

Elettronica In

www.elettronica.in.it

€ 7,50 - Bimestrale, anno XXIX - n. 272 - APRILE / MAGGIO 2023

160 PAGINE



Indicatore prezzi carburante

- ▶ Matter, il nuovo protocollo per la domotica
- ▶ Le novità di KiCad 7
- ▶ MKR-UNO Arduino carrier board
- ▶ Scheda FTR2040LoRa
- ▶ Mini driver LED



Ganimede.E12

- ▶ Minion Robot
- ▶ IoT controllo
- ▶ Saldare a stagno
- ▶ Riflettometria nel dominio del tempo
- ▶ Controllo smart per tapparelle



Affidabilità e precisione!

Generatore di forme d'onda arbitrarie 2 Canali-25 MHz

- Canali: 2
- Lunghezza record: 2 Mpts
- Range frequenza: 25 MHz
- Frequenza di campionamento: 300 MSa/s
- Risoluzione di tensione: 16 Bits
- Uscita forma d'onda standard: sinusoidale, quadra, triangolare, pulsante, noise, armoniche
- Uscita forma d'onda arbitraria: 160 tipi di forme d'onda arbitrarie, tra cui: exponential rise, exponential fall, ECG signal, Gauss, half vector, Lorentz, dual tone multi-frequency, DC voltage, etc.
- Noise (3 db): Larghezza di banda 100 MHz
- Onda Arbitraria: 1 μ Hz ~ 15 MHz
- Risoluzione: 1 μ Hz
- Precisione: ± 1 ppm, 18~28 °C



CON
TECNOLOGIA
DDS

cod. HDG3022B

€ 388,00

Hantek



Generatore di segnali con frequenzimetro 2 CH-60 MHz

- Canali: 2
- Frequenza max. di uscita: 60 MHz
- Frequenza di campionamento: 100 MSa/s
- Forme d'onda standard: sinusoidale, quadra, rampa, triangolare, impulso, noise, arbitraria, impulso di Lorenz, aumento/diminuzione esponenziale, CMOS
- Interfaccia: USB
- Display: 2.4" TFT LCD colori

cod. JDS6600

€ 189,00

JOY-IT

Generatore di funzioni e forme d'onda arbitrarie 2 Canali-25 MHz

- Display: TFT LCD a colori da 4,3 pollici
- Frequenza di campionamento: 125 MS/s
- Forme d'onda standard: sinusoidale, quadra, rampa, triangolare, impulso, rumore gaussiano, DC, arbitrarie
- Ampiezza del segnale di uscita: fino a 5 Vpp con impedenza di 50 Ω o fino a 10 Vpp in modalità ad alta impedenza
- Impedenza di uscita regolabile
- Tipo di modulazione: AM, FM, PM, ASK, FSK, PSK, PWM
- Modalità Sweep e Burst
- Distorsione armonica totale: 20 kHz, 1Vpp (valore tipico): <0,2%
- Interfaccia: USB



cod. TP388

€ 498,00

UNI-T



La tecnologia che unisce

L'IoT (o internet delle cose) è stato creato per migliorare la nostra esperienza quotidiana, rendendo casa e ufficio interattivi e facili da gestire, ma anche suggerendoci abitudini di vita più salutari. Tuttavia, fino ad oggi, ogni produttore di dispositivi IoT ha utilizzato un proprio protocollo di comunicazione, rendendo difficile l'interazione tra dispositivi di marche diverse.

La tecnologia Matter rappresenta una svolta in questo senso, creando un protocollo di comunicazione comune che prescinde dal produttore; così è possibile utilizzare una sola app per controllare tutti i dispositivi, invece di una per ogni produttore.

Matter è un esempio di come la tecnologia possa essere utilizzata per unire le persone e semplificare la loro vita

Matter è un esempio di come la tecnologia possa essere utilizzata per unire le persone e semplificare la loro vita, creando nel contempo un'interazione tra dispositivi eterogenei che altrimenti

sarebbero estranei gli uni agli altri e incapaci di "parlarsi". Ma non è solo una questione di facilità d'uso: la sicurezza dei dati degli utenti è un tema centrale nella tecnologia Matter, che si preoccupa di proteggere la privacy dei suoi utenti, dimostrando che la tecnologia può essere utilizzata in modo etico e responsabile. Matter è quindi un esempio di tecnologia che non si limita all'aspetto tecnologico, ma che raggiunge lo scopo per cui la tecnologia dovrebbe esistere: aiutare realmente le persone e farlo in tutte le loro attività. Ne sentiremo parlare sempre di più, ma intanto potete anticipare i tempi leggendo l'articolo che in questo numero abbiamo dedicato a Matter e all'impiego in Domotica.


Boris Landoni
 @boris.landoni@elettronica.in

In copertina:
 Il nostro indicatore di prezzi del carburante ci aiuta a trovare il fornitore più economico vicino a casa.



ON LINE

@elettronica.in

SI RINGRAZIA

Futura Elettronica
 Microchip
 Mouser

Anno XXIX n. 272
 APRILE / MAGGIO 2023

EDITORIALE

Direttore Responsabile

Boris Landoni

@boris.landoni@elettronica.in

Redazione

Stefano Garavaglia, Paolo Gaspari,
 Davide Scullino, Gabriele Daghetta

@redazione@elettronica.in

Grafica

Alessia Sfulcini

@alessia.sfulcini@elettronica.in

Ufficio Pubblicità

Monica Premoli (0331-752668)

@monica.premoli@elettronica.in

Ufficio Abbonamenti

Tel. 0331-752668

@abbonati@elettronica.in

Direzione, Redazione, Pubblicità

FUTURA GROUP srl - Divisione Editoriale

via Adige 11 - 21013 Gallarate (VA)

Tel. 0331-752668

Abbonamenti

Annuo 6 numeri Euro 39,00

Annuo 6 numeri digitale Euro 29,00

Le richieste di abbonamento vanno inviate a:

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

Tel. 0331-752668

oppure tramite il sito

<https://futuragroup.it/abbonamenti>

Distribuzione per l'Italia

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

via Bettola 18 - 20092 Cinisello Balsamo (MI)

Tel. 02-660301 Fax 02-66030320

Stampa

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



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

Prezzo di copertina Euro 7,50.

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

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

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

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

Tutti i contenuti della Rivista sono protetti da Copyright.

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

© 2023 Futura Group srl

Futura
 Group
 Edizioni

Contenuti

01

RUBRICHE

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

Progetto di Copertina

**66****11**

ARTICOLI

DOMOTICA

11 Matter, il nuovo protocollo per la domotica

Matter, il nuovo standard per la domotica che ha come obiettivo principale quello di far "collaborare" tutti i dispositivi connessi già presenti in una casa intelligente, unificandone tutti i protocolli.

IN VETRINA

17 Le novità di Kicad 7

Conosciamo l'ultima versione dell'ECAD prediletto dal mondo open, evidenziando le nuove funzionalità che porta con sé.

ARDUINO

25 MKR-UNO Arduino Shield

Adattatore che consente la compatibilità tra la serie di board MKR e gli shield di Arduino UNO.

RASPBERRY PI

35 Scheda FTR2040LoRa

La scheda FTR2040LoRa unisce le potenzialità del chip RP2040 di Raspberry Pi e del modulo RFM95W allo scopo di creare un end device LoRa/LoRaWAN in grado di comunicare con The Things Network e altri cloud simili.

LIGHTING

49 Mini driver LED

I LED sono diventati sempre più comuni nell'elettronica di consumo, ma per farli funzionare in modo efficiente e sicuro, è necessario utilizzare un driver LED.

DOMOTICA

Matter, la nuova domotica

**11**

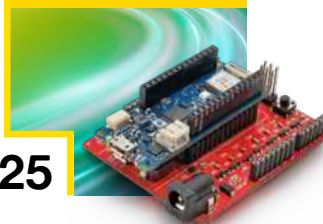
IN VETRINA

Le novità di KiCad 7

**17**

ARDUINO

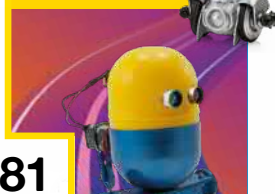
MKR-UNO Arduino shield

**25**

RASPBERRY PI

Scheda FTR2040LoRa

**35**

LIGHTING**Mini driver LED****49****NETWORKING**
Ganimede E.12**55****ROBOTICA**
Robot Minions**81****DOMOTICA**
Smart Roller Shutter**89****55****NETWORKING****GanimedeE.12**

Versatile ed economica scheda di sviluppo basata su ESP12, dotata di USB, WiFi, Bluetooth, slot mikroBus. Programmabile tramite IDE Arduino, Micropython o Javascript.

66**NETWORKING****Fuel price monitor**

Controllo attraverso ESP32 di un pannello touch interattivo che espone i prezzi dei distributori di carburante in una determinata zona.

81**ROBOTICA****Robot Minions**

Simpatico automa personalizzabile, con cui sperimentare la programmazione e il controllo del movimento.

89**DOMOTICA****Smart Roller shutter**

Permette il comando di tapparelle motorizzate mediante assistente vocale Amazon Alexa, Google assistant oppure tramite App; connesso in WiFi, è programmato con un firmware che si appoggia al server Sinric Pro.

105**MISURE****Riflettometria nel dominio del tempo**

Misuriamo l'impedenza caratteristica di un cavo coassiale e rileviamo le condizioni di adattamento, mediante un semplice hardware ed un oscilloscopio.

115**DIDATTICA****Saldare a stagno**

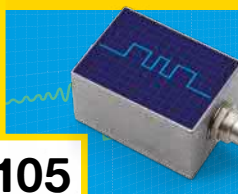
Impariamo a saldare e dissaldare i componenti elettronici, da quelli più tradizionali a foro passante fino agli SMD nei package più moderni.

129**IN VETRINA****IoT controllo**

Una suite di dispositivi di controllo e acquisizione da remoto gestiti tramite un'app attraverso un servizio NBIoT che si appoggia alla rete cellulare LTE.

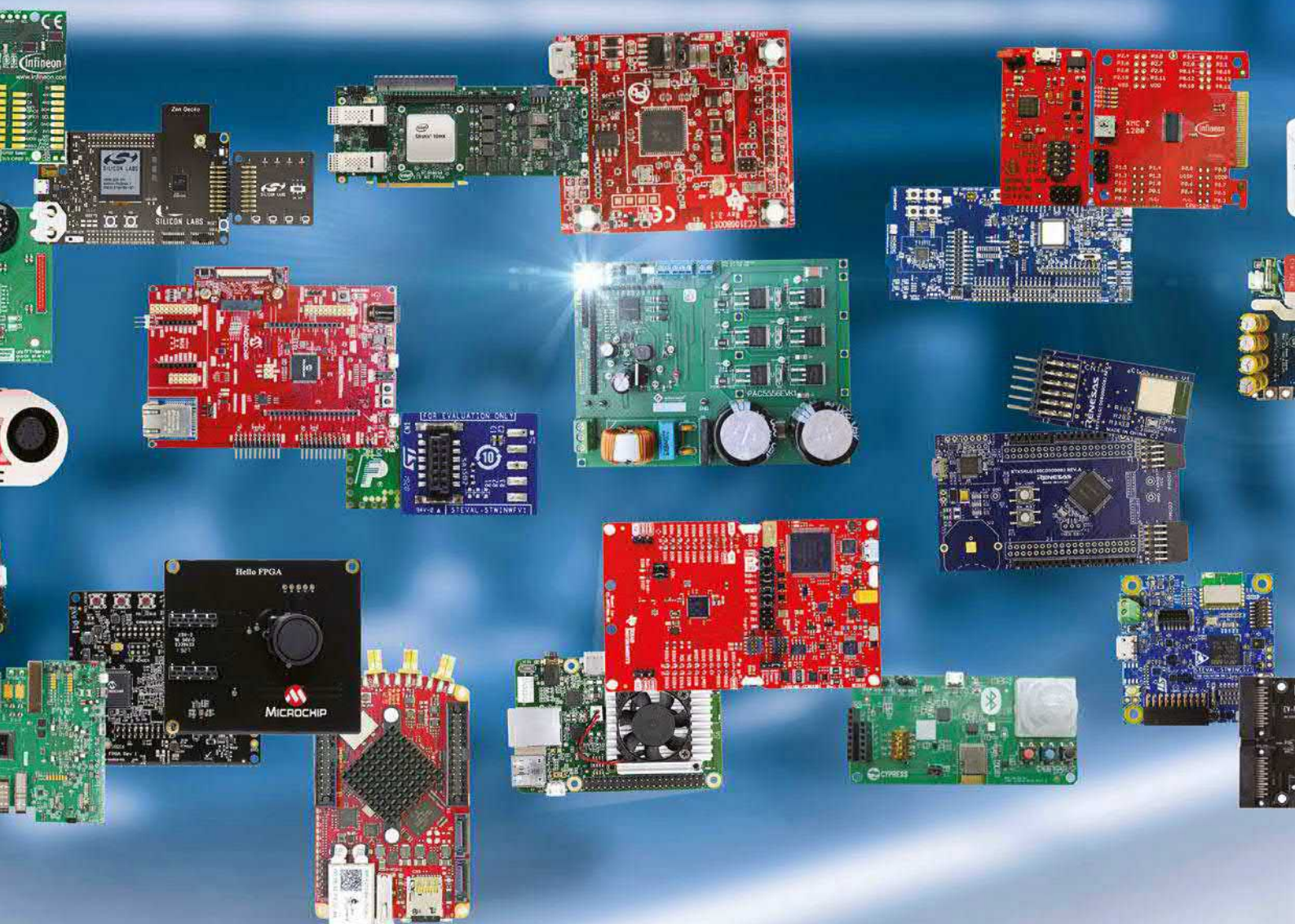
137**DIDATTICA****Impariamo ad utilizzare Eagle**

Cominciamo il nostro viaggio alla scoperta di un potente strumento di progettazione elettronica, che permette di coprire l'intero ciclo di sviluppo di una scheda elettronica. Prima Puntata.

MISURE**Riflettometria nel dominio del tempo****105****DIDATTICA****Saldare a stagno****115****IN VETRINA****IoT controllo****129****DIDATTICA****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

OTA per ESP32

Per i dispositivi che dispongono di una connessione dati (ad esempio gli smartphone, che contano sulla connettività cellulare) esiste da tempo una funzionalità chiamata OTA (Over-the-Air) che consente di aggiornare il firmware tramite la rete, invece di doverlo fare connettendoli fisicamente a un computer. La funzione OTA è disponibile anche per device e moduli WiFi come l'ESP32, da noi molto utilizzato nei nostri progetti.

Purtroppo sembra che la OTA disponibile per tale hardware e utile quando occorre aggiornare il codice su un ESP32, non sia facilmente accessibile.

Una possibilità consiste nell'eseguire la programmazione da remoto con l'ESP32 e il browser posti sulla stessa rete locale.

Vediamo come utilizzare, allo scopo, lo strumento OTA Web Updater, che però richiede di aggiungere il codice per l'OTA in ogni sketch che si vuole caricare: questo permetterà di utilizzare l'OTA all'occorrenza per poter caricare successivamente il codice tramite browser.

L'OTA Web Updater creerà un server web al quale potete accedere per caricare un nuovo sketch tramite browser.

Successivamente, bisognerà implementare, come già accennato, le routine OTA, in modo da poter fare gli aggiornamenti successivi in modalità wireless.

Per poter implementare tutto ciò, come prima cosa nell'IDE di Arduino bisogna installare l'add-on ESP32, il che automaticamente installerà la libreria ArduinoOTA.

Fatto ciò bisogna impartire dall'IDE il comando di menù File > Examples > ArduinoOTA > OTAWebUpdater; ciò determinerà il caricamento del codice corrispondente.

A questo punto bisogna identif-

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.

care nello sketch caricato le linee seguenti e scrivere le credenziali di accesso alla rete WiFi che si utilizzerà, al posto di ssid e password:

```
const char* ssid = "";  
const char* password = "";
```

Si deve quindi salvare lo sketch e caricarlo nella board ESP32. L'OTAWebUpdater per the ESP32 crea così un web server asincrono dove si potrà caricare il nuovo codice (quello di aggiornamento) senza

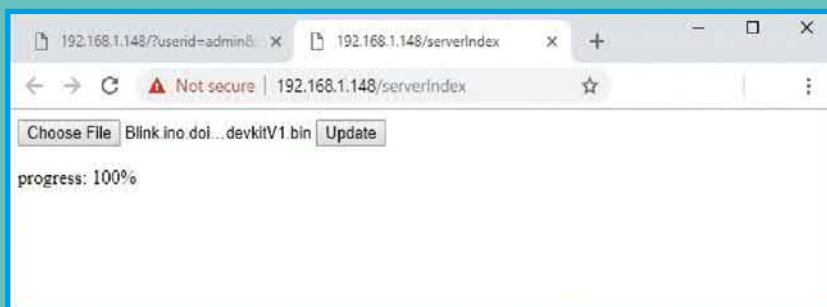
la necessità della connessione seriale. Dopo l'upload, bisogna aprire Serial Monitor con un baud rate di 115.200, premere il pulsante di enable del modulo ESP32 ed a quel punto si potrà ottenere l'IP address del modulo stesso, come mostrato nell'immagine in queste pagine nella cornice rossa.

Fatto ciò, bisogna aprire il browser su un computer affacciato sulla stessa rete e scrivere tale IP nella barra degli indirizzi, quindi premere Invio.

Apparirà la pagina web ESP32 Login Page nella quale inserire le credenziali di

↑ Esposizione dell'IP address della scheda ESP32 per l'OTA Updater.





↑ Caricamento del file .bin nel web server e aggiornamento dello sketch.

default che sono:

- **Username:** admin
- **Password:** admin

Potete cambiare queste credenziali nel codice, se volete. Dopo aver cliccato sul pulsante Login, verrete ridiretti alla URL del server web e nel browser si aprirà una nuova scheda che permette di fare l'upload del codice per l'aggiornamento, scegliendo il relativo file con il pulsante Choose File; i file importabili sono **.bin**.

Va ricordato che quando si carica un nuovo sketch via OTA, questo deve sempre essere completato con la porzione di codice per eseguire aggiornamenti futuri; la soluzione migliore è aprire lo sketch OTAServerUpdater e includervi il codice desiderato. Dopo aver copiato il codice, nell'IDE Arduino bisogna generare un file .bin: ciò può essere fatto dal comando di menù Sketch>Export compiled binary, ovvero con la scorciatoia da tastiera Ctrl+Alt+S, allorché nella cartella Sketch (comando Sketch>Show sketch folder) apparirà il nuovo file, che andrà caricato attraverso la pagina del web server con Choose file, quindi con Update. Dopo alcuni secondi, il codice dovrebbe essere caricato correttamente. Il LED integrato dell'ESP32 dovrebbe lampeggiare a conferma dell'avvenuto aggiornamento.

CAT-M, un link per IoT

Tra gli standard wireless impiegati nell'IoT spicca CAT-M, l'ultima generazione delle tecnologie di Mobile IoT, che utilizza la rete cellulare 4G complementando l'NB-IoT; la sua importanza e versatilità sta nell'ampia

diffusione che consente, grazie al fatto che come veicolo di connessione dati sfrutta la rete di telefonia cellulare 4G. Il CAT-M è supportato in Italia da Vodafone, garantisce ampia copertura a velocità e consumi di energia ridotti per le comunicazioni Machine-to-Machine (M2M), e 5G, con latenza minore di 10 millisecondi e banda ultra-larga per applicazioni in tempo reale o con contenuti in alta definizione. Con CAT-M i dati degli oggetti in movimento vengono trasferiti con tempi di latenza inferiori e velocità superiori rispetto al Narrowband IoT, caratteristiche che la rendono una tecnologia particolarmente adatta allo scambio di piccole e medie quantità di dati in mobilità, con una velocità massima di 800 kbps.

La tecnologia CAT-M abilita anche l'utilizzo dei dispositivi indossabili (i cosiddetti wearable device) e alcune importanti applicazioni di telemedicina, consentendo un monitoraggio continuo delle funzioni vitali e sanitarie garantendo la massima sicurezza nella trasmissione di dati grazie alla crittografia end-to-end dello standard LTE.

Megapixel e sensibilità delle fotocamere

Talvolta capita di fare un confronto tra fotografie e riprese effettuate tramite la fotocamera di uno smartphone e di rimanere perplessi di fronte alla constatazione che a parità di risoluzione della fotocamera stessa, le immagini ottenute non hanno la stessa qualità e non appaiono ugualmente definite. Questo accade perché la qualità di una fotografia non dipende solamente dalla risoluzione (i famigerati megapixel o

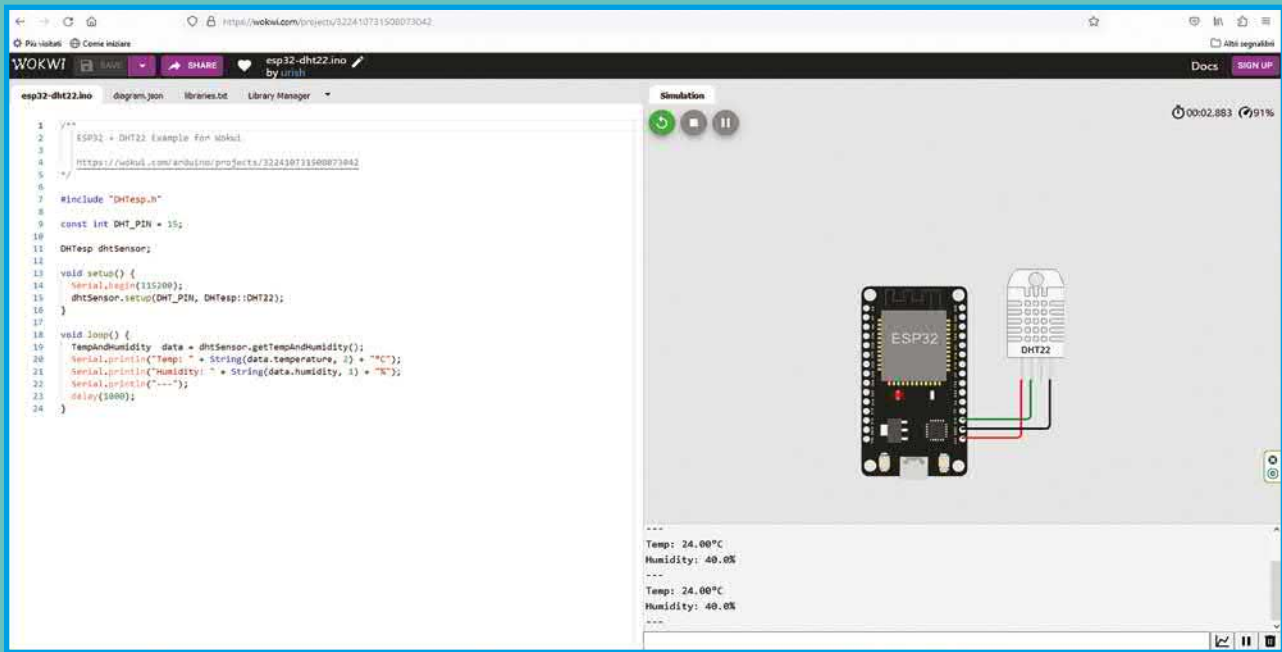
Mpx che dir si voglia) ma è condizionata da altri parametri, uno dei quali (il più importante) è la sensibilità, tipicamente espressa in lux.

La sensibilità rappresenta la capacità della fotocamera di riprodurre immagini di una certa qualità in base alle condizioni d'illuminazione della scena ripresa e corrisponde al minimo livello di illuminamento (i lux, appunto) per fornire un'immagine leggibile. Si potrebbe dire che la sensibilità determina a partire da che livello di illuminamento della scena i pixel della fotocamera iniziano ad attivarsi; per questa ragione, si può avere un sensore d'immagine con un numero e una densità molto grande di pixel, ma ottenere immagini che ingrandite e schiarite non hanno la qualità che ci si aspetta: perché alcuni o tanti di quei pixel, in condizioni di scarsa illuminazione non si attivano e ciò porta, di fatto, a riprese a risoluzione effettiva inferiore. Una maniera per elevare la qualità delle fotografie in caso di scarsa luminosità della scena consiste nell'allungare il tempo di esposizione, fermo restando che ciò può esporre all'effetto mosso. Sugli smartphone i valori ISO sono spesso predefiniti e molto alti (800/1600 ISO) e questo rende le foto molto luminose. È anche possibile modificare il valore predefinito, ottenuto dalla modalità "Automatico".

Simulare i progetti IoT basati su Arduino

L'Internet delle cose, meglio noto come IoT (Internet of Things) è uno dei trend più gettonato degli ultimi anni ed è per questo che un po' tutti i progettisti vogliono sviluppare qualcosa che rientri in tale concetto e che tutte le principali piattaforme e schede di prototipazione puntano ad avere un supporto in grado di renderlo possibile. Il pubblico di Arduino, che tipicamente è composto da maker e progettisti alle prime armi, potrebbe trovare arduo lo sviluppo di applicazioni IoT ed è per questo che sono disponibili strumenti come Wokwi, un simulatore on-line che permette di simulare un progetto IoT. Wokwi è un simulatore di circuiti elettronici che permette di sperimentare e migliorare l'esperienza in fatto di nell'IoT. Consente di simulare il comportamento di schede come Arduino ed ESP32 (perché i moduli ESP32





↑ Simulazione di un tipico progetto IoT basato su Wokwi.

possono essere programmati con l'IDE di Arduino) interfacciate a componenti e sensori. Permette, tramite un'interfaccia web accessibile da un browser puntando all'indirizzo wokwi.com, di iniziare a scrivere codice per applicazioni in pochi secondi, senza alcun rischio di danneggiare l'hardware come avverrebbe se si stesse testando uno sketch con un circuito materiale.

La piattaforma consente sia di creare un proprio circuito e farne scrivere e simulare il firmware, sia di partire da esempi consultabili scorrendo in basso nella home page alla sezione Quick Start Templates, ma anche di aprire e simulare o personalizzare progetti esistenti caricati dagli utenti del sito. È anche possibile partire da zero scegliendo la piattaforma (tra esse figura anche Raspberry Pi Pico) desiderata nella sezione Start from Scratch.

Di ciascun progetto creato o aperto è possibile consultare in tab dedicate, lo sketch e lo schema di cablaggio, il JSON, le librerie Arduino interessate, nonché gestire le librerie stesse, magari aggiornandole o editandole.

Una cosa interessante è che il sito Wokwi consente, per le schede ESP32, anche di creare progetti con codice scritto in Micropython, invece che con il classico linguaggio di Arduino. Per avviare la simulazione del progetto è previsto il pulsante di playback

(triangolino nel cerchio verde) in alto a sinistra nella sezione contenente lo schema di cablaggio del circuito; accanto al pulsante di playback ce n'è uno di arresto (quadrato) della simulazione ed uno di pausa. Ogni simulazione determina in basso nello schermo una serie di parametri, nel caso il progetto preveda l'acquisizione di grandezze fisiche da sensori, ovvero la comparsa nel display di testo scritto nel codice, oppure lo spostamento del braccetto di un servomotore in base alle istruzioni contenute nel codice.

La protezione UVLO

Circuiti come gli alimentatori e integrati regolatori di tensione o di carica delle batterie, quando al loro ingresso si presenta una tensione insufficiente a garantire i requisiti nominali (tensione e corrente) in uscita, vengono dotati di una protezione chiamata UVLO (Under Voltage Lock Out); si tratta di un circuito in grado di disattivare l'uscita per evitare malfunzionamenti dell'utilizzatore, problemi alla batteria connessa, ma anche utilizzati nei sistemi alimentati da batterie al litio per evitare che queste vengano scaricate oltre un valore che porterebbe al degrado delle loro celle.

Proprio in tema di batteria al litio, una protezione del genere è integrata nell'elettronica dei pacchi al litio montati nei computer notebook, per fare in modo che quando la batteria è troppo scarica il polo positivo venga sconnesso attraverso un MOSFET enhancement-mode posto in serie.

Anche sulle pagine di Elettronica In abbiamo pubblicato circuiti di protezione da sotto-tensione all'ingresso: un esempio è il circuito di disconnessione dell'utilizzatore quando la tensione di alimentazione è troppo bassa, pubblicato nel fascicolo n° 270; quest'ultimo è dotato di un trimmer per impostare la soglia di distacco.

Web Forum e supporto tecnico

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

Collegati a:

www.elettronica.in/webforum



ABBONAMENTO
ANNUALE



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



Visita: www.elettronica.in.it

6 NUMERI
A SOLI

€29

ANZICHÉ € 45,00

ABBONAMENTO ANNUALE PER RIVISTA DIGITALE

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

COME ABBONARSI

→ **on-line**

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

→ **e-mail**

scrivendo a
abbonati@elettronicaain.it

→ **telefono**

telefonando a
+39-0331-752668

NUOVO FORMATO 160 PAGINE CON PIÙ CONTENUTI

ABBONAMENTO ANNUALE PER RIVISTA **CARTACEA**

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

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

BUONO SCONTO DA 5€ valido 12 mesi
da utilizzare su **Futuranet.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

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.

MATTER, IL NUOVO PROTOCOLLO PER LA DOMOTICA

dell'Ing. **VINCENZO GERMANO**



Matter, il nuovo standard per la domotica che ha come obiettivo principale quello di far "collaborare" tutti i dispositivi connessi già presenti in una casa intelligente, unificandone tutti i protocolli.

Q

uando si parla di domotica si pensa sempre ad un sistema unificato che gestisce e controlla ogni aspetto del vivere abitativo, ad esempio un appartamento all'interno del quale sono stati installati dei dispositivi interconnessi che ne permettono il monitoraggio e la gestione da remoto (e non solo). Tali dispositivi devono svolgere funzioni differenti che vanno dalla gestione dell'illuminazione al controllo del riscaldamento in autonomia, dalla sicurezza dell'appartamento alla regolazione della continuità tra ambienti interni ed esterni, integrando l'esperienza high-tech della propria abitazione con estrema semplicità. Ciò viene fatto sia per fornire un nuovo concetto di benessere abitativo ma anche e soprattutto per poter ridurre i consumi e gli

 **Fig. 1**
Matter, The Foundation for Connected Things



sprechi che possono derivare da vari aspetti come il riscaldamento/raffrescamento, l'illuminazione e altro. Negli ultimi anni, il mercato della casa intelligente e quindi della domotica, è cresciuto in modo esponenziale, attraversando un periodo di enorme sviluppo, in un ambito che un tempo era dominato da utenti esperti di tecnologia, mentre a oggi la maggior parte delle famiglie possiede un dispositivo per la gestione di una casa intelligente. Tutto ciò è stato trainato da vari aspetti, tra cui il salto tecnologico dei dispositivi dettato da una sempre costante evoluzione dell'elettronica ma anche dalla continua riduzione economica per renderli più convenienti. Tuttavia, fino a qualche tempo fa, farli funzionare bene insieme continuava ad essere una sfida. Infatti, sia per gli appassionati di settore sia per gli utenti meno esperti che hanno deciso di addentrarsi nella domotica sperimentando come poter rendere il proprio alloggio intelligente, si saranno immediatamente accorti di quali possono essere gli straordinari vantaggi della compatibilità tra dispositivi e di quanto sia fondamentale il tema dell'interoperabilità. Questo perché, quando acquistiamo un qualunque dispositivo intelligente, non dovremmo preoccuparci della sua compatibilità all'interno del nostro sistema già installato, tuttavia oggi è una verifica che va fatta e il timore che il nuovo dispositivo non funzioni è giustificato. In quest'ambito, alla fine del 2019 ha visto la luce il progetto CHIP, acronimo anglosassone che sta per "Connected Home over IP", ovvero 'Casa connessa su IP' (dove IP sta per Internet Protocol), realizzato da colossi di settore che hanno spinto e accelerato la diffusione dei prodotti smart home come Apple, Google, Amazon e Zigbee Alliance. Ciò è stato fatto con l'intento di realizzare uno standard comune per consentire ai prodotti (ma anche ai servizi) per le case intelligenti, di dialogare facilmente tra di loro; con un conseguente obiettivo di facilitare i consumatori nel costruire il loro ambiente di domotica ideale. D'altro canto questo ha anche incentivato le aziende a sviluppare e produrre nuovi prodotti, permettendo di arrivare ad un sistema standardizzato in grado di superare le barriere tra dispositivi e produttori differenti.

In poche parole, questo significa che quando si andrà ad acquistare un dispositivo intelligente per la casa non ci sarà più la necessità di chiedersi se è compatibile con il sistema già installato, perché comprando un dispositivo certificato Matter si avrà la certezza che sarà sicuramente compatibile. Una bella svolta, ma vediamo più nei dettagli di cosa si tratta.

COSA È MATTER?

Il progetto CHIP (negli anni) ha poi assunto il nome di Matter (lanciato ufficialmente nel novembre 2022) **Fig. 1**, gestito dalla *Connectivity Standards Alliance* (CSA) – precedentemente nota come *Zigbee Alliance* –, esente da royalty, con l'intento di incoraggiare l'interoperabilità tra dispositivi e piattaforme. Perciò il progetto non ha cambiato lo scopo iniziale, ovvero un protocollo standard volto a ridurre la frammentazione esistente tra i vari produttori di dispositivi per la domotica (puntando alla totale interoperatività dei vari apparati domestici intelligenti interconnessi di produttori diversi). Perciò Matter si impone come nuovo standard per le case intelligenti e va a risolvere molti dei punti deboli di interoperabilità creati da software e hardware proprietari, oltre che alla necessità di una connessione costante ad Internet. Infatti, gli attuali dispositivi utilizzati nel settore della domotica (prima di Matter), fanno affidamento esclusivamente sulla connessione ad Internet e sul cloud per qualsiasi funzionalità, rendendoli di fatto inutili in assenza di connessione. Matter al contrario consente di funzionare anche offline senza la richiesta continua dell'accesso al cloud e ai vari servizi associati. Ciò implica anche una maggiore sicurezza per tutti i dispositivi, soprattutto in ottica di funzionalità sensibili come le serrature intelligenti o le telecamere di sicurezza, aggiungendo un ulteriore punto a favore nell'utilizzo di Matter. Inutile sottolineare gli enormi vantaggi del suo utilizzo, visto che questo standard Open-Source permette a tutti i dispositivi (certificati Matter) della smart home di poter dialogare tra di loro (anche e soprattutto a livello locale), senza aver bisogno di una connessione ad Internet o di un cloud, andando a sfruttare tecnologie come il Wi-Fi, il Bluetooth, l'Ethernet o il Thread. Con un tale vantaggio è facile immaginare come sarà possibile da ora in poi avere una casa intelligente semplicemente acquistando dispositivi certificati Matter (basati su questo nuovo standard) e un access point Wi-Fi. Essendo Open-Source, è possibile contribuire direttamente su Github, visto che è totalmente

aperto a qualsiasi contribuzione esterna e senza bisogno di pagare i diritti di utilizzo; infatti i privati o le aziende che vogliono partecipare al progetto sono libere di farlo e aggiornare i dispositivi senza costi aggiuntivi (tranne quelli per le certificazioni). Per verificare se i dispositivi hanno compatibilità con Matter lo si può fare facilmente verificando sul sito ufficiale quali sono i brand e i rispettivi produttori che adottano già o adotteranno questo standard; nel caso il prodotto o il brand ricercato non è presente nella lista non è detto che in futuro non offrirà la compatibilità ma semplicemente che al momento non offre il supporto (quindi non è compatibile). Per i consumatori, tutto ciò si traduce facilmente nel verificare se l'etichetta riporta la compatibilità con Matter o meno, visto che garantirà il loro funzionamento con altri dispositivi della rete in modo affidabile, permettendo all'utente di scegliere tra una gamma ampia di marchi con la certezza che questi funzionino/dialoghino all'interno della smart home in sicurezza e senza interruzioni.

Un ulteriore punto a favore, da non sottovalutare, è rappresentato dalla configurazione semplificata offerta, ovvero basterà collegare il nuovo dispositivo alla rete domotica affinché lui si possa connettere in automatico al resto della casa intelligente; di sicuro un passo in avanti determinante che permetterà anche agli utenti meno esperti di poter costruire una casa smart senza per forza essere esperti di settore.

Visti i principali vantaggi di Matter, scendiamo maggiormente in alcuni dettagli tecnici con

riferimento alla **Fig. 2**. Per essere precisi Matter è un *Application Layer* basato su IP, ovvero un layer superiore di comunicazione e non un intero standard di connessione (per semplificare, è un software che va a definire il linguaggio con cui i dispositivi dialogano tra loro). A oggi, essendo ancora in sviluppo, presenta un set di istruzioni ridotto, uno dei motivi per cui è in grado di funzionare solo con determinate tipologie di prodotti. Alcuni esempi possono essere gli interruttori, le lampadine o la sensoristica, tuttavia sta "maturando" per supportare prodotti più complessi come gli elettrodomestici e molto altro, che a differenza di prodotti semplici (es. la sensoristica) possono avere centinaia di funzioni.

MATTER E THREAD

Lo standard Matter non ha la pretesa di sostituirsi agli attuali sistemi proprietari utilizzati per il controllo e la configurazione dei dispositivi smart home, ma piuttosto vuole affiancarsi ad essi (andando ad utilizzare un linguaggio comune e comprensibile a tutti i prodotti e i brand), in modo tale da poter offrire soluzioni sia hardware che software. Infatti, con riferimento alla **Fig. 3**, è facilmente intuibile come il livello di trasporto scelto per Matter usa i protocolli già noti TCP e UDP, che a loro volta usano il livello di rete IP (Internet Protocol), mentre il livello di collegamento dati e il livello fisico (cioè i sistemi usati per trasmettere i dati), si affidano a tecnologie differenti come Ethernet, Wi-Fi, Bluetooth LE e Thread.

Considerando il Bluetooth, esso viene usato princi-



Fig. 2
Il layer
Matter, un
Application
Layer

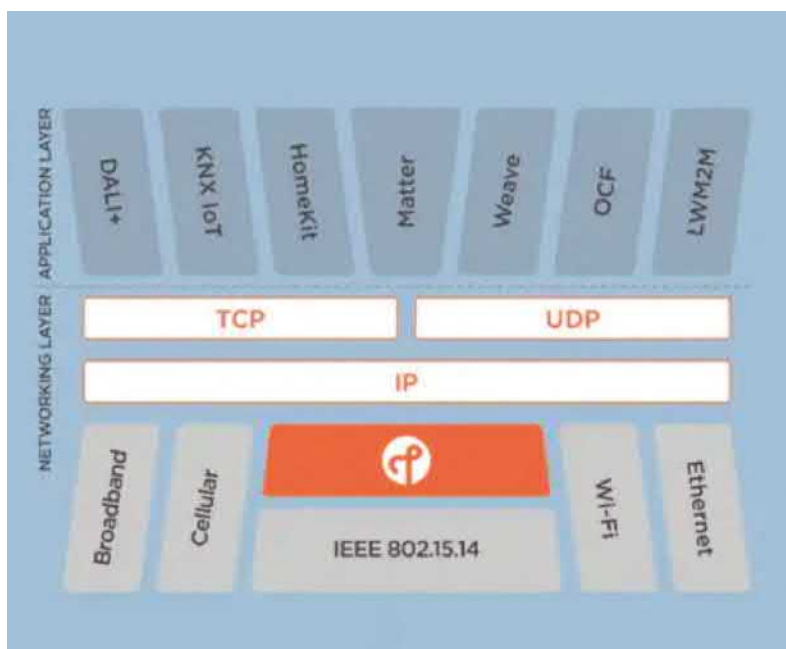


Fig. 3
Le tipologie
di protocolli

palmente per l'operazione di *commissioning*, ovvero l'inserimento del dispositivo all'interno della rete, grazie al quale il dispositivo viene "installato" nella smart home. Possiamo considerarlo come un'attività paragonabile al *pairing* delle cuffie Bluetooth, grazie al quale vengono associate ad un determinato dispositivo.

Questa funzionalità di *commissioning* è possibile anche svolgerla agevolmente via *QR Code*. Al contrario, sia l'Ethernet che il Wi-Fi vengono utilizzati soprattutto per i dispositivi che hanno necessità di una banda di trasmissione dati molto ampia, l'esempio più immediato può essere il video in streaming di una telecamera di videosorveglianza. Con riferimento alla **Fig. 4**, di sicuro la "novità" tra questi protocolli riguarda il Thread, che si colloca nella gestione della connessione dei dispositivi di rete a bassa potenza e che generalmente non richiede una banda di trasmissione ampia (visto che la dimensione dei dati inviati e ricevuti è contenuta). Viene utilizzato principalmente per interruttori, sensori, serrature, luci, prese e molto altro. Creato da Nest e altre aziende nel 2014 si basa sullo stesso protocollo di comunicazione wireless di Zigbee con connessione di rete "mesh" (a maglia, una singola rete Thread può ospitare oltre 250 dispositivi) a basso consumo energetico, viene generalmente utilizzato per dispositivi a batteria o che richiedono un'alta efficienza energetica.

A livello di tipologia di rete, la rete mesh permette di utilizzare i singoli dispositivi domotici come

elementi di rete "passivi" e "attivi", con i quali si possono ricevere comandi via radio ed attivare specifiche funzioni. Oltre a queste due tipologie, i dispositivi della rete possono anche essere degli "estensori di rete" (bridge) permettendo di ampliare la copertura dei segnali, ottimizzare i consumi energetici e ridurre la latenza (tre i comandi e le notifiche passa meno tempo). Perciò a differenza del protocollo Zigbee, Thread non richiede un hub dedicato, sia esso esterno o integrato direttamente nei dispositivi ma è sufficiente un "border router" (o anche Router Thread) ovvero un circuito elettronico integrato dedicato in qualsiasi dispositivo della rete.

Come si nota dalla **Fig. 4**, Thread può supportare più border router in modo tale da creare una ridondanza anche in caso di guasti o malfunzionamenti, offrendo molti vantaggi rispetto a protocolli e tecnologie già adottate nella Smart Home come Bluetooth, Wi-Fi e Zigbee.

Da quanto detto possiamo concludere che Thread opera su un livello hardware più che un livello software come accade per il Matter. Ciò significa che deve essere necessariamente previsto in fase di progettazione e industrializzazione del dispositivo che sarà border router e non semplicemente aggiunto in un secondo momento via software (senza la necessaria predisposizione di un hardware dedicato). Questo implica che solo i dispositivi già commercializzati che hanno a bordo l'hardware potranno essere aggiornati a Thread mentre gli altri non lo potranno essere. Tuttavia il mancato hardware di Thread non impedisce a questi dispositivi domotici comunque di supportare Matter, dato che rappresenta il "linguaggio comune" a livello software e quindi aggiornabile/utilizzabile grazie al protocollo IP mediante standard già affermati e universalmente implementati come il Wi-Fi e il Bluetooth.

LA SICUREZZA PRIMA DI TUTTO

Man mano che la domotica connette sempre più dispositivi e sistemi tra loro e le case intelligenti continuano ad evolversi, aumentano di pari passo i rischi di attacchi informatici che suscitano preoccupazioni da parte degli utenti finali e ne limitano l'adozione. Basti pensare ad attacchi DDoS (Distributed Denial of Service), violazioni dei dati e della privacy con potenziale danno agli umani e molto altro. Tuttavia proprio per superare questo ostacolo, Matter è stato concepito con massimi livelli di sicurezza e privacy che sono stati considerati come principali chiavi del design; infatti sono

state implementate tra le altre cose, comunicazioni sicure, autenticità dell'identità e connessioni con dispositivi comprovati e controllo dell'accesso. Questa sfida legata alla sicurezza e alla privacy ha di fatto richiesto l'implementazione delle migliori pratiche di sicurezza su larga scala a 360 gradi, andando a curare aspetti chiave come:

Produzione sicura: nella fabbrica o nella catena di approvvigionamento, i dispositivi possono essere soggetti a una varietà di attacchi come l'iniezione di codice dannoso;

Operazioni: una volta all'interno della rete i dispositivi sono suscettibili a una vasta gamma di attacchi remoti ma anche in alcuni casi ad attacchi fisici, perciò le comunicazioni devono essere crittografate e autenticate per proteggere dagli attacchi remoti/locali una volta installati; per cui proteggere dagli attacchi significa garantire che i dati arrivino nella destinazione prevista in una forma riservata, autenticata e inalterata;

Manutenzione: il processo di aggiornamento del software per i dispositivi deve essere sicuro per evitare l'inserimento di software non autorizzato, inclusi aggiornamenti del firmware e delle credenziali. Un processo di aggiornamento sicuro aiuta a prevenire il caricamento di malware o firmware

non autorizzato;

Regolamenti sulla sicurezza: le iniziative di regolamentazione domotica per la sicurezza e la privacy differiscono da paese a paese e continuano a evolversi, perciò i dispositivi intelligenti devono essere progettati con funzionalità dedicate per poter supportare queste modifiche.

CONCLUSIONI

Arriviamo a termine di questo articolo all'interno del quale abbiamo visto come la rivoluzione del nuovo standard Matter è appena cominciata, una nuova epoca per la domotica, con la promessa di migliorare la vita delle persone e soddisfarne le esigenze tramite strumenti in continua evoluzione. Di sicuro il supporto agli altri protocolli oggi disponibili e alle attuali tecnologie continuerà ad esserci, perciò si potranno ancora utilizzare i vecchi dispositivi senza la necessità di sostituire per intero il sistema costruito negli ultimi anni.

Per quanto è solo una questione di tempo prima che gli utilizzatori si convincano ad abbracciare il nuovo standard Matter per risolvere una volta per tutte i problemi di compatibilità affrontati negli ultimi anni, ma anche godere di tutti i vantaggi di questa nuova tecnologia. □

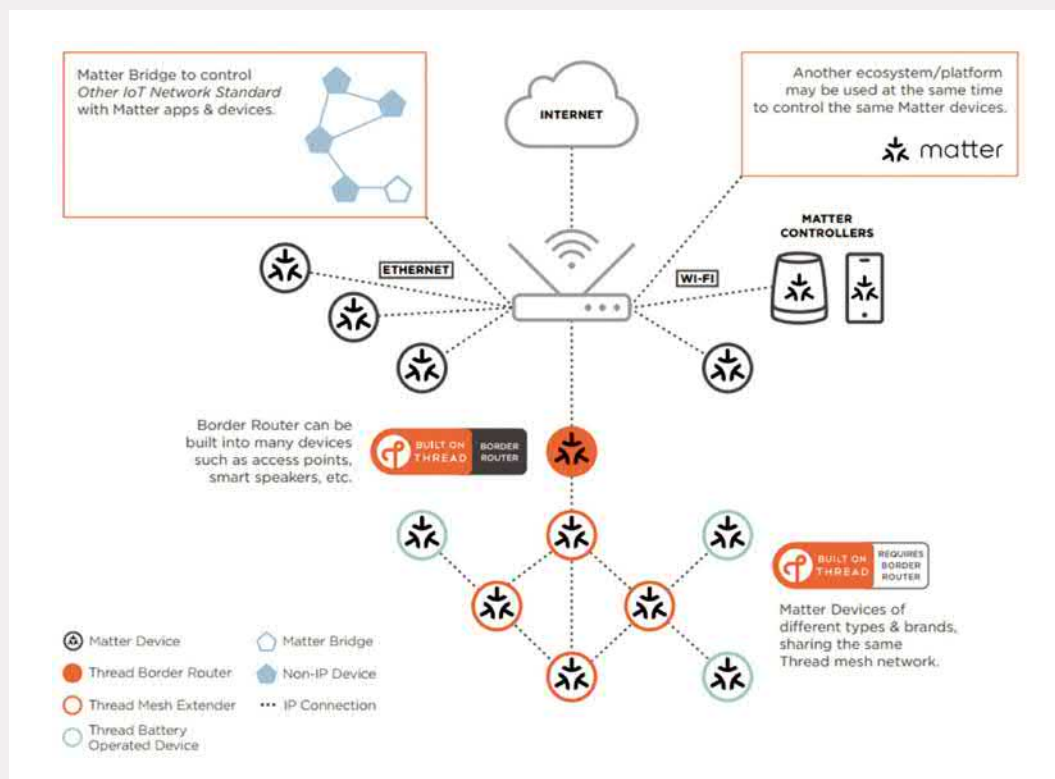


Fig. 4
Matter e Thread, una possibile rete.



