la sola uscita cuffie disponibile nel modulo DAC. Sempre nell'ottica di risparmio energetico tutte le periferiche, compreso il modulo WiFi della scheda ESP vengono spente per poi essere riabilitate solo alla riaccensione. Anche la scheda SD è posta in modalità a basso consumo e la luminosità del display viene diminuita, in questo modo l'intero sistema assorbe solo pochi milliampere. In questa fase il display visualizza data e ora (**Fig. 25**) e ad ogni minuto viene modificata la posizione delle scritte come avviene negli smartphone quando è attiva la funzione di always on display per evitare di danneggiare il display OLED.

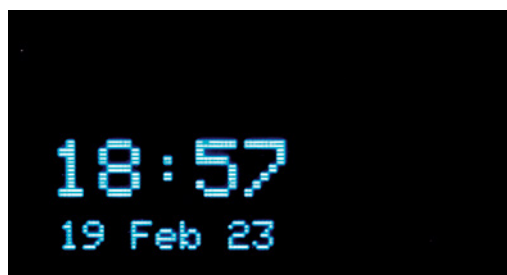
Alla riaccensione il lettore si predispose nelle esatte condizioni in cui si trovava prima dello spegnimento, non è necessario salvare in memoria alcun dato, visto che la CPU non viene mai riavviata. Al momento non abbiamo ancora implementato



← **Fig. 23**
Il display durante la riproduzione di una internet radio.



← **Fig. 24**
Il display durante la riproduzione di un file in rete.



← **Fig. 25**
Il display durante la modalità di stand by.

funzioni avanzate come lo spegnimento dopo un certo tempo o l'accensione automatica ad un certo orario, come avviene nelle radiosveglie. Nella trattazione abbiamo volutamente omesso il listato che essendo parecchio lungo avrebbe reso la lettura assai prolissa, esso comunque è disponibile e ben commentato per una facile comprensione, anche se presenta una certa complessità. Non ci resta che augurarvi un buon ascolto della vostra musica preferita. □



Cosa occorre?

I componenti utilizzati in questo progetto e riportati in Tabella 1 sono tutti disponibili presso Futura Elettronica all'indirizzo futuraishop.it

Il materiale va richiesto a:

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



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

€ 27,90



COD. PLIB6000

3,7V-6000 mAh.
Connettore 2 mm

€ 6,90



COD. PLIB1000-2MM

3,7V-1000 mAh.
Connettore 2 mm

€ 7,90



COD. PLIB1100-2MM

3,7V-1100 mAh
connettore 2 mm

€ 21,00



COD.
25C2000MAH7V4

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

€ 34,00



COD.
25C2200MAH11V1

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

€ 4,90



COD. PLIB500NEW

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

€ 6,90



COD. PLIB1000-1.25

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

€ 21,90



COD. PLIB5000MAH

3,7V-5000 mAh.
Connettore 2 mm

€ 21,90



COD. PLIB5000-2.54

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

€ 12,50



COD. PLIB430

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

€ 12,90



COD. PLIB2000

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

€ 7,90



COD. PLIB1100

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

Prezzi IVA inclusa.

RICEVITORE

1 CH 434 MHz

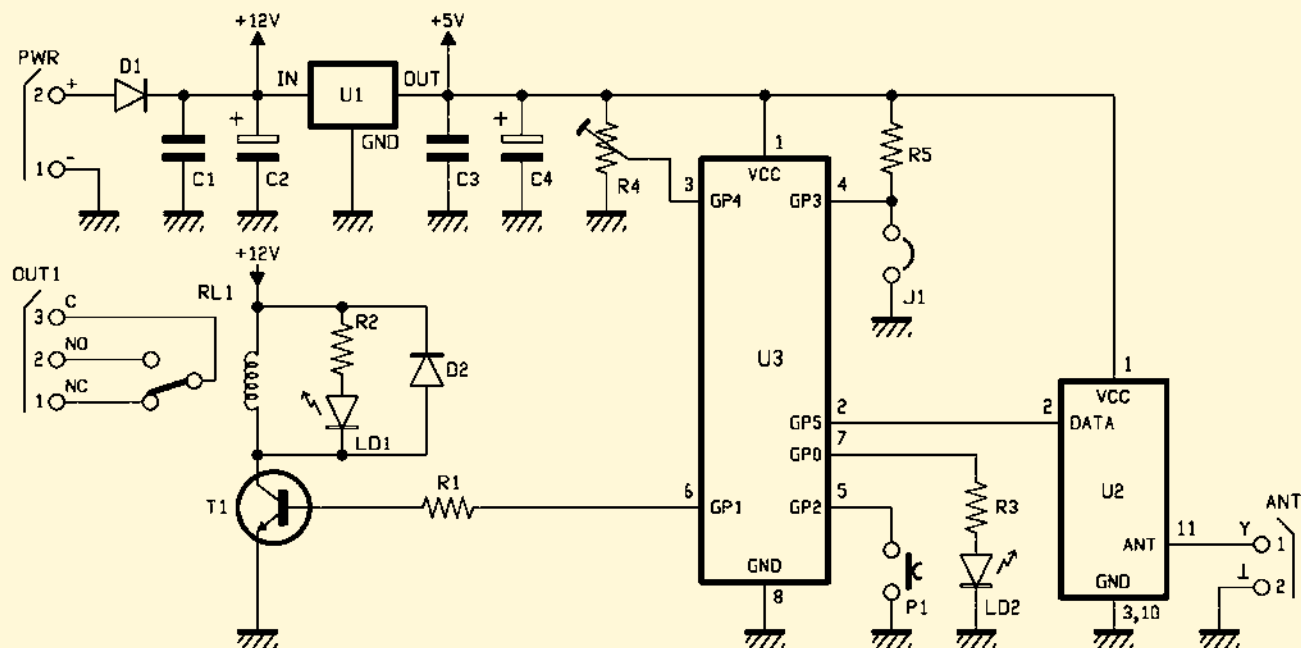
di ALESSANDRO SOTTOCORNOLA



Abbinato a un trasmettitore commerciale, consente di controllare a distanza utilizzatori di vario genere mediante un relé che può operare in modalità sia impulsiva che a permanenza.

Il telecomando, più specificatamente quello via radio e quindi il radiocomando, è ormai qualcosa cui ci siamo abituati per azionare, senza essere fisicamente sul posto né muovere interruttori, deviatori e simili, ogni genere di dispositivo o sistema: cancelli motorizzati, tapparelle e serrande motorizzate, cancelli elettrici, illuminazione, impianti di allarme e tanto altro.

Se è vero che oggi molte di queste cose sono integrate nei controlli domotici e gestite dalle app per smartphone, esiste ancora una gran varietà di situazioni in cui serve disporre del puro e semplice radiocomando esclusivo e indipendente dalle reti dati, dai cellulari e dal mondo della comunicazione globalizzata che si appoggia ad Internet. Ecco quindi spuntare la canonica accoppiata trasmittente e ricevente a radio-



frequenza, operante sulle bande assegnate ad apricancelli e affini, che nel nostro caso prende la forma del progetto qui proposto: un radiocomando a singolo canale, capace di commutare un circuito elettrico in modalità sia impulsiva (ossia fin quando viene tenuto premuto il pulsante del trasmettitore, ovvero finché viene ricevuto il relativo segnale codificato) sia bistabile (l'uscita inverte la propria condizione alla ricezione di ogni segnale codificato), idoneo a controllare a distanza qualsiasi cosa possa essere gestita elettricamente. Naturalmente, vista la gran quantità di dispositivi simili che operano simultaneamente in zone anche ristrette come i quartieri delle città, abbiamo pensato a qualcosa di codificato, in modo che il ricevitore qui proposto risponda esclusivamente a un segnale il cui codice rispecchi quello atteso. Diversamente dai radiocomandi più datati, per la codifica non c'è da impostare alcun dip-switch perché il sistema funziona ad autoapprendimento: tramite una semplice procedura è possibile abbinare un ricevitore ad uno specifico trasmettitore affinché risponda solo ad esso; anzi, la stessa procedura consente di memorizzare fino a 10 codici e altrettanti canali o dispositivi purché prodotti dall'azienda EMYLO, il che significa poter comandare uno stesso ricevitore da 10 differenti trasmettitori. Siccome EMYLO produce non solo mini TX formato portachiavi ma anche altri dispositivi, questo significa poter abbinare a un ricevitore più categorie di dispositivi trasmettenti.

SCHEMA ELETTRICO

Dopo questa premessa possiamo andare a vedere come è fatto e in che modo funziona il ricevitore, analizzandone lo schema elettrico: il tutto gira intorno all'integrato U3, che è un microcontrollore, in quanto non sarebbe possibile gestire decodifica, funzioni di autoapprendimento e memoria con un encoder integrato tradizionale; il micro è un PIC12F683-I/P nel quale viene caricato il firmware MF1702 (che potete scaricare dalle pagine di presentazione di questo numero della rivista all'indirizzo www.elettronica.in.it), il quale all'inizializzazione imposta la linea GP2 come ingresso digitale dotato di pull-up per leggere il pulsante P1, GP3 come ingresso digitale per acquisire lo stato del ponticello J1, GP4 come input analogico per leggere il trimmer (spiegheremo tra breve a cosa serve) e GP5 come input digitale per la lettura dei dati in arrivo dall'uscita del modulo RF ricevente U2. Sempre all'inizializzazione, i GPIO GP0 e GP1 come uscite per il controllo, rispettivamente, dell'uscita a relé (tramite il transistor NPN T1) e del LED di segnalazione LD2, polarizzato attraverso la resistenza R2.

Il segnale radio viene ricevuto da una piccola antenna tipicamente ad 1/4 d'onda (che sui 434 MHz corrisponde a circa 17 cm di lunghezza) e portato all'ingresso RF dell'integrato U2, un modulo ibrido CY121-433.92-ASK (**Fig. 1**) prodotto dalla CY wireless technology (www.rficy.com) contenente un completo ricevitore superete-

rodina compatibile con segnali a modulazione ASK/OOK/FSK con stadio di sintonia accordato a 433,92 MHz e conversione di frequenza, più amplificatore di media frequenza e demodulatore AM OOK con squadratore (comparatore) del segnale demodulato che raddrizza i fronti di commutazione, la cui uscita fa capo al piedino 2 (DATA). Lo stadio d'antenna ha un'impedenza d'ingresso di 50 ohm e un'eccezionale sensibilità di -108,5 dBm, tale da garantire un'ampia portata pur con trasmettitori da pochi milliwatt.

La piedinatura del modulo è la seguente:

- 1) alimentazione 5Vcc;
- 2) uscita dati 0/5V;
- 3) GND;
- 10) GND;
- 11) ANT (antenna).

Va notato che in realtà il modulo ha 11 pad a passo 2,54 mm dei quali però quelli interni sono utilizzati solo in particolari versioni ed hanno il seguente significato:

- 4) 3Vcc;
- 5) GND;
- 6) CSB configurazione/programmazione;
- 7) SDA configurazione/programmazione;
- 8) SCL configurazione/programmazione;
- 9) FCSB configurazione/programmazione.

Andiamo adesso a vedere come il modulo funziona nel circuito: ogniqualvolta in antenna si presenta un segnale RF di ampiezza adeguata e di frequenza pari a 433,92 MHz, lo stadio supereterodina lo sintonizza e converte di frequenza, andando poi a demodularlo; se il segnale ricevuto è stato modulato in ampiezza, all'uscita di segnale (piedino 2) il modulo restituisce impulsi di forma e frequenza pari a quella del segnale modulante del trasmettitore.

A questo punto entra in gioco il microcontrollore U3, il cui firmware dopo il power-on reset e l'inizializzazione degli I/O, fa pulsare per 5 volte rapidamente la linea GP0 e poi per prima cosa legge lo stato del piedino GP3 e quindi quello del jumper: se lo trova basso predispone la modalità bistabile per l'uscita GP1, altrimenti imposta il modo monostabile; dopo GP3, il firmware acquisisce la lettura dell'A/D converter interno al microcontrollore per determinare l'impostazione dell'intervallo a livello alto dell'uscita GP1 nella modalità a impulso. Espletate queste due operazioni, il firmware si pone "in ascolto" del piedino GP5 in attesa di una commutazione, che identificherà come l'arrivo

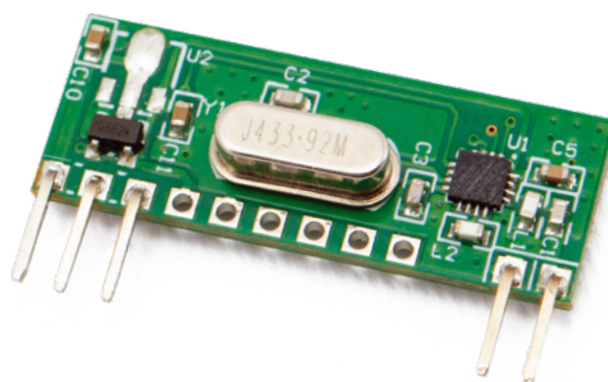


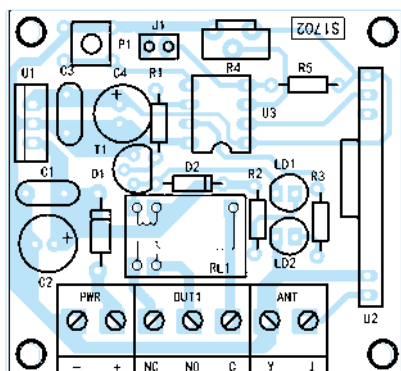
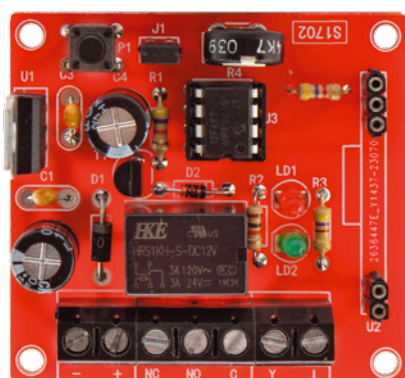
Fig. 1
Il modulo
radoricevente
CY121.

di una stringa di dati; il firmware carica l'intera stringa e va a leggerne il contenuto, dopodiché se si tratta di dati in formato differente da quello previsto (quello dei sistemi EMYLO, per intendersi...) svuota la RAM e si pone in attesa di nuove stringhe, altrimenti decodifica la stringa, che deve contenere 24 bit binari, confrontandola poi con i codici già memorizzati con la fase di autoapprendimento. In caso il codice ricevuto corrisponda a uno di quelli memorizzati in EEPROM procede con la subroutine di controllo dell'uscita, la quale verifica innanzitutto se il relé dovrà operare in modalità bistabile o impulsiva, quindi comanda di conseguenza il GPIO GP1.

Per l'esattezza, se è previsto il controllo bistabile il firmware invertirà l'attuale condizione logica di GP1 (quindi porrà il pin 6 a 1 logico se sarà a zero e viceversa), mentre se è impostata la modalità monostabile andrà a leggere il valore di tensione determinato dal trimmer R4 all'ingresso GP4 e lo convertirà in un valore numerico esprimente un

CARATTERISTICHE TECNICHE

Tensione di alimentazione: 12 Vcc	Modalità uscita: monostabile e bistabile
Corrente assorbita max: 40 mA	Abbinabile a: 10 trasmettenti o pulsanti
Frequenza operativa: 434 MHz	Codifica: 24 bit EMYLO
Numero di uscite: 1	



Elenco Componenti:

R1, R5: 4,7 kohm
 R2: 1 kohm
 R3: 470 ohm
 R4: Trimmer 4,7 kohm MV
 C1, C3: 100 nF ceramico
 C2, C4: 100 µF 35 VL elettrolitico
 U1: 7805
 U2: CY121-433.92-ASK
 U3: PIC12F683-I/P (MF1702)
 D1: 1N4007
 D2: 1N4148
 LD1: LED 5 mm rosso
 LD2: LED 5 mm verde
 T1: BC547
 P1: Microswitch
 RL1: Relè 12V

Varie:

- Morsetto 2 poli (2 pz.)
- Morsetto 3 poli
- Zoccolo 4+4
- Strip maschio 2 vie
- Jumper
- Circuito stampato S1702 (53x49 mm)

tempo, che sarà quello per il quale porterà a livello alto la linea GP1; infatti abbiamo previsto che nella modalità monostabile sia possibile regolare, con il trimmer R4, il tempo per cui il relé rimarrà eccitato, fra un minimo di 500 ms e un massimo di 9 secondi (il tempo minimo corrisponde alla posizione del cursore tutto verso massa). In modalità monostabile, quindi, il relé di uscita verrà attivato alla pressione del pulsante corrispondente sul trasmettitore (quello il cui codice sarà stato abbinato in fase di autoapprendimento) e tornerà a riposo trascorso, dal rilascio, un intervallo dipendente dalla posizione del cursore del trimmer R4.

L'uscita GP1 del microcontrollore pilota la base del transistor NPN T1 (configurato a emettitore comune) attraverso la resistenza di polarizzazione R1, ed il suo collettore alimenta la bobina del relé, il cui altro estremo è posto sul positivo della linea di alimentazione; il diodo D2 serve a proteggere la giunzione base-collettore del transistor quando quest'ultimo va in interdizione, allorché la bobina reagisce alla brusca interruzione della corrente generando un'extratensione inversa che se non venisse soppressa dal diodo causerebbe il breakdown elettrico del T1.

L'attivazione del relé viene segnalata dall'accensione del LED rosso LD1, polarizzato attraverso R2 quando il transistor va in saturazione. Del relé rendiamo disponibile l'intero scambio (tutti e tre i contatti C, NC e NO) in modo da adattare il radiocomando a ogni situazione di installazione.

Durante il funzionamento, il firmware testa ciclicamente anche lo stato del piedino 5, in modo da verificare se viene premuto il pulsante P1, che serve ad avviare la procedura di autoapprendimento del codice ricevuto via radio: quando si preme P1, il firmware avvia la routine di apprendimento che consiste nel leggere il codice in arrivo dal modulo RF U2 e, laddove questo sia nel formato compatibile con quello del sistema, decifrarlo e salvare i bit corrispondenti in un'apposita area della propria EEPROM interna.

Il ricevitore può essere abbinato a 10 codici, che vengono tutti stivati in memoria; tentando l'apprendimento di un ulteriore codice, l'operazione non viene portata a termine, nel senso che il LED LD2 lampeggia lentamente a indicare che pur essendo valido il codice ricevuto, la memoria è piena. Per apprendere nuovi codici occorre resettare la memoria e ricominciare da capo.

L'intero circuito si alimenta a tensione continua,

meglio se stabilizzata, di valore compreso fra 12 e 15V; l'alimentazione applicata tra i morsetti 1 e 2 (+ e - PWR) passa dal diodo D1 (che protegge il resto del circuito dall'inversione accidentale di polarità) e raggiunge l'ingresso del regolatore integrato a tre terminali U1 (un comune 7805) dove viene filtrata dai condensatori C1 e C2; all'uscita del regolatore troviamo 5V ben stabilizzati che, opportunamente filtrati dai condensatori C3 e C4, vanno a far funzionare il microcontrollore e ciò che lo contorna, modulo radiricevente compreso. Fa eccezione il circuito del relé, che funziona con la tensione a valle del diodo D1 (tipicamente 12V).

REALIZZAZIONE PRATICA

Bene, lasciamo ora da parte la teoria per occuparci degli aspetti pratici di questo progetto: iniziamo con la preparazione del circuito, per la quale abbiamo previsto un PCB appositamente disegnato e del quale vi rendiamo disponibili per il download la traccia lato rame che potrete utilizzare per procedere con la fotoincisione.

Lo stampato è a singola faccia, quindi di facile realizzazione anche da parte di chi ha poca esperienza nei montaggi elettronici ed inoltre la componentistica richiesta è tutta a montaggio tradizionale (THT=Through Hole Technology) il che semplifica ulteriormente il lavoro e lo rende davvero alla portata di tutti, anche dei principianti. Una volta incisa e forata la basetta, seguendo il piano di montaggio pubblicato in queste pagine disponete i componenti occorrenti a iniziare dalle resistenze e dai diodi al silicio (cioè tutti i diodi tranne i LED...) e dallo zoccolo per il microcontrollore, che orienterete con la tacca di riferimento rivolta a D2, quindi passate al pulsante miniatura e ai condensatori non polarizzati, proseguendo con il transistor (il cui lato piatto andrà rivolto al D2) e i LED, per i quali ricordiamo che il lato smussato è quello dove si trova il terminale di catodo (negativo).

Ora montate un pin strip a 2 poli per realizzare il ponticello J1, il trimmer verticale, il relé, i condensatori elettrolitici (per i quali va rispettata la polarità indicata nel piano di montaggio) il regolatore integrato (da tenere con il lato metallico rivolto all'esterno dello stampato) quindi le morsettiere a passo 5 mm (due bipolari e una tripolare) e il modulo ibrido: quest'ultimo va tenuto con il lato dei componenti rivolto all'interno dello stampato. Notate che in alternativa alla saldatura diretta al PCB, è possibile montare il modulo ibrido su uno zoccolo composto da una fila di strip femmina

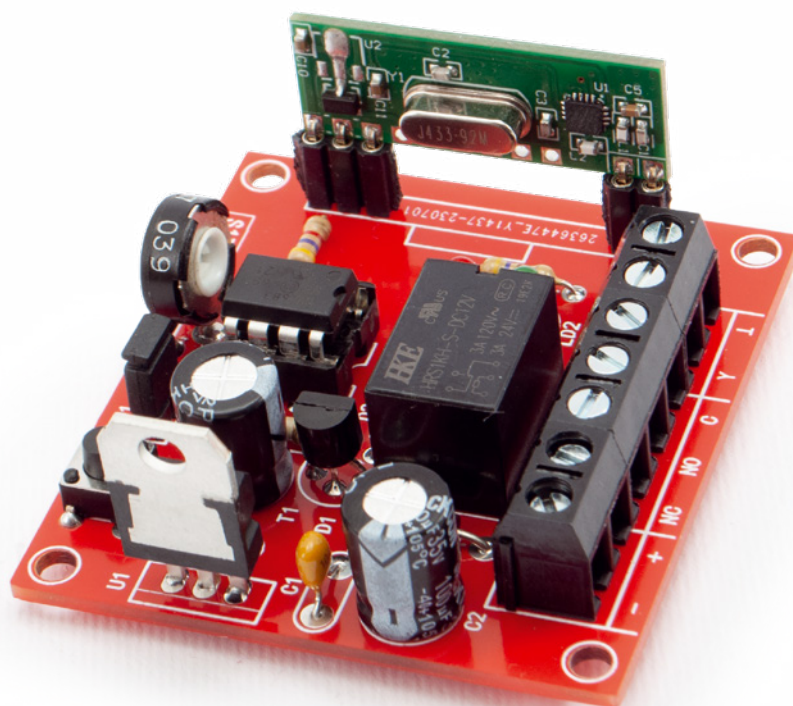
a passo 2,54 mm, ovvero su porzioni di essi da inserire e saldare nelle piazzole del PCB.

Completate le saldature potete inserire il microcontrollore già programmato con l'apposito firmware nel rispettivo zoccolo, badando di orientare il suo riferimento dalla stessa parte di quello dello zoccolo e accertandovi che non si pieghi alcuno dei piedini sotto il corpo durante l'operazione; il circuito al completo si presenterà come in **Fig. 2**.

Dovrete solo collegargli l'alimentazione e un'antenna ricevente, che potrà essere composta semplicemente da uno spezzone di filo in rame rigido lungo 17 cm avvitato nel morsetto di destra della morsettieria ANT; la massa serve solo se intendete montare un'antenna esterna collegata con del cavo coassiale schermato, che in tal caso dovrà avere impedenza caratteristica di 50 ohm.

Dopo aver verificato con il piano di montaggio che tutti i componenti siano orientati correttamente, potete procedere all'apprendimento dei codici. Per prima cosa dovete alimentare il circuito con un alimentatore in grado di fornire una tensione continua, preferibilmente stabilizzata, del valore di 12V ed una corrente di almeno 100 milliampere, così da avere una riserva per fronteggiare i picchi di assorbimento.

 **Fig. 2**
Il circuito montato,
privo di antenna.



Per semplificare lo schema del circuito abbiamo optato per una sezione radiricevente in forma ibrida e contenuta nel modulo ricevitore supereterodina CY121 della CY wireless technology.

Questo dispositivo integra su un substrato di allumina un completo radiorecettore supereterodina compatibile con le modulazioni ASK/OOK/FSK disponibile con tuner accordati su varie bande di frequenza, tra cui la ISM.

All'interno del modulo troviamo un amplificatore d'antenna e un sintonizzatore ad elevatissima sensibilità (si parla di -108 dBm) ed elevata selettività, nel nostro caso accordato a 433,92 MHz, con oscillatore locale quarzato, un amplificatore di media frequenza, un rivelatore AM/OOK e uno squadratore per raddrizzare i fronti del segnale AM demodulato. Le principali caratteristiche di questo modulo radio ibrido sono:

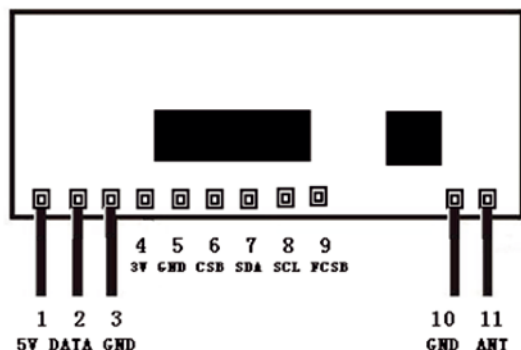
- frequenza di lavoro

300 ÷ 960 MHz;

- selettività di 160 kHz;
- modulazioni ASK/FSK;
- banda passante BF di 25 kHz;
- sensibilità in antenna -108,5 dBm;
- tensione di alimentazione da 3,6 a 5,5 Vcc;
- assorbimento tipico di 4 mA;
- temperatura di esercizio -20 °C ~ 70 °C;
- alta reiezione delle spurie dell'oscillatore locale;
- dimensioni di 1,5x12x5 mm.

La piedinatura del modulo CY121, visto dal lato dei componenti, è quella proposta nella Fig. A. Il modulo ha un foot-print single-in-line a 11 pin dei quali ne sono utilizzati tipicamente i 5 proposti nella figura.

Fig. A
Pin-out
dell'ibrido
CY121.



Fatto ciò verificate che fornendo alimentazione LED2 emetta 5 lampeggi, ad indicare il corretto avvio del firmware; a questo punto siete pronti per operare.

Procuratevi un trasmettitore della EMYLO ed eseguite la seguente procedura di apprendimento dei codici:

1. premete e tenete premuto il pulsante P1;
2. premete il pulsante del radiocomando che desiderate apprendere e attendete che il LED

LD2 (verde) lampeggi velocemente, segno che l'apprendimento è andato a buon fine.

Notate che se il lampeggio del LED risulta più lento, il codice è valido ma la memoria è piena; questo naturalmente non si verificherà al primo avvio del circuito ma dopo che sono stati appresi vari codici. In questo caso, vale a dire se la memoria è piena, non è possibile rimuovere uno o più codici selettivamente ma occorre cancellare l'intera memoria; idem se dovete rimuovere il codice di un trasmettitore non più in vostro possesso, per ragioni di sicurezza.

In tutti i casi in cui occorra riprogrammare la memoria dei codici bisogna eseguire la seguente procedura di cancellazione:

1. togliere alimentazione al dispositivo;
2. premere il pulsante P1 e tenerlo premuto;
3. fornire alimentazione e verificare che il LED LD2 rimanga acceso fisso per 2 secondi a indicare l'avvenuta cancellazione della memoria.

Lo stesso LED verde emetterà 5 lampeggi ad indicare l'uscita dalla procedura di cancellazione e l'avvio normale del ricevitore, che potrà quindi essere abbinato a nuovi (o gli stessi) trasmettitori con la medesima procedura descritta qualche paragrafo indietro.

Per il normale utilizzo, prima di alimentare il circuito dovete decidere in che modalità far lavorare l'uscita, quindi impostare il ponticello J1 di conseguenza: chiudete quindi i relativi pin-strip con un jumper a passo 2,54 mm se volete che il relé di uscita segua la modalità bistabile, altrimenti lasciate il ponticello aperto (ottenendo così la modalità impulsiva).

Alimentate quindi il ricevitore e sarà operativo. ■



Cosa occorre?

Il ricevitore monocanale (cod. FT1702K) è disponibile in scatola di montaggio a Euro 15,00, il telecomando a un canale (cod. EMYTX1) è disponibile a Euro 11,00. Il modulo radio (cod. CY121-433ASK) è disponibile anche separatamente a Euro . 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**

OSCILLOSCOPIO DA LABORATORIO 2 CANALI-100 MHz

COD. DSO2C10
€ 298,00



CARATTERISTICHE TECNICHE OSCILLOSCOPIO

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

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



COD. DSO2D10
€ 358,00

CARATTERISTICHE TECNICHE GENERATORE FORME D'ONDA

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

Le caratteristiche tecniche della parte relativa all'oscilloscopio sono le stesse del modello DSO2C10.

Batteria scarica? Niente paura!

Powerbank per bici e moto da 5000 mAh

- Supporto per il fissaggio al manubrio girevole a 360° per una facile installazione.
- Funzione Quick Charge QC 3.0 e Power Delivery per una ricarica fino a 4 volte più veloce.
- Torcia a LED integrata per l'uso in emergenza.
- Indicatore di stato a LED per una facile visualizzazione della carica rimanente.
- Compatto e facile da trasportare, ideale per viaggi e campeggio.
- Tensioni di ingresso 5Vdc/3A – 9Vdc/2A 12Vdc/1.5A.
- Tensione di uscita:
 - USB-A QC 3.0 (Quick Charge): 5Vdc/3A/15W – 9Vdc/2A/18W
 - USB type C (Power Delivery): 5Vdc/3A/15W – 9Vdc/2A/18W
- Protezione contro corto circuito e sovraccarico.
- Lunghezza cavo USB 0,3 m.
- Temperatura di funzionamento da 0°C a +45°C.
- Dimensioni: lunghezza 108 mm, diametro 30 mm.
- Peso 117 grammi.

Ricarica completa in meno di 1 ora e mezza

due input di ingresso per una maggiore flessibilità di ricarica



Skateboard elettrici



Bici elettriche



Laptop



cod. WT60656

€ 29,00



Monopattini

Disponibili anche:

Power Bank Slim 10.000 mAh



cod. WT40891

€ 26,50

Power Bank 20.000 mAh – Quick Charge



cod. WT53939

€ 46,00



Prezzi IVA inclusa.

TOMBINGO

IL GIOCO ELETTRONICO PER TUTTI

del Prof. **MICHELE MENNITI**



Una rivisitazione elettronica della Tombola o del Bingo che unisce tradizione e tecnologia per un divertimento coinvolgente.

P

robabilmente non tutti conoscono le curiose origini del gioco della Tombola, nato a Napoli nel XVIII secolo, durante il regno dei Borboni. Il Re Carlo di Borbone, che voleva stabilire il controllo pubblico sul gioco del Lotto, al fine di bloccare il proliferarsi di bische clandestine, si scontrò con il frate domenicano Gregorio Maria

Rocco, molto benvenuto dal popolo, il quale però auspicava l'abolizione ed il divieto della pratica del gioco del Lotto, che egli riteneva immorale e causa di traviamiento dei fedeli dal credo religioso. Alla fine trovarono un compromesso che stabiliva il divieto di gioco nel solo periodo natalizio. Da questa situazione nacque il gioco della Tombola, che, in sostituzione al gioco

del Lotto, veniva organizzata nelle case durante le festività di Natale.

Il Bingo si può considerare la versione americana della Tombola, ma con le caratteristiche del gioco d'azzardo, non a caso in Italia è permesso solo in apposite sale autorizzate dallo Stato. Conoscendo le tipiche problematiche di una partita domestica a Tombola, in cui la confusione spesso genera dubbi sui numeri chiamati o usciti in precedenza, oppure gli "incidenti" che capitano per la momentanea perdita di un bussolotto con un numero, con tutto ciò che ne consegue, abbiamo pensato ad una soluzione elettronica che ci garantisse prima di tutto la necessaria casualità e poi la certezza del numero estratto e dei numeri chiamati in precedenza, per le verifiche in caso di vincita o di distrazione del concorrente.

L'eccellente risultato che mostriamo in **Fig. 1** è stato ottenuto utilizzando un microcontrollore come cuore del circuito, due piccoli display e due tabelloni esterni con display di grandi dimensioni, per l'indicazione del nuovo numero estratto e 90 LED per l'indicazione fissa dei numeri già estratti.

SCHEMA ELETTRICO LOGICA

La logica di controllo del nostro TomBingo si basa su un microcontrollore ATmega328P-PU, il cuore del celeberrimo Arduino Uno, configurato con un quarzo esterno da 16 MHz.

Il micro è stato opportunamente programmato per generare, alla pressione del pulsante START (P1), un numero casuale da 1 a 90, che sarà istantaneamente mostrato sui due piccoli display DSP1 e DSP2 e, contemporaneamente, sui due tabelloni esterni,

ognuno costituito da due display giganti, mediante una codifica inviata tramite protocollo I²C (connettore CN2). Inoltre il microcontrollore invierà, sempre tramite I²C (connettore CN1), le istruzioni per pilotare la matrice di 90 LED, in modo da accendere quello corrispondente al numero appena estratto e, contemporaneamente, tutti quelli chiamati in precedenza.

Il pilotaggio dei LED è possibile grazie alla memorizzazione dei numeri estratti in un array, che viene letto ad ogni ciclo di aggiornamento. A fine partita il pulsante RESET (P2) permetterà di azzerare l'elenco dei numeri chiamati, mostrare "0" sui display ed iniziare una nuova partita. I due piccoli display DSP1 e DSP2 hanno il catodo comune collegato a massa tramite un resistore che limita la corrente di scorrimento nei LED che costituiscono i sette segmenti, mentre ogni singolo anodo è pilotato direttamente da un pin digitale del micro.

SCHEMA ELETTRICO TABELLONE LED (PARTE SINISTRA E PARTE DESTRA)

Il tabellone è costituito da due circuiti, uno per le prime 5 colonne (sezione sinistra) e l'altro per le restanti 5 colonne (sezione destra), i rispettivi PCB sono uniti tra loro mediante una coppia di connettori maschio-femmina da 13 pin, a formare quindi un unico tabellone di 10 colonne e 9 righe. Spieghiamo il funzionamento del tabellone, considerandolo come un unico circuito; per facilitarne la lettura abbiamo disegnato lo schema completo omettendo i due connettori, lo riportiamo in **Fig. 2**.

Le istruzioni I²C arrivano contemporaneamente ai due integrati U1 e U2, due MCP23017 che svolgono



Fig. 1
Il TomBingo
completo.

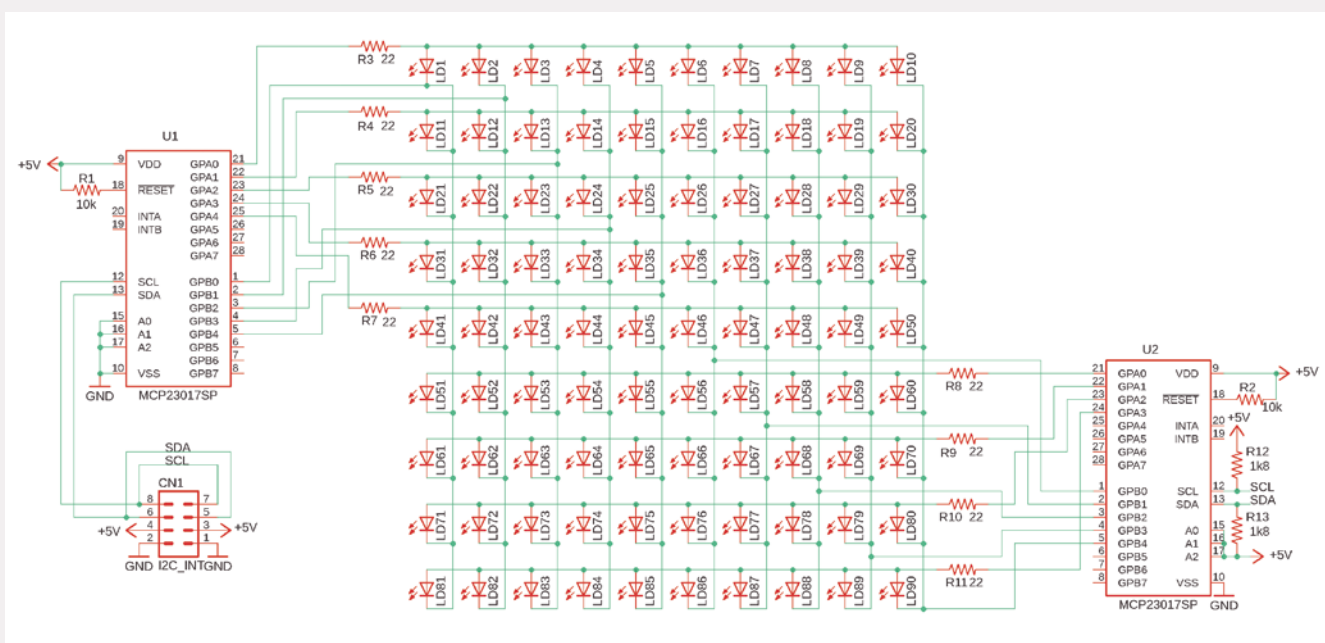


Fig. 2
Schema elettrico
unificato del tabel-
lone LED completo.

la funzione di expander I/O a 16 bit; in pratica ogni integrato dispone di 16 uscite che possono essere pilotate per via seriale (I²C) mediante opportune combinazioni di bit, quindi con valori 0 o 1, in cui il valore 1 serve per pilotare l'anodo del LED ed il valore 0 per pilotare il catodo.

Poiché non si poteva certo pensare di gestire 180 uscite separate (quindi un totale di ben 15 integrati), abbiamo fatto ricorso alla tecnica del multiplexing, riuscendo così ad utilizzarne solo 2. Un paio di esempi chiariranno meglio la tecnica. Prendiamo la prima riga di LED LD1÷LD10, come si vede chiaramente dallo schema, tutti i loro anodi sono uniti assieme e sono pilotati dal pin GPA0 di U1, mentre i 10 catodi sono pilotati da altrettanti pin distribuiti tra i due integrati U1 e U2 e precisamente GPB0÷GPB4 di U1 e GPB0÷GPB4 di U2. L'accensione di un LED richiede lo scorrimento di una corrente dal suo anodo verso il suo catodo, quindi il pin che pilota l'anodo deve trovarsi a livello High, mentre quello che pilota il catodo deve trovarsi a livello Low, qualsiasi altra combinazione manterrà il LED spento. Per esempio, per accendere il LED LD3 occorre che GPA0/U1 sia a livello High e GPB2/U1 sia a livello Low, mentre per accendere il LED LD9 occorre che GPA0/U1 sia a livello High e GPB3/U2 sia a livello Low; se devono restare accesi solo questi due LED tutti gli altri GPB di U1 e U2 devono trovarsi a livello High.

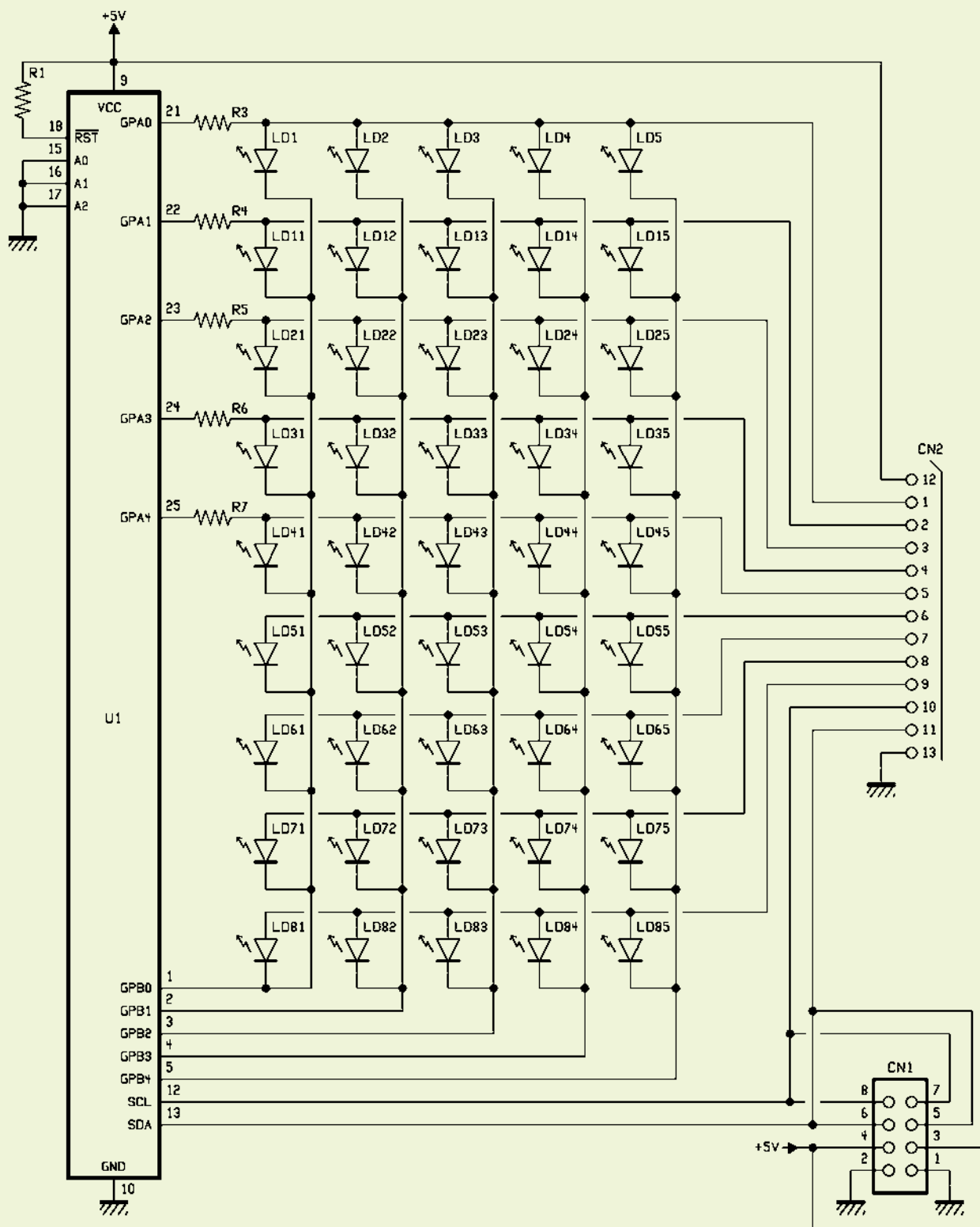
Se si trattasse di una sola fila di LED la cosa sarebbe molto semplice, come abbiamo appena visto, ma la

struttura a matrice, se non pilotata opportunamente, farebbe accendere sempre tutti i LED di una stessa colonna.

Ecco che ci viene in aiuto la tecnica del multiplexing. Essa consiste nel pilotare, a grande velocità, ognuno dei LED da accendere, singolarmente, l'uno dopo l'altro, mantenendo spenti gli altri 89. In pratica se eseguiamo questa operazione al ritmo di 1 secondo (1 Hz) vedremo accendere ogni singolo LED in base alla sequenza di estrazione, ma non sarebbe possibile vedere contemporaneamente accesi tutti i LED corrispondenti ai numeri estratti. Invece questa operazione viene eseguita alla massima velocità prevista per l'I²C, che è di ben 400 kHz, e il resto lo fa l'occhio umano, mediante il cosiddetto fenomeno della persistenza retinica, in virtù della quale l'occhio che visualizza un'immagine la mantiene memorizzata per un certo periodo di tempo, facendo sì che tutta la sequenza di accensioni appaia come un'accensione unica.