Rilevamento ad Alta Precisione per applicazioni moderne

Famiglia di Microcontroller AVR[®] EA

I Microcontroller AVR[®] EA semplificano l'integrazione di controllo intelligente nei sistemi embedded con sensori analogici. Gli MCU, appositamente costruiti, offrono un ADC differenziale a 12 bit (con Programmable Gain Amplifier integrato), un DAC a 10 bit e un comparatore on-chip per un'incredibile flessibilità nell'interfacciamento con i componenti analogici.

In combinazione con la potente CPU AVR e le periferiche digitali "core independent", la famiglia AVR EA consente di creare progetti compatti con flessibilità integrata e costi inferiori di sistema.

Caratteristiche Salienti

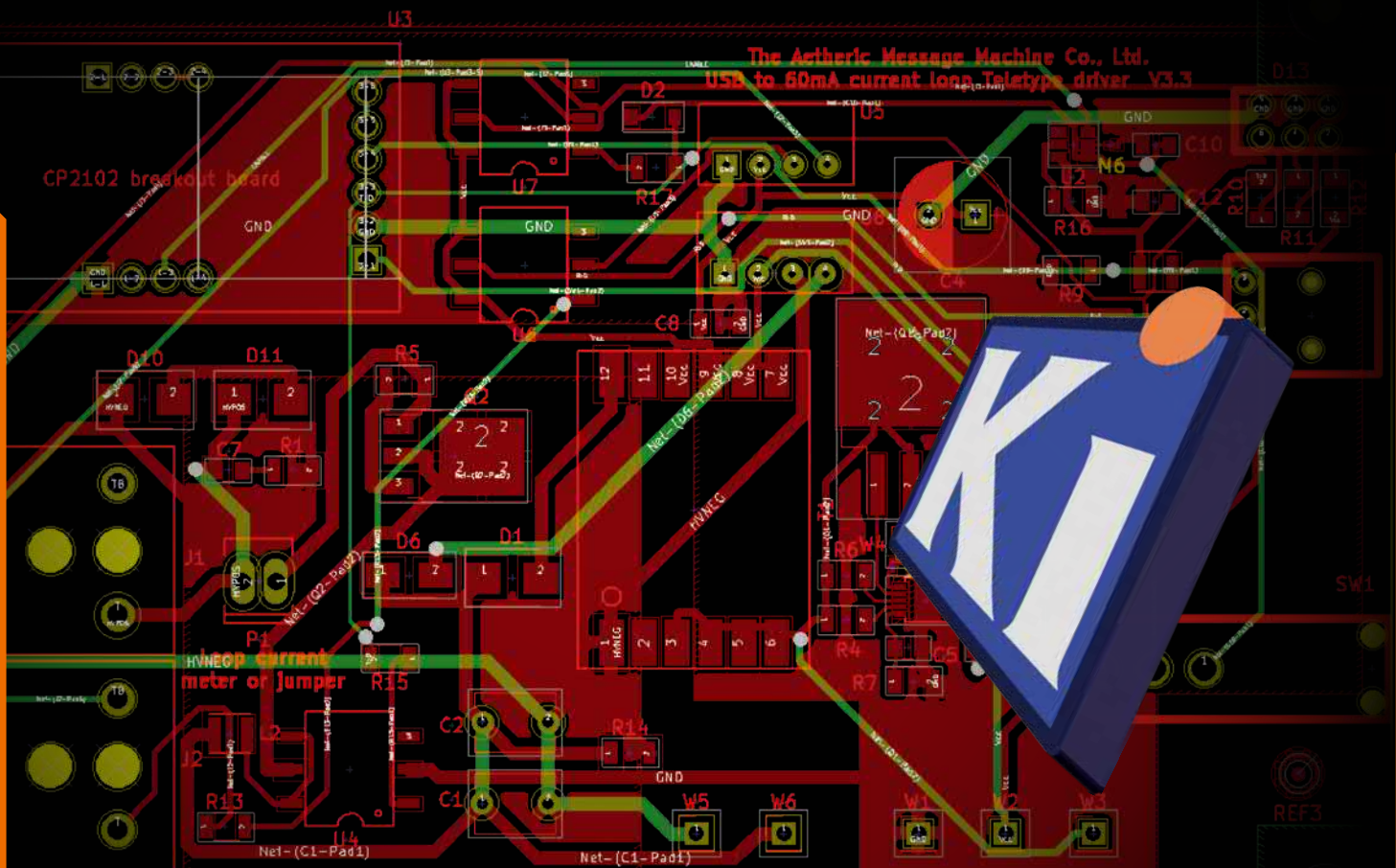
- CPU AVR fino a 20 MHz
- Memoria flash da 16 64KB con sezione Read-While-Write (RWW) dedicata
- 2- 6KB SRAM
- 512 B EEPROM
- 28 a 64 pin nei package SPDIP, SSOP, TQFP e VQFN
- Temperature di esercizio Standard ed Estesa: da -40°C a +85°C e da -40°C a +125°C



microchip.com/AVREA



Il nome e il logo Microchip e il logo Microchip sono marchi industriali registrati di Microchip Technology Incorporated negli U.S.A. e in altri Stati. Tutti gli altri marchi appartengono ai loro rispettivi titolari.
© 2023 Microchip Technology Inc. Tutti i diritti riservati.
MEC2478A-ITA-02-23



KICAD 7

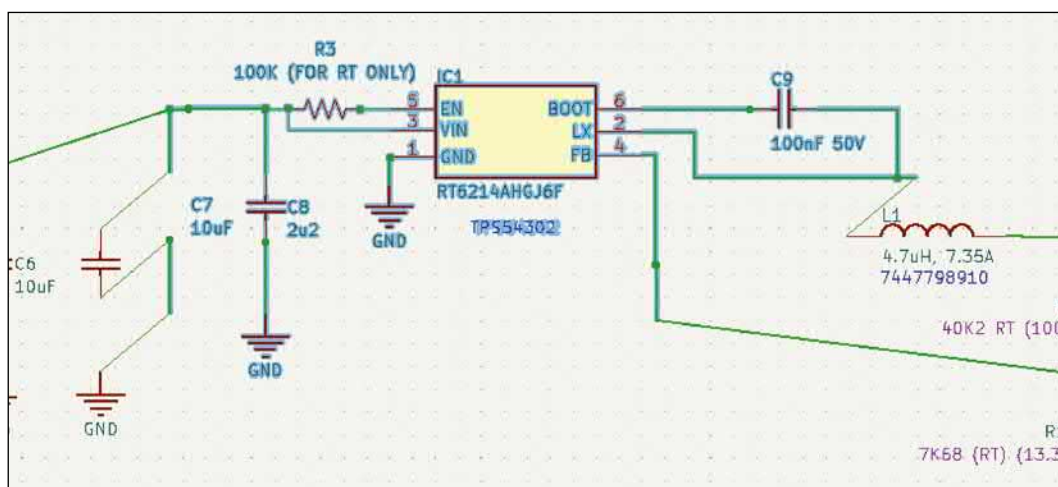
dell'Ing. **VINCENZO MENDOLA**

Conosciamo l'ultima versione dell'ECAD prediletto dal mondo open, evidenziando le nuove funzionalità che porta con sé.

Negli ultimi anni i software open source hanno raggiunto un livello qualitativo e di maturità davvero notevole; tra questi, a chi come noi è interessato al mondo dell'elettronica, è di grande interesse KiCad, un insieme completo di applicativi per tracciare schemi elettrici, realizzare i layout delle schede elettroniche e per visualizzare i file GERBER (ossia i file di produzione) corrispondenti. La storia di questo incredibile tool di progettazione elettronica inizia più di 30 anni fa: la prima release risale al 1992 e fu sviluppata da Jean-Pierre Charras ed ora il suo sviluppo è affidato ad un team dedicato chiamato "KiCad Development Team". Il nome di questa squadra di sviluppatori deriva dalle prime lettere di un'azienda di un amico del suo ideatore, che

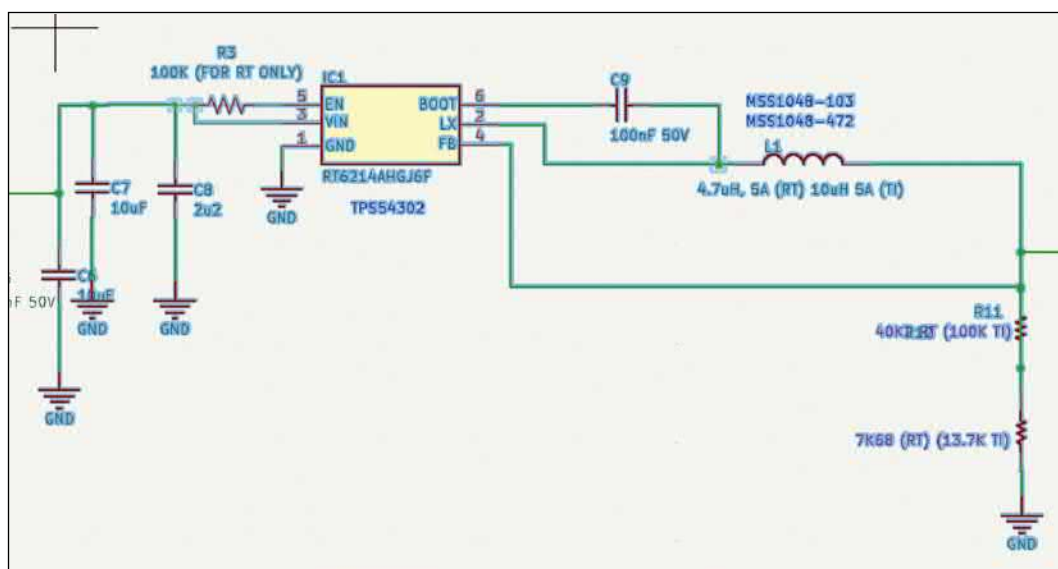
➔ Fig. 1

La precedente funzione di spostamento dei componenti manteneva le connessioni, ma disordinate.



➔ Fig. 2

Connessioni ottenute con lo spostamento del componente nella versione 7.



sono "Ki"; trattandosi di un software di progettazione assistita da computer, a queste due lettere è stato aggiunto "Cad". Ecco nascere, così, il nome KiCad.

PRESENTE E FUTURO DI KICAD

Come tutti i software di progettazione, dovendo tenere da conto l'evoluzione tecnologica e il conseguente cambiamento o incremento delle esigenze dell'utenza, anche KiCad viene costantemente aggiornato.

Il 12 febbraio 2023 è stata rilasciata la versione 7.0, che introduce molte novità e miglioramenti rispetto alla precedente versione 6, uscita appena un anno fa, esattamente il 25 dicembre 2021. Come dichiarato dallo stesso team che ha contribuito allo sviluppo della versione 7, questa

presenta grandi miglioramenti e nuove funzionalità rispetto alla precedente versione 6. Gli sviluppatori hanno dichiarato di voler continuare lo sviluppo di KiCad con l'attuale cadenza, rendendo disponibile una nuova major release all'anno, anticipando fin d'ora la disponibilità della nuova versione 8 nel 2024. Anche la politica di supporto rimarrà la stessa: prendendo come esempio l'attuale versione 7, la versione 6 non verrà più aggiornata, mentre la 7 verrà mantenuta e i bachi che dovessero emergere saranno risolti fintantoché non verrà rilasciata la successiva versione, che sarà la 8.

LICENZA D'USO

Come abbiamo accennato, KiCad è un software gratuito, il cui codice sorgente viene rilasciato sotto licenza GNU General Public License (GPL) version

3 o superiore; ci possono essere parti di codice di terze parti rilasciate sotto licenza MIT o altre, per i cui dettagli si rimanda al codice sorgente stesso. Per quanto riguarda le librerie di KiCad, queste sono rilasciate con licenza CC-BY-SA 4.0, la quale permette un utilizzo libero delle stesse, per cui è possibile usarle anche per attività commerciali e per applicazioni “closed-source” ed il loro utilizzo non comporta la necessità di condividere il lavoro fatto o di rilasciarlo con la stessa tipologia di licenza.

A chi fosse interessato ad un utilizzo professionale di KiCad o fosse semplicemente curioso, consigliamo di leggere con attenzione la documentazione ufficiale, nella quale vengono forniti tutti dettagli relativi alle licenze delle varie parti di KiCad, compresa la documentazione ed il sito stesso, anch'essi rilasciati sotto licenza.

VERSIONI PRECOMPILE

Per la maggior parte degli utenti, che non sono interessati ai sorgenti ma direttamente ai file binari precompilati, nella pagina di download del sito ufficiale sono disponibili i file per Windows, MacOS, Ubuntu, Debian, Fedora e altre distribuzioni Linux tra le più diffuse; ma non solo, perché ci troviamo anche i file per FreeBSD e Raspberry PI OS. Quindi possiamo affermare che KiCad punta a tutte le piattaforme che toccano, oltre al mondo professionale, quello dei Maker.

NOVITÀ VERSIONE 7 EDITOR DEGLI SCHEMI

Veniamo ora alle novità introdotte, partendo dall'editor di schemi elettrici. Come si vede chiaramente

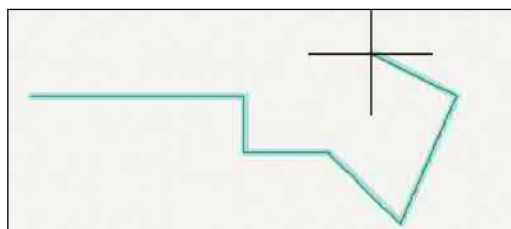


Fig. 3
Modalità di disegno dei collegamenti a 45°.

dalla **Fig. 1**, nella versione precedente alla 7, il trascinamento di un componente o di un insieme di componenti, comportava lo spostamento anche delle connessioni collegate alla parte interessata allo spostamento, che avveniva in maniera, per così dire, “irregolare”: KiCad si preoccupava solamente di mantenere intatte le connessioni ma non di mantenerle nella forma in cui erano e si venivano a creare angolazioni varie, obbligando il progettista a sistemarle una ad una; ora questo non è più necessario, in quanto nella versione 7 viene mantenuta l'ortogonalità delle stesse, velocizzando e semplificando il lavoro (**Fig. 2**).

Sempre riguardo alle connessioni, nella versione 7 è stata aggiunta una modalità per tracciare i fili (o “wires” in inglese) a 45°, oltre alla consueta possibilità di mantenere angoli retti o di tracciare connessioni con qualunque angolo; si può passare da una modalità all'altra, in maniera ciclica, tramite la combinazione di tasti Shift + Spazio (**Fig. 3**). È stato aggiunto il supporto alla funzionalità “Do Not Populate” (“DNP”, cioè non montare), per poter esplicitare direttamente nello schema elettrico quei componenti che vogliamo che siano chiaramente identificati come da non montare: KiCad li disegnerà in grigio e verranno barrati con una croce

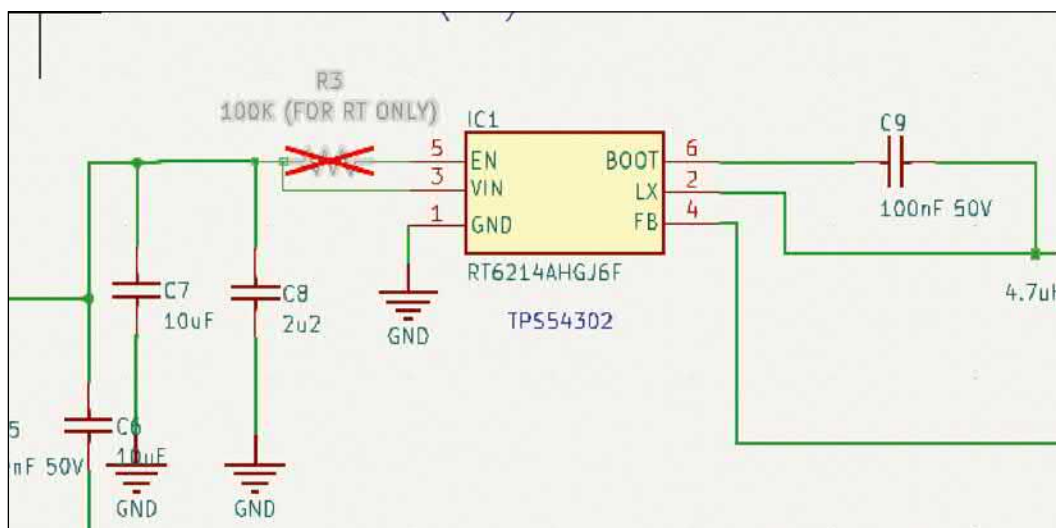
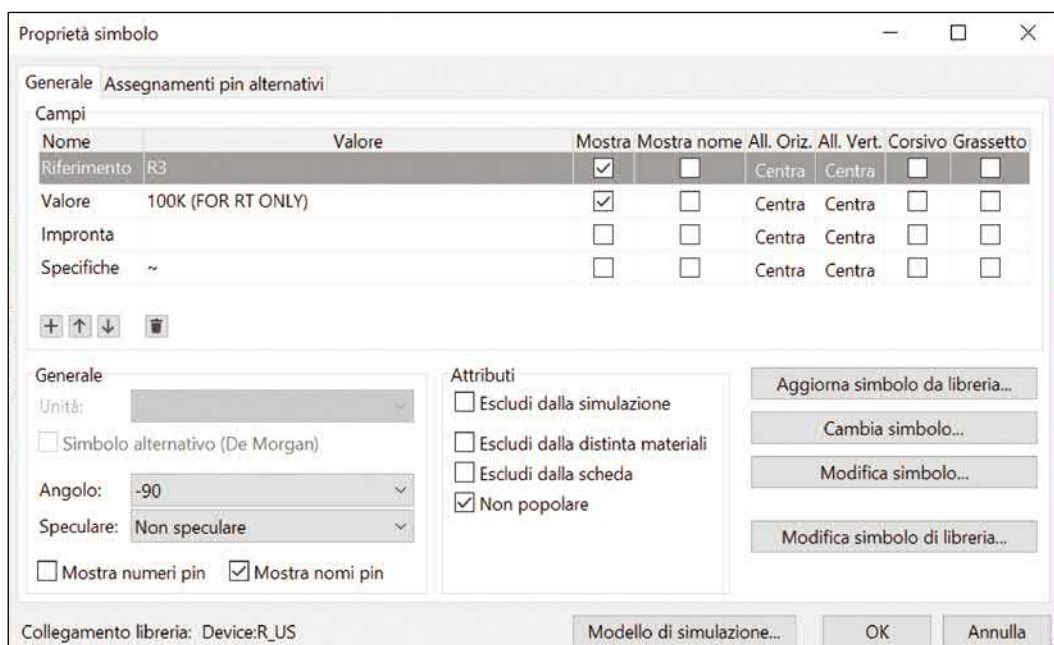


Fig. 4
Indicazione dei componenti da non montare.

Fig. 5
Impostazioni in
Proprietà per il
componente
da non montare.



rossa, come esemplificato nella **Fig. 4**. I componenti contrassegnati come "DNP" saranno esclusi dai file di piazzamento che verranno generati da KiCad. Per indicare che si vuole marcare un componente come "DNP", è necessario entrare nella finestra di dialogo delle sue proprietà: per fare ciò è possibile cliccare sopra il simbolo del componente e premere il tasto "E" della tastiera del computer, oppure selezionare il componente, cliccarvi con il pulsante destro del mouse e selezionare, dal menù contestuale che si apre, la voce "proprietà"; questo comando fa apparire la finestra di dialogo Proprietà simbolo nella sezione Attributi della quale, si deve spuntare "Non popolare" (come visibile in **Fig. 5**) e poi cliccare sul pulsante OK per confermare.

KiCad dispone di un simulatore circuitale integrato open source (chiamato *ngspice*) che può accettare vari modelli, compatibili con spice, Pspice, LTspice e Hspice; questo simulatore di circuiti può essere lanciato dal menù a tendina "Ispeziona", impar-

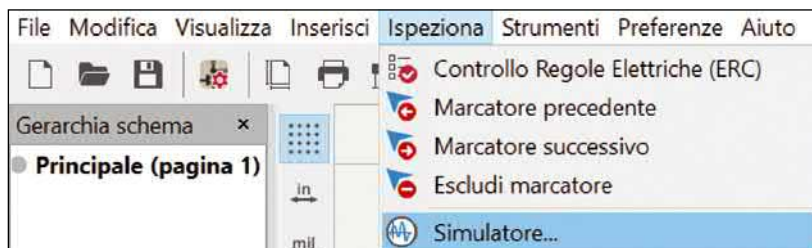
tendo da esso il comando "Simulatore..." dello "Schematic Editor" (**Fig. 6**).

Nella versione 7 è stato introdotto un editor del modello utilizzato per la simulazione, raggiungibile all'interno delle proprietà del simbolo, selezionabile cliccando sopra "Modello di simulazione"; una volta aperto è possibile definire i parametri del modello tramite una semplice interfaccia grafica a finestra piuttosto che inserendo direttamente il codice spice nello schema (**Fig. 7**).

Sono disponibili dei modelli standard, come ad esempio quelli di condensatori, resistori, diodi e transistor, ma è anche possibile caricare delle proprie librerie da cui poi andare a selezionare un componente specifico.

Una nuova funzionalità importante e largamente richiesta, che avvicina KiCad ai software commerciali, è la possibilità di definire un sistema centralizzato di librerie di simboli e di footprint, che può essere condiviso con tutti quanti hanno accesso al database, all'interno del quale è possibile effettua-

Fig. 6
Avvio del simulatore
di KiCad 7.



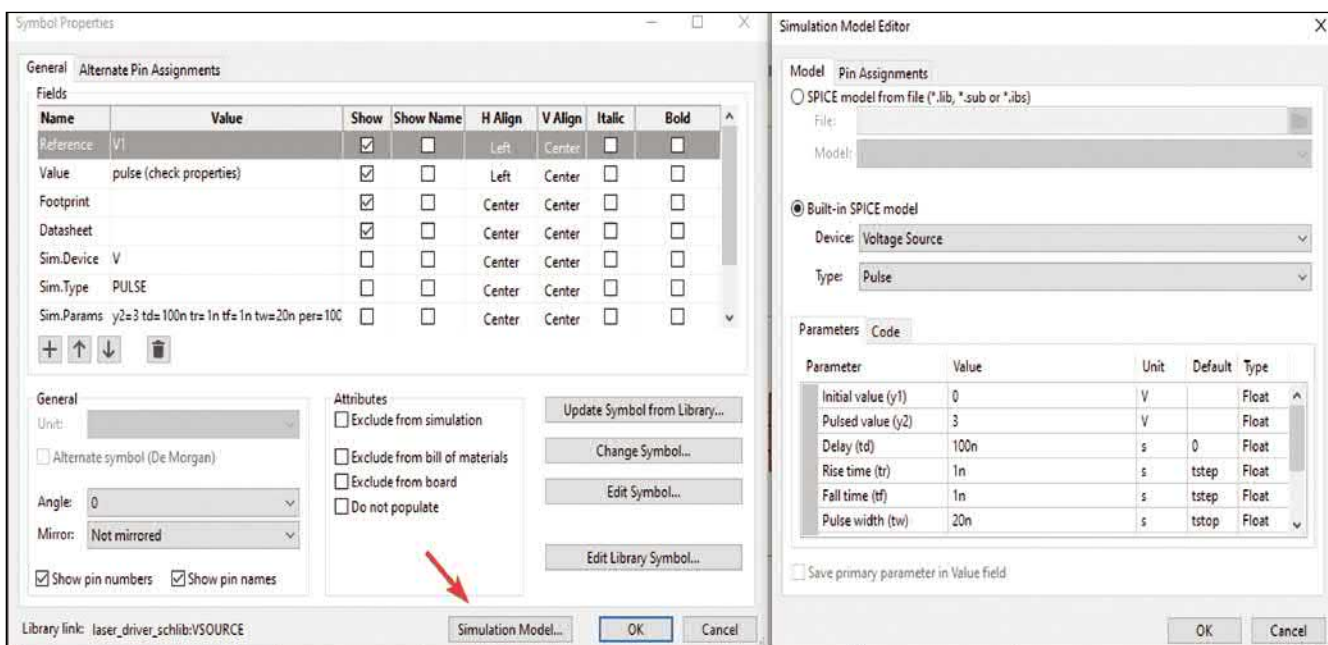


Fig. 7
Gestione dei parametri del modello per la simulazione, tramite finestra.

re una ricerca selezionando a piacimento i campi di ricerca, in modo parametrico.

Altre modifiche introdotte nell'editor degli schemi elettrici sono:

- possibilità di inserire hyperlink nello schema elettrico;
- possibilità di inserire delle forme geometriche come cerchi o rettangoli, ad esempio per contornare ed evidenziare parti dello schema elettrico;
- possibilità di effettuare automaticamente il controllo dell'eventuale disponibilità di nuove versioni di un pacchetto installato (PCM Automatic Updates);
- possibilità di inserire delle caselle di testo sia nello schematico che nell'editor di PCB;
- potenziamento del sistema di "drag & drop";
- miglioramento della tabella in cui si editano i pin dei simboli, in cui ad esempio si definiscono le caratteristiche grafiche e la posizione in cui verranno disegnati i pin;
- aggiunta di un nuovo sistema che mostra il piazzamento di un componente fuori griglia, cosa che potrebbe comportare problemi in quanto le connessioni non sono correttamente effettuate, causando errori nel PCB;
- aggiunta di una nuova funzione di tracciamento assistito, grazie alla quale le piste possono ora essere parzialmente inserite a mano e l'editor si occupa poi del tracciamento delle restanti, collegando la pista alla connessione più vicina

appartenente alla stessa NET;

- aggiunta di un nuovo e potente tool che permette di selezionare una parte dello schema e tramite un nuovo menù raggiungibile tramite il tasto destro del mouse o con il tasto "P" all'interno del PCB editor, permette lo spostamento in gruppo dei footprint dei componenti selezionati, invece che singolarmente;
- miglioramento nella generazione di PDF che vengono generati con un indice che contiene i link alle singole pagine ed ai singoli elementi costituenti lo schema elettrico;
- aggiunta all'interno dei PDF, di alcune informazioni relative ai componenti quando ci si clicca sopra, visualizzate quando si utilizza Adobe Acrobat Reader (al momento la compatibilità con gli altri lettori di PDF non è garantita); un esempio è proposto nella **Fig. 8**.

NOVITÀ VERSIONE 7: PCB EDITOR

Anche per l'editor dei circuiti stampati esistono miglioramenti e novità:

- aggiunta di una nuova opzione che permette di verificare la consistenza tra il footprint utilizzato nel layout della scheda che si sta realizzando, con il footprint presente nella libreria del componente stesso;
- introduzione nella tabella riassuntiva del DRC (Design Rule Checking) di un tab separato in cui sono presenti i test che sono stati ignorati;
- introduzione di due nuove regole per il DRC

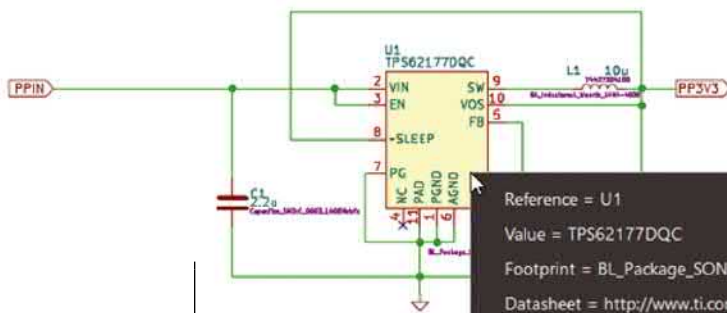


Fig. 8
Pop-up con le informazioni sul componente.

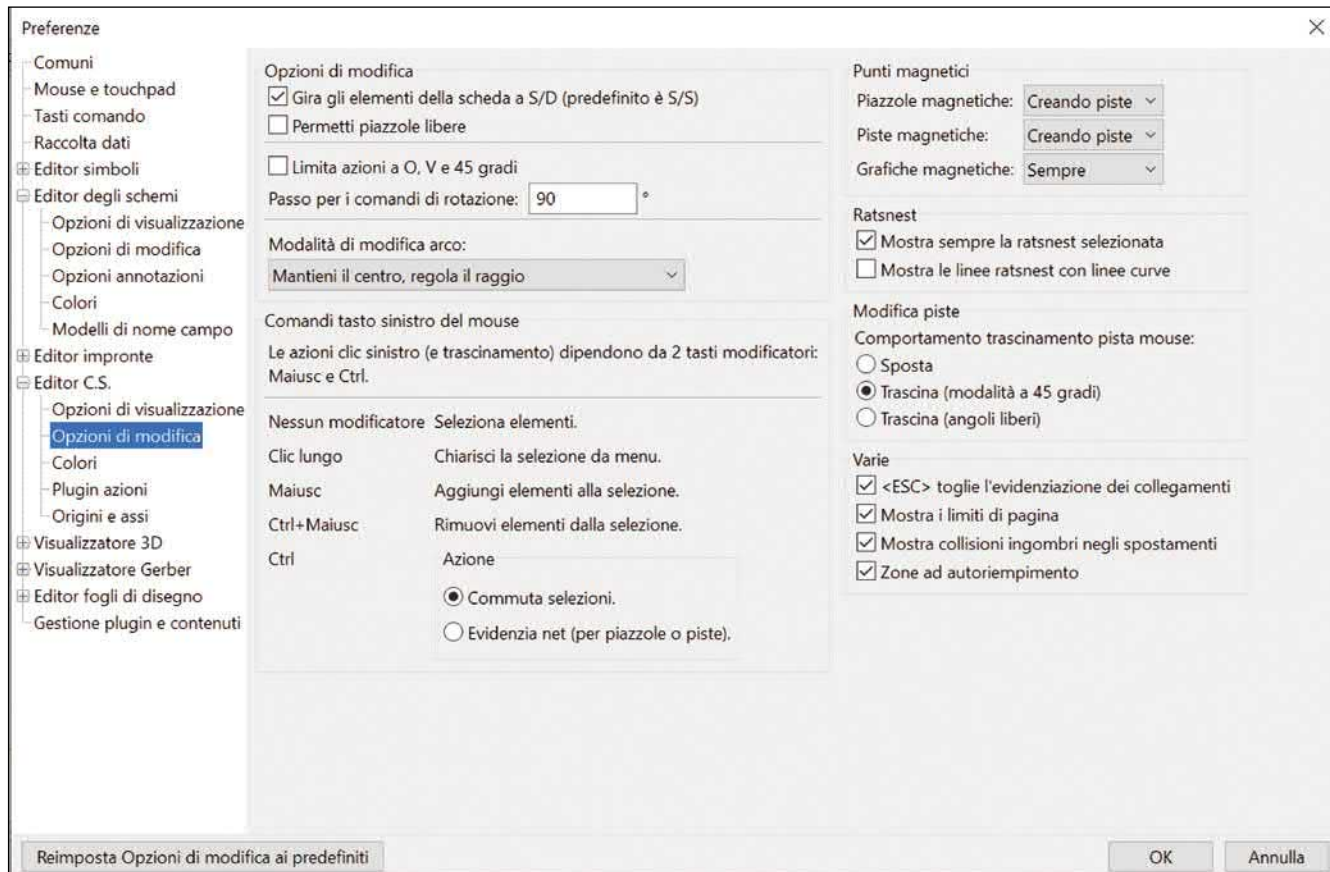
Reference = U1
Value = TPS62177DQC
Footprint = BL_Package_SON:WSON-10-1EP_2x3mm_P0.5mm_EP0.84x2.4mm_ThermalVias
Datasheet = <http://www.ti.com/lit/ds/symlink/tps62177.pdf>
Description = 500mA Step-Down Converter with Sleep Mode, fixed 3.3V Output Voltage, 4.75-28V input
Keywords = step-down dc-dc buck regulator

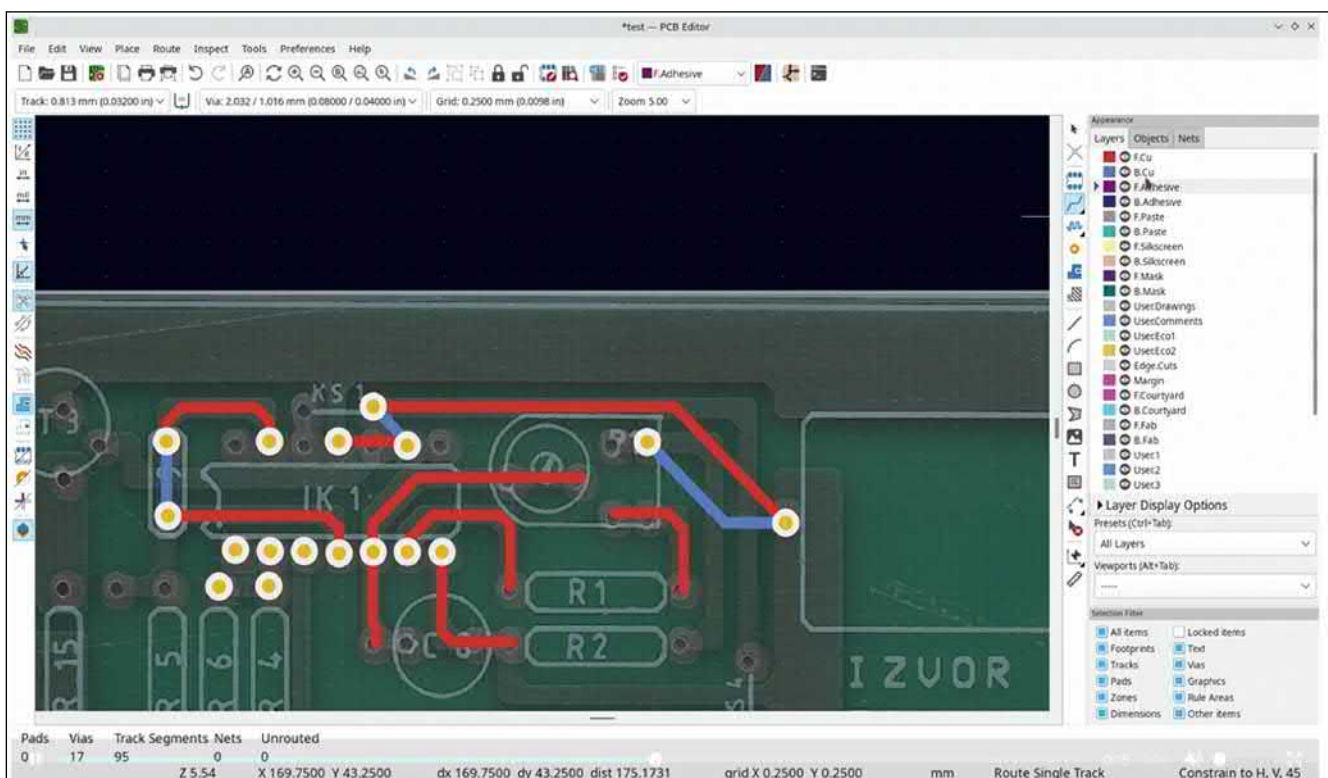
Fig. 9
Impostazioni di riempimento automatico delle piste.

("mechanical_clearance" e "mechanical_hole_clearance"), aggiunte all'interno delle regole personalizzate relative alle "clearance" e "hole_clearance constraints";

- introduzione della funzione che permette di effettuare l'unroute delle piste selezionate, utile sia per correggere errori senza rifare il PCB da zero, sia per modificare il percorso creato;

- inserimento di un nuovo pannello delle proprietà che permette un rapido accesso a molti dei parametri di un oggetto e che permette di editare più oggetti allo stesso tempo; di tale funzionalità è previsto un ulteriore sviluppo nelle future versioni di KiCad;
- aggiunta di un nuovo pannello di ricerca, che permette di cercare e filtrare i risultati, utilizzando i caratteri "jolly";





- introduzione di un'opzione, abilitata per impostazione predefinita, che permette di riempire automaticamente le zone di riempimento quando viene rilevato un cambiamento della scheda, invece di dover effettuare nuovamente a mano tale operazione; l'opzione può essere eventualmente deselezionata entrando in "Preferenze", "Editor c.s.", "Opzioni di modifica", deselezionando nella sezione "Varie", "Zone ad autoriempimento" (**Fig. 9**).

Un'opzione interessante che sarà molto apprezzata da quanti, tra voi, potrebbero avere necessità di effettuare un reverse engineering di una scheda esistente, di cui magari si è persa la documentazione, è quella di selezionare un'immagine da porre in background, magari del PCB da replicare, in modo da poter piazzare i componenti e tracciare le piste, usando come riferimento la figura o la foto sottostante (la **Fig. 10** esemplifica l'utilizzo di tale strumento). Tale funzionalità semplifica enormemente il lavoro, risparmiando al progettista l'onere di dover misurare tutte le quote del PCB da ridisegnare e replicarle sullo schermo del computer a rischio di errori: è sufficiente ripassare le piste ed esportare il Gerber per avere un circuito identico all'originale.

CONCLUSIONI

I software open source hanno raggiunto un livello di maturità e una diffusione senza precedenti, rendendo disponibili a chiunque dei tool di sviluppo veramente potenti, senza dover sostenere l'onere economico di acquistare licenze di prodotti professionali, ingiustificati quando magari l'ECAD ci serve per realizzare a livello scolastico o dilettantistico i nostri progetti.

KiCad spicca nel panorama di questi tool.

In questo articolo vi abbiamo esposto gli sviluppi di KiCad, un ottimo strumento di utilizzo gratuito che permette di realizzare circuiti stampati anche multistrato, ormai arrivato alla versione 7, la quale porta, a poco più di un anno di distanza dal rilascio della versione 6 (già molto ricca di caratteristiche e miglioramenti rispetto alla precedente versione, la 5...) ulteriori migliorie e nuove funzionalità indubbiamente utili a chi deve disegnare schemi e circuiti stampati.

Siamo certi che vorrete subito provare dal vivo quello che uno strumento come kiCad può offrirvi e che sfrutterete le nuove funzionalità delle quali vi abbiamo messo al corrente in queste pagine. Scaricate quindi l'ultima versione dal sito www.kicad.org e... buon lavoro con il vostro nuovo CAD per circuiti elettronici! ■

Fig. 10
Tracciatura da un'immagine importata.

Solo su
futuranet.it

Arduino, la piattaforma open source per tutti!

Entra
nell'affascinante
mondo di Arduino
e impara ad
utilizzarlo al meglio
con i nostri kit!
Scegli quello più
adatto a te.

cod. ARDUKITLEVEL1

ARDUKITLEVEL1

ARDUINO UNO R3 ORIGINALE
+ SET COMPONENTI



€
35,90

cod. ARDUKITLEVEL2

ARDUKITLEVEL2

ARDUINO UNO R3 ORIGINALE
+ SET COMPONENTI



€
45,90

cod. ARDUKITABC

ARDUKITABC

STARTER KIT CON ARDUINO
UNO REV3 E COMPONENTI



€
58,00

IL PIÙ
COMPLETO!

cod. ARDUKITV6

ARDUKITV6

ARDUINO ORIGINALE +
LIBRO L'ABC DI ARDUINO +
COMPONENTI



€
68,00

cod. ARDUINOSTARKIT

ARDUINOSTARKIT

ARDUINO ORIGINALE +
LIBRO IN ITALIANO +
COMPONENTI



€
118,00

Prezzi IVA inclusa.

**FUTURA
ELETTRONICA®**

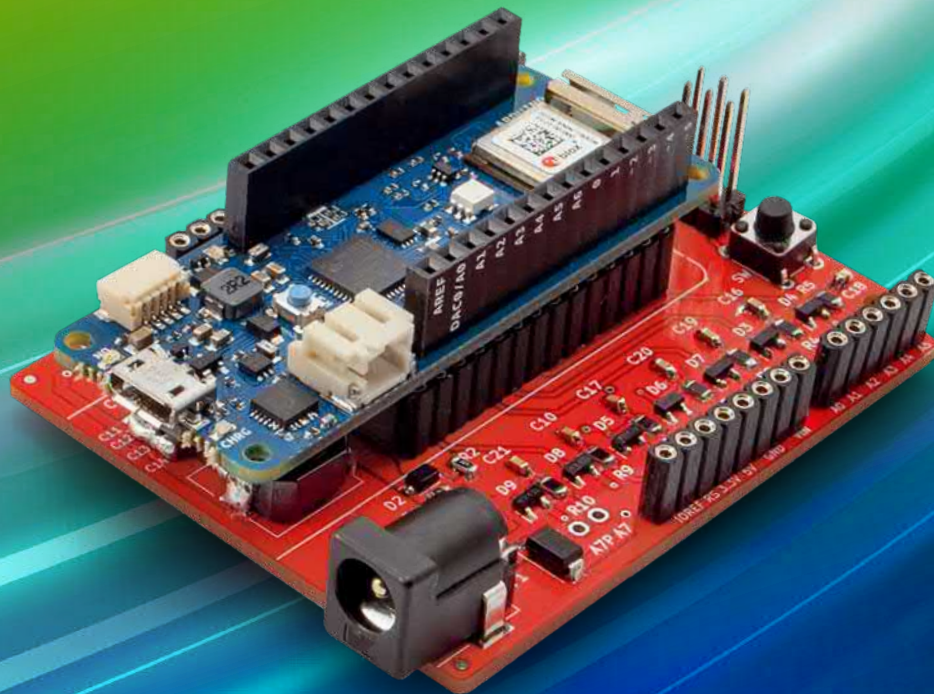
www.futurashop.it

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

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

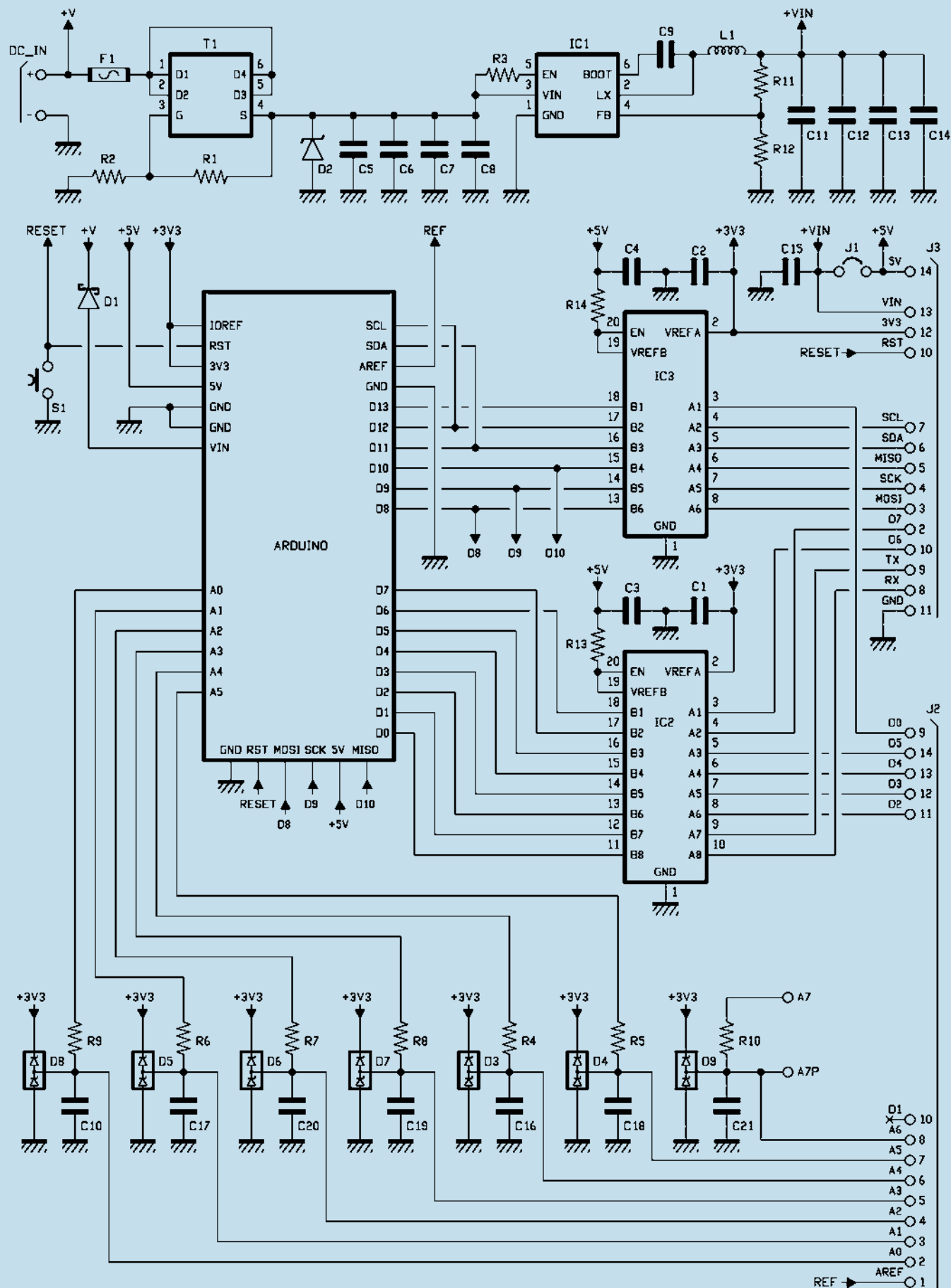
MKR-UNO ARDUINO SHIELD

dell'Ing. **VINCENZO MENDOLA**



Adattatore che consente la compatibilità tra la serie di board MKR e gli shield di Arduino UNO.

Dalla fondazione di Arduino ad oggi sono cambiate tantissime cose, partendo dalle caratteristiche hardware e arrivando alla tipologia di utilizzatori ed applicazioni; nata come una piccola scheda a microcontrollore ad 8 bit per avvicinare i giovani al mondo della tecnologia, anche se non provenienti dal mondo dell'elettronica, Arduino è diventato un dispositivo hardware con moltissimi modelli, schede di espansione (i cosiddetti "shield"), utilizzato sia dagli hobbisti, i cosiddetti "makers", sia dai professionisti. Visto il gran numero di shield ufficiali e non, realizzati in questi anni per la famiglia Arduino UNO, abbiamo pensato di fare cosa gradita pensando ad una scheda di adattamento che permettesse di utilizzare quanto creato per le board Uno e compatibili, come ad esempio le



Fishino UNO e Fishino 32, ma anche per la recente famiglia MKR, che ha espanso moltissimo le potenzialità di Arduino, sia in termini di connettività (aggiungendo nativamente BLE e WiFi) sia in termini di prestazioni, utilizzando dei microcontrollori a 32 bit. Ad onor del vero esiste già uno shield di adattamento ufficiale ma ci è sembrato veramente troppo "basic". Abbiamo deciso pertanto di pensarne uno nuovo realizzato partendo da zero, garantendo comunque la compatibilità elettrica e meccanica con le schede esistenti.

MKR VS UNO

Dopo anni in cui gli sviluppatori di Arduino sono stati legati ai microcontrollori ad 8 bit della Atmel (ora parte di Microchip), in particolar modo al famosissimo ATmega328P, hanno finalmente deciso di utilizzare anche MCU a 32 bit, più veloci e prestanti, affiancandole a moduli radio per estenderne la connettività (BLE, WiFi, LoRa, Sigfox, GSM/3G, LTE), dando vita alla famiglia MKR, caratterizzata, oltre ai miglioramenti appena visti, da un nuovo fattore di forma, più piccolo e compatto del classico "UNO". Per completezza facciamo presente che esistono anche dei modelli con FPGA, come ad esempio l'Arduino MKR Vidor 4000 oppure i prodotti della linea professionale Arduino Pro, con caratteristiche ancora più spinte, nati per applicazioni particolari e dai prezzi meno "hobbistici".

La transizione verso i microcontrolli ARM a 32 bit ha comportato, per prima cosa, il passaggio della tensione di alimentazione e dei bus di comunicazione, come ad esempio I²C, allo standard a 3,3V, invece dei 5V a cui siamo stati abituati con la famiglia UNO e derivati. Anche l'ADC lavora conseguentemente a questa tensione massima per cui è stato necessario tenerne conto duran-

te la progettazione dello shield di adattamento. Per quanto riguarda la dimensione delle schede MKR, essendo queste come abbiamo già detto di dimensioni inferiori (è il formato più piccolo delle attuali schede ufficiali di Arduino), non è stato affatto problematico inserire nella nostra scheda due file di strip femmina a passo 2,54mm in cui introdurre la Arduino MKR a 32 bit, rispettandone le connessioni native. Nella progettazione dello shield, si è cercato di seguire il più possibile le linee guida per la realizzazione di schede compatibili con Arduino UNO, a partire dal fattore di forma. Per questo motivo, chi ha già utilizzato Arduino UNO e i suoi shield, non farà fatica a trovarsi tra le mani un oggetto familiare: a parte il connettore "DC IN", cui si connette la tensione d'ingresso, ed i due strip J2 e J3 ai quali va connessa la Arduino MKR, le altre connessioni sono quelle standard, con le porte analogiche da A0 ad A5, i GPIO da A2 a A13, compresi RX e TX della connessione seriale, le connessioni del bus I²C e, ovviamente, vicino al connettore degli ingressi analogici, c'è quello delle alimentazioni, anche questo con la stessa pin-out della Arduino UNO. Per mantenere la compatibilità è stato inserito anche il connettore a 6 pin ICSP, nella stessa posizione in cui lo troviamo in Arduino UNO; appena sotto abbiamo messo il pulsantino di RESET. Ovviamente anche i fori di fissaggio sono stati messi nelle posizioni originali, a parte quello in basso, vicino al jack di alimentazione, che è stato eliminato perché troppo vicino ai componenti. Per completezza, rimandiamo allo schema elettrico e al piano di montaggio.

SCHEMA ELETTRICO

Passiamo ora all'analisi dello schema elettrico, partendo innanzitutto dalla sezione di alimentazio-

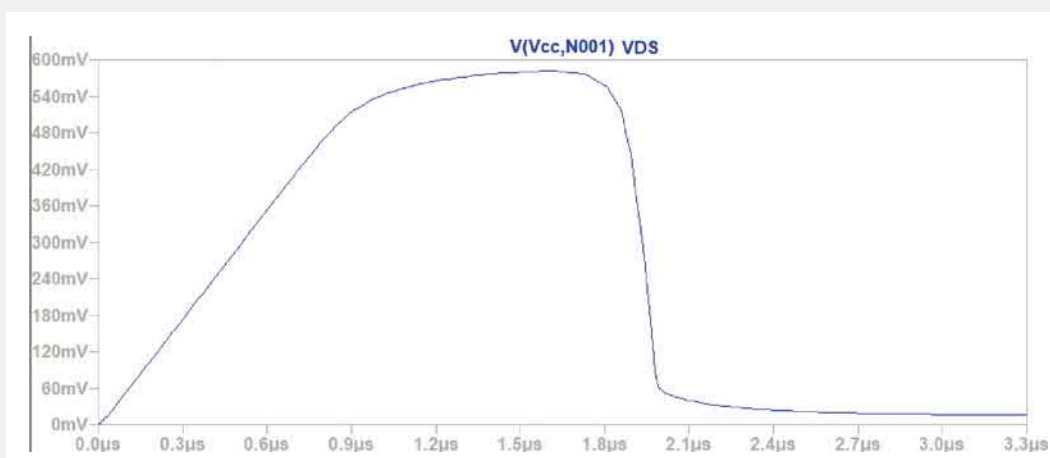


Fig. 1
Simulazione del comportamento dell'ingresso di alimentazione.

→ **Fig. 2**
Misura attraverso la
sonda differenziale.



ne: questa è costituita essenzialmente da un robusto buck converter da 3A (IC1), che, vista l'attuale difficoltà nel reperimento dei componenti, è stato scelto in modo da poter essere sostituito almeno da un alternativo: la scelta originaria era per il TPS54302 della Texas Instruments, al momento difficilmente reperibile, caratterizzato da una V_{in} compresa tra 4,5V e 28V, pin to pin compatibile con l'integrato RT6214 della Richtek, che accetta tensioni d'ingresso tra i 4,5V e 18V, inferiori all'integrato della Texas Instruments, ma comunque adeguate per l'utilizzo che vogliamo farne nel nostro shield; sarà necessario tenere conto di questa differenza solo nel caso di applicazioni particolari, in cui si voglia ad esempio utilizzare una tensione esterna di 24V. Entrambi i produttori hanno messo all'interno di IC1 varie protezioni, tra cui una protezione in caso di sovracorrente in uscita e una protezione in caso di riscaldamento eccessivo. All'ingresso dell'integrato, subito dopo il jack a cui connettiamo la tensione continua di alimentazione, abbiamo un fusibile PTC (autoripristinante), seguito dal mosfet a canale P indicato con T1 che, insieme ad R1, R2 e allo zener D2, realizza la protezione contro le inversioni di polarità.

Il funzionamento è il seguente: applicando una tensione con polarità corretta, cioè con il positivo sul drain e negativo a massa, il diodo di body è polarizzato positivamente ed entra pertanto in conduzione, portando il source a un potenziale pari a quello d'ingresso diminuito dalla caduta sul diodo

stesso; il partitore costituito da R1 e R2 si assicura, insieme allo Zener D2 da 12V, che la tensione tra source e gate, non sia mai superiore ai 12V, tensione più che sufficiente a portare correttamente in conduzione il MOSFET, garantendo al contempo un buon margine di sicurezza ($V_{GSmax} = +/ - 20V$ per il MOSFET in uso). Essendo la V_{GS} in modulo maggiore del valore di soglia V_{GSth} , il MOSFET entra in conduzione, cortocircuitando il diodo di body dello stesso, portando la caduta tra DRAIN e SOURCE a zero; tutto questo avviene in maniera pressoché istantanea.

Nella **Fig. 1** è possibile vedere il risultato di una simulazione con LTSPICE della sezione d'ingresso del circuito di alimentazione, con una corrente circolante nel PMOS di circa 600mA, che dimostra quanto appena detto: si vede inizialmente la caduta sul body diode pari a circa 600mV ma dopo appena una manciata di microsecondi il MOSFET entra in conduzione e la caduta di tensione tra Drain e Source scende a zero.

In **Fig. 2** vede invece la misura effettuata tramite una sonda differenziale che mostra l'andamento della VDS all'applicazione della tensione d'ingresso: a parte una transizione iniziale non presente nella simulazione, si vede anche in questo caso la caduta di tensione che inizialmente è di circa 680mV ma poi va a zero; l'unica vera differenza con la simulazione è il tempo in cui questo avviene: possiamo dire che il modello utilizzato è piuttosto "ottimistico", calcolando questo transitorio della

durata di circa 2 μ s, mentre in realtà abbiamo misurato circa 2 ms, quindi 3 ordini di grandezza in più, differenza quindi non trascurabile ma di impatto nullo nella nostra applicazione reale.

Ricordiamo che le simulazioni servono per orientarci nelle scelte progettuali ma non vanno prese come "oro colato" perché i modelli non sono mai perfetti e soprattutto non sempre è possibile considerare e parametrizzare tutte le grandezze e gli elementi in gioco, dato che a volte ci possono essere effetti di ordine superiore che magari noi stessi trascuriamo creando il modello, che si sommano creando degli effetti che possono "polarizzare" il risultato della misura.

Aggiungiamo che la misura si può effettuare anche usando due sonde comuni opportunamente collegate, cioè una sul Drain e una sul Source con entrambe le masse delle stesse connesse alla massa del circuito; per ottenere la misura voluta è necessario far calcolare la funzione differenza all'oscilloscopio, implementabile in diversi modi a seconda dello strumento in uso. Si ricorda che la massa dell'oscilloscopio è collegata direttamente alla terra e che collegare questo terminale (la massa della sonda) ad un potenziale diverso da zero, può causare gravi danni allo strumento e al dispositivo connesso.

La **Fig. 3** mostra invece la simulazione del caso in cui si applichi all'ingresso una tensione negativa di -15V: in essa si vede che l'uscita rimane a zero.

Per quanto riguarda C5, C6, C7, C8, C11, C12, C13, C14, sono i condensatori di ingresso e di uscita

previsti per il corretto funzionamento dell'integrato IC1. R11 e R12 sono le resistenze di feedback, che servono per impostare la tensione in uscita e variano a seconda dell'integrato: R11 è 100K per l'integrato della Texas Instruments e 40.2K per la Richtek, mentre R12 è del valore di 13,3 kohm per la Texas Instruments e di 7,68 kohm per la Richtek, ovviamente calcolati in entrambi i casi per avere 5V in uscita; questo perché i due convertitori utilizzano internamente Vref di valore differente. Anche per l'induttore L1 bisogna distinguere tra il caso in cui venga utilizzato il TPS54302, nel qual caso si utilizzerà un induttore da 10 μ H, ed il caso del RT6214, in cui servirà un induttore da 4,7 μ H; in entrambi i casi il circuito è stato dimensionato per poter erogare una corrente in uscita fino a 3A e quindi l'induttore dovrà avere con una corrente nominale di saturazione adeguata. C9 è il condensatore di bootstrap, previsto per entrambi gli integrati. R3 è la resistenza di pull-up dell'enable, indispensabile per far funzionare gli RT6214, mentre per i TPS54302 non è necessaria e non va montata. L'uscita a 5V di IC1 è la VIN sul pin 13 di J3 dell'Arduino MKR; tramite J1 è possibile, qualora necessario, connettere la VIN sul pin 13 alla 5V sul pin 14, connessa anche al 5V sul connettore ICSP, in questo modo si può fornire la 5V all'arduino MKR senza connetterlo alla presa USB.

Come per i 4 connettori dell'Arduino UNO visti in precedenza, anche quelli a cui connettiamo l'Arduino MKR rispettano le connessioni standard: su J2 ci sono i 7 ingressi analogici A0-A6 ed i primi

Fig. 3
Simulazione con tensione d'ingresso negativa.



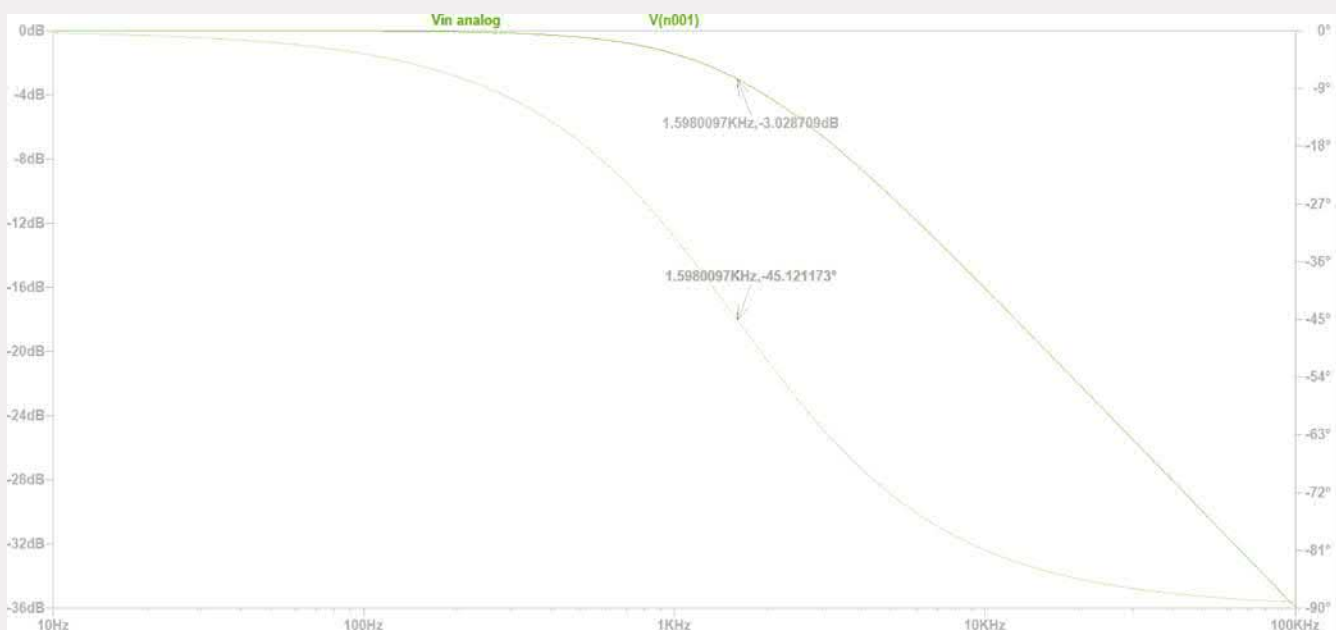


Fig. 4
Il diagramma
di Bode
simulato.

6 GPIO (D0-D5) mentre su J3 ci sono gli altri GPIO, le connessioni del Bus I²C, SPI, UART e le tensioni 5V e 3,3V. Facciamo notare alcune differenze tra la famiglia "UNO" e quella "MKR": sugli Arduino MKR il pin corrispondente a D11 e D12 sono rispettivamente "SDA" e "SCL", mentre il D14 e il D13 sono "TX" e "RX" e sono stati collegati a D1 e D0 dell'Arduino UNO; il pin D0 MKR è stato collegato al D13 dell'UNO, mentre il pin D1 del MKR non è stato collegato, non corrispondendo direttamente ad una funzionalità dell'UNO.

Nella realizzazione dello shield è stato necessario prendere alcune decisioni, tipo quelle appena viste, abbiamo, per così dire, dato una nostra interpretazione nel cercare di mappare i pin-out delle due famiglie di Arduino; è chiaro che per applicazioni particolari e meno universali potrebbe essere necessario avere una corrispondenza differente delle porte, nel qual caso le modifiche andranno fatte da voi, in base alle vostre esigenze.

Come abbiamo già spiegato, la famiglia MKR utilizza come tensione di alimentazione del microcontrollore, 3,3V invece dei canonici 5V e di conseguenza anche i BUS seriali, i GPIO e gli ingressi dell'ADC funzionano a 3,3V; per questo motivo nello shield sono presenti i due integrati IC2 e IC3 che effettuano la traslazione bidirezionale da 5V sul lato connesso all'Arduino UNO ai 3,3V della Arduino MKR per le porte digitali. A tal proposito è necessaria una precisazione: sul lato a 5V è necessaria una resistenza di pull-up su tutte le porte utilizzate, per permettere al traslatore di livello di

lavorare correttamente ed operare la conversione del livello alto da 3,3V a 5V; in molti shield queste resistenze sono già inserite, per cui si è scelto di non inserirle nella nostra scheda; in ogni caso è molto facile aggiungerle, qualora necessario, utilizzando delle resistenze a foro passante, connesse al connettore femmina delle porte digitali a cui serve il pull-up e la 5V, disponibile ad esempio sullo strip delle alimentazioni.

Per quanto riguarda le porte analogiche, sono previste delle protezioni, costituite, analizzando ad esempio la porta analogica A0, dal doppio diodo schottky D8 che taglia le tensioni superiori a 3,3V (a cui si aggiunge la tensione di soglia del diodo che entra in conduzione) e protegge la porta contro le inversioni di polarità e da R9 che limita la corrente in ingresso alla porta ed insieme al condensatore C10 (che può anche non essere montato, accettando di rinunciare all'azione di filtro) realizza il filtro passa basso antialiasing. In particolare, se si sceglie (come abbiamo fatto noi per il nostro test) di utilizzare per R9 il valore di 1KΩ e per C10 il valore di 100 nF, avremo un filtro passa-basso del primo ordine, con una frequenza di taglio di 1,59 kHz, determinabile con la formula:

$$f_t = \frac{1}{2\pi RC}$$

Otterremo anche una fase di -45°, che deriva dall'applicazione della formula seguente:

$$\omega = -\tan^{-1} 2\pi fRC$$



Fig. 5
Andamento
della tensione
di uscita del
filtro.

come chiaramente mostrato dal diagramma di Bode proposto nella **Fig. 4**, in cui è visualizzato l'andamento sia del modulo che della fase del nostro sistema, ottenuto dalla simulazione fatta con lo strumento LTSPICE.

In questo caso le misure rispecchiano perfettamente l'andamento delle simulazioni.

Nella **Fig. 5** si vede l'andamento della tensione in uscita del filtro, al cui ingresso è applicata una sinusoide di ampiezza pari a 250 millivolt di picco

e frequenza pari a 1,59 kHz; l'ampiezza della componente in uscita è pari a 178 mV, corrispondente all'ingresso a -3dB; le formule seguenti chiariscono da dove derivano tali valori:

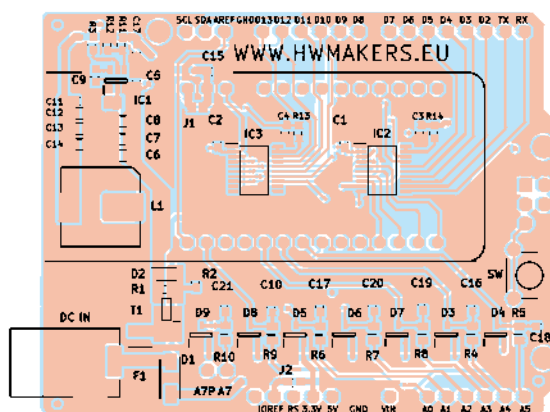
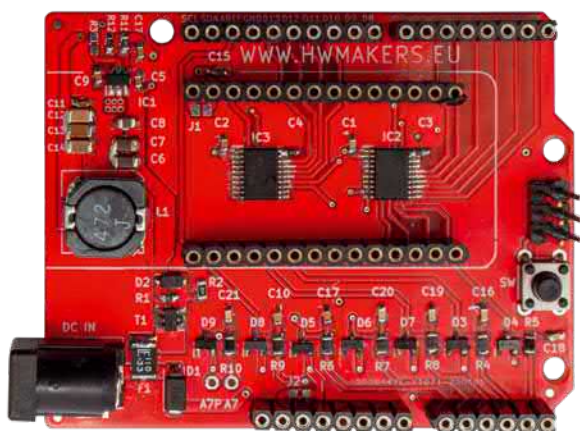
$$V_{in(dB)} = 20 \log V_{in}$$

$$V_{in[-3dB]} = V_{in} * 10^{-\frac{3}{20}} = V_{in} * 0,707 = 177mV$$

Per quanto riguarda la fase, è facile calcolare che un ritardo di 78 μs corrisponde ad uno sfasamento



Fig. 6
Sfasamento
di 45°
all'uscita del
filtro.



Elenco Componenti:

R1: 15 kohm (0603)
 R2: 4,7 kohm (0603)
 R3: 100 kohm (0603) vedi testo
 R4, R5, R6, R7, R8, R9, R10: 1 kohm (0805)
 R11: 40,2 kohm / 100 kohm (0603) vedi testo
 R12: 7,68 kohm / 13,7 kohm (0603) vedi testo
 C1, C2, C3, C4, C5, C9, C10, C11, C15, C16, C17, C18, C19, C20, C21: 100 nF ceramico (0603)
 C6, C7: 10 μ F 25 VL ceramico (1206)
 C8: 2,2 μ F 25 VL ceramico (1206)
 C12, C13, C14: 22 μ F 16 VL ceramico (1206)
 D1: MBRA320T3G
 D2: Zener 12V PTVS12VS1UR,115

D3, D4, D5, D6, D7, D8, D9: BAT54S
 IC1: TPS54302 TI / RT6214AHGJ6F vedi testo
 IC2, IC3: LSF0108PW
 L1: 4,7 μ H / 10 μ H vedi testo
 S1: Microswitch
 T1: CPH6354
 F1: PTC 1812L110_33MR

Varie:

- Plug DC
- Strip Femmina torniti passo 2,54 mm
- Circuito stampato S1672 (69 x 54 mm)

di 45°, che risulta visibile nell'oscillogramma proposto nella **Fig. 6**. Le formule che consentono di arrivare a questa conclusione, ossia di determinare la corrispondenza tra sfasamento e ritardo, sono proposte qui di seguito (il ritardo viene definito con T e lo sfasamento con φ):

$$T = \frac{1}{f} = 629\mu s$$

$$\varphi = \frac{629\mu s}{360} * 45 = 78\mu s$$

Sulle schede Arduino MKR è disponibile anche un ingresso analogico in più rispetto a quelli delle schede standard (per esempio la UNO), chiamato A6, che è stato collegato anch'esso al circuito di protezione e filtraggio appena analizzato ed è disponibile sui pin A7P - A7.

Se si volesse eliminare del tutto il filtraggio agli ingressi analogici, si può decidere di non montare il condensatore e utilizzare una resistenza da 0 ohm; si ricorda che la resistenza ha anche una funzione di protezione degli ingressi analogici del micro-

controllore, limitando la corrente che può circolare negli stessi; il suo valore, qualora venisse scelto troppo alto, potrebbe interferire con il corretto funzionamento del circuito di Sample & Hold interno al microcontrollore, causando una diminuzione dell'accuratezza della misura, per cui si consiglia di non aumentarlo troppo, mantenendolo possibilmente al di sotto di 1,5 K Ω .

Chiaramente montando resistori da 0 ohm bisogna accettare il rischio di non disporre della protezione da sovracorrente implementata sugli ingressi proprio da essi.

REALIZZAZIONE PRATICA

Bene, con la teoria e l'analisi dello schema elettrico abbiamo concluso; adesso possiamo occuparci degli aspetti costruttivi, vale a dire della preparazione dello shield/adattatore qui descritto.

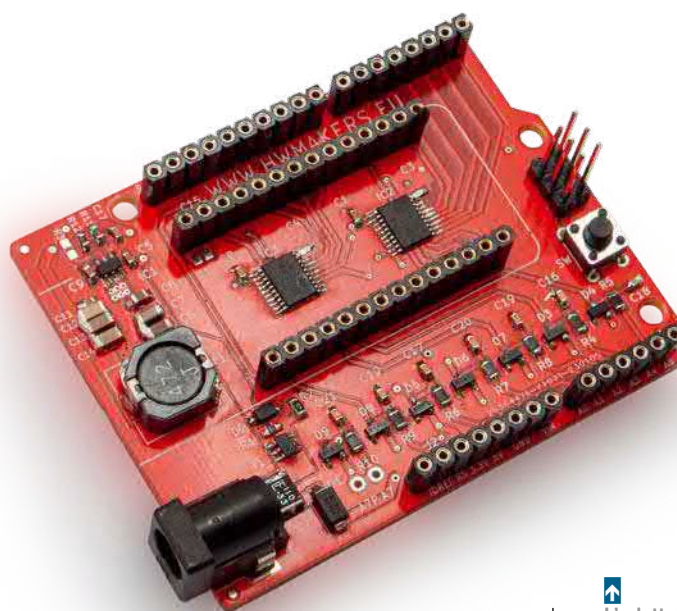
Per la realizzazione della scheda abbiamo scelto di utilizzare un PCB a 4 strati, non tanto per la complessità del circuito, tutto sommato piuttosto limitata, ma più che altro per il grande numero di

componenti e di connessioni necessari. Ci rendiamo conto che a livello prototipale e hobbistico non è possibile realizzare un circuito stampato del genere, che è un multilayer, pertanto suggeriamo a chi fosse intenzionato a realizzare il progetto, di appoggiarsi a un servizio di PCB industriale, magari di quelli on-line. Allo scopo si possono utilizzare i file gerber caricati sul nostro sito www.elettronica.in.it ed utilizzare il servizio di prototipazione rapida dei PCB offerto da Futura Elettronica (PCBPRODUCTION), sul sito www.futuranet.it; anche i componenti principali sono facilmente reperibili presso la stessa azienda. Una volta in possesso del circuito stampato e dei componenti, la saldatura non presenta particolari difficoltà per chi ha una buona esperienza nei montaggi elettronici, essendo tutti i componenti di dimensioni e di tipologia facilmente saldabili a mano con un saldatore a stilo e una lega saldante di buona qualità, quest'ultima di diametro preferibilmente inferiore o uguale a 0,7 mm.

Come sempre consigliamo di partire dai componenti SMD a più basso profilo, per poi montare per ultimi i componenti a montaggio tradizionale (foro passante) e quindi i pin-strip, il jack DC e gli strip femmina. In tutte le fasi del montaggio, per il corretto orientamento degli elementi polarizzati fate riferimento al piano di montaggio visibile in queste pagine. Per il collaudo della scheda si procederà verificando innanzitutto la correttezza delle saldature, l'assenza di cortocircuiti causati dallo stagno (che nel caso possono essere rimossi aiutandosi con la treccia dissaldante ed il saldatore ed eventualmente anche con del fluxante) per poi passare alla verifica della tensione di 5V a vuoto, ossia senza shield e senza board MKR connessa. Una volta fatto ciò, si potrà inserire, nello zoccolo formato dagli strip femmina, la Arduino MKR e a quel punto, alimentata la scheda, si potranno testare le porte digitali e quelle analogiche on-board attraverso gli header dello shield. Come vedete, passate dalla MKR alla UNO senza fatica... Con questo è tutto: abbiamo terminato.

CONCLUSIONI

In questo articolo abbiamo visto come sia possibile realizzare uno shield di adattamento tra la nuova serie di Arduino MKR e la storica Arduino UNO ed i suoi shield, mantenendo la compatibilità con forme, dimensioni e tensioni caratteristiche delle due versioni, una con microcontrollore ad 8 bit, l'ormai conosciutissimo ATmega 328P, l'altra con i nuovi microcontrollori Cortex a 32 bit.



↑
L'adattatore trasla la pinnatura delle Arduino MKR su quella di Arduino UNO.

Tale compatibilità è stata raggiunta utilizzando un DC/DC converter in grado di fornire una tensione di 5V e di erogare fino a 3A, due traslatori di livello per i segnali digitali ed i GPIO i cui livelli vanno convertiti in maniera bidirezionale da 3,3V a 5V, cui sono state aggiunte una serie di protezioni sugli ingressi analogici ed eventualmente anche dei filtri antialiasing.

La grande praticità della scheda è nella possibilità di utilizzare la serie Arduino MKR come fosse una Arduino UNO, quindi interfacciandola fisicamente e meccanicamente come fosse Arduino UNO, ma beneficiando delle prestazioni più elevate della MKR. Avrete così in mano un prezioso strumento per eseguire l'upgrade di progetti esistenti e mantenere gli shield della Arduino UNO, con evidente risparmio di tempo e denaro! ■



Cosa occorre?

La serie MKR è disponibile presso Futura Elettronica. Il PCB della scheda presentata in questo articolo può essere realizzato con il servizio di realizzazione circuiti stampati (cod. PCBPRODUCTION) utilizzando il gerber scaricabile dalla pagina di questa rivista. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

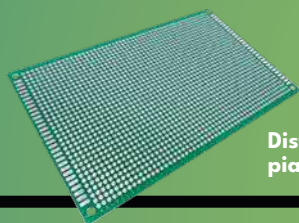
Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 - www.futurashop.it

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



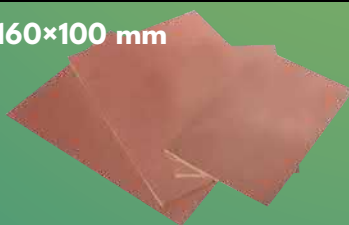
COD.	MISURA (mm)	N. CONT.	PREZZO (€)	COD.	MISURA PIASTRA (mm)	MISURA BASE (mm)	N. CONT.	PREZZO (€)
BB830	165×55	830	5,50	BB3220	188×182×9,5	240×210	3220	21,00

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

€5,50



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.

SCHEDA FTR2040LoRa



di PIER CALDERAN

La scheda FTR2040LoRa unisce le potenzialità del chip RP2040 di Raspberry Pi e del modulo RFM95W allo scopo di creare un end device LoRa/LoRaWAN in grado di comunicare con The Things Network e altri cloud simili.

N

el corso dedicato al microcontrollore RP2040 (vedi le sei puntate dal numero 266 al numero 271 di Elettronica In) abbiamo spiegato come collegare la scheda FTR2040 a una breakout LoRa/LoRaWAN della HopeRF RFM95W, con collegamenti "volanti".

Per semplificarvi le cose, abbiamo

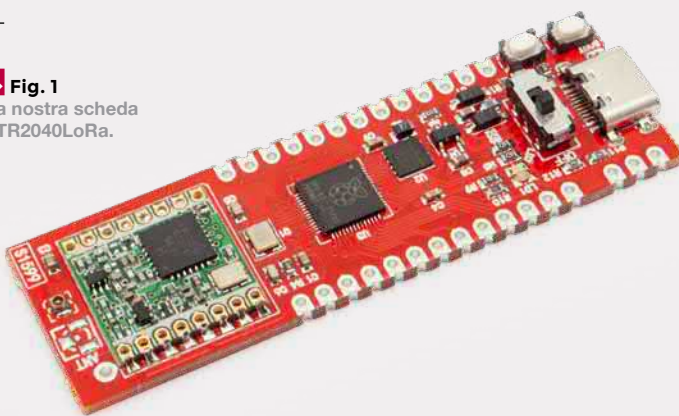
pensato a una nuova scheda che comprendesse già il modulo HopeRF RFM95W che qui vi presentiamo.

Il nuovo prodotto è disponibile nello shop di Futura Elettronica con il codice FTR2040_LORA868 (**Fig. 1**).

Oltre al modulo LoRa, abbiamo apportato anche numerose migliorie:

- aggiunta di una porta USB con connettore USB-C per l'alimentazione e la programmazione;

Fig. 1
La nostra scheda
FTR2040LoRa.



- aggiunta di un pulsante di accensione e spegnimento;
- aggiunta di un pulsante di reset;
- aggiunta di un pulsante per il BOOTSEL;
- due file di connettori per il collegamento esterno a sensori, attuatori e quant'altro.

LA LIBRERIA LORAWAN

Come già detto nel suddetto corso RP2040, il supporto per la connettività LoRaWAN è stato messo a disposizione da Sandeep Mistry, l'autore della libreria Arduino LoRa, che più recentemente ha fornito anche il supporto LoRa per Pico e altre schede basate su RP2040, utilizzando il modulo radio Semtech SX1276.

Questo rende la nostra scheda FTR2040LoRa completamente compatibile.

Se non avete già configurato la toolchain C/C++ di Raspberry Pi Pico nel vostro Raspberry Pi, fate riferimento alla seconda puntata del corso. Se avete già configurato la toolchain C/C++, clonate il repository dal link seguente:

```
git clone --recurse-submodules https://github.com/sandeepmistry/pico-lorawan.git
```

Assicuratevi di avere il percorso PICO_SDK_PATH impostato prima di procedere.

Per esempio, se state compilando su un Raspberry Pi e avete seguito la seconda puntata, dovrete avere già eseguito il comando seguente (senza spazi prima e dopo il segno di uguale):

```
export PICO_SDK_PATH=/home/pi/pico/pico-sdk
```

REGISTRAZIONE DEL GATEWAY SULLA RETE TTN

Nella terza puntata del corso RP2040 abbiamo spiegato in dettaglio la procedura per la registrazione di un gateway su The Things Network (TTN) e come creare un'applicazione con un end device. Pertanto oggi diamo solo alcune rapide indicazioni riguardanti le operazioni che dovete fare (**Fig. 2**).

1. Creazione di un account su TTN
<https://www.thethingsnetwork.org>
2. Accesso a TTN facendo clic su Sign Up.
3. Accedere alla console e fare clic su Go to gateways.
4. Registrare un gateway. Attenzione! Nel nuovo stack V3 di TTN i gateway a singolo canale non sono più supportati. Per questo motivo dovrete acquistare un gateway disponibile in commercio o costruirvi da soli un gateway multicanale.
5. Digitate l'EUI ovvero un identificatore unico per il gateway. Se usate un gateway iC880A che utilizza il Semtech UDP Packet Forwarder, copiatelo dal file global_conf.json. Per esempio, l'ID di default AA555A0000000000, può diventare semplicemente BB555A0000000000 e deve essere unico.
6. Cliccate su confirm.
7. Inserite un nome qualsiasi per il campo Gateway name.
8. La voce Frequency plan deve essere impostata con Europe 863 - 870 MHz (SF9 for RX2 - recommended).

CREAZIONE DI UN'APPLICAZIONE E DI UN END DEVICE

Facendo clic su Applications della console, potete aggiungere una nuova applicazione (**Fig. 3**).

Fig. 2
Pagina di
registrazione
del gateway.

Register gateway

Register your gateway to enable data traffic between nearby end devices and the network. Learn more in our Gateway Guide.

Gateway EUI

Gateway ID

Gateway name

Frequency plan

☐ Require authenticated connection
Choose this option e.g. if your gateway is powered by LoRa Basic Station.

Share gateway information
Select which information can be seen by other network participants, including Packet Broker.

☒ Share status within network

☒ Share location within network

Ecco un riepilogo rapido delle operazioni che dovete compiere.

1. Inserite un identificativo univoco per Application ID (per esempio, my-rp2040).
2. Inserite un nome qualsiasi per Application name (per esempio my-rp2040).
3. Inserite una descrizione a piacere per il campo Description.
4. Fate clic sul pulsante "Create application".
5. Nella sezione End devices delle Applications fate clic sul collegamento Add end device (**Fig. 4**).
6. Scegliete l'opzione Manually per compilare manualmente i campi come segue:
 - Frequency plan: Europe 863-870 MHz (SF9 for RX2 - recommended).
 - LoRaWAN version: selezionare 1.0.2.
 - Regional Parameters version: RP001 Regional Parameters 1.0.2.
 - Activation mode: lasciare selezionato OTAA.
 - DevEUI: un ID generato automaticamente.
 - AppEUI: riempire l'ID con tutti 0 (zero).
 - AppKey: un chiave generata automaticamente.
 - End device name: un nome qualsiasi, per esempio, rp2040-otaa.
7. Premere sul tasto "Register end device" per registrare il dispositivo.
8. Apparirà una pagina simile alla **Fig. 5**.

SCHEMA ELETTRICO

Prima di procedere vi vogliamo illustrare la nostra scheda FTR2040LoRa iniziando dallo schema elettrico.

Come potete vedere nella breakout sono portati all'esterno i GPIO del chip RP2040, siglato U3 e le linee USB sono collegate direttamente alla porta Type-C. Il clock che scandisce il funzionamento è ricavato grazie al quarzo Q1 e ai condensatori C5 e C6, connessi ai piedini di temporizzazione 20 e 21, che appartengono all'oscillatore principale e cui si connettono i componenti esterni. L'alimentazione per il chip è ricavata dal regolatore a basso drop-out U4, il quale, ricava 3,3V stabilizzati. Grazie ai diodi Schottky (utilizzati per minimizzare la caduta di tensione) D1 e D2, come fonte di alimentazione principale si possono utilizzare i 5V in arrivo dalla connessione USB o quelli provenienti dal contatto 5V.

In altre parole, il circuito può essere alimentato attraverso la porta USB o il pin 5V, ma è anche possibile connettere il cavo USB mentre il circuito è alimentato dai suoi pad di alimentazione, proprio grazie alla presenza degli Schottky, che impediscono ad una tensione di andare sulla linea dell'altra. Il modulo RFM95W è collegato a un porta SPI con

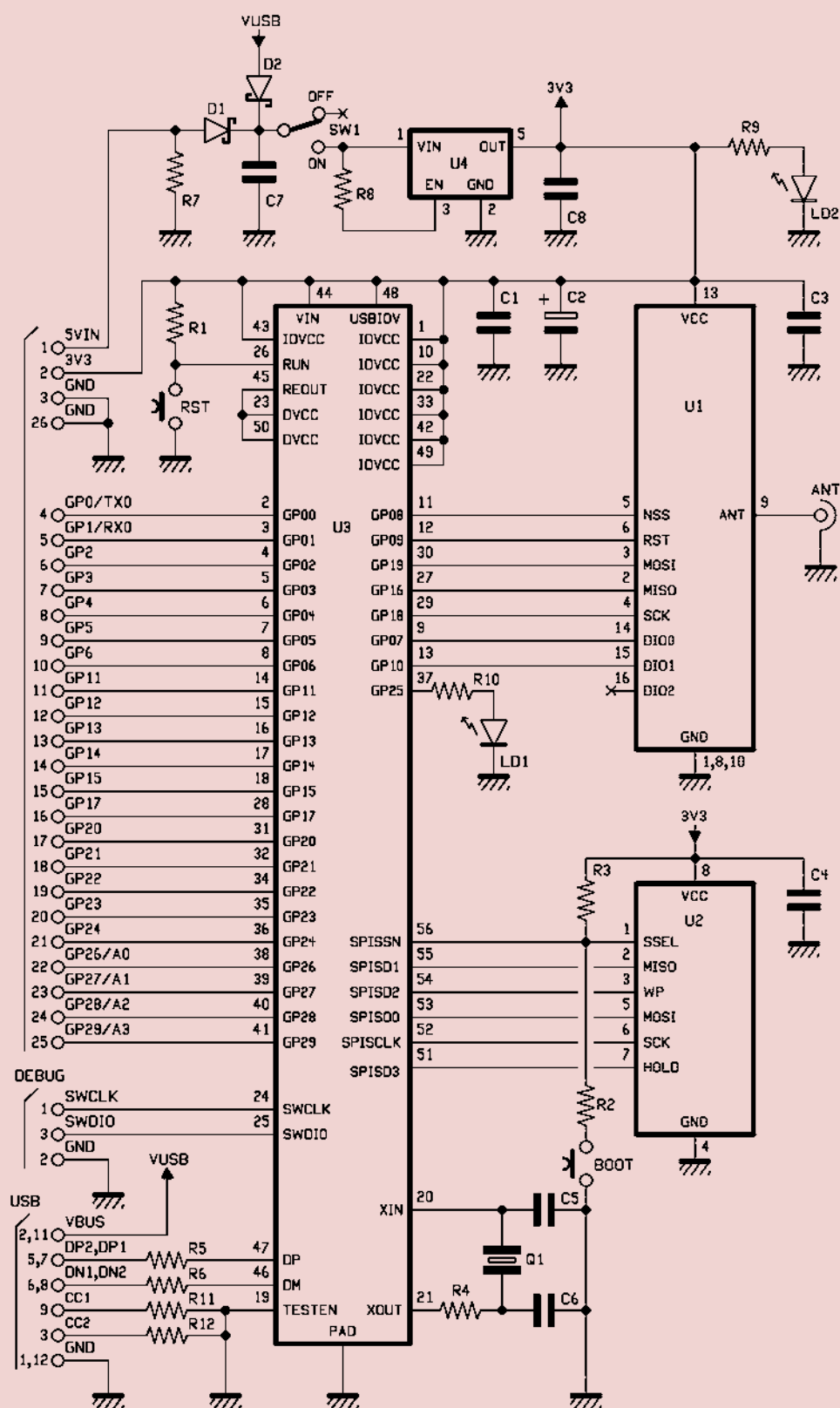
Fig. 3
Pagina Add Application.

Fig. 4
Pagina Manually Register end device.

Fig. 5
Pagina di registrazione dell'end device.

le tipiche connessioni NSS, RST, MOSI, MISO, SCK, rispettivamente ai pin 11, 12, 30, 27 e 29. Le porte DIO0 e DIO1 del modulo sono controllate dai pin 9 e 13 e servono al protocollo LoRa per le funzioni di





TX/RX e per impostare la modalità LoRa. Completiamo l'analisi dello schema elettrico con l'integrato siglato U2, il quale altro non è che l'indispensabile memoria Flash EPROM da 64Mbit ad accesso seriale, con modalità DUAL e QUAD SPI; come vedete, è interfacciata all'RP2040 attraverso una connessione Serial Peripheral Interface a 4 linee più 2 segnali di controllo. Questa memoria Flash serve a caricare i firmware che scriveremo e a contenere eventuali dati da memorizzare a permanenza, utili alle applicazioni che andremo a realizzare con la nostra board, sia durante questo corso, sia in autonomia.

REALIZZAZIONE DELLA SCHEDA FTR2040LORA

Il modulo FTR2040LoRa può essere autocostruito seguendo il piano di montaggio dopo aver realizzato il relativo PCB.

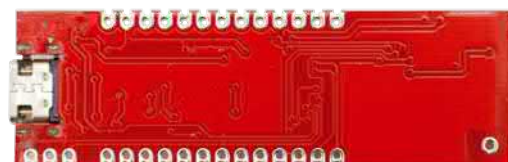
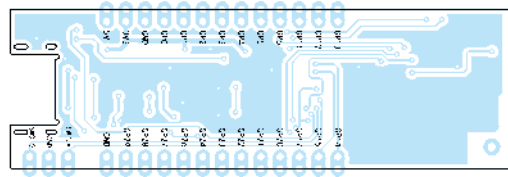
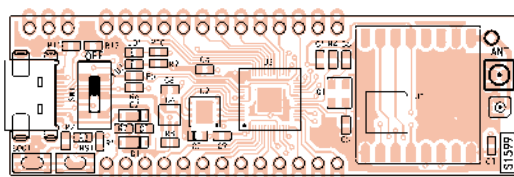
Il circuito stampato è un doppia faccia che richiede

una certa competenza per essere realizzato, ma che potete farvi preparare da un servizio di circuiti stampati on-line come il PCBPRODUCTION di Futura Elettronica.

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

Il montaggio deve iniziare dal chip RP2040, che è il componente più impegnativo e che va posizionato, rispettando il verso indicato nel piano di

piano di MONTAGGIO

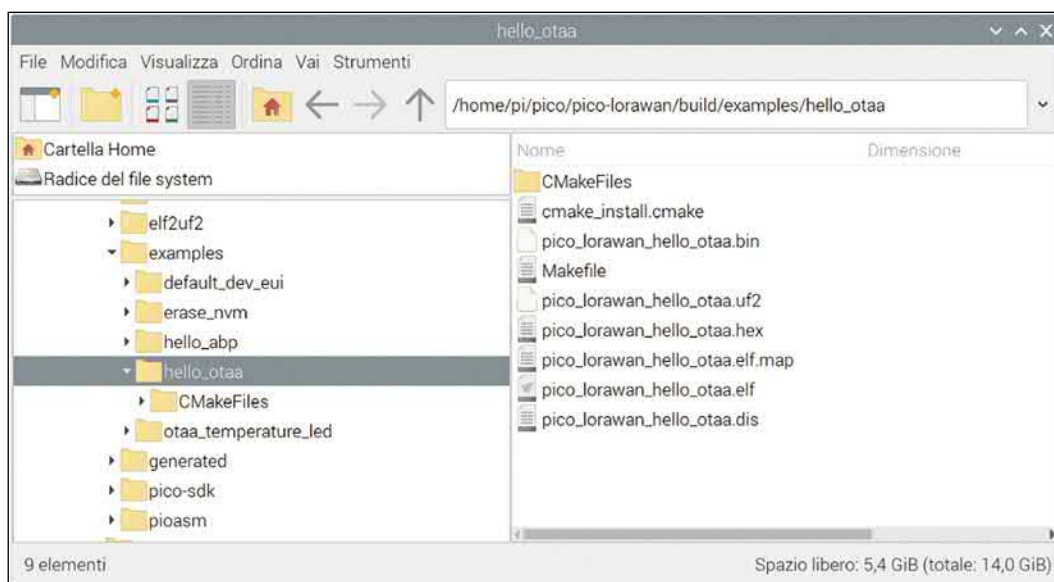


Elenco Componenti:

R1, R7, R8: 10 kohm (0603)
R9, R10: 330 ohm (0603)
R2, R4: 1 kohm (0603)
R5, R6: 27 ohm (0603)
R3: -
R11, R12: 5,1 kohm (0603)
C1, C3, C4, C9: 100 nF ceramic (0603)
C2: 10 µF 6,3 VL tantalum (2012)
C5, C6: 22 pF ceramic (0603)
C7, C8: 10 µF ceramic (0603)
LD1: LED red (0603)

LD2: LED green (0603)
D1, D2: MBR0520LT1G (SOD123)
Q1: Crystal 12MHz (3.2x2.5 mm)
RST, BOOT: PTS810SJK250SMTRLFS
SW1: DPDT slide switch JS202011JCQN
U1: Module RFM95 868 MHz
U2: W25Q64JVXGIQ TR
U3: RP2040
U4: MIC5317-3.3YM5-TR
USB: Molex USB connector 2169900002
ANT: IPEX connector RECE.20369.001E.01

Fig. 6
La cartella con
l'esempio
compilato.



montaggio, ben centrato sulle rispettive piazzole, preventivamente cosparse di stagno e poi di pasta fluxante; fatto ciò, scaldandolo con la stazione ad aria calda, si salderà quando avrete visto lo stagno brillare, allorché toglierete il getto d'aria calda. La saldatura della Flash dovrà essere fatta in modo simile.

Sicuramente più facile è la saldatura del modulo LoRa RFM95W che ha la spaziatura delle piazzole di 2 mm. Per il resto dei componenti, tutti molto semplici, potrete procedere con il saldatore, sempre dopo aver stagnato e cosperso di fluxante le piazzole. Terminato il montaggio e verificato, con l'aiuto della lente d'ingrandimento che sia tutto a posto, la scheda è pronta per l'uso.

Possiamo quindi scordarci l'hardware ed entrare nel vivo della sua programmazione.

Chi, tra voi, non ha l'attrezzatura o non se la sente di costruire da sè il circuito, non deve disperare perché Futura Elettronica (www.futurashop.it) rende disponibile la scheda FTR2040LoRa già montata e collaudata (cod. FTR2040_LORA868).