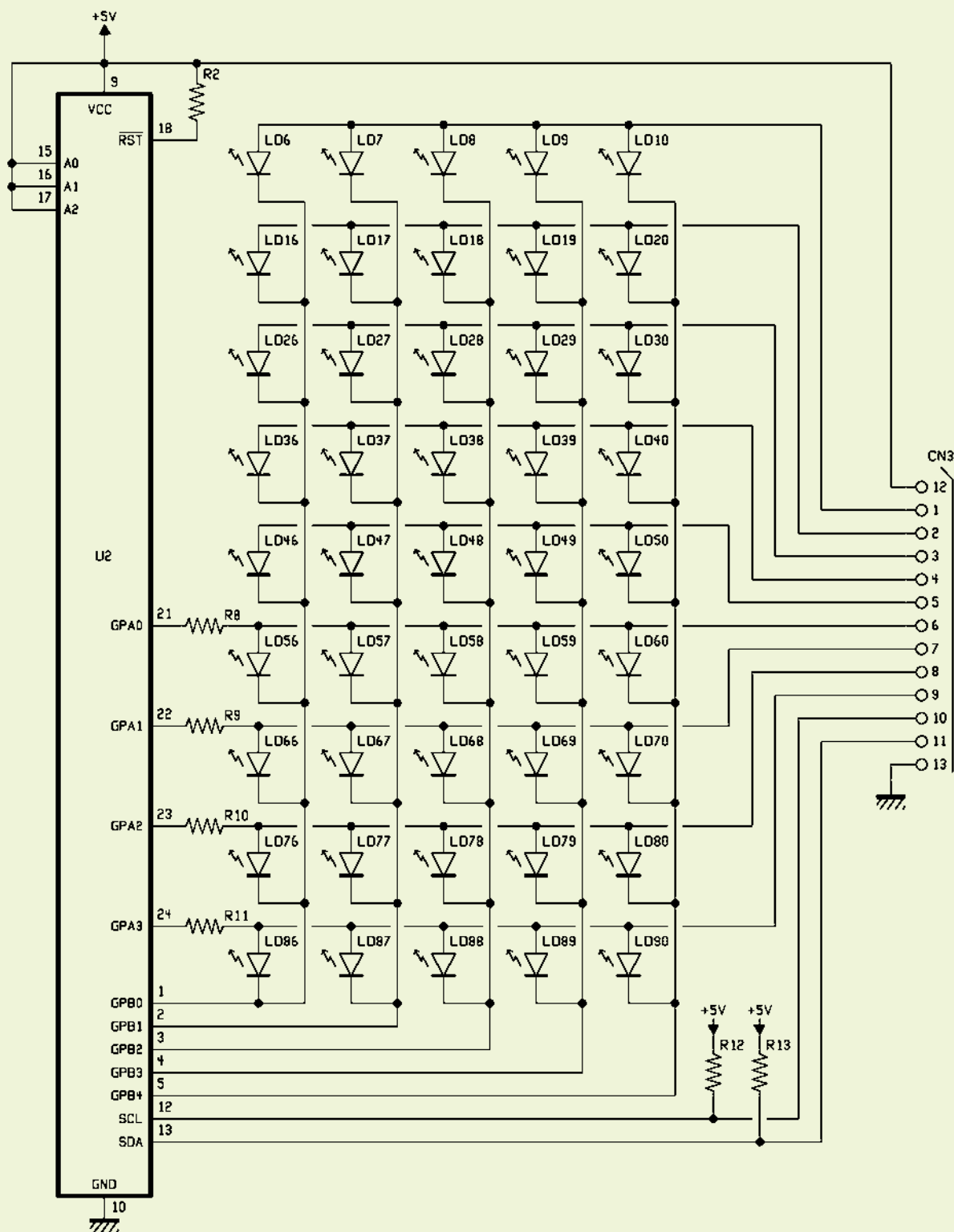
Questa cosa diventa evidente dopo l'accensione di circa 50-60 LED (dipende da chi li osserva), allorché essi iniziano a mostrare un leggero sfarfallio, segno appunto che ognuno di essi si sta accendendo e spegnendo molto rapidamente.

Lo sfarfallio aumenta con l'aumentare dei numeri estratti, tuttavia questo effetto non crea alcuna difficoltà nella lettura del tabellone, anche perché, statisticamente, una tombolata alla quale partecipano almeno 6 giocatori viene completata tipicamente dopo 60-70 estrazioni e maggiore è il numero di



giocatori e maggiore è la probabilità che la "Tombola!" venga chiamata con meno estrazioni. Una nota importante riguardo i LED: devono necessariamente essere del tipo ad alta luminosità, in

quanto il multiplexer tende a ridurla, e i LED normali potrebbero non emettere luce sufficiente per una buona visibilità. Ogni riga di LED presenta un resistore da 22 ohm,



che ha la funzione di limitare la corrente di alimentazione dei LED; in condizioni normali ogni singolo LED dovrebbe avere un proprio resistore, ma in questo caso, come abbiamo chiarito prima, ogni LED, in un

dato istante, viene acceso da solo, ed è solo per il fenomeno della persistenza visiva che li vediamo accesi tutti contemporaneamente, quindi questa soluzione va benissimo.

spettivamente ai pin fisici 17-16-15 dell'integrato), secondo la configurazione fornita dal datasheet, che riportiamo in **Fig. 4**.

Come si vede l'indirizzo dell'integrato si basa sul corrispondente valore esadecimale di una combinazione di 7 bit, dei quali i primi 3 meno significativi (partendo da destra) sono A0-A1-A2, quindi quelli modificabili da noi, mentre gli altri 4 sono fissi e impostati su "0100".

Se consideriamo inizialmente A2-A1-A0 uguali a 0 abbiamo un valore di base "0100000" che corrisponde a "0x20" esadecimale, questa cosa è facilmente verificabile cercando su Internet la voce "convertitore binario esadecimale".

La condizione esattamente opposta è data da A2-A1-A0 uguali a 1, con un valore binario "0100111" che corrisponde invece a "0x27" esadecimale. Considerando le varie combinazioni dei 3 bit, sappiamo che possiamo disporre di $2^3=8$ valori possibili, quindi da "0x20" a "0x27".

Vediamo quali abbiamo usato nei nostri circuiti.

L'integrato U1 della parte sinistra del tabellone ha i tre pin settati a 0 (tutti collegati a massa) quindi il suo indirizzo è "0x20"; l'integrato U2 invece ha i tre pin settati a 1 (tutti collegati a +5V) quindi il suo indirizzo è "0x27". Nel caso dei due blocchi di display giganti, avendo realizzato un unico PCB identico per entrambi, abbiamo previsto dei jumper a saldare che, in fase di montaggio, dovranno essere settati mediante una goccia di stagno, vedremo più avanti come; per ora è sufficiente sapere che uno dei due avrà A2-A1-A0 su 0-1-1 quindi "0x23" e l'altro avrà A2-A1-A0 su 1-0-1 quindi "0x25".

Un'altra nota riguardante il bus I²C: com'è ormai noto a chiunque lo abbia usato almeno una volta, i suoi due segnali SDA e SCL richiedono l'ausilio di una corrente, specialmente quando le informazioni viaggiano via cavo, come nel nostro caso.

Questa funzione viene svolta da due resistori di pull-up che, nel nostro caso, sono posizionati sul PCB della parte destra del tabellone dei LED, si tratta delle R12 e R13; non ha importanza in quale punto delle linee esse siano collocate, l'importante è che siano presenti.

Chiudiamo la parte relativa agli schemi elettrici del TomBingo con la motivazione della presenza del modulo step-up nello schema del tabellone con i display giganti. I segmenti dei display da noi adottati in realtà sono costituiti ognuno da tre LED in serie, che richiedono una tensione di circa 6 V per funzionare correttamente. Il nostro intero circuito è invece alimentato a 5 V, quindi abbiamo dovuto aggiungere questo piccolo modulo (8300-YXA347

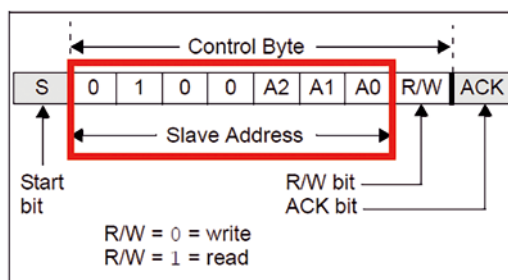
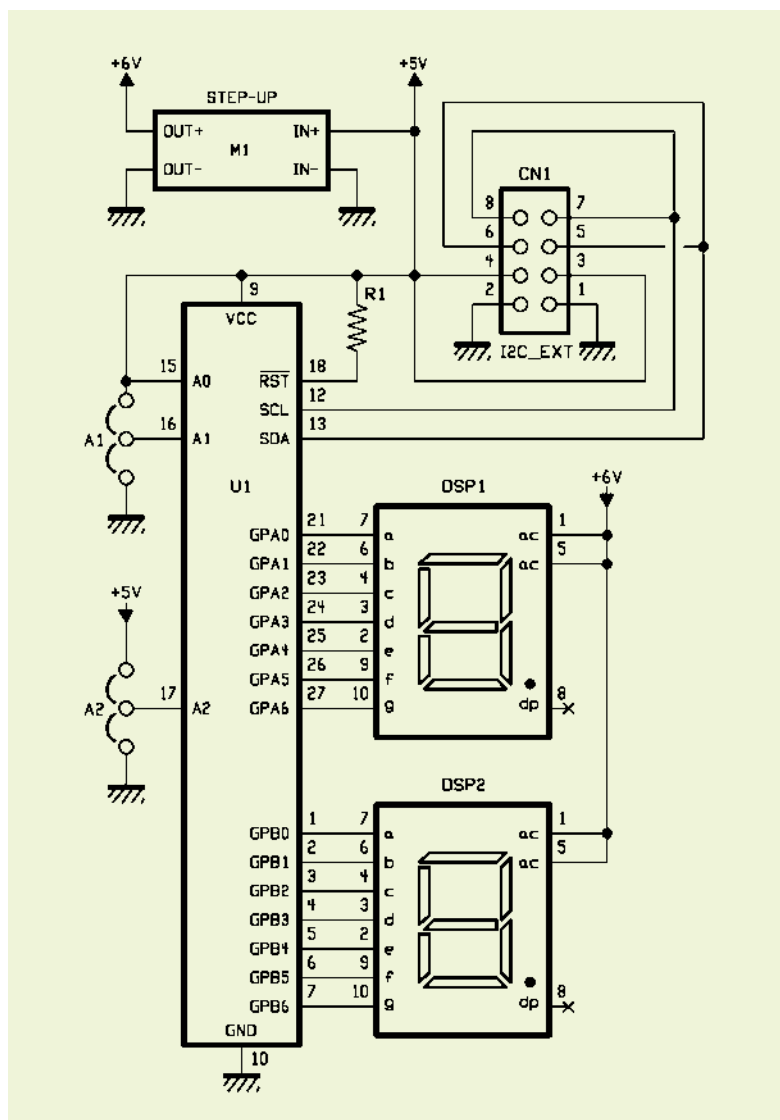


Fig.4
Indirizzi del-
l'MCP23017.

schema elettrico TABELLONI ESTERNI



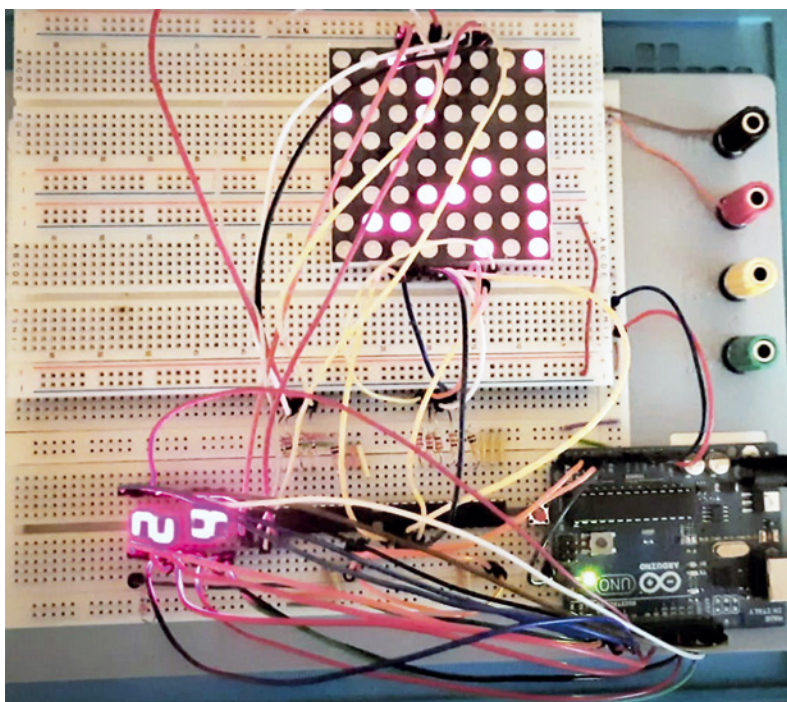


Fig. 5
Prototipo base
del TomBingo.

su futuranet.it) per elevare la tensione di 1 V ed alimentare la coppia di display. Poiché li alimentiamo esattamente con la tensione richiesta dal costruttore, non servono i classici resistori di limitazione in serie ai LED.

MONTAGGIO

A titolo di curiosità in **Fig. 5** mostriamo il prototipo base su breadboard col quale abbiamo testato i collegamenti ed il firmware, prima di passare al disegno degli schemi elettrici definitivi e dei relativi PCB professionali; a dimostrazione che dietro ogni progetto presentato sulla nostra rivista c'è sempre molto lavoro di progettazione e di test, per raggiungere il miglior risultato possibile!

Per la realizzazione dei PCB necessari per realizzare il TomBingo, abbiamo messo, nella sezione download della pagina di presentazione di questo numero della rivista tutti i necessari file gerber, ricordiamo che è attivo il servizio realizzazione PCB a prezzi molto contenuti.

Una volta in possesso dei PCB, le semplici regole per il montaggio dei componenti sono sempre le stesse, si inizia dai componenti più bassi: resistori, zoccoli degli integrati, quarzo, condensatori e via via quelli più alti.

Di seguito daremo alcune indicazioni specifiche per i vari PCB; le misure di distanziamento che indicheremo sono riferite al successivo montaggio nei contenitori che abbiamo previsto, che potranno essere

realizzati mediante una stampante 3D. Ovviamente se saranno adottate soluzioni diverse bisognerà regolarsi di conseguenza.

Tutti i componenti della parte Logica vanno montati sul piano superiore del relativo PCB, ad eccezione del connettore CN1, che va montato sul piano inferiore, come da serigrafia, in modo da farlo corrispondere con l'equivalente della parte sinistra del tabellone dei LED.

I due deviatori a pulsante vanno saldati direttamente sul PCB, al contrario dei due piccoli display, per i quali conviene prevedere due strisce di header femmina, in modo che la loro superficie arrivi a 16 mm dalla superficie del PCB, allo stesso livello della parte terminale della filettatura dei pulsanti; in tal modo si troveranno esattamente a filo del pannello, una volta montato tutto nel contenitore.

Sul PCB della parte sinistra del tabellone bisogna montare tutti i componenti sulla faccia inferiore del PCB, come da serigrafia, quindi resistori, zoccolo del circuito integrato e connettore CN1.

Il connettore CN2 può essere montato indifferentemente sulla faccia superiore o inferiore, a patto di fare la stessa cosa con quello del PCB della parte destra.

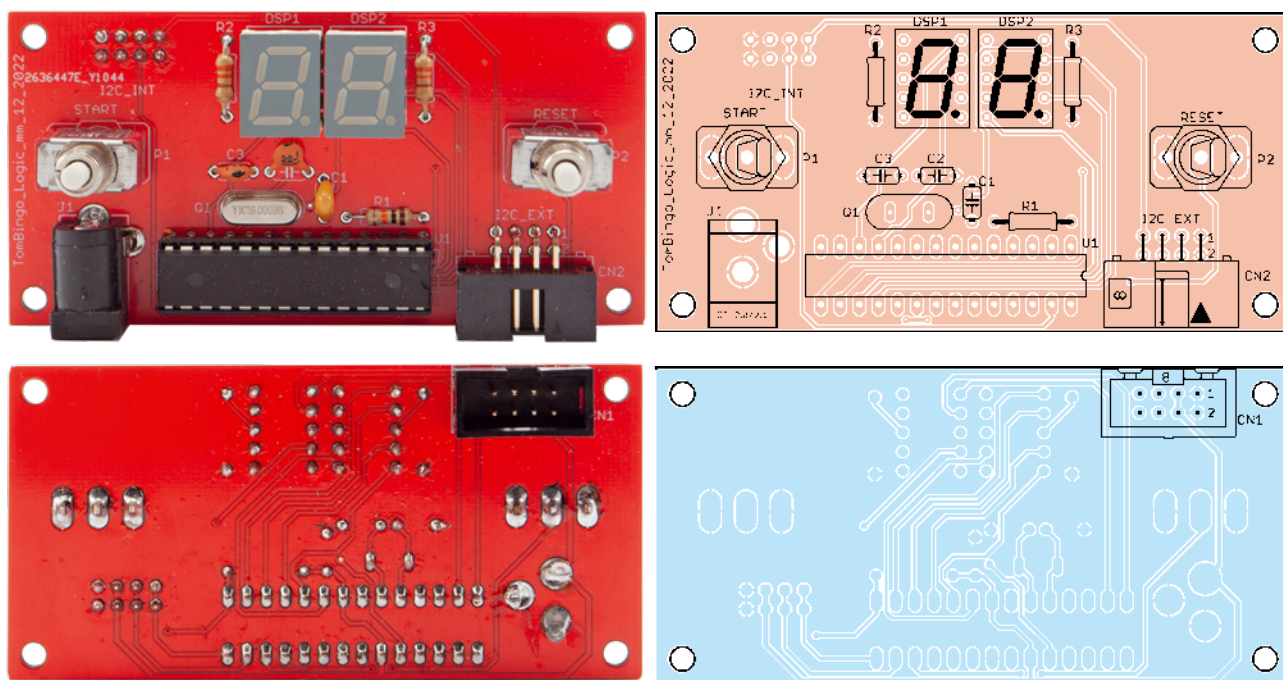
Invece tutti i 45 LED andranno montati sulla faccia superiore; poiché i LED devono avere tutti la stessa altezza, per fuoriuscire dal pannello in modo omogeneo, noi abbiamo usato questo sistema: abbiamo montato nei quattro fori di fissaggio del PCB altrettanti distanziali da 15-16 mm, rivolti verso l'alto, quindi sulla faccia superiore del PCB, quindi abbiamo inserito i LED nei rispettivi fori (ATTENZIONE alla polarità!) ed abbiamo capovolto il PCB, in tal modo i LED sono scesi fino a toccare la superficie del piano di lavoro tutti alla stessa altezza; quindi abbiamo saldato un solo reoforo per ogni LED, così da poter verificare il corretto posizionamento ed apportare i correttivi ove necessario; a quel punto abbiamo saldato tutti i reofori ancora liberi, ottenendo una perfetta omogeneità di altezza.

Per il PCB della parte destra occorre seguire le stesse indicazioni della parte sinistra, quindi resistori e zoccolo dell'integrato sulla faccia superiore del PCB, il connettore CN3 sulla stessa faccia del CN2 della parte sinistra, i 45 LED sulla faccia superiore, come visto in precedenza.

Il nostro progetto, come spiegato in precedenza, prevede due tabelloni esterni, ognuno costituito da due display giganti ma, volendo adottarne solo uno, la cosa è fattibilissima.

La prima operazione da fare è quella di impostare l'indirizzo dell'MCP23017 (vedi spiegazione teorica

piano di montaggio PARTE LOGICA



Elenco Componenti:

R1: 10 kohm
R2, R3: 330 ohm
C1: 100 nF ceramico
C2, C3: 22 pF ceramico
P1, P2: deviatore a pulsante
DSP1, DSP2: Display 7 segmenti

10x13 mm catodo comune
U1: ATMEGA328P-PU (MF1686)
Q1: Quarzo 16 MHz a basso profilo

Varie:
- Connettore maschio da c.s.

IDC 4+4 vie
- Connettore maschio da c.s. IDC
4+4 vie 90°
- Plug DC
- Zoccolo 14+14
- Circuito stampato S1686

nella sezione degli schemi elettrici). Come abbiamo visto, uno dei due tabelloni va settato su 0x23 e l'altro su 0x25. Queste due condizioni si ottengono rispettivamente con: A0=1, A1=1, A2=0 e A0=1, A1=0, A2=1. Poiché A0 è sempre su 1, abbiamo collegato il pin 15 direttamente al positivo dell'alimentazione, mentre per A1 e A2 abbiamo previsto due jumper a saldare con tre pad ciascuno. Il pad centrale di sinistra è collegato al pin 17 (A2), mentre quello di destra al pin 16 (A1). I pad laterali posti alla sinistra del centrale sono collegati a massa, mentre i due laterali posti a destra sono collegati al positivo di alimentazione. Quindi, per attribuire il valore 1 ad un pin occorre depositare una goccia di stagno tra il pad centrale e quello di destra, mentre per attribuirgli il valore 0 occorre depositare una goccia di stagno tra il pad centrale e quello di sinistra.

In **Fig. 6** abbiamo riportato il particolare dei due PCB settati correttamente: a sinistra quello con A2=0 e A1=1 (indirizzo 0x23) e a destra quello con A2=1 e A1=0 (indirizzo 0x25).

Bisogna fare attenzione a due cose: la prima, la più importante è evitare assolutamente di mettere in corto tutti e tre i pad del singolo jumper, perché significherebbe chiaramente cortocircuitare

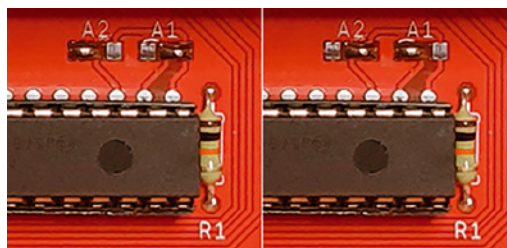


Fig. 6
Set degli indirizzi dei tabelloni esterni.

l'alimentazione! La seconda è rispettare queste due combinazioni, altrimenti si attribuirebbero indirizzi che creerebbero problemi.

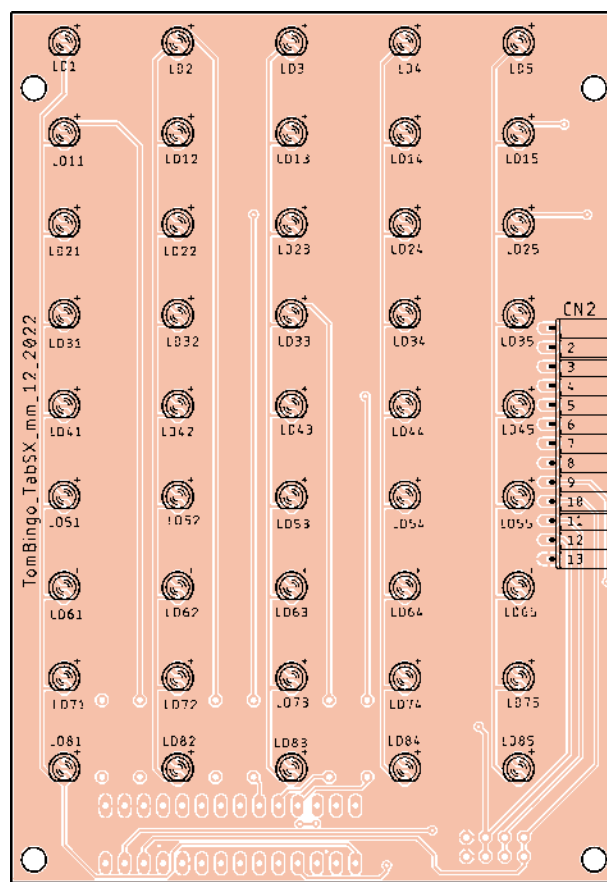
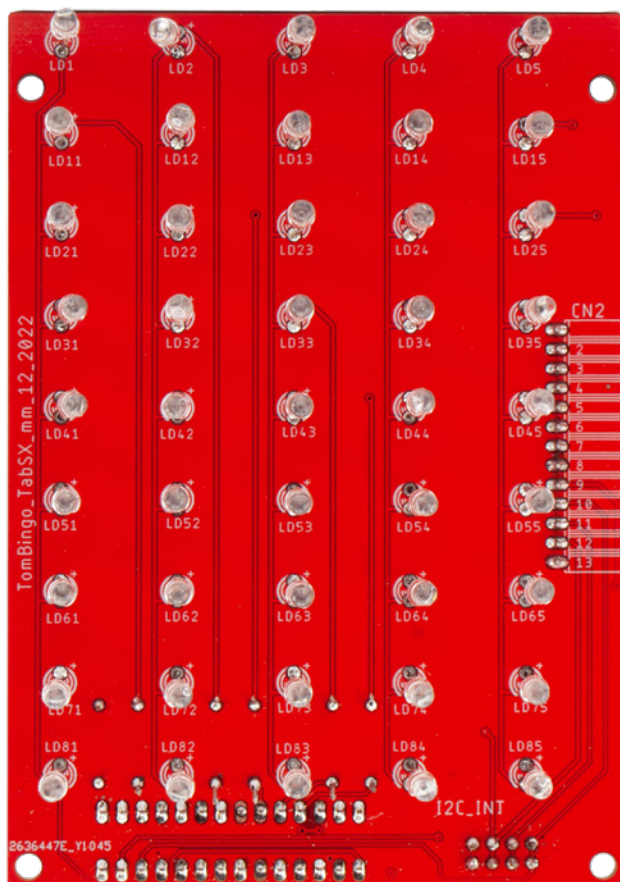
Infatti se ipoteticamente si impostasse A2=1 e A1=1 si replicherebbe l'indirizzo dell'integrato presente sulla parte destra del tabellone dei LED (0x27) creando un conflitto tra i due integrati; settando invece A2=0 e A1=0 si otterrebbe un indirizzo 0x21 non previsto nel firmware ed i due display non funzionerebbero correttamente.

Tutti i componenti vanno montati sulla faccia superiore del PCB, ad eccezione del modulo step-up, che va montato sulla faccia inferiore.

Prima del montaggio però il modulo dovrà essere correttamente tarato, applicando una tensione di 5V agli ingressi Vin+ e Vin- (ATTENZIONE alla polarità!) e agendo sul trimmer di precisione fino ad ottenere sui pin Vout+ e Vout- una tensione di 6 V.

Un buon sistema per montare il modulo sul PCB è quello di inserire nei suoi quattro forellini altrettanti pin strip e saldarli sul lato componenti, e poi innestare gli strip nei fori del PCB del display, saldandoli sul lato componenti di quest'ultimo, ma facendo molta ATTENZIONE a far combaciare le serigrafie Vin e Vout del modulo con quelle del PCB, altrimenti il modulo si danneggerà irrimediabilmente! (**Fig. 7**)

piano di montaggio TABELLONE PARTE SINISTRA



Elenco Componenti:

R1: 10 kohm

R3, R4, R5, R6, R7: 22 ohm

LD1, LD2, LD3, LD4, LD5, LD11, LD12,

LD13, LD14, LD15, LD21, LD22, LD23,
LD24, LD25, LD31, LD32, LD33, LD34,
LD35, LD41, LD42, LD43, LD44, LD45,

LD51, LD52, LD53, LD54, LD55,
LD61, LD62, LD63, LD64, LD65,
LD71, LD72, LD73, LD74, LD75, LD81,

Per quanto riguarda i display, bisognerà usare degli header a basso profilo come zoccoli, per far sì che la loro superficie venga distanziata dalla faccia superiore del PCB di circa 15 mm, così verranno a trovarsi a filo della mascherina dello specifico contenitore previsto (**Fig. 8**).

IL FIRMWARE

Alcune indicazioni sul firmware che abbiamo scritto per questo progetto.

Aprendo il file *Tombingo.ino* dall'IDE Arduino troverete 5 schede:

1. Tombingo: è la sezione principale, nella quale

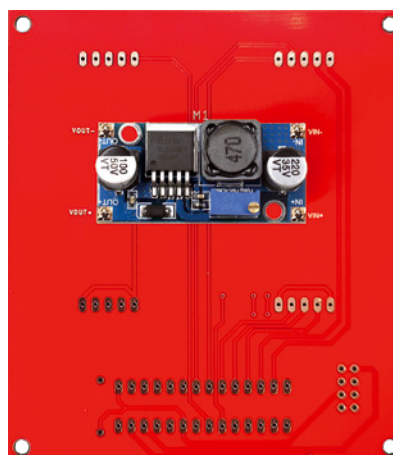
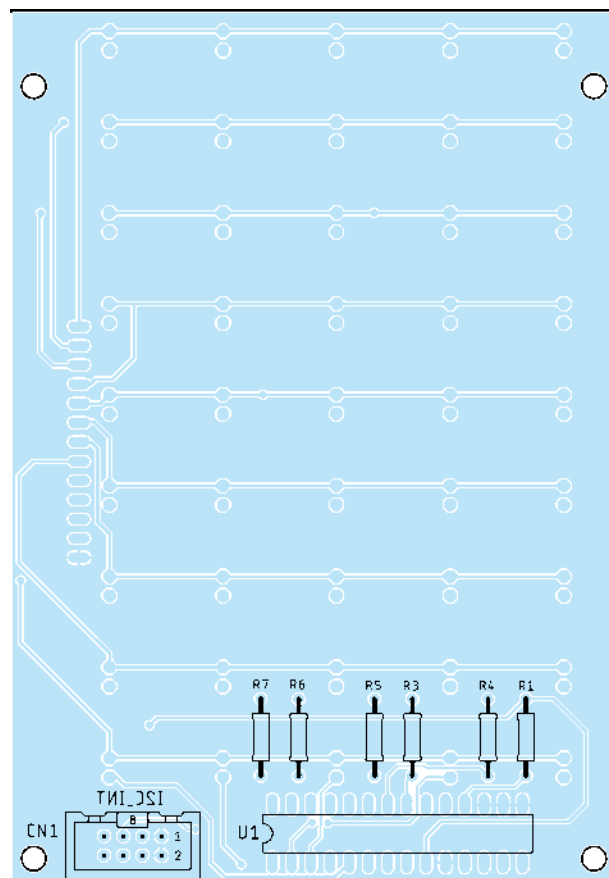
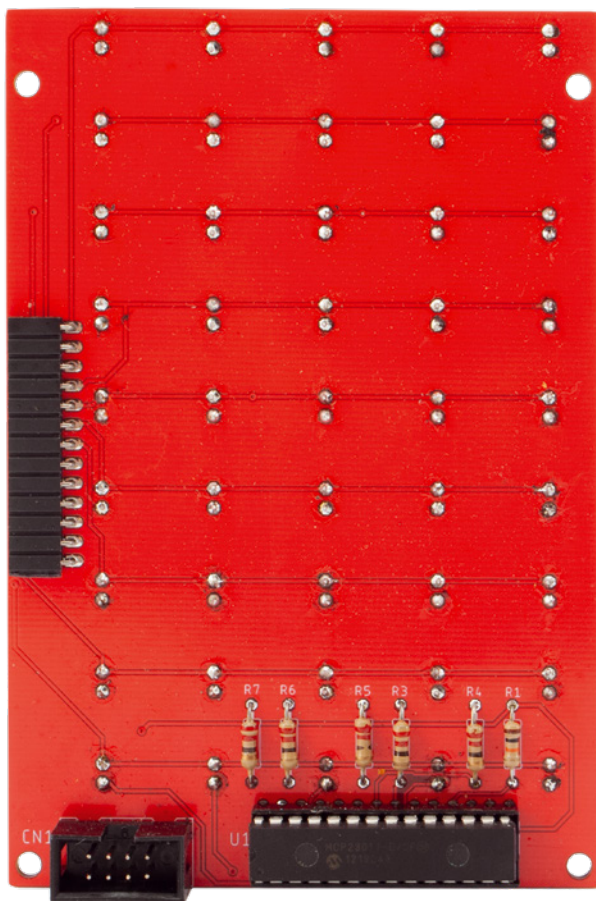


Fig.7
Il modulo
montato sulla
faccia inferiore
del PCB.

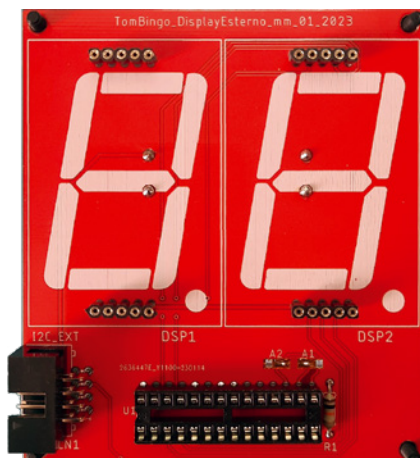


LD82, LD83, LD84, LD85:
LED 3 mm rosso ad alta luminosità
U1: MCP23017-E/SP

Varie:
- Strip femmina 13 vie 90°
- Connettore maschio da c.s. IDC 4+4 vie

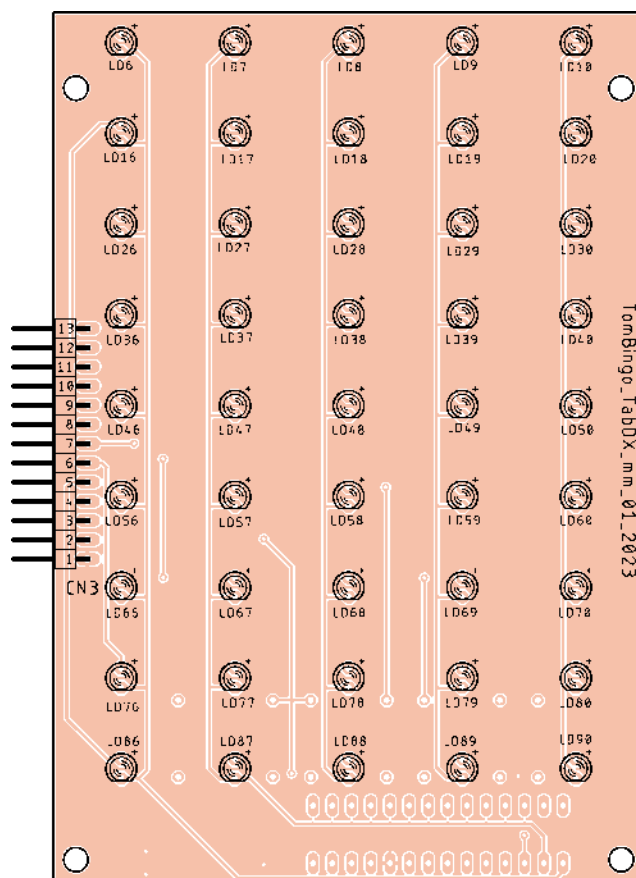
- Zoccolo 14+14
- Circuito stampato
S1687_SX

Fig. 8
PCB dei display gi-
ganti con zoccolo,
IDC ed header.



sono incluse le librerie <Wire.h> e <MCP23017.h>; seguono l'attribuzione degli indirizzi per i vari MCP23017, alcune variabili, e la dichiarazione dei pin usati per pilotare i due display interni della Logica. Nel Setup vengono inizializzati i collegamenti I²C, per i quali il comando "Wire.setClock(400000L)" setta la massima velocità possibile (400 kHz), lo stato iniziale delle porte degli MCP23017 ed i pin del microcontrollore; inoltre viene attivata la sequenza random per le estrazioni dei numeri, collegandola al pin analogico AO; infine vengono eseguite le istruzioni per mostrare all'accensione, sui display interni

piano di montaggio TABELLONE PARTE DESTRA



Elenco Componenti:

R2: 10 kohm
R12, R13: 1,8 kohm
R8, R9, R10, R11: 22 ohm

LD6, LD7, LD8, LD9, LD10, LD16, LD17,
LD18, LD19, LD20, LD26, LD27, LD28, LD29,
LD30, LD36, LD37, LD38, LD39, LD40,

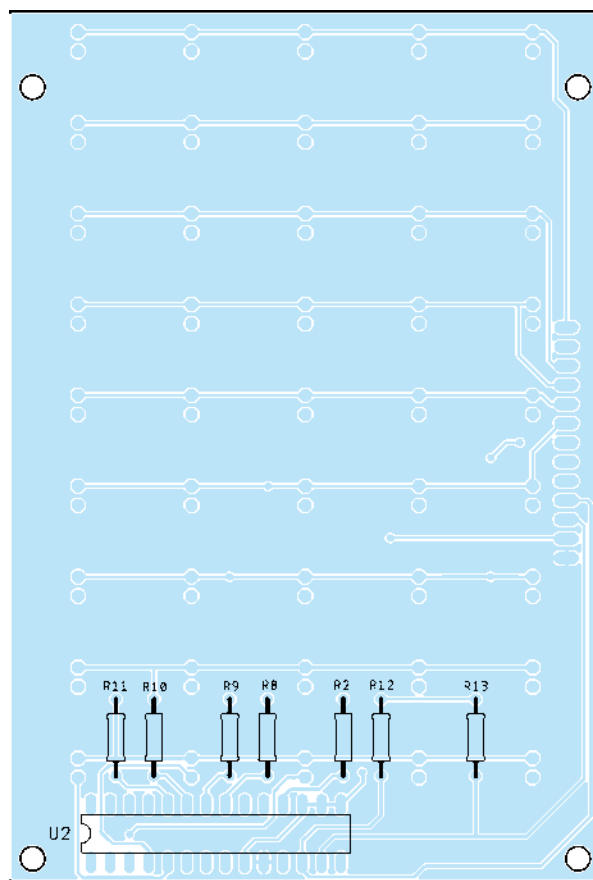
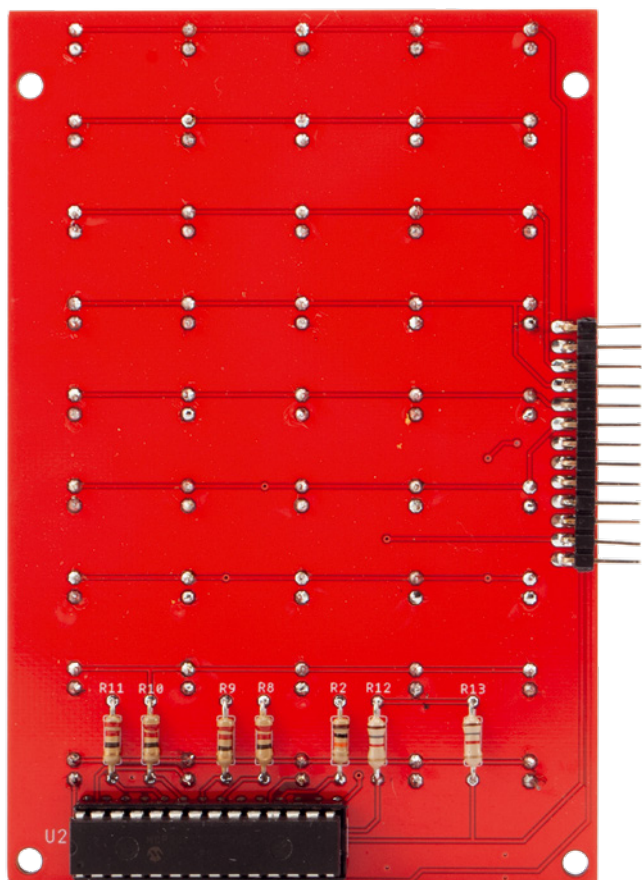
LD46, LD47, LD48, LD49, LD50, LD56,
LD57, LD58, LD59, LD60, LD66, LD67,
LD68, LD69, LD70, LD76, LD77, LD78,

e/o esterni, il numero "0" o "00" (leggi più avanti); queste opzioni si possono disabilitare commentando le relative righe, ma consigliamo di lasciarne attiva una per avere sempre conferma della corretta accensione dell'apparecchio. Nel Loop ci sono la routine di estrazione casuale del numero, l'impostazione di un blocco del pulsante START per 3 secondi dopo ogni estrazione, per evitare sovrapposizioni dei numeri, e una routine che mostra sui display tre linee orizzontali e accende tutti i LED, quando sono stati chiamati tutti i 90 numeri del tabellone.

2. Display: contiene tutte le combinazioni binarie

delle porte del microcontrollore che pilotano i due piccoli display interni, in modo da accendere i segmenti corretti per ogni numero chiamato. Nelle prime righe di questa scheda è presente la doppia combinazione per lo "0", è possibile cioè decidere se all'accensione e ad ogni nuovo inizio partita lo "0" debba essere mostrato su entrambi i display o solo sulla cifra meno significativa (quella a destra); di default è settato il doppio "0", per ottenere l'altra opzione bisogna commentare le due righe attive e decommentare le due precedenti.