IL CODICE DI ESEMPIO

Nella stessa cartella della libreria che avete clonato in precedenza, troverete una cartella examples. Al suo interno troverete una sottocartella hello_otaa con dentro tre file:

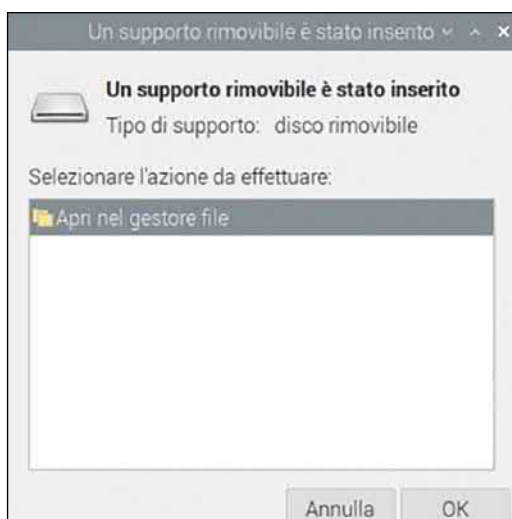
- CMakeLists.txt
- Config.h
- Main.c

Aprirete il file config.h e modificalo in questo modo:

```
#define LORAWAN_REGION      LORAMAC_REGION_EU868
#define LORAWAN_DEVICE_EUI  "70B3D57ED0054DEF"
#define LORAWAN_APP_EUI    "0000000000000000"
#define LORAWAN_APP_KEY    "9CABA5322B-
2630BE471F0C769B81A977"
```

Ovviamente, dovete sostituire le stringhe DEVICE EUI e APP KEY con quelle dell'end device che avete creato in precedenza su TTN (fate riferimento alla **Fig. 5**). La voce APP EUI può rimanere con tutti zero, perché l'attivazione OTAA viene gestita da TTN. Quindi, dopo aver modificato il file config.h, potete andare avanti e creare l'applicazione di esempio per il vostro end device.

Fig. 7
La cartella del
dispositivo.



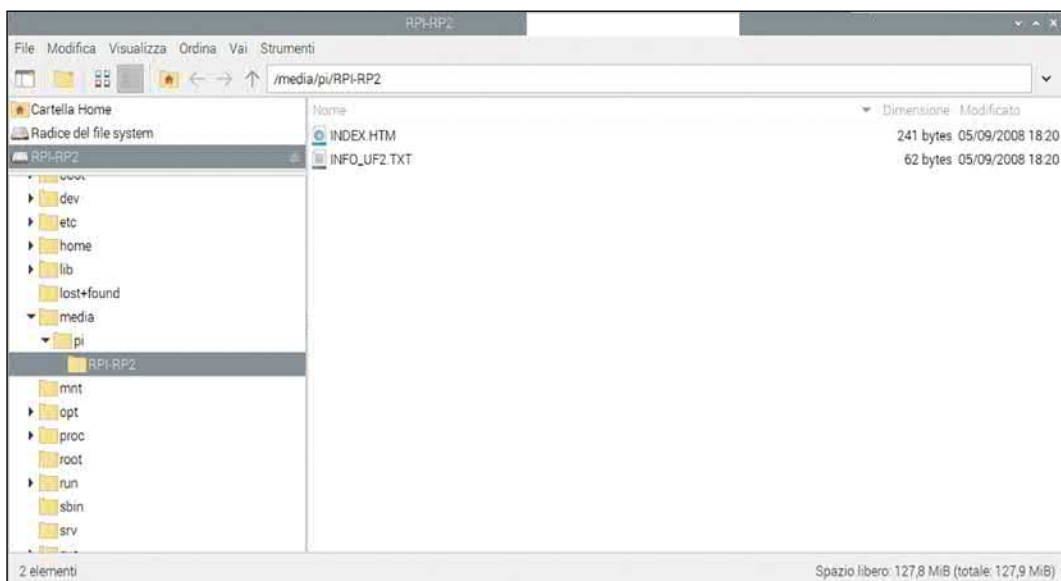


Fig. 8
La cartella
con i file
INDEX.HTM
e INFO_UF2.
TXT.

Per fare questo, entrate nella cartella pico-lorawan, create una cartella build, entrate nella cartella build ed eseguite prima cmake e quindi make.

La compilazione potrebbe richiedere qualche minuto:

```
cd home/pi/pico-lorawan.
mkdir build
cd build
cmake .. -DPICO_BOARD=pico
make
```

Se tutto va bene, dovrete avere il file pico_lorawan_hello_otaa.uf2 nella cartella build/examples/hello_otaa (**Fig. 6**).

A questo punto, potete caricare questo file UF2 nella scheda nel modo consueto, ma con la facilità offerta da questa nostra nuova scheda:

- collegate la scheda tramite un cavetto USB-C;
- spegnete la scheda con il pulsante ON/OFF;
- tenete premuto il pulsante BOOT e accendete la scheda con il pulsante ON/OFF;

In questo modo, si aprirà la cartella del dispositivo (**Fig. 7**).

Nella cartella che apparirà, contenente i due file INDEX.HTM e INFO_UF2.TXT (**Fig. 8**), trascinate al suo interno il suddetto file UF2.

La cartella si chiuderà e il programma si avvierà. Dal terminale potete lanciare il comando seguente

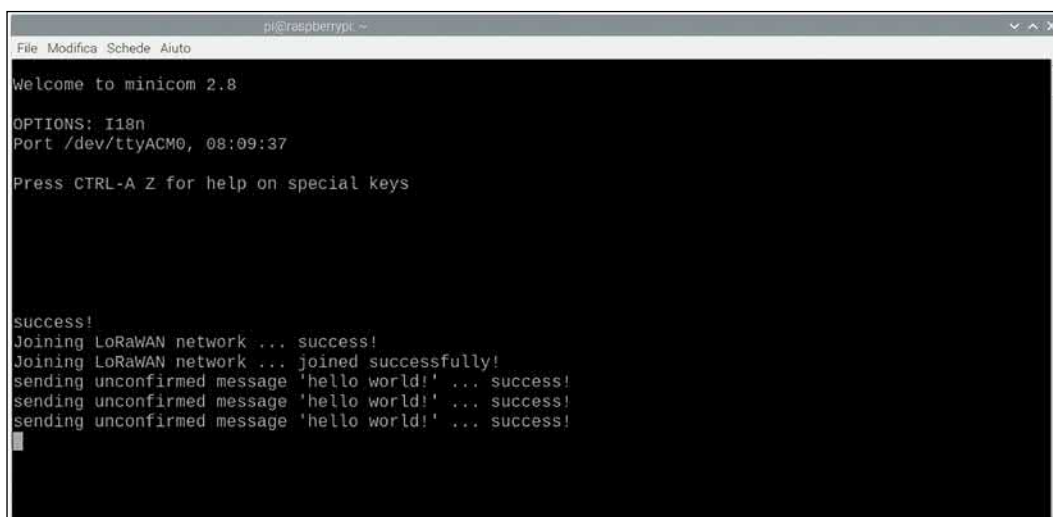


Fig. 9
Il debug del
dispositivo
sul terminale.

Fig. 10
I dati dell'end-
device in arrivo
nell'applicazione
TTN.

per vedere il debug della porta seriale USB:

```
minicom -D /dev/ttyACM0
```

Se il collegamento OTAA ha successo, potrete vedere i messaggi seguenti (**Fig. 10**):

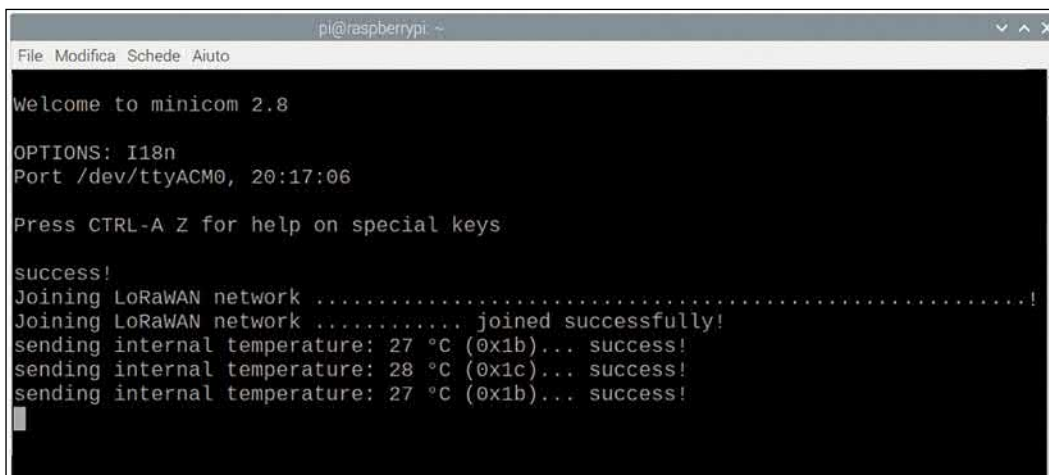
- *success!* (significa che l'inizializzazione del dispositivo ha avuto successo)
- *Joining LoRaWAN network ... joined successfully!* (significa che il collegamento con l'applicazione su TTN ha avuto successo)
- *sending unconfirmed message 'hello world!' ... success!* (significa che il messaggio è stato inviato con successo a TTN)

Essendo l'esempio hello_otaa un collegamento di tipo OTAA (Over The Air Activation) bisognerà

aspettare TTN per il JOIN, cioè la connessione dell'end device all'applicazione. Quando apparirà "success", significa che il dispositivo è collegato all'applicazione e dovreste vedere arrivare i dati nella pagina dell'applicazione e del device (**Fig. 11**). Attenzione! L'esempio hello_otaa prevede di mandare i dati della stringa "hello world!" ogni 5 secondi. Dato che TTN pone delle limitazioni nella Community Edition, ovvero alla versione gratuita, vi consigliamo di portare l'invio ad almeno 60 secondi. Per fare questo, andate alla riga seguente del file main.c nella cartella dell'esempio, e cambiate il valore dei millisecondi fra un messaggio e l'altro portandolo a 60000:

```
if ((now - last_message_time) > 60000)
```

Fig. 11
Il messaggio hello
world! decodificato.



```
pi@raspberrypi ~  
File Modifica Schede Aiuto  
Welcome to minicom 2.8  
OPTIONS: I18n  
Port /dev/ttyACM0, 20:17:06  
Press CTRL-A Z for help on special keys  
success!  
Joining LoRaWAN network .....joined successfully!  
Joining LoRaWAN network .....joined successfully!  
sending internal temperature: 27 °C (0x1b)... success!  
sending internal temperature: 28 °C (0x1c)... success!  
sending internal temperature: 27 °C (0x1b)... success!
```

Fig. 12
Il terminale
con i dati di
temperatura.

DECODIFICA DEI MESSAGGI

I messaggi inviati dal dispositivo sono ovviamente una serie di byte incomprensibili all'uomo. Se andate nella pagina Live data del device e fate clic su un messaggio, dovreste vedere la stringa hello world! esattamente in questo modo:

68 65 6c 6c 6f 20 77 6f 72 6c 64 21

... ovvero la serie di numeri esadecimali corrispondenti ai valori ASCII della stringa hello world!. Nella puntata 4 del corso RP2040 viene spiegato in dettaglio come decodificare in automatico i messaggi del dispositivo. Per il momento potete decodificare il messaggio manualmente copiando il messaggio e andando in uno dei tanti siti dedicati alla conversione hex to string, come per esempio <https://codebeautify.org/hex-string-converter>. Dovreste vedere il messaggio decodificato come nella **Fig. 11**.

VISUALIZZARE I DATI IN NODE-RED

Se avete seguito il corso RP2040 o il corso Node-RED della nostra rivista sapete già come fare. Per vostra comodità riassumiamo in breve come poter visualizzare, per esempio, i dati di temperatura del sensore interno al chip RP2040 tramite il browser di Node-RED.

Il codice di esempio per la scheda si trova nella stessa cartella examples che avete clonato. Modificate l'esempio otaa_temperature_led seguendo la stessa procedura che abbiamo già spiegato. Dalla cartella build copiate l'esempio pico_lora-
wan_otaa_temperature_led.uf2 nella scheda esattamente come prima.
Dopo il Join OTAA, dovreste vedere arrivare nel terminale i dati di temperatura (**Fig. 12**):

```
Joining LoRaWAN network .....joined successfully!  
sending internal temperature: 27 °C (0x1b)... success!  
sending internal temperature: 28 °C (0x1c)... success!  
sending internal temperature: 27 °C (0x1b)... success!  
sending internal temperature: 27 °C (0x1b)... success!  
sending internal temperature: 27 °C (0x1b)... success!
```

INTEGRAZIONE MQTT E NODE-RED

Prima di visualizzare i dati con Node-RED, è necessario attivare l'integrazione MQTT in TTN.

Se aprite il menu a sinistra alla voce Integrations, fra le numerose voci, troverete anche l'integrazione MQTT.

Facendo clic su MQTT, si aprirà una pagina simile alla **Fig. 13**.

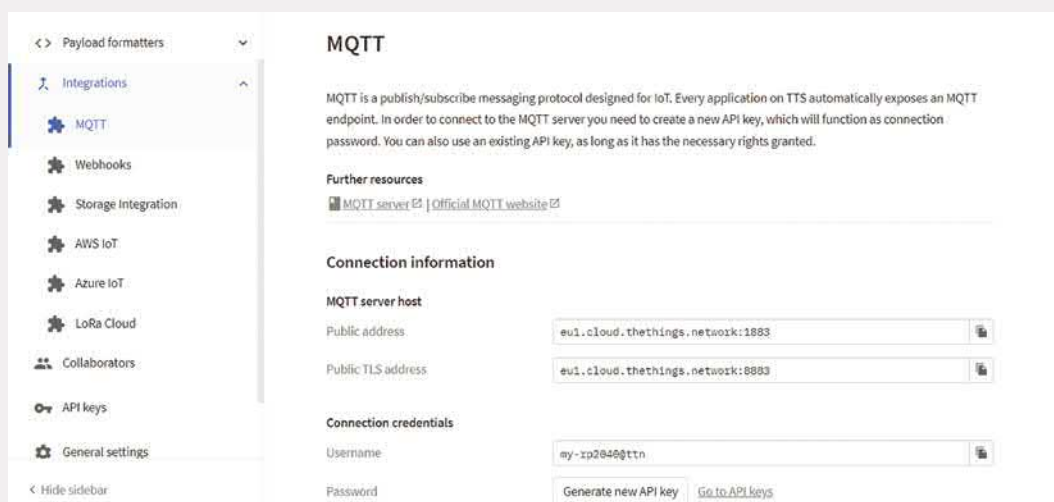
Qui troverete tutte le informazioni da inserire in seguito in Node-RED per l'integrazione MQTT di TTN.

```
MQTT server host  
Public address eu1.cloud.thethings.  
network:1883  
Public TLS address eu1.cloud.thethings.network:8883  
Connection credentials  
Username my-rp2040@ttn  
Password Generate new API key
```

Oltre all'indirizzo del server MQTT e la porta di connessione, vengono date le credenziali di accesso, che sono relative al nome utente, ovvero il nome dell'applicazione che state utilizzando seguito da @ttn, e la password che viene generata tramite il pulsante Generate new API key.
Dovrete copiare queste informazioni dopo aver creato un flow su Node-RED, come spieghiamo nel prossimo paragrafo.



Fig. 13
Pagina Integrations
per l'integrazione
MQTT di TTN.



MQTT E NODE-RED

Per approfondire l'uso di Node-RED vi consigliamo il corso Node-RED pubblicato su Elettronica In dal numero 258 al numero 265.

Per la connessione con TTN, dovete innanzitutto creare un nodo MQTT In, cioè un nodo che rimarrà in ascolto dei dati del broker MQTT di TTN. In un flow di Node-RED trascinate il nodo MQTT In (**Fig. 14**) dalla barra laterale sinistra ed editate il contenuto in questo modo:

Server: Add new mqtt-broker...
Action: Subscribe to single topic
Topic: # (significa la sottoscrizione a qualsiasi topic)
QoS: 2
Output: auto-detect (string or buffer)

Name: mqtt TTN (o qualsiasi altro nome)

Facendo clic sull'icona matita del menu Add new mqtt-broker... si aprirà una finestra in cui dovete inserire le informazioni dell'integrazione MQTT di TTN, viste prima.

Nel tab Connection inserite l'indirizzo del broker TTN e un nome qualsiasi (**Fig. 15**):

Name: ttn broker

Server: eu1.cloud.thethings.network:1883 Port 1883

Nel tab Security inserite Username e password generati da TTN (**Fig. 16**).

Username: my-rp2040@ttn

Password: ***** (l'API key generata da TTN)

Fig. 15
Il tab Connection
del nodo MQTT.

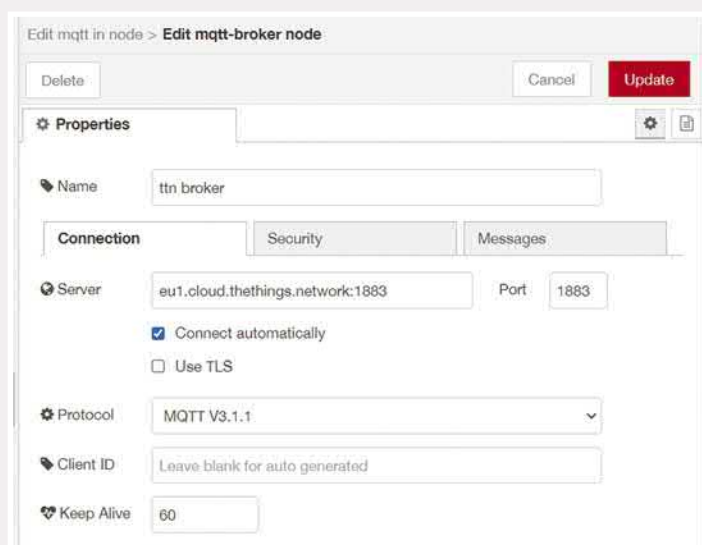


Fig. 14
Il nodo MQTT.



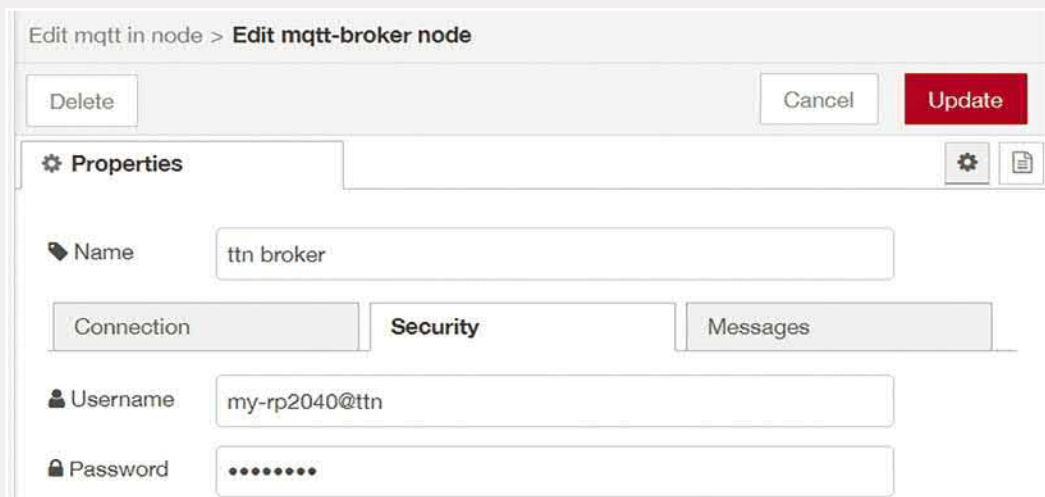


Fig. 16
Il tab
Security del
nodo MQTT.

Fate clic Update e Done per chiudere l'editor e quindi su Deploy per distribuire il nodo.

Dovreste vedere la connessione attivarsi, ovvero visualizzare connected in verde in corrispondenza del nodo MQTT In.

Senza entrare nei dettagli, possiamo solo dirvi che i dati del nodo vanno convertiti in un oggetto JSON da un nodo json collegato all'uscita del nodo MQTT In. Collegando un nodo debug all'uscita del nodo json dovreste vedere nella finestra di debug qualcosa di simile alla **Fig. 17**.

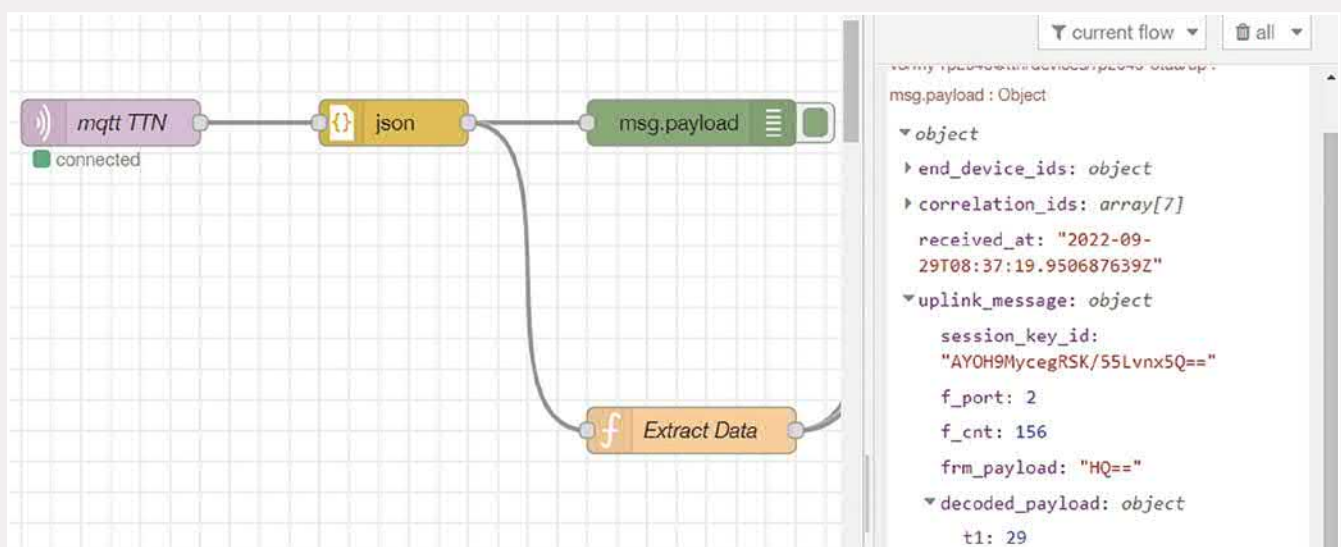
Sono tutti i dati provenienti dall'end device, compreso la coppia chiave-valore che abbiamo decodificato con il Payload formatter. Il Payload formatter è una funzione Javascript che

va inserita nella pagina Payload formatter dell'end device. Una volta aperta la pagina Payload formatters dovreste vedere qualcosa di simile alla **Fig. 18**. Nella finestra Custom Javascript formatter copiate il codice che riportiamo nel **Listato 1** e che potete trovare nella nostra pagina di download (payload_formatter.txt).

A questo punto possiamo collegare un nodo function e scrivere una funzione che "isoli" solo il dato che ci interessa, ovvero il valore di temperatura. Per fare questo, all'interno del nodo function scriviamo una funzione che trovate nel **Listato 2**.

Nella variabile obj creiamo una coppia chiave-valore composta dalla chiave "temp" e dal valore msg.payload.uplink_message.decoded_

Fig. 17
Il debug di
Node-RED.



Listato 1

```
function Decoder(bytes, port) {
  var decoded = {};
  decoded.t1 = bytes[0]; // assegnazione del byte di temperatura alla variabile
  return decoded;
}

function decodeUplink(input) {
  var data = input.bytes;
  var valid = true;

  if (typeof Decoder === "function") {
    data = Decoder(data, input.fPort);
  }

  if (typeof Converter === "function") {
    data = Converter(data, input.fPort);
  }

  if (typeof Validator === "function") {
    valid = Validator(data, input.fPort);
  }

  if (valid) {
    return {
      data: data
    };
  } else {
    return {
      data: {},
      errors: ["Invalid data received"]
    };
  }
}
```

payload.t1, ovvero dal valore di t1, che corrisponde alla chiave di temperatura.

Con la variabile `var res_temp = obj["temp"]` assegniamo il valore decodificato della temperatura e con `var msg1 = { payload:res_temp }` creiamo l'oggetto da restituire con `return [msg1]`. Collegando un nodo debug all'uscita del nodo function dovreste vedere solo il valore di temperatura (**Fig. 19**).

A questo punto, per visualizzare il valore di

temperatura nella dashboard di Node-RED, basta aggiungere un nodo gauge e collegarlo all'uscita del nodo function.

Ovviamente dovreste creare anche un gruppo e tab per visualizzare il nodo gauge nella dashboard. Andando all'indirizzo della dashboard (solitamente 127.0.0.1:1880/ui) dovreste vedere qualcosa di simile alla **Fig. 20**.

Ogni qualvolta il valore di temperatura del vostro end device cambierà, vedrete l'ago del gauge muo-

Listato 2

```
var obj =
{
  "temp":msg.payload.uplink_message.decoded_payload.t1
}

var res_temp = obj["temp"]
var msg1 = { payload:res_temp };

return [msg1]
```



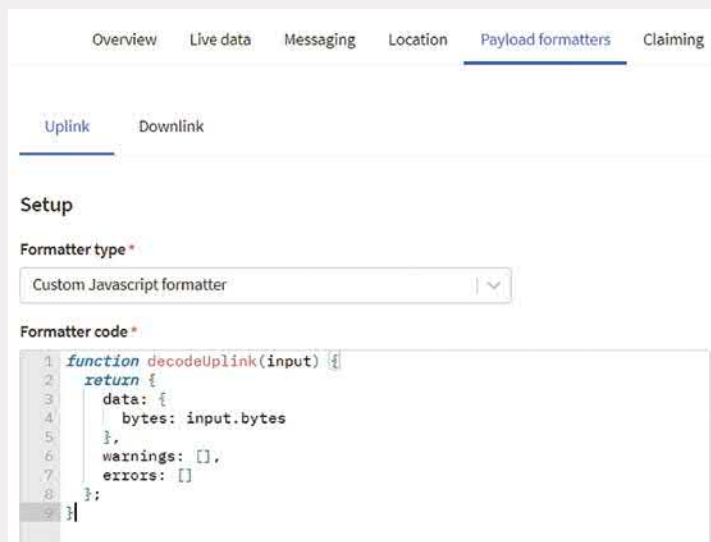


Fig. 18
La pagina Payload formatters dell'end device.

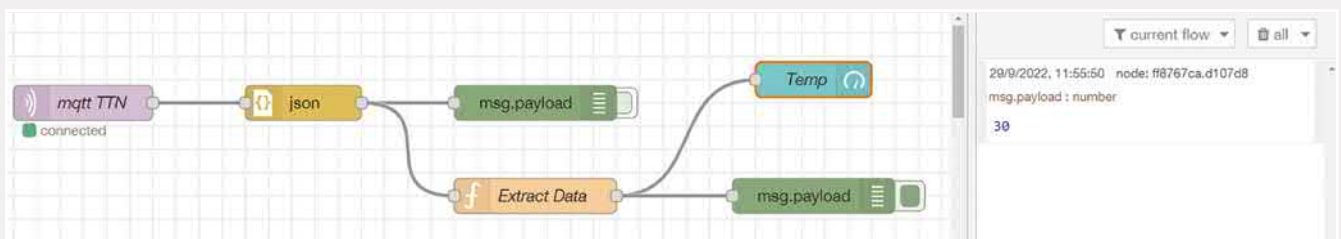


Fig. 19
Il debug di Node-RED con solo il valore di temperatura.



Fig. 20
I dati di temperatura nel terminale.

versi e indicare la temperatura esattamente come viene rilevata.
Il flow di Node-RED pronto all'uso è disponibile come risorsa scaricabile gratuitamente.

CONCLUSIONI

Con la speranza di essere stati sufficientemente chiari, vi invitiamo a sperimentare con la nostra scheda FTR2040LoRa e a costruire la vostra rete LoRaWAN senza sforzo.



Cosa occorre?

La breakout board con il chip RP2040 e il modulo LoRa 868 MHz (cod. FTR2040_LORA868) è in vendita a Euro 28,00 mentre la breakout senza il modulo radio (cod. FTR2040) è disponibile a Euro 13,90. Il microcontrollore RP2040 (cod. RP2040) è in vendita a 2,50 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

Dai la giusta tensione al tuo progetto.

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

STEP UP

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

cod. MODSTEPUP

€ 5,00



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

cod. BOOSTDCDC

€ 25,00



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

cod. YXA347

€ 6,00



STEP DOWN

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

cod. STEPDOWNDISP

€ 6,90



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

cod. DCDCSTEPDOWN

€ 6,50



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

cod. YB728

€ 14,00



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

cod. STEP5V32

€ 5,90



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

cod. MODULODCDC

€ 4,30



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

cod. YM355

€ 5,00



STEP UP/DOWN

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

cod. STEPUPDOWN

€ 16,00



BREAKOUT BOARD

DRIVER LED DIODES

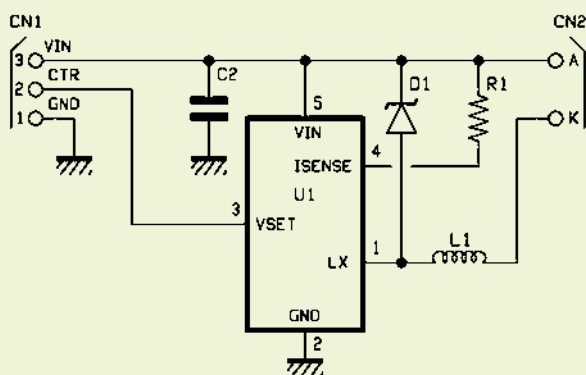
AL8861



dell'Ing. **VINCENZO GERMANO**

I LED sono diventati sempre più comuni nell'elettronica di consumo, ma per farli funzionare in modo efficiente e sicuro, è necessario utilizzare un driver LED.

Per quanto i LED (dall'acronimo anglosassone Light Emitting Diodes ovvero diodo a emissione di luce) hanno una storia ormai di varie decine di anni, il loro pieno potenziale si è realizzato solo negli anni sessanta da ricercatori e ingegneri che cominciarono a concentrarsi sulla realizzazione e poi industrializzazione di LED più efficienti. Ad oggi questo componente viene comunemente utilizzato nella maggior parte di dispositivi elettronici, solo per citarne alcuni, li ritroviamo nei display di dispositivi tipo computer, TV, tablet, smartphone, semafori, fotocamere digitali, elettronica di consumo, illuminazione ambientale, domotica e molto altro ancora. Com'è facile immaginare non ha filamenti né tantomeno gas (come nelle vecchie lampadine a filamento) ma è un



dispositivo a semiconduttore che va ad emettere luce nello spettro del visibile (e non solo) quando riceve tensione tra i suoi due terminali (o pin) esterni; semplificando estremamente, con una polarizzazione di tensione "positiva" tra i due terminali, si polarizza positivamente la sua giunzione interna e viene generata luce. I vantaggi di questa tecnologia rispetto al filamento è indubbia, offrendo un consumo energetico molto basso, una lunga durata d'utilizzo e sicuramente maggior "rispetto" verso l'ambiente.

Tuttavia, diversamente dall'illuminazione tradizionale, uno degli elementi più importanti nel funzionamento dei LED sono i driver. Dato che i LED non vengono collegati direttamente alla corrente elettrica, per il loro pilotaggio hanno la necessità di essere gestiti mediante un dispositivo elettronico identificato come driver LED, che adatta tensione e corrente alle necessità di utilizzo del particolare LED in base alle caratteristiche richieste.

Andando a considerare il settore dell'illuminazione e non la gestione magari di un singolo LED di segnalazione, essendo i LED un tipo di carico molto particolare, questa funzione apparentemente semplice in realtà non lo è, richiedendo lo sviluppo di soluzioni non banali da progettare e gestire caso per caso. Due principali fattori complicano ulteriormente lo scenario, il primo riguarda il panorama di questo settore in continua evoluzione dovendo proporre un'elettronica di pilotaggio sempre al passo con nuovi sviluppo e tecnologie LED, il secondo fattore è la moltitudine di differenti soluzioni per il pilotaggio di LED mediante le proposte più svariate di tipologie di driver con caratteristiche e requisiti specifici da valutare con attenzione in base al progetto d'interesse. Prima di affrontare le caratteristiche principali del driver della Diodes AL8861 e la breakout board

associata, vediamo a grandi linee le caratteristiche principali dei driver LED.

TIPOLOGIE E CARATTERISTICHE PRINCIPALI DEI DRIVER LED

Per quanto di distinzioni e classificazioni se ne possono fare molte sulla tipologia di driver, possiamo generalizzare considerando che sul mercato si possono distinguere due macro tipologie differenti di driver LED, quella a corrente costante e quella a tensione costante, che, com'è facile immaginare, hanno applicazioni e utilizzi diversi tra loro. I primi sono progettati per LED che richiedono una corrente fissa (costante) ma possono sopportare una gamma di tensioni differenti, che trovano maggiore applicazione per alimentare faretti a LED o ad alta potenza.

Mentre il driver a tensione costante, è progettato per LED che richiedono una tensione fissa e una corrente regolabile; generalmente vengono utilizzati per strisce LED, illuminazione d'ambiente e applicazioni affini in cui si ha una tensione costante per mantenere sia il colore che l'intensità dell'illuminazione.

Possiamo considerare che la maggior parte degli usi a LED sono a bassa tensione, tipo le strisce LED utilizzate per l'illuminazione ambientale che tendenzialmente sono a 12V o 24V (ovviamente bisogna associare il corretto driver con la tensione corretta per poterle gestire) ma oltre alla caratteristica in tensione bisogna tener presente anche i Watt di un driver, che stanno ad indicare la sua capacità massima in uscita (come relazione corrente-tensione). In generale si scelgono driver in grado di erogare una potenza superiore di almeno il 10% rispetto alla potenza del LED o della striscia di LED alimentata, questo per garantire anche una maggiore durata del sistema. Tuttavia per quanto la potenza indica la quantità di energia consumata durante l'utilizzo, non conviene scegliere un driver con potenza troppo alta per evitare deterioramenti/danneggiamenti nel tempo.

IL DRIVER LED AL8861

Tralasciando ulteriori dettagli legati ai driver in generale, procediamo con un focus particolare sul part number della Diodes AL8861.

Questo driver LED è un convertitore step-down DC-DC con isteresi; questa tipologia di convertitori sono utilizzati in un'ampia gamma di sistemi nella nostra vita quotidiana e il loro obiettivo principale è aumentare o diminuire la tensione in continua in base all'applicazione in questione, fornendo

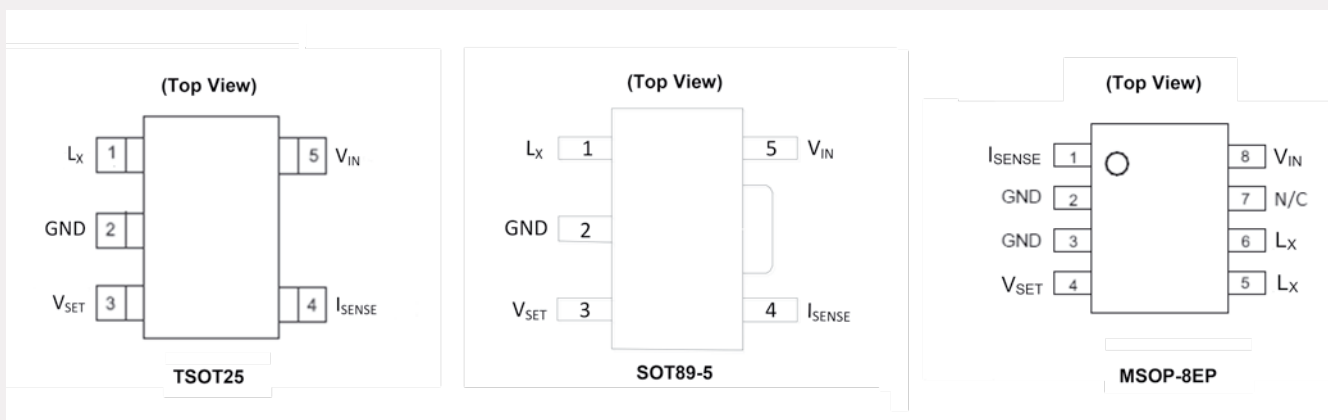


Fig. 1
AL8861, tipologie di package

al contempo anche la regolazione della tensione. Nel particolare, il codice Diodes è uno step-down, perciò riduce in uscita una tensione in ingresso all'interno di un dato range più alto, in questo caso il dispositivo può funzionare con tensioni di input tra 4,5V e 40V.

Progettato per la gestione di LED singoli o multipli, riesce a pilotarli in modo efficiente a partire da una tensione superiore a quella richiesta dal/dai LED, fornendo una corrente di uscita regolabile esternamente fino ad arrivare a 1A per il package TSOT25 e 1,5A per i due package SOT89-5 e MSOP-8EP (con riferimento alla **Fig. 1**). Ovviamente la scelta del package dipende dalla corrente erogabile ma anche dall'ingombro del componente sul circuito stampato e soprattutto dalla capacità di dissipazione di potenza necessaria. Infatti è consigliabile utilizzare la versione SOT89-5 con correnti LED più elevate e circuiti stampati più piccoli per massimizzare la corrente LED consentita in un intervallo di temperatura ambiente più ampio.

Riguardo la potenza erogata, a seconda della tensione di alimentazione e dei componenti esterni con cui è progettato, questo convertitore può fornire fino a 40W di potenza di uscita.

L'AL8861 integra l'interruttore di alimentazione (integrated power switch) e un circuito di rilevamento della corrente di uscita, impostabile mediante una resistenza esterna che configura la corrente di uscita media nominale. Ulteriori caratteristiche salienti sono la possibilità di dimmerare mediante un segnale PWM (Pulse Width Modulation o modulazione a larghezza d'impulso) di controllo sul pin VSET che può essere anche gestito in continua nel caso non si voglia il dimming. Inoltre si può gestire il tempo di soft start mediante il valore di un condensatore esterno tra il pin di VSET e il GND. Infine l'applicazione di una tensione di 0,2V

o inferiore al pin VSET può disattivare l'uscita e far entrare il dispositivo nello stato di standby con un basso consumo di corrente.

Con riferimento alla **Fig. 2** in cui è riportato un possibile schema con la componentistica esterna per il pilotaggio del driver, è possibile notare come il pin Lx (1 nel TSOT e 5,6 nel MSOP) è relativo al drain del NMOS integrato nel driver per il pilotaggio dei LED (ad esso va collegato il catodo del LED).

Il pin VIN è relativo alla tensione d'ingresso ed ha la necessità di un condensatore di disaccoppiamento C1 da 10uF o maggiore (collocato vicino al dispositivo).

Il pin Isense che grazie a un resistore Rs (tra questo pin a VIN) permette di definire la corrente di uscita media nominale, infine il pin VSET che è multifunzionale, ovvero:

- se viene lasciato non collegato il driver avrà un comportamento di normale funzionamento;
- con una tensione inferiore a 0,2V disattiva la corrente di uscita;
- con una tensione continua ($0.3V < VSET < 2.5V$) viene regolata la corrente di uscita da 0 a 100% rispetto al nominale (regolazione lineare dal 5% al 100% del nominale);

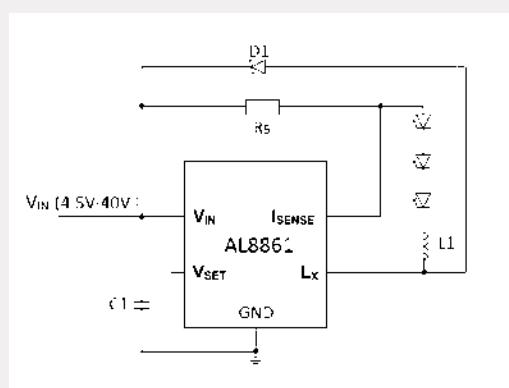
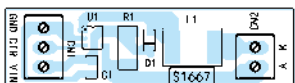


Fig. 2
AL8861, possibile componentistica esterna



Elenco Componenti:

- R1 0,1 ohm (2010)
- C1 2,2 μ F 50 VL ceramico (0805)
- D1 CDBVF240-HF
- U1 AL8861WT-7
- L1 Induttanza 22 μ H 2A

Varie:

- Morsetto 2 vie 2.54 mm
- Morsetto 3 vie 2.54 mm
- Circuito stampato S1667 (39x11 mm)

- con segnale PWM si riesce a regolare l'uscita (regolazione lineare da 1% a 100% del nominale per $f < 500\text{Hz}$)
- se si prevede un condensatore da questo pin a GND, si può aumentare il tempo di soft-start, ovvero un avvio graduale, il tempo predefinito è di 0,1 ms, mentre il tempo di avvio graduale aggiuntivo è di circa 1,5 ms/1 nF

Passiamo a questo punto all'analisi della breakout board acquistabile su futuranet.it con il codice FT1667 per comprenderne la sua implementazione e il suo funzionamento.

IL DRIVER LED

Facendo riferimento allo schema elettrico, possiamo notare come siano stati riportati tutti i compo-

nenti necessari per il corretto utilizzo del driver e collegato il pin VSET sul connettore CN1 pin 2 CTR, in modo tale che l'utente possa utilizzare tutte le funzionalità appena descritte.

Mentre sul connettore CN2 è possibile collegare uno o più LED in cascata.

Durante il normale funzionamento, quando viene applicata la normale tensione di ingresso su VIN (pin 3 del CN1), l'interruttore all'interno del driver AL8861 si accende e la corrente inizia a fluire attraverso il resistore di rilevamento R1, l'induttore L1 e i LED collegati al connettore CN2.

La corrente comincia ad aumentare linearmente e la velocità di rampa è determinata dalla tensione di ingresso VIN e dall'induttore L1, producendo una rampa di tensione attraverso R1. Il circuito interno dell'AL8861 rileva la tensione su R1 e applica una tensione proporzionale all'ingresso del comparatore interno. Quando questa tensione raggiunge una soglia superiore impostata internamente, l'interruttore interno viene disattivato. La corrente dell'induttore continua a fluire attraverso R1, L1, i LED e il diodo D1 e torna alla linea di alimentazione, ma decade con la velocità determinata dalla caduta di tensione diretta dei LED e del diodo D1. Questa corrente in diminuzione produce una tensione in caduta su R1, che viene rilevata dall'AL8861. All'ingresso del comparatore interno verrà applicata una tensione proporzionale alla tensione di rilevamento su R1. Quando questa tensione scende alla soglia inferiore impostata internamente, l'interruttore interno viene riattivato. Questo ciclo di accensione e spegnimento continua a fornire la corrente LED media impostata dal resistore di rilevamento R1.

Come appena visto, la corrente di uscita media nominale nei LED è determinata dal valore del resistore di rilevamento della corrente esterna (R1) collegato tra VIN e ISENSE ed è data dalla seguente relazione $I_{out(nominale)} = 0,1/R1$; la **Tabella 1** fornisce i valori della corrente di uscita media nominale per diversi valori preferiti del resistore di impostazione della corrente (R1), nel caso specifico della breakout board S1667 la corrente è stata impostata su 1A, quindi con resistenza 0,1Ohm.

Questi valori presuppongono che il pin VSET sia non collegato e ad una tensione di riferimento nominale per il comparatore interno. È possibile utilizzare diversi valori di R1 se il Pin VSET è pilotato da un segnale di regolazione esterno. Considerando l'applicazione di una tensione DC da 0,3 V a 2,5 V sul pin VSET, il driver può regolare

Tabella 1
Valori di R1
configurabili
rispetto la corrente
di uscita.

RS (Ω)	NOMINAL AVERAGE OUTPUT CURRENT (mA)
0.066	1,500
0.1	1,000
0.13	760
0.15	667
0.3	333

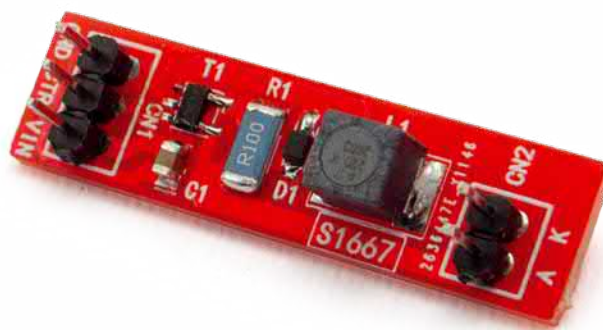
la corrente di uscita dallo 0 al 100% del nominale, come mostrato nella **Fig. 3**, tuttavia l'intervallo di regolazione consigliato va dal 5% al 100%. Se il Pin VSET viene portato al di sopra di 2,5V, la corrente del LED verrà bloccata al 100% del nominale mentre se la tensione VSET scende al di sotto della soglia di 0,3V, l'interruttore di uscita si spegnerà. In più, la corrente del LED può essere regolata digitalmente applicando un segnale logico modulato in larghezza di impulso (PWM) a bassa frequenza al pin VSET per accendere e spegnere il dispositivo. Ciò produrrà una corrente di uscita media proporzionale al ciclo di lavoro del segnale di controllo. Per ottenere una risoluzione elevata, è raccomandabile che la frequenza PWM sia inferiore a 500 Hz, tuttavia è possibile utilizzare frequenze di regolazione più elevate a scapito della gamma dinamica e della precisione della regolazione. Tipicamente, per una frequenza PWM di 500Hz la precisione è migliore dell'1% per PWM che va dall'1% al 100%. La precisione del dimming a basso ciclo di lavoro è influenzata sia dalla frequenza PWM che dalla frequenza di commutazione dell'AL8861; perciò per una migliore precisione/risoluzione, la frequenza di commutazione dovrebbe essere aumentata mentre la frequenza PWM dovrebbe essere ridotta.

Qualche ulteriore considerazione sulla selezione dei componenti è, per la massima efficienza e prestazioni, il diodo D1 è stato scelto come diodo Schottky veloce a bassa capacità con bassa corrente di dispersione inversa che fornisce inoltre una migliore efficienza rispetto ai diodi al silicio, grazie alla minore tensione diretta e al ridotto tempo di ripristino.

La maggiore tensione diretta e l'overshoot dovuto al tempo di recupero inverso nei diodi al silicio possono aumentare la tensione di picco sull'uscita Lx, perciò se veniva utilizzato un diodo al silicio era necessario prestare maggiore attenzione per garantire che la tensione totale sul pin Lx (inclusa l'ondulazione dell'alimentazione), non superasse il valore massimo specificato.

Infine rispetto la scelta del condensatore C1, è stato selezionato un condensatore a bassa ESR (Equivalent Series Resistance) per il disaccoppiamento dell'ingresso, poiché l'ESR di questo condensatore appare in serie con l'impedenza della sorgente di alimentazione e può ridurre l'efficienza complessiva.

E' possibile utilizzare il driver LED con Arduino per controllare la luminosità di un led sfruttando un'uscita PWM.



CONCLUSIONI

Abbiamo visto a grandi linee cosa sono i driver LED, analizzate le loro caratteristiche principali per poi fare un focus diretto su un componente della Diodes, ovvero il driver LED AL8861 che è stato utilizzato per la realizzazione della breakout board S1667, acquistabile sul sito futuranet.it. Grazie a questa board è possibile pilotare LED con il semplice ausilio di un Arduino e di qualche rigo di codice come visto precedentemente. ■

Fig. 3
Il driver LED con
AL8861



Cosa occorre?

Il driver LED (cod. FT1667M) è disponibile già montato a Euro 8,50 IVA compresa.

Il materiale va richiesto a:

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

Entra nel mondo dell'elettronica!