3. Display_EXT: contiene tutte le combinazioni binarie delle porte dell'MCP23017 che gestisce



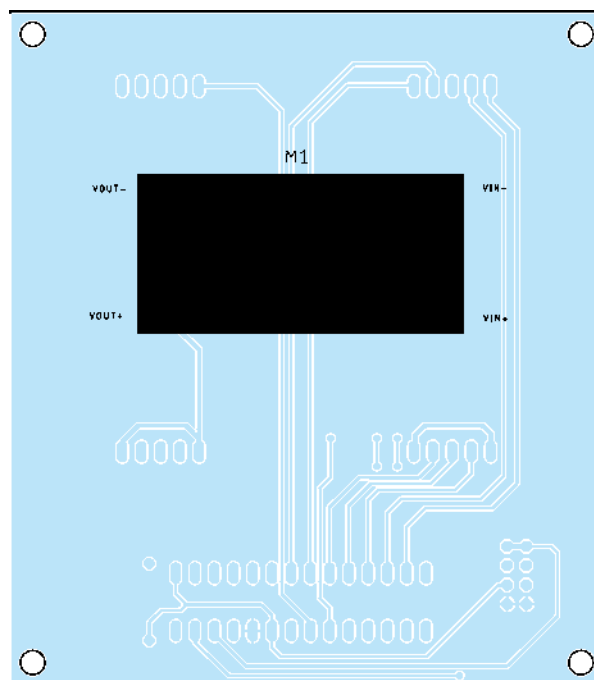
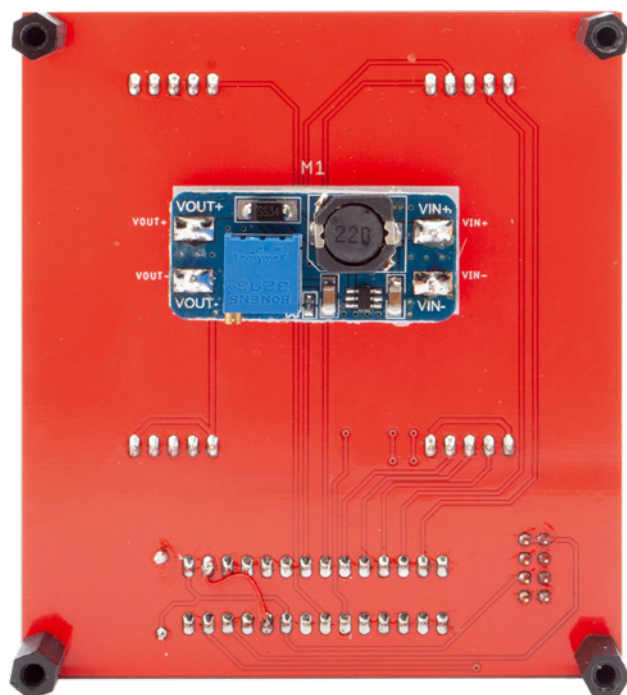
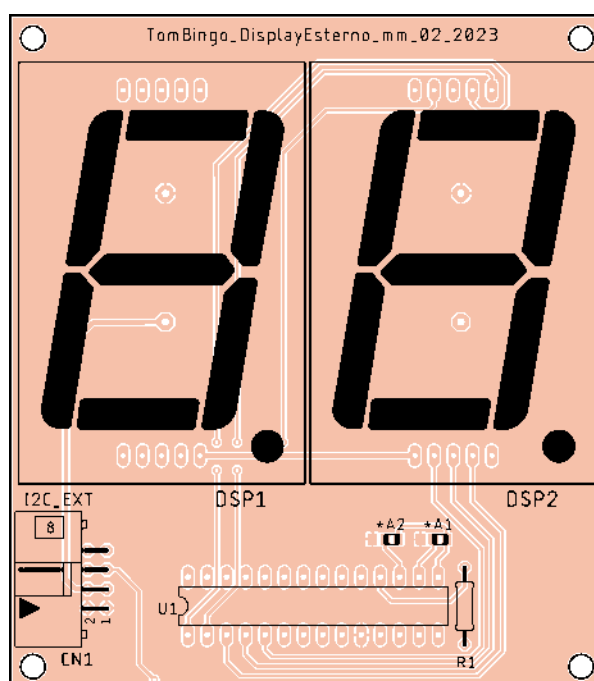
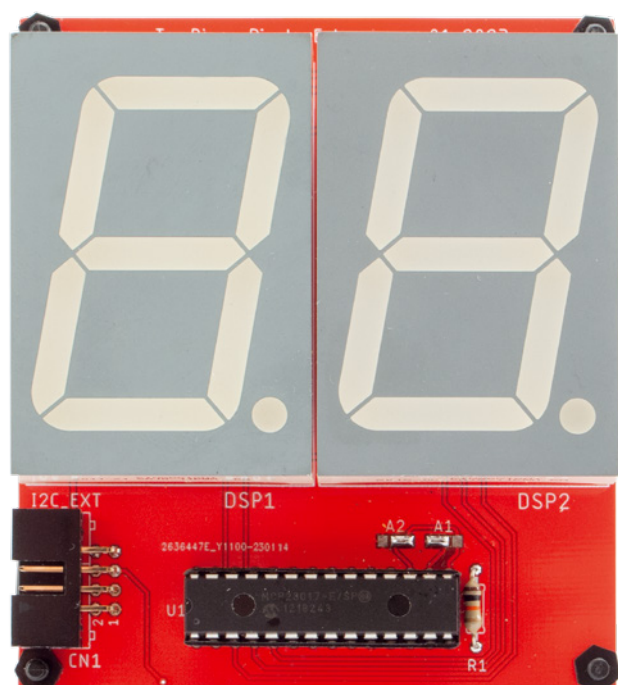
LD79, LD80, LD86, LD87, LD88, LD89,
LD90: LED 3 mm rosso ad alta luminosità
U2: MCP23017-E/SP

Varie:
- Strip maschio 13 vie 90°
- Zoccolo 14+14

- Circuito stampato
S1687_DX

piano di montaggio

TABELLONE ESTERNO



Elenco Componenti:

R1: 10 kohm
 DSP1, DSP2: Display 7 segmenti
 anodo comune (cod. Mouser
 604-SA18-11EWA)

U1: MCP23017-E/SP

Varie:
 - Modulo Step-Up (cod. YXA347)

- Connettore IDC 4+4 vie 90°
 maschio da c.s.
 - Zoccolo 14+14
 - Circuito stampato S1688

la coppia di display di ogni tabellone esterno, in modo da accendere i segmenti corretti per ogni numero chiamato. Per ogni numero esistono due righe per "mcp3" e due righe per "mcp4", che corrispondono rispettivamente agli indirizzi 0x23 e 0x25, quindi è già tutto predisposto per pilotare uno o due tabelloni esterni. Anche in questa scheda esiste la possibilità di scegliere tra le opzioni di accensione "0" o "00", come spiegato in precedenza per i display piccoli, solo che in questo caso le righe da commentare o decommentare sono quattro.

4. Estrazione: ci sono le operazioni necessarie per scartare i numeri già estratti e memorizzare il primo nuovo numero; infatti la chiamata random dei numeri non garantisce l'univocità, per cui ogni numero potrebbe essere estratto più volte durante il gioco, ecco perché ogni nuovo numero viene memorizzato in un array e dopo ogni estrazione il numero viene confrontato con quelli già presenti nell'array stesso: se c'è viene scartato e il programma chiama automaticamente un altro numero, se invece non c'è viene memorizzato a sua volta.

5. Matrice: contiene tutte le combinazioni binarie delle porte dei due MCP23017 che gestiscono i 90 LED del pannello, in modo da accendere e mantenere accesi quelli corrispondenti ai numeri estratti.

COLLEGAMENTI

I vari connettori IDC a 8 pin presenti nei PCB di Logica, parte sinistra del tabellone dei LED e tabelloni esterni servono per trasportare l'alimentazione principale ed i segnali I²C dalla Logica agli altri circuiti. Questi i collegamenti da effettuare:

1. CN1 Logica con CN1 parte sinistra del tabellone dei LED;

2. CN2 Logica con CN1 dei due tabelloni esterni con i display giganti.

Bisogna quindi realizzare due cavetti piattina, uno lungo 3,5 cm (o quanto serve nel caso si usi un contenitore diverso) con due spinotti IDC femmina alle due estremità, e l'altro lungo in tutto 45 cm con due spinotti IDC femmina alle due estremità più uno all'interno, distanziato di circa 19 cm da uno qualsiasi dei due capi esterni.

Gli spinotti devono essere orientati tutti allo stesso modo, altrimenti i segnali saranno invertiti, con conseguenze imprevedibili.

In **Fig. 9** si possono vedere i due cavetti realizzati per il nostro prototipo.

Osservando attentamente la foto si vede chiaramente che tutte le linguette di riferimento degli spinotti sono orientate verso destra. Per quanto riguarda i collegamenti dei due cavetti tra i tre circuiti si può far riferimento alla **Fig. 10**.

Una volta completato il montaggio di tutti i PCB, l'ultima operazione da fare è di inserire i circuiti integrati ed i vari display nei rispettivi zoccoli o header. Il solo microcontrollore dovrà essere preventivamente programmato, mediante tecnica ISP, utilizzando lo sketch presente nella solita sezione download, oppure potrà essere acquistato già programmato sul sito futuranet.it.

COLLAUDO

Il nostro TomBingo assorbe una corrente massima di circa 420 mA se si usano i display esterni, altrimenti essa scende a circa 50 mA. Nel primo caso è necessario un alimentatore da parete con uscita stabilizzata di 5 V 1+2 A, nel secondo caso potrebbe essere alimentato con un pacco di 4 batterie tipo AA ricaricabili in serie (1,2 V x 4 = 4,8 V) oppure 3 batterie tipo AA non ricaricabili ad alta energia in serie (1,5 V x 3 = 4,5 V), anche se la

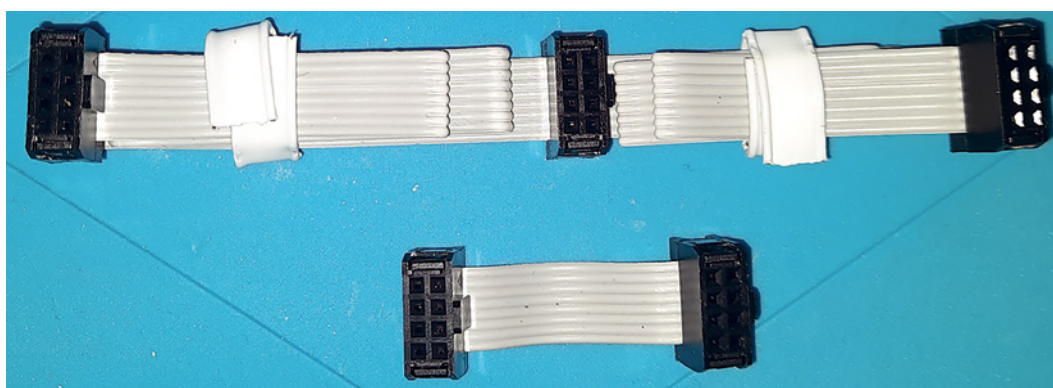


Fig. 9
Cavetti di collegamento.

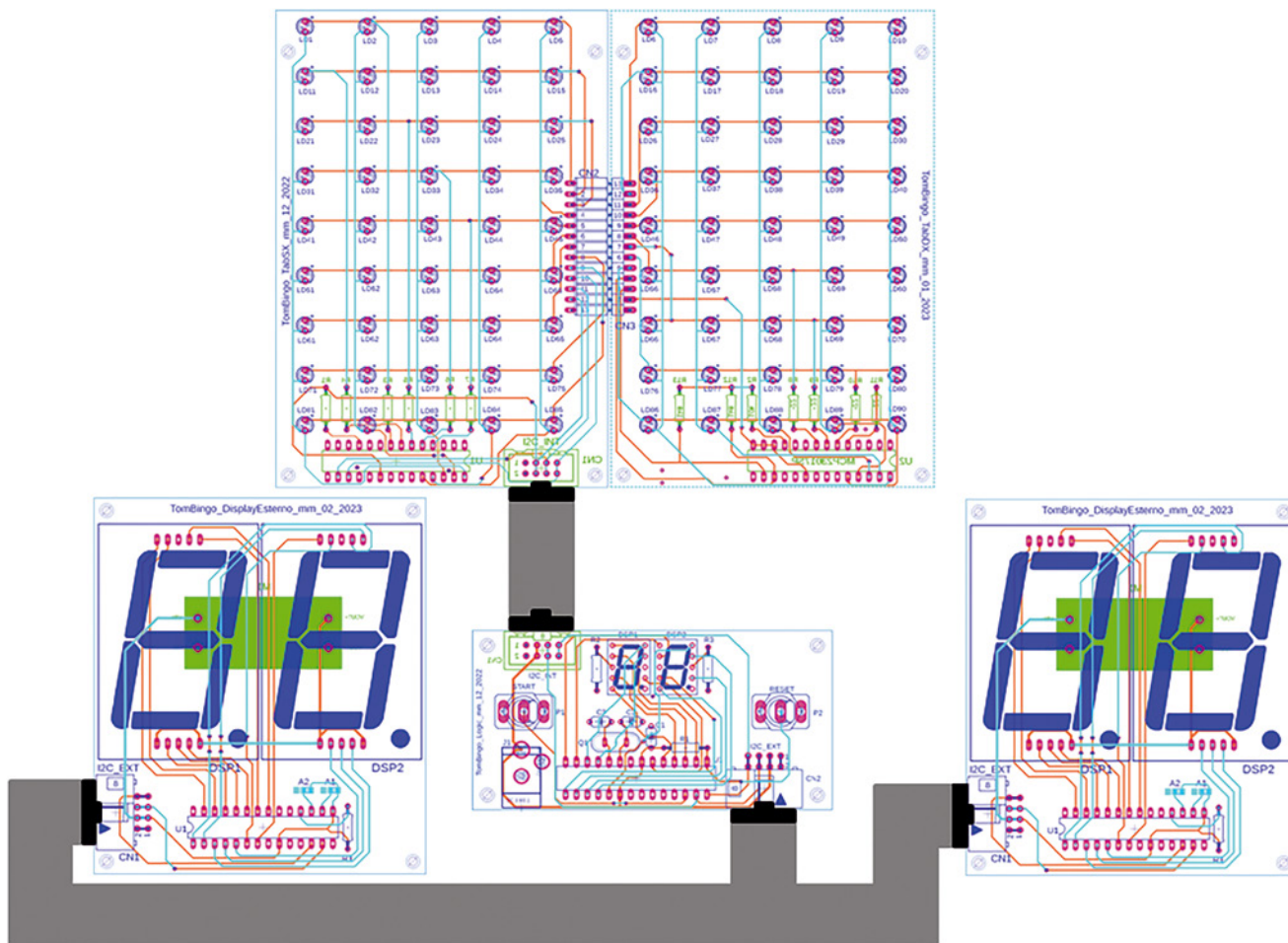


Fig. 10
Schema di
cablaggio

tensione finale è inferiore ai 5 V richiedi il circuito funzionerà ugualmente, senza particolari problemi, peraltro diminuirà anche la corrente assorbita (35 mA circa nel caso di 4,5 V) aumentando l'autonomia. L'alimentatore esterno può essere il modello 8822-AL5V2A (reperibile su futuranet.it) che ha ben 2 A di corrente erogabile, ad un costo molto contenuto. Il test del nostro TomBingo è piuttosto semplice: una volta effettuati i collegamenti, come da schema, fornire l'alimentazione alla presa PLUG DC e istantaneamente le tre coppie di display mostreranno il numero "0" o "00" (in base a come sarà stato impostato il firmware) che, essendo un numero non previsto nei due giochi, svolge semplicemente la funzione di "spia" di accensione e di test positivo. Qualsiasi altra situazione, display spenti o che mostrano caratteri casuali o altri valori, sarà indicativa della presenza di un problema, quindi occorrerà spegnere immediatamente e verificare tutto. Come ulteriore prova si dovrà premere il pulsante START e uscirà il primo numero casuale, che sarà mostrato su tutti i display, mentre sul

tabellone si accenderà il LED corrispondente. Premendo infine il pulsante RESET il LED acceso si spegnerà ed i display mostreranno di nuovo lo 0. Per testare il funzionamento dei 90 LED, occorre premere e mantenere premuto il pulsante START: ogni 3 secondi sarà estratto un nuovo numero e si accenderà il relativo LED, se non dovesse accendersi bisogna prendere nota del numero per poi controllare alla fine quelli che non vanno. Comunque le uniche possibilità di errore sono una mancata saldatura di uno dei reofori o l'inversione del LED stesso.

CONTENITORI

Abbiamo disegnato, espressamente per questo progetto, due box: uno per la Logica ed il tabellone dei LED, ed un altro per il tabellone esterno con i display giganti, con due supporti che permetteranno l'aggancio fisico al box principale e la sua rotazione. Il box principale presenta due fori per lo spinotto dell'alimentatore e per il connettore IDC, ed ha anche una mascherina superiore già dotata dei 90



← **Fig. 11**
Box e sup-
porti stampati
in 3D.

fori per i LED da 3 mm, per il doppio display piccolo e per i due deviatori a pulsante, mentre lateralmente, sia a destra che a sinistra, presenta due piccoli fori per il fissaggio dei supporti dei box dei tabelloni esterni mediante 4 viti M3 con relativi dadi. Il box di ogni tabellone esterno (bisognerà realizzarne due) presenta un foro laterale per il connettore IDC e due piccoli fori per il supporto, mentre la sua mascherina ha un riquadro dal quale dovranno sporgere i due display giganti.

Per la realizzazione di questi box mediante stampa 3D abbiamo caricato tutti i necessari file STL nella sezione download, ma chi non fosse dotato di questo tipo di periferica potrà richiedere la stampa al

service di *futuranet.it*. Poiché la mascherina del box principale non ha una serigrafia, occorrerà crearla, per esempio stampando i 90 numeri e i due testi START/RESET su un foglio adesivo, applicandolo poi alla mascherina.

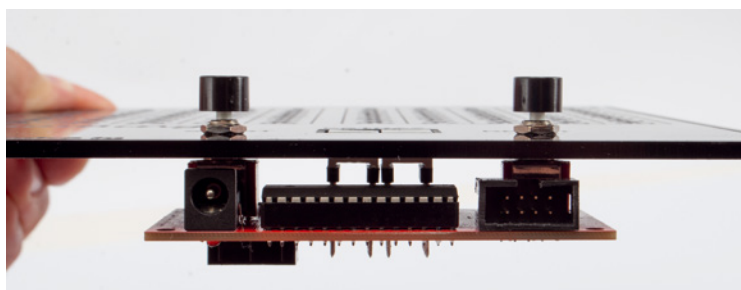
Una eccellente alternativa, quella che si può osservare nelle nostre foto, è offerta sempre dallo stesso service, e consiste nella preparazione di una mascherina in plexiglass con tanto di serigrafia, mediante incisione al laser, studiata espressamente per il box principale.

In **Fig. 11** abbiamo riportato tutti i box ed i supporti da noi stampati, mentre in **Fig. 12** si può vedere la mascherina incisa al laser.

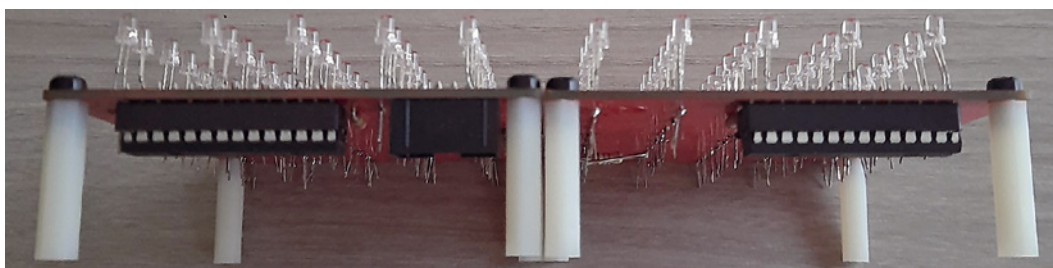


← **Fig. 12**
Mascherina
tabellone dei LED
incisa al laser.

↓ **Fig. 13**
Fissaggio PCB
Logica alla
mascherina.



→ **Fig. 14**
Il tabellone dei LED
con i distanziali.



→ **Fig. 15**
Il tabellone esterno
nel box.



A questo punto resta solo da assemblare i vari circuiti nei rispettivi box. Vediamo qualche dettaglio. Il PCB della Logica andrà fissato esclusivamente alla mascherina mediante i due dadi in dotazione ai deviatori a pulsante (**Fig. 13**) e questo garantirà la perfetta corrispondenza del connettore di alimentazione e di quello IDC con i fori presenti sul box principale. I due PCB del tabellone dei LED non necessitano di alcun fissaggio, ma solo di 8 distanziali da 25 mm da inserire nei relativi fori: garantiranno al PCB l'altezza utile per far fuoriuscire i 90 LED dai fori della mascherina, e questo sarà più che sufficiente a mantenere tutto ben bloccato nel box (**Fig. 14**). Il PCB dei display giganti invece dovrà essere fissato sul fondo del suo box, mediante 4 distanziali da 15 mm in corrispondenza dei quali occorrerà praticare 4 fori da 3 mm di diametro, per far passare le viti filettate (**Fig. 15**); la lunghezza dei distanziali è obbligata dalla posizione dell'apertura laterale per il connettore IDC, altrimenti non combacerà. Come sempre, se non saranno adottati i nostri file stl, ovviamente poi ci si regolerà di

conseguenza. A fine assemblaggio i due tabelloni esterni potranno essere facilmente ruotati in qualsiasi direzione facilitando così il compito di chi effettuerà l'estrazione, perché il numero sarà ben visibile a chiunque.

Non ci resta che augurarvi buon divertimento col nostro TomBingo!



Cosa occorre?

I componenti principali di TomBingo sono disponibili presso Futura Elettronica. Il microcontrollore programmato (cod. MF1688) è in vendita a Euro 13,00, l'IO expander (cod. MCP23017-E/SP) a Euro 4,80, il modulo step-up (cod. YXA347) a Euro 7,00, il display 7 segmenti catodo comune 10x13 mm (cod. 1440-DS1013CC) a Euro 1,20, il deviatore a pulsante (cod. 8220-8701) a Euro 3,50. 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 TENSIONE DI RETE OVUNQUE TU SIA

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

FORMA D'ONDA MODIFICATA



cod. FR670M300
€ 48,00

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

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



cod. FR670M600
€ 69,00

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

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



cod. FR670M1500
€ 148,00

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

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

FORMA D'ONDA PURA



cod. FR668P300
€ 98,00

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

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



cod. FR668P600
€ 178,00

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

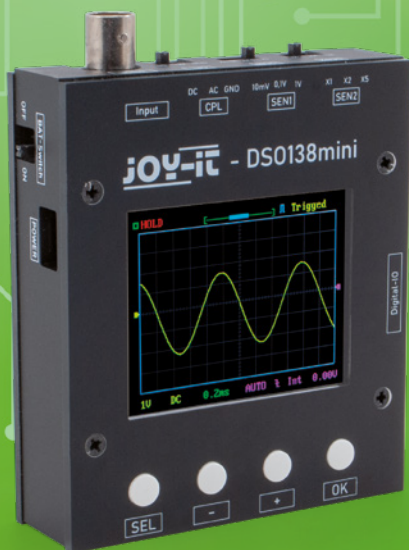
- Tensione di ingresso: 12 VDC nominali
- Tensione di uscita: 220-240 VAC - 50 Hz
- Allarme batteria scarica: 10,5-11,5 Vdc
- Dimensioni (mm): 220x115x55
- Peso: 980 grammi

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

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



€ 89,00
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



€ 199,00
cod. DCA75

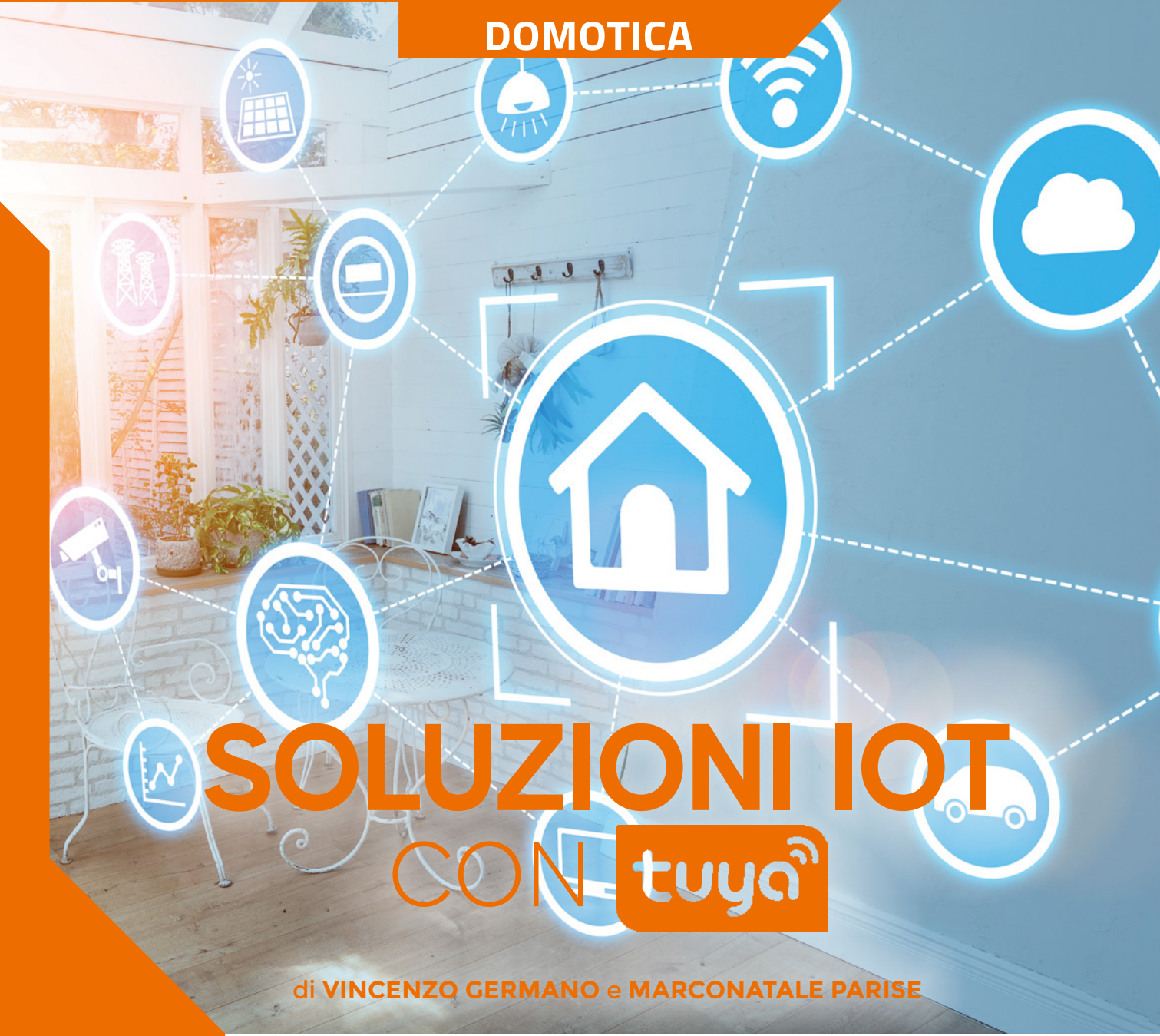
Analizza svariati tipi di semiconduttori. Può essere interfacciato al PC e, grazie al software in dotazione, fornisce un'analisi dettagliata del componente.

Misuratore di induttanza, capacità, resistenza e impedenza LCR45



€ 145,00
cod. LCR45

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



SOLUZIONI IOT CON **tuya**

di **VINCENZO GERMANO** e **MARCONATALE PARISE**

Scopriamo la tecnologia di Tuya per interconnettere i dispositivi in rete e realizzare un sistema di automazione casalingo in perfetto stile IoT.

S

e si pensa a sistemi di automazione domestica, possono venire in mente varie soluzioni, che vanno dalle opzioni fai-da-te a quelle commerciali. In ogni caso una cosa è certa: grazie a questi sistemi, è possibile ottenere numerosi vantaggi in termini di costo ed efficienza energetica. Un esempio è la possibilità di programmare il riscaldamento, il raffreddamento e l'illuminazione in base alle preferenze personali e alle effettive necessità, evitando così sprechi di energia. A tutto ciò si aggiunge il rilevamento della presenza, con una conseguente attivazione/disattivazione automatica dei dispositivi quando una persona entra o esce da una stanza. Questo aiuta a evitare che le luci o gli apparecchi rimangano

accesi inutilmente. L'immediata conseguenza è un risparmio economico, che implica minori spese energetiche e un prolungamento della vita utile degli apparecchi, riducendo la necessità di sostituzioni premature.

Un altro aspetto da tenere in considerazione è il controllo remoto e la programmazione. Grazie all'implementazione di questi sistemi si riescono a controllare i dispositivi domestici da remoto, tramite smartphone o tablet, permettendo di gestire l'ambiente domestico anche quando non si è fisicamente presenti. Si riesce ad attuare una programmazione personalizzata, andando ad agire sul funzionamento di dispositivi come l'illuminazione, il riscaldamento e l'aria condizionata in base a orari specifici o alle routine quotidiane.

Ad esempio, si può impostare il sistema per accendere automaticamente le luci all'imbrunire o abbassare la temperatura durante la notte. Tutto ciò porta a un aumento di comfort grazie alle possibilità di personalizzazione degli ambienti e all'automazione di determinate *routine* per eseguire una serie di azioni in sequenza a seconda degli ambienti d'interesse.

Infine un altro vantaggio rilevante da non sottovalutare in questo contesto è la sicurezza e la prevenzione, infatti un sistema di automazione casalingo può includere sistemi di sicurezza come telecamere di sorveglianza, sensori di allarme e monitoraggio delle porte/finestre. Questi dispositivi aiutano a prevenire le intrusioni indesiderate inviando notifiche e avvisi in tempo reale sullo stato della propria abitazione, come ad esempio avvisi di fumo rilevato, allarmi di sicurezza attivati o anomalie nei consumi energetici. Come già anticipato, facendo una considerazione generale, le possibili implementazioni possono essere commerciali (con sistemi già realizzati e pronti all'uso), oppure hobbistiche (con soluzioni "home made"). Considerando la seconda delle due opzioni, ovvero l'implementazione del sistema sviluppando la propria interfaccia di controllo, questo può risultare non banale, complicato e costoso.

Se si prende in considerazione di dover progettare e sviluppare il proprio circuito stampato, andare ad implementare un protocollo di comunicazione tra i dispositivi e il server, sviluppare la propria applicazione per comunicare con il server e via dicendo, le cose si complicano ulteriormente.

Tuttavia c'è una possibilità intermedia per tutto ciò, ovvero l'utilizzo di Tuya Smart (www.tuya.com) associata a dispositivi di utilizzo comune nell'Open Hardware.

Questa strada permette di avere già a disposizione (tra le altre cose) un server sviluppato per i sistemi IoT (*Internet of Thing*), moduli pre-programmati per una facile connessione al cloud, un'applicazione mobile personalizzabile con diverse funzionalità (con un design visivo accattivante e facile da usare), e una registrazione gratuita del proprio dispositivo su Internet in pochi clic.

Con tale tecnologia andiamo a realizzare un semplice sistema per la gestione di un relè che può essere utilizzato per vari scopi, come l'apertura di una porta mediante l'attivazione di una serratura, oppure per la gestione di "carichi" d'interesse all'interno di un proprio sistema di automazione casalingo. Nell'articolo si vogliono fornire le linee generali per poter utilizzare tale tecnologia e gestirla al meglio per i propri progetti, perciò verranno presentati i punti chiave e i passaggi principali per poter poi riadattare questa tecnologia alle proprie esigenze. Tuttavia prima di presentare una possibile implementazione pratica, approfondiamo brevemente le potenzialità e le possibilità di Tuya.

TUYA

L'azienda è stata fondata nel 2014 da un gruppo di imprenditori cinesi con l'idea di rendere l'*Internet of Things* (IoT) accessibile a tutti e semplificare la creazione di dispositivi smart connessi. Questo perché molti dispositivi intelligenti erano costosi e complessi da installare e configurare, mentre con l'utilizzo di Tuya si ha una piattaforma (All-in-One o tutto in uno) che semplifica la creazione e l'integrazione di dispositivi smart, includendo anche SDK (Software Development Kit), API e altre risorse che i produttori di dispositivi stessi possono utilizzare per aggiungere funzionalità IoT ai propri prodotti. Questo ha permesso alle aziende di concentrarsi sulla creazione di prodotti innovativi piuttosto che sulla "complessità" della connettività. Un tale ecosistema consente quindi ai produttori di elettrodomestici, illuminazione, sicurezza domestica e molti altri dispositivi, di trasformare "rapidamente" i loro prodotti in dispositivi smart e connessi. Ciò significa che, grazie all'integrazione con Tuya, questi dispositivi possono essere ad esempio controllati e monitorati da remoto tramite smartphone o tramite assistenti vocali come Amazon Alexa, Google Assistant, o altri.

Facendo una breve panoramica di alcune delle caratteristiche chiave come piattaforma IoT, ritroviamo:

1. SDK e API: forniti per semplificare il processo di sviluppo di soluzioni IoT, consentendo ai prodotto-

ri di dispositivi di aggiungere facilmente funzionalità di connettività e controllo remoto ai loro prodotti esistenti o di sviluppare nuovi dispositivi intelligenti.

2. App Tuya Smart: un'applicazione per smartphone che permette agli utenti di controllare e gestire i dispositivi smart collegati alla piattaforma Tuya; l'app è disponibile per dispositivi iOS e Android ed è compatibile con un'ampia gamma di dispositivi smart.

3. Integrazione con assistenti vocali: i dispositivi smart basati su Tuya possono essere integrati con assistenti vocali, permettendo agli utenti di controllarli usando semplicemente comandi vocali, creando un'esperienza utente più completa e connessa.

4. Sistemi operativi compatibili: supporta diversi sistemi operativi, tra cui Linux, FreeRTOS e Android, rendendo la piattaforma adatta a una vasta gamma di dispositivi smart.

5. Sicurezza: pone una forte enfasi sulla sicurezza dei dati e della connessione, infatti gli standard adottati aiutano a garantire che i dati degli utenti e i dispositivi connessi siano protetti da eventuali minacce di sicurezza.

Tutto ciò ha accelerato la transizione verso la produzione di dispositivi maggiormente intelligenti e connessi, contribuendo all'espansione e alla diffusione dell'IoT a livello globale, permettendo a Tuya di raggiungere una vasta portata e presenza internazionale. Questo è stato possibile anche grazie a collaborazioni con un'ampia gamma di marchi e produttori di dispositivi smart.

IL PROGETTO

Passiamo a questo punto alla realizzazione di un progetto concreto partendo dal presentare l'idea, per poi utilizzare la piattaforma online per la generazione delle informazioni utili alla connettività dei dispositivi (*Tuya IoT Platform*). Grazie a questi passaggi andremo a comprendere come il focus principale di Tuya sia orientato alla connessione dei dispositivi a Internet (permettendo in poco tempo di creare un'intera soluzione automatizzata). Vedremo come tutto il controllo viene effettuato tramite smartphone mediante un'interfaccia accattivante, pulita e piacevole, offrendo una soluzione completa mediante l'utilizzo di Arduino, ESP8266, ESP32 o affini.

Come anticipato sin qui con Tuya si ha la possibilità di utilizzare board Open Hardware per i propri progetti. Facendo riferimento alla **Fig. 1A**, vediamo l'idea di concetto di un tale sistema, in cui vengono

riportati i dispositivi in gioco:

1. Una board: (MCU) che può essere una scheda come Arduino o altre, sulla quale si andrà ad inserire il proprio codice di progetto (nel nostro caso ci sarà il software per la gestione della scheda relè).

2. Modulo WiFi con codice Tuya: (Tuya Module) una board generica del tipo ESP32 o ESP8266 o altre (che abbia una connettività WiFi o Bluetooth), nella quale si andrà a scaricare il software Tuya. Grazie a questo modulo ci si potrà collegare alla piattaforma Tuya opportunamente configurata e si potrà gestire il tutto mediante un'applicazione su smartphone.

In **Fig. 1B** viene riportata una possibile implementazione con un Arduino UNO ed un NodeMCU collegati tramite seriale.

L'hardware necessario è possibile acquistarlo sul sito di futurashop.it, con i seguenti codici: ARDUINO NOUNOREV3, NODEMCU e RELAY1CH.

Per la connessione tra le due board si faccia riferi-

Fig. 1A/1B
Il funzionamento di Tuya.

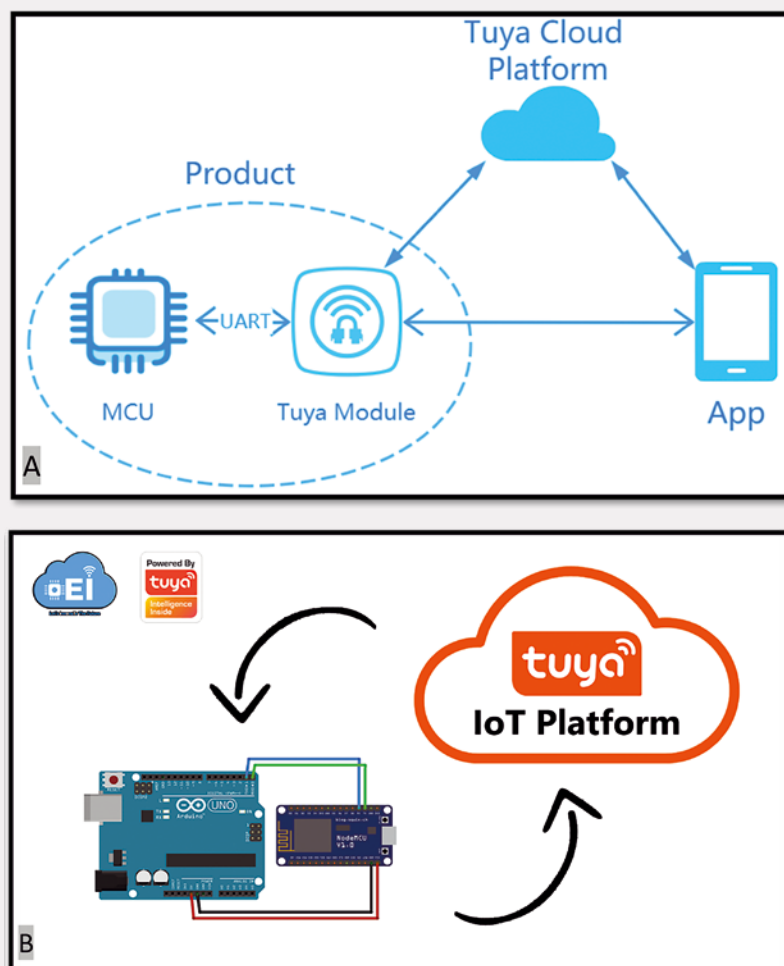


Tabella 1
Connessione
tra Arduino ed
ESP8266.

ESP8266	ARDUINO
TX	RX
RX	TX
GND	GND
VIN	5v

mento alla **Tabella 1**. Un ulteriore collegamento va fatto tra la board Arduino e il pin della scheda relè, in questo caso utilizzeremo il pin 6 di Arduino per attivarlo.

A questo punto l'hardware è pronto, dobbiamo solo configurare il tutto per poterlo utilizzare. Per fare ciò partiamo dalla *Tuya IoT Platform*

TUYA IOT PLATFORM

Per poter utilizzare i servizi offerti dalla piattaforma Tuya, bisogna collegarsi al sito <https://iot.tuya.com> e registrarsi gratuitamente (i passaggi sono obbligati perciò vengono tralasciati).

Fatto ciò si può effettuare il login al sito e ci si troverà davanti la pagina generale della piattaforma, sulla barra a sinistra bisogna cliccare su **"Product"** e poi selezionare **"Development"**, in modo tale da essere riportati nella schermata di creazione del nuovo prodotto, come riportato in **Fig. 2**.

La piattaforma aiuta la creazione di prodotti *IoT* in un processo semplice ed integrato con tutte le operazioni necessarie per ottenere la connessione al cloud. Questo processo di sviluppo altamente integrato consente di eseguire facilmente la definizione delle funzioni, la creazione del modello di dati

dei prodotti, il pannello del dispositivo, lo sviluppo dell'hardware, il debug del dispositivo, la configurazione del prodotto, il test del prodotto e il marketing del prodotto, tutto in un unico posto.

Come da **Fig. 2**, viene offerta una serie di soluzioni chiavi in mano per risparmiare tempo nella scelta del protocollo e delle funzioni giuste in base alle esigenze specifiche. Queste soluzioni collaudate e pronte all'uso coprono quasi tutti i campi come illuminazione, sensori, elettrodomestici, sistemi di sicurezza, serrature delle porte, hub di controllo, risparmio energetico, nonché protocolli popolari come Wi-Fi, Bluetooth e Zigbee.

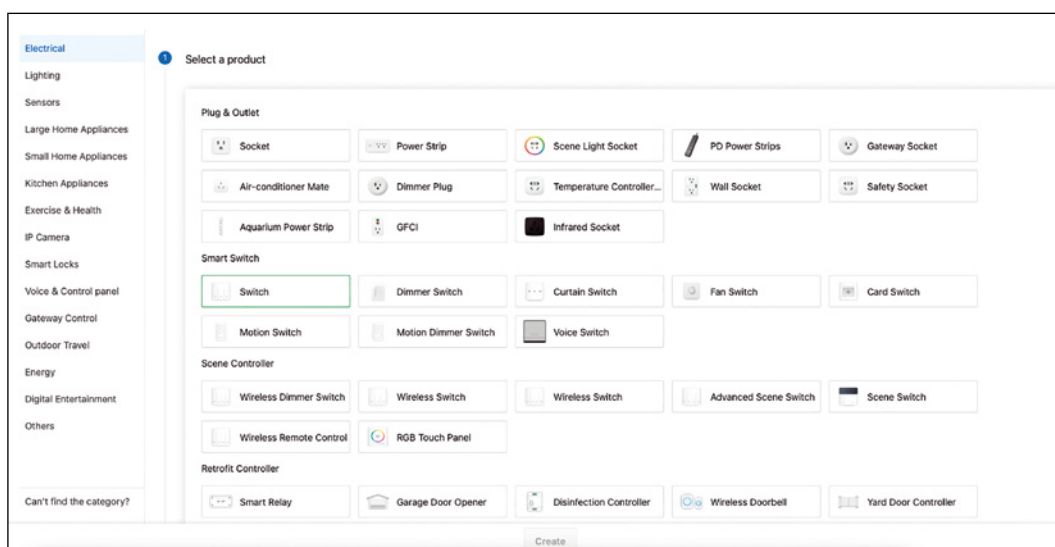
Un "prodotto" è una rappresentazione astratta di una raccolta di dispositivi fisici che hanno le stesse configurazioni e proprietà. A ogni prodotto creato sulla piattaforma di sviluppo Tuya IoT viene assegnato un **ID prodotto univoco (PID)**. Tale numero viene assegnato dopo la creazione del prodotto stesso e solo successivamente si può procedere con il suo sviluppo.

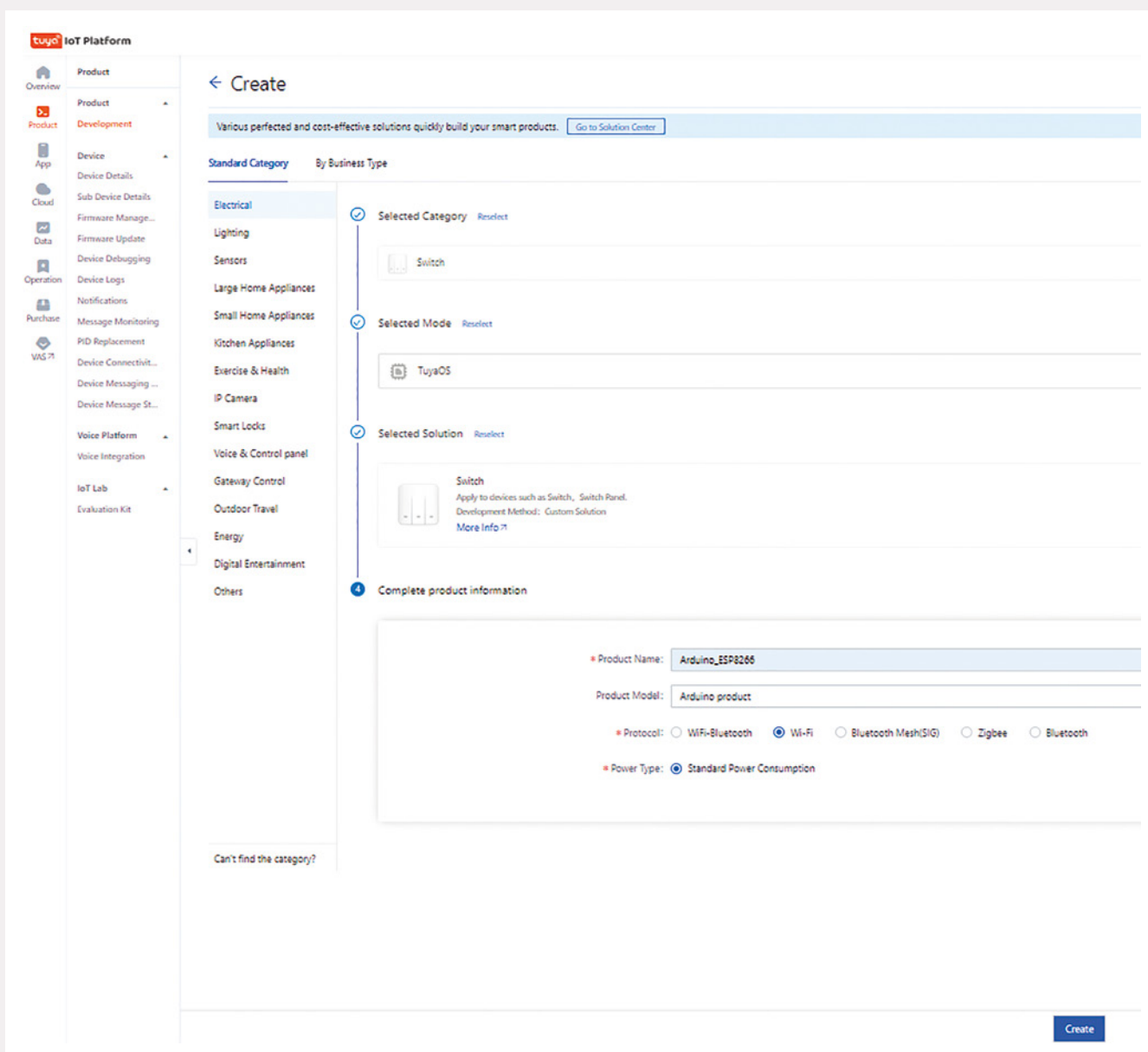
Vediamone i vari passaggi.