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

Interruttore Crepuscolare



cod. FK001 | € 9,00



Modulo Intermittenza



cod. FK002 | € 12,00

Alimentatore stabilizzato da 1,5 a 35 V



cod. FK003 | € 10,50

Alimentatore stabilizzato duale e regolabile

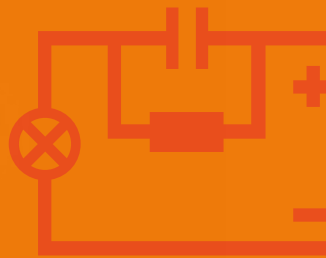


cod. FK004 | € 14,00

Timer Start e Stop da 1 a 240 secondi



cod. FK005 | € 14,00



Interruttore Battimani



cod. FK006 | € 14,00



Controllo automatico ventilazione



cod. FK007 | € 14,00

Regolatore di velocità per ventole



cod. FK008 | € 17,80



Luce psichedelica a LED



cod. FK009 | € 12,00



Protezione bassa tensione



cod. FK010 | € 11,50

Regolatore PWM per pannelli fotovoltaici 12/24V



cod. FK011 | € 18,00

GANIMEDE.E12

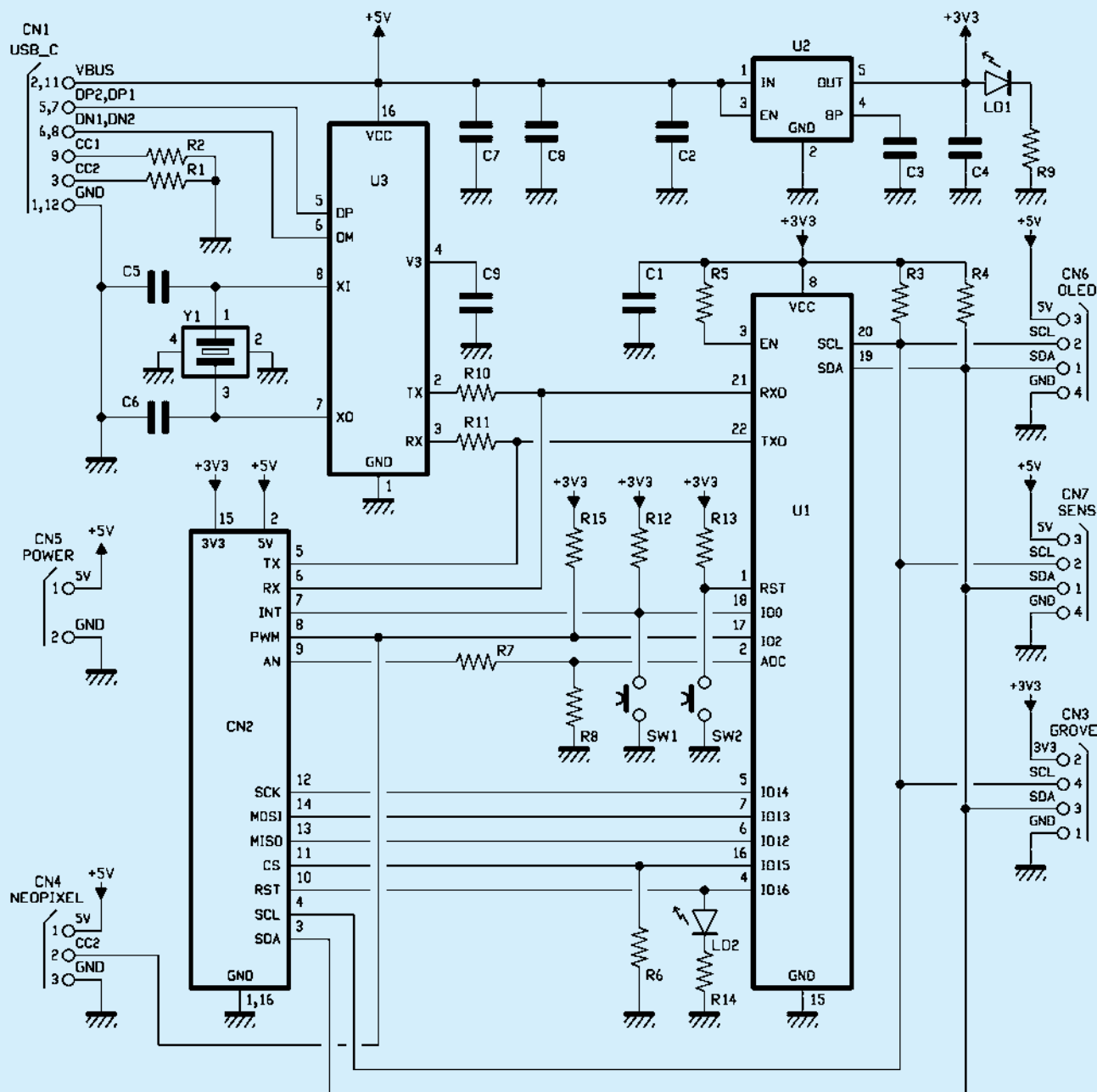
di FRANCESCO FICILI



Versatile ed economica scheda di sviluppo basata su ESP12, dotata di USB, WiFi, slot mikroBus e Grove. Programmabile tramite IDE Arduino, Micropython o Javascript.

N

egli ultimi anni abbiamo assistito ad una importante proliferazione di nuovi strumenti di sviluppo, che hanno sempre di più allargato il perimetro delle possibilità progettuali dei Maker. Questi nuovi strumenti hanno consentito di migrare il range applicativo dai semplici progettini con LED e pulsanti, verso sistemi sempre più complessi e connessi. Se Arduino è sicuramente definibile come il capostipite di questa famiglia di strumenti, sistemi di sviluppo come NodeMCU, ESP32 Thing e Raspberry Pi Pico ne costituiscono sicuramente la nuova generazione. Le schede citate in precedenza sono tutti strumenti validi, ma spesso presentano il difetto di non supportare un ben definito standard di interfacciamento hardware, esponendo semplicemente quanti più pin



possibili verso l'esterno tramite delle comuni strip a passo 2,54 mm. Questa configurazione hardware richiede quindi che l'utente utilizzi collegamenti cablati tramite cavetti tipo dupont e breadboard per la realizzazione delle applicazioni finali basate su queste schede, portando spesso a realizzazioni hardware molto sommarie, ingombranti e scomode da gestire sul campo (**Fig. 1**). Per ovviare a questo problema abbiamo pensato di realizzare una semplice scheda di sviluppo che

utilizzasse standard di interfacciamento hardware comuni e per i quali esistono già svariate breakout board con i vari sensori ed attuatori necessari per i tipici progetti hobbistici. Abbiamo inoltre voluto che questa scheda mantenesse le caratteristiche dei sistemi descritti in precedenza, e quindi che fosse economica, semplice da utilizzare, dotata di interfacciamento USB e WiFi ed in grado di supportare una vasta gamma di linguaggi di programmazione e frameworks.

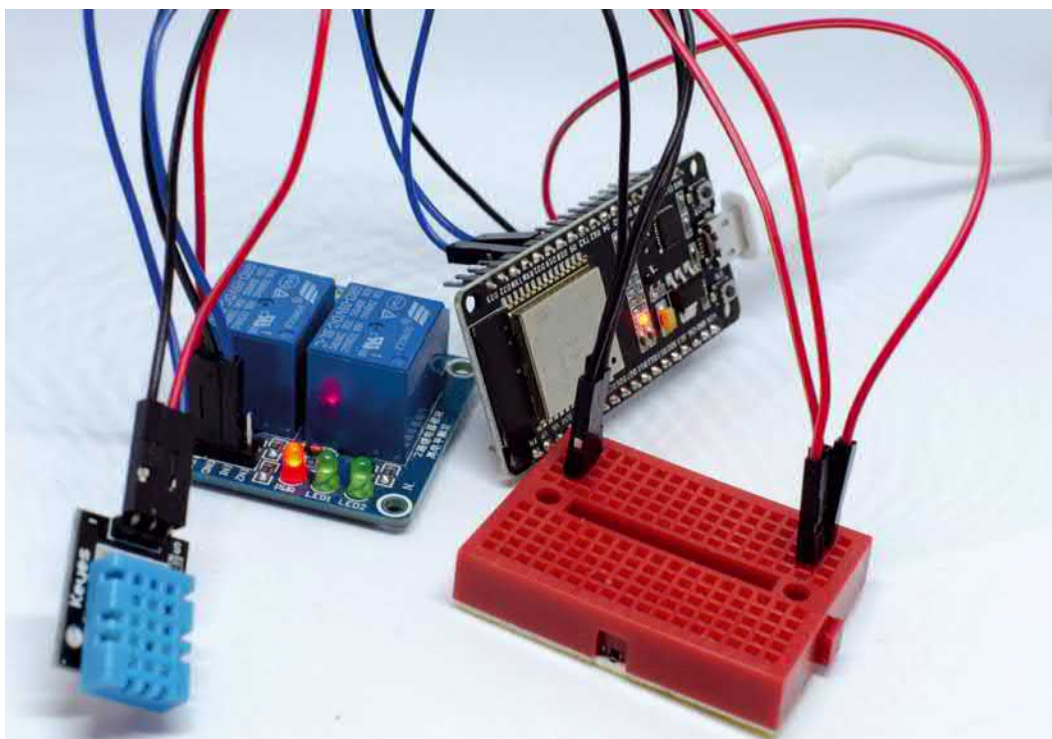


Fig. 1
Tipica
realizzazione
hardware
basata su
NodeMCU.

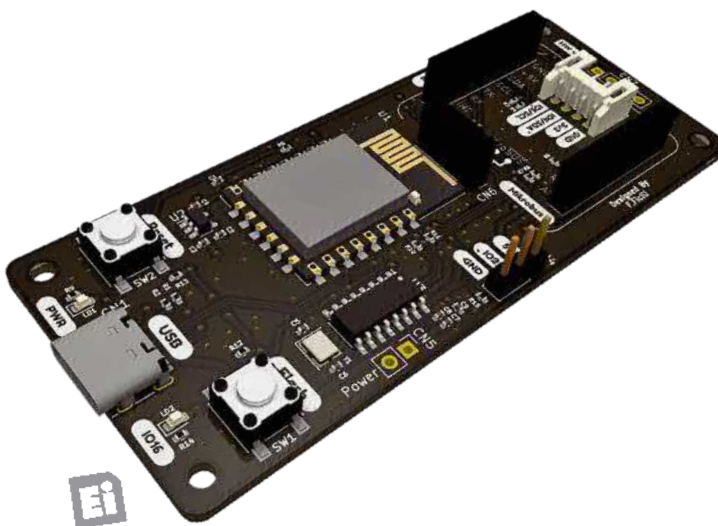
GANIMEDE.E12

Il risultato dei nostri ragionamenti è una famiglia di schede con hardware totalmente open, basata sul modulo ESP12, prodotto da Ai-Thinker (a sua volta basato sul SoC ESP8266, con l'aggiunta di memoria Flash QSPI, antenna WiFi integrata e tutta la circuiteria necessaria al funzionamento del SoC stesso); ciascuna scheda è dotata di interfacciamento hardware verso gli standard Mikroelettronika MikroBus (Click Boards) e SeedStudio Grove, in modo da poter utilizzare tutte le breakout board messe a disposizione da questi due standard. Ogni board è dotata inoltre, di porta USB per programmazione, debug ed interfacciamento verso Host PC, di connettore per interfacciamento con standard NeoPixel e 1-Wire, nonché per display OLED da 0,96" I²C-Bus. Abbiamo deciso di chiamare questa famiglia Ganimede, ed il modello specifico Ganimede.E12, per evidenziare il fatto che sia stata sviluppata partendo dal modulo ESP12. Ganimede è una piccola luna di Giove ed il nome è stato scelto per rappresentare il parallelismo tra il macrosistema costituito dagli strumenti di sviluppo per maker ed hobbisti (Giove) ed il piccolo ecosistema derivato che intendiamo costruire. Nella **Fig. 2** è riportata la scheda Ganimede.E12, dalla quale si vede la dotazione hardware.

SCHEMA ELETTRICO

Dopo aver descritto a grandi linee il progetto, passiamo alla descrizione hardware di dettaglio della scheda, per la quale ci riferiamo allo schema elettrico visibile nella pagina qui accanto. Come già anticipato, il cuore della nostra scheda è il modulo ESP12 (indicato nello schema elettrico come U1). E' stato scelto questo particolare modello in quanto risulta estremamente diffuso ed economico, oltre ad essere dotato di antenna WiFi integrata (e certificata), risolvendo di conseguenza la problematica dell'implementazione dell'antenna su PCB. Sebbene la maggior parte della circuiteria necessaria per il funzionamento dell'ESP8266 sia effettivamente racchiusa all'interno del modulo, ci

Fig. 2
La board Ganimede
E.12.



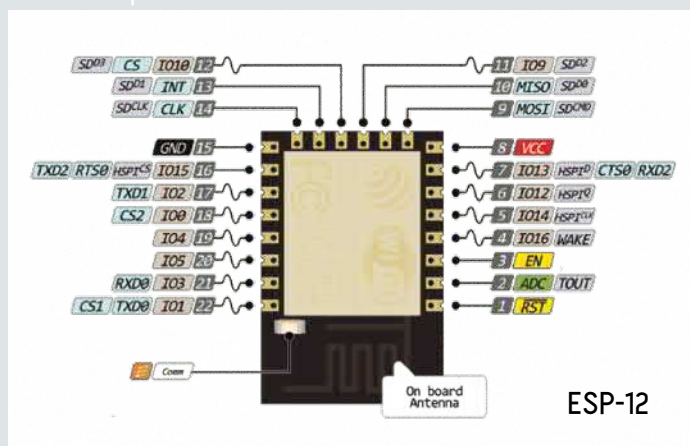
Il modulo ESP-12 è una scheda di sviluppo basata sul System-on-Chip (SoC) ESP8266 prodotto da Espressif Systems. La piccola board è stata progettata per fornire una soluzione compatta ed economica per l'implementazione di dispositivi connessi a Internet.



Il SoC ESP8266 è un microcontrollore a basso costo che integra un processore a 32 bit basato su processore Tensilica Xtensa LX106, un modulo Wi-Fi, una memoria flash e una serie di periferiche I/O. Queste caratteristiche lo rendono un'ottima soluzione per la costruzione di dispositivi connessi ad internet, come ad esempio sensori remoti, dispositivi di automazione domestica e prodotti IoT (Internet of Things). Il Soc ESP8266 è dotato di una interfaccia UART, un modulo ADC, un modulo SPI, un modulo I²C, una serie di GPIO e un modulo PWM. Queste interfacce permettono

di collegare il modulo ESP-12 a vari sensori e dispositivi esterni per creare progetti di vario tipo. L'ESP8266 supporta diversi protocolli di rete come TCP/IP, HTTP, DNS, DHCP, UDP e molti altri, il che lo rende una soluzione versatile per la connettività Wi-Fi. Dal punto di vista della programmazione, è possibile programmare l'ESP8266 utilizzando il linguaggio di programmazione C/C++, come anche linguaggi di scripting come LUA, uPython e JavaScript, rendendolo accessibile a una vasta gamma di sviluppatori. Inoltre, il SoC supporta vari ambienti di programmazione integrati, come l'IDE Arduino e PlatformIO, sistemi operativi real-time come FreeRTOS e vari framework, come i popolari framework di sviluppo NodeMCU e Tasmota. In sintesi, il modulo ESP-12 è una soluzione economica e compatta per creare dispositivi connessi ad Internet, che sfrutta le potenti funzionalità del SoC ESP8266. Grazie alla sua versatilità e alla vasta gamma di periferiche disponibili, il modulo ESP-12 è una scelta ideale per molti progetti di sviluppo di prodotti IoT. La pin-out del modulo ESP-12 è proposta nella Fig. A in questo riquadro.

Fig. A
Piedinatura
ESP-12.



sono alcuni accorgimenti importanti da implementare dal punto di vista hardware affinché il modulo funzioni correttamente. Per prima cosa ci sono alcuni pin speciali che devono essere opportunamente gestiti. Il primo è il pin EN (o chip-enable), che permette di mettere il chip in basso consumo dall'esterno, portando il pin stesso ad un valore logico basso. Non prevedendo di utilizzare questa funzionalità, abbiamo portato costantemente il pin ad un valore logico alto tramite la resistenza di pull-up da 10kohm R5. L'altro pin speciale è il pin RST, che permette di resettare il modulo dall'esterno. Noi abbiamo deciso di fornire la possibilità all'utente di resettare tramite un pulsante, quindi abbiamo collegato il pin RST al pulsante SW2, e mantenuto per impostazione predefinita lo stesso pin ad un valore logico alto tramite la resistenza R13, sempre da 10 kohm. In questo modo è possibile resettare la scheda premendo il pulsante SW2. I pin 9, 10, 11, 12, 13 e 14 sono da non collegare, perché sono utilizzati all'interno del modulo ESP12 per interfacciare la memoria programma QSPI interna da 4MB, nella quale viene salvato il codice da eseguire. Ci sono poi una serie di GPIO standard dotati di doppia funzionalità, potendo funzionare come normali GPIO in running, ma che vengono letti dal SoC allo startup per determinare la modalità di boot. Questi pin sono GPIO0, GPIO2 e GPIO15. In sostanza, allo startup, il SoC può eseguire il codice precedentemente flashato sulla Flash QSPI internal (Flash Boot) oppure, in alternativa, entrare in modalità bootloader UART (indicato come UART nella **Tabella 1**) e permettere l'aggiornamento della memoria Flash con un nuovo programma. La sezione di codice che implementa il bootloader UART è memorizzata in una ROM interna al SoC, e quindi non modificabile accidentalmente, rendendo il sistema molto robusto. Per entrare nelle due modalità, i pin GPIO 0, 2 e 15 devono avere ben determinati valori, come indicato nella **Tabella 1**, estratta dal datasheet del modulo ESP12.

Tabella 1
Stato dei GPIO 0, 2 e 15 per determinare i due boot mode del modulo ESP-12.

MODE	GPIO15	GPIO0	GPIO2
UART	low	low	high
Flash Boot	low	high	high

Come si può vedere, GPIO15 e GPIO2 devono stare sempre, rispettivamente, a GND e VCC, cosa che è stata ottenuta nel nostro schema tramite l'uso della resistenza di pull-down R6 e della resistenza di pull-up R15. Lo stato di GPIO0 determina effettivamente la modalità di startup, quindi abbiamo collegato questo pin al pulsante SW1 (Flash) e lo abbiamo portato a livello logico alto tramite R12. In questo modo la scheda parte sempre in modalità Flash Boot (ossia esegue il programma contenuto nella memoria Flash esterna), a meno che all'avvio non sia premuto SW1, nel qual caso viene avviato il bootloader UART, consentendo l'aggiornamento del programma (bootloading).

Infine un altro pin degno di attenzione è ADC_IN, che è l'unico pin analogico dell'ESP8266, ma ha una dinamica di tensione che va da 0 a 1V, mentre il SoC è alimentato a 3,3V. Di conseguenza abbiamo dovuto adattare la dinamica di tensione in ingresso a questo pin tramite il partitore di tensione costituito da R7 ed R8, rispettivamente da 220 kohm e 100kohm, ottenendo:

$$V_{adc} = \frac{R8}{R7+R8} \cdot V_{adc} \quad 0,31 \cdot V_{adc}$$

Un ultimo pin con funzione speciale è GPIO16, che è utilizzato come pin di wake-up. Noi abbiamo deciso di non implementare questa funzionalità, preferendo utilizzare GPIO16 per il controllo del LED LD2. Infine, tra Vcc e GND del modulo è stato collegato il condensatore di bypass C1, da 100nF.

Passiamo adesso alla descrizione della sezione USB: Ganimede.E12 dispone di un'interfaccia USB 2.0 che può essere utilizzata sia come porta di comunicazione seriale, che per la programmazione della scheda, in quanto è collegata alla porta UART che viene utilizzata come canale di comunicazione dal ROM bootloader. Questa porta è costituita da due elementi: il connettore USB CN1 ed il convertitore USB-Seriale U3 (CH340G).

Il connettore CN1 è un connettore USB-C, adattato allo standard USB2.0. Questo vuol dire che può essere utilizzato per il collegamento, un cavetto USB-C, ma non tutte le funzionalità dello standard USB 3.0 sono supportate (ad esempio non è possibile utilizzare la piena banda di comunicazione o negoziare con l'host livelli di corrente USB 3.0). Tuttavia sono mantenute le caratteristiche meccaniche, ossia la robustezza e la facilità di inserimento del connettore e la sua simmetria, ossia la possibilità di inserire il connettore in entrambi i versi. L'altro elemento dell'interfaccia è il convertitore

USB-seriale CH340G. Questo dispositivo è in grado di convertire una linea differenziale USB 2.0 in una porta UART standard, implementando di fatto una specifica classe di comunicazione USB, la classe CDC: Communication Device Class, pensata per soppiantare tramite lo standard USB, le vecchie porte seriali standard RS232.

Per funzionare il CH340G necessita solamente del circuito di oscillazione costituito dal cristallo Y1 da 12MHz e dai due condensatori C5 e C6 da 12pF, oltre che dai condensatori C7 e C8, rispettivamente da 10uF e 100nF e C9 da 100nF.

Il dispositivo è collegato tramite linea differenziale ai pin D+/D- del connettore USB e tramite una linea standard UART (TXD ed RXD, pin 2 e 3) ai pin GPIO1 e GPIO3 del modulo ESP-12.

In questo modo il modulo vede una porta standard UART, mentre un qualsiasi dispositivo USB Host (ad esempio un Personal Computer) vede la scheda come un dispositivo USB 2.0.

Passiamo adesso alla descrizione della sezione di alimentazione. La porta USB appena descritta è anche la principale sorgente di alimentazione della scheda, ma i 5V forniti dal dispositivo USB host (o da un alimentatore USB standard), necessitano di essere adattati a 3,3V, in quanto quella è la tensione di lavoro del modulo ESP12 (il CH340G funziona invece a 5V). Per adattare questa tensione è stato utilizzato il regolatore lineare SPX3819M5-L-3-3 (U2), in grado di erogare fino a 500mA, più che sufficienti per il funzionamento della Ganimede. Per il suo funzionamento il regolatore necessita della circuiteria di condensatori costituita da C2 (1uF), C3 (470pF) e C4 (10uF).

Il LED LD1, con la sua resistenza di limitazione R9, è stato posto a valle del regolatore per indicare la presenza della corretta tensione di funzionamento. La parte restante della scheda è costituita da una serie di connettori che permettono di interfacciare standard di espansione hardware.

In particolare la scheda è dotata di:

- un socket Mikroelettronika mikroBus (CN2), che permette di interfacciare le popolari Click board prodotte da Mikroelettronika (<https://www.mikroe.com/mikrobis>), per il quale sono disponibili più di 1000 schede attualmente in commercio;
- un socket Seed Studio Grove (CN3), che permette di interfacciare i dispositivi dello standard Grove (<https://www.seeedstudio.com/category/Grove-c-1003.html>), per il quale esistono circa 200 breakout board in commercio;
- un header a 3 poli (CN4) con il quale è possibile



Pinout specification

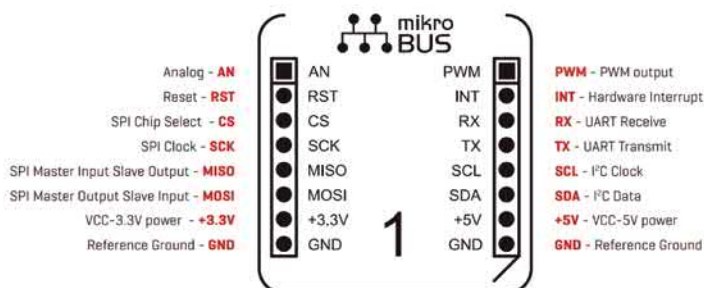


Fig. 3
Piedinatura
del
connettore
standard
Mikrobus.

interfacciare dispositivi NeoPixel, quindi LED Strip o Moduli LED Neopixel di vario tipo, ma anche dispositivi 1-wire, quindi sensori di temperatura tipo DS18B20 o dispositivi simili;

- due header a 4 poli (CN6 e CN7) pensato per l'interfacciamento di OLED da 0,96" tramite protocollo I²C o dispositivi similari.

GLI STANDARD MIKROBUS E GROVE

I due connettori di espansione della Ganimede.E12 che offrono maggiori possibilità sono CN2 e CN3, che permettono di interfacciare rispettivamente gli standard Mikrobus e Grove. Analizziamo più in dettaglio le caratteristiche di questi due standard. MikroBus è uno standard di connettività hardware per microcontrollori e dispositivi embedded, sviluppato dall'azienda Serba Mikroelektronika. È stato progettato per fornire un modo semplice e conveniente per connettere sensori, attuatori e altri componenti a microcontrollori di vario tipo. Lo standard è basato su un connettore (o socket), con pin-out standardizzata, riportato in **Fig. 3**. Il prodotto più noto che fa uso dello standard sono

le schede della linea Click, prodotte sempre da Mikroelektronika, delle quali esistono più di 1.000 diverse varianti in commercio, ognuna con una specifica funzione e con un proprio hardware. La specifica dello standard è open source e disponibile al seguente indirizzo web:

<https://download.mikroe.com/documents/standards/mikrobus/mikrobus-standard-specification-v200.pdf>

La specifica permette che anche altri produttori di schede possano adoperare la specifica nelle loro realizzazioni hardware o anche contribuire nella realizzazione di nuovi dispositivi.

L'utilizzo di uno standard di connettività come Mikrobus semplifica notevolmente lo sviluppo di progetti embedded. Gli sviluppatori possono concentrarsi sulle loro attività di codifica, poiché il connettore Mikrobus fornisce una soluzione semplice e conveniente per la connessione di componenti. Inoltre, questo standard è molto comodo per la prototipazione rapida, poiché gli sviluppatori possono facilmente sostituire o aggiungere componenti al loro progetto senza dover preoccuparsi della connessione hardware.

Fig. 4
Ganimede.E12
con Click Relay
montata sul suo
socket Mikrobus.

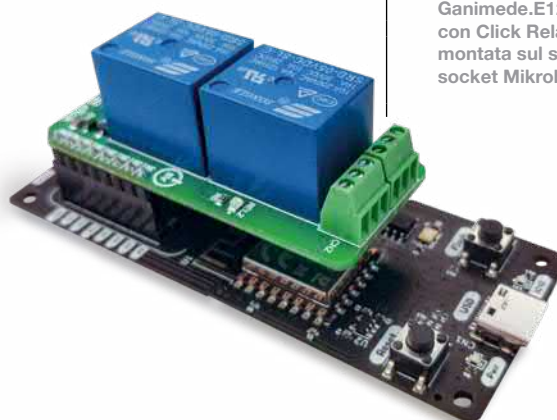


Fig. 5
Sistema Grove.



In **fig. 4** è rappresentata la scheda ganimede.e12 connessa ad una click board Relay. Grove è un altro standard molto diffuso nel panorama hobbystico/maker creato da SeedStudio. Si tratta di un altro standard di espansione hardware che però, a differenza di MikroBus, che utilizza un socket con 16 poli e breakout dotate di connettori equivalenti (è quindi possibile pluggare le schede su socket), usa un connettore standard a 4 poli e breakout connesse via cavo. I 4 pin del connettore standard grove portano potenza e segnali e questi ultimi hanno delle connessioni che variano a seconda della tipologia di scheda connessa:

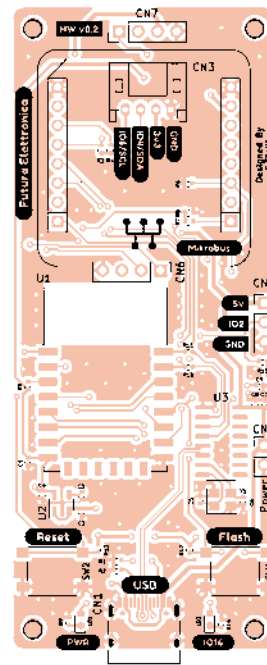
- pin 1 – Giallo – Segnale 1; può essere l'SCL del

bus I²C, Digital I/O, Analog, UART RX a seconda della scheda connessa;

- pin 2 – Bianco – Segnale 2; può essere SDA dell'I²C, Digital I/O, Analog, UART TX a seconda della scheda connessa;
- pin 3 - Rosso – VC; comune a tutti i connettori;
- pin 4 - Nero – GND; ripetuto su tutti i connettori.

Anche in questo caso abbiamo una vasta gamma di dispositivi, circa 200-300 unità (**Fig. 5**) che possono essere connessi alla nostra scheda Ganimede. Maggiori info possono essere reperite al seguente indirizzo web: https://wiki.seeedstudio.com/Grove_System/.

piano di MONTAGGIO



Elenco Componenti:

R9, R14: 1 kohm (0402)
 R10, R11: 470 ohm (0402)
 R8: 100 kohm (0402)
 R3, R4, R5, R6, R12, R13, R15: 10 kohm (0402)
 R7: 220 kohm (0402)
 R1, R2: 5,1 kohm (0402)
 C1, C8, C9: 100 nF ceramico (0402)
 C3: 470 pF ceramico (0402)
 C5, C6: 12 pF ceramico (0402)

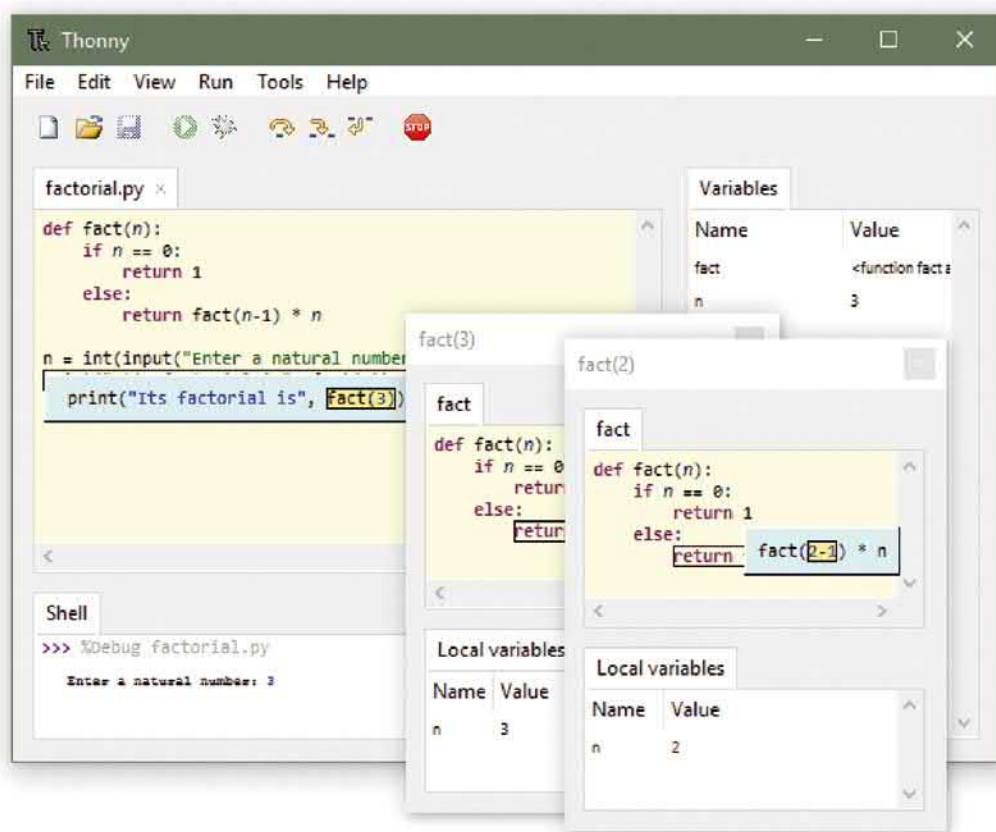
C2: 1 µF ceramico (0402)
 C4, C7: 10 µF ceramico (0402)
 LD1: LED verde (0603)
 LD2: LED rosso (0603)
 U1: ESP-12F
 U2: SPX3819M5-L-3-3
 U3: CH340G
 SW1, SW2: Microswitch
 Y1: Quarzo 12 MHz (3.2x2.5mm)

CN1: Connettore Usb Type-C

Varie:

- Strip Femmina 4 vie (1 pz.)
- Strip Femmina 8 vie (2 pz.)
- Strip Maschio 3 vie (1 pz.)
- Connettore JST 4 vie passo 2 mm da CS
- Circuito stampato S1673 (82x34 mm)

Fig. 6
L'editor Python
Thonny.

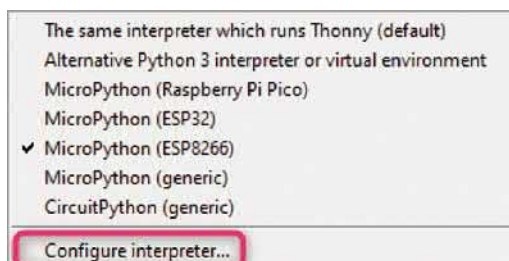


INSTALLAZIONE DELL'INTERPRETE MICROPYTHON

Come abbiamo già anticipato, la scheda Ganimede E12 è programmabile con una varietà di linguaggi di programmazione, tutti molto validi. Sicuramente una scelta ideale per chi comincia ad utilizzarla è MicroPython, che è un ambiente semplice da apprendere, ma sorprendentemente potente. Una descrizione dettagliata degli step necessari all'installazione dell'ambiente MicroPython, con anche una serie di esempi, può essere trovata nella sezione Python del repository GitHub della scheda: https://github.com/ffich/Ganimede/tree/main/10_Python.

Per vostra comodità, riportiamo qui di seguito l'intera procedura.

Fig. 7
Selezione delle
opzioni di configu-
razione dell'inter-
prete micropython.



Step 1: per prima cosa occorre scaricare il firmware Micropython dal seguente link: <https://micropython.org/download/esp8266/>.

Step 2: ora che abbiamo il firmware, occorre flasharlo sulla scheda. Per poterlo fare abbiamo bisogno di un tool fornito alla espressif, che si chiama ESptool, che può essere scaricato dal seguente link: <https://github.com/espressif/esptool>. Il tool è a sua volta basato su Python, quindi occorre averlo installato sulla macchina che verrà effettivamente utilizzata per flashare la scheda. Istruzioni utili all'installazione del tool stesso ed alla soluzione di eventuali problemi possono essere reperite al seguente link: <https://docs.espressif.com/projects/esptool/en/latest/esp32/>.

Step 3: a questo punto possiamo procedere alla programmazione, e per prima cosa va attivato il bootloader. Come abbiamo visto possiamo farlo tenendo premuto il tasto SW2 all'accensione o al riavvio della scheda. Una volta attivato il bootloader possiamo flashare, aprendo una finestra di comando nella directory nella quale abbiamo scaricato ed estratto ESptool, scrivendo le seguenti righe:

```
esptool.py --port [VCP] erase_flash
```

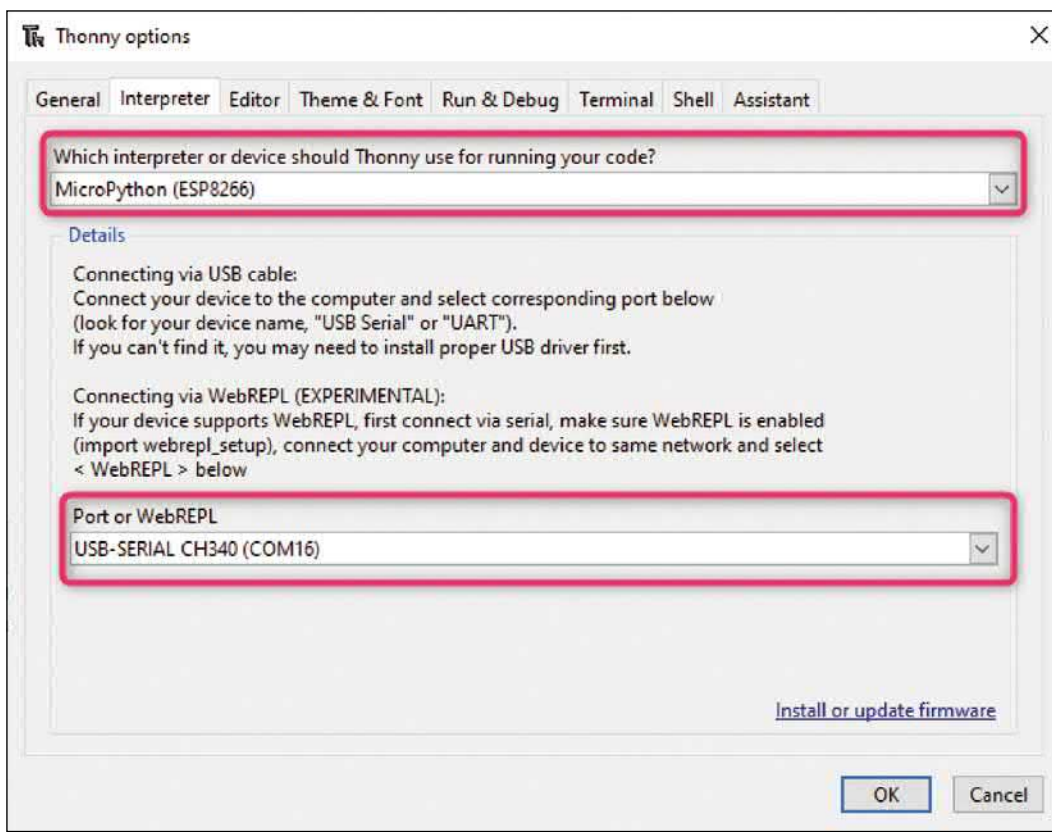



Fig. 8
Selezione
delle opzioni
dell'inter-
prete.

per la cancellazione della memoria e:

```
esptool.py --port [VCP] --baud 460800 write_flash
--flash_size=detect 0 esp8266.bin
```

per il flashing dell'interprete MicroPython.

In entrambe le righe [VCP] sta ad indicare il numero della porta COM virtuale associata alla scheda (ad esempio COM3 o COM5).

Invece **esp8266.bin** rappresenta il file binario dell'interprete Micropython, che deve essere copiato nella stessa directory (cartella) dalla quale eseguirete la finestra di comando.

A questo punto l'interprete è presente sulla scheda e possiamo utilizzare un editor Python per programmarla.

Come editor suggeriamo Thonny (<https://thonny.org/>), perché è molto semplice da utilizzare, del quale la **Fig. 6** propone una schermata. Una volta installato Thonny, potete aprirlo e configurare l'interprete per lavorare con Ganimede. Per farlo, cliccare con il tasto sinistro del mouse nell'angolo in basso a destra e selezionate l'opzione "Configure Interpreter", come illustrato in **Fig. 7**. A questo punto navigate nel tab "Interpreter" sele-

zionate (**Fig. 8**):

- nel primo combobox, MicroPython(Esp8266);
- nel secondo combobox, USB-SERIAL CH340 (VCP).

Dove VCP è la porta COM virtuale associata alla vostra scheda Ganimede dal vostro Personal Computer. Da questa stessa finestra, tramite l'opzione "Install or update Firmware", è possibile installare una nuova versione di MicroPython, senza dover chiudere l'editor Thonny o dover ricorrere agli script batch presentati nella sezione precedente.

Selezionando l'opzione "Install or update Firmware", si aprirà una finestra di dialogo come quella proposta in **Fig. 9**, dalla quale potrete selezionare il binario MicroPython da caricare sulla scheda e la porta COM virtuale.

Se tutto è andato a buon fine, una volta resettata o scollegata e poi ricollegata la scheda Ganimede.E12, nella Shell di Thonny dovrete vedere un prompt simile a quello di **Fig. 10** (la versione di MicroPython potrebbe essere differente).

Il prompt che vedrete è i REPL, acronimo di Read Evaluate Print Loop, ed è il nome che è stato dato alla shell di MicroPython con la quale è possibile interagire.



MicroPython è un linguaggio di programmazione open source che mira a fornire una versione semplificata e più piccola di Python per microcontrollori e altri dispositivi embedded. MicroPython è stato progettato per essere efficiente in termini di memoria e di velocità sui dispositivi embedded, il che lo rende un'ottima scelta per progetti IoT (Internet delle cose). Una delle caratteristiche più interessanti di MicroPython è la sua compatibilità con il linguaggio Python, che lo rende accessibile a molte persone che hanno già familiarità con Python. Questo significa che gli sviluppatori possono utilizzare la maggior parte delle conoscenze acquisite con Python anche su MicroPython. MicroPython ha molte applicazioni nell'ambito dell'IoT. Può essere utilizzato per creare dispositivi embedded come sensori ambientali, sistemi di automazione domestica, dispositivi di sicurezza e molto altro ancora. Inoltre, offre una vasta gamma di librerie integrate

per supportare molte funzionalità di base, come la connessione Wi-Fi, l'elaborazione di protocolli di rete, la gestione della memoria, la creazione di interfacce grafiche e molto altro. Il port per ESP8266 di MicroPython offre un modo semplice e conveniente per programmare il modulo ESP-12 basato su ESP8266 ed è stato progettato per fornire una soluzione completa e integrata per la programmazione di progetti IoT con MicroPython. Inoltre, il port per ESP8266 offre un ambiente di sviluppo integrato che include un'interfaccia REPL (Read-Eval-Print Loop), un editor di codice integrato e molte altre funzionalità che semplificano la vita degli sviluppatori.



HELLO WORLD IN MICROPYTHON

Ora che abbiamo MicroPython installato sulla nostra Ganimede.E12, possiamo implementare il nostro primo, semplice script. Come nel più classico dei casi, vogliamo far lampeggiare il LED utente alla frequenza di 1Hz. Apriamo Thonny e scriviamo lo script riportato qui di seguito:

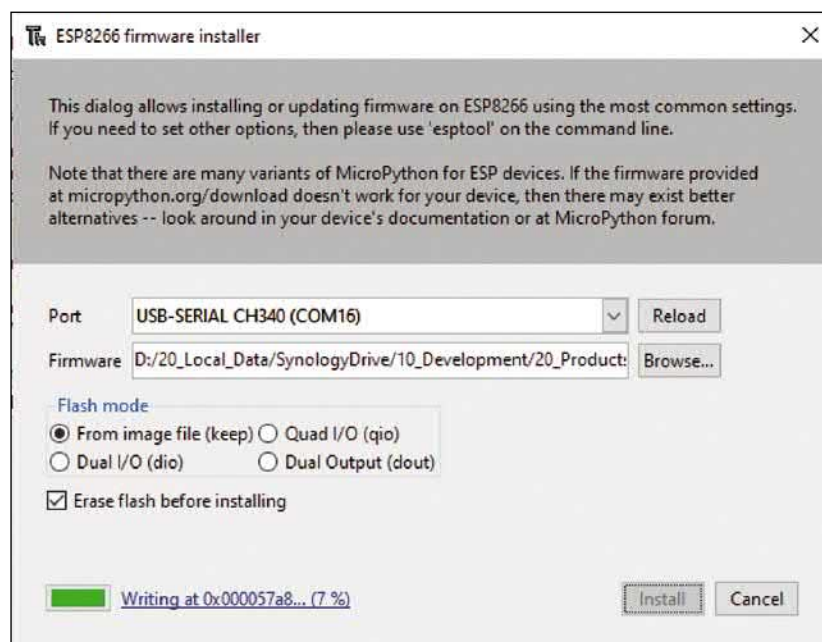
```
from machine import Pin
import time
led = Pin(16, Pin.OUT)

while True:
    led.value(not led.value())
    if (led.value()):
        print("Il LED è acceso!!!")
    else:
        print("Il LED è spento!!!")
    time.sleep(1)
```

Lo script inizia importando le librerie "machine" e "time" che forniscono rispettivamente l'accesso alle funzionalità del microcontrollore e alle funzioni di timing. Quindi, si crea un oggetto "led" utilizzando la classe Pin. L'oggetto "led" viene inizializzato con il numero di pin 16 (collegato allo user LED, come abbiamo visto in precedenza) e il modo "Pin.OUT", il che significa che il pin viene configurato come uscita. La sezione di codice all'interno del ciclo "while" viene eseguita in continuazione. Il codice all'inter-

Fig. 9

Aggiornamento o installazione del FW MicroPython direttamente da Thonny.



```

Shell x
MicroPython v1.19.1 on 2022-06-18; ESP module with ESP8266
Type "help()" for more information.
>>>

```

← Fig. 10
REPL di
MicroPython.

```

Thonny - D:\20_Local_Data\SynologyDrive\10_Development\40_Publications\10_Magazines\10_Articles\FFAR2301_GanimedeE12\20_Output\10_Article\Samples\blinky.py @ 13:1
File Edit View Run Tools Help

blinky.py
1 from machine import Pin
2 import time
3
4 led = Pin(16, Pin.OUT)
5
6 while True:
7     led.value(not led.value())
8     if (led.value()):
9         print("Il LED è acceso!!!")
10    else:
11        print("Il LED è spento!!!")
12    time.sleep(1)

Shell x
MicroPython v1.19.1 on 2022-06-18; ESP module with ESP8266
Type "help()" for more information.
>>> %run -c $EDITOR_CONTENT

Il LED è spento!!!
Il LED è acceso!!!
Il LED è spento!!!
Il LED è acceso!!!
Il LED è spento!!!
Il LED è acceso!!!

```

← Fig. 11
Script di
esempio in
funzione.

no del ciclo cambia il valore del pin utilizzando la funzione "led.value(not led.value())". Questo codice inverte il valore corrente del pin 16 ad ogni ciclo del ciclo "while", spegnendo o accendendo il LED ad ogni iterazione. L'istruzione condizionale *if...else* serve a stampare sulla Shell lo stato del LED, tramite l'istruzione *print*. Infine, la funzione "time.sleep(1)" fa sì che il programma attenda un secondo prima di passare alla prossima iterazione del ciclo "while". Questo fa in modo che il LED lampeggi alla frequenza di 1Hz. In Fig. 11 è riportato l'output della shell con lo script in funzione.

CONCLUSIONI

Come avete potuto vedere in queste pagine, la Ganimede.E12 è una scheda estremamente versatile e semplice da utilizzare, specialmente utilizzando linguaggi di programmazione evoluti come Micropython. Allo stesso tempo la scheda è molto

economica ed è facilmente espandibile tramite gli standard MikroBus e Grove e gli altri connettori di cui dispone, e permette di realizzare applicazioni complete senza fare ricorso all'utilizzo di breadboard o cavetti volanti. Stiamo già lavorando a una serie di progetti pratici sviluppati con Ganimede.E12, che contiamo di proporvi a breve. □



Cosa occorre?

La board Ganimede (cod. GANIMEDE-E12) è disponibile presso Futura Elettronica a Euro 13,00, il modulo ESP12 (cod. ESP12F) è in vendita anche separatamente a Euro 7,50. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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







di DANIEL SCAINI

FUEL PRICE MONITOR

Controllo attraverso ESP32 di un pannello touch interattivo che espone i prezzi dei distributori di carburante in una determinata zona.

I continui rincari delle materie prime che colpiscono la nostra vita quotidiana cominciano ad essere un problema non trascurabile. Alcuni di questi sono inevitabili, mentre per altri riusciamo ad avere un minimo di scelta e a far fronte trovando alternative. Quello che da alcuni mesi ci preoccupa molto è il rincaro dei carburanti, tanto che è cresciuta la sensibilità circa la ricerca del distributore che esponga il prezzo migliore. Ma quante volte subito dopo aver appena fatto il pieno, magari in una zona che frequentiamo d'abitudine, notiamo lungo la strada un distributore di gran lunga più conveniente? La soluzione potrebbe essere un dispositivo che ci aiuti in queste decisioni e ci consigli il migliore. In questo articolo proveremo a sviluppare un assistente a cui, passati in ingresso una città oppure le sue coordinate, ci proponga a video tutti i distributori e ci comunichi in tempo reale quale o quali siano i migliori dove poter fare rifornimento. Per far ciò utilizzeremo, lato software, un'app di messaggistica già nota a tutti: Telegram. La quale potrà inviare messaggi di testo oppure condividere la propria posizione. Lato hardware invece utilizzeremo una delle schede più conosciute ed utilizzate, ovvero la ESP32. Quest'ultima sarà all'interno di un circuito più complesso che prevede al suo fianco un utilissimo display touch LCD, in modo da poter mostrare a video i risultati e da poter interagire con essi attraverso un'interfaccia semplicissima, facilmen-



 **Fig. 1**
Interfaccia
utente.

te consultabile anche da chi è poco avvezzo alla tecnica (**Fig. 1**).