Nel nostro caso selezioniamo **"switch"** come da **Fig. 3**, poi nella selezione successiva **"Mode"** clicchiamo su **"TuyaOS"**, poi in **"Selected Solution"** clicchiamo su **"custom"** e infine su **"Switch"**.

Nelle informazioni di prodotto inseriamo il nome che preferiamo per il nostro prodotto (nell'esempio in **Fig. 3** Arduino_ESP8266) e come protocollo utilizziamo solo il WiFi, per arrivare ad una configurazione come da **Fig. 3**. A questo punto viene generato il prodotto d'interesse e verremo riportati in una nuova schermata per l'associazione delle funzionalità dello specifico prodotto **"Switch"** come da **Fig. 4**.

Fig 2
Schermata
di selezione
del prodotto.





Nel nostro esempio associamo solo la funzionalità di accensione/spengimento dello switch ma in realtà si possono assegnare tutte le funzionalità della lista o aggiungerne di nuove, creandole con l'apposito editor presente nella piattaforma a seconda delle necessità di progetto. Con riferimento alla **Fig. 5**, bisogna sempre tener presente che:

1. **Data Point**: rappresenta le funzioni di dispositivi intelligenti e i corrispondenti parametri;
2. **DP ID**: indica il codice di un "Data Point"; il trasferimento dei dati delle funzioni tra il dispositivo e il cloud avviene tramite questo parametro ID;
3. **DP Name**: indica il nome di una funzione;
4. **Identifier**: indica il valore del codice di un "Data

Point", visualizzando il nome della funzione in più lingue sull'app;

5. Data type.

- a. **Boolean**: indica il valore della variabile binaria (vero/falso); ad esempio: funzione di commutazione (On o Off), come nel nostro caso;
- b. **Integer**: rappresenta i dati che possono essere regolati linearmente; ad esempio: regolazione della temperatura, da 20 a 40 °C;
- c. **Enum**: indica il valore dell'insieme finito personalizzato; ad esempio: livello di lavoro (basso, medio e alto);
- d. **Fault**: specificato per la segnalazione dei guasti e le statistiche; segnala solo quando si

Fig. 3
Selezione
dello switch.



Fig. 4
Funzionalità
da associare
allo switch.

Standard Functions	Other Functions	Select All	Selected Function(1)	Remove Optional Function
DP ID: 22 Identifier: cur_voltage		>	Switch DP ID: 1 Identifier: switch_1	X
Current power DP ID: 23 Identifier: cur_power		>		
test bit DP ID: 24 Identifier: test_bit		>		
voltage coe DP ID: 25 Identifier: voltage_coe		>		
electric coe DP ID: 26 Identifier: electric_coe		>		
power coe DP ID: 27 Identifier: power_coe		>		
electricity coe DP ID: 28 Identifier: electricity_coe		>		

OK

verificano più guasti;

- e. Char type:** indica il trasferimento dei dati nel tipo char;
- f. RAW type:** indica il trasferimento di dati in un formato binario non elaborato; in generale, si consiglia di utilizzare il trasferimento di dati grezzi quando gli altri cinque tipi non sono in grado di far fronte alle complicate funzioni del prodotto.

6. Data transfer type:

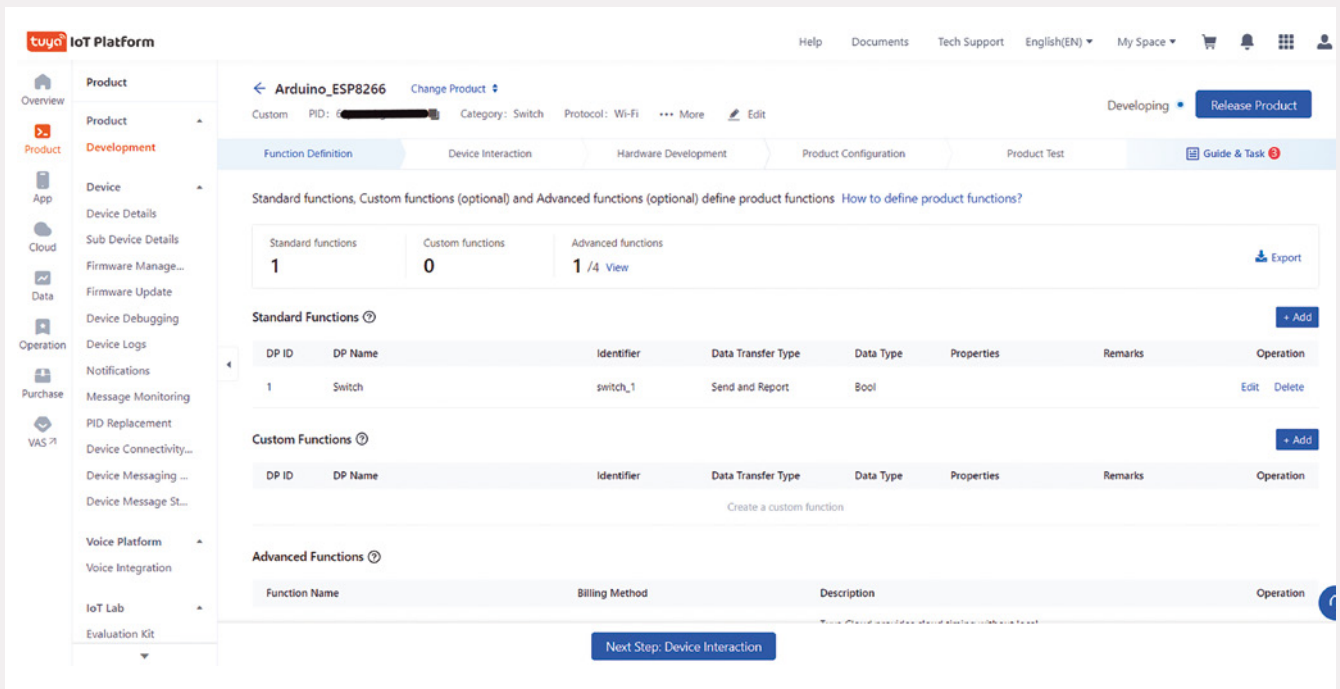
- a. Issue and report:** indica il trasferimento bidirezionale dei dati tra il dispositivo e il cloud;
- b. Only report:** indica il trasferimento di dati unidirezionale; possono essere riportati solo dal dispositivo;
- c. Only issue:** indica il trasferimento di dati unidirezionale; possono essere emessi solo dal cloud.

Se si volessero aggiungere o variare le funzionalità del dispositivo bisognerà farlo sempre in "Function Definition", che è l'astrazione di una funzionalità fornita da un device intelligente. È possibile utilizzare le funzioni del prodotto per descrivere le caratteristiche e i parametri. Sulla base delle funzioni, la piattaforma Tuya IoT genera modelli di dati nel cloud per connettersi e controllare i dispositivi intelligenti. Ciò semplifica la comprensione delle funzioni e delle interazioni con i dati.

A questo punto, sempre facendo riferimento alla **Fig. 5**, selezionando il tasto blu in fondo alla pagina, ovvero "Next Step: Device Interaction", si viene riportati nella schermata di **Fig. 6**, che riguarda la configurazione del pannello di controllo su smartphone. Ci sono già dei pannelli standard da poter selezionare che verranno popolati a seconda delle funzioni utilizzate nel passo precedente; in questo caso si è scelto di utilizzare il "Light Switch 4". Una volta confermata la selezione si viene riportati in una nuova pagina, ovvero quella di **Fig. 7**. Un "pannello" è un pannello di controllo sull'applicazione mobile di un prodotto intelligente, esso mostra l'interfaccia utente interattiva per consentire di accedere alle funzioni del prodotto; ad esempio, se il prodotto aggiunge un timer, il suo pannello mostra anche il timer corrispondente. Dalla schermata di **Fig. 7**, si può decidere di fare un editing dell'interfaccia finale sullo smartphone in modo custom, cambiare il tema oppure associare il "Voice Control". Questi argomenti saranno tralasciati non essendo legati al nostro progetto, tuttavia nella sezione "Doc" del sito ufficiale sono presenti delle guide esaustive per tutti i passaggi d'interesse.

Se si installa sul proprio smartphone l'app "Tuya Smart", e dopo aver inserito il proprio account si effettua la scansione del QR-code appena generato, si ottiene l'interfaccia di **Fig. 8**.

Mediante questa interfaccia si può gestire il



progetto che stiamo creando ovvero la gestione remota di un relè. Ultimo passaggio è cliccare sul tasto blu in fondo alla pagina di **Fig. 7** e nella schermata successiva di "Hardware Development" selezionare "MCU SDK".

LA PIATTAFORMA SMART PMS

Abbiamo tutto per passare allo step successivo grazie al **PID** appena generato e recuperabile in alto a sinistra come da **Fig. 5**.

Per generare il Token ID da utilizzare successivamente, bisogna mandare una mail all'indirizzo devops@tuya.com inserendo i seguenti dati:

- **Account:** l'email inserita per creare l'account al passo precedente su *Tuya IOT Platform*;
- **PID:** il codice alfanumerico appena recuperato.

Dopo un tempo variabile che può andare da svariate ore a qualche giorno, in risposta a questa mail vi verrà inviato il Token ID associato da inse-

Fig. 5
Funzionalità associate allo switch.

Fig. 6
Selezione del pannello di controllo dello smartphone.

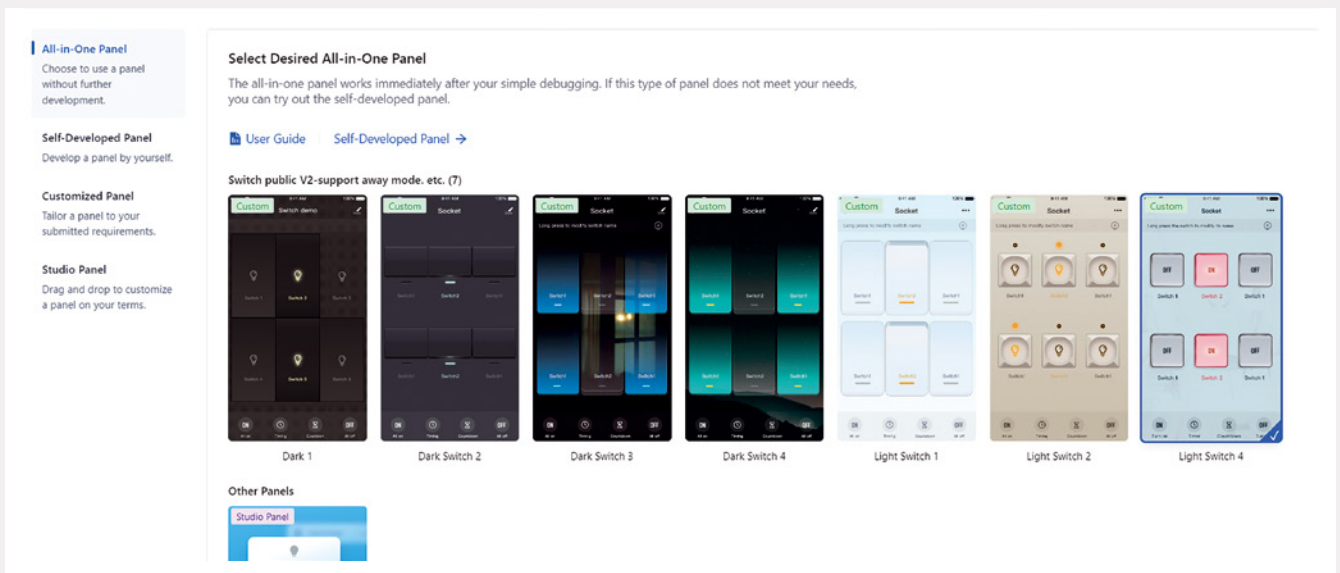


Fig. 7
Pannello di controllo per smartphone autogenerato.

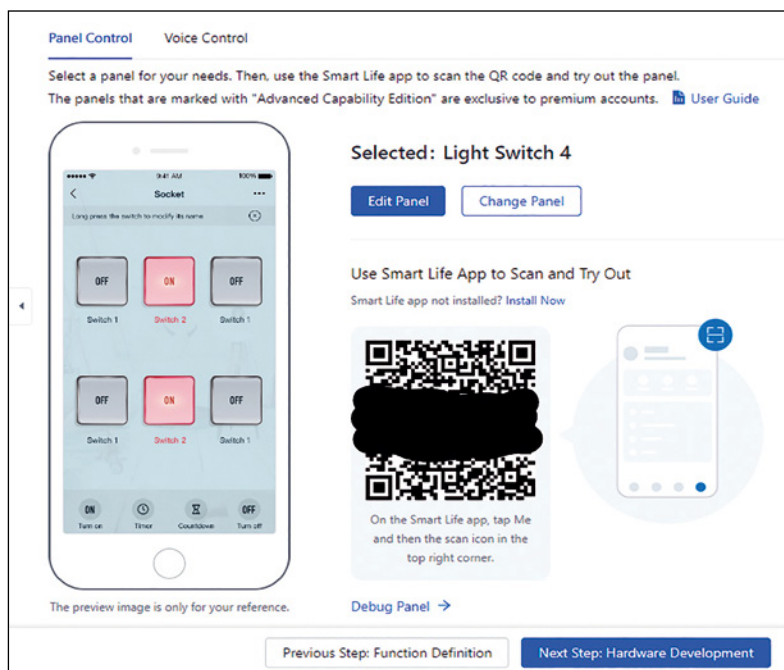


Fig. 8
Pannello di controllo da smartphone.

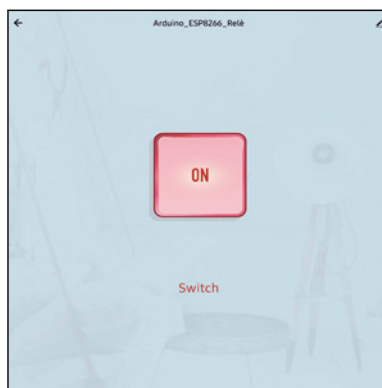


Fig. 9
Download del TYProduction Toolkit.

Smart PMS					Help document	ENGLISH	SmartPMS.com
Production Manage	Audio Production Test Tool	-	V1.0.0	38.075MB	download		
Quality Mngmt	Timer Gadget	-	V1.0.0	0.97MB	download		
Knowledge Base	BK Dongle Verification Tool	-	V1.0.0	8.34MB	download		
Email Subscription	RF Gold Plate File Processing Tool	-	V1.0.0		download		
Tutorials	XR806 Dongle programming calibration tool	-	V1.0.0	162.985MB	download		
SMT Material	Dual Gateway Zigbee RF Tools	-	V1.0.0	114.121MB	download		
Production Test Guide	Tuya RF Test Custom.BK7271	-	V1.1.2		download		
Software Download	TYProductionToolkit	-	V1.1.15.0	69.621MB	download		
Download Driver	Firmware Upgrade tool	-	V1.0.0	0.689MB	download		
Download Firmware	Base firmware tool	-	V1.0.0	0.867MB	download		
Document Management	dongle-S4	-	V1.1.1	0.009MB	download		
Service Support	BK7231N	-	V2.0.4	1.093MB	download		
	AK801803 Chip Burning Tool	-	V1.0.2	4.869MB	download		
	RTL8721DN_KING_WLAN Facility	-	V1.0.1	173.98MB	download		
	dongle-S1	-	V1.0.1	0.007MB	download		
	AlmaTool	-	V1.1.5.1	16.805MB	download		
	None	-	V2.1.0	0MB	download		

rire nel sito <https://pms.tuya.com>.

A seguito di opportuna registrazione al sito, per attivare il certificato del prodotto (una volta effettuato l'accesso), portarsi nella schermata principale e cliccare in sequenza su "Production Manage" poi su "Work Order Management" ed infine su "Activation Code Verification".

A questo punto inserire il Token ID ricevuto via mail e cliccare sul tasto "Confirm"; fatto ciò, vi verrà visualizzata una schermata di conferma. Solo allora l'ESP8266 è pronto per essere aggiornato con il software corrispondente ed autorizzato come Tuya Module di **Fig. 1**.

Per aggiornare l'ESP8266, si possono seguire due strade, la prima prevede di scaricare il pacchetto con il software dedicato dal seguente indirizzo drive:

<https://drive.google.com/file/d/1SF-rM5qDLAPJiuMHrkqFauVvRXSSHMkj/view?usp=sharing>; all'interno di tale pacchetto è presente solo ed esclusivamente il software d'interesse con l'aggiunta di driver USB CP210x universali.

Mentre la seconda strada prevede il download del pacchetto software direttamente dal sito <https://pms.tuya.com>, selezionando "Knowledge Base Management" e poi "Software Download" ci si ritrova nella schermata di **Fig. 9**. Ricercare il pacchetto "TYProductionToolkit" e avviare il download. Dopo averlo installato, avviarlo ed inserire il proprio account di registrazione al sito che vi permetterà di visualizzare quanto riportato in **Fig. 10**.

In configurazione "Production" selezionare l'icona TYPB, che andrà a scaricare ed avviare un tool in versione aggiornata di quanto presente nel

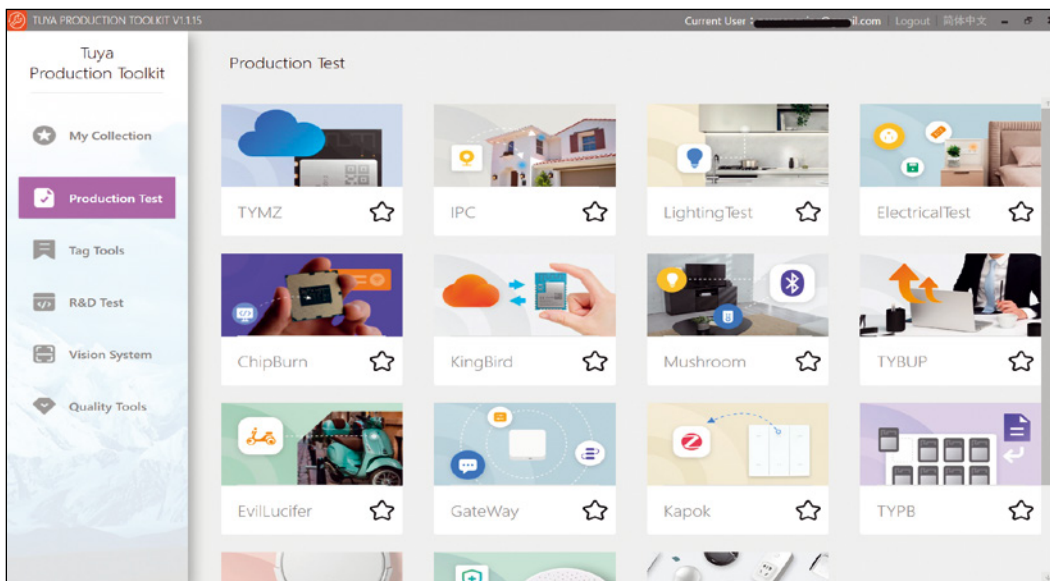


Fig. 10
Interfaccia principale del programma TYProduction Toolkit.

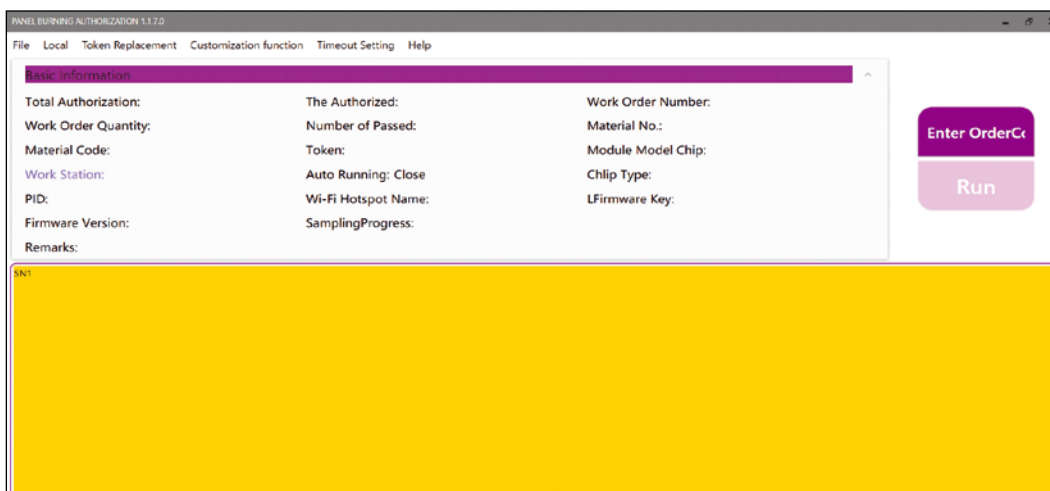


Fig. 11
Il programma di aggiornamento software TYPB.

pacchetto drive appena riportato.
Non rimane altro che collegare il NODEMCU alla porta USB, inserire il Token ID verificato al passo precedente cliccando "Enter OrderCode" e nel menù a scorrimento impostare "Burning Authorization". Infine cliccando su "RUN", previa selezione della porta COM corretta, viene avviata la procedura di download del software su ESP8266 che durerà qualche secondo; a procedura completata potete scollegare il dispositivo pronto all'uso come *Tuya Module*.

LA CONFIGURAZIONE DI ARDUINO

Passiamo brevemente alla configurazione di Arduino UNO utilizzando un esempio già presente e ben strutturato, disponibile nelle librerie

Listato 1

```
unsigned char pid[] = {"ma6719sgmdyg3d2k"};
unsigned char mcu_ver[] = {"1.0.0"};

/* Current LED status */
unsigned char led_state = 0;

/* Connect network button pin */
int key_pin = 7;

/* Gestione Relè */
int relay_pin = 6;
unsigned char relay_state = 0;

/* Data point define */
#define DPID_SWITCH 1
```

Listato 2

```
void setup()
{
  Serial.begin(9600);
  //Initialize led port, turn off led.
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, LOW);
  //Inizializziamo l'uscita per la gestione Relè
  pinMode(relay_pin, OUTPUT);
  digitalWrite(relay_pin, LOW);
  //Initialize networking keys.
  pinMode(key_pin, INPUT_PULLUP);
  //Enter the PID and MCU software version
  my_device.init(pid, mcu_ver);
  //incoming all DPs and their types array, DP numbers
  my_device.set_dp_cmd_total(dp_array, 1);
  //register DP download processing callback function
  my_device.dp_process_func_register(dp_process);
  //register upload all DP callback function
  my_device.dp_update_all_func_register(dp_update_all);
  last_time = millis();
}
```

Listato 3

```
unsigned char dp_process(unsigned char dpid, const unsigned char value[],
unsigned short length)
{
  switch(dpid) {
    case DPID_SWITCH:
      relay_state = my_device.mcu_get_dp_download_data(dpid, value, length);
      if (relay_state) {
        //Turn on
        digitalWrite(relay_state, HIGH);
      } else {
        //Turn off
        digitalWrite(relay_state, LOW);
      }
      //Status changes should be reported.
      my_device.mcu_dp_update(dpid, value, length);
      break;
      default: break;
  }
  return TY_SUCCESS;
}
```

dedicate di Tuya. Nell'IDE di Arduino, avviare il gestore librerie, ricercare ed installare la libreria "TUYA_WIFI_MCU_SDK". A procedura completa, tra i vari esempi disponibili per questa libreria selezionare ed avviare "Start". Per un corretto funzionamento, abbiamo la necessità di cambiare il PID fornito in questo codice di esempio con il PID appena generato sulla piattaforma *Tuya IoT* ed in più anche la versione MCU. Nel codice del **Listato 1** troviamo le due stringhe da modificare come *pid[]* ed *mcu_ver[]*. Mentre per la gestione del relè dichiarare una nuova variabile *int relay_pin* (che utilizzeremo come porta di Arduino e la imponiamo come 6) e una *relay_state* (per lo stato del relè). Infine cambiamo la variabile *DPID_SWITCH* in 1 (al posto di 20), perché definita così nella **Fig. 5**.

Passando al **Listato 2**, all'interno del blocco di *setup()* definiamo come uscita il pin per la gestione relè appena dichiarato con "*pinMode(relay_pin, OUTPUT);*" e impostiamolo come "LOW". Infine nel **Listato 3** configuriamo le azioni da compiere nell'interazione con l'app su smartphone andando a gestire la variabile *relay_state* a seconda del *DPID_SWITCH* definito in precedenza. Con quest'ultimo passaggio abbiamo chiuso il progetto d'esempio. Per quanto sembrano macchinosi e lunghi i vari passaggi, in realtà prendendo confidenza con gli strumenti della piattaforma le modifiche che si possono apportare al progetto sono immediate, ad esempio per

la gestione di LED aggiuntivi anche RGB con funzionalità di dimming, timer e molto altro ancora.

CONCLUSIONI

L'approccio innovativo e l'impegno di Tuya nell'abbattere le barriere all'ingresso nell'IoT hanno contribuito al suo successo. La piattaforma Tuya ha aiutato numerosi produttori a creare dispositivi intelligenti in settori come l'illuminazione, la sicurezza domestica, il riscaldamento, la climatizzazione e molti altri. L'obiettivo di Tuya di democratizzare la tecnologia IoT e di creare un ecosistema connesso ha avuto un impatto significativo sull'evoluzione delle case intelligenti e dei dispositivi smart in tutto il mondo. ■



Cosa occorre?

La board Arduino UNO utilizzata nel progetto (cod. ARDUINOUNOREV3) è disponibile a Euro 27,50, la scheda con ESP8266 (cod. NODEMCUESP) a Euro 9,50, la scheda relè (cod. RELAY1CH) a Euro 5,00. 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

IoTicontrollo

DISPOSITIVI INNOVATIVI STUDIATI PER METTERE IN COMUNICAZIONE APPARATI E OGGETTI CON LA RETE INTERNET PER SEMPLIFICARE LA VITA DI TUTTI I GIORNI.

POWER CONTROL IOT



cod. PWC3NV000

€119,00

Collegato alla rete elettrica, permette di essere avvisati da remoto, tramite apposita APP, di una eventuale mancanza di corrente e di intervenire tempestivamente. Il dispositivo dispone anche di una uscita a contatto pulito molto utile per attivare da remoto dispositivi elettrici. Dispone di una SIM on chip con durata di connettività illimitata.

SECURITY IOT BOX



cod. SEC3NV000

€119,00

Ideale per proteggere gli oggetti dai furti. Basato su un accelerometro, Security IOT BOX è in grado di rilevare e segnalare tempestivamente, tramite l'apposita APP, eventuali spostamenti degli oggetti sui quali viene posizionato. Per funzionare non necessita di Bluetooth o Wi-Fi; dispone infatti di una SIM on chip con durata di connettività illimitata. Funziona con due batterie a stilo tipo AA (incluse). Autonomia circa 18 mesi.

SMOKE DETECTOR IOT



cod. SMK3NV000

€149,00

È in grado di rilevare la presenza di fumi potenzialmente pericolosi segnalandolo, tempestivamente tramite l'apposita APP. Per funzionare non necessita di Bluetooth o Wi-Fi; dispone infatti di una SIM on chip con durata di connettività illimitata. Funziona con una batteria al litio da 3V (incluse). Autonomia circa 18 mesi.

CONNETTIVITÀ ILLIMITATA

nessun abbonamento



APP GRATUITA



Prezzi IVA inclusa.

Precisione fino all'ultimo grammo!

Celle di carico ed elettronica necessari per la misura del peso di oggetti in moltissimi ambiti dalle bilance elettroniche alle pesi industriali.

Cella di carico 50 kg

Composta da una struttura in metallo e da piccoli elementi chiamati estensimetri (strain gauges) montati in precisi punti sulla struttura. Fornisce in uscita un segnale elettrico che varia in funzione della deformazione che la forza, applicata sul componente, produce.



€ 9,90

Cod. LC50KG

Cella di carico tipo "S" range 1 Tonnellata

Cella di carico di tipo "S" realizzata in acciaio inossidabile, classe di precisione C3, range 1 tonnellata e due fori filettati M12. Fornisce in uscita un segnale elettrico che varia in funzione della deformazione che la forza (trazione o compressione) applicata sul componente produce.



€ 64,00

Cod. LCST1TONS

DISPONIBILE ANCHE IN DIFFERENTI RANGE:

500 Kg	cod. LCST500KG	€ 64,00
200 Kg	cod. LCST200KG	€ 64,00
100 Kg	cod. LCST100KG	€ 64,00

Cella di carico 10 kg

Composta da una struttura in metallo e da piccoli elementi chiamati estensimetri montati in precisi punti sulla struttura. Fornisce in uscita un segnale elettrico che varia in funzione della deformazione che la forza applicata sul componente produce.



€ 7,00

Cod. 10KGLOADCELL

DISPONIBILE ANCHE IN DIFFERENTI RANGE:

20 Kg	cod. 20KGLOADCELL	€ 7,00
5 Kg	cod. 5KGLOADCELL	€ 7,00
1 Kg	cod. 1KGLOADCELL	€ 7,00

Con elettronica integrata o separata

Kit cella di carico con elettronica

Set composto da una scheda elettronica di interfaccia e da una cella di carico da 20 kg montata all'interno di una struttura meccanica. La scheda elettronica è basata sul chip HX711, un convertitore analogico-digitale a 24 bit, specifico per interfacciare la cella di carico a un microcontrollore come Arduino, Fishino o compatibili.

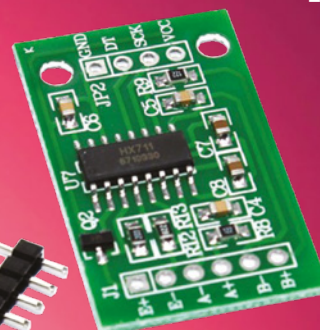


€ 28,00

Cod. LOADCELLKIT

Scheda elettronica con chip HX711 per cella di carico

Scheda elettronica basata sul chip HX711, un convertitore analogico-digitale a 24 bit, specifico per interfacciare la cella di carico a un microcontrollore come Arduino, Fishino o compatibili. Dimensioni: 33,6x20,5x3 mm.



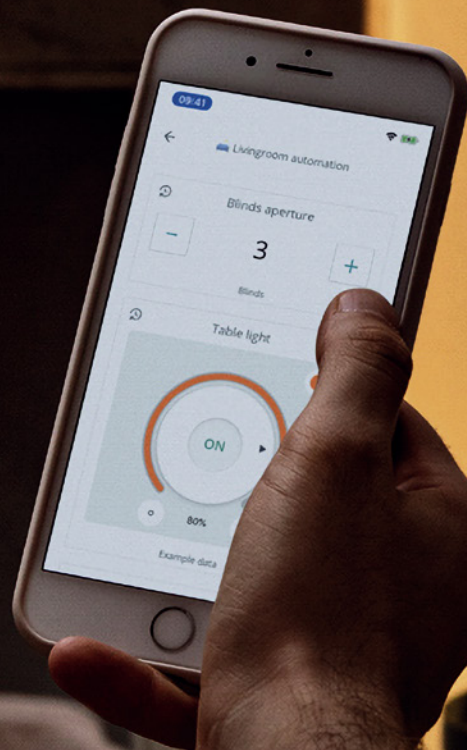
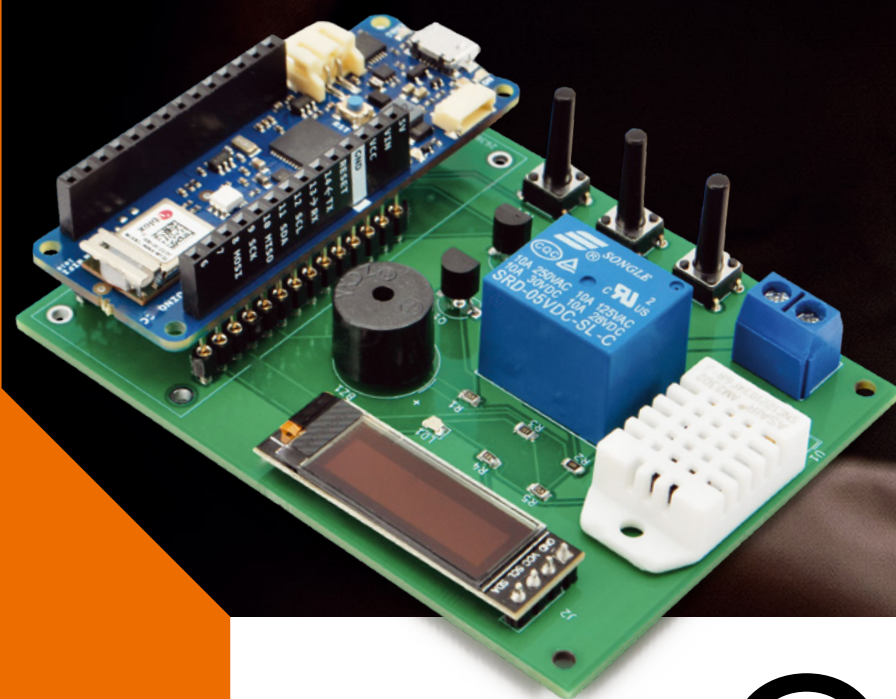
€ 5,00

Cod. HX711BOARD

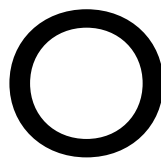
Prezzi IVA inclusa.

TERMOSTATO AMBIENTE SUL CLOUD

di TIZIANO ROSSI



Utilizziamo un sensore DHT22 per controllare la temperatura ambiente e l'umidità relativa e inviamo i dati ad Arduino IoT Cloud.



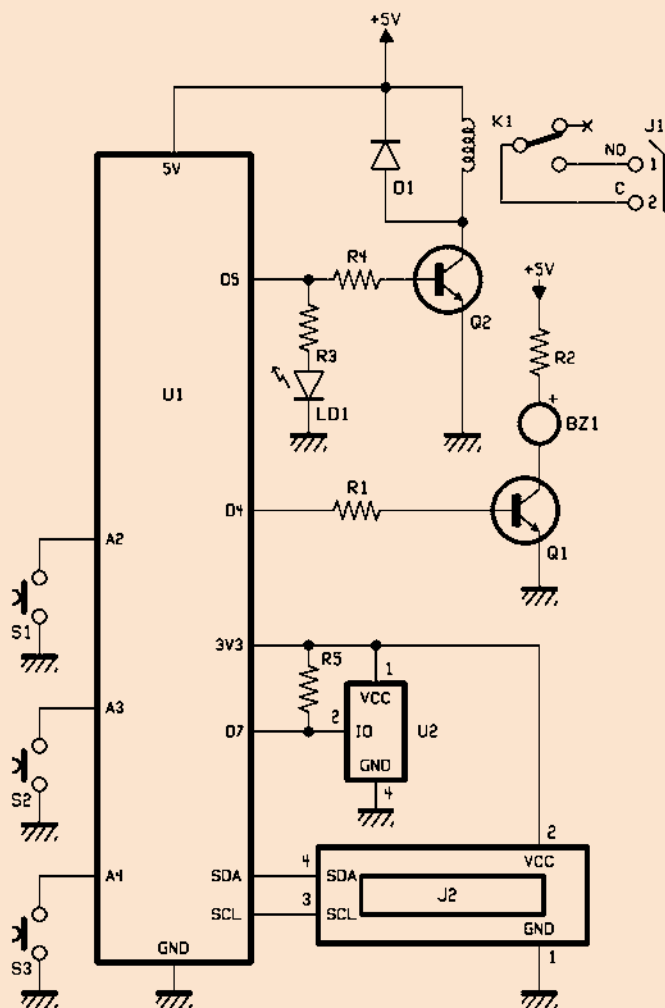
Oggigiorno sempre più attenzione è rivolta al risparmio energetico. Qualsiasi trasmissione, sia essa televisiva, radiofonica o altro, consiglia di ridurre e/o mantenere una temperatura media negli ambienti in cui viviamo non oltre i 19°C ma, come spesso accade, non abbiamo nessun mezzo che possa

darci questo tipo d'informazione.

Non esiste un unico termostato per ambienti: sono disponibili molti modelli e le loro caratteristiche variano molto: cablato o senza fili, elettronico o no, collegato, ecc.

Per ogni esigenza, ogni casa e ogni budget, c'è un termostato digitale adeguato.

A meno che, per ridurre al minimo il costo, non acquistiate un semplice termostato con manopola graduata (senza nessuna



informazione se non la temperatura impostata), un termostato con più funzioni e un display può costare anche oltre i 200€.

Partendo da queste considerazioni abbiamo realizzato un termostato che potesse abbinare alcune caratteristiche dei vari termostati in commercio con le seguenti peculiarità:

- costo contenuto;
- visualizzazione della temperatura ambiente;
- visualizzazione umidità ambiente;
- visualizzazione della temperatura impostata per termoregolazione;
- visualizzazione delta impostato per termoregolazione.

La possibilità di potere controllare la temperatura e il delta di termoregolazione avviene tramite il sensore DHT22 dandoci, tramite un contatto normalmente aperto di un relè, la possibilità di controllare l'apertura o la chiusura dell'elettrovalvola che controlla la circolazione dell'acqua calda nei radiato-

ri o nella serpentina di riscaldamento a pavimento (d'ora in poi denominati "circuiti di riscaldamento"), per cui abbiamo sviluppato un dispositivo che, chiudendo il contatto del relè, fosse atto a comandare il circuito di riscaldamento.

Oltre all'informazione della temperatura ambiente il sensore DHT22 ci permette di avere anche l'informazione dell'umidità relativa.

Come premesso nel titolo, l'impostazione e la lettura della temperatura può essere fatta localmente o tramite Arduino IoT Cloud.

LO SCHEMA ELETTRICO

Il cuore del termostato riportato nello schema elettrico in queste pagine è rappresentato dall'MKR WIFI 1010 di Arduino (vedi **Fig. 1**).

Sfruttando i due connettori strip maschio incorporati nell'MKR WIFI 1010, viene montato sul dispositivo tramite due connettori strip femmina, questo ci permette di poterlo rimuovere con facilità dal dispositivo per ogni evenienza.

Alimentato tramite la porta micro USB incorporata, permette a sua volta, tramite l'uscita 5V, di alimentare il relè e il buzzer; mentre il sensore di temperatura e umidità DHT22 e il display LCD vengono alimentati tramite l'uscita a 3,3V.

Qualora vogliate mantenere in funzione il termostato in mancanza di alimentazione al micro USB dell'MKR WIFI 1010, vi consigliamo di aggiungere una batteria ai polimeri di litio 3,7V 1000mA con connettore 2mm, che potete acquistare sul sito www.futurashop.it con il codice PLIB1000-2MM. Lo schema elettrico è molto semplice e intuitivo, ma un minimo di informazione va data.

Fermo restando che l'MKR WIFI 1010 (U1), come detto in precedenza, è il cuore del nostro progetto, nel paragrafo che tratterà lo sketch vedremo come il MKR WIFI 1010 controlla i vari componenti ad esso collegati.

Prima di partire con una panoramica sulla funzionalità dei vari componenti facciamo una piccola deviazione su cosa, per noi, significa **delta** e cosa, pur essendo un termine molto specifico, si intende quando indichiamo la **temperatura**:

- **delta**: ci permette di controllare il range di termoregolazione che va da un valore minimo 0,1°C ad un massimo di 1°C, per cui se impostiamo un delta di 0,5°C e una temperatura ambiente di 19°C al raggiungimento della stessa il contatto normalmente aperto si chiuderà per permettere al circuito di riscaldamento di riscaldare l'ambiente. Raggiunti i 19,5°C di temperatura (19°C impostati + il delta di 0,5°C) il contatto si riaprirà,

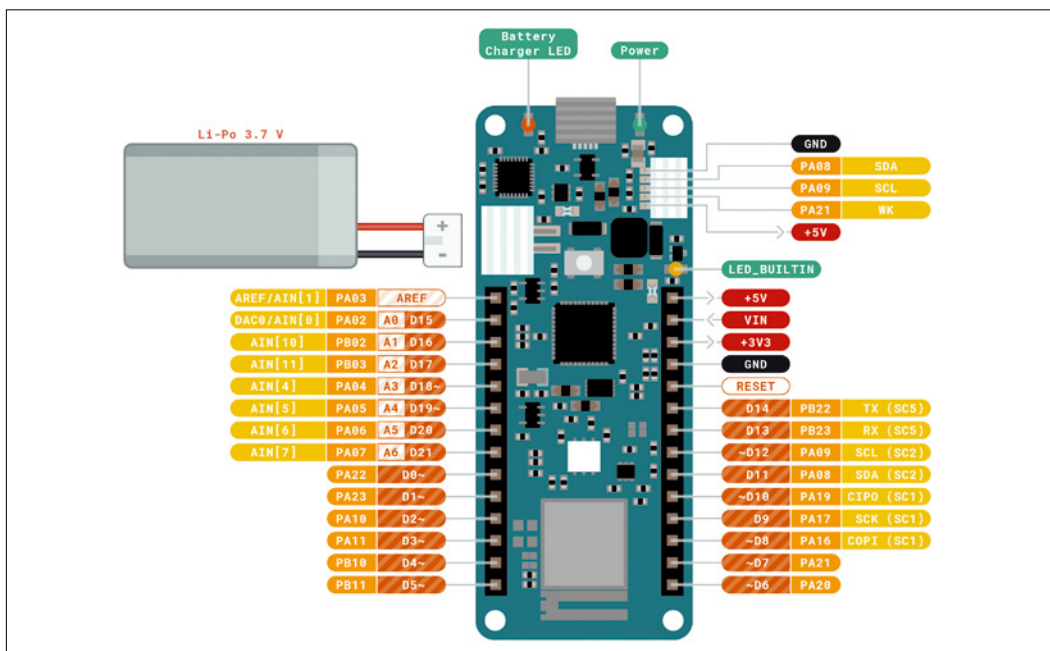


Fig. 1
Pin out
MKR WIFI 1010.

chiudendo il circuito di riscaldamento.

Alla prima accensione del termostato il valore di default del delta è di 0,3°C.

- **temperatura:** in questo caso abbiamo la temperatura impostata e la temperatura letta. Per quanto riguarda la temperatura impostata abbiamo un range che va da un minimo di 5°C ad un massimo di 30°C. Per quanto riguarda la temperatura letta si tratta, ovviamente, della temperatura reale ambiente in cui è installato il termostato. Alla prima accensione del termostato la temperatura impostata è di 20°C.

Concentriamoci ora sulle funzionalità dei vari componenti che compongono lo schema elettrico. Il display LCD viene controllato dall'MKR WIFI 1010 tramite la porta I2C e ha una triplice funzionalità.

1. visualizzare la temperatura e umidità ambiente in tempo reale;
2. visualizzare i valori di delta e temperatura impostati per la termoregolazione;
3. visualizzare l'incremento o il decremento dell'impostazione dei valori di delta e temperatura.