I SERVIZI

Come accennato durante l'introduzione l'algoritmo si basa su latitudine e longitudine e sfrutta due servizi per poter estrarre i dati di cui abbiamo bisogno.

Il primo servizio si occupa di risalire, dall'inserimento di un campo testuale, alle coordinate vere e proprie. Su Internet oggi giorno vengono offerti numerosi servizi che si occupano di geo tracking o di reverse geo tracking.

Fra i più famosi vi sono senza dubbio quelli offerti da società come Google (<https://developers.google.com/maps/documentation/geocoding/overview>), Here, uno dei leader nel campo dei navigatori (<https://developer.here.com/tutorials/learn-the-reverse-geocoding-using-postman/>) oppure OpenWeather (<https://openweathermap.org/api/geocoding-api>). Nel nostro caso sfrutteremo un quarto fornitore free. La nostra scelta ricade infatti su Geocode.xyz. Questo ci offre, previa iscrizione, la possibilità di fare diverse richieste gratis a patto che non superiamo la frequenza di 1 per secondo (<https://geocode.xyz/pricing>).

Una volta registrati sul sito, attraverso l'inserimento di una username valida ed una password e ricevuta la e-mail di conferma, viene rilasciata API_KEY personale per poter invocare le richieste. L'engagement con l'API avviene attraverso una chiamata GET al seguente URL: https://geocode.xyz/Milano?json=1&auth=API_KEY.

Una volta completata la richiesta, abbiamo accesso al seguente codice JSON che riporta, oltre a latitudine e longitudine, anche la precisione del dato di ritorno:

```
{
  "standard": {
    "address": {},
    "statename": {},
    "city": "Milano",
    "prov": "IT",
    "countryname": "Italy",
    "postal": {},
    "confidence": "0.90"
  },
  "longt": "9.18023",
  "alt": {
    "loc": [
      {
        "longt": "9.16903501162404",
        "prov": "LOMBARDIA",
        "city": "MILANO",
        "postal": "20161",
        "score": "3011",
        "latt": "45.5186721687148"
      },
      {
        "longt": "9.20334553448277",
        "prov": "IT",
        "city": "Milano",
        "postal": "20124",
        "score": "580",
        "latt": "45.4855161034483"
      },
      {
        "longt": "9.16904",
        "prov": "IT",
        "city": "Milano",
        "countryname": "Italy",
        "postal": "20161",
        "region": {},
        "latt": "45.51867"
      }
    ]
  },
  "elevation": {},
  "remaining_credits": "-7",
```

↓ Listato 1

```
{
  "success": true,
  "center": {
    "lat": 44.90938025959227,
    "lng": 10.66000690845997
  },
  "results": [
    {
      "id": "Tamoil Gualtieri",
      "name": "8976",
      "fuels": [
        {
          "id": 47742999,
          "price": 1.819,
          "name": "Benzina",
          "fuelId": 1,
          "isSelf": true
        },
        {
          "id": 47742998,
          "price": 1.859,
          "name": "Gasolio",
          "fuelId": 2,
          "isSelf": true
        }
      ],
      "location": {
        "lat": 44.889557326884855,
        "lng": 10.617287009954453
      },
      "insertDate": "2023-01-17T11:16:11+01:00",
      "address": "VARIANTE S.S. 62 9 42044 - GUALTIERI RE",
      "brand": "Tamoil"
    },.....
  ]
}
```

```
"latt": "45.47547"
}
```

Il secondo servizio, una volta impostati la posizione ed un raggio intorno ad essa, ci restituisce tutti i distributori della zona interessata. Per ottenere questa lista sfruttiamo un API pubblica presente anche all'interno del sito del Ministero delle Imprese e del Made in Italy (<https://carburanti.mise.gov.it/ospzSearch/zona>). All'interno del portale è possibile, attraverso un'interfaccia grafica basilare, scegliere il proprio comune di residenza sulla mappa, impostare un raggio di azione e ottenere così la lista completa di tutti i distributori di benzina nelle vicinanze. A proposito del raggio, va precisato che il portale del MISE consente di impostare un'area di ricerca limitata, quindi se intendete modificare il raggio ricordate che il raggio impostato nel nostro firmware è 5 km e se andate oltre è possibile che i risultati restituiti contengano un errore; nessun problema,

invece, se impostate un raggio inferiore.

La riga di codice interessata è riportata nel paragrafo qui sotto e il parametro passato è "radius": 5. L'API alla base di questa piattaforma, a differenza della precedente, è una chiamata POST, accessibile anche dall'esterno all'indirizzo: <https://carburanti.mise.gov.it/ospzApi/search/zone>. Trattandosi di una funzione POST, vi è un dato in entrata e un codice JSON che conterrà tutto il necessario; questa è la stringa corrispondente:

```
{"points": [{"lat": "44.917653509784884", "lng": "10.65221077251436"}], "radius": 5}
```

In risposta ad essa avremo accesso ad un oggetto con la struttura proposta dal **Listato 1**.

L'HARDWARE DEL PROGETTO

L'hardware su cui ci basiamo, che potete trovare nello store on-line di Futura Elettronica (www.futurashop.it), è un Esp32-S3 con display





Fig. 2
Il display con
Esp32 utilizzato.

TFT da 3,5". Il core di questa scheda è un processore operante a 2,4 GHz di frequenza di clock, con interfacce I²C-Bus, I²S, SPI, PWM, UART, USB; dispone inoltre di due porte USB di tipo C, uno slot per SD-Card ed è dotato connettività WiFi e Bluetooth. Nella **Fig. 2** vedete la nostra board da entrambe le facce.

Un altro aspetto non trascurabile è sicuramente la memoria, che può essere una Flash, RAM oppure EEPROM.

Sfrutteremo quest'ultima, Electronically Erasable Programmable Read Only Memory, per poter immagazzinare informazioni capaci di sopravvivere al riavvio della macchina stessa.

Il display ha un touchscreen capacitivo e sfrutta il controller grafico ILI9488 e interfacciamento SPI. Quest'ultima caratteristica lo rende decisamente migliore di altri display dotati di interfaccia di comunicazione parallela.

Arduino IDE ci mette a disposizione una libreria molto importante per poter interagire con questo

schermo; pertanto, una volta aperto il nostro IDE entriamo nella finestra di dialogo del Gestore Librerie (comando di menù Strumenti>Gestione Librerie) e da essa andiamo ad installare la versione 1.1.2 di LovyanGFX.

L'uso di questa libreria è veramente elementare: si crea un oggetto LGFX, si inizializza, si settano alcuni parametri, che vedremo dopo, quindi questo diventa la "tela" su cui possiamo disegnare o scrivere. Un esempio può essere il codice contenuto in HowToUse→simple_use.ino che si trova nella libreria LovyanGFX. Ci sono anche altre entità, come per esempio le Sprite, ma servono in caso di utilizzi più avanzati.

IL BOT DI TELEGRAM

Telegram è un'applicazione di messaggistica istantanea, tra le più utilizzate in Italia insieme con Facebook Messenger, WhatsApp, Signal e Instagram Direct.

Nasce dall'idea di Pavel Durov, giovane imprenditore russo ed ex studente dell'Università statale di San Pietroburgo, insieme a suo fratello Nikolai. Tra le altre cose, i fratelli Durov hanno fondato anche il social network russo VK (VKontakte, che in italiano significa letteralmente "in contatto"). L'anno di nascita di Telegram risale al 2013, quattro anni più tardi rispetto al suo diretto concorrente WhatsApp. Pavel Durov sapeva che per ottenere successo avrebbe dovuto proporre qualcosa di diverso, come effettivamente ha fatto.

Nel corso degli ultimi anni, numerosi utenti si sono avvicinati a Telegram perché delusi dai "down" che hanno colpito l'app concorrente.

Un altro fattore importante da considerare nella vicenda è la sicurezza di WhatsApp, più volte finito sotto attacco dai suoi detrattori, i quali spingono molto sulle polemiche legate alla composizione dell'impero Zuckerberg (WhatsApp, Instagram, Facebook).

Una delle peculiarità più interessanti di questo servizio di messaggistica è insita nei gruppi: questi possono raggiungere dimensioni enormi, anche oltre i 100.000 membri.

Inoltre ci dà la possibilità di creare dei veri e propri canali di trasmissione che possono avere un numero illimitato di utenti e possono essere sia privati che pubblici.

Tra quelli privati ci sono i bot ed è proprio quest'ultima caratteristica che possiamo sfruttare per creare un filo diretto fra noi e la nostra scheda con ESP32.

Un bot è un "utente" di Telegram non umano, fatto

da righe di codice e capace di svolgere azioni più o meno complesse in modo automatico, ricevendo i comandi dell'utente sotto forma di richieste HTTP. Le tipologie di bot Telegram sono due:

- Bot stand alone, dotato di una finestra di chat autonoma in cui l'utente può impartire i comandi; per raggiungere un bot stand alone basta aggiungerlo attraverso un link;
- Bot inline, che può essere richiamato da altre finestre di chat (singole o di gruppi) tramite il comando @nomebot.

Ciascun bot Telegram è identificato univocamente da un token: una sequenza di numeri e lettere usata dalle applicazioni web e dalle API per riferirsi a quello stesso Bot.

Apriamo l'app e clicchiamo sulla lente di ricerca nella barra superiore. Qui digitiamo BotFather. (Fig. 3). Clicchiamo su questa e apriamo la chat nella quale andremo a scrivere questo messaggio: /newbot, e seguiamo le istruzioni a video.

Dopo questa procedura copiamo il token rilasciato da BotFather.

Questa è una delle due variabili, SECRET_BOT_TOKEN, che ci servirà per comunicare con le API di Telegram.

La seconda invece la possiamo ottenere in diversi modi. Quello che consigliamo è mandare un messaggio con scritto "Ciao" all'interno della chat appena creata e inserire all'interno di un browser questo link: <https://api.telegram.org/bot<BOTToken>/getUpdates>. Dove <BOTToken> è il token del passo precedente.

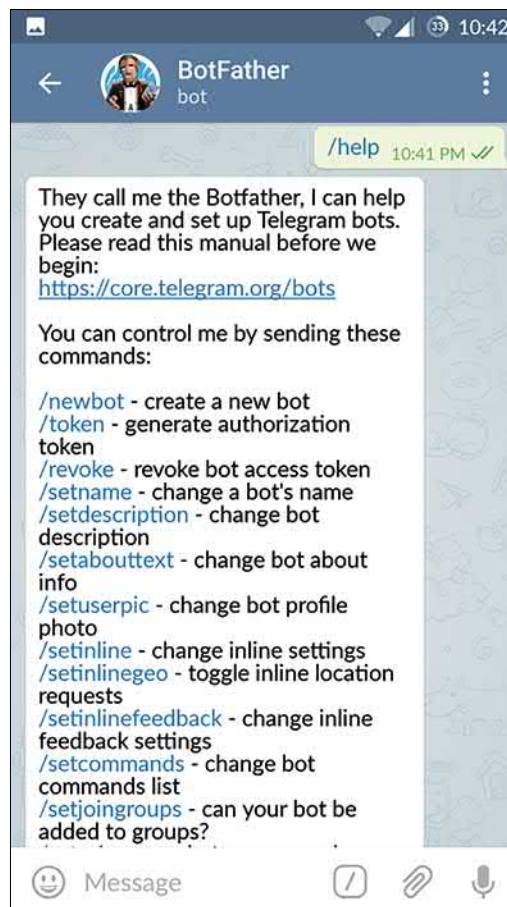
Gli step dovranno esser fatti esattamente in questo ordine.

In risposta, sul browser apparirà un JSON, ovvero una struttura dati formattata in un determinato modo, simile al seguente:

```
{
  "ok":true,
  "result":[
    {
      "update_id":YYYYYY,
      "message":{
        "message_id":18,
        "from":{
          "id":XXXXXXX,
          "is_bot":false,
          "first_name":"NOME COMPLETO TELEGRAM",
          "username":"IL TUO USERNAME TELEGRAM",
          "language_code":"it"
        }
      }
    }
  ]
}
```

```
},
"chat":{
  "id": XXXXXXXX,
  "first_name":"NOME COMPLETO TELEGRAM",
  "username":"IL TUO USERNAME TELEGRAM",
  "type":"private"
},
"date":1600324371,
"text":"Ciao"
}
}
}
```

Questa struttura rappresenta le interazioni con il Bot da parte degli utenti che vi si collegano. Ovviamente noi abbiamo interagito con lui precedentemente con il messaggio "Ciao". L'ultima variabile, SECRET_CHAT_ID, si trova proprio all'interno di questo json nel campo in grassetto.



← Fig. 3
Interfaccia di BotFather



Fig. 4
La schermata
dell'IDE di Arduino.



le librerie necessarie:

```
#include <ArduinoJson.h>
#define LGFX_AUTODETECT
#define LGFX_USE_V1
#include <LovyanGFX.hpp>
#include <LGFX_AUTODETECT.hpp>
#include <WiFi.h>
#include <EEPROM.h>
#include <StreamUtils.h>
#include "arduino_secrets.h"
#include "display_touch.h"
#include "times.h" //FACOLTATIVA
#include "telegram_bot.h"
#include "coordinates.h"
#include "fuelprice.h"
```

IL SOFTWARE

L'hardware scelto per il progetto si programma con gli stessi tool di Arduino e quindi con il relativo IDE; per poter programmare la nostra scheda, dobbiamo verificare che l'IDE abbia già i relativi driver di terze parti, altrimenti andiamo ad impostare il nostro ambiente IDE per poter riuscire ad uploadare correttamente il nostro firmware. Ci rechiamo quindi in File->Impostazioni e qui inseriamo il seguente URL: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json e facciamo clic su OK.

Andando in Strumenti->Scheda->Gestore schede cerchiamo esp32 e installiamo o aggiorniamo con la versione 2.0.6.

Uscendo ora selezioniamo la nostra scheda di riferimento ovvero ESP32S3 Dev Module.

Ora possiamo cominciare a dedicarci al vero codice, ossia quello di Arduino; per prima cosa importiamo

Le prime otto importate riguardano classi standard come quella per trattare i JSON (Javascript Object Notation) per poter interagire con l'esterno oppure quella già citata per poter lavorare con il display integrato nella scheda; invece le ultime saranno le nostre classi personalizzate che servono per poter mantenere il codice ordinato, facilitandone la lettura, nonché migliorare la manutenibilità e la modularità.

Ora salviamo il file, in modo tale da poter creare già la cartella del progetto.

Creiamo la prima classe personalizzata: arduino_secrets.h e per semplicità lo collocheremo sul Desktop. A questo punto importiamo il all'interno del progetto Arduino: nella barra degli strumenti dell'IDE clicchiamo su Sketch e poi su Importa file. Selezioniamo un file alla volta.

Lo stesso procedimento lo faremo anche per le altre librerie personalizzate. Per queste, oltre a cre-

Listato 2

```
#define WIFI_SECRET_SSID "WiFi SSID"
#define WIFI_SECRET_PASSWORD "WiFi Password"

#define TIME_ZONE "http://worldtimeapi.org/api/timezone/Europe/Rome"

#define COORDINATES_PATH "https://geocode.xyz/"
#define COORDINATES_KEY "?json=1&auth=AUTH_KEY"

#define PRICE_PATH "https://carburanti.mise.gov.it/ospzApi/search/zone"

#define SECRET_BOT_TOKEN "Telegram_CHAT_TOKEN"
#define SECRET_CHAT_ID "Telegram_CHAT_ID"

#define VERS_SW 1
```

↓ Listato 3

```
boolean Coordinates::getLatLong(String city0){
    HTTPClient http;
    boolean ret = false;

    String path = url0 + city0 + key0;

    http.begin(path.c_str());

    int httpResponseCode = http.GET();

    if (httpResponseCode>0) {
        //Serial.print("HTTP Response code: ");
        //Serial.println(httpResponseCode);
        String payload = http.getString();
        //Serial.println(payload);

        DynamicJsonDocument doc(2048);
        DeserializationError error = deserializeJson(doc, payload);
        if (error) {
            Serial.print(F("deserializeJson() failed: "));
            Serial.println(error.f_str());
            return false;
        }

        if(doc["longt"]=="0.00000" && doc["latt"]=="0.00000"){
            ret = false;
        }else{
            long0 = doc["longt"].as<String>();
            lat0 = doc["latt"].as<String>();
            ret = true;
        }
    }
    else {
        Serial.print("Error code: ");
        Serial.println(httpResponseCode);
        ret = false;
    }
    http.end();
    return ret;
}
```

are un file .h creiamo anche il suo corrispondente .cpp nel quale inseriremo il body e l'implementazione delle varie funzioni dichiarate in precedenza. Il risultato ottenuto dovrà essere simile a quanto proposto nella **Fig. 4**.

Come già accennato, il primo include è molto facile poiché conterrà solamente i define delle varie variabili usate a livello globale. In caso cambiasse la configurazione della rete o di Telegram potremmo agire solamente su questo blocco di codice (**Listato 2**) senza minimamente intaccare il programma principale. A riguardo vi ricordiamo che per far funzionare il sistema è indispensabile editare la porzione di codice contenente i define (quindi quella proposta nel **Listato 2**) e scrivere al posto del termine WiFi SSID tra virgolette l'SSID della rete WiFi cui il vostro hardware si connetterà per comunicare con il servizio on-line

<https://carburanti.mise.gov.it/ospzApi/search/zone;> sempre per garantire la connessione, dovrete scrivere al posto del virgolettato WiFi Password la passkey utilizzata per l'accesso al WiFi. Una volta personalizzato il codice potete salvare il file e caricarlo nella scheda.

Andiamo ora a disegnare quello che sarà il controller responsabile di tradurre in coordinate un'eventuale città che verrà inviata dall'esterno.

Apriamo la tab con coordinates.h e compiliamo la prima classe come proposto qui di seguito:

```
#define coordinates_h

#if ARDUINO >= 100
#include "Arduino.h"
#else
#include "WProgram.h"
```



```
#endif

#include <ArduinoJson.h>
#include <WiFi.h>
#include <HttpClient.h>

class Coordinates{
public:
    Coordinates();
    Coordinates(String url1, String key1);
    boolean getLatLng(String city0);
    String getLatt();
    String getLongt();
    void setLatt(String d0);
    void setLongt(String d0);

private:
    String url0;
    String key0;
    String lat0;
    String long0;
};
```

Qui, come possiamo notare, abbiamo tutti i prototipi delle funzioni che andremo a richiamare nel nostro main.ino e che implementeremo nel file **.cpp**. Si tratta di una classe abbastanza “asciutta” e fra le funzioni corrispondenti vi è il costruttore:

```
Coordinates::Coordinates(){
}

Coordinates::Coordinates(String url1, String key1){
    url0 = url1;
    key0 = key1;
}
```

Ovviamente in override, per poter istanziare sia senza parametri che con, dando una maggiore flessibilità. Tutte le classi a seguire sfrutteranno questo aspetto.

In ingresso abbiamo il path per la chiamata e la api_key. In realtà avremmo potuto inserire anche solo la chiave lasciando il path come costante interno alla classe.

A seguire (**Listato 3**) troviamo il vero cuore pulsante di questo oggetto, ovvero la funzione responsabile di agganciare il servizio esterno Geocode.xyz. In entrata abbiamo solo una stringa che rappresenta la città inserita dall'utente mentre in uscita un boolean che ci permette di capire se la chiamata è andata bene o meno.

Questa rappresenta una chiamata REST ad un API in GET. Sicuramente è il più semplice e il più immediato. È consigliato soprattutto in quelle richieste in cui è utile salvare nell'URL i parametri, ad esempio per poter essere cachati. Un classico caso d'uso è una query di ricerca.

L'ingaggio di queste richieste API può essere anche in POST, che vedremo più avanti e in questo caso aumenta di gran lunga la sicurezza.

Il metodo POST si differenzia da GET in quanto i parametri della richiesta non vengono passati in query string e quindi non possono essere tracciati nemmeno negli access log dei web server.

Caso d'uso comune di una richiesta in POST è un form che invia dati personali, come in una registrazione, login form.

Inoltre è importante chiamare una sola volta il metodo **http.getString()** ed assegnarlo ad una variabile per poterlo utilizzare in più momenti nel corso della funzione.

In questo caso per esempio sia per leggerlo che passarlo alla funzione di decodifica JSON.

Questo perché tale metodo va a scaricare il dato e se dopo una prima chiamata il dato sarebbe quello che ci ha risposto realmente il server, nella seconda risulterebbe nullo andando ad invalidare il nostro codice.

Occorre sempre liberare la risorsa attraverso la funzione **http.end()**.

Andiamo a rifinire questo oggetto con le funzioni pubbliche di set e get delle due variabili lat0 e long0 in modo da poterle usare ovunque; il codice interessato è quello proposto nel **Listato 4**.

Un'altra classe veramente elementare e responsabile dell'ingaggio di un altro servizio è la fuelprice.

Il servizio in questione, di cui abbiamo già accennato nel capitolo precedente, recupererà tutti i distributori di benzina di una specifica zona.

Troviamo un header, fuelprice.h, nel quali sono dichiarati i prototipi delle funzioni.

Oltre alla dichiarazione troviamo un body nel quale, così come prima, si intravedono poche funzioni, prime fra tutti i costruttori:

```
FuelPrice::FuelPrice(){
    doc = new DynamicJsonDocument(16500);
}

FuelPrice::FuelPrice(String url1)
{
    doc = new DynamicJsonDocument(16500);
    url0 = url1;
}
```



Listato 4

```
#if ARDUINO >= 100
#include "Arduino.h"
#else
#include "WProgram.h"
#include "pins_arduino.h"
#include "WConstants.h"
#endif

#include "coordinates.h"
#include <ArduinoJson.h>
#include <WiFi.h>
#include <HTTPClient.h>

Coordinates::Coordinates(){

}

Coordinates::Coordinates(String url1, String key1){
    url0 = url1;
    key0 = key1;
}

boolean Coordinates::getLatLong(String city0){
    HTTPClient http;
    boolean ret = false;

    String path = url0 + city0 + key0;

    http.begin(path.c_str());

    int httpResponseCode = http.GET();

    if (httpResponseCode>0) {
        Serial.print("HTTP Response code: ");
        Serial.println(httpResponseCode);
        String payload = http.getString();
        Serial.println(payload);

        DynamicJsonDocument doc(2048);
        DeserializationError error = deserializeJson(doc, payload);
        if (error) {
            Serial.print(F("deserializeJson() failed: "));
            Serial.println(error.f_str());
            return false;
        }

        if(doc["longt"]=="0.00000" && doc["latt"]=="0.00000"){
            ret = false;
        }else{
            long0 = doc["longt"].as<String>();
            lat0 = doc["latt"].as<String>();
            ret = true;
        }
    }
    else {
        Serial.print("Error code: ");
        Serial.println(httpResponseCode);
        ret = false;
    }
    // Free resources
    http.end();
    return ret;
}

String Coordinates::getLatt(){
    return lat0;
}

String Coordinates::getLongt(){
    return long0;
}

void Coordinates::setLatt(String d0){
    lat0 = d0;
}

void Coordinates::setLongt(String d0){
    long0 = d0;
}
```

↓ Listato 5

```
DynamicJsonDocument FuelPrice::getItems(){
    return (*doc);
}

void FuelPrice::getPrice(String lat0, String long0){
    WiFiClientSecure client;
    client.setInsecure();

    HTTPClient http;

    http.begin(client, url0);

    //http.addHeader("Content-Type", "application/x-www-form-urlencoded");
    http.addHeader("Content-Type", "application/json");
    String httpRequestData = "{\"points\": [{\"lat\": \""+lat0+"\", \"lng\": \""+long0+"\"}], \"radius\": 5}";
    int httpResponseCode = http.POST(httpRequestData);

    //Serial.print("HTTP Response code: ");
    //Serial.println(httpResponseCode);

    String payload = http.getString();
    //Serial.println(payload);

    DeserializationError error = deserializeJson((*doc), payload);
    if (error) {
        Serial.print(F("deserializeJson() failed: "));
        Serial.println(error.f_str());
    }

    //Serial.println((*doc)["results"][1]["name"].as<String>());

    http.end();
}
```

↓ Listato 6

```
#define EEPROM_SIZE 1024
//display_touch display0;
static LGFX lcd;
static LGFX_Sprite sprite(&lcd);
static LGFX_Sprite body(&lcd);

DisplayTouch dis;
TimesUNI timeUNI; //FACOLTATIVO
Coordinates myCoords;
FuelPrice myPrice;
telegram_bot myBot;
E le costanti necessarie:
const uint32_t BG_BAR_COLOR = lcd.color888(42, 59, 201);
const uint32_t TXT_BAR_COLOR = lcd.color888(255, 255, 255);
const uint32_t BG_CARD_COLOR = lcd.color888(192, 192, 192);
const uint32_t TXT_CARD_COLOR_M = lcd.color888(0, 0, 0);
const uint32_t TXT_CARD_COLOR_L = lcd.color888(127, 127, 127);
const int HOME_PAGE = 0;
const int MENU_PAGE = 1;
const int CONNECT_PAGE = 2;
const int LOAD_PAGE = 3;
const int TOUCH_BACK_BT_X0 = 10;
const int TOUCH_BACK_BT_Y0 = 180;
const int TOUCH_BACK_BT_X1 = 50;

const int TOUCH_FOR_BT_X0 = 470;
const int TOUCH_FOR_BT_X1 = 430;

String botToken = SECRET_BOT_TOKEN;
String CHAT_ID = SECRET_CHAT_ID;
String COORDINATES_PATH0 = COORDINATES_PATH;
String COORDINATES_KEY0 = COORDINATES_KEY;
String PRICE_PATH0 = PRICE_PATH;
const int TIME_REFRESH = SECRET_TIMEREFRESH;
int versionSW = VERS_SW;
```



Anche in questi costruttori, così come per i precedenti, prendiamo in ingresso la URL del path remoto cui punteremo per accedere alla risorsa. Inizializziamo anche la variabile `doc`, che sarà l'oggetto che conterrà un array con tutti i vari distributori di benzina, che renderemo fruibile e pubblica attraverso la `getItems()` nella maniera proposta nel **Listato 5**.

Quest'ultima funzione consente di inviare una richiesta POST ad un secure provider, ovvero ad un fornitore che espone su un canale https. Poi a differenza della GET, qui dobbiamo aggiungere un HEADER per dichiarare il content-type e una REQUEST DATA.

Dopo aver salvato la risposta in una variabile di tipo String cerco di deserializzarla all'interno dell'oggetto `doc`. L'ultima fra le classi obbligatorie è quella per il controllo dei click che vengono fatti sul nostro display. Questa potrebbe essere arricchita anche con tutte le funzioni di disegno del pannello ma a scopo didattico si è preferito lasciarle all'interno del main. Ed infine passiamo quindi al main `.ino`, ossia lo sketch principale, vale a dire il vero cuore pulsante del progetto firmware, che andrà a mescolare i vari connettori di cui abbiamo spiegato fino a qui. Al di là degli include, già menzionati, istanziamo tutti gli oggetti come si vede nel **Listato 6**. Come ben sappiamo, il ciclo di Arduino si basa su due macro funzioni: Setup e Loop. La prima è responsabile di tutte le inizializzazioni e della prima fase di "messa in moto" dell'apparato. Come mostra il **Listato 7**, troviamo infatti l'inizializzazione della EEPROM, memoria residente

sull'ESP, che salverà i vari stati e le richieste inviate per poter dare dei valori predefiniti nel caso di riavvio del dispositivo.

Dove ovviamente per leggere e scrivere usiamo due funzioni basate sulla `EepromStream`, per facilitare le operazioni:

```
void writeStringToEEPROM(int con, String res){
    EepromStream eepromStream(0, 1024);
    //serializeJson(docRes, res);
    eepromStream.print(res);
    eepromStream.flush();
    EEPROM.commit();
}

String readStringFromEEPROM(int addrOffset){
    EepromStream eepromStream(0, 1024);
    return eepromStream.readString();
}
```

Sempre a questo livello disegniamo all'interno del nostro LCD.

```
dis.setMode(HOME_PAGE);
drawHeaderBar(dis.getMode(), timeUNI.getHours(),
timeUNI.getDays());
drawBody(dis.getMode(), true, 0);
lcd.startWrite();
```

Il nostro template di layout si basa principalmente su due aree: la prima, header, è la tipica barra in alto nel display tipica del mondo embedded, la ritroviamo infatti in molti sistemi per smartphone o PLC che devono generare un output a video.

↓ Listato 7

```
EEPROM.begin(EEPROM_SIZE);
String myMem = readStringFromEEPROM(0);
Serial.println(myMem);
DynamicJsonDocument doc(2048);
DeserializationError error = deserializeJson(doc, myMem);
if (error) {
    myBot.setOffsetString(428401491);
    myBot.setCity("Guastalla");
    myCoords.setLatt("");
    myCoords.setLongt("");
    writeStringToEEPROM(0, "{\"vers\":\""+String(versionSW)+"\", \"update_id\":\"428401491\", \"city\"
: \"Guastalla\", \"latt\":\"\", \"longt\":\"\"}");
} else {
    myBot.setOffsetString(doc["update_id"]);
    myBot.setCity(doc["city"]);
    myCoords.setLatt(doc["latt"]);
    myCoords.setLongt(doc["longt"]);
}
```

Listato 8

```

int ind = 0;

void loop() {
    std::int32_t tx, ty, tc;
    tc = lcd.getTouch(&tx, &ty);
    getEvent(tc, tx, ty);

    //FACOLTATIVA
    if(timeUNI.getEnvolveTIME()){
        drawHeaderBar(dis.getMode(), timeUNI.getHours(), "");
    }
}

void getEvent(int tc, int tx, int ty){
    if(tc==1 && tx >= (TOUCH_BACK_BT_X0-5) && tx <= (TOUCH_BACK_BT_X1+5) && ty >= 55 &&
        ty <= 310 && (dis.getIndex()-1)>=0){
        dis.setIndex(dis.getIndex()-1);
        drawBody(dis.getMode(), false, 0);
        delay(500);
    } else if(tc==1 && tx >= (TOUCH_FOR_BT_X1-5) && tx <= (TOUCH_FOR_BT_X0+5) && ty >= 55 &&
        ty <= 310 && (dis.getIndex()+1)<myPrice.getItems()["results"].size()){
        dis.setIndex(dis.getIndex()+1);
        drawBody(dis.getMode(), false, 0);
        delay(500);
    } else if(tc==1 && tx>= 370 && tx <= 460 && ty >= 10 && ty <= 40){
        dis.setIndex(0);
        int resMsg = myBot.getMessages();
        if(resMsg==1){
            drawBody(Load_PAGE, false, 0);
            String city = myBot.getCity();
            if(city!=" " && myCoords.getLatLong(city)){
                myPrice.getPrice(myCoords.getLatt(), myCoords.getLong());
                myBot.sendMessage("[INFO] Settings changed");
            } else if(city==" " && myBot.getLatt()!=" " && myBot.getLong()!=" "){
                myBot.setCity("");
                myCoords.setLatt(myBot.getLatt());
                myCoords.setLongt(myBot.getLongt());
                myPrice.getPrice(myCoords.getLatt(), myCoords.getLongt());
            } else{
                myBot.sendMessage("[INFO] No city found");
            }
        } else if(resMsg==2){
            float pb = 100000;
            float pd = 100000;
            String namePB = "";
            String namePD = "";
            String type1= "Benzina";
            String type2= "Gasolio";
            for(int i=0;i<myPrice.getItems()["results"].size();i++){
                String tmpName = myPrice.getItems()["results"][i]["name"].as<String>();
                for (int j=0; j < myPrice.getItems()["results"][i]["fuels"].size(); j++){
                    String type = myPrice.getItems()["results"][i]["fuels"][j]["name"].as<String>();
                    float tmpPrice = myPrice.getItems()["results"][i]["fuels"][j]["price"].as<String>().toFloat();
                    if(type.equals(type1) && tmpPrice<pb){
                        namePB = tmpName;
                        pb = tmpPrice;
                    } else if(type.equals(type2) && tmpPrice<pd){
                        namePD = tmpName;
                        pd = tmpPrice;
                    }
                }
            }
            myBot.sendMessage("[INFO] Il miglior prezzo Benzina è di "+String(pb)+"", presso
                "+namePB+"\nIl miglior prezzo Gasolio è di "+String(pd)+"", presso "+namePD);
        } else{
            myPrice.getPrice(myCoords.getLatt(), myCoords.getLongt());
        }
        drawBody(dis.getMode(), false, 0);
        String tmp = "{\n\"vers\":\n"+String(versionSW)+"\n,\n\"update_id\":\n"+myBot.getOffsetString()+"\n",
            \n\"city\":\n"+myBot.getCity()+"\n,\n\"latt\":\n"+myCoords.getLatt()+"\n,\n\"longt\":\n"+
            myCoords.getLongt()+"\n}";
        writeStringToEEPROM(0,tmp);
    }
}
    
```


Qui, volendo possiamo arricchirlo con l'orario oppure un tasto per commutare la modalità e passare dalla sola visualizzazione ad un menu di configurazione.

La seconda area, vale a dire il body, è il corpo nel quale visualizzeremo, in primis, messaggi per l'utente e poi, una volta scelto il punto, i vari distributori e magari due frecce per scorrerli in tutta comodità.

L'altra macro area in cui si suddivide il ciclo di vita di un programma per Arduino o micro di questo tipo è la funzione di Loop. Questa viene chiamata costantemente ed è la vera "vita" del programma: un ciclo continuo (un loop...) delle istruzioni contenute al suo interno; come si vede chiaramente dal **Listato 8**, tale ciclo è davvero corto.

Di tipo void, senza alcun ritorno, è quella che svolge praticamente tutti i controlli e la parte di ridisegno. In ingresso prende i punti cliccati dall'utente e se avviene il click.

Questi punti sono poi all'interno di if che servono al firmware per determinare, in base alle coordinate sul display, quale tasto è stato premuto e quali azioni correlare.

I tasti principali sono tre: i primi due sono controlli, infatti riguardano le frecce "avanti" e "indietro" che visualizzeranno il distributore successivo o precedente. Qui si ha solo una fase di disegno, finalizzata ad aggiornare la grafica.

Mentre rimane più complesso l'ultimo tasto, ovvero "aggiorna". Questo avremmo potuto evitarlo controllando ad ogni ciclo di clock ma è oneroso e poco significativo ai fini didattici e in caso di informazioni abbastanza statiche come queste.

In questo tasto vi è inserita anche la parte che controlla il bot di Telegram e ci informa se arrivano o meno messaggi per cambiare la posizione attuale o richieste per conoscere i prezzi più convenienti. Come si può intuire dalla lettura e analisi del codice, se il messaggio in arrivo è di tipo "città" o "propria posizione" allora vengono ricercati tutti i distributori della zona e vengono disegnati a video tramite una paginazione; invece alla richiesta di un TAG specifico, nel nostro esempio "prezzi", vengono valutati attraverso un ciclo tutti i prezzi della zona e viene inviato un messaggio di risposta contenente i valori più bassi e a che distributore tra quelli dell'area selezionata fanno capo.

CONCLUSIONI

Bene, possiamo fermarci qui, dopo avervi introdotto la finalità di questo progetto ed averne snocciolato gli aspetti del firmware, che potete

scaricare per intero dal nostro sito www.elettronica.it insieme ai file del progetto. L'applicazione che è stata qui descritta è una base che fornisce il servizio essenziale, ma può costituire, per chi fra voi se la sente di "mettere mano" al codice, un punto di partenza per arricchirla e personalizzarla. A riguardo, qui di seguito vi forniamo qualche spunto, iniziando dall'oggetto TimesUNI che trovate all'interno del codice; questa domanda ad un server online l'orario e poi lo mantiene ad ogni ciclo di clock, funzionando a tutti gli effetti da orologio aggiornato in Rete.

Nella funzione di loop viene richiamata la getEnvolveTime che si occupa esattamente di questo: tenere allineati i minuti e le ore che passano.

Si può sviluppare una parte aggiuntiva che in caso di non utilizzo del dispositivo per un tot di minuti lo mette in standby oppure lo trasforma in un vero e proprio orologio.

Un ulteriore sviluppo potrebbe essere quello di programmare SSID e PASSWORD della rete direttamente da un menu di configurazione all'interno del touch. Nella parte di header può essere quindi aggiunto un tasto MENU in grado di passare ad una schermata di impostazioni. A tale scopo è importante capire come poter aggiungere, attraverso il versioning, i campi nel database temporaneo che storiemo nella EEPROM e come poter scrivere poi all'interno di questi.

Entrambe sono sfide molto complesse da superare, che però potrebbero migliorare il sistema e portarci una conoscenza maggiore del dispositivo e della programmazione su di esso. □



Cosa occorre?

Il display TFT da 3,5" con ESP32-S3 è disponibile presso Futura Elettronica (cod. ESP32S335D) a Euro 89,00. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

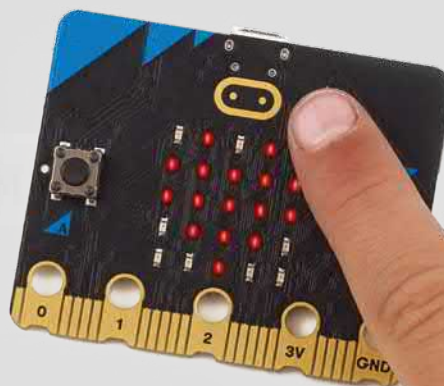
**Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 - www.futurashop.it**



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

ROBOT MINIONS



di GRETA
GALLI

Simpatico automa personalizzabile, con cui sperimentare la programmazione e il controllo del movimento.

L'estate scorsa è uscito al cinema un nuovo film dedicato ai Minions, dei personaggi immaginari gialli che parlano una lingua tutta loro e che seguono solo ed esclusivamente il loro capo nelle sue avventure, ma che sono anche molto amati dai bambini. In questo articolo vi proponiamo un'idea di progetto da sviluppare nelle vacanze di Pasqua che renda felici anche i più piccoli, perché si tratta di costruire un "robottino" il cui aspetto è quello di uno dei suggestivi personaggi della serie Minions e la cui realizzazione, oltre che semplice e in stile Maker, è un'occasione per fare un po' di robotica didattica.

MATERIALE NECESSARIO

Perché è da fare proprio nelle vacanze di Pasqua questo progetto? Perché abbiamo bisogno di un uovo di plastica delle sorprese, proprio quello che per l'appunto si trova nell'uovo di Pasqua della Kinder.

I Minions sono gialli e indossano delle casacche blu, quindi con una bomboletta spray possiamo colorare di conseguenza il nostro uovo delle sorprese, ma vi consigliamo di fare prima i fori per i componenti, così da non rovinare la vernice.

I componenti da utilizzare non sono vincolanti,

Fig. 1
Schematizzazione del ponte ad H: gli interruttori simulano il funzionamento dei transistor.

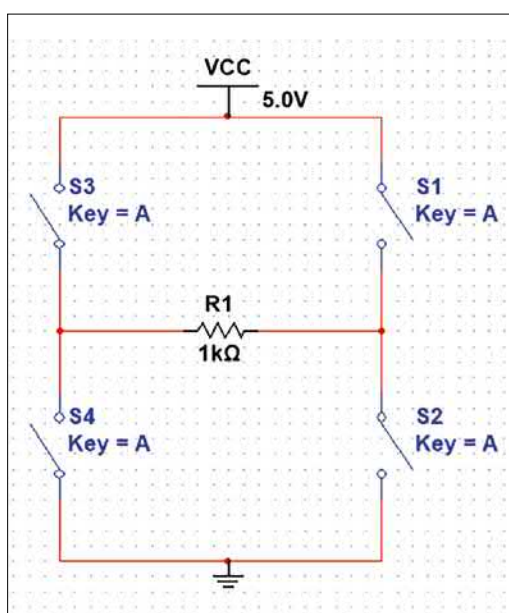
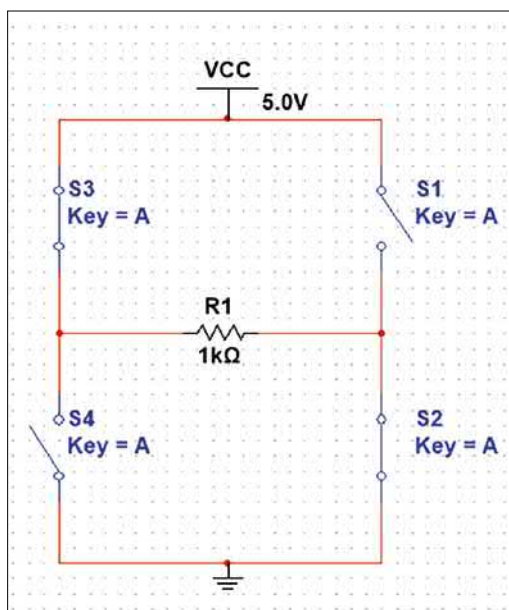


Fig. 2
Attivazione degli interruttori per avere 5V positivi sul lato sinistro del ponte.



quindi potete personalizzare il progetto sulla base della vostra creatività. Ma ora vi spieghiamo come l'abbiamo pensato noi.

Come unità logica dei Minions abbiamo optato per un Arduino Nano, una scheda a microcontrollore della famiglia Arduino conosciuta soprattutto per le sue dimensioni ridotte. Grazie a questa caratteristica diventa molto semplice nascondere all'interno dell'uovo. In alternativa potete usare un Arduino nano 33 IoT o un Esp32 nel caso vogliate avere anche una connettività e controllare da tablet o smartphone il robot.

Abbiamo poi bisogno di due motori per il movimento; a riguardo precisiamo che non dovranno spostare un grande peso e non avendo bisogno di chissà quale velocità, possiamo utilizzare due unità del motore reperibile on-line su www.futurashop.it con il codice prodotto YJFF280. Si tratta di un motore che si "accontenta" di tensioni di lavoro molto basse, come quelle ottenibili da batterie al litio o comuni pile a secco.

A questo punto viene spontaneo chiedersi come si pilotano questi motori? Potremmo collegarli direttamente ad Arduino? No, semplicemente perché questi motori necessitano sì di poca corrente, ma che una board Arduino non è in grado di erogare (tipicamente si parla di 20mA per ogni pin); inoltre necessitano di una tensione compresa tra i 6V e i 12V, e Arduino nano è in grado di erogare 5V dai pin.

Quindi come possiamo fare? Ebbene, abbiamo due metodi da scegliere: il primo è quello di utilizzare dei transistor, mentre il secondo consiste nell'usare un ponte H. Nel caso della prima scelta potete utilizzare un transistor di tipo NPN, come il BC337, calcolare la resistenza di base e aggiungere un diodo di protezione per ogni motore, ma il problema di questa soluzione è il fatto che il motore sarà in grado di girare solo in un verso.

Nel caso invece della seconda scelta si può utilizzare un ponte H per ogni motore, un circuito sempre fatto da dei transistor ma che ci consente di direzionare il verso della corrente e di conseguenza di far girare il motore in entrambi i sensi di marcia. Nella **Fig. 1** è illustrato lo schema di un ponte H fatto con degli interruttori, che rappresentano i transistor (bipolari o ad effetto di campo) normalmente utilizzati come interruttori allo stato solido nei ponti ad H, per capire meglio il funzionamento di questo circuito; proviamo dunque a spiegare il ponte ipotizzando in una prima fase che, come nella **Fig. 2**, andiamo a chiudere gli interruttori S3 e S2. Ebbene, in tale condizione avremo la corrente

che circolerà da VCC, attraverso S3, passerà nel carico (una resistenza, in questo caso, ma nella nostra applicazione il carico è un motore elettrico a spazzole in continua), arriverà da S2 e infine a GND, determinando la rotazione in un senso di marcia; invece nel caso, schematizzato nella **Fig. 3**, andiamo a chiudere gli interruttori S1 e S4, la polarità ai capi del carico si inverte e, nel caso del motore, l'albero gira al contrario rispetto al caso precedente.

Ma come si realizza in pratica un ponte ad H? Il circuito è facile, come vedete nelle figure appena citate, però siccome ciò richiederebbe la realizzazione di un circuito stampato (sebbene si possano montare i componenti su un pezzetto di basetta millefori...) è più comodo utilizzare una breakout board (se ne trovano diverse in commercio) già pronta che dispone dei contatti di uscita e di ingresso comandabili da livelli logici TTL. Nel nostro caso abbiamo scelto di utilizzare un modulo che già dispone di due ponte H, proprio quelli che ci servono per controllare i nostri motori, ma nel caso voleste crearvi il ponte H da zero vi lasciamo lo schema elettrico del circuito da realizzare: lo trovate nella **Fig. 4**.

Il modulo ponte H, da noi utilizzato e dettagliato nello schema di cablaggio visibile in **Fig. 5**, potete acquistarlo on-line su futuranet.it con il codice HBRIDGEMOD. Abbiamo detto che i motori necessitano di un'alimentazione superiore ai 5V e per questo motivo utilizzeremo un porta-batteria esterno per pile stilo che permette di disporre di una tensione di 6V se vi si montano quattro stilo, o

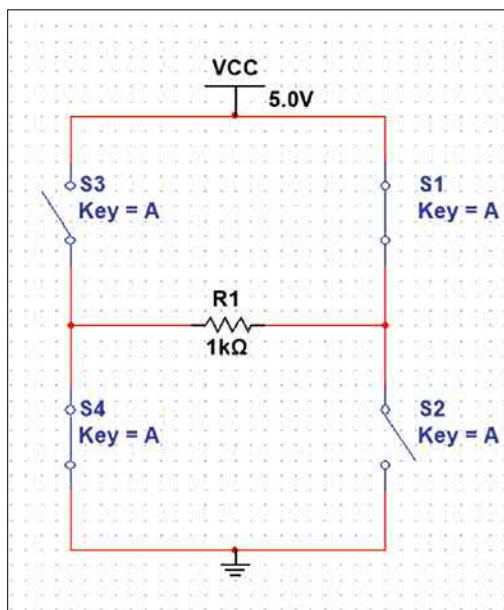


Fig. 3
Attivazione degli interruttori per avere 5V la polarità positiva sul terminale di sinistra della resistenza.

di 12V utilizzando due porta-batterie in serie (cod. BH343B).

Come ultimo elemento aggiungiamo un sensore ad ultrasuoni, ovvero un sensore che ci consente di misurare la distanza da un oggetto e inoltre, dato l'aspetto che i due trasduttori gli conferiscono, simile a due occhi, permette di creare anche gli occhi del nostro Minion robotico.