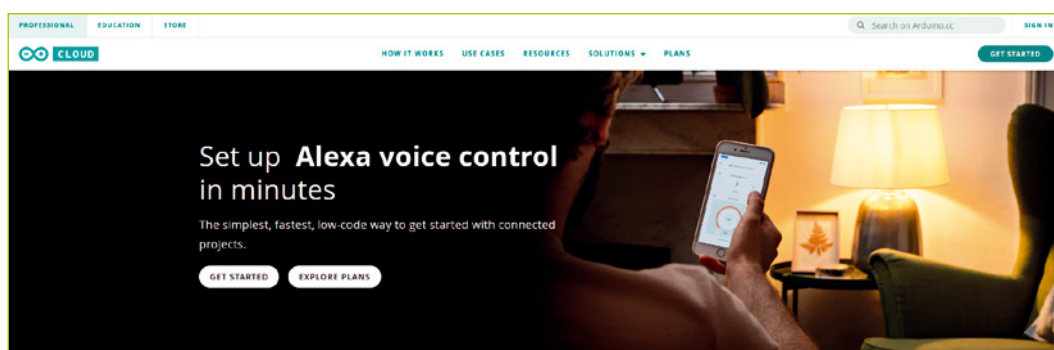
Per allungarne la vita abbiamo deciso che dopo due minuti circa venga spento. Vedremo in seguito come avviene questo controllo e come eventualmente possiamo variarlo. Fermo restando che principalmente sul display LCD viene visualizzata la temperatura e l'umidità

attuale, i tre mini-tasti S1, S2 e S3 ci permettono di:

1. S1: mantenendolo premuto potremo visualizzare sul display LCD i valori che abbiamo impostato per la termoregolazione di delta e temperatura.
2. S2: ha una doppia funzionalità. Una volta premuto comparirà sul display la dicitura "Temp-". Rilasciando il mini-tasto vedremo automaticamente decrementare il valore di temperatura impostato a step di 0,1°C alla volta. Una volta raggiunta la temperatura di nostro interesse, ripremendo il mini-tasto usciremo dalla funzione di decremento della temperatura impostata. Sul display LCD comparirà la scritta "EXIT". Rilasciando il mini-tasto S2 ritorneremo alla visualizzazione di temperatura e umidità attuale. Se invece manteniamo premuto il mini-tasto S2 per più di due secondi, dopo la scritta "Temp-" comparirà la scritta "Delta-". Rilasciando il mini-tasto S2 vedremo decrementare il valore di DELTA a step di 0,1°C alla volta. Raggiunto il valore desiderato per uscire dalla funzione "Delta-" basta premere il mini-tasto S2. Sul display LCD comparirà la scritta "EXIT" e, come prima, rilasciandolo verrà visualizzato la temperatura e umidità attuale.
3. S3: è esattamente uguale a S2, ma con la differenza che incrementa i valori da impostare di temperatura e/o delta anziché decrementarli, come visto in precedenza per S2.

Il buzzer BZ1 viene attivato tramite Q1, producendo un allarme sonoro di sicurezza che serve ad avvisarci che la temperatura rilevata dal sensore è

→ **Fig. 2**
Registrazione al
Cloud Arduino



superiore ai 30°C. Potrebbe significare che il contatto del relè rimane sempre attivo (non termoregola) o il sensore DHT22 non funziona, contemporaneamente sul display LCD compare una scritta di "Allarme!!" ad intervalli di due secondi.

Uscirà dal loop di allarme una volta raggiunta una temperatura rilevata dal sensore minore di 25°C. Come potremo vedere più avanti, le temperature di allarme possono essere cambiate a nostro piacimento.

Non rimane altro che concentrarci sul relè K1 che riporta sul connettore J1 il suo contatto normalmente aperto.

Controllato da Q2 quando eccitato viene chiuso il contatto del relè, permettendo in questo modo di potere comandare il circuito di riscaldamento.

Lo sketch di TERMOSTATO_FUTURA è sviluppato tramite L'IDE di Arduino. Sfruttando la libreria denominata DHT.h e DHT_U.h ci permette di controllare il sensore di temperatura e umidità DHT22 senza doverci scervellare su come ottenere ciò che ci serve. Il range di utilizzo del sensore va da -40 a +80°C; mentre per quanto riguarda l'umidità relativa andiamo da 0 a 100%.

La risoluzione è di 0,1 sia per la temperatura che per l'umidità.

Sapendo che il termostato viene utilizzato nel periodo autunno/inverno, abbiamo implementato uno switch, controllabile solo da Arduino IoT Cloud tramite la APP del vostro telefono mobile o trami-

te il sito WEB, che, in stato ON (inverno), permette di sfruttare le funzionalità di termoregolazione della temperatura ambiente; mentre in stato OFF (estate) il termostato funziona solo come lettore di temperatura e umidità, permettendo, in caso di utilizzo di un eventuale climatizzatore, di informarci sulle condizioni ambientali.

Utilizzando il LED RGB dell'MKR WIFI 1010 potremo avere l'informazione se lo switch è ON (verde) o OFF (spento) e di colore rosso in caso di attivazione del circuito di riscaldamento.

ARDUINO IOT CLOUD

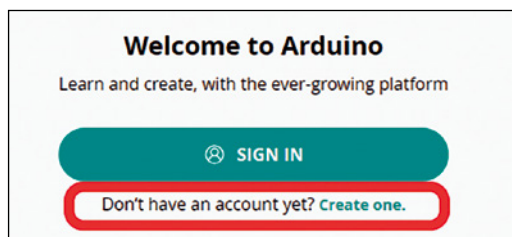
La scelta di utilizzare Arduino IoT Cloud è dettata dalla facilità nell'interfacciamento con board Arduino, dalla sobrietà del Cloud in oggetto e, soprattutto, dal costo limitato in caso si voglia passare da un piano di utilizzo "FREE" (nostro caso) a un piano superiore (esempio ENTRY).

La differenza tra i diversi piani di utilizzo la potete ricavare dal sito WEB <https://cloud.arduino.cc/plans>. Ovviamente il piano "FREE", come tutti i piani gratuiti, ha delle limitazioni.

Il punto fondamentale a cui prestare attenzione è che anche se si possono registrare più di due dispositivi, è possibile assegnare un massimo di cinque variabili ciascuno per non più di due dispositivi. Ovviamente, utilizzare board Arduino, come vedremo più avanti, risulta essere la condizione più semplice per essere velocemente on-line e, soprattutto, non è necessario aggiungere nessun simbolo di identificazione per connetterci con Arduino IoT Cloud se non, ovviamente, l'SSID e PASSWORD del vostro router.

Ma andiamo con ordine: la prima cosa da fare è registrarsi per potere ottenere un vostro account andando alla pagina WEB: <https://cloud.arduino.cc/>. Vi troverete la pagina come da **Fig. 2** in cui potete, utilizzando il menu di navigazione, esplorare le funzionalità del Cloud di Arduino.

→ **Fig. 3**
Come creare
un account.



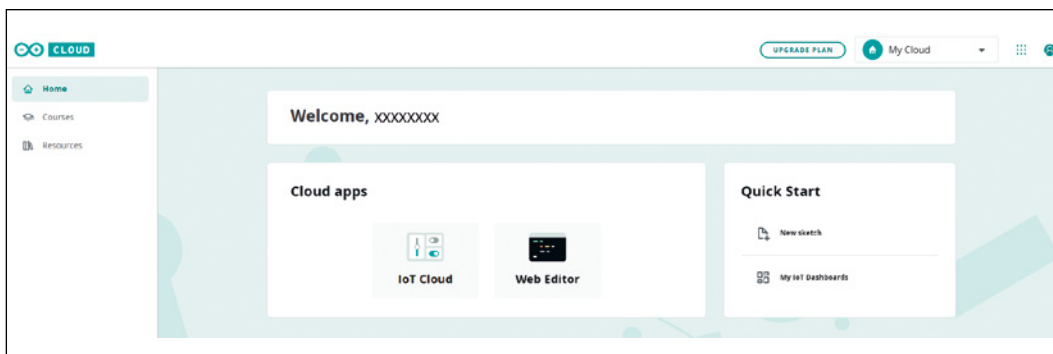


Fig. 4
Welcome
Arduino
Cloud.

Se, come accennato all'inizio, cliccate sulla voce "EXPLORE PLANS" potrete vedere quali sono, e cosa comprendono, i quattro piani di utilizzo. Innanzitutto, come ogni cloud, bisogna procedere alla registrazione che, nel nostro caso, avviene cliccando su "GET STARTED".

Vi verrà visualizzato quanto contenuto in **Fig. 3**, cliccate su "Don't have an account yet? Create one" e proseguite compilando quanto vi viene richiesto. Se avete fatto tutto correttamente riceverete una mail di WELCOME TO ARDUINO dove vi chiede di confermare il vostro indirizzo mail.

Finalmente abbiamo un account per, mattoncino dopo mattoncino, costruire il vostro Arduino IoT Cloud, dove potrete visualizzare e comandare l'impostazione di temperatura, umidità e, tramite il widget "switch", passare dalla condizione ON (Inverno) o OFF (Estate).

Entrate nell'account cliccando su "GET STARTED" (vedi **Fig. 2**), oppure, nel menù in alto a destra, cliccate su "SIGN IN".

Una volta entrati vi comparirà un popup in cui vi chiede di partire nel realizzare un Arduino IoT Cloud. Il nostro consiglio è di saltarlo cliccando su "DO IT LATER" e vi troverete nel vostro account (vedi **Fig. 4**), dove compare la scritta WELCOME, XXXXXXXX (le X non sono altro che il vostro Username). Collegate il vostro MKR WIFI 1010 a una porta micro USB del vostro PC e, nella pagina del vostro account, cliccate sull'icona "IOT CLOUD", vi comparirà una pagina con un menù dove potete fare le seguenti scelte:

- Things
- Dashboard
- Devices
- Integrations
- Templates

La prima cosa da fare è registrare un dispositivo, per cui cliccate su "Devices", "ADD DEVICE" e vi compari-

rà un'icona dove potete scegliere un Arduino Device o un Third party Device (vedi lista dispositivi supportati al seguente indirizzo WEB): Getting Started With the Arduino IoT Cloud | Arduino Documentation alla sezione "Compatible Hardware".

Cliccate su Arduino Device: se il vostro MKR WIFI 1010 è collegato correttamente al vostro PC vi comparirà il messaggio come da **Fig. 5**.

Continuate cliccando su "CONFIGURE" e date un nome al vostro dispositivo (nel nostro caso lo abbiamo nominato TERMOSTATO_FUTURA).

Al termine del settaggio del dispositivo vi troverete una pagina come da **Fig. 6**, il che vuol dire che il settaggio si è concluso positivamente.

Possiamo ora determinare quali sono le variabili (massimo cinque) che comporranno il nostro sketch. Dal menu principale (vedi **Fig. 6**) clicchiamo su *Things*, si aprirà una pagina nella quale potete, tramite l'icona "ADD", dare un nome alle cinque variabili di nostro interesse e determinare se è una variabile floating, integer, boolean ecc...

La **Fig. 7** riporta la pagina compilata con le cinque variabili e a che dispositivo sono assegnate.

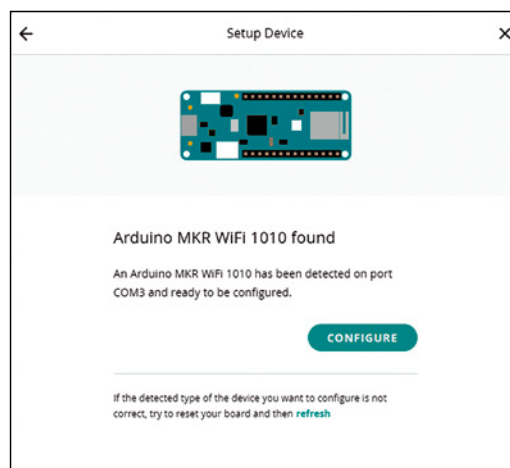
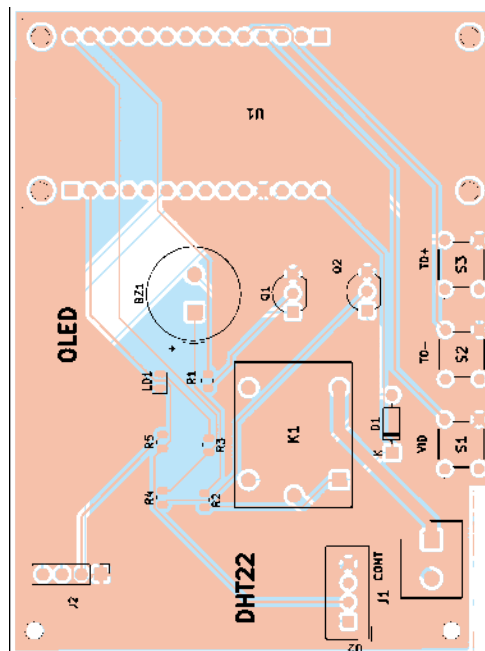
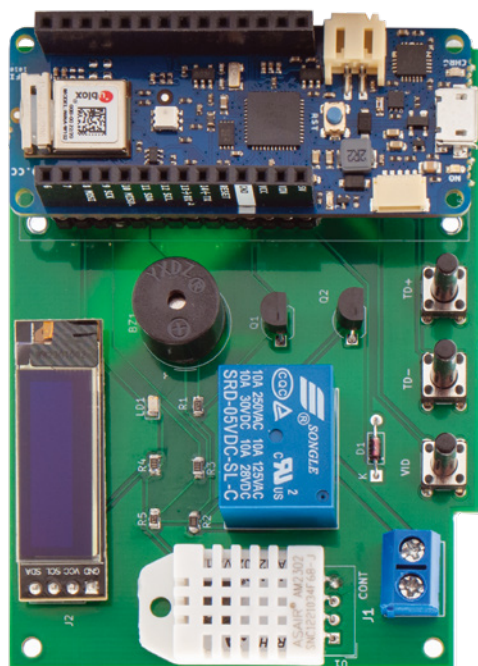


Fig. 5
Setup Device.



Elenco Componenti:

R1, R4 4,7 kohm
 R2 10 ohm
 R3 330 ohm
 R5 10k ohm
 D1 1N4148
 BZ1 Buzzer 3V con elettronica
 S1, S2, S3 Microswitch (h 13mm)
 Q1, Q2 BC547
 K1 Relè 5V - 250VAC/10A

U1 Arduino MKR1010
 U2 Sensore DHT22
 LD1 LED rosso (0805)
 J2 Display OLED I2C

Varie:

- Morsetto 2 poli 5,08 mm
- Strip femmina 14 vie passo 2,54 mm (2 pz.)
- Circuito stampato S1706 (86x65 mm)

Se avete commesso qualche errore nella compilazione delle variabili non vi preoccupate in quanto puntando con il mouse sui tre punti potete scegliere di modificarla.

Non rimane altro che assegnare l'SSID e la PASSWORD, che utilizzate per entrare in rete tramite il vostro router, nella sezione "Network".

Arduino IoT Cloud ha automaticamente creato uno sketch da potere scaricare o testare direttamente nel Cloud.

Per fare ciò clicchiamo sull'icona in alto a destra (vedi **Fig. 7** cerchiato in rosso), scegliendo "WEB EDITOR" vi verrà aperta una pagina (vedi **Fig. 8**) con, nella spalla sinistra, la lista degli sketch creati (nel nostro caso, cerchiato in rosso, TERMOSTATO_FUTURA con la data di creazione); mentre al centro vi troverete lo sketch con quattro folder

denominati:

- TERMOSTATO_FUTURA_XXXXXX
- README
- THINGPROPERTIES
- SECRET

Cerchiato in verde sono riportate le stesse funzioni che potete trovare nell'IDE di Arduino, per cui, se volete, potete testare lo sketch direttamente nella pagina WEB.

La velocità di verifica e upload dello sketch dipendono dalla velocità della vostra rete internet, per cui il nostro consiglio è di fare il download, cliccando col tasto destro del mouse sul nome dello sketch TERMOSTATO_FUTURA_XXXXX.

Nella cartella compressa che avete scaricato tro-

verete esattamente ciò che vedete nella pagina di WEB EDITOR. Salvate la cartella che è stata creata nella directory di vostro interesse; rinominatela senza la data di creazione in modo da avere una cartella con solo: TERMOSTATO_FUTURA e un file chiamato TERMOSTATO_FUTURA.ino.

Tramite l'IDE di Arduino aprite lo sketch, vi troverete lo stesso sketch visualizzato nella pagina di WEB EDITOR.

Ovviamente lo sketch TERMOSTATO_FUTURA.ino è "spoglio", contenendo solamente le variabili create nel WEB Cloud di Arduino, per cui, e lo vedremo in seguito, dovrà essere completato in modo da trasmettere e ricevere i dati di nostro interesse all'Arduino IoT Cloud.

Ultima cosa che ci rimane da fare è completare la dashboard con i widget che verranno abbinati alle variabili create nella pagina "Things", per cui dal menu di **Fig. 7** dobbiamo selezionare "Dashboards". Selezionando "BUILD DASHBOARD" daremo un nome che, in questo caso, dovrebbe essere il nome della stanza dove posizioneremo il termostato (nel nostro caso il -salotto-).

Cliccando su "ADD" si aprirà una finestra con tutti i widget che possiamo aggiungere.

Scelto il widget (per temperatura e umidità utilizziamo un Gauge, per Sel. Temp e Delta abbiniamo due Slider e lo switch per ON/OFF) dobbiamo fare

alcune cose fondamentali nella finestra "Widget Settings":

- dare il nome al widget;
- associarlo ad una variabile utilizzando "Linked Variable". Si aprirà una finestra con la lista di tutte le variabili create.

Se si tratta del widget di visualizzazione temperatura scegliete la variabile "temp" e automaticamente il widget verrà abbinato alla visualizzazione della temperatura ambiente;

- determinate il range di visualizzazione del dato nella sezione "Value range". Trattandosi di una temperatura ambiente abbiamo selezionato da 0 a 50°C, ricordando che il range del DHT22 va da -40 a +80°C. Utilizzandolo per una termoregolazione ambientale si presume che non si vada sottozero, non oltre i 25°C d'Inverno e, sperabilmente, i 40°C in Estate.

Se fatto tutto correttamente vi dovreste trovare nelle condizioni riportate in **Fig. 9**.

Per ogni dubbio vi rimandiamo ai vari tutorial o manuali che potete trovare cliccando su "Resources" posto sulla spalla sinistra della pagina iniziale (vedi **Fig. 4**).

Ultimo step da fare è scaricare dal telefono mobile la APP "Arduino IoT Cloud Remote" (vedi **Fig. 10**) e registrarsi utilizzando lo stesso username e pas-

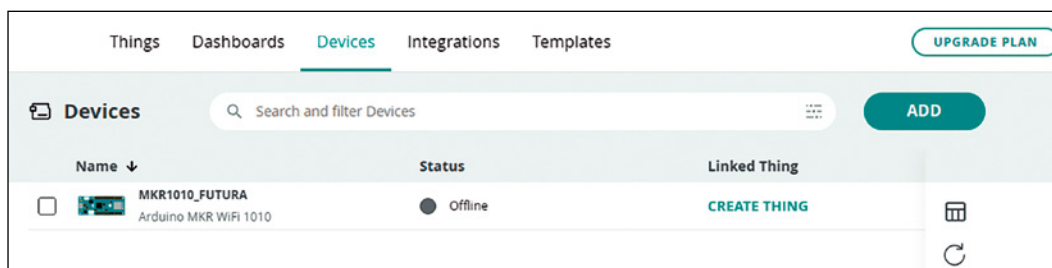


Fig. 6
Devices controllati in Arduino Cloud

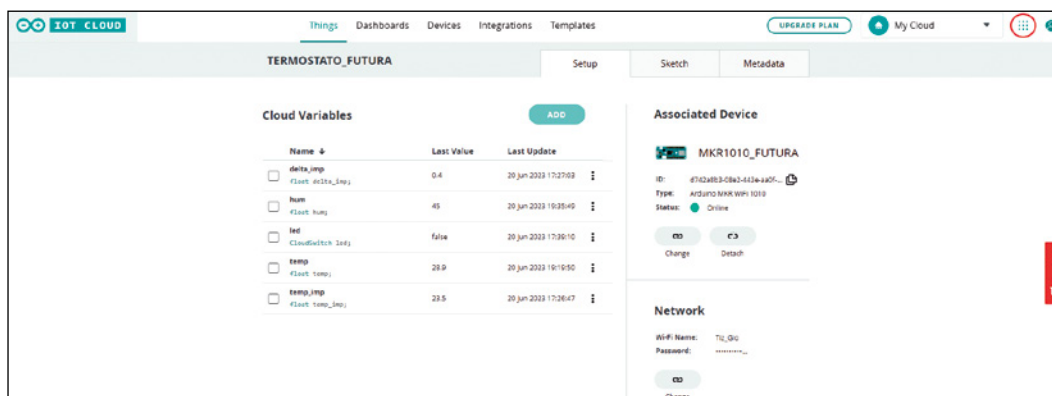


Fig. 7
Variabili associate al Device.



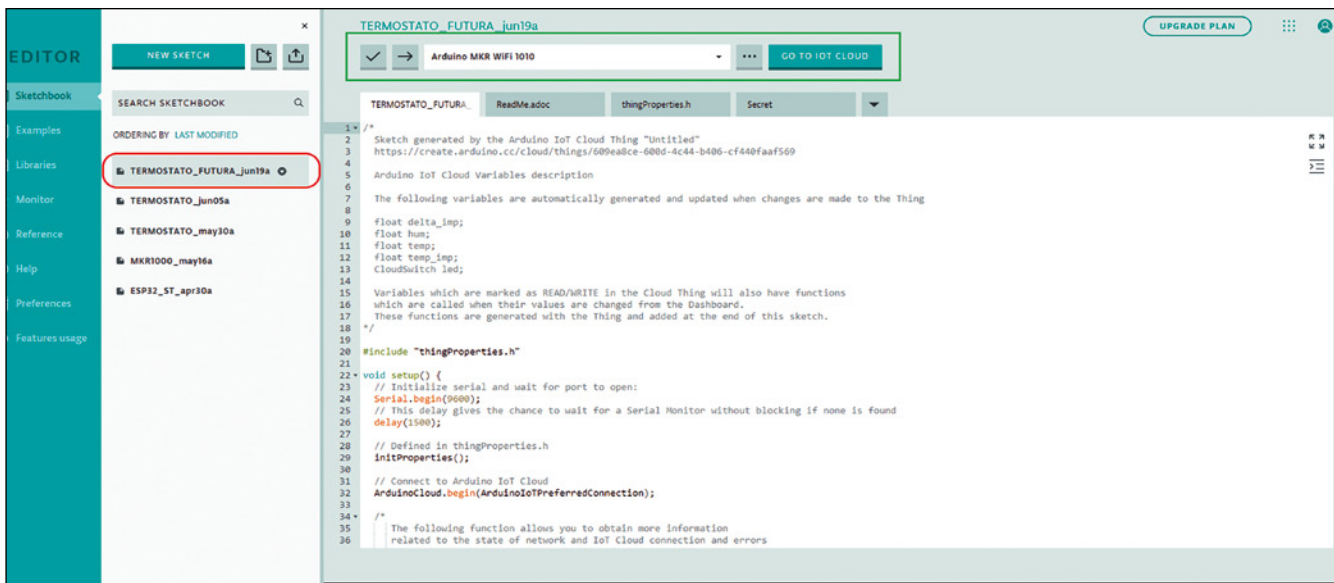


Fig. 8
Web Editor.

sword utilizzati per creare il vostro account.

Una volta entrati vedrete una icona con il nome dato nella pagina della dashboard (in questo caso salotto), cliccandoci sopra visualizzerete gli stessi widget (vedi **Fig. 9**).

Non meravigliatevi se ancora non siete on-line con il vostro Cloud. Dovete portare pazienza finché lo sketch THERMOSTATO_FUTURA.ino viene completato e caricato sul vostro MKR WIFI 1010.

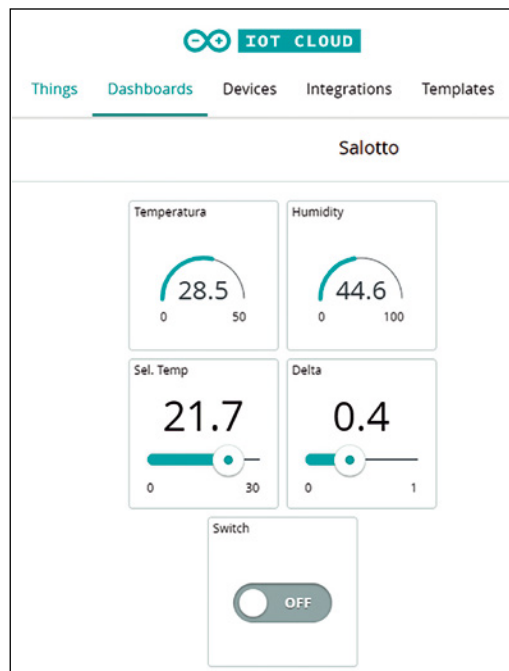


Fig. 9
Dashboard.

IL CODICE

Prima di entrare nel cuore dello sketch vediamo come impostare l'interrupt che ci permette di spegnere il display dopo circa due minuti. Questa parte è necessaria per prolungare la vita del Display LCD e, inoltre, se colleghiamo il PCB ad una batteria ricaricabile, ci permette di mantenere acceso il nostro dispositivo in caso di venuta a meno della tensione di alimentazione.

La parte del codice cerchiata in rosso (vedi **Fig. 11**) serve a settare il clock di conteggio che, partendo dal clock con frequenza 8MHz, viene diviso per otto per ottenere una frequenza da 1MHz. Oltre a settare il clock dell'MKR WIFI 1010 abilitiamo il timer da utilizzare (nel nostro caso è il T4).

Considerando il prescaler di 1024 (vedi **Fig. 11** cerchiato in verde), il numero dei conteggi di overflow cerchiati in verde (`REG_TC4_COUNT8_PER = 0xFF`;) e il numero della variabile `cnt` (1000) riportata nella routine di overflow chiamata **T4_Handler** (vedi **Fig. 14** cerchiato in azzurro), il display rimane acceso per circa due minuti.

Le altre variabili introdotte nel SETUP fanno parte della normale programmazione per ottenere le funzioni (che vedremo in seguito), per cui facili da comprendere; inoltre, alcune funzioni sono dettate dai componenti utilizzati come:

- display LCD;
- DHT22;
- Cloud di Arduino.

Per iniziare dovete aggiungere alla cartella THERMOSTATO_FUTURA il file header chiamato **disp.h**

Fig. 13
Variabili da inviare
all'Arduino Cloud.

```
#include <ArduinoIoTCloud.h>
#include <Arduino_ConnectionHandler.h>

const char SSID[] = SECRET_SSID; // Network SSID (name)
const char PASS[] = SECRET_OPTIONAL_PASS; // Network password (use for WPA, or use as key for WEP)

void onDeltaImpChange();
void onHumChange();
void onTempChange();
void onTempImpChange();
void onLedChange();

float delta_imp;
float hum;
float temp;
float temp_imp;
CloudSwitch led;

void initProperties(){
  ArduinoCloud.addProperty(delta_imp, READWRITE, ON_CHANGE, onDeltaImpChange);
  ArduinoCloud.addProperty(hum, READ, ON_CHANGE, onHumChange);
  ArduinoCloud.addProperty(temp, READ, ON_CHANGE, onTempChange);
  ArduinoCloud.addProperty(temp_imp, READWRITE, ON_CHANGE, onTempImpChange);
  ArduinoCloud.addProperty(led, READWRITE, ON_CHANGE, onLedChange);
}

WiFiConnectionHandler ArduinoIoTPreferredConnection(SSID, PASS);
```

Fig. 14
Interrupt
T4_Handler.

```
void TC4_Handler() // Interrupt Service Routine (ISR) for timer TC4
{
  cnt++;
  if (cnt > 1000){
    display.clearDisplay();
    display.display();
    NVIC_DisableIRQ(TC4_IRQn); // Connect TC4 to Nested Vector Interrupt Controller (NVIC)
  }else{
    REG_TC4_INTFLAG = TC_INTFLAG_OVF; // Clear the OVF interrupt flag
  }
}
```

cerchiata in rosso di **Fig. 13**). Per quanto riguarda lo sketch TERMOSTATO_FUTURA.ino riportiamo solo delle sezioni perché troppo lungo per essere riportato in queste pagine, per cui, partendo dalla variabile **count**, vediamo come si sviluppano i vari blocchi. Trattandosi di una funzione che si ripete nel tempo (vedi **Fig. 15** cerchiata in rosso) per il controllo

dei tre mini-tasti, abbiamo pensato di creare il file header **disp.h** in modo da rendere più fruibile e semplice alla lettura il file TERMOSTATO_FUTURA.ino. Nel file header vengono controllati i due mini-tasti S2 e S3 per l'incremento o il decremento della temperatura e del delta. La funzione **vdt** contenuta nel file header viene


```

//Inizio settaggio temperatura e/ Delta Positivo
if(digitalRead(up) == 0){
    REG_TC4_INTFLAG = TC_INTFLAG_OVF; // Clear the OVF interrupt flag
    cnt = 0;
    NVIC_EnableIRQ(TC4_IRQn); // Connect TC4 to Nested Vector Interrupt Controller (NVIC)

    count = 1;
    display.clearDisplay();
    display.setCursor(0, 0);
    display.write("Temp+");
    display.display();
    delay(2000);
    if(digitalRead(up) == 0){
        count = 2;
        display.clearDisplay();
        display.setCursor(0, 0);
        display.write("Delta+");
        display.display();
        delay(2000);
    }
    display_i2c();
}

//*****

```

Fig. 15
Routine
controllo
mini-tasti.

chiamata ogni volta che si preme il mini-tasto S1 che, oltre ad attivare la visualizzazione dei dati di **temperatura** e **delta** impostati, riaccende il display LCD dopo lo spegnimento.

Come accennato nella costruzione della dashboard in Arduino IoT Cloud, abbiamo aggiunto uno switch che ci permette di selezionare la modalità Inverno (ON) ed Estate (OFF).

In modalità ON il termostato funziona come termoregolatore della temperatura ambiente. Nell'evenienza ci possa essere un problema nel controllo della temperatura (sensore o relè non funzionanti), abbiamo introdotto un limite di 30.0°C, superati i quali compare sul display la scritta "Allarme !!!" seguita da un suono, ad intervalli di due secondi, emesso dal buzzer.

Ovviamente se il TERMOSTATO_FUTURA è settato per l'estate l'allarme è disinserito in quanto, in questo caso, funziona solo come visualizzazione della

temperatura e umidità ambiente.

Trattandosi di uno sketch "aperto" è inutile dirvi che potete variare il valore limite di 30.0°C (visualizzati nello sketch con un valore di 300) a vostro piacimento, portandolo ad un valore a voi più consono. Ultimo sguardo va rivolto all'area che riguarda l'aggiornamento della temperatura e/o del delta impostati nell'Arduino IoT Cloud (vedi **Fig. 16** cerchiato in nero).

Ogni volta che da Cloud a termostato, o viceversa, si effettua un cambiamento dei valori di impostazione della temperatura e/o del delta si attivano le funzioni cerchiato in nero per aggiornare il dato sull'MKR WIFI 1010 e visibili sul display LCD.

È ora di caricare lo sketch: eseguite l'upload di TERMOSTATO_FUTURA.ino sul vostro MKR WIFI 1010 (ricordatevi di inserire SSID e Password nel folder header chiamato **arduino_secrets.h**) e dovrete, finalmente, essere online nell'Arduino IoT Cloud.



Un termostato ambiente è un dispositivo cablato o wireless che regola la temperatura della casa. La regolazione del riscaldamento fatta dal termostato della stanza fornisce un calore ideale e una temperatura che offre un grande comfort termico. Questa regolazione avviene tramite il sensore di temperatura permettendo di scaldare l'ambiente solo se necessario.

Perché è importante la posizione del termostato della stanza?

Impostare il termostato della tua stanza non è un atto banale. Infatti, la scelta della sua ubicazione deve essere attentamente pensata per soddisfare due obiettivi essenziali:

1. Ridurre il consumo di energia e non surriscaldare inutilmente
2. Ottimizzare e controllare il comfort termico in ogni stanza

È importante posizionare correttamente il termostato ambiente: non è consigliabile posizionarlo vicino a un radiatore o a una fonte

di calore. È meglio tenerlo il più lontano possibile per non distorcere la misurazione della temperatura nella stanza.

Non è affatto opportuno mettere un termostato ambiente in cucina quando altre stanze sono riscaldate. Infatti, in certi momenti della giornata la cucina non ha bisogno di essere riscaldata a differenza di una stanza come una camera da letto. L'uso di apparecchi elettrici come un forno elettronico o una lavastoviglie contribuisce ad aumentare il calore in modo artificiale, distorcendo la temperatura misurata dal termostato.

È quindi essenziale utilizzare un termostato per ogni stanza, o posizionare il termostato in una stanza "neutra", per esempio il soggiorno.

Evitare anche tutte le correnti d'aria posizionando il termostato sulla parete a 30-40 cm dalle aperture (porte o finestre). Le correnti d'aria generate dalle aperture negli accessi alla casa possono influenzare il termostato.

CONCLUSIONI

Fin dall'inizio di questo articolo vi abbiamo informato sulla possibilità di scegliere fra una

vasta gamma di termostati atti a svolgere quanto descritto, ma non esiste un termostato che possa svolgere le funzioni essenziali senza, lasciateci dire, costare un occhio della testa. Ovviamente il progetto è stato pensato per controllare la temperatura ambiente nel periodo di utilizzo della vostra caldaia: in poche parole, cambiamento climatico permettendo, da autunno inoltrato all'inizio della primavera. Avendo incorporato un display LCD, nel periodo di non utilizzo (periodo estivo) vi può essere utile per visualizzare la temperatura ambiente e l'umidità relativa, per cui vi permette eventualmente di controllare se il vostro climatizzatore funziona correttamente. Vi ricordiamo che Arduino IoT Cloud può supportare più di un device a cui, ad ognuno, possiamo abbinare cinque Things per un massimo di dieci Things, per cui potete, se volete, aggiungere un device (esempio..... un altro MKR WIFI 1010) da installare in una stanza diversa dal soggiorno. Un esempio potrebbe essere il bagno al quale va impostata, per ragioni logistiche, una temperatura più alta rispetto agli altri ambienti. ■

Fig. 16
Variazione
impostazione
Temperatura e
Delta per invio
all'Arduino Cloud.

```
/*
  Since Temp is READ_WRITE variable, onTempChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onTempChange() {
  // Add your code here to act upon Temp change
}

/*
  Since Hum is READ_WRITE variable, onHumChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onHumChange() {
  // Add your code here to act upon Hum change
}

/*
  Since VarTemp is READ_WRITE variable, onVarTempChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onTemplmpChange() {
  // Add your code here to act upon Templmp change
  temp_i = temp_imp * 10.0;
}

/*
  Since Delta is READ_WRITE variable, onDeltaChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onDeltalmpChange() {
  // Add your code here to act upon Delta change
  delta = delta_imp * 10.0;
}

/*
  Since Led is READ_WRITE variable, onLedChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onLedChange() {
  // Add your code here to act upon Led change
}
```



Cosa occorre?

La scheda Arduino MKR WIFI 1010 (cod. MKRWIFI1010) è disponibile a Euro 47,00, il display oled (cod. I2COLED091) a Euro 5,90, il sensore di temperatura e umidità (cod. DHT22) a Euro 9,00. Il circuito stampato può essere realizzato con il servizio PCB (cod. PCBPRODUCTION). I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

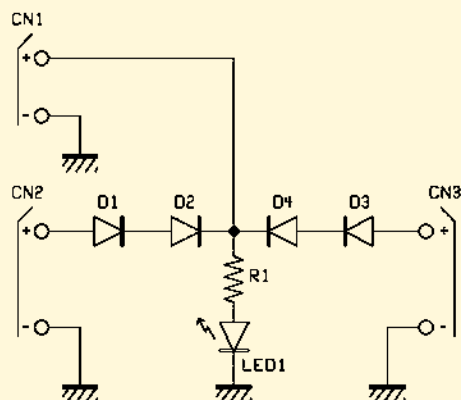
LUMINO A ENERGIA SOLARE

del Prof. MICHELE MENNITI



Costruiamo un lumino elettronico alimentato da pannelli solari e batterie ricaricabili per una soluzione eterna, sostenibile, eco-friendly e senza costi energetici.

Il giorno della ricorrenza dei defunti si rende omaggio ai propri cari che non ci sono più con una visita al cimitero, una preghiera, un fiore, un lumino che resterà acceso fino all'esaurimento dello stoppino immerso nella cera. Ma dopo qualche giorno, nella maggior parte dei casi, tutto tornerà come prima, perché si torna al tran tran quotidiano e raramente si riesce a trovare il tempo per una nuova visita, rimandando tutto all'anno successivo. Una luce accesa davanti ad un nome o una foto è simbolo di ricordo ma anche di speranza, allora tanti ricorrono ai lumini elettronici, alimentati a batterie, ma anche questi ad un certo punto smettono di funzionare e bisognerebbe poi trovare il tempo di andare a sostituire le batterie, cosa che difficilmente succede. Per queste ragioni abbiamo pensato di realizzare un prodotto a funzionamen-



to illimitato, ricorrendo agli ormai diffusissimi pannelli solari.

SCHEMA ELETTRICO

Il circuito è estremamente semplice, come si può notare osservando lo schema elettrico. Abbiamo adottato un pannello solare (cod. SOLPAN05W) di dimensioni abbastanza contenute, con tensione nominale di 5,5 V e potenza di 0,55 W, corrispondente ad una corrente massima erogabile di 100 mA, più che sufficiente per caricare un pacco di tre batterie ricaricabili, di tipi NiMH messe in serie, per una tensione nominale di circa 3,6-3,9V.

I due diodi D1 e D2 in serie svolgono il duplice compito di abbassare la tensione di carica a

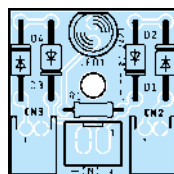
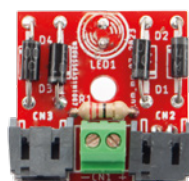
circa 4,2-4,3 V e di impedire il ritorno di corrente dalle batterie al pannello, in assenza di luce solare. In serie ai due diodi il resistore R1 limita la corrente di alimentazione del diodo LED (cod. LEDG5FIAMMA), un particolare tipo di LED che emette una luce gialla tremolante, che dà un bellissimo effetto "fiammella", emulando proprio la luce emessa da una candela di cera.

Quindi la presenza della luce solare permette di accendere il LED e contemporaneamente di ricaricare le batterie; quando la luce ambientale diventa insufficiente allora sono le batterie a continuare ad alimentare il LED, per tutta la notte e fino al mattino successivo, quando il pannello riprenderà a generare la necessaria tensione di funzionamento per il circuito.

Naturalmente il pannello deve essere posizionato in modo tale da captare la luce solare in modo ottimale e per il maggior tempo possibile, altrimenti la ricarica giornaliera potrebbe essere insufficiente a compensare il consumo notturno e nel tempo le batterie tenderebbero a scaricarsi, vanificando di fatto il sistema da noi pensato. Come vedremo, il contenitore che abbiamo disegnato espressamente per il nostro lumino prevede delle superfici idonee al fissaggio di uno o due pannelli solari, in quanto un secondo pannello potrebbe aiutare a risolvere eventuali difficoltà di posizionamento a causa del diverso orientamento della luce solare durante la giornata.

Il circuito opzionale aggiuntivo prevede semplicemente un ulteriore pannello con i due propri diodi in serie (D3 e D4), la cui uscita sarà collegata assieme a quella del primo pannello; in tal modo si potrà orientare il contenitore in modo che ci sia sempre almeno un pannello operativo nell'arco della giornata.

piano di MONTAGGIO



Elenco Componenti:

D1, D2, D3, D4: 1N4007
R1: 22 ohm
LED1 LED Fiamma 5 mm

Varie:

- Morsetto 2 vie passo 2,54 mm
- Connettore JST PH passo 2 mm (2 pz.)
- Circuito Stampato S1705 (22x22 mm)

REALIZZAZIONE DEL CIRCUITO

Abbiamo predisposto un piccolissimo PCB (appena 24x23 mm) che potrà essere realizzato mediante i file gerber, caricati nella nostra sezione download, e ricorrendo al servizio di produzione di circuiti stampati di Futura Elettronica (cod. PCBPRODUCTION).

Una volta in possesso del circuito stampato bisognerà saldare i due (o quattro) diodi raddrizzatori, il resistore e i connettori sul lato superiore del PCB, mentre il solo LED dovrà essere saldato sul lato inferiore, come richiesto dal montaggio nel contenitore (**Fig. 1**); occorre fare attenzione alla polarità dei diodi e del LED altrimenti il circuito non potrà funzionare.

COLLAUDO

I collegamenti tra il PCB e i componenti esterni sono piuttosto semplici: i pannelli solari sono già dotati di un cavetto con uno spinotto di tipo JST PH femmina con passo da 2 mm; seguendo la **Fig. 2** basterà innestarli nelle relative prese CN2 e CN3; le tre batterie (di tipo AA NiMH) dovranno essere inserite in un portabatterie idoneo (cod. 8220-BH331B o simili) i cui due fili dovranno essere collegati al connettore CN1, facendo ben attenzione a non invertire la polarità, basterà collegare il filo rosso al punto contrassegnato con "+" e quello nero al "-".

Se tutto è stato eseguito correttamente il LED si accenderà istantaneamente con il suo particolare effetto luminoso, anche effettuando il test al chiuso, in quanto saranno le batterie a fornire l'energia necessaria. Infatti misurando la tensione su CN1 sarà rilevato il valore del solo pacco batterie, che tenderà a diminuire lentamente durante il funzionamento. Spostandosi sotto la luce del sole la tensione inizierà ad aumentare, a dimostrazione che è in atto la ricarica delle batterie.



Fig. 1
Il LED sul lato inferiore.

Il nostro progetto prevede il posizionamento del lumino all'esterno, per captare la luce ambientale; ci sono casi in cui (p.es. le cappelle di famiglia) potrebbe essere necessario collocare il lumino all'interno, in questi casi occorrerà prolungare i fili dei pannelli solari in modo da portarli nei pressi di una finestra o meglio ancora all'esterno, altrimenti il circuito non potrà funzionare correttamente.

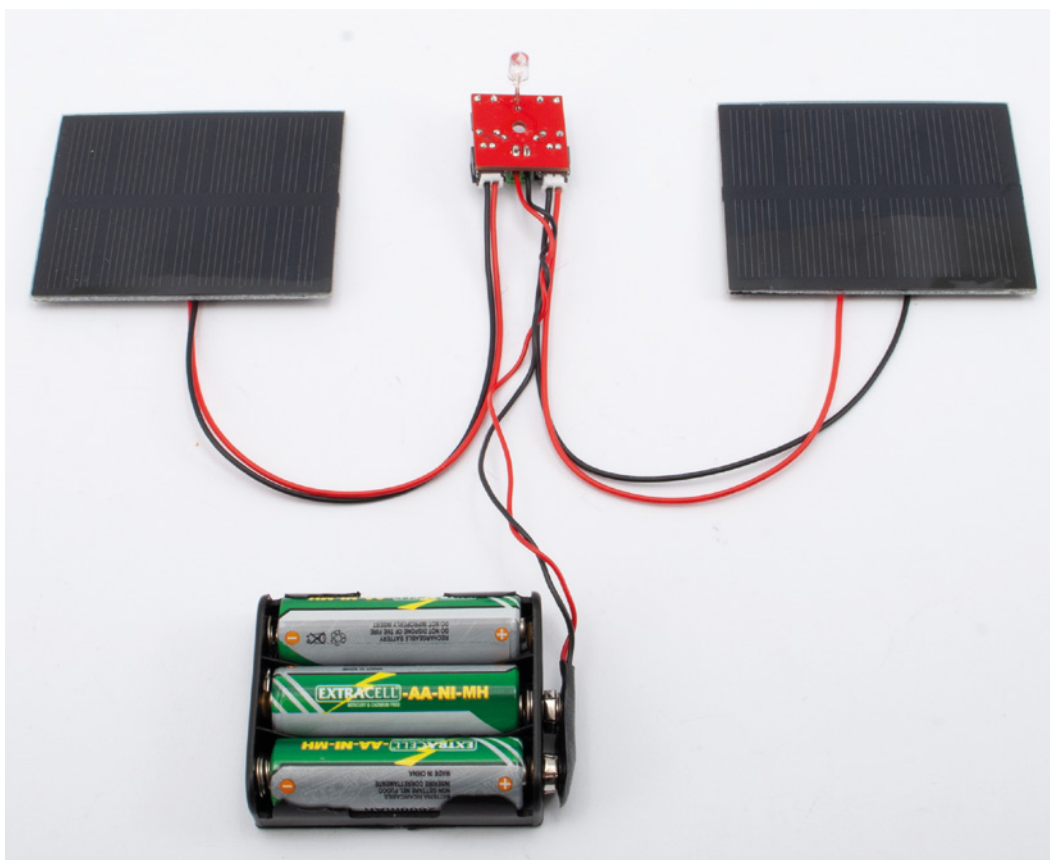


Fig. 2
I collegamenti dei pannelli e del pacco batterie.