Il sensore a ultrasuoni cui ci riferiamo è acquistabile on-line sempre su www.futuranet.it (cod. MISDIST04); è in grado di misurare distanze comprese tra 2 cm e 4,5 m (questo significa che il nostro robottino potrà riconoscere ostacoli a questa

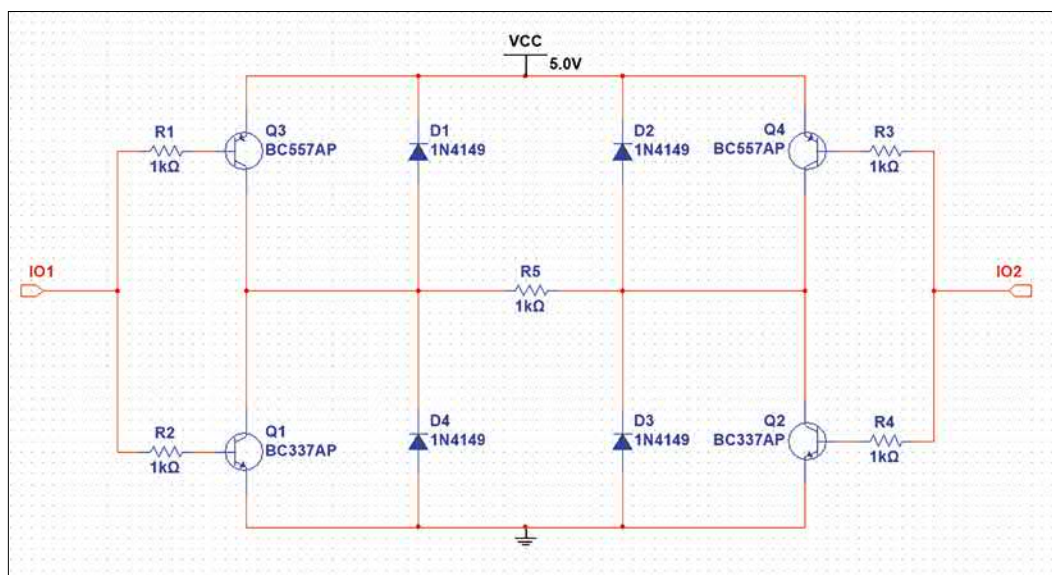
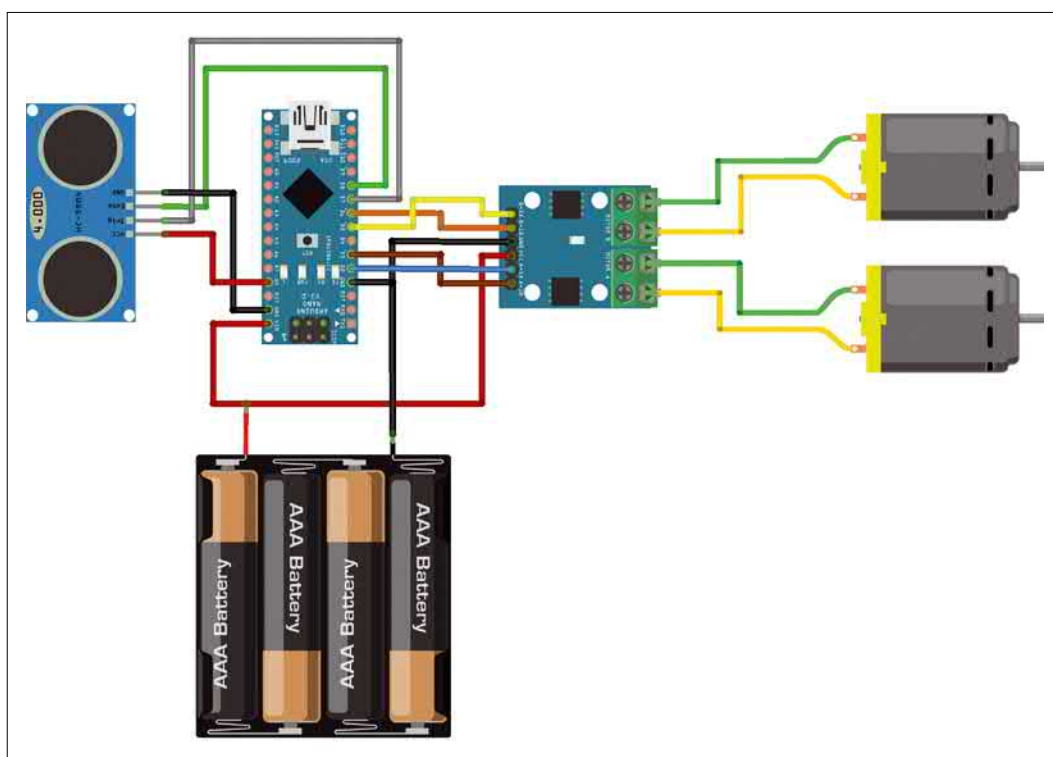


Fig. 4
Schema elettrico del ponte ad H.

➔ **Fig. 5**
Cablaggio dei componenti utilizzati nel robot.



distanza). È composto da due capsule ad ultrasuoni a 40 kHz (una trasmittente e una ricevente) e dall'elettronica di controllo, della quale tra breve spiegheremo il funzionamento.

STRUTTURA MECCANICA

Per quanto riguarda la struttura fisica del robot potete dare sfogo alla vostra fantasia: potrete usare delle ruote che si trovano in commercio, ma creare tutto da zero è un ottimo modo per rendere ancora più personale il nostro robot.

Per l'assemblaggio potete trarre spunto dalle foto del prototipo; nel nostro caso abbiamo utilizzato stampa 3D, viti, colla a caldo e cartoncino colorato, ma per fare le ruote potete utilizzare anche dei tappi delle bottiglie!

Nella sezione download della pagina di questa rivista sul nostro sito www.elettronica.in.it trovate, insieme ai file del progetto, i file .STL per stampare le ruote ed il "carrello" posteriore che supporta il portatile.

COLLEGAMENTI

I componenti e gli elementi che realizzano la parte elettronica del nostro Robot Minion devono essere interconnessi seguendo lo schema di cablaggio proposto nella **Fig. 5**.

Sulla base di questo disegno, come prima cosa col-

leghiamo l'alimentazione di ogni componente, dal vano batteria prendiamo il cavo nero, ovvero GND, e lo colleghiamo alla massa di Arduino, del modulo ponte H e del sensore ad ultrasuoni.

Il cavo rosso, ovvero VCC, lo colleghiamo al pin Vin della scheda Arduino e al pin VCC del ponte H, mentre il pin di alimentazione (VCC) del sensore ad ultrasuoni lo colleghiamo all'uscita 5V di Arduino Nano.

Il sensore ad ultrasuoni ha due contatti chiamati *trig* ed *echo* (oltre all'alimentazione) i quali consentono al sensore di funzionare come il sensore che il pipistrello utilizza per percepire gli ostacoli intorno ad esso: con un trasduttore piezoelettrico emittente irradia frontalmente, entro un cono ristretto, delle onde ultrasoniche alla frequenza di 40 kHz e quando queste colpiscono un corpo riflettente nel raggio di azione del dispositivo, parte di esse torna indietro e viene captata dalla seconda capsula piezo, stavolta in funzione di ricevente. Il segnale ai capi di quest'ultima viene amplificato e trattato da un comparatore per ottenere degli impulsi la cui larghezza è proporzionale alla distanza rilevata. Per generare un impulso di ultrasuoni il dispositivo richiede che il suo pin *trig* venga posto a livello alto per 10 ms (ogni commutazione 0/1 logico su di esso determina l'emissione di un treno di 8 onde ultrasoniche); se degli ultrasuoni tornano con

ampiezza sufficiente alla capsula ricevente, un apposito circuito genera i predetti impulsi rendendoli disponibili sul pin *echo*.

La durata dell'impulso di risposta al *trig* restituito dal modulo sul pin di uscita *echo* varia da 150 μ s (alla distanza di 2 cm) a 25 ms (in corrispondenza della massima distanza rilevabile, ossia 4,5 m) proporzionale alla distanza dell'oggetto.

Se il dispositivo non rileva alcun ostacolo entro i 4,5 m, produce un impulso di durata di circa 38 ms. Per calcolare la distanza in centimetri bisogna dividere la durata dell'impulso in μ s per 58. Il firmware che abbiamo scritto per il nostro robot si basa su tale proporzione per calcolare la distanza.

Nel nostro circuito i due pin *trig* ed *echo* vengono collegati a due pin digitali di Arduino.

Vediamo adesso il modulo ponte H, il quale possiede quattro morsetti divisi in due coppie, ad ognuno dei quali va collegato uno dei due cavi del motore (da saldare ai morsetti del motore); nel nostro caso non ci sono particolari problemi di polarità, perché anche scambiando i fili il motore gira ugualmente. Chiaramente il motore elettrico ha una polarità, nel senso che normalmente collegando il positivo sul positivo l'albero ruota in senso orario. Diciamo che la polarità è abbastanza irrilevante, perché alla peggio basta scambiare i fili degli ingressi del ponte, ovvero invertire il verso nel firmware, alle sezioni **void avanti()** e **void indietro()**.

Sul modulo troviamo poi quattro pin dedicati ai

quattro morsetti, che collegheremo ad altri quattro pin digitali di Arduino Nano.

Per le connessioni potete utilizzare i classici cavetti jumper, ad esempio quelli reperibili da Futura Elettronica con il codice JUMPERFF40P.

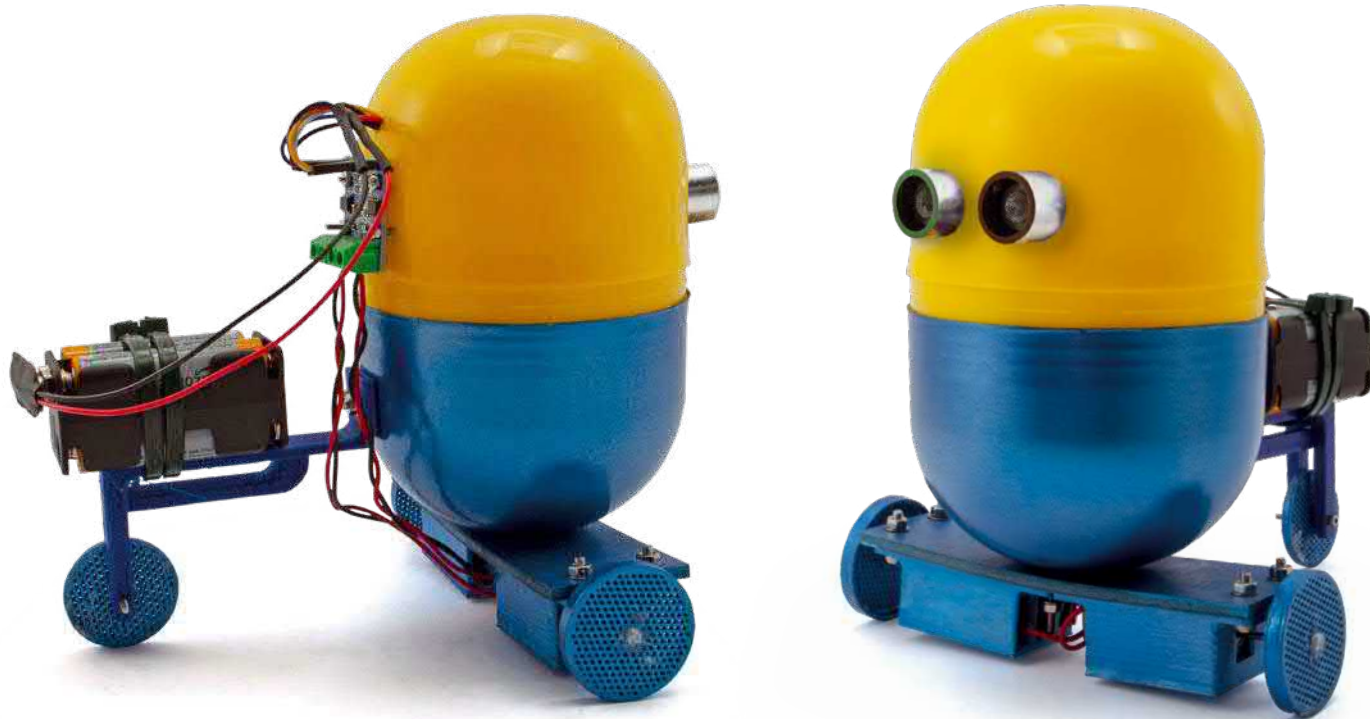
PROGRAMMAZIONE

Bene, vista la parte meccanica e l'elettronica (ossia l'hardware) del robot, passiamo agli aspetti firmware, quindi al caricamento dello sketch nel microcontrollore della scheda Arduino Nano.

Apriamo dunque l'Arduino IDE e iniziamo col dichiarare tutto ciò che abbiamo collegato, vale a dire le variabili associate al sensore a ultrasuoni e in generale le linee di I/O utilizzate per la gestione di esso e dei motori.

Riguardo al sensore di distanza, notate che non abbiamo fatto uso della specifica libreria per Arduino ma lo gestiamo a basso livello, quindi ci basiamo sugli impulsi che esso produce e le cui caratteristiche sono correlate alla distanza rilevata. Riguardo alle dichiarazioni dei pin utilizzati abbiamo due pin digitali per ogni motore e i pin *trig* ed *echo* del sensore. Abbiamo poi due variabili che ci serviranno per leggere la distanza letta dal sensore: *distanza* e *durata*, *distanza* può essere dichiarata come *int* in quanto conterrà un valore abbastanza piccolo da poter essere memorizzato in un *int*, mentre *durata*, che memorizzerà il tempo che passa da quando il pin *trig* manda un

← Il nostro robot montato.



▼ Listato 1

```
#define A1A 2
#define A1B 3
#define B1A 5
#define B1B 6

const int Trig = 7;
const int Echo = 8;

long durata;
int distanza;

void setup() {
  // put your setup code here, to run once:
  pinMode(A1A, OUTPUT);
  pinMode(A1B, OUTPUT);
  pinMode(B1A, OUTPUT);
  pinMode(B1B, OUTPUT);
  pinMode(Trig, OUTPUT);
  pinMode(Echo, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:

  avanti();

  digitalWrite(Trig, LOW);
  delayMicroseconds(2);

  digitalWrite(Trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Trig, LOW);

  durata = pulseIn(Echo, HIGH);
  distanza = durata * 0.0343 / 2;

  digitalWrite(Echo, LOW);

  Serial.println(distanza);

  if(distanza < 30){
    fermo();
    delay(1000);
    indietro();
    delay(800);
    fermo();
    delay(1000);
    destra();
    delay(1000);
    fermo();
    delay(500);
  }
}

void avanti(){
  digitalWrite(A1A, HIGH);
  digitalWrite(A1B, LOW);
  digitalWrite(B1A, LOW);
  digitalWrite(B1B, HIGH);
}

void indietro(){
  digitalWrite(A1A, LOW);
  digitalWrite(A1B, HIGH);
  digitalWrite(B1A, HIGH);
  digitalWrite(B1B, LOW);
}

void destra(){
  digitalWrite(A1A, HIGH);
  digitalWrite(A1B, LOW);
  digitalWrite(B1A, LOW);
  digitalWrite(B1B, LOW);
}
```

segnale e il pin echo lo memorizza, sarà di tipo long in quanto non sarebbe possibile utilizzare un int (negli int si possono memorizzare valori che vanno da -32.768 a 32.767).

Nel setup dichiariamo input e output: in questo caso come input abbiamo solo il pin echo del sensore.

Per semplificare il codice abbiamo deciso di creare una funzione per ogni movimento, ossia una avanti, indietro, destra e sinistra.

Chiaramente in avanti e indietro i motori gireranno uno opposto all'altro a pari velocità.

Per far ruotare un motore con il ponte H, come visto precedentemente, bisogna dare alimentazione ad un pin e mettere a massa l'altro, se si inverte il pin alimentato con quello a massa si ottiene il movimento del motore nel senso di marcia opposto, e se per caso vi trovate con un motore che non gira come dovrebbe vi basta invertire i cavi collegati ai morsetti oppure quelli collegati ai pin.

Per andare avanti ci devono quindi essere due pin alimentati e due spenti (uno per motore), per andare indietro si invertono i pin e per girare abbiamo due metodi: attiviamo solo un motore (quello opposto al senso di rotazione, quindi se si deve girare a destra si attiverà solo il motore di sinistra) oppure possiamo attivare un motore in un senso e uno nel senso opposto.

Nel primo caso il robot compierà un arco con raggio pari alla larghezza del robot in quanto il motore spento fa da perno, mentre nel secondo caso avremo un robot che gira su se stesso, come fanno i robot delle pulizie che abbiamo in casa per fare un paragone.

Alla domanda "quale metodo conviene?" rispondiamo che non ce n'è uno preferibile: dipende da quanto spazio il robot può occupare nel suo spostamento.

Nel loop come prima operazione andiamo a richiamare la funzione **avanti()** così da far muovere il nostro robot, successivamente andiamo a controllare la distanza; questa operazione può essere scritta nel loop oppure può avere una funzione dedicata **int distanza()** che poi viene richiamata.

Per calcolare la distanza dobbiamo attivare, ossia porre a livello alto il pin di *trig* per 10 millisecondi allo scopo di far generare al misuratore a ultrasuoni; le righe di codice che attivano e disattivano il pin trig sono le seguenti:

```
digitalWrite(Trig, LOW);
delayMicroseconds(2);
digitalWrite(Trig, HIGH);
```



```
delayMicroseconds(10);
digitalWrite(Trig, LOW);
```

Successivamente Arduino si pone in ascolto dello stato logico fornito dal pin *echo* e andrà a memorizzare la durata del relativo impulso nella variabile **durata**; una volta acquisita la durata, viene applicata la formula:

```
distanza = durata * 0,0343 / 2
```

contenuta nella riga di codice corrispondente. In pratica moltiplicando per 0,0343 e dividendo il risultato per 2 otteniamo lo stesso valore che otterremmo dividendo per 58 la durata dell'impulso. Notate che il fattore 0,0343 deriva dalla velocità del suono, che è 340 m/s ed è insito nella correlazione tra distanza, che il sensore fornisce raddoppiata perché il tempo trascorso tra l'attivazione del *trig* e l'arrivo della risposta su *echo* corrisponde all'andata e al ritorno delle onde a 40 kHz. Siccome la distanza (*d*) è data dal prodotto della velocità del suono (*v*, pari a circa 344 m/s nell'aria a 20 °C) per il tempo trascorso (*t*), ogni millisecondo di ritardo indica una distanza di 0,344/2 m, quindi 17,24 cm. Infatti se nella formula:

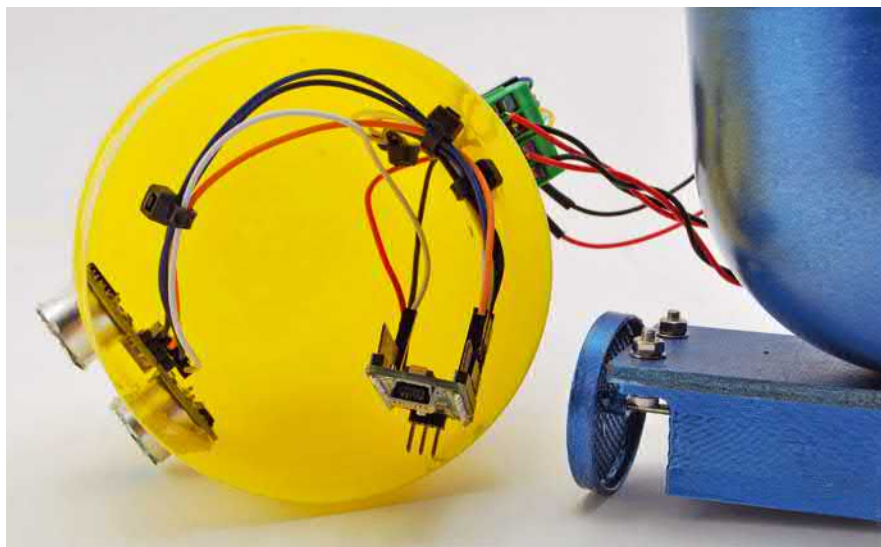
$$d = v \times t$$

sostituiamo i valori, otteniamo:

$$d = 344 \text{ m/s} \times 10^{-3} \text{ s} = 0,344 \text{ m}$$

Dividendo per due risultano 17,2 cm.

Con un costrutto **if** controlliamo se questa distanza è minore di un valore a nostra scelta e in caso affermativo fermiamo il robot, lo facciamo tornare indietro e girare verso un'altra direzione, in modo



da evitare l'ostacolo.

Tutto quanto appena spiegato apparirà più chiaro leggendo il **Listato 1**, che riporta il codice Arduino utilizzato per gestire il nostro piccolo robot.

 L'elettronica all'interno del robot.

CONCLUSIONI

Bene, con questo abbiamo terminato.

Con la speranza che questo progetto di riciclo creativo vi sia piaciuto e che gli abbiate dato la forma che più vi aggrada, vi ricordiamo che nessun robot è perfetto, ma l'importante è che piaccia a voi! Questo vale anche per il firmware, che potrete personalizzare a vostro piacimento per adattare ancora di più il robot al vostro gusto o alle vostre esigenze, scegliendone il comportamento. 

Listato 1 - continua

```
void sinistra(){
  digitalWrite(A1A, LOW);
  digitalWrite(A1B, LOW);
  digitalWrite(B1A, LOW);
  digitalWrite(B1B, HIGH);
}

void fermo(){
  digitalWrite(A1A, LOW);
  digitalWrite(A1B, LOW);
  digitalWrite(B1A, LOW);
  digitalWrite(B1B, LOW);
}
```



Cosa occorre?

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

La board Arduino Nano (cod. ARDUINONANO) è disponibile a Euro 29,00, il sensore ad ultrasuoni (cod. MISDIST04) a Euro 5,00, il ponte H (cod. HBRIDGEMOD) a Euro 3,90, il motore (cod. YJFF280) a Euro 2,90 l'uno.

Il materiale va richiesto a:

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

Stop ai fili con gli interruttori wireless a 433 MHz!

Interruttori wireless da utilizzare in abbinamento al ricevitore wireless a 433 MHz a 1 canale (FR790). Non richiedono nessuna alimentazione in quanto si auto-alimentano tramite un generatore elettrodinamico posto al loro interno. Ideali per controllare facilmente senza fili le luci in abitazioni, garage, negozi o altri dispositivi funzionanti a 220 VAC con carico inferiore a 1000 W e in tutte le situazioni in cui è necessario installare un interruttore ad un impianto già esistente senza aggiungere cavi di collegamento. La procedura di abbinamento tra ricevitore e pulsante è molto semplice e offre la possibilità di memorizzare fino a 5 dispositivi.

Interruttori wireless a 1 o 2 canali a 433 MHz autoalimentati

Ricevitore wireless 433 MHz 1 Canale - 220 VAC



€ 11,00

cod. FR791



€ 19,00

cod. FR792



€ 18,00

cod. FR790

Prezzi IVA inclusa



Ideali per:



Illuminazione



Ventilatori



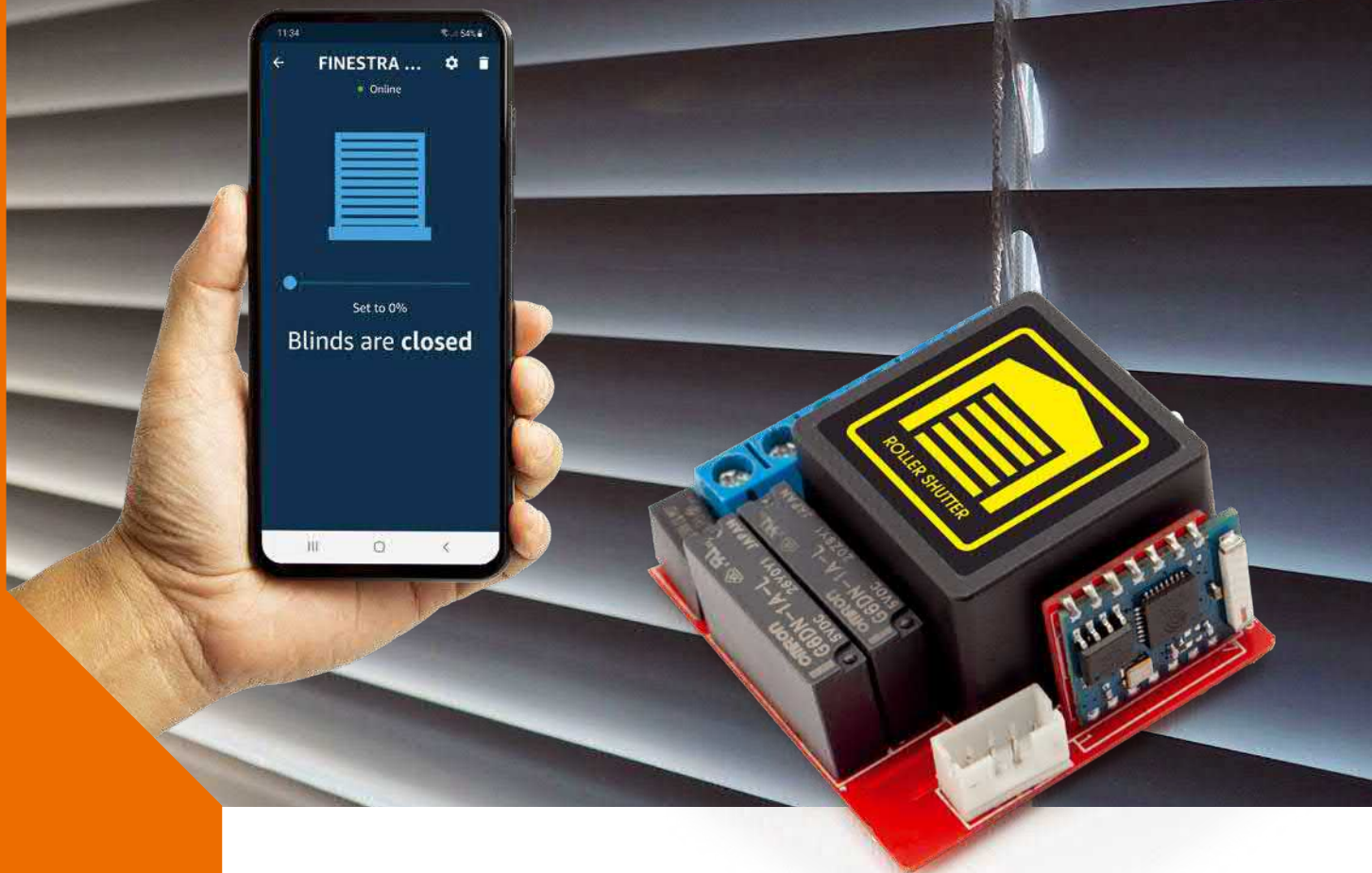
Elettrodomestici



Utilizzi industriali

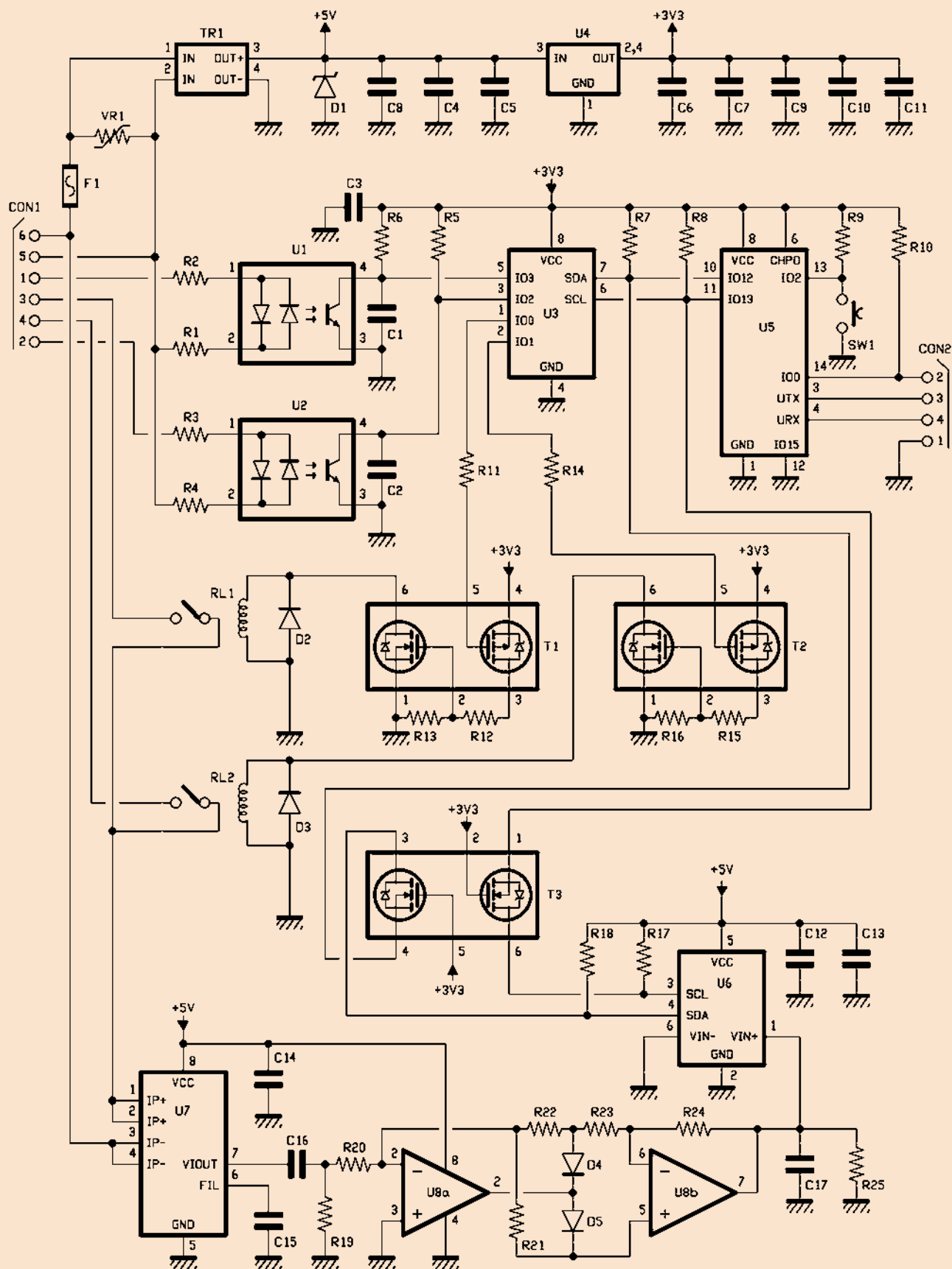
SMART ROLLER SHUTTER

di MAURIZIO ŠKERLIČ



Permette il comando di tapparelle motorizzate mediante assistente vocale Amazon Alexa, Google assistant oppure tramite App; connesso in WiFi, è programmato con un firmware che si appoggia al server Sinric Pro.

Il tema wireless è strettamente legato alla domotica e all'IoT, ambiti che affrontiamo con entusiasmo e per cui vi proponiamo mensilmente progetti basati su soluzioni di prototipazione già disponibili e dotate di connettività di rete. In questo articolo ci riallacciamo alla "saga" dei dispositivi Smart basati sul modulo WiFi ESP8266 (Smart Switch, Smart Dimmer e via di seguito), proponendovi un nuovo progetto per la domotica, che abbiamo chiamato "Smart roller shutter": si tratta di un dispositivo "intelligente" dedicato al controllo delle tapparelle elettriche mediante l'assistente vocale Amazon Alexa o Google assistant oppure tramite App da smartphone.



L'installazione non prevede assolutamente nessuna modifica dell'impianto elettrico, né tantomeno la sostituzione del motore della tapparella oppure la sostituzione dei pulsanti o interruttori auto-bloccanti preesistenti. Basterà cablare opportunamente il nostro dispositivo e in pochi minuti avrete trasformato la vostra tapparella in un dispositivo smart. Le sue dimensioni contenute gli permettono di essere installato in qualsiasi scatola di derivazione per impianti elettrici oppure montato sul supporto portafrutto nelle scatole rettangolari 503 utilizzando gli appositi adattatori, stampabili mediante stampante 3D, scaricabili dal nostro sito.

SCHEMA ELETTRICO

Chi ci segue da tempo avrà notato la somiglianza tra lo "Smart roller shutter" e il "Double smart switch" presentato nella pubblicazione di febbraio 2022, numero 261.

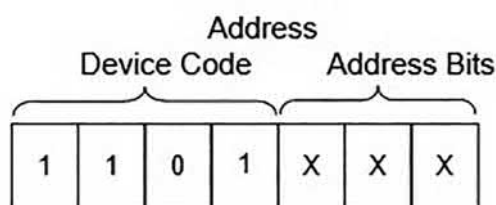
Andiamo quindi ad analizzarlo partendo dal modulo ESP03, siglato U5 e basato sul microprocessore Espressif ESP8266 con architettura RISC a 32 bit, caratterizzato da una frequenza di funzionamento a 80 MHz ed equipaggiato con una memoria flash da 1MB. Il circuito viene alimentato da un alimentatore switching integrato a bordo scheda, che eroga 5 V di tensione e 1 A di corrente; nello schema elettrico, tale modulo è siglato TR1. La tensione di uscita di questo dispositivo viene abbassata a 3,3 V dal regolatore di tensione NPC1117, nello schema elettrico siglato U4. La tensione a 5 V viene filtrata dai condensatori ceramici multistrato C4, C5 e C8 mentre quella a 3,3 V dai condensatori C6, C7, C9, C11 e C12 indispensabili per evitare reset o crash di sistema casuali. Ricordiamo che il processore Espressif è molto sensibile ai cali di tensione, ai disturbi elettromagnetici, detti EMI (Electro Magnetic Interference), e soprattutto all'eccessivo ripple tipico degli alimentatori switching in particolar modo quelli a basso costo. Queste sono le principali cause per il mancato funzionamento dei progetti realizzati dai neofiti che si cimentano per la prima volta con i processori della famiglia ESP, soprattutto quando i prototipi vengono montati su breadboard e i collegamenti sono realizzati con cavetti in modo caotico e approssimativo. Oppure dove, per la fretta o per pigrizia, l'alimentazione non viene quasi mai disaccoppiata o filtrata con i condensatori. Chiusa questa doverosa parentesi continuiamo con l'analisi dello schema elettrico. I piedini 10 (GPIO 12) e 11 (GPIO 13) dell'ESP03, rispettivamente SDA e SCL, sono utilizzati per la comunicazione I²C con

l'integrato PCA9536 un espansore di porte input output a 4 bit, nello schema elettrico siglato U3, e con l'integrato ADC (analog to digital converter) da 18 bit MCP3421 siglato U6. Facciamo presente che vengono commercializzati ben sette versioni di questo chip che si differenziano unicamente per il diverso indirizzo I²C. Quello utilizzato nel nostro progetto risponde all'indirizzo I²C 0x69 ed è siglato MCP3421A1. Se non lo trovate presso il vostro negozio di fiducia potete tranquillamente utilizzarne uno con indirizzo I²C diverso, in questo caso dovrete opportunamente modificare lo sketch inserendo il corretto indirizzo di comunicazione. Ad esempio se acquisterete il chip MCP3421A2, come descritto nel datasheet, al numero binario 1101XXX sostituite le "X" con l'address bits del chip, che è definito dagli ultimi due valori della sigla. Nel nostro caso specifico è A2 che equivale all'address bit numero 2, il quale espresso in formato binario è 010. Dopo l'opportuna sostituzione l'indirizzo I²C diventa 1101010 che convertito in esadecimale equivale a 0x6A (vedere Fig. 1).

Analizziamo ora la parte più importante del nostro dispositivo cioè quella che permette di impostare la posizione della tapparella senza l'ausilio di un sensore di posizione. Questo avviene grazie a un sensore di corrente a effetto Hall ACS712 prodotto dall'Allegro Microsystems, nello schema elettrico siglato U7, che rileva lo spegnimento del motore una volta raggiunto il finecorsa. Indispensabile soprattutto in fase di calibrazione per poter cronometrare il tempo che la tapparella impiega per chiudersi e aprirsi. Questo chip viene commercializzato in tre versioni a seconda della corrente massima rilevabile; da 30 A con una sensibilità di 66 mV/A, da 20 A con una sensibilità di 100 mV/A e da 5 A con una sensibilità di 185 mV/A. Nel nostro prototipo abbiamo utilizzato la versione da 5 A poiché ha la massima sensibilità (185 mV/A) e permette di rilevare l'intensità di corrente con una risoluzione migliore. Anche se in commercio esistono diversi sensori con sensibilità e caratteristiche anche migliori abbiamo optato l'utilizzo

CARATTERISTICHE TECNICHE

- ◆ Montaggio in scatola da incasso oppure su supporti Vimar, Gewiss o BTicino.
- ◆ Carico massimo 550 W.
- ◆ Protezione termica, da sovracorrenti e sovratensioni.
- ◆ Compatibile con gli assistenti vocali Amazon Alexa e Google Assistant.
- ◆ Configurazione tramite interfaccia web.
- ◆ Collegamento alla rete WiFi domestica.
- ◆ Funzionamento con pulsanti o interruttori interbloccati.
- ◆ Protezione con password.
- ◆ Reset password tramite pulsante.
- ◆ Rilevamento della corrente e della potenza assorbita.



◀ Fig. 1
Procedura per ricavare l'indirizzo I²C del convertitore A/D MCP3421.

Si tratta di un sensore di corrente prodotto dalla Allegro Microsystems e basato sull'effetto di Hall.

L'Allegro ACS712 è una soluzione economica e precisa per il rilevamento della corrente CA o CC in applicazioni industriali, commerciali, e sistemi di comunicazione. Applicazioni tipiche includono il controllo di motori elettrici, il rilevamento e la gestione del carico, modalità di alimentazione e protezione dai guasti da sovracorrente. Il dispositivo non è destinato ad applicazioni automobilistiche. Il dispositivo è costituito da un circuito Hall preciso, a basso offset e lineare con un percorso di conduzione in rame situato vicino alla "superficie del die, cioè la sottile piastrina di materiale semiconduttore sulla quale è stato realizzato il circuito elettronico del circuito integrato. La corrente scorre attraverso il percorso in rame e genera un campo magnetico che il sensore di Hall converte in una tensione proporzionale al campo.

La precisione del dispositivo è ottimizzata attraverso la stretta vicinanza del segnale magnetico al trasduttore di Hall e alla posizione fissa e stabile rispetto ad esso. L'uscita del dispositivo ha una pendenza positiva ($>V_{OUT}(Q)$) quando una corrente crescente scorre attraverso il percorso di conduzione primario di rame (dal pin 1 e 2, ai pin 3 e 4), che è il percorso utilizzato per il campionamento della corrente. La resistenza interna di questo percorso conduttivo è tipico di 1,2 mΩ e fornisce una bassa perdita di potenza. Lo spessore del conduttore di rame permette al dispositivo di sopportare fino a volte le condizioni di sovracorrente. I terminali del percorso conduttivo sono elettricamente isolati dai terminali del segnale (pin da 5 a 8). Ciò consente di utilizzare l'ACS712 nelle applicazioni che richiedono l'isolamento elettrico

senza l'uso di optoisolatori o altre costose tecniche di isolamento. L'ACS712 è fornito in un piccolo contenitore SOIC8 a montaggio superficiale. La tabella seguente descrive la funzione degli 8 piedini.

NUMERO PIN	NOME	DESCRIZIONE
1 e 2	IP+	Terminale per il rilevamento della corrente
3 e 4	IP-	Terminale per il rilevamento della corrente
5	GND	Terminale del segnale di terra
6	FILTER	Terminale per il condensatore esterno che imposta la larghezza di banda
7	VOUT	Uscita analogica
8	VCC	Terminale di alimentazione del dispositivo

Le principali caratteristiche del componente sono:

- segnale analogico a basso rumore;
- tempo di salita dell'uscita di 5 μs;
- larghezza di banda di 80 kHz;
- errore di uscita totale 1,5% a 25°C;
- resistenza del conduttore interno di 1,2 mΩ;
- tensione di isolamento minima dai pin 1-4 ai pin 5-8 di 2,1 kVRMS;
- funzionamento con alimentazione singola di 5V;
- Sensibilità di uscita da 66 a 185 mV/A.

dell'ACS712 esclusivamente per la sua facile reperibilità.

Questo chip misura la corrente alternata che fluisce attraverso i piedini 1, 2 e 3, 4, e la trasforma in una tensione proporzionale di 185 mV/A prelevabile dal piedino 7. Il segnale alternato viene raddrizzato dall'operazionale MCP6402, nello schema elettrico siglato U8, montato in configurazione di raddrizzatore a doppia semionda per misurare il valore efficace della corrente alternata.

Il segnale raddrizzato in uscita dal piedino 7 di U8 viene prelevato e convertito dal convertitore analogico digitale U6, che è un MCP3421. Poiché il sensore di corrente ACS712 funziona a 5 V, anche il segnale in uscita dal pin 7 deve essere raddrizzato e convertito da un chip alimentato con la stessa tensione. Difatti sia l'operazionale U8 in configurazione di raddrizzatore a doppia semionda e il convertitore A/D U6 vengono alimentati da una tensione a 5 V, mentre il livello logico del bus I²C del modulo ESP03, che comunica con il convertitore A/D, non è a 5V ma bensì a 3,3 V.

Per rendere compatibili i due segnali logici, che

presentano tensioni diverse, abbiamo dovuto usare un traslatore di tensione (*voltage translator*) detto level shifter, composto da un doppio MOSFET a canale N nello schema elettrico siglato con T3, e quattro resistenze R7, R8, R17 e R18 che a loro volta hanno anche la funzione di mantenere alto il livello del bus I²C.

I piedini 1 (IO0) e 2 (IO1) di U3 sono configurati come porte OUTPUT mentre il piedino 3 (IO3) e 5 (IO4) sono configurati come porta INPUT. Queste sono mantenute ad alto livello dalle resistenze R5 e R6 e i segnali, proveniente dal collettore dei transistor presenti nei fotoaccoppiatori U1 e U2, sono filtrati dai condensatore C1 e C2.

La tensione alternata a 230 V proveniente dai pulsanti collegati ai piedini 1 e 2 del connettore CON1, rispettivamente salita e discesa, viene disaccoppiata grazie ai doppi diodi LED, dei fotoaccoppiatori U1 e U2, e dalle resistenze limitatrici di corrente R1, R2 e R3, R4. Le resistenze R5 e R6, usate come resistenze di pullup, mantengono a potenziale alto rispettivamente le linee SCL e SDA del bus I²C. Il pulsante SW1 collegato al piedino

9 (GPIO2) del modulo ESP03, mantenuto ad alto livello dalla resistenza R9, serve a reimpostare la password predefinita di sistema "123456789", in caso dimenticaste quella memorizzata. I piedini 1 e 2 dell'integrato PCA9536 (U3), pilotano una coppia di relé elettromeccanici mediante due doppi MOSFET a canale N e canale P (una coppia incapsulata in un unico contenitore), nello schema elettrico siglati con T1 e T2. In parallelo alla bobina dei relé troviamo i diodi di riciclo (Flyback), siglati D2 e D3, i quali limitano l'ampiezza delle sovratensioni induttive, sopprimendo le extratensioni di segno opposto generate dalle bobine quando i relé vengono diseccitati e quindi proteggendo i due transistor T1 e T2 montati a valle. Il carico, nel nostro caso il motoriduttore della taparella, viene collegato sui piedini 3 salita (OUT1), 4 discesa (OUT2) e 5 comune (NEUTRO), mentre la tensione di rete, neutro e fase, viene collegata

rispettivamente ai piedini 5 e 6 di CON1. Il circuito è protetto dalle sovratensioni e sovracorrenti mediante il varistore VR1 e il fusibile F1, mentre il diodo transil unidirezionale D1 protegge il circuito da una tensione eccessiva in uscita dall'alimentatore in caso di guasto.

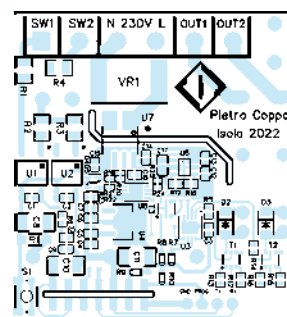
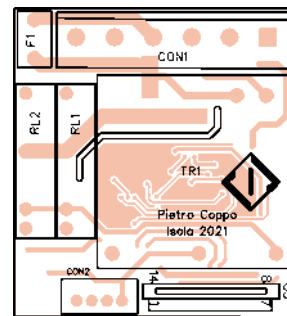
REALIZZAZIONE PRATICA

Prima di addentrarci nella realizzazione pratica vogliamo informarvi che a causa delle dimensioni ridotte del circuito stampato (imposte dalla necessità di renderlo compatibile con i frutti dei marchi più diffusi) per riuscire a far stare tutti i componenti sul PCB siamo stati costretti a utilizzare alcuni componenti SMD con package 0201. Ciononostante, la realizzazione del nostro circuito non richiederà conoscenze particolari né un'estrema precisione. Come sempre per il montaggio dei componenti suggeriamo l'utilizzo di un saldatore stilo a punta

piano di MONTAGGIO

Elenco Componenti:

R1, R2, R3, R4: 150 kohm 300V (0805)	U1, U2: TLP184
R5, R6: 470 kohm (0402)	U3: PCA9536TK
R7, R8: 2,2 kohm (0603)	U4: NCP1117ST33T3G
R9, R10: 10 kohm (0603)	U5: ESP03
R11, R14: 100 ohm (0402)	U6: MCP3421A1T-E/CH
R12, R15: 10 ohm (0402)	U7: ACS712ELCTR-05B-T
R13, R16, R23: 10 kohm (0402)	U8: MCP6402
R17, R18: 2,2 kohm (0402)	VR1: V430CH8
R19: 5,6 kohm (0201)	T1, T2: BSD235CH
R20, R21, R22, R24: 10 kohm (0201)	T3: BSS 138 DW
R25: 1 Mohm (0201)	D1: SMF5.0A
C1, C2, C16: 1 µF 6,3 VL ceramico (0402)	D2, D3: 1N4148W
C3, C4, C7, C9, C13, C14: 100 nF 6,3 VL ceramico (0402)	D4, D5: PMEG4002EB,115
C5, C6, C12: 10 µF 6,3 VL ceramico (0402)	RL1, RL2: G6DN-1A-SL
C8: 220 µF 6,3 VL ceramico (1206)	DC5
C10, C11: 100 µF 6,3 VL ceramico (1206)	F1: Fusibile T 2,5 A - 7,4x4,5x5,1mm
C15: 1 nF 6,3 VL ceramico (0402)	SW1: Pulsante SMD 4,6x2,8mm
C17: 22 µF 6,3 VL ceramico (0603)	TR1: RAC05-05 (5V / 5W)
	CON1: Morsettiere 6 vie passo 5 mm
	CON2: Connettore Maschio ad inserzione 4 poli passo 2 mm
	Varie:
	- circuito stampato S1665 (41x37 mm)



Il doppio diodo Schottky tipo BAT54 con cui potete sostituire la coppia di diodi D5 e D6 esiste in varie versioni, tutte incapsulate in package formato SOT-323; ciò che le distingue è la piedinatura, che differisce per offrire al progettista varie soluzioni. Ciascuna pin-out è contraddistinta da

un suffisso, ossia una coppia di lettere che segue la sigla. Nel nostro caso abbiamo scelto la SW e pertanto il circuito stampato è progettato per ospitare questa e non altre. Le possibili versioni, associate alle rispettive pin-out, sono dettagliate nell'immagine in questo riquadro.