Fig. 3
Il circuito montato
nel contenitore con
due pannelli.



CONTENITORE

Come accennato in precedenza, abbiamo disegnato uno specifico contenitore, di forma prismatica, che prevede la possibilità di attaccare uno o due pannelli solari sulle sue superfici esterne, mentre il LED potrà fuoriuscire dal foro della piccola base superiore, alla quale potrà essere fissato il PCB mediante una vite M3 con relativo dado (**Fig. 3**).

Le superfici laterali del contenitore 3D presentano due fori rettangolari per il passaggio degli spinotti JST, che saranno utilizzabili in qualsiasi modo si deciderà di posizionare i pannelli, per poter collegare gli spinotti alle prese del PCB.

Per quanto riguarda il LED, abbiamo predisposto uno specifico file STL per realizzare il tappo trasparente visibile nella foto, utilizzando del PLA neutro per la stampa.

Il pacco batterie invece troverà alloggio nell'am-

pio spazio interno del contenitore (**Fig. 4**), senza alcuna necessità di fissaggio ma, volendo, si potrà utilizzare del nastro adesivo di tipo strip in nylon. I file STL per la stampa in 3D sono scaricabili dalla sezione download, ricordando comunque che è disponibile il servizio di stampa 3D sul sito futuranet.it.

Il progettino da noi proposto è semplice, economico e facilmente realizzabile e una volta completato il montaggio ed il relativo collaudo sarà subito utilizzabile. L'unico accorgimento da adottare dovrebbe essere quello di controllare lo stato delle batterie dopo alcuni giorni di funzionamento, per verificare se il posizionamento dei pannelli sia idoneo alla loro completa ricarica giornaliera; se ciò non dovesse avvenire bisognerà modificare opportunamente l'orientamento dei pannelli, in caso contrario le batterie non avranno una lunga durata in quanto, ad un certo punto, la loro tensione scenderà sotto la soglia minima e si danneggeranno irrimediabilmente. Concludiamo dicendo che la particolare forma prismatica che abbiamo dato al contenitore si presta anche ad applicazioni diverse da quella descritta, per esempio nel periodo natalizio potrebbe essere un oggetto decorativo sul tavolo, in questo caso si potrebbe optare per l'uso di comuni batterie alcaline, senza montare il pannello solare, oppure posizionare, durante il giorno, la candela elettronica vicino una finestra, per far ricaricare le batterie. □

Fig. 4
L'interno del
contenitore.



Cosa occorre?

Il lumino elettronico può essere facilmente autocostruito utilizzando i pannelli solari (cod. SOLPAN05W) in vendita a Euro 4,50 l'uno, il led con effetto fiamma (cod. LEDG5FIAMMA) a Euro 0,80, il porta batterie (cod. BH331B) a Euro 0,95, le batterie AA ricaricabili (cod. THR6AA2100) a Euro 15,80.

Il PCB può essere realizzato utilizzando il servizio di produzione circuiti stampati (cod. PCBPRODUCTION).

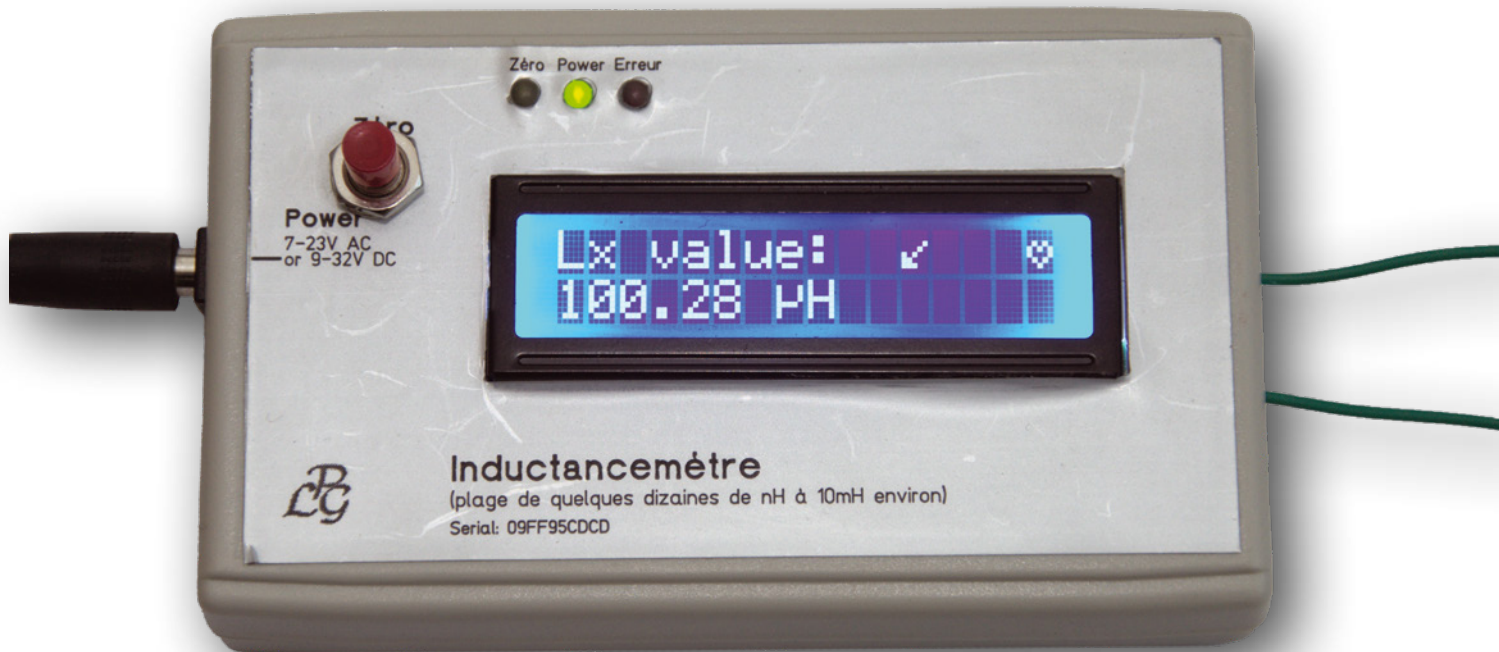
I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

UN MISURATORE D'INDUTTANZE

di PHILIPPE LE GUEN

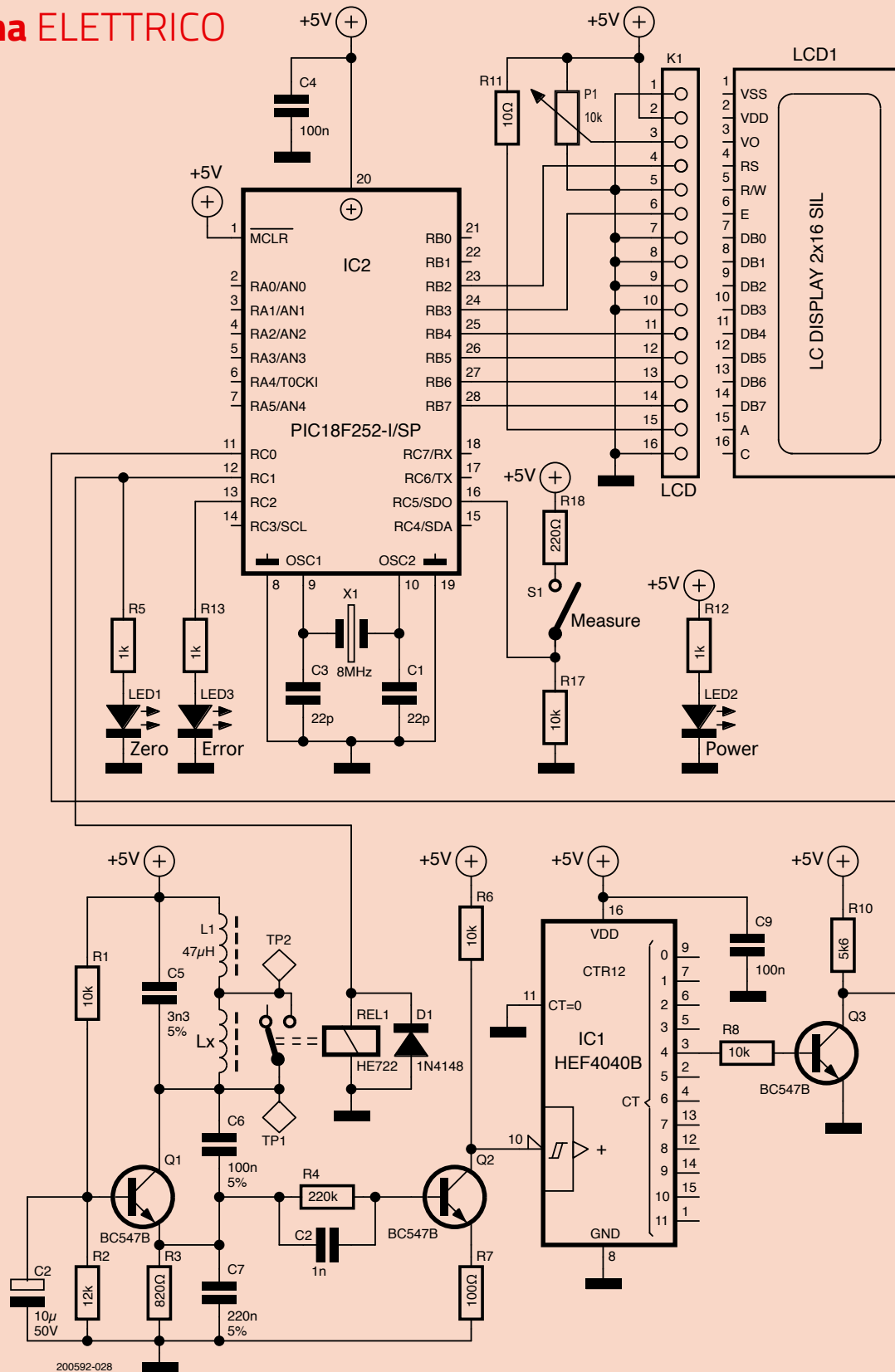


Hai bisogno di uno strumento per testare e misurare induttori? Questo strumento fai-da-te è un'economica alternativa rispetto alle costose soluzioni disponibili sul mercato.

Uno strumento affidabile per testare e misurare induttori è stato nella mia lista dei desideri per il laboratorio per molti anni, ma il costo di un dispositivo del genere è proibitivo. Ho quindi deciso di costruire qualcosa da solo. Il piccolo

strumento presentato qui non può competere con i complessi strumenti polifunzionali disponibili sul mercato; si tratta solo di un modesto misuratore di induttanze che mi consente di ottenere il valore di un'induttanza sconosciuta. Ci sono molti dispositivi di questo tipo disponibili online, a volte basati su Arduino.

schema ELETTRICO



Alcuni anni fa, ho scoperto il sito web di F. Kudelsko (http://kudelsko.free.fr/inductance_usb/sommaire.htm), che descrive un piccolo misuratore di induttanze fai-da-te in grado di misurare induttanze da poche decine di nH a circa 10 mH. Un piccolo programma per Windows recupera il valore dell'induttore da testare tramite USB e lo visualizza. Anche se ho trovato questo approccio interessante, stavo cercando un dispositivo autonomo. Tuttavia, desidero ringraziare l'autore per aver condiviso il suo lavoro, su cui ho basato il mio progetto.

PRINCIPIO DI FUNZIONAMENTO

Il principio di funzionamento del misuratore di induttanze è piuttosto semplice. L'induttanza sconosciuta viene utilizzata per creare un oscillatore LC. Misurando la frequenza dell'oscillatore, è possibile determinare il valore dell'induttanza. Esistono molti modi per costruire un oscillatore LC. Qui viene utilizzato un oscillatore di Colpitts (**Fig. 1**). Il circuito utilizza un amplificatore a transistor a base comune (Q1) con l'ingresso sull'emettitore e l'uscita sul collettore. Senza entrare nei dettagli su come funziona questo oscillatore, la sua frequenza di uscita è data dalla formula di Thomson (nota anche come Lord Kelvin):

$$f = 1 / (2\pi \sqrt{LC})$$

Dopo un adeguamento, otteniamo:

$$L = 1 / (4\pi^2 f^2 C)$$

Qui C è il valore dei due condensatori C6 e C7 in serie:

$$C = C6 \times C7 / (C6 + C7)$$

Conosciuti i valori di questi condensatori, che ho misurato con il mio misuratore di capacità per maggiore precisione nei calcoli, dobbiamo solo inserire il valore di C nella formula (insieme alla frequenza misurata) per trovare il valore di L. Si consiglia di utilizzare condensatori MKT al 5%.

SCHEMA ELETTRICO

Il circuito del misuratore di induttanza è illustrato nello schema elettrico. Ho sostituito il microcontrollore PIC18F2550 dello schema originale di Kudelsko con un PIC18F252, in quanto non avevo bisogno di un'interfaccia USB, e ho aggiunto un LCD alfanumerico 16x2. Il valore C5 è stato ereditato dal progetto origi-

nale e ha una leggera influenza sulla frequenza dell'oscillatore.

Con C5 da 3,3nF il valore di C da utilizzare nei calcoli è dato da:

$$C = C5 + C6 \times C7 / (C6 + C7)$$

C5 dovrebbe essere della stessa precisione e qualità di C6 e C7.

Con i valori dei miei componenti (e Lx cortocircuitato dal relè, vedi sotto), la frequenza operativa teorica dell'oscillatore è di 83,821 kHz (86,488 kHz con componenti ideali). La misurazione con un oscilloscopio ha mostrato una frequenza di 88,652 kHz, con una differenza del 5,4%. Tutti i calcoli sono eseguiti dal microcontrollore che misura la frequenza dell'oscillatore attraverso la sua porta RCO. A tal fine è essenziale dividere la frequenza in valori accettabili per l'MCU. Un contatore binario di tipo 4040 (IC1) si occupa di questo realizzando un rapporto di divisione di 1/32. Ho utilizzato un HEF4040B per IC1 perché era quello che avevo disponibile in magazzino, ma è possibile utilizzare anche un 74HCT4040.

L1 E IL RELÈ

Per garantire l'avvio rapido dell'oscillatore indipendentemente dal valore dell'induttanza incognita Lx, è stata posta un'altra induttanza (L1) in serie ad essa. Quando si effettua una misura, la frequenza dell'oscillatore è determinata da queste due induttanze. Volendo però conoscere solo il valore di Lx, il software esegue prima una misura con Lx in corto-

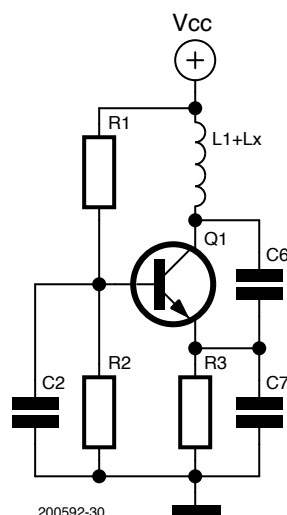


Fig. 1
Un circuito LC determina la frequenza dell'oscillatore di Colpitts.

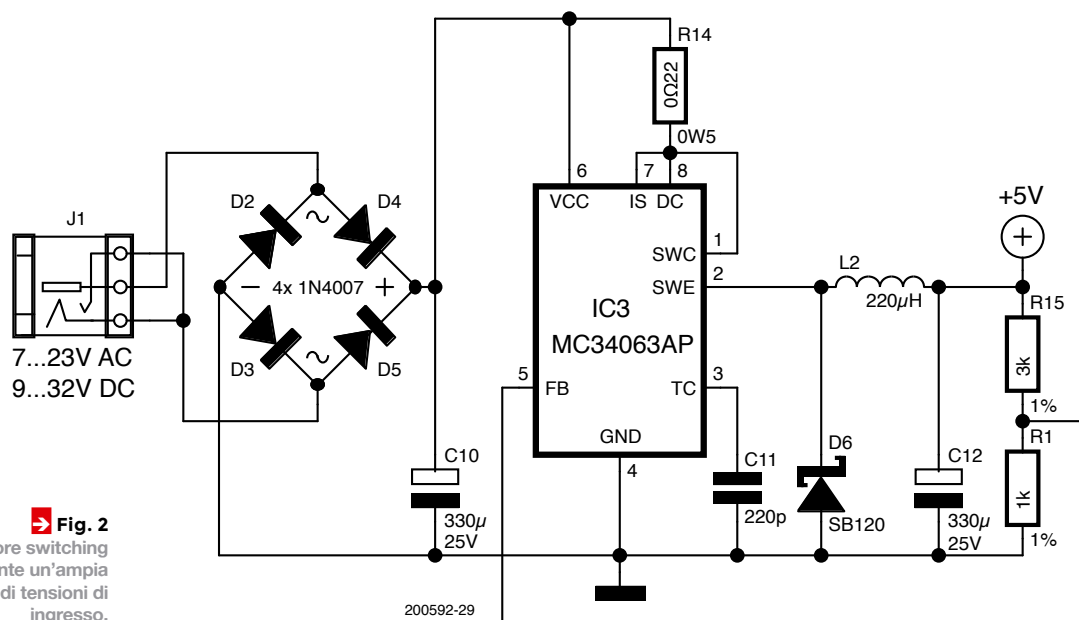


Fig. 2
L'alimentatore switching
consente un'ampia
gamma di tensioni di
ingresso.

circuito per avere un valore di riferimento. Questo valore, il riferimento zero, viene memorizzato e utilizzato in seguito per calcolare il valore di Lx.

L'ALIMENTAZIONE

L'alimentazione è basata su un MC34063 (IC3), un regolatore buck di tensione a commutazione che fornisce una tensione di alimentazione di +5V (**Fig. 2**). La sorgente di alimentazione applicata all'ingresso del circuito viene prima raddrizzata e poi filtrata dal condensatore C10 prima di entrare nell'IC3. Il motivo del raddrizzatore D2...D5 è quello di consentire l'alimentazione sia in AC che in CC senza doversi preoccupare della polarità. In questo modo il dispositivo accetta in ingresso da 7 VAC a 23 VAC o da 9 VDC a 32 VDC. La frequenza di lavoro è di circa 30 kHz. Il consumo del circuito è di soli 35 mA; sicuramente non è un problema per l'alimentatore previsto che può erogare una corrente massima di 1,2 A. Il pin centrale del connettore dell'alimentatore ha un diametro di 2 mm, il che consente di utilizzare molti adattatori di alimentazione standard.

FIRMWARE

Ho sviluppato un piccolo programma in mikroC (scaricabile dalla pagina di presentazione di questo numero di Elettronica In) che, come il progetto originale, consente di misurare qualsiasi induttanza e di visualizzarne il valore (in nH, µH o mH) sul display LCD. La selezione del campo di misura è automatica. La misurazione del valore di Lx avviene in due fasi:

1. Sequenza di calibrazione (con relè chiuso, **Fig. 3**).
2. Sequenza di misura con visualizzazione del valore calcolato.

Il Timer1 dell'MCU è configurato per iniziare il conteggio sul primo fronte di salita di RCO; conterà per un secondo (controllato dal Timer0). Dopo un secondo, il valore del Timer1 rappresenta la frequenza del segnale presente al suo ingresso. Ora il processore può eseguire tutti i calcoli necessari per determinare il valore di L1 (Fase 1) o di Lx (Fase 2). Il display LCD mostra ciò che sta accadendo. Durante il passo 1 si accende il LED giallo (LED1, Zero). Se l'induttanza da misurare è scollegata, viene visualizzato il messaggio "no coil detected" e si accende il LED rosso (LED3, Errore). In questo caso, verificare che la bobina sia collegata correttamente e riavviare la sequenza di misura premendo il pulsante.

L'uso del sistema è semplice e comodo in quanto completamente automatico. L'unica cosa da fare è premere brevemente (circa 0,5 s) il pulsante collegato a RC5 per riavviare la procedura, se necessario. Il display visualizza una freccia per indicare una nuova misurazione.

IL PROCESSO DI COSTRUZIONE

Una volta convalidato il prototipo, ho progettato un circuito stampato a doppia faccia (scaricabile dal sito della rivista). È stato progettato per essere inserito in un contenitore Multicomp MCRH3135 (**Fig. 4**). L'occhio esperto avrà notato le piccole

Per maggiori
informazioni
visita la pagina
elektormagazine.com/magazine/elektor-260/60545

dimensioni dei pad. La maggior parte di esse ha un diametro di 1,4 mm per un foro di 0,8 mm. Pertanto, consiglio vivamente di utilizzare un buon saldatore per montare i componenti o di ridisegnare la scheda in modo diverso.

Anche se ho fatto del mio meglio per realizzare un buon progetto di PCB, alla fine ho avuto due problemi che non ho risolto:

- Ho dimenticato di controllare le dimensioni e la forma delle piazzole del connettore di alimentazione J1. Se utilizzate lo stesso modello che ho utilizzato io, dovreste rielaborare un po' la scheda con un utensile tipo Dremel (**Fig. 5**). Poiché il connettore non è più fissato correttamente dai terminali saldati, l'ho incollato alla scheda (dopo averlo saldato).
- Non ho previsto un connettore ICSP per la programmazione in-circuit dell'MCU, il che è poco pratico. Tuttavia, è possibile programmare l'MCU rimuovendola dalla scheda e posizionandola su una breadboard o una scheda di prototipazione.

L'LCD è montato su un connettore a 16 vie a fila singola ed è fissato saldamente alla scheda con quattro distanziali in nylon M3 × 10 mm e bulloni e dadi adatti.

I circuiti integrati sono montati su zoccoli che, ovviamente, non sono assolutamente necessari (a meno che non si voglia riprogrammare il microcontrollore...). Il relè è saldato direttamente sulla scheda, soprattutto per evitare capacità e induttanze parassite indesiderate. I tre LED sono saldati quasi a filo con la scheda, ho utilizzato delle guide di luce (fibra ottica) per migliorarne la visibilità. Il file binario prodotto dal compilatore deve essere programmato nella memoria flash del microcontrollore con un programmatore come il PICKit3.

Dopo aver progettato il pannello frontale utilizzando lo strumento gratuito Front Panel Designer (<https://www.frontpanelexpress.com>), l'ho stampato su un foglio di alluminio autoadesivo (3M) con la mia stampante laser. Il risultato è pulito, ma tagliare correttamente il pannello frontale e incollarlo nella posizione corretta è un po' complicato.

DETTAGLI E RIFINITURE

Si raccomanda di mantenere i collegamenti tra il dispositivo e l'induttanza da testare Lx il più corti possibile, perché possono introdurre errori dovuti a capacità e induttanze parassite. I miei puntali sono fili multifilo da 12 cm 0,2 mm² con ganci a un'estremità. Come per qualsiasi strumento di misura, prima di effettuare qualsiasi misurazione accende-



Fig. 3
Per garantire la massima precisione, ogni misura viene eseguita in due fasi. Qui è illustrata la fase 1, la sequenza di calibrazione del sistema.

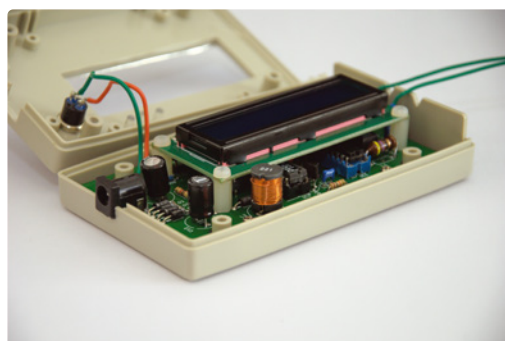


Fig. 4
La scheda si inserisce perfettamente all'interno del contenitore.

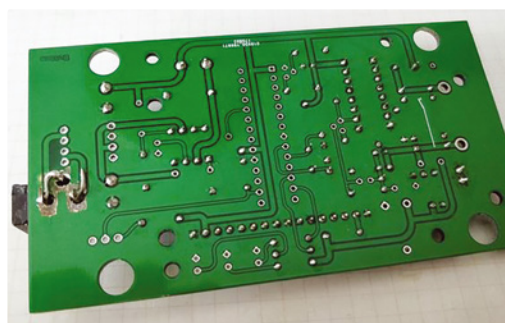


Fig. 5
Errare humanum est, e quindi il circuito stampato ha richiesto alcune modifiche per far sì che il plug di alimentazione potesse essere utilizzato.

re il dispositivo per alcuni minuti per stabilizzare la sua temperatura di esercizio.



Cosa occorre?

Il circuito stampato e il codice di questo progetto sono scaricabili dalla pagina di presentazione di questo numero. Il PCB può essere realizzato con il servizio PCBPRODUCTION disponibile dal sito www.futurashop.it

MAKER UNO

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

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

+

50
video
tutorial



€ 58,00

cod. MAKERKITBASE



IMPARIAMO A UTILIZZARE EAGLE

di DAVIDE MADDALENA e dell'ing. VINCENZO GERMANO

4

Proseguiamo il nostro viaggio alla scoperta di EAGLE, un potente strumento di progettazione elettronica, che permette di coprire l'intero ciclo di sviluppo di una scheda elettronica. Quarta puntata.

Siamo giunti finalmente allo sbroglio del circuito stampato mediante l'ambiente *board*, dopo aver affrontato ed analizzato le basi per disegnare lo schema elettrico. Sono state create le librerie con le quali si è disegnato lo schema elettrico all'interno dell'ambiente *schematic*, per poi documentarlo rendendolo meglio leggibile ed infine si è attuato il controllo delle regole elettriche prima di poter passare a disegnare la scheda *NordIOT*. Il passaggio da *schematic* a *board*, visto nella precedente puntata, è stato abbastanza immediato data l'equiva-

lenza dei due ambienti, perciò ci siamo focalizzati maggiormente verso i *layer* che assumono molta importanza soprattutto nel piazzamento, sbroglio e documentazione (su due o più *layer*). Passiamo a questo punto, dato che tutto è pronto, allo sbroglio della scheda analizzando le due possibilità per metterlo in pratica, ovvero lo sbroglio manuale ed automatico.

SBROGLIO MANUALE

Cominciamo dunque con l'argomentare lo sbroglio manuale a volte detto "tirare le piste" o, per usare



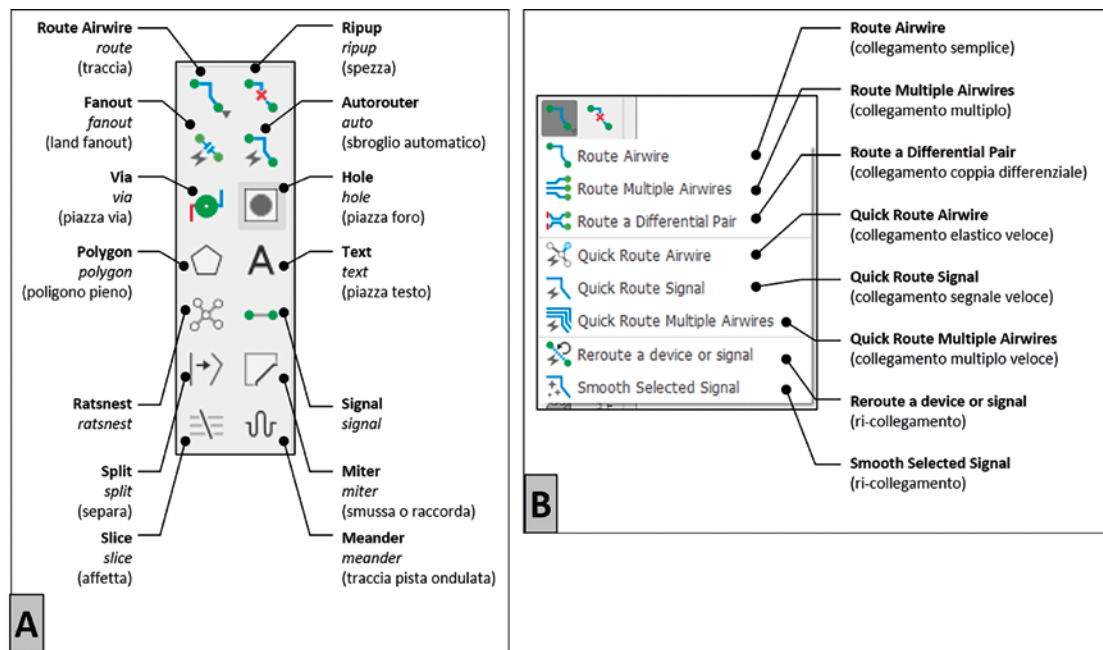


Fig. 1 Gli strumenti per routing.

la dizione inglese, "routing". Gli strumenti per lo sbroglio vanno di pari passo con quelli per disegnare le linee. Tuttavia ci sono due evidenze che distinguono questi strumenti: le piste corrispondono ad una net, vengono segnalate dagli elastici e, come abbiamo introdotto nelle puntate precedenti, rispondono alle regole di progetto ("design rule"). Le linee invece sono degli oggetti grafici che non assolvono ad alcuna funzione di collegamento. Un discorso analogo

vale tra piani (in rame) e poligoni. Per poter tracciare un collegamento tra due punti facciamo uso del comando "route" o dello strumento "Route Airwire" mostrato in Fig. 1A. In Fig. 1B vengono mostrati invece gli strumenti immediatamente disponibili quando si espande lo strumento di route. In Fig. 2 viene descritto come si modifica la barra degli strumenti quando si seleziona lo strumento "route". Notiamo, come già anticipato, che la barra strumenti è mol-

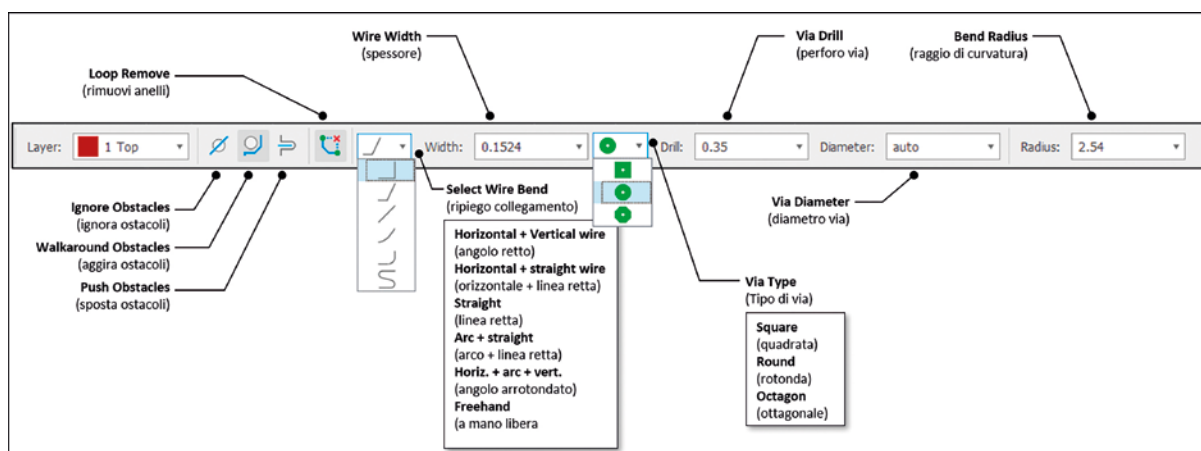


Fig. 2 Lo strumento route e le sue configurazioni.

to simile a quella per il disegno delle linee o dei poligoni. Ci sono però alcune aggiunte che riguardano soltanto le tracce (o route). In particolare durante il tracciamento (piste o via), di fronte a degli ostacoli (componenti, piazzole, piani, fori, altre tracce) si può scegliere:

- **Ignore obstacles:** in questo caso l'ostacolo mantiene la sua posizione e viene sovrapposto dalla nuova traccia indipendentemente dalle regole di disegno. In generale con questa scelta si può dar luogo a corti che dovranno poi essere valutati opportunamente.
- **Walkthrough obstacles:** in questo caso l'ostacolo mantiene la sua posizione e viene aggirato dalla traccia o dal piano. L'aggiramento, in base al tipo di ostacolo, verrà fatto rispettando le regole minime di disegno quindi generalmente questo tipo di scelta non genera DRC.
- **Push obstacles:** in questo caso si è completamente conformi alle regole di progettazione e l'ostacolo viene spostato (eventualmente anche gli altri oggetti intorno) in modo che la nuova traccia attraversi l'area che prima occupava l'ostacolo.

L'opzione "loop remove" consente di rimuovere automaticamente i loop ridondanti ritracciando completamente il percorso tra due pad. Possiamo iniziare il re-indirizzamento in corrispondenza di un pad, di una via, di un vertice o nel mezzo di un collegamento e terminarlo nello stesso modo. L'opzione "Select Wire Bend" permette di selezionare il tipo di piegatura che deve prendere il collegamento; allo stesso modo di una linea, selezionando il tasto destro del mouse mentre si traccia, si cambia il verso di piegatura (e.g.: da angolo retto in senso antiorario ad angolo retto in senso orario). Notiamo che la selezione dello spessore parte con un valore (0,1524"mm) che corrispondono ai 6mils impostati come "Minimum Width" nelle design rules.

Le opzioni successive riguardano il via previsto nel passaggio da uno strato all'altro; dunque si può scegliere il tipo di via (tra rotondo, quadrato o esagonale), il diametro del perforo (drill) e il diametro della piazzola del via. L'impostazione "auto" prevederà una piazzola in proporzione più grande del drill, come impostato nel tab "Annular Ring" delle design rules, purché si rimanga tra una dimensione minima ed una massima, come rappresentato in Fig. 3.

L'ultima impostazione nella barra degli strumenti riguarda il raggio di curvatura ("corner radius" o "bend radius") della pista. Impostando un valore all'interno della casella combinata, quando si sceglieranno dei tipi di curvatura diversi da "angolo retto", l'angolo della pista risulterà smussato o raccordato, a seconda del tipo di "wire bend", con raggio pari a quello impostato. Nella Fig. 4 viene esemplificato quanto sopra descritto, avendo impostato un raggio di curvatura di 0,5mm e una griglia (principale) di 0,5mm.

Una volta fatta una panoramica sullo strumento "route" e le sue impostazioni possiamo passare finalmente allo sbroglio: proviamo a collegare due pad (ovviamente appartenenti alla stessa net) con una pista lasciando le impostazioni predefinite. La sequenza è rappresentata nella Fig. 5 e di seguito riassunta.

1. Ipotizziamo di dover collegare direttamente due piazzole, nell'esempio hanno net N\$12, di due componenti posti adiacenti tra loro.
2. Selezioniamo lo strumento "route" mantenendo le impostazioni di sbroglio predefinite.
3. Posizioniamoci e clicchiamo sulla prima piazzola, nell'esempio quella a destra. Notiamo che dopo aver cliccato sulla piazzola appare una croce in corrispondenza a segnalare l'inizio della pista.
4. Spostiamoci verso la seconda piazzola, nell'esempio quella a destra. Notiamo come la sagoma della pista,

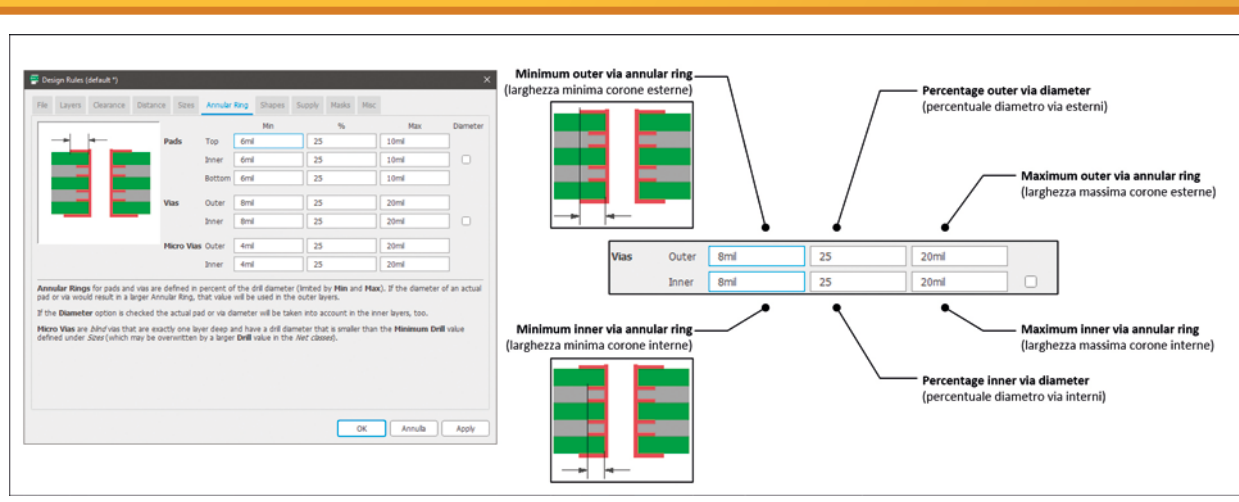


Fig. 3 Le impostazioni dell'Annular Ring.

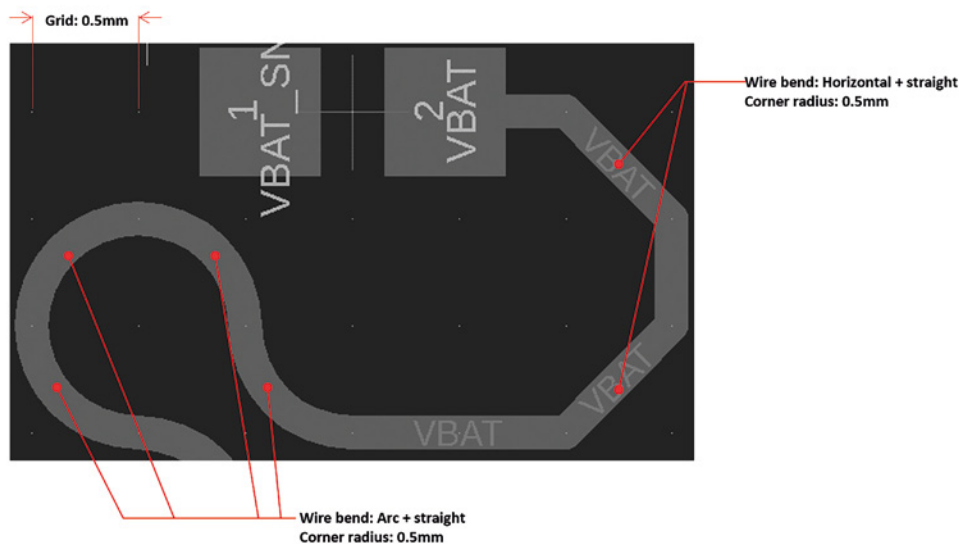


Fig. 4 Il raggio di curvatura e il wire bend.

scontornata dello stesso colore della piazzola, segue il movimento del puntatore a passi di dimensioni della griglia.

5. Posizioniamoci sulla piazzola di destinazione. In questo caso l'aggancio del puntatore viene segnalato tramite una croce ancor prima di cliccare.
6. Clicchiamo sulla piazzola e la sagoma della pista diventa solida quindi potremo ritenere concluso il collegamento.

Abbiamo così creato il nostro primo collegamento avendo mantenuto le impostazioni predefinite.

Se però nasce la necessità, anche a posteriori, di modificare le caratteristiche di una pista è sufficiente cliccare con il tasto destro sulla pista interessata e selezionare "Properties"; apparirà la finestra delle proprietà come mostrato in Fig. 6. Come possiamo vedere, nella finestra di Fig. 6, è possibile modificare diversi parametri della pista come la posizione,

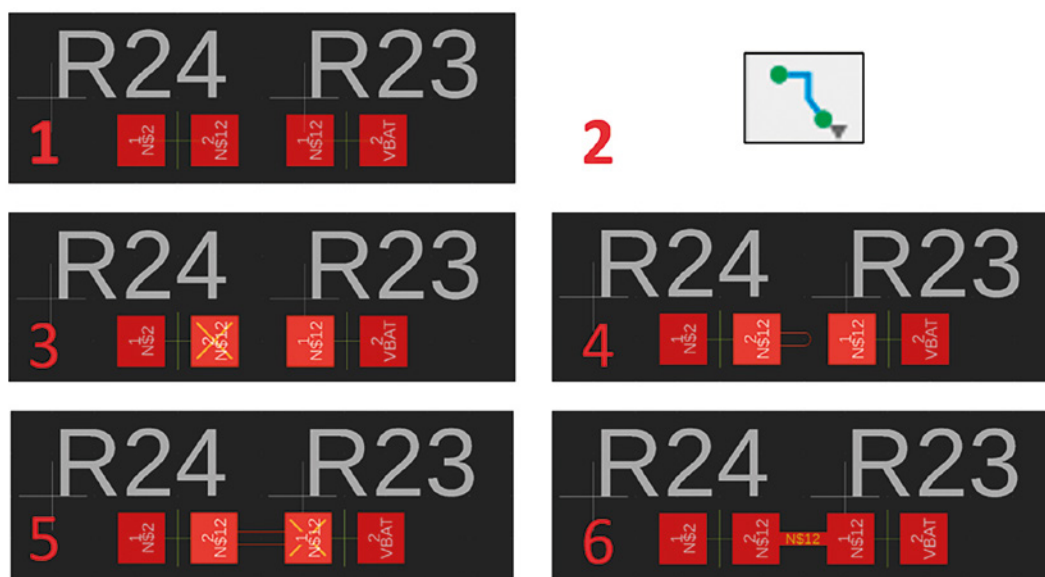


Fig. 5 La sequenza per il route.

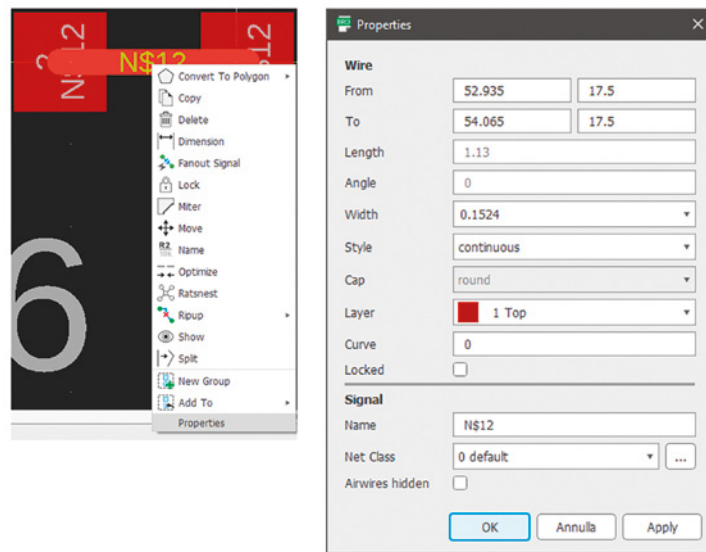


Fig. 6 La modifica delle caratteristiche di una pista.

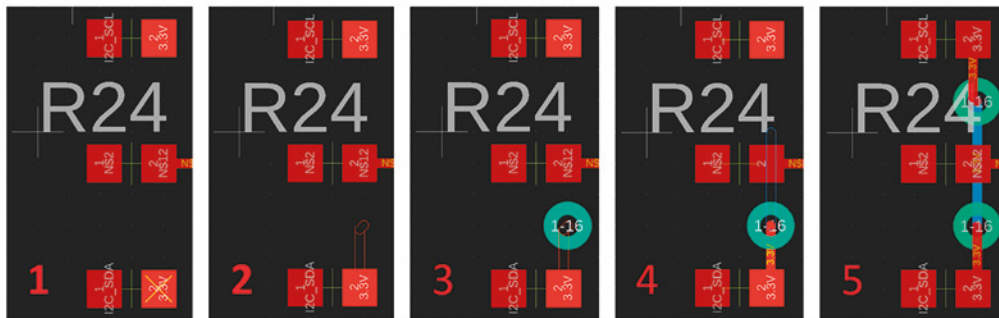


Fig. 7 Sequenza di route con l'aggiunta di un via.

dimensione, stile ma anche le caratteristiche della net. Se invece abbiamo la necessità di passare per un altro layer (per esempio in caso di presenza di ostacoli), quindi di introdurre dei via, la nuova sequenza è descritta in Fig. 7 e si modifica come segue.

1. Sempre dopo aver selezionato lo strumento "route", posizioniamoci e clicchiamo sulla prima piazzola. La solita croce indicherà l'aggancio alla piazzola e la partenza della pista.
2. Spostiamo il puntatore nella posizione dove piazzare il via. La sagoma della pista di colore "rosso" ci indicherà la posizione.
3. Clicchiamo con il tasto centrale del mouse oppure selezioniamo la barra spaziatrice quindi apparirà il via con le caratteristiche impostate. Notiamo che il via è ancora

mobile; infatti la pista non è ancora solida e spostando il puntatore la piazzola del via segue il movimento.

4. Clicchiamo sulla posizione voluta. Notiamo che il via viene piazzato e la pista che collega il pad diventa solida. Da questo momento la pista passa sull'altro layer della scheda infatti vedremo che il colore che la sconta è cambiato (nell'esempio è blu cioè il colore del bottom).
5. Ripetiamo gli stessi ragionamenti fatti per le piste e i via fino alla successiva piazzola.

Anche per il via si possono modificare le proprietà a posteriori come rappresentato in Fig. 8.

A questo punto siamo in grado di tracciare manualmente le nostre piste e, eventualmente, modificarne le proprietà o addirittura assegnarne una classe di net.

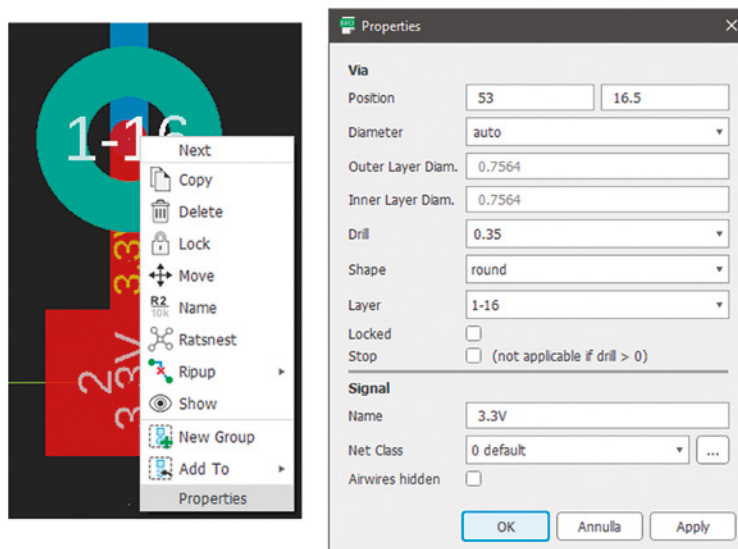


Fig. 8 Le proprietà di un via.

Nel caso in cui si voglia eliminare un collegamento allora si può ricorrere al classico **CTRL+Z** per i più recenti oppure al comando **"ripup"** rappresentato nella Fig. 9.

Il significato delle opzioni selezionabili è abbastanza intuitivo; tuttavia gli ultimi due strumenti hanno particolare utilità quando si vuole eliminare in un colpo solo rispettivamente tutti i collegamenti o tutti i piani.

SBROGLIO AUTOMATICO

Come suggerisce la parola stessa, per sbroglio (o anche **"routing"**) automatico si intende la modalità di tirare le piste in modo automatico, cioè senza la mano dell'utente se non per le impostazioni iniziali delle regole di disegno e caratteristiche dello sbroglio automatico.

Questo strumento si applica alle net ovvero ai collegamenti e può essere usato in toto su tutte le net della scheda (azione che può aumentare enormemente il tempo di elaborazione) oppure selettivamente per alcune net. Vediamo come funziona in quest'ultimo caso poiché la celerità di elaborazione dello sbroglio automatico (detto brevemente **"autorouter"**) è efficace se usato localmente. Per lanciare l'autorouter selezioniamo lo strumento riportato in Fig. 10A oppure usiamo il comando **"auto"**. Apparirà la finestra di impostazione dello strumento **"Autorouter Main Setup"**.

Nella finestra individuiamo le seguenti opzioni principali:

1. **Preferred direction:** selezioniamo (per ogni layer) questa opzione se necessitiamo una direzione preferenziale di sbroglio altrimenti l'impostazione auto è sufficiente.

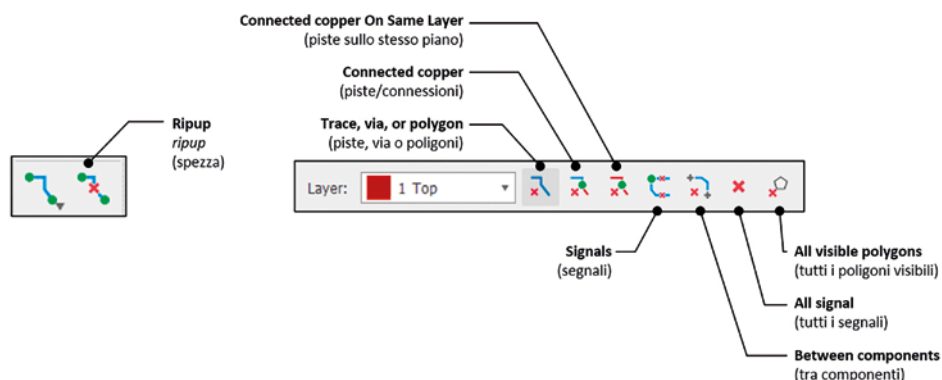


Fig. 9 Lo strumento ripup.

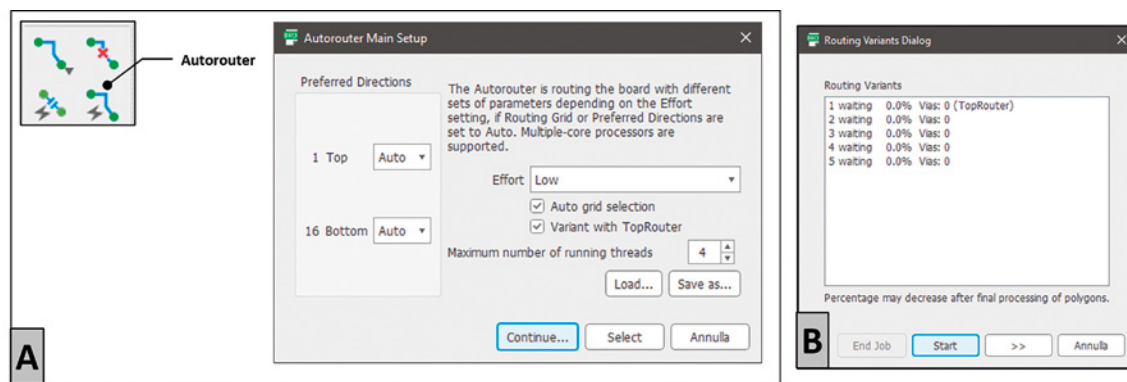


Fig. 10 Lo strumento autorouter.

2. **Effort:** con questa opzione definiamo come deve lavorare l'autorouter. Il tempo di lavoro sarà tanto maggiore quanto più alto è il livello di impegno richiesto ("effort").
3. **Number of threads:** questa opzione imposta il massimo numero di thread (carico) per la CPU. Di sicuro un livello più alto inficia il tempo di elaborazione.

A questo punto si possono scegliere due strade:

- Selezionare "Continue..." per procedere allo sbroglio dell'intera scheda. Quest'azione, come già anticipato è fortemente sconsigliata a meno che non si tratti di una scheda con pochi collegamenti. Questa scelta porta direttamente alla finestra "Routing Variants Dialog" che vedremo tra un attimo.
- Selezionare il tasto "Select" che permette la selezione delle net di cui si vuol procedere con lo sbroglio. In questo caso vediamo più avanti i passi intermedi da seguire prima di avviare lo sbroglio.

Qualunque sia la scelta, facendo un passo avanti, prima di processare lo sbroglio automatico si passa per la finestra "Routing Variants Dialog" mostrata nella Fig. 10B. In questa finestra, selezionando il pulsante ">>", si possono impostare ulteriori parametri per lo sbroglio come mostrato in Fig. 11.

In questo corso non approfondiremo l'impostazione dello sbroglio automatico al di fuori delle impostazioni predefinite infatti ci siamo limitati solo ad introdurre i concetti fondamentali per il suo utilizzo. Una volta pronti basta selezionare il pulsante "Start" per avviare il processo. Il termine del processo dipende sicuramente dalle opzioni che abbiamo

selezionato ma anche dalle caratteristiche della nostra macchina quindi potrebbero variare anche sostanzialmente. Quando selezioniamo il tasto "Select" nella finestra "Autorouter Main Setup" il focus torna sul nostro ambiente layout. Notiamo che in basso a sinistra nella barra di stato compare il messaggio "Select signals or click GO to start". Come mostrato in Fig. 12, è quindi possibile selezionare le net per cui si vuole eseguire lo sbroglio (nell'esempio sono le net GND e VIN) dunque si seleziona lo strumento "GO" posto in alto nella barra degli strumenti.

Si consiglia di tenere accesi gli elastici in modo da poter selezionare più rapidamente le net.

A questo punto di nuovo appare la finestra "Routing Variants Dialog" nella quale si può lanciare lo sbroglio automatico questa volta per le sole net selezionate. Tuttavia, seppure rimane uno strumento validissimo e allo stesso tempo utilissimo, è bene precisare che l'autorouter non è perfetto e quindi necessita del controllo (a posteriori) dell'utente per

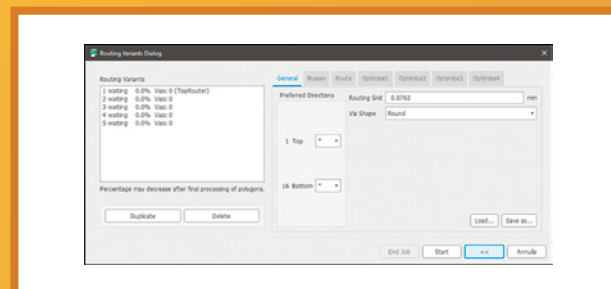


Fig. 11 Le impostazioni dell'autorouter.

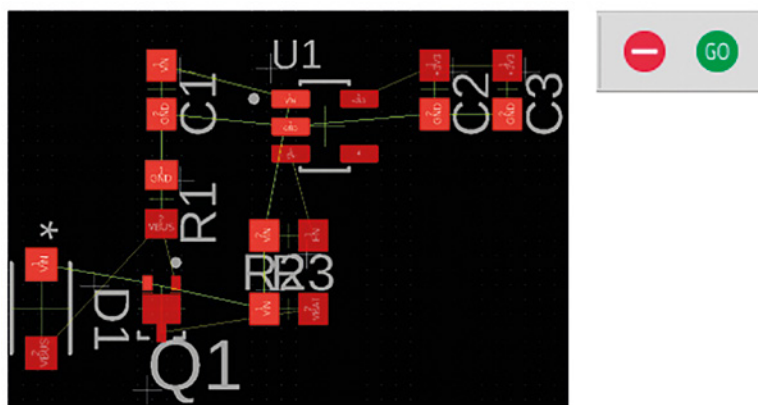


Fig. 12 Le impostazioni dell'autorouter.



Fig. 13 Lo strumento polygon.

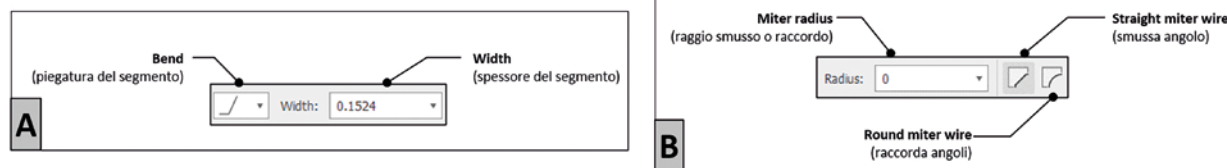


Fig. 14 Le varie impostazioni di polygon.

lo meno per quanto riguarda l'esclusione di problemi come le "acid trap", collegamenti ad angolo retto, e via dicendo.

PIANI CONDUTTORI E RIEMPIMENTI

Durante lo sbroglio possiamo trovarci in una situazione nella quale è necessario collegare più punti, ovviamente appartenenti alla stessa net, distribuiti però come in una "costellazione" sulla scheda. Nella maggior parte dei casi questi punti appartengono alle linee di alimentazione e di riferimento, ma più piccole "costellazioni", appartenenti a qualsiasi net, si potrebbero comunque trovare localizzate sulla scheda. Questi tipi di collegamento sono possibili sicuramente attraverso delle piste; spesso però il modo migliore per il loro collegamento sarebbe attraverso un piano di conduttore (generalmente indicati come piani di alimenta-

zione e di massa). In questa sezione andiamo a vedere come si crea un piano conduttore e quali tipologie di riempimento sono possibili.

Per creare un piano conduttore bisogna selezionare il layer (ovviamente di conduttore) sul quale disegnare e poi selezionare lo strumento "polygon" o il comando "POLYGON" mostrato in Fig. 13 insieme alla barra degli strumenti contestualizzata.

Abbiamo già introdotto questo strumento nella parte relativa allo schema ma nell'ambiente "board" assume delle opzioni aggiuntive per quanto riguarda i piani conduttori. Ovviamente un poligono è formato da diversi lati dunque la prima parte della barra degli strumenti, riportata in Fig. 14A, rappresenta nient'altro che il tipo di piegatura e lo spessore di ciascun lato alla stregua di quanto visto per le piste, men-

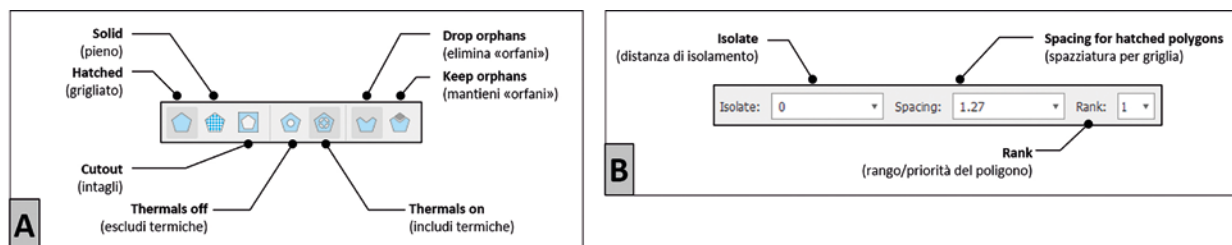


Fig. 15 Le impostazioni rilevanti di polygon.

tre l'ultima parte, riportata in Fig. 14B, si riferisce a quanto abbiamo visto durante la creazione del contorno scheda e più in particolare allo strumento "miter".

Le opzioni più interessanti, per quanto riguarda i poligoni pieni per piani conduttori, sono localizzate nella parte centrale della barra degli strumenti e riportati nelle Fig. 15A e 15B.

In particolare, nel disegnare un piano conduttore, possiamo scegliere se debba essere pieno, grigliato o intagliato all'interno di un altro piano. La Fig. 16A mostra le differenze tra le tre opzioni appena citate.

Tra un piano pieno ed uno grigliato concettualmente non sussistono grosse differenze e comunque la scelta di uno o l'altro esula dagli scopi di questo corso. Un piano intagliato invece è un piano posto all'interno di un altro quindi generalmente appartiene ad una net differente posta all'interno

di una meno prioritaria. Quest'ultima osservazione ci aiuta a definire il concetto di "rank" o, se vogliamo, di rango/priorità di un piano conduttore come mostrato nella Fig. 16B.

Questo concetto si applica sia ai piani intagliati che non. EAGLE mette a disposizione fino a 6 livelli di priorità e, a discapito del numero, più è elevato il rank di un piano minore sarà la sua priorità rispetto ad altri piani adiacenti o intagliati. La Fig. 16C mostra il comportamento di due piani a seconda della priorità attribuita. Rimanendo ancora sulle opzioni della parte centrale andiamo a descrivere i concetti di "isolate" e "spacing" che fanno gruppo con lo "width":

- **Width (Fig. 14A):** rappresenta lo spessore dei lati del bordo poligono e, nel caso di grigliato, anche delle linee interne.
- **Spacing (Fig. 15B):** si applica solo ad un piano grigliato e rappresenta lo spazio che intercorre tra una linea e l'altra.

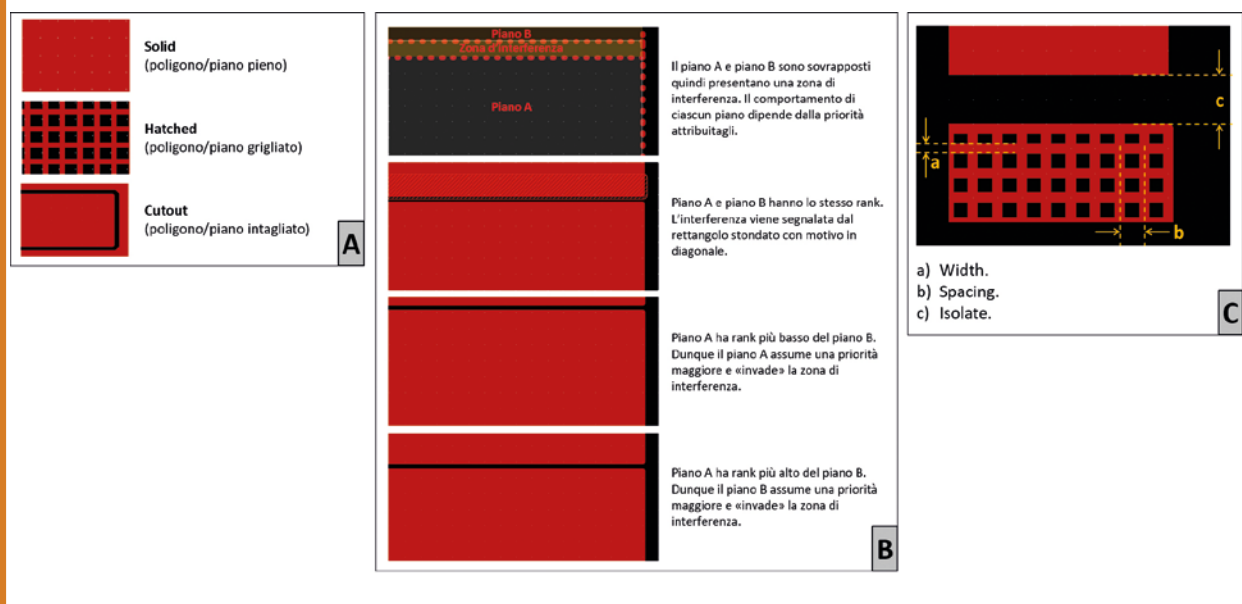


Fig. 16 Le differenze tra i piani.

- **Isolate (Fig. 15B):** si applica a qualunque tipo di poligono e rappresenta la dimensione della zona di isolamento che il piano manterrà rispetto agli altri conduttori di net diversa. Se questo parametro viene mantenuto a zero allora prevarranno le impostazioni di "design rule".

La Fig. 16C riassume le principali impostazioni sopra elencate. Le ultime impostazioni riguardano le termiche ("thermals") e gli orfani ("orphans") mostrate nelle Fig. 17A e 17B. Le termiche sono delle tipologie di approccio verso le piazzole che permettono un discreto collegamento elettrico con i piani conduttore ma garantiscono un certo isolamento termico della piazzola rispetto allo stesso piano. In effetti un piano conduttore si comporta termicamente come aletta di raffreddamento quindi una piazzola affogata nel piano (impostazione "thermals off") verrebbe continuamente raffreddata. Il problema si pone durante la saldatura perché le dimensioni del piano potrebbero non garantire una corretta concentrazione di calore sulla piazzola con il risultato di una saldatura carente o fredda. Per questo motivo si vanno ad attivare le termiche (impostazione "thermals on") come mostrato in Fig. 17A. Le dimensioni delle termiche dipendono dall'impostazione "width". Gli orfani sono dei piani conduttore o residui di piano che, per qualsiasi ragione, rimangono isolati dalla net. Nella

Fig. 17B viene mostrato il risultato dell'impostazione "drop orphans" in alto e "keep orphans" in basso. Notiamo anche che, nella stessa Fig. 17B, il piano GND collegante le due piazzole a destra non si comporta come orfano nonostante sia isolato dal piano principale; questo comportamento è dovuto al fatto che tale piano rimane concettualmente collegato alla net GND nonostante non lo sia fisicamente. Infine nella Fig. 18 viene rappresentata la finestra proprietà di un piano conduttore creato con un poligono. Notiamo che le stesse proprietà vengono riportate nel tab "Inspector" una volta che il piano interessato viene selezionato.

Una volta illustrate le proprietà e le opzioni di un piano conduttore proviamo a creare il nostro piano, per esempio partendo dalla net GND. Nella Fig. 19 vengono illustrati i passaggi.

Notiamo che una volta creato il poligono (nell'esempio è un semplice rettangolo) non appare immediatamente riempito (pieno o grigliato) ma solo tratteggiato. Per riempire il poligono appena creato è sufficiente selezionare lo strumento "ratsnest" o il relativo comando. A questo punto la procedura di creazione è terminata. Nel caso in cui si volessero fare delle modifiche al piano è opportuno selezionare l'opzione "All visible polygons" dello strumento "ripup" dunque, una volta che il piano riappare nuovamente tratteggiato, utilizzare

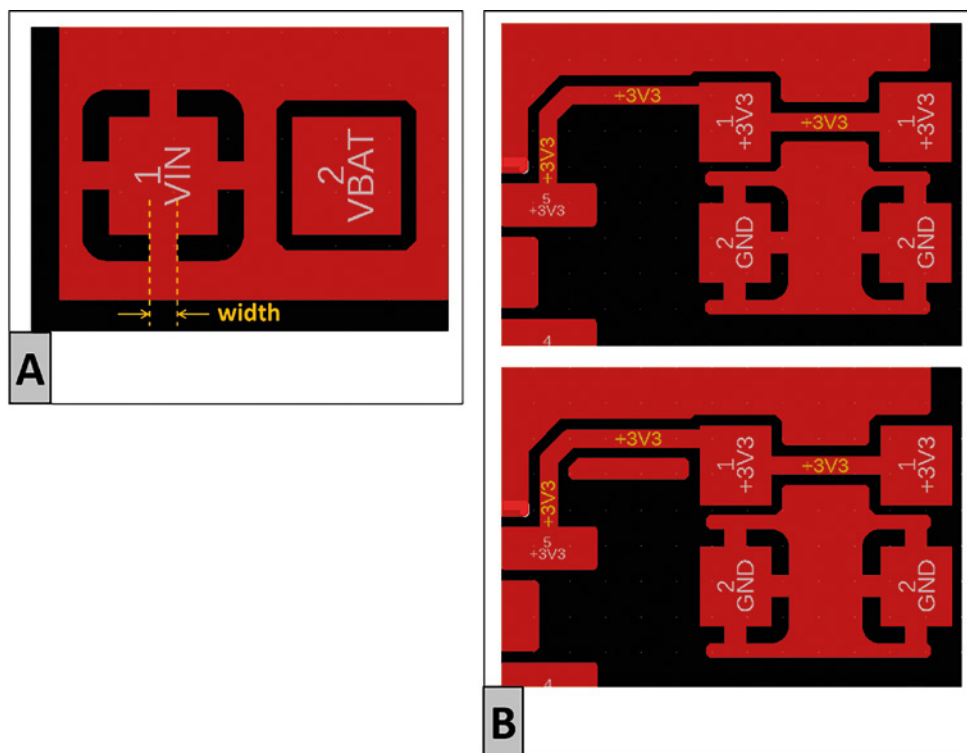


Fig. 17 Le differenze tra i piani.

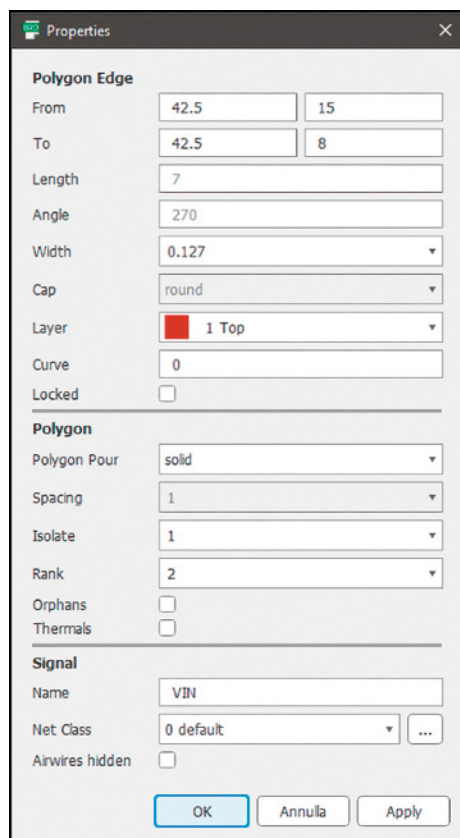
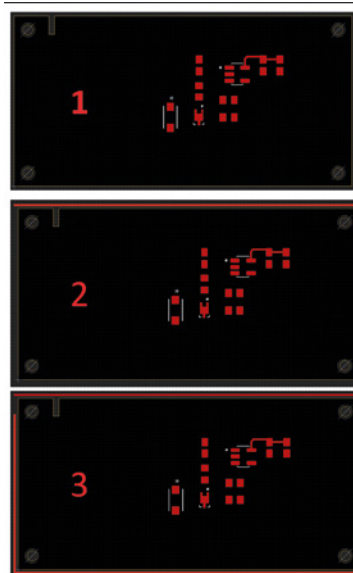


Fig. 18 Le proprietà di un piano conduttore

gli strumenti "split" o "miter" per realizzare le modifiche. Seppure è possibile realizzare la creazione dei piani conduttori in qualsiasi momento è consigliabile crearli una volta che tutti i restanti collegamenti sono stati ultimati. In altri termini la creazione dei piani conduttori è un'attività da riservarsi al termine di tutte le attività di sbroglio prima di passare al controllo delle regole di progetto quindi alla generazione dei file di fabbricazione.

INSERIMENTO LOGO E TESTO

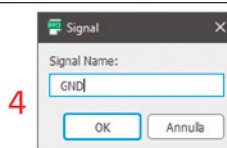
Prima delle attività di controllo e fabbricazione di un circuito stampato, possiamo approfondire un ultimo passaggio. Spesso è necessario che sulla scheda si vogliano inserire un segno identificativo (logo), una mini descrizione della scheda o la versione di layout. Queste informazioni possono essere inserite direttamente su rame oppure su altra tipologia di layer (21 tPlace, 22 bPlace), in genere, la serigrafia o "silkscreen". Che sia su rame o su serigrafia un logo ha la necessità di essere importato mentre un qualunque testo può essere inserito mediante lo strumento "text" o il relativo comando TEXT. Tuttavia se si decide di inserire queste informazioni direttamente su rame un logo o un testo non devono andare in interferenza con altri oggetti sulla scheda mentre utilizzando una serigrafia, che solitamente è "in-chiostro", non ci sono grossi problemi di interferenza. Vediamo come inserire un qualsiasi testo sulla scheda. Per completezza supponiamo di volerlo inserire su rame e, soprattutto, sul piano conduttore appena creato. Chiaramente questo testo non avrà nessuna net quindi, per non andare in interferenza con il piano, sarà necessario



Una volta selezionato lo strumento poligono clicchiamo (all'esterno del contorno scheda) in alto a sinistra.

Supponendo di aver selezionato la piegatura ad angolo retto, muoviamo il puntatore verso destra e, sempre all'esterno del contorno scheda, arriviamo in alto a destra poi spostiamo il puntatore verso il basso.

Clicchiamo poi in basso a destra e ripetiamo le operazioni sopra spostando il puntatore verso sinistra fino all'angolo in basso a sinistra poi spostiamo il puntatore verso l'alto.



Una volta raggiunto nuovamente l'angolo in alto a sinistra (il punto di partenza) clicchiamo. Il poligono si chiude e verrà richiesto a quale net si vuole associare il piano.

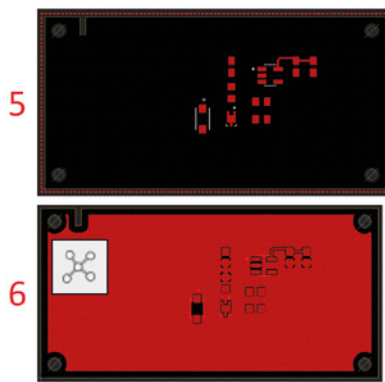
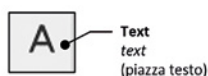
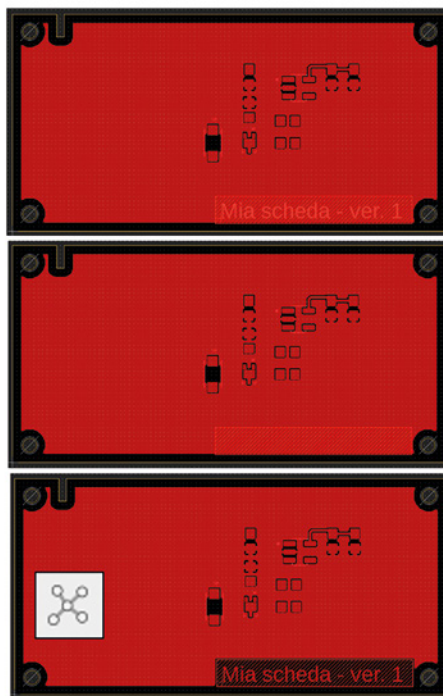
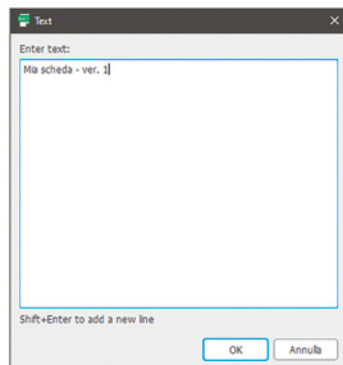


Fig. 19 I passaggi per la creazione di un piano conduttore.



Una volta selezionato il layer (in questo esempio 1 Top) si seleziona lo strumento testo quindi appare la finestra per l'inserimento testo.



Appena si seleziona OK nella finestra, il testo appare all'interno di un rettangolo riempito a righe ed è possibile spostarlo in qualsiasi posizione del piano di lavoro.

Individuata la posizione desiderata per il piano clicchiamo con il tasto sinistro dunque il testo si fissa. Appare però affogato nel piano e ovviamente non visibile. Tuttavia il rettangolo in cui è contenuto individua facilmente la sua posizione.

Lo strumento o il comando RATSNEST permette di aggiornare la definizione del testo all'interno del piano quindi di renderlo visibile. Il rettangolo che lo racchiude rimarrà comunque visibile.

Fig. 20 I passaggi per l'inserimento su rame di un testo.

intagliarsi nel piano la zona dove inserirlo. Dunque potremo applicare le regole di modifica del poligono viste prima, operazione però statica che non tiene in conto della dimensione del testo, oppure inserire il testo nella zona desiderata (anche nel mezzo del piano) e poi selezionare lo strumento "ratsnest". Il procedimento è molto semplice ed esemplificato nella Fig. 20.

È bene sottolineare che il rettangolo contenente il nostro testo compare solo se il testo è piazzato su un layer di conduttore (top, bottom o interno che sia). Nel caso si scelga un layer di serigrafia non apparirebbe alcun rettangolo. A questo punto, se volessimo modificare il contenuto del testo o le sue caratteristiche (e.g.: le dimensioni) è sufficiente selezionare le proprietà del testo (click destro sul testo e poi "Properties") oppure modificarle direttamente nel tab "Inspector". Ovviamente ricordiamoci di selezionare sempre "Ratsnest" dopo averlo modificato ed eventualmente spostato. In Fig. 21 viene mostrato il risultato finale comparato con quanto descritto per la serigrafia.

Prima di inserire il logo bisogna spendere qualche parola a riguardo. EAGLE permette l'importazione sia di file vettoriali (*.dxf) sia di immagini (*.bmp) tramite apposite ULP. Spesso il logo è disponibile in formato di immagine dunque procediamo con questo tipo di formato, premesso che il file di origine deve essere un bitmap di massimo 256

colori dei quali solo 32 possono essere importati su 32 layer differenti. Il formato di uscita sarà una serie di rettangoli che ricalcheranno l'immagine; EAGLE consiglia di utilizzare questo formato di uscita come base per il disegno del logo sotto forma di serigrafia o di rame. Tuttavia EAGLE offre una discreta flessibilità nel generare i file di fabbricazione dunque le immagini, una volta importate, possono essere lasciate nel formato d'uscita senza problemi.

Per descrivere l'inserimento del logo partiamo da un'immagine monocromatica (bianco e nero) come mostrato nella Fig. 22A; nella stessa figura sono riportate le caratteristiche dell'immagine che è sempre bene tenere in mente per l'importazione.

Selezioniamo poi l'elenco delle ULP ed in particolare lo



Fig. 21 Risultato finale dell'inserimento su rame di un testo.

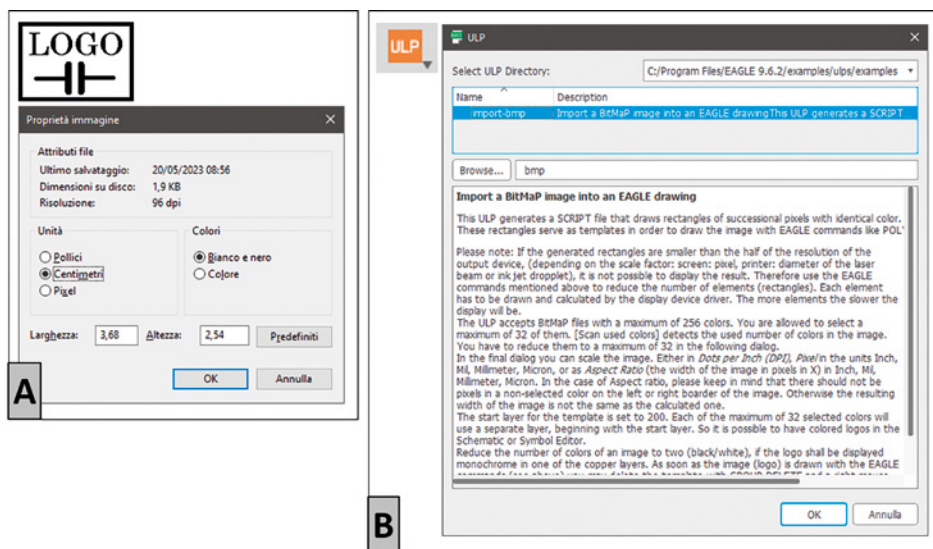


Fig. 22 L'inserimento di un logo.

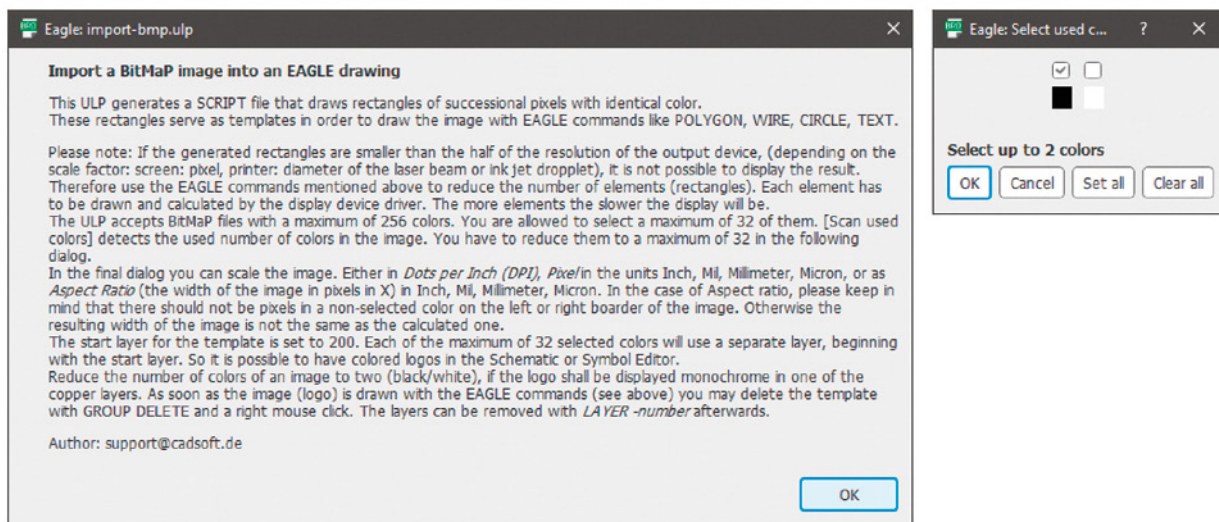
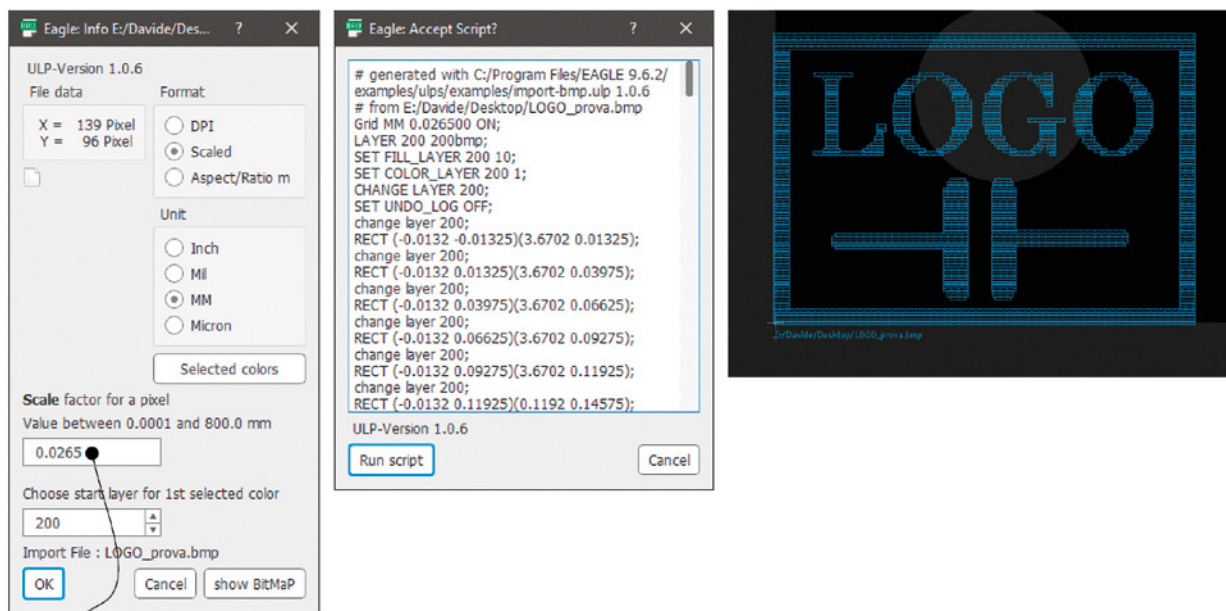


Fig. 23 La finestra di selezione dei colori per un logo.

strumento "Import-bmp" come mostrato in Fig. 22B. Nel selezionare OK appare una finestra di descrizione e riepilogo dello strumento; selezionando ancora OK e, dopo aver selezionato l'immagine bitmap da importare, apparirà una finestra per la selezione dei colori come mostrato in Fig 23. L'immagine d'esempio è monocromatica quindi sono

presenti solo due colori: nero e bianco. Ecco spiegato perché nella finestra di selezione saranno presenti solo due selezioni (delle 32 possibili). Ciascuna selezione verrà importata in un layer diverso a partire dal layer 200. Siccome il contenuto dell'immagine è visibile nella parte in nero è conveniente importare solo questo colore quindi selezioniamo



$$scala = \frac{\text{dimensione finale}}{\text{pixel}} = \frac{2,54\text{cm}/10}{96\text{pixel}} = \frac{2,54\text{mm}}{96\text{pixel}} = 0,0265$$

Fig. 24 Le impostazioni per l'inserimento di un logo.

mo la casella in alto del quadrato nero e selezioniamo OK. Apparirà una finestra di impostazione per l'importazione mostrata in Fig. 24 insieme al risultato finale.

Per quest'ultimo passaggio è bene spendere qualche parola aggiuntiva. Abbiamo già detto che è sempre meglio tenere traccia delle caratteristiche dell'immagine soprattutto per la scala. Sappiamo che l'immagine di esempio è alta 2,54cm (ovvero 24,5mm) o 96pixel. Supponiamo di volerla importare in modo tale che l'altezza del logo sia un decimo di quella originale. In altri termini supponiamo che il logo debba essere altro circa 2,54mm (dimensione finale). Dunque nella finestra di impostazione selezioneremo Format→Scaled, Unit→MM e nella casella "Scale factor for a pixel" (fattore di scala per pixel) inseriremo il valore 0,0265 come riportato nella precedente figura. Ovviamente si può fare un ragionamento simile nei confronti della lunghezza. A questo punto possiamo ritenere concluso l'importazione del logo. Il layer o eventualmente i layer appena creati si possono raccogliere in un gruppo nominale in modo che sarà più semplice spostare il logo a piacere. Si potrebbe pensare anche di importare l'immagine una sola volta creando un componente

logo, sottoforma di footprint ed inserirlo all'occorrenza nel layout. A voi la scelta.

CONTROLLO DELLE REGOLE DI PROGETTO

Abbiamo già affrontato, nella parte di schema elettrico, il controllo finale assistito da ciò che abbiamo definito ERC (electrical rule check). In effetti questo strumento può aiutarci ad individuare parti di disegno un po' ambigue o addirittura errori. Nella parte di layout esiste uno strumento altrettanto utile che prende il nome di DRC (design rule check) ovvero il controllo delle regole di progetto. Si tratta delle stesse regole di progetto che abbiamo introdotto insieme all'ambiente board e richiamate nei vari passaggi di piazzamento e sbroglio. Per avviare il controllo delle regole di progetto è sufficiente usare il comando DRC o lo strumento "drc" mostrato nella Fig. 25A. Apparirà nuovamente la finestra delle regole di progetto (Fig. 25B) dove, questa volta, saranno visibili anche i pulsanti "Check" e "Select" al posto di "OK". La differenza tra questi due pulsanti è molto semplice ed esemplificata in Fig. 25C: "Check" avvia il controllo delle regole di progetto



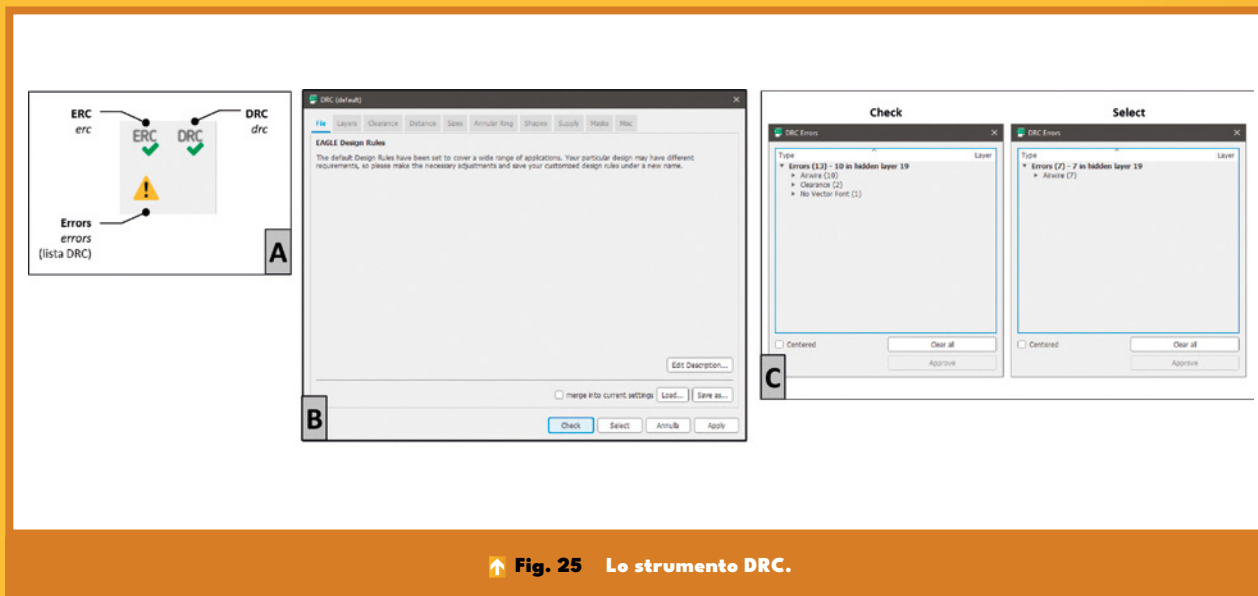


Fig. 25 Lo strumento DRC.

su tutta la scheda, mentre "Select" permette di selezionare un'area all'interno della quale avviare il controllo. Prima di avviare il controllo, totale o parziale, è comunque possibile modificare le regole di progetto. Una volta terminato il controllo delle regole di progetto apparirà la finestra "DRC Errors" che, come per l'ambiente schematic, riepiloga e segnala tutte le violazioni delle regole sia che siano errori sia che siano avvisi, come mostrato in Fig. 25C; possiamo dunque controllarli oppure approvarli, tramite il pulsante "Approve". Ovviamente risolvere l'errore/avviso significa lavorare manualmente. Nell'esempio di Fig. 25C l'ultimo punto rappresenta un errore il cui messaggio cita: "No Vector Font". Questo errore è relativo al testo che abbiamo introdotto come incisione su rame e segnala un Font non vettoriale. Per risolverlo basta entrare nelle proprietà del testo e selezionare "vector" nella combo box "Font". Tuttavia "errors" e "warnings" non scompaiono se approvati ma vengono spostati più in basso nella stessa lista "DRC Errors" sotto la voce "Approved". Non è possibile approvare tutti i messaggi in una sola volta ma è necessario controllarli uno per uno. È invece possibile cancellare la lista dei messaggi tramite il pulsante "Clear all". In questo caso non si cancellano "errors" e "warnings" ma solo il contenuto della lista; lanciando nuovamente DRC si riproporrà la lista. Quindi lo strumento clear è bene usarlo quando si risolve un errore ma poiché la lista non si aggiorna dinamicamente, è necessario cancellare tutta la lista e poi rilanciare DRC.

GENERAZIONE DEI FILE DI OUTPUT

Una volta terminato il progetto e controllati tutti gli errori del caso passiamo alla generazione dei file detti di output per la produzione della scheda. Esistono generalmente

diversi file necessari per la produzione, di seguito ne elenchiamo alcuni.

- **GERBER:** sono costituiti da una serie di documenti in formato standard ed indipendente dal CAD che i produttori utilizzano per la realizzazione del circuito stampato e del loro montaggio.
- **DRILL:** conosciuti anche come NC Drills (ovvero "Numeric Control Drills") rappresenta un file che contiene, in formato numerico, tutte le informazioni relative ai perfori di via e fori (hole). Dunque include la posizione, la dimensione e il tipo di tool per la creazione del foro.
- **Pick & Place data:** questi file descrivono le posizioni (in coordinate x-y) dei riferimenti componente, la rotazione rispetto al proprio centro e la specchiatura rispetto al layer (Top e Bottom). Sono file che si interfacciano direttamente con la macchina Pick-and-place che accetta il formato ASCII.
- **Fabmaster, ODB++:** rappresentano file ad uso industriale (comunque fuori dagli scopi di questo corso) dotati di algoritmi per l'automazione del test e di database per la produzione della scheda.

Approfondiremo alcuni dei punti descritti sopra in modo più dettagliato all'occorrenza; per il momento diciamo che questi file (spesso chiamati "file d'uscita") possono essere definiti come un lavoro del processore CAM ("CAM Processor"), il cui strumento è presentato in Fig. 26A.

Avviando il processore CAM appare la finestra mostrata in Fig. 26B. Potendo generare diversi file di output, la finestra del CAM processor ha una parte generica e propria della finestra mentre la parte a destra cambia dinamicamente a seconda del file selezionato. In alto troviamo il nome del template cioè il modello utilizzato per la generazione dei

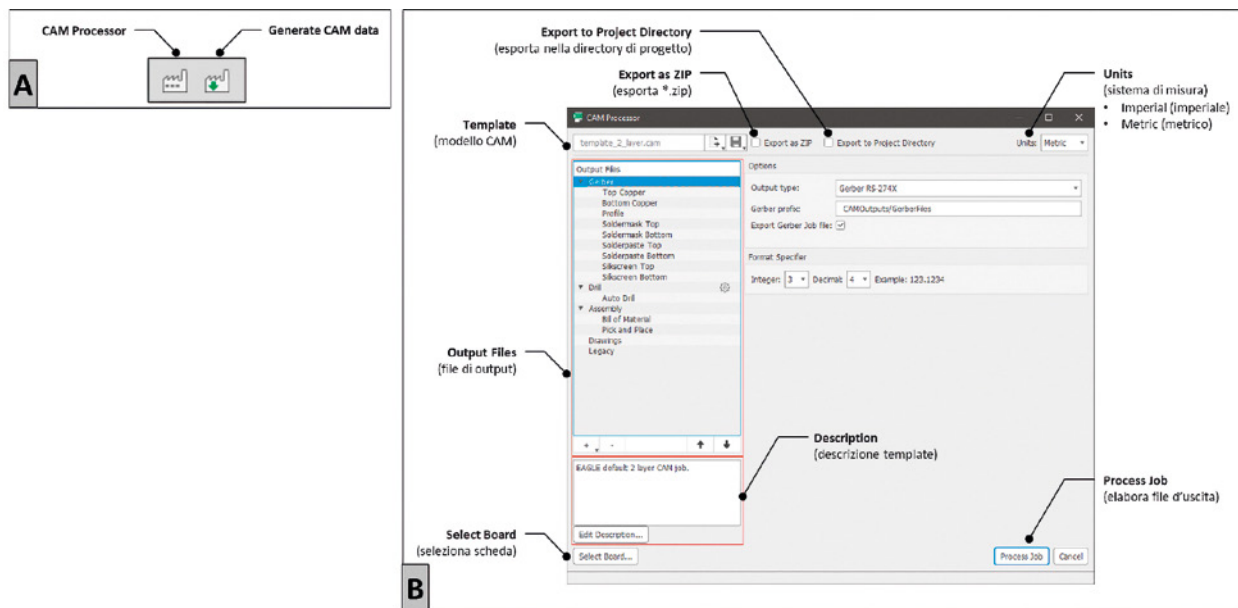


Fig. 26 Lo strumento CAM Processor.

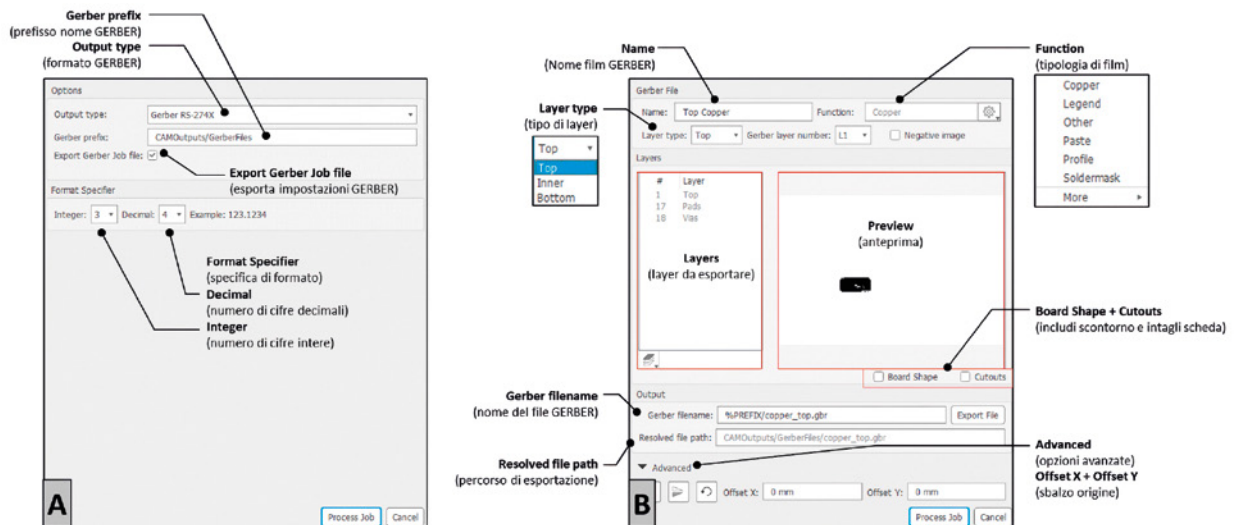


Fig. 27 La gestione dei GERBER.

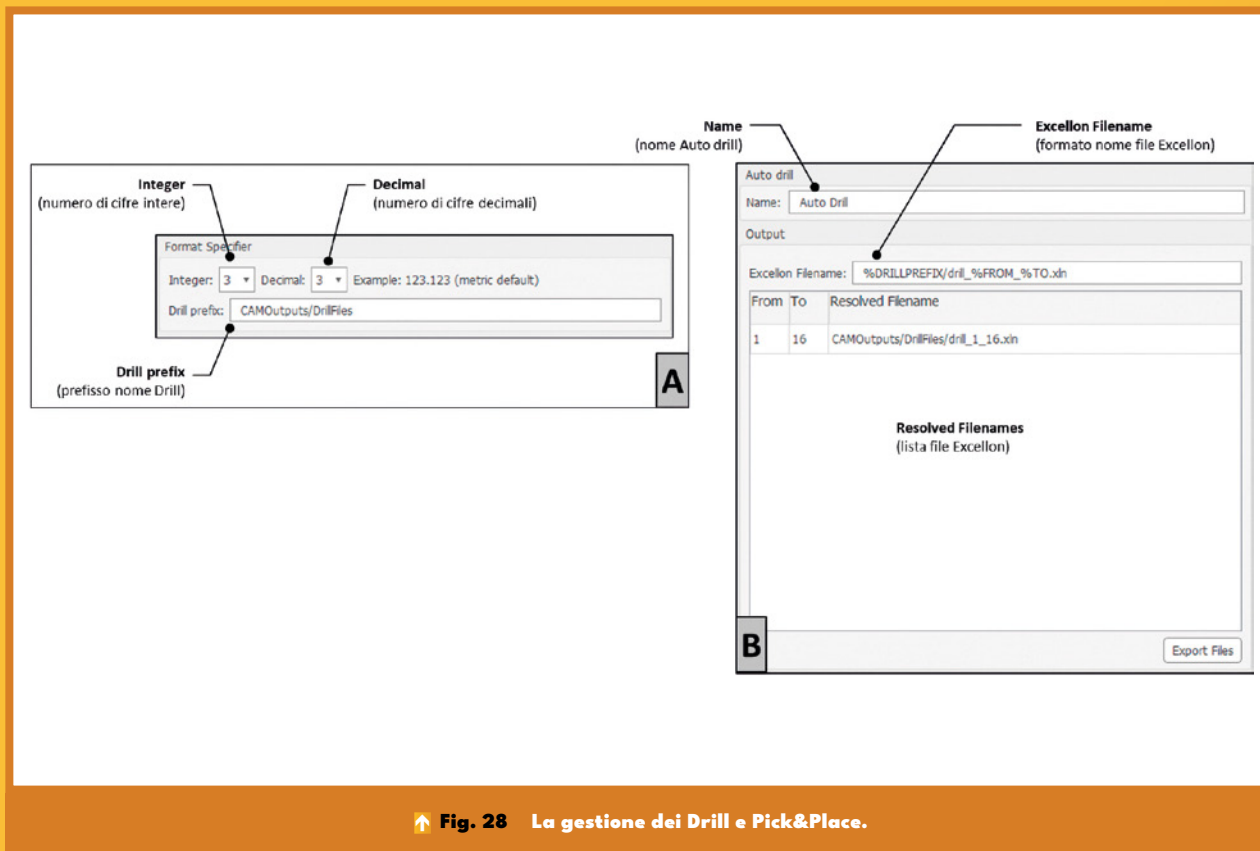


Fig. 28 La gestione dei Drill e Pick&Place.

file d'uscita. Questo modello è possibile modificarlo o caricarlo ed eventualmente salvarlo tramite i due pulsanti posti a destra della casella indicante il nome.

Subito a destra troviamo le opzioni di esportazione del lavoro come unico file ZIP e destinazione nella directory di progetto. Infine troviamo la casella combinata "Units" che permette di selezionare il sistema di misura, quindi l'unità di misura, per l'esportazione file. In pratica se si seleziona il sistema imperiale i file di output saranno quotati in pollici (in) o frazione di pollice, mentre se si seleziona il sistema metrico i file d'uscita saranno quotati in millimetri (mm) o frazione di millimetro. Subito in basso a sinistra è presentata una lista dei file di output; nell'esempio sottolineiamo principalmente i GERBER, i DRILL e lo ASSEMBLY che sono gli attori principali per la generazione di file di fabbricazione. Ancora più in basso troviamo la descrizione del modello utilizzato e, tramite il pulsante "Edit Description...", la possibilità di modificare o argomentare meglio la descrizione. Finora abbiamo descritto la finestra CAM processor nelle sue parti "fisse" ma, come accennato, il cuore della finestra si adatta dinamicamente alla selezione del file di output ed alle sue sotto-parti seppur, in linea generale, quasi mai è necessario modificarle; vedremo di seguito l'impostazione di queste parti al termine delle quali si possono processare i file d'uscita tramite il pulsante "Process Job".

IMPOSTAZIONE DEI FILE GERBER

Cominciamo con il descrivere meglio i file GERBER e la loro esportazione. Come abbiamo già accennato sono una serie di documenti in formato RS-274X (attualmente in ultima revisione); questa versione comprende i parametri di configurazione, le aperture, le coordinate x,y ed i comandi Draw & Flash della scheda nuda.

Come formato di file rimane comunque molto semplice perché scritto in ASCII quindi riproducibile da qualsiasi lettore di testo.

Si consiglia tuttavia l'apertura di questi file con un visualizzatore GERBER ("viewer") poiché, come è facile immaginare, orientarsi in un file di testo di questo tipo non risulta molto intuitivo. Nelle Fig. 27A e 27B sono descritti i dettagli delle impostazioni generali dei file GERBER e di un singolo elemento ("Top Copper").

Dunque i file GERBER vengono descritti con delle impostazioni generali come il formato (pocanzi accennato), un eventuale prefisso al file e il numero di cifre intere e decimali con le quali esprimere le quote. Nelle impostazioni generali è anche possibile esportare il file di impostazione dei GERBER. Inoltre per i GERBER, essendo composti da più file che indicheremo come "film", è possibile specificare le caratteristiche di ciascuno dei film (nell'esempio in figura è riportato il "Top Copper").

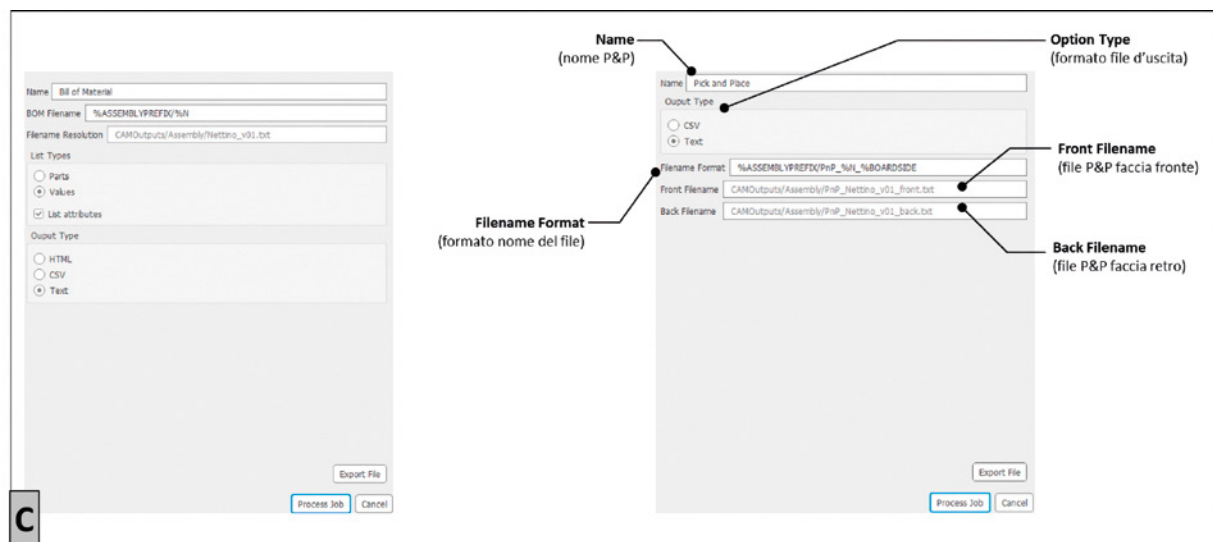


Fig. 29 Le impostazioni dei file di output "Assembly".

IMPOSTAZIONE DEI DRILL E PICK&PLACE

Concludiamo con la breve descrizione delle impostazioni di "Drill" e "Pick&Place" come fatto per i GERBER. Nelle Fig. 28A e 28B sono descritti i dettagli delle impostazioni generali e particolari per i singoli file Excellon (file Auto Drill). I file excellon rappresentano una sottoclasse del formato RS-274X e sono formattati in modo tale da interfacciarsi direttamente con le foratrici e fresatrici CNC.

In Fig 29 sono riportate le impostazioni dei file di output "Assembly". Le prime, a sinistra, riguardano la BOM ("bill of material") che abbiamo affrontato precedentemente durante la descrizione dello schema elettrico e dell'ambiente "schematic". A destra invece sono riportate le impostazioni per i file di "pick and place" (raccolgi e piazza) spesso abbreviati con P&P. Come possiamo vedere si generano due file uno per la faccia fronte (TOP) e l'altra per la faccia retro (BOTTOM). Nell'esempio ciascun nome del file è formattato secondo la stringa: %ASSEMBLYPREFIX/PnP_%N_%BOARDSIDE. Notiamo nella stringa tre segnaposto: %ASSEMBLYPREFIX, %N e %BOARDSIDE. Più in generale è possibile formattare il nome di qualsiasi file di output utilizzando i seguenti segnaposto:

- %N: è il nome del file "board".
- %NAME: il valore del campo "Name".
- %ASSEMBLYPREFIX: è il valore del campo della sezione "Assembly prefix" (file BOM e P&P).
- %DRILLPREFIX: è il valore del campo "Drill prefix" (file Drill).

- %PREFIX: è il valore del campo "Gerber prefix" (file GERBER).
- %BOARDSIDE: verrà sostituito con uno dei lati della scheda (front, back).
- %DATE e %DATEUTC: è la data corrente formattata come <<yyyymmdd>> (anno-mese-giorno).
- %TIME e %TIMEUTC: è l'ora corrente formattata come <<hhmmss>> (ore-minuti-secondi).

Una volta processate queste impostazioni tutti i file verranno salvati nelle opportune sottocartelle (dettate dai prefissi) e con la formattazione specificata.

CONCLUSIONI

Siamo dunque giunti al termine di questa puntata, partendo dai fondamenti di EAGLE abbiamo introdotto le librerie, esplorato l'ambiente schematic con le sue impostazioni e i file di output; abbiamo poi introdotto l'ambiente board fino ad arrivare alla generazione dei suoi file di output. Arrivati a questo punto siamo in grado di realizzare un progetto partendo da zero fino a generare i file di fabbricazione necessari per la prototipazione della scheda.

Rimane tuttavia un ultimo passaggio da affrontare, cioè la migrazione verso l'ambiente Fusion 360 che in futuro sostituirà l'attuale versione di EAGLE.

Come avremo modo di apprezzare nella prossima puntata conclusiva del corso, il passaggio è immediato, perché ritroveremo nel nuovo ambiente di sviluppo, tutti i vari strumenti analizzati sin qui.



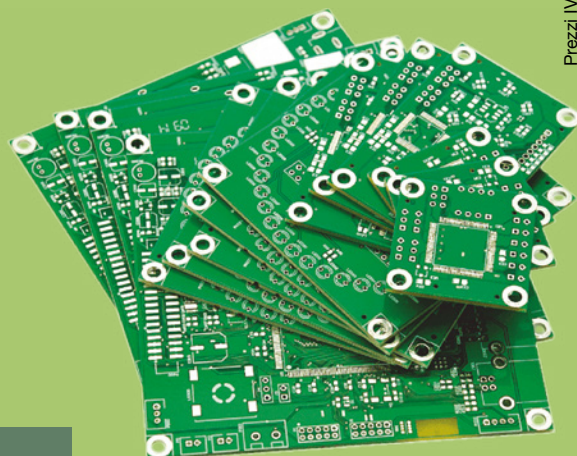
I tuoi PCB a partire da

1€

Con il servizio PCB di Futura Elettronica, potrai realizzare facilmente il circuito stampato del tuo progetto!

Vai su www.futurashop.it, inserisci nella ricerca il codice "PCBPRODUCTION" e carica il tuo Gerber scegliendo le dimensioni e le caratteristiche preferite.

Prezzi IVA inclusa.



cod. PCBPRODUCTION

COLORE SERIGRAFIA



BIANCA



NERA

La serigrafia nera è disponibile solamente per PCB bianchi e gialli.

COLORE SOLDER



VERDE



ROSSO



GIALLO



BLU



BIANCO



NERO

SPESSORE (mm)

mm



0.2

0.4

0.6

0.8

1.0

1.2

1.6

2.0

2.4

2.8

3.2

NUMERO DI LAYERS

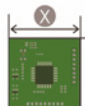
1 Layer

2 Layers

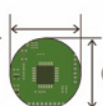
4 Layers

6 Layers

DIMENSIONI (cm)



Single Size



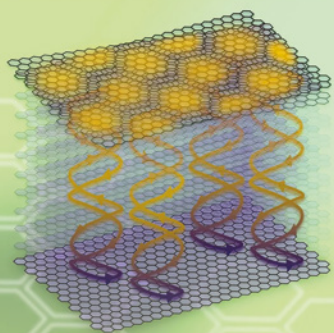
Pannel Size

- 5x5 - 5x10 - 5x15 - 5x20 - 5x25 - 5x30
- 10x10 - 10x15 - 10x20 - 10x25 - 10x30
- 15x15 - 15x20 - 15x25 - 15x30
- 20x20 - 20x25 - 20x30
- 25x25 - 25x30
- 30x30

ECCO QUALCHE ESEMPIO:

PCB con 1 layer
da 5x5 cm
spessore 1,6 mm
solder verde
serigrafia bianca
€ 1 cad.

PCB da 2 layer
da 10x10 cm
spessore 1,6 mm
solder verde
serigrafia bianca
€ 6 cad.



La grafite con le proprietà del grafene

Impilare due fogli di grafene disposti di traverso ha reso possibile la realizzazione di superconduttività sintonizzabile, memoria quantistica e nuovi stati della materia che coinvolgono gruppi di elettroni. Il comportamento dei sistemi siffatti, basati su grafene a doppio strato ritorto, emerge con un angolo di "torsione" tra il modello esagonale nei due fogli di appena 1,1 gradi. Scienziati negli Stati Uniti e nel Regno Unito hanno iniziato a esplorare i limiti di questi sistemi 2D sovrapposti, ipotizzando cosa accadrebbe combinando un materiale 3D (ad esempio la grafite) con uno spesso un atomo (il grafene). Così hanno scoperto che alcune delle proprietà del doppio strato persistevano anche all'interno di pile di grafite spesse e tridimensionali. I comportamenti osservati nella combinazione di grafite + grafene, consentono alla grafite tridimensionale di presentare il comportamento di un materiale bidimensionale. I team di ricerca dell'Università di Washington e dell'Università di Manchester hanno creato due configurazioni leggermente diverse; il primo ha sormontato una spessa pila di grafite con un singolo strato di grafene, con una leggera angolazione. Il team dell'Università di Manchester ha utilizzato una pila di grafite simile, applicata con un singolo strato bidimensionale di nitrato di boro. Entrambi gli approcci creano il cosiddetto super reticolo moiré. Sottoposti a forti campi magnetici, i superreticoli moiré subiscono notevoli trasformazioni a causa dell'aumentata spaziatura reticolare. Man mano che il campo viene sintonizzato, la conduttività del foglio oscilla periodicamente (oscillazioni Brown-Zak). I fenomeni descritti, caratteristici di una coppia di fogli 2D, emergono quindi anche nella grafite, che pure è tridimensionale.

www.washington.edu

Euclid: un occhio sul lato oscuro dello Spazio



È in pieno svolgimento la missione del veicolo spaziale scientifico Euclid dell'Agenzia spaziale europea (ESA) che è stato lanciato con successo dal Kennedy Space Centre in luglio e trasporta un telescopio ad alta precisione di 1,2 m di diametro e un modulo di payload che gli consentiranno di esplorare la composizione e l'evoluzione dell'Universo, compreso il ruolo della materia oscura e dell'energia oscura. Costruito da Thales Alenia Space per l'ESA, Euclid raccoglierà immagini ad alta risoluzione nelle lunghezze d'onda del visibile e dell'infrarosso. Entro sei anni di osservazione, coprendo più di un terzo dell'intero cielo, Euclid misurerà le forme e le distanze di oltre 1 miliardo di galassie. Euclid è un'ulteriore missione spaziale che ispira il mondo e che aiuterà l'umanità a comprendere la struttura e l'evoluzione dell'Universo. Euclid creerà una mappa della struttura su larga scala dell'Universo ed esplorerà come l'Universo si è espanso e come si è formata la struttura nel corso della storia cosmica, rivelando di più sul ruolo della gravità e dell'energia oscura. Euclid esaminerà anche gli effetti del "lente debole", un effetto che distorce le forme delle immagini di galassie lontane a causa della presenza di materia oscura tra la Terra e quelle galassie. In questo modo mapperà la distribuzione della materia oscura nell'Universo con una precisione senza precedenti. Il telescopio interamente in carburo di silicio (SiC) dovrà operare in condizioni estremamente fredde di 100 Kelvin (-170°C), si basa sull'esperienza leader mondiale di Airbus nella tecnologia del carburo di silicio

per lo spazio, già dimostrata in orbita sull'Herschel dell'ESA e Missioni Gaia. Euclid è una missione di "classe media" nel programma Cosmic Vision dell'ESA. Thales Alenia Space era il prime contractor industriale per il satellite e Airbus era responsabile del modulo di carico utile, di cui il telescopio è lo strumento principale. Euclid ha già raggiunto la sua orbita operativa attorno a Lagrange 2, quando inizieranno i test, con l'inizio delle operazioni complete previsto dopo tre mesi.

Nel mese successivo al suo lancio (avvenuto il 1° luglio) Euclid ha percorso 1,5 milioni di chilometri dalla Terra verso il punto di Lagrange Sol-Terra L2, raggiungendo la propria orbita di destinazione; la posizione dello specchio secondario, situato vicino all'ingresso del telescopio, è stata regolata dal produttore di telescopi Airbus Defence and Space France. Il sensore di guida fine di Euclid (FGS), che fornirà input su dove sta puntando Euclid utilizzerà le stelle per puntare il telescopio in modo estremamente stabile verso un bersaglio nello spazio, garantendo immagini nitidissime per il suo rilevamento. Il modello di luce innaturale rilevato dai sensori VIS è ancora oggetto di indagine. È in corso un'indagine approfondita per ottenere una panoramica completa dell'effetto di questo pattern sulle immagini VIS. Euclide sarà puntato sistematicamente in di-



650kW per la hypercar elettrica

Helix, azienda inglese produttrice di motori elettrici, ha presentato il suo ultimo powertrain basato sulla Scalable Core Technology (SCT), soluzione scalabile per adattarla a più auto. Il powertrain nasce dalla X-Division Helix (specializzata in motori customizzati) ed è stato chiamato REB. Basato sull'SPX177, il più potente motore per veicoli elettrici a batteria prodotto dall'azienda, pesa solo 28 kg ma eroga 650 kW di potenza continua e 398 Nm di coppia motrice: le prestazioni di una vettura di F1 o Formula E. L'insieme pesa solo 41 kg, compreso l'inverter che controlla il motore elettrico, il quale pesa 13 kg. Il motore è un trifase 2x, quindi la sua corrente è condivisa tra due inverter, un approccio necessario per soddisfare le richieste di corrente di fase a tensioni CC "normali" a questo livello di potenza estremamente elevato. Sia il motore che l'inverter hanno una densità di potenza estremamente elevata. Sei cavi ad alta tensione collegano l'inverter al motore, mentre un connettore LV trasporta i segnali di controllo. Nei test il motore ha raggiunto più di 700 kW; il massimo regime di giri raggiunto è 25.000 rpm. Rispetto al motore a combustione interna installato nella McLaren Artura, ad esempio, l'SPX177 pesa solo 1/7 a parità di potenza continua, anche se la batteria pesa più del serbatoio di carburante. Altri vantaggi includono la massa ridotta, che dà libertà di posizionamento per ottimizzare il baricentro e l'aerodinamica. Helix ha cercato un'architettura che riducesse al minimo le perdite e il calore generato, soprattutto ad alta velocità, e questo ha comportato un notevole cambiamento nel modo in cui il motore è stato avvolto, poiché la riduzione delle perdite di resistenza si traduce in una macchina a induttanza molto bassa, necessaria a minimizzare le interferenze dovute alla commutazione nell'inverter.

www.ehelix.com



verse angolazioni rispetto al Sole. Ad ogni orientamento VIS e NISP, così come FGS, acquisiranno immagini sensibili per scoprire quali angoli sono adatti per osservare l'Universo. Euclid dispone di un sistema di micropropulsione composto da sei propulsori ridondanti di micro-propulsione a gas freddo, fondamentali per garantire a Euclid un puntamento molto preciso e stabile e fornire immagini della massima qualità. Il team operativo di Euclid ha inviato

il comando per rilasciare l'antenna ad alto guadagno in banda K, parte del sistema di comunicazioni nello spazio profondo del veicolo spaziale. Questa antenna verrà utilizzata per inviare giornalmente più di 100 GB di dati compressi dal veicolo spaziale a una grande antenna nella rete di comunicazione nello spazio profondo Estrack dell'ESA. Euclid è ora il più grande trasmettitore di dati (in termini di data rate di circa 74 Mbps) dallo spazio translunare. Questo è circa 2-3 volte i tassi utilizzati dal James Webb Space Telescope che finora ha detenuto il trofeo.

www.esa.eu

Al per migliorare l'efficienza delle fuel-cell

Quantinuum, realtà specializzata nelle tecnologie quantistiche, collabora con BMW Group e Airbus nello sviluppo di un workflow ibrido quantistico-classico utilizzando quantum computer per simulare sistemi quantistici, concentrandosi sulle reazioni chimiche di ossidazione nei catalizzatori delle fuel-cell; in breve, si punta a modellare accuratamente la reazione di riduzione dell'ossigeno (ORR)

sulla superficie di un catalizzatore a base di platino, per elevare l'efficienza di conversione delle fuel-cell con membrana a scambio protonico. Il catalizzatore serve a scindere le molecole di idrogeno e ossigeno per far funzionare la pila a combustibile, processo lento (il che implica tempi di avvio relativamente lunghi per i veicoli a fuel-cell) e poco efficiente e per

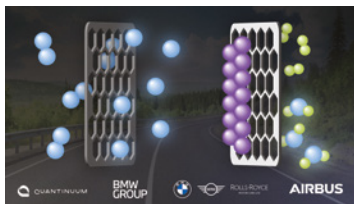
migliorarlo servono grandi quantità di platino, metallo costosissimo e non così diffuso. Utilizzando il computer quantistico serie H di Quantinuum è possibile accelerare la comprensione della ORR

e arrivare presto a nuovi materiali capaci di processi più rapidi ed efficienti.

Il team di ricerca spera che la comprensione della reazione ORR fornisca aiuti a identificare materiali alternativi

che potrebbero migliorare le prestazioni e ridurre i costi di produzione delle celle a combustibile. La modellazione accurata di reazioni chimiche come l'ORR è un compito non eseguibile con i computer classici, a causa delle proprietà quantistiche dei meccanismi chimici coinvolti.

www.quantinuum.com



La ISS cederà il passo a Starlab

Voyager Space e Airbus Defence and Space coopereranno alla costruzione di Starlab, una stazione spaziale commerciale progettata per succedere alla ISS, l'attuale stazione spaziale internazionale.

Voyager ha ricevuto uno Space Act Agreement (SAA) da 160 milioni di dollari dalla National Aeronautics and Space Administration (NASA) nel dicembre 2021 tramite Nanoracks, parte del segmento di esplorazione di Voyager. Parte del programma commerciale di sviluppo dell'orbita terrestre passa dalla NASA, questo SAA pone le basi per creare Starlab, una stazione spaziale a volo libero con equipaggio continuo per servire la NASA e una base di clienti globale di agenzie spaziali e ricercatori.

La missione del programma è mantenere la presenza umana continua e la leadership americana nell'orbita terrestre bassa (LEO). Oltre all'entità statunitense, Starlab avrà una filiale europea in joint venture per servire direttamente l'Agenzia spaziale europea (ESA) e le agenzie spaziali dei suoi stati membri.

Questo annuncio segue un'importante pietra miliare nello sviluppo di Starlab, la Systems Requirements Review (SRR) che definisce i principali sistemi spaziali, la prontezza tecnica e la capacità di soddisfare i requisiti di missione e sicurezza della NASA. Lo Starlab SRR è stato completato nel giugno 2023 in coordinamento con il team del programma di sviluppo LEO commerciale della NASA. Voyager Space offre soluzioni di esplorazione, tecnologia e difesa.

voyagerspace.com

Alle Canarie si volerà con l'idrogeno

Surcar Airlines, compagnia aerea che serve l'area delle Isole Canarie, ha deciso di scegliere i motori ecologici di ZeroAvia per i propri idrovolanti. ZeroAvia, pioniere dell'aviazione a idrogeno, ha siglato un accordo con Surcar Airlines per fornire i suoi sistemi di propulsione a idrogeno ZA600 ai velivoli dell'operatore aereo, che saranno usati nelle Canarie sulle brevi rotte da isola a isola.

L'obiettivo della partnership è quella di sostituire gradualmente i tradizionali mezzi a combustibili fossili per ridurre le emissioni di carbonio e migliorare l'impatto ambientale di una delle destinazioni turistiche più popolari in Europa.

Surcar prevede di installare il motore ZA600 sugli idrovolanti Twin Otter prodotti dall'azienda canadese De Havilland, con cui ZeroAvia ha siglato un MoU nel Dicembre 2021 per lo sviluppo congiunto di velivoli dimostrativi a idrogeno.

ZeroAvia ha completato con successo una serie di 10 voli prova sul prototipo Dornier 228 dotato di ZA600 a idrogeno presso la sua base britannica di Kemble, nel Gloucestershire. I motori elettrici vengono alimentati da fuel-cell PEM, che producono come residuo solamente acqua calda.

zeroavia.com



HDMI
↓
VGA

CONVERTITORE DA HDMI A VGA 1080P PLUG AND PLAY

Permette di convertire un segnale digitale HDMI (compatibile con HDMI1.3) in un segnale VGA (SUB-D) ad alta definizione con audio in uscita. Con questo convertitore, il computer, il lettore video portatile, le console di gioco e alcuni altri dispositivi video con uscita HDMI possono connettersi a schermi HD di grandi dimensioni oppure a proiettori HD con ingresso VGA.

€ 11,50

Cod. **BOXHDMITOVGA**

Convertitori di segnale

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

CONVERTITORE DA AV A HDMI PLUG AND PLAY

AV
↓
HDMI

€ 14,00

Cod. **BOXAVTOHDMI**

Permette di convertire un segnale audio/video CVBS (PAL e NTSC) in ingresso in un segnale HDMI in uscita. Con questo convertitore, il computer, il lettore video portatile, le console di gioco e alcuni altri dispositivi video con uscita CVBS possono connettersi a schermi con ingresso HDMI. L'uscita HDMI supporta una risoluzione fino a 1080P 60Hz.

VGA
↓
HDMI

CONVERTITORE DA VGA A HDMI PLUG AND PLAY

Permette di convertire un segnale audio e video VGA (SUB-D) in un segnale HDMI (tipo A) ad alta definizione con audio. Con questo convertitore, il computer, il lettore video portatile, le console di gioco e alcuni altri dispositivi video con uscita VGA possono connettersi a schermi HD di grandi dimensioni oppure a proiettori HD con ingresso HDMI. L'ingresso VGA supporta video fino a 1920x1080 a 60 Hz, l'uscita HDMI può supportare video fino a 1080P.

€ 11,50

Cod. **BOXVGATOHDMI**

CONVERTITORE DA HDMI AD AV PLUG AND PLAY

Permette di convertire un segnale HDMI in ingresso in un segnale video composito CVBS (PAL e NTSC) in uscita. Con questo convertitore, il computer, il lettore video portatile, le console di gioco e alcuni altri dispositivi video con uscita HDMI possono connettersi a schermi con ingresso CVBS. L'ingresso HDMI supporta una risoluzione fino a 1920x1080 a 60 Hz.

HDMI
↓
AV

€ 11,00

Cod. **BOXHDMITOAV**

SERVIZIO DI PROGETTAZIONE

L'area R&D di Futura Group studia, progetta e realizza prototipi per differenti settori utilizzando tecnologie innovative. Il team Futura Group è composto, oltre che dal personale dei nostri laboratori tecnici, anche da un numero elevato di collaboratori esterni, ciascuno specializzato nel proprio settore.



SERVIZIO DI REALIZZAZIONE PCB

Grazie al Servizio di realizzazione PCB potrai ricevere i tuoi circuiti stampati semplicemente caricando sul nostro sito il tuo Gerber e scegliendo le dimensioni e le caratteristiche preferite. È possibile ordinare schede a una singola faccia oppure multistrato, scegliendo anche il colore del solder.



Trasformiamo le tue idee in realtà!

FUTURA ELETTRONICA È IL TUO PARTNER DI RIFERIMENTO PER MOLTEPLICI SERVIZI. I NOSTRI TECNICI SONO A TUA DISPOSIZIONE PER FORNIRE TUTTE LE INFORMAZIONI E UNA VALUTAZIONE PERSONALIZZATA IN BASE ALLE TUE NECESSITÀ.

SERVIZIO TAGLIO LASER

Futura Elettronica offre la possibilità di realizzare mediante taglio laser i tuoi progetti in plexiglass (o metacrilato) con spessore massimo di 10 mm e con dimensioni massime lavorabili di 130x90 cm. Carica sul nostro sito il tuo file .DXF, scegli le dimensioni e le caratteristiche preferite.



SERVIZIO STAMPA 3D

Futura Elettronica mette a disposizione le proprie stampanti 3D FDM (fused deposition modeling), per chi intende realizzare un oggetto tridimensionale con dimensioni fino a 40x40x40 cm partendo da un modello 3D digitale. Carica sul nostro sito il tuo file .STL, scegli le dimensioni e le caratteristiche preferite.



**FUTURA
ELETTRONICA®**

www.futurashop.it

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

Scuola di saldatura!

**Stazione saldante
48 W con display**



cod. VTSSC40N

€ 94,00

**Stazione saldante/
dissaldante per
componenti SMD**



cod. VTSS230

€ 165,00

**Saldatore stilo
30 Watt**



cod. TP0055

€ 9,90

**Aspirafumi da banco
per saldatura – 23W**



cod. TP0065

€ 56,00

**Pompetta dissaldante
con riscaldatore**



cod. VTDESOL3

€ 11,80

**Pompetta
succhiastagno**



cod. VTD7

€ 3,60

**Stagno per saldatura
senza piombo
100GR – 0,56 mm**



cod. SOLD100G6SPB

€ 23,00

**Pulisci punta
per saldatore**



cod. WT45239

€ 6,00

**Pasta saldante
per SMD – Vol. 10cc**



cod. SMDPASTASALD

€ 8,90

**Pasta per saldare
50 g**



cod. WT45238

€ 3,90

**Pasta saldante per SMD
(senza piombo)- 50g**



cod. SOLDER50G

€ 37,00

**Anti flussante spray
200 ml**



cod. F02

€ 10,80

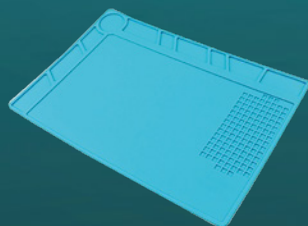
**Lacca isolante
spray 200 ml**



cod. V66

€ 11,50

**Tappeto in silicone per
saldatura 337x227 mm**



cod. AS10

€ 9,00

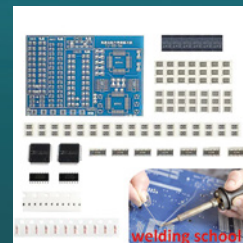
**Set di componenti per
imparare a saldare**



cod. YX286

€ 4,00

**Set di componenti SMD
per imparare a saldare**



cod. YX672

€ 5,00

ROME

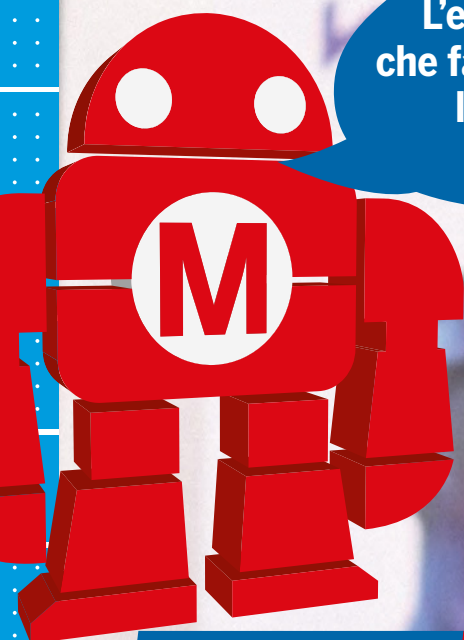
Maker Faire

THE EUROPEAN EDITION

An event powered by



L'evento europeo
che facilita e racconta
l'innovazione
tecnologica



20-22
OTTOBRE
2023

SCOPRI, INNOVA, CREA!
FIERA DI ROMA

MAKERFAIREROME.EU - #MFR2023

Promosso
e organizzato da



Main Partner



Gold Partner



Silver Partner



Bronze Partner



Technical
Partner