Pin	Description	Simplified outline	Graphic symbol
BAT54AW			
1	cathode (diode 1)		
2	cathode (diode 2)		
3	common anode		
BAT54CW			
1	anode (diode 1)		
2	anode (diode 2)		
3	common cathode		
BAT54SW			
1	anode (diode 1)		
2	cathode (diode 2)		
3	cathode (diode 1), anode (diode 2)		
BAT54W			
1	anode		
2	not connected		
3	cathode		

sottile con potenza non troppo elevata (20 ÷ 25 watts) e del filo di lega saldante possibilmente non superiore a 0,7 mm di diametro. Consigliamo di usare un flussante di ottima qualità che non lascia scorie residue, possibilmente di bassa o media densità. Per posizionare i componenti aiutatevi con delle pinzette a punta fine possibilmente amagnetiche di acciaio inox oppure con le punte in plastica o ceramica, altrimenti magnetizzandosi, i componenti saranno attratti dalle punte metalliche, complicandovi il lavoro.

Fig. 2
Montaggio del modulo ESP03 in posizione verticale sul PCB.



Per chi fosse interessato a realizzare da solo lo stampato, facciamo presente che sul nostro sito web www.elettronica.in.it trovate i file Gerber del circuito stampato a doppia faccia ottenibile per fotoincisione, nonché il firmware; il tutto è disponibile nella sezione Download insieme ai file di questo numero della rivista.

Come al solito iniziate a saldare i semiconduttori e successivamente tutti i componenti passivi, facendo attenzione al corretto orientamento dei semiconduttori, come visibile nel piano di montaggio proposto nella pagina seguente. Nel saldare i componenti passivi, stagnate prima quelli con il contenitore più piccolo e successivamente quelli più grandi. Per ottimizzare lo spazio il modulo ESP03, che nello schema elettrico è siglato U5, andrà prima stagnato sul PCB adattatore e successivamente montato verticalmente come mostrato qualche pagina indietro, nella **Fig. 2**. Per ultimi andranno montati i componenti a foro passante, ossia le morsettiere, i relé e il modulo alimentatore switching.

Per chi non riuscisse a reperire i due diodi schottky "PMEG4002EB,115" abbiamo previsto sul PCB che si possano sostituire con un doppio diodo Schottky tipo BAT54, che è in formato SOT-323. Nell'acquisto di quest'ultimo ricordiamo di fare attenzione al suffisso della sigla, che indica la disposizione interna dei diodi; nel nostro caso va utilizzato il tipo SW. Per aiutarvi, dettagliamo le versioni nel riquadro dedicato al BAT54.

Una volta terminato il montaggio di tutti i componenti consigliamo sempre di pulire il PCB dal flussante con alcol isopropilico. Con una lente di ingrandimento controllate attentamente che non si siano formati ponticelli di stagno tra i piedini dei componenti in particolar modo tra i piedini di U1 e le piazzole dei componenti con case 0201. A questo punto potete alimentare il prototipo, seguendo il cablaggio illustrato nello schema elettrico.

Come sempre, e non ci stancheremo di ripeterlo, non toccate con le mani nude i circuiti quando sono allacciati alla rete elettrica a 230 V, altrimenti rischiate di rimanere folgorati. Siccome gli impianti elettrici domestici per legge hanno l'interruttore differenziale, impropriamente chiamato "salvavita", molti di voi pensano che questo interverrà e lo proteggerà da eventuali scosse. In realtà il differenziale scatta unicamente se ci sono dispersioni di corrente verso terra, quindi se toccate con le dita la fase e il neutro probabilmente il differenziale non interverrà e voi vi beccherete

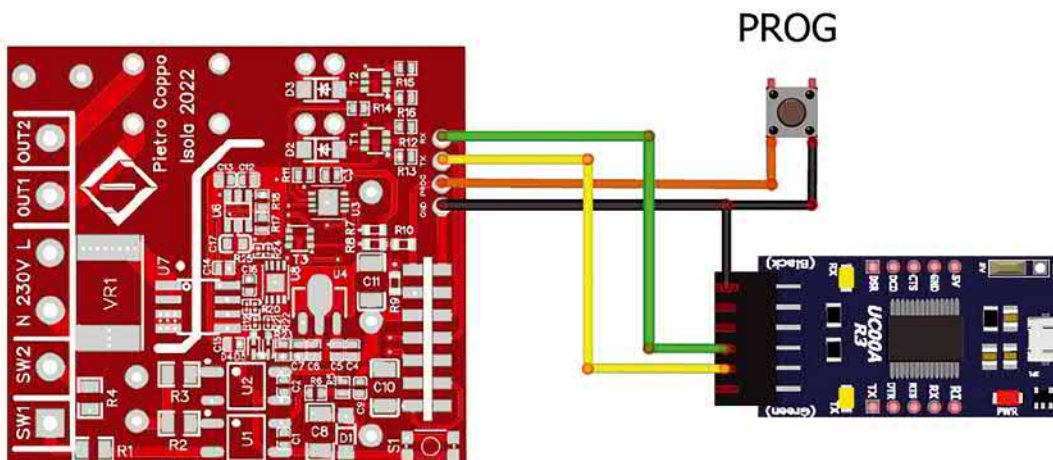


Fig. 3
Collegamento del programmatore seriale allo "Smart roller shutter".

una scossa che potrebbe essere anche mortale: la prudenza non è mai troppa.

PROGRAMMAZIONE E COLLAUDO

Poiché la configurazione e il collaudo dello "Smart roller shutter" è identica a quella degli altri dispositivi smart proposti in passato su questa rivista, per non essere ripetitivi, rimandiamo il lettore a leggere gli articoli già pubblicati, riportando in questo numero solo i passaggi più importanti.

La programmazione del modulo ESP03 deve essere effettuata mediante un adattatore USB per ESP8266 oppure con un qualsiasi convertitore USB TTL, purché compatibile con logica a 3,3 V, usato per programmare i moduli Arduino mini. Per il collegamento fate riferimento alla figura **Fig. 3**. Collegate l'uscita TX del convertitore all'entrata RX pin 4 del connettore CON2 e l'entrata RX del convertitore all'uscita TX pin 3 di CON2. Per mettere lo "Smart roller shutter" in modalità di programmazione, prima di allacciarlo alla corrente elettrica, premete il pulsante Prog collegato al pin 2 del connettore CON2 rilasciandolo dopo un secondo.

Per il corretto cablaggio del pulsante sul connettore CON2 fate riferimento al cablaggio di **Fig. 3**.

A scopo informativo precisiamo che lo sketch è stato compilato utilizzando il package core ESP8266 versione 2.7.4 e le librerie "lotwebconf" versione 3.2.1, "SinricPro" versione 2.10.0, "SparkFun_PCA9536_Arduino_Library" versione 1.2.2 e "ArduinoJson" versione 6.19.4. Fate attenzione che la versione del package core dell'ESP8266 in Arduino Ide sia aggiornato almeno alla versione 2.7.4, altrimenti avrete problemi di funzionamento con la libreria *lotWebConf*. Al momento della stesura dell'articolo la versio-

ne del package core dell'ESP8266 è arrivata alla versione 3.0.2. Con questa versione abbiamo riscontrato problemi con la rete WiFi, probabilmente con gli aggiornamenti futuri verrà risolto.

All'inizio dello sketch ci sono delle righe commentate che servono per impostare la compilazione del sorgente.

Ad esempio se la memoria Flash montata sul modulo ESP03 è marchiata *Puya*, e volete abilitare il supporto di tale memoria in fase di compilazione, dovete togliere il commento "//" davanti a "#define PUYA_SUPPORT 1", e se volete attivare il debug togliete il commento davanti a "#define ENABLE_DEBUG".

La libreria *lotwebconf*, per impostazione predefinita fa visualizzare sulla seriale le chiavi in chiaro del vostro account IoT. Per motivi di sicurezza vi consigliamo di disabilitare tale opzione aggiungendo all'inizio del file *lotWebConfSettings.h*, che si trova nella cartella *SRC*, la seguente riga di testo "#define IOTWEBCONF_DEBUG_DISABLED".

Tra i file scaricabili dal sito web www.elettronica.in.it, nella cartella STL, trovate i file per la stampa 3D degli adattatori e frutti per i supporti della serie Vimar Plana, Gewiss System, Gewiss Playbus e BTicino living, visibile in **Fig. 7**, che faciliteranno l'installazione dello "Smart roller shutter" nel vostro impianto elettrico domestico.

Fate attenzione al corretto orientamento del coperchietto di chiusura del contenitore, in modo che il forellino laterale sia posizionato sopra al pulsante SW1, per permettere di schiacciarlo in caso vogliate resettare il dispositivo se scordate la password.

Per quelli che vogliono realizzare da soli il contenitore con la stampante 3D, consigliamo di impostare l'altezza del layer a un valore non superiore di

Fig. 4
Home page
dello "Smart roller
shutter".

0,2 mm e, a seconda della versione del software di slicing, attivare l'utilizzo del riempimento a linee nel riempimento a griglia ad alta densità, altrimenti la parete del contenitore, spessa appena 1,5 mm, non verrà riempita correttamente presentando uno spazio vuoto.

Benché con il filamento PLA si ottengono ottimi risultati, consigliamo comunque l'utilizzo di filamento in ABS che sopporta temperature di fusione più alte, oppure quello ABS V0 autoestinguente certificato dallo standard UL94 V0.

In alternativa, per chi ha difficoltà a realizzare i contenitori, oppure non possiede una stampante

te 3D, può usufruire del servizio di stampa 3D messo a disposizione sul nostro sito all'indirizzo <https://furanet.it/servizio-di-stampa-3d/> e chiedere un preventivo. In caso di conferma le stampe 3D verranno spedite nelle 24 ore successive al termine della stampa.

UTILIZZO DELLO SMART ROLLER SHUTTER

Una volta acceso lo "Smart roller shutter" potete accedere alla pagina web iniziale, come visibile in **Fig. 4**, dove viene visualizzato il numero IP del dispositivo, il numero MAC (Media Access Control), la corrente assorbita dal carico espressa in ampere, il consumo espresso in watt e la posizione della tapparella espressa in percentuale. Per accedere alla pagina di configurazione vi verrà richiesto il nome utente e la password. Come nome utente inserite "admin" e come password "123456789". Una volta entrati nella sezione "Settings" potete impostare i parametri, come visibile in **Fig. 5**.

I settaggi sono uguali a quelli degli altri dispositivi IoT analizzati negli altri articoli già pubblicati. Per questo motivo andremo a elencare le varie impostazioni soffermandoci solo su quelle presenti esclusivamente in questo dispositivo.

Nella sezione "Shutter calibration" trovate il

Fig. 5
Schermata della
pagina web di
configurazione
dello "Smart roller
shutter".

pulsante per la calibrazione, che bisogna necessariamente avviare alla prima accensione, affinché il dispositivo acquisisca il tempo che il motore impiega per avvolgere completamente la tapparella. In poche parole deve apprendere quanto tempo intercorre dal finecorsa di apertura a quello di chiusura.

Per il posizionamento della tapparella, il nostro dispositivo, si affida al tempo trascorso e non a sensori di posizione da montare sull'asse del motore rendendo complicata l'installazione. Sicuramente il posizionamento calcolato, utilizzando il tempo come metodo di misura, non sarà accurato, ciononostante non pregiudica il corretto funzionamento del dispositivo.

Stiamo parlando pur sempre di una tapparella e non di una macchina CNC. Anche se usassimo dei sensori, avremmo comunque un errore di posizionamento causato dall'aumento progressivo del diametro di arrotolamento dato dallo spessore della tapparella che si avvolge sul rullo. L'errore ovviamente aumenta proporzionalmente alla lunghezza della tapparella e allo spessore delle sue doghe. Man mano che la tapparella si avvolge sul rullo il suo diametro aumenta, aumentando di conseguenza la velocità di avanzamento della

tapparella che non sarà mai lineare ma logaritmica. Difatti, una volta installato il nostro dispositivo, vi accorgete che al comando di posizionamento della tapparella a 50% questa non si posizionerà a metà corsa dell'apertura della finestra. Questo sarà più evidente su una porta finestra piuttosto che su una finestra dove l'altezza luce è quasi doppia. Per ovviare a questo inconveniente è sufficiente modificare il sorgente implementando un algoritmo che calcoli la percentuale di apertura prendendo in considerazione il diametro del rullo, lo spessore delle doghe e la lunghezza della tapparella.

Nella sezione "Roller shutter start position" la casella di spunta "Start at last position" vi permette di abilitare la funzione di ripristino dell'ultima posizione memorizzata. Dopo un riavvio, dovuto ad esempio a un blackout, il dispositivo, per risincronizzarsi con i fine corsa, prima chiuderà la tapparella e successivamente la riposiziona sull'ultima posizione impostata.

La casella di spunta "Start at custom position" vi permette di abilitare il posizionamento della tapparella su un valore in percentuale impostabile dal menu a tendina "Set custom position". Anche in questo caso dopo un riavvio il dispositivo si

Fig. 6
Schema di cablaggio dello "Smart roller shutter" con il pulsante interbloccato e il motore della tapparella.

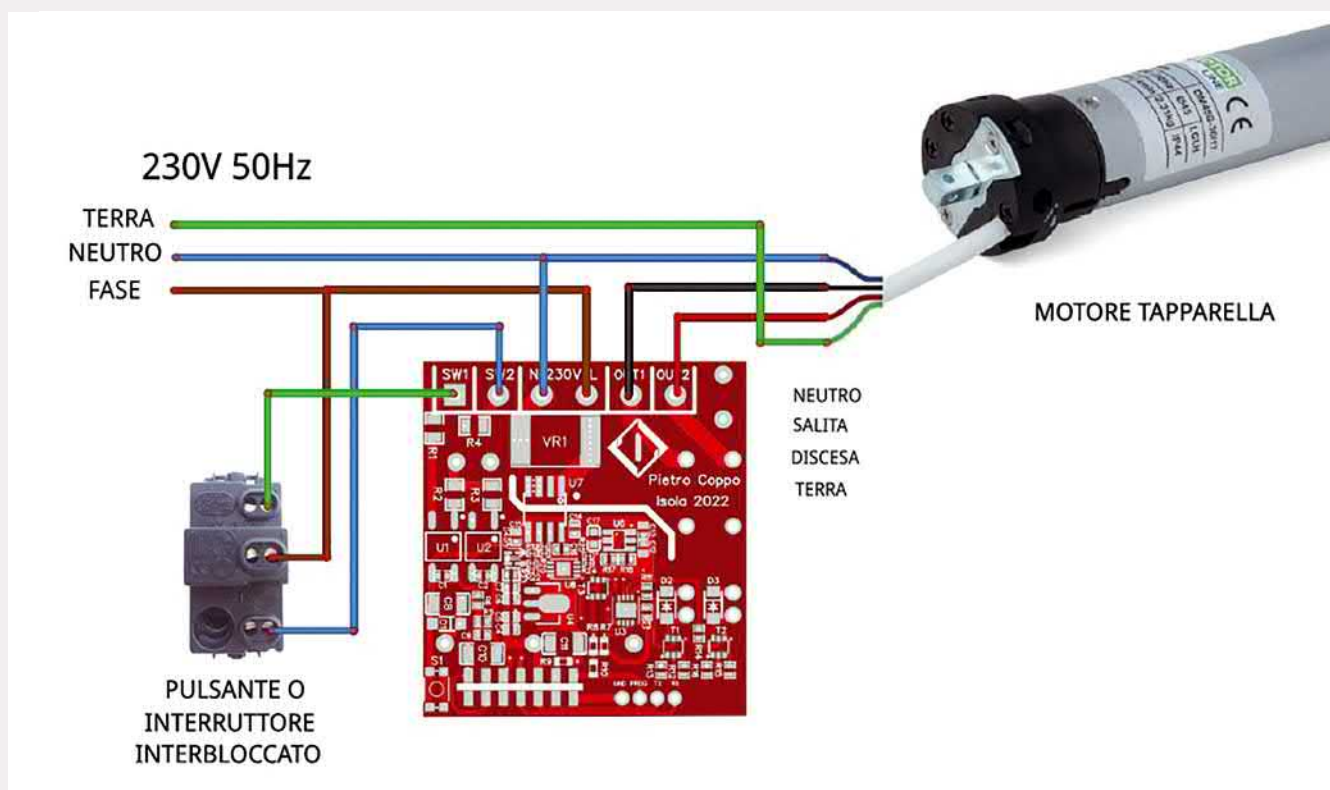


Fig. 7
Contenitori,
adattatori e frutti
stampati in 3D
per placche Vimar
Plana, Vimar Idea,
Gewiss Basic e
Gewiss PlayBus.



risincronizzerà chiudendo la tapparella e successivamente si posizionerà nella posizione impostata. Se non si seleziona nessuna di queste funzioni all'avvio la tapparella rimarrà ferma. In questo caso la sincronizzazione dovrà essere effettuata portandola, con un comando vocale o con il pulsante, in una delle due posizioni di finecorsa aperto o chiuso. Facciamo presente che il dispositivo, per minimizzare gli errori, si risincronizza ogni volta raggiunto uno dei finecorsa.

La casella di spunta *"Enable external pushbutton"* vi permette di abilitare o disabilitare la regolazione mediante pulsanti o interruttori esterni. La casella di spunta *"Enable assistant control"* vi permette di abilitare o disabilitare il controllo mediante assistente vocale Amazon Alexa o Google assistance. La casella di spunta *"Reboot device on save"*, abilita il riavvio del dispositivo dopo il salvataggio delle impostazioni schiacciando il tasto *"Apply"*.

Una volta configurato il dispositivo vi consigliamo di impostare lo *"Startup delay"* presente nella sezione *"System configuration"* a 1 secondo, così

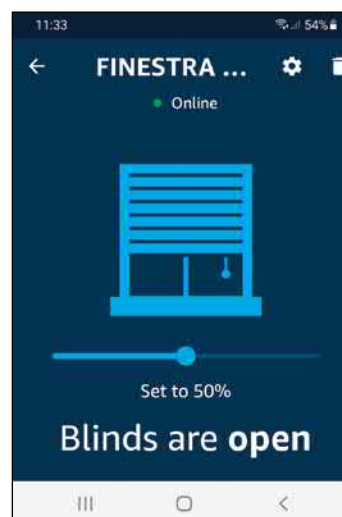
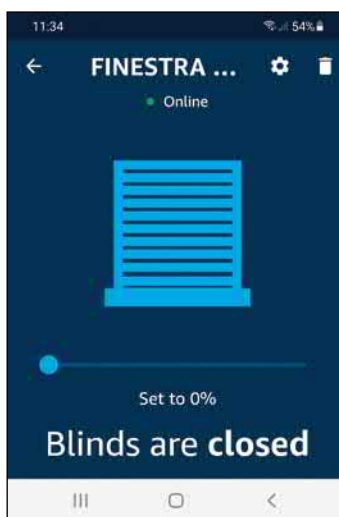
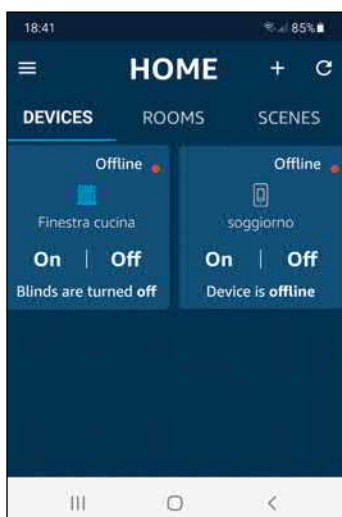
se lo *"Smart roller shutter"* si riavvia, ad esempio a causa di un calo momentaneo di tensione sulla linea elettrica, invece di aspettare i preimpostati 30 secondi di delay, si collegherà alla rete WiFi immediatamente e sarà attivo più in fretta. Il tempo di 30 secondi serve esclusivamente al primo avvio in fase di configurazione, quando il dispositivo diventa un punto di accesso (access point) per permettervi di individuarlo e selezionare la sua SSID. In **Fig. 6** trovate lo schema esemplificativo di collegamento dello *"Smart roller shutter"*.

Se sul vostro smartphone installate l'applicazione Google Home, lo *"Smart roller shutter"* potrà essere comandato anche con l'assistente vocale Google Assistant.

Di seguito riportiamo alcuni comandi riconosciuti dall'assistente vocale Amazon Alexa. Supponiamo che in <https://sinric.pro> al nostro dispositivo abbiate assegnato il nome *"Finestra cucina"*.

Per alzare tutta la tapparella pronunciate: *"Alexa, apri finestra cucina"* oppure *"Hey Google, apri finestra cucina"*. Per abbassare tutta la tapparella

Fig. 8
Interfaccia dell'app
Sinric pro.



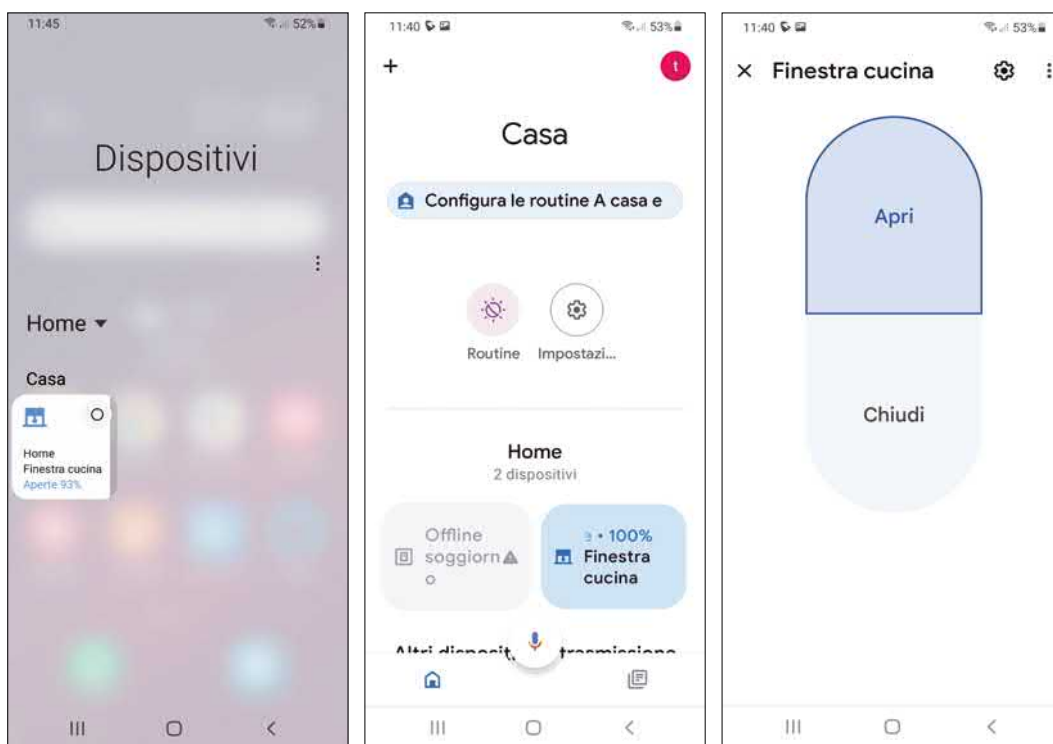


Fig. 9
Interfaccia
dell'app
Google
Home.

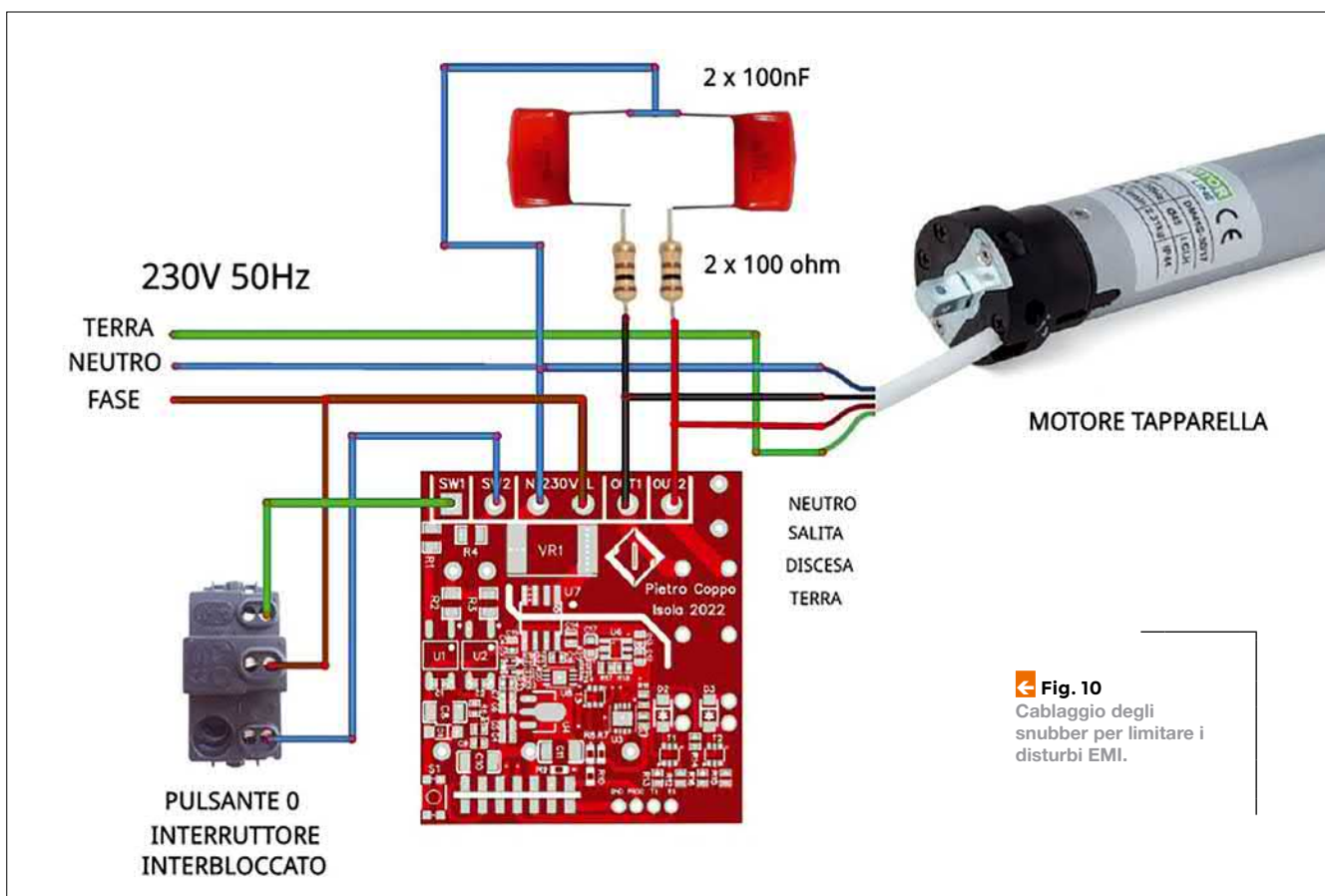
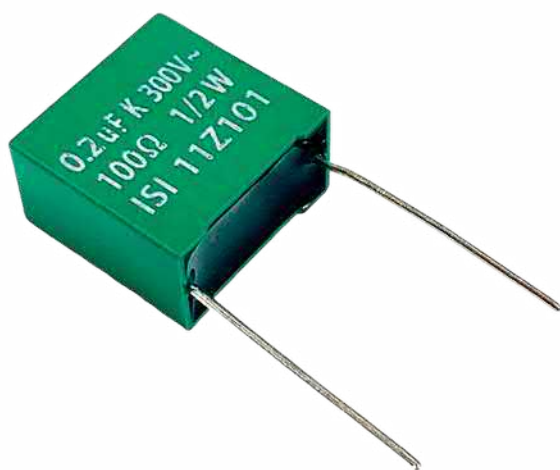


Fig. 10
Cablaggio degli
snubber per limitare i
disturbi EMI.

Fig. 11
Snubber integrato
reperibile in com-
mercio.



pronunciate: "Alexa, chiudi finestra cucina" oppure "Hey Google, chiudi finestra cucina".
Per posizionare la tapparella a metà corsa: "Alexa, imposta finestra cucina a 50%" oppure "Hey Google, imposta finestra cucina a 50%".
Per aumentare l'apertura di 10%: "Alexa, incrementa di 10% finestra cucina" oppure "Hey Google, incrementa di 10% finestra cucina". Per diminuire l'apertura di 10%: "Alexa, decrementa di 10% finestra cucina" oppure "Hey Google, decrementa di 10% finestra cucina".

In **Fig. 8** potete osservare le immagini dell'interfaccia dell'app di Sinric pro, mentre in **Fig. 9** potete osservare le immagini dell'app Google home.
Prima di terminare vogliamo informarvi che a causa dello spazio ridotto non siamo riusciti a implementare gli snubber antidisturbo sullo stampato. Se il funzionamento del motore della tapparella disturba altri dispositivi o apparecchi domestici, oppure volete proteggere i contatti del relé dal deterioramento causato dalle correnti indotte, vi consigliamo di inserire in parallelo agli avvolgimenti del motore uno snubber formato da un condensatore e una resistenza da 1-2 W, come visibile in **Fig. 10**.
Va specificato che il condensatore deve essere di classe X2 possibilmente MKP, mentre la resistenza meglio se a strato metallico e assolutamente non a filo. In alternativa in commercio potete trovare snubber già pronti e integrati, con componenti di diversi valori e caratteristiche specifiche (un esempio è visibile in **Fig. 11**) il cui contenitore assomiglia a un comune condensatore rettangolare a film di polipropilene o poliestere.

Lo potete collegare vicino al motoriduttore nel cassetto della tapparella oppure, se avete spazio, nella scatola di derivazione che ne realizza le connessioni.

SOPPRIMERE LE EXTRATENSIONI

Quando l'interruttore si apre, la carica immagazzinata nell'induttanza del carico provoca un'extratensione dovuta al fatto che la natura inerziale degli induttori nei riguardi della corrente tende a far permanere il regime dell'istante precedente; è possibile calcolare l'extratensione generata utilizzando la seguente formula:

$$V = \frac{L \times I}{t}$$

dove V è la tensione da calcolare, L è il valore dell'induttanza della bobina, I è la corrente nominale della bobina e t corrisponde al tempo di rilascio dell'alimentazione.

Tale tensione si presenta tra i contatti dell'interruttore, il quale nel momento dell'apertura presenta una distanza tra i contatti minima.

Pertanto, la formazione di un arco tra i contatti può verificarsi quasi immediatamente cioè nell'istante in cui questi si stanno separando e non si toccano più. Il fenomeno può verificarsi anche con carichi resistivi, ma è accentuato in quelli induttivi, i quali determinano un arco elettrico intenso che riduce la vita dell'interruttore.

Anche se i contatti sono rivestiti o realizzati in Tungsteno (detto anche Wolframio) ossia un metallo che fonde ad altissime temperature ($3.4 \cdot 10^3$ °C) con il tempo comunque possono erodersi.

Nei circuiti a corrente continua per eliminare la corrente indotta ad alta tensione di solito viene utilizzato semplicemente un diodo chiamato diodo di flyback. Sfortunatamente, l'applicazione di questo diodo nei circuiti a corrente alternata non è possibile. Quando abbiamo a che fare con la corrente alternata, abbiamo tre possibili soluzioni: possiamo usare un varistore di ossido di metallo (metal oxide varistor - MOV), un diodo soppressore di tensione transitoria bidirezionale (transient voltage suppression - TVS) oppure una rete di soppressione RC, nota anche come snubber.

Oltre a queste soluzioni esiste anche quella di non utilizzare alcun soppressore, però in questo caso la vita dei contatti ne risentirà.

I diodi MOV e TVS conducono corrente unicamente quando viene superata la tensione di soglia.

Normalmente questi diodi sono collegati in parallelo al contatto dell'interruttore e sono in grado di funzionare in modo efficiente sia a basse che alte tensioni. Lo snubber RC invece ha il vantaggio di limitare la tensione esattamente durante l'aper-

tura dell'interruttore cioè quando la distanza tra i contatti è ridotta. È realizzato da un condensatore e da una resistenza collegate tra loro in serie e possono essere montati in parallelo al contatto dell'interruttore oppure in parallelo al carico, come visibile in **Fig. 12**.

Anche se il montaggio dello snubber in parallelo al contatto è la configurazione ideale, c'è un enorme svantaggio quando questi sono aperti, poiché provoca un passaggio di corrente verso il carico.

Al contrario se lo snubber è installato in parallelo al carico l'impedenza di quest'ultimo può influenzare sull'efficienza del circuito RC.

Quindi quando lo snubber è in parallelo ai contatti dell'interruttore, la resistenza deve avere un valore sufficientemente alto da limitare la corrente di scarica del condensatore quando i contatti dell'interruttore si chiudono.

Allo stesso tempo, deve avere un valore abbastanza piccolo da limitare la tensione indotta quando i contatti dell'interruttore si aprono.

Se scegliamo un condensatore con valore elevato, sicuramente ridurrà meglio le extra tensioni quando i contatti dell'interruttore si aprono; ma allo stesso tempo un condensatore di questo tipo può essere più costoso, più ingombrante e potrebbe causare danni maggiori durante la sua scarica sui contatti quando questi si chiudono.

Quando invece lo snubber è in parallelo al carico, per diminuire il degrado dei contatti dovuto all'arco, dobbiamo assicurarci che il valore della resistenza sia minimo. Allo stesso tempo però, per ridurre l'arco sul contatto, dovuto alla corrente di spunto del condensatore, il valore di quest'ultima deve essere maggiore.

Come avete potuto constatare molti sono i fattori che determinano la scelta del valore di R e di C . Poiché lo snubber, nel nostro caso, deve proteggere i contatti di un interruttore, e non un MOSFET in uno switching ad alta frequenza, la scelta del valore RC non sarà critica. In linea di massima potremmo scegliere i valori della resistenza R tra 10 ohm a 220 ohm e per il condensatore C tra 100 nF a 330 nF.

Per motivi di reperibilità di solito si scelgono per R un valore di 100 ohm e per C un valore di 100nF. Oltre a questo metodo empirico e poco scientifico vediamo assieme come si dovrebbero scegliere i valori di questi componenti.

PROGETTAZIONE DELLO SNUBBER

Di solito per trovare i valori RC si collega in parallelo al carico un oscilloscopio e si misura la forma

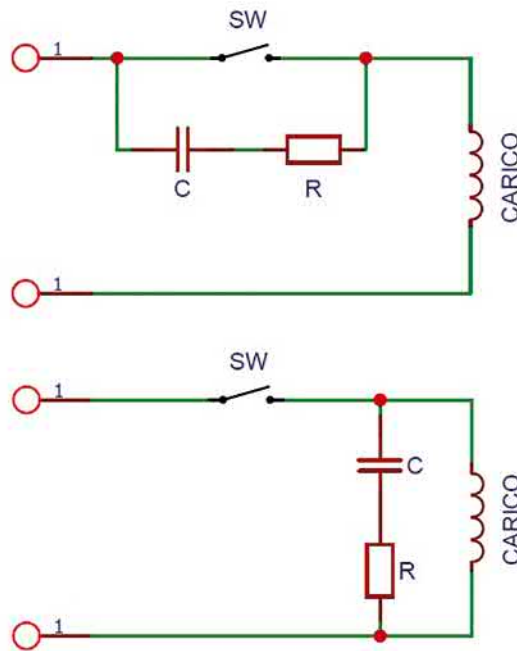


Fig. 12
Cablaggio dello Snubber parallelo all'interruttore (in alto) e parallelo al carico (in basso).

d'onda della tensione indotta generata quando viene spento un carico induttivo, chiamata frequenza di risonanza, che indichiamo con $F_{RING1'}$. In parallelo al carico colleghiamo un condensatore di valore basso che deve essere gradualmente incrementato fino a riuscire a dimezzare l'ampiezza della frequenza di risonanza iniziale.

Questa frequenza dimezzata la chiamiamo $F_{RING2'}$. Il valore di questo condensatore sarà il condensatore C_s del nostro snubber.

Stimiamo l'induttanza parassita L_p , utilizzando la frequenza di risonanza iniziale $F_{ring1'}$ senza condensatore, e quella con l'aggiunta del condensatore $F_{ring2'}$ utilizzando l'equazione:

$$L_p = \frac{1}{(2\pi(F_{RING1} - F_{RING1}))^2 \times C_s} = \frac{1}{4\pi^2(F_{RING1} - F_{RING2})^2 \times C_s}$$

Stimiamo la capacità parassita C_p utilizzando l'equazione:

$$C_p = \frac{1}{(2\pi F_{RING1})^2 \times L_p} = \frac{1}{4\pi^2(F_{RING1})^2 \times L_p}$$

Calcolata l'induttanza parassita L_p e la capacità parassita C_p calcoliamo il valore della resistenza dello snubber R_s utilizzando l'equazione:

$$R_s \geq Z = \sqrt{\frac{L_p}{C_p}}$$

Per attenuare la frequenza di risonanza è necessario scegliere per lo snubber una resistenza R_s uguale o maggiore dell'impedenza Z , mentre per C_s scegliere un valore da 1 a 4 volte maggiore di quello della capacità parassita C_p . Questo procedimento richiede anche degli accorgimenti non indifferenti quando si fanno misurazioni con l'oscilloscopio sulla linea elettrica a 230 Vca. Bisogna utilizzare un oscilloscopio particolare con ingressi galvanicamente isolati o alimentato a batterie, utilizzare particolari sonde di tipo differenziale o sonde isolate galvanicamente, in alternativa utilizzare un trasformatore di isolamento con rapporto di trasformazione 1:1, con potenza adeguata, capace di alimentare il circuito sul quale vogliamo fare le misurazioni.

Esiste un ulteriore metodo, che non dovremmo neanche menzionare poiché pericoloso, ed è quello di togliere il collegamento a terra dell'oscilloscopio e fare le misurazioni collegando sempre il coccodrillo della sonda sul neutro della linea elettrica (non la fase, perché ci si ritroverebbe la fase sulla carcassa dell'oscilloscopio e se la tocchiamo sono guai). Noi consigliamo l'utilizzo del trasformatore, che tra l'altro è quello più sicuro e che costa meno. Un metodo alternativo è quello di utilizzare l'equazione sviluppata da C.C. Bates e pubblicata nell'articolo "Contact protection of electromagnetic relays" sulla rivista "Electro-mechanical design" nel lontano agosto del 1966.

$$C = \frac{I^2}{10}$$

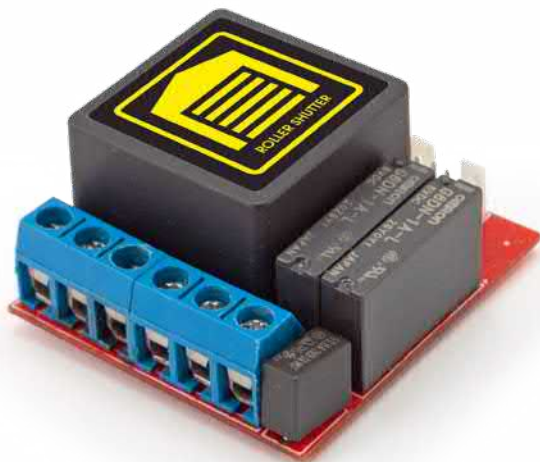


Fig. 13
Le morsettiere
connettono il
motore della tappa-
rella e i pulsanti.

$$R_s = \frac{V_p}{10 \times I \times \left(1 + \left(\frac{50}{V_p}\right)\right)}$$

- V_p = Tensione picco
- I = Corrente di carico (prima dell'apertura del contatto)
- R_s = Resistenza dello snubber RC espresso in μF
- C = Capacità dello snubber RC
- R_L = Resistenza di carico

Per capire come utilizzare l'equazione, facciamo un esempio di calcolo per la progettazione di uno snubber RC per un motore elettrico con potenza di 300 W, funzionante con una tensione alternata di 230 V. I parametri di progetto su cui ci basiamo sono $P = 300$ W, $V_{rms} = 230$ V, $V_p = 325,27$ V. Iniziamo con il calcolo della tensione di picco:

$$V_p = V_{rms} \times \sqrt{2}$$

$$V_p = 230 \text{ V} \times 1,414213562373 = \mathbf{325,27 \text{ V}}$$

Andiamo a determinare la tensione picco-picco:

$$V_{pp} = V_{rms} \times 2 \times \sqrt{2}$$

$$V_{pp} = 230 \text{ V} \times 2 \times 1,414213562373 = \mathbf{650,54 \text{ V}}$$

Proseguiamo con il calcolo della tensione efficace:

$$V_{rms} = \frac{V_p}{\sqrt{2}} = \frac{325,27 \text{ V}}{1,414213562373} = \mathbf{230 \text{ V}}$$

Passiamo all'equazione per calcolare la corrente assorbita dal carico:

$$P = I \times V_p \Rightarrow I = \frac{P}{V_p}$$

$$I = \frac{300 \text{ W}}{325,27 \text{ V}} = \mathbf{0,922 \text{ A}}$$

Ora vediamo l'equazione per calcolare il valore del condensatore della rete snubber:

$$C = \frac{I^2}{10} = \frac{0,922 \text{ A}^2}{10}$$

$$C = \mathbf{85 \text{ nF}}$$

Invece per determinare il valore della resistenza della rete snubber applichiamo la seguente equa-

zione svolgendone i calcoli:

$$R_s = \frac{V_p}{10 \times I \times \left(1 + \left(\frac{50}{V_p}\right)\right)}$$

$$R_s = \frac{325,27}{10 \times 0,922 \times \left(1 + \left(\frac{50}{325,27}\right)\right)}$$

$$R_s = \frac{325,27}{10 \times 0,922 \times (1 + 0,1537)}$$

$$R_s = \frac{325,27}{9,22 \text{ A} \times 1,1537}$$

$$R_s = \frac{325,27}{10,637}$$

$$R_s = 30,579 \Omega \cong \mathbf{30,58 \Omega}$$

Vediamo adesso come calcolare la corrente che passa attraverso lo snubber RC: per prima cosa dobbiamo calcolare la reattanza capacitiva (X_C) del condensatore appartenente alla rete (la frequenza di rete elettrica è 50 Hz) rammentando che il termine π vale 3,1415926:

$$X_C = \frac{1}{2\pi \times f \times C}$$

$$f = 50 \text{ Hz}$$

$$C = 0,085 \mu\text{F} \Rightarrow 0,000000085 \text{ F}$$

$$X_C = \frac{1}{2\pi \times 50 \text{ Hz} \times 0,000000085 \text{ F}}$$

$$X_C = \frac{1}{2 \times 3,14159 \times 50 \text{ Hz} \times 0,000000085 \text{ F}}$$

$$X_C = \frac{1}{0,0000267 \text{ F}} = 37313,432 \Omega$$

$$X_C = 37313,43 \Omega$$

A questo punto scriviamo l'equazione per calcolare la corrente che passa attraverso la rete snubber:

$$I = \frac{V}{R} = \frac{V_p}{(X_C + R_s)}$$

$$I = \frac{325,27 \text{ V}}{(37313,43 \Omega + 30,58 \Omega)}$$

$$I = \frac{325,27 \text{ V}}{37344 \Omega}$$

$$I = 0,00871 \text{ A} \Rightarrow \mathbf{8,71 \text{ mA}}$$

Infine utilizziamo l'equazione occorrente per calcolare la potenza dissipata dalla resistenza dello snubber RC, la quale, essendo un componente dissipativo, va dimensionata opportunamente, onde

evitare che surriscaldi e bruci:

$$P = V \times I$$

$$V = R \times I$$

$$P = V \times I = (R \times I) \times I = R \times I^2$$

$$I = 0,00871 \text{ A}$$

$$P_R = I^2 \times R_s$$

$$P_R = 0,00871 \text{ A}^2 \times 30,58 \Omega$$

$$P_R = 0,0023 \text{ W} \Rightarrow \mathbf{2,319 \text{ mW}}$$

CONCLUSIONI

Con questo dispositivo speriamo di aver soddisfatto i nostri lettori che volevano realizzare un interruttore smart per tapparella facilmente gestibile e comandabile da remoto con lo smartphone o con l'assistente vocale, nonché essere di dimensioni contenute per facilitare l'installazione negli impianti domestici. Ricordiamo che il progetto "Smart roller shutter" fa parte di una serie di moduli e attuatori domotici progettati e realizzati dagli alunni dell'Istituto Tecnico Pietro Coppi Isola Slovenia, alcuni dei quali già pubblicati sulla nostra rivista. Rientrano tutti nel concetto di IoT e dei dispositivi connessi, nello specifico rivolti all'automazione degli ambienti domestici tramite il collegamento WiFi e l'integrazione con app e assistenti vocali, oggi molto in voga. ■



Cosa occorre?

I componenti principali utilizzati in questo progetto sono disponibili presso Futura Elettronica. Il modulo ESP8266 (cod. ESP03) è in vendita a Euro 8,00, il programmatore/debugger (cod. YM203) è disponibile a Euro 2,50, il convertitore USB/TTL (cod. UC00A) costa Euro 15,50. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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



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

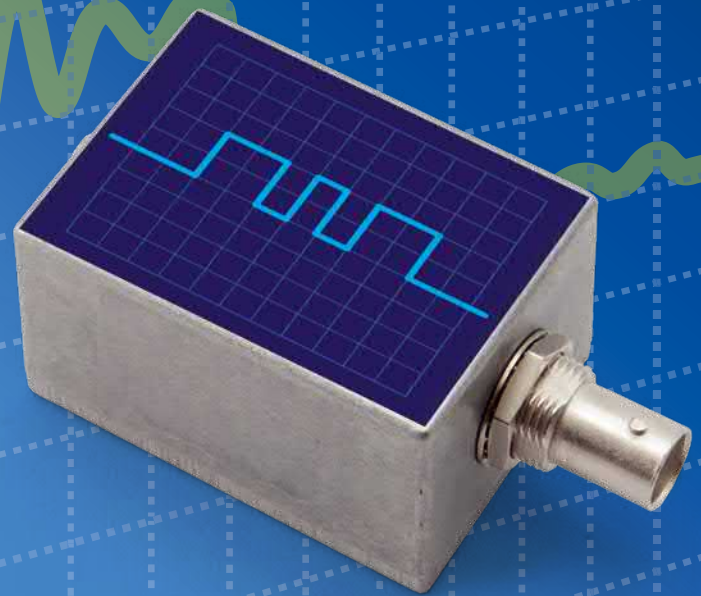
Particolarmente utile (ad esempio prima di praticare un foro su una parete) per rilevare la presenza di cavi elettrici sotto tensione, strutture in legno (telai, travetti, barre, ecc), oggetti metallici (travi, telai, tubi, tondini, perni, barre in metallo).



Prezzo IVA inclusa.

RIFLETTOMETRIA NEL DOMINIO DEL TEMPO

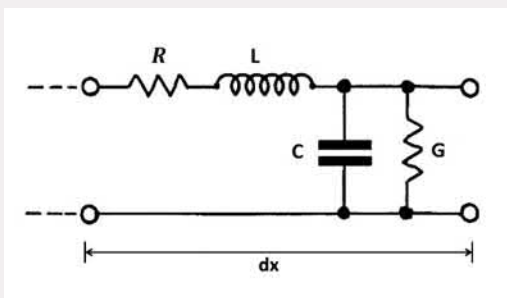
di VALERIO CONICELLA



Misuriamo l'impedenza caratteristica di un cavo coassiale e rileviamo le condizioni di adattamento, mediante un semplice hardware ed un oscilloscopio.

La Riflettometria nel Dominio del Tempo (TDR) è una tecnica di misura basata sull'analisi del segnale viaggiante lungo una linea di trasmissione e riflesso da un generico carico. Il principio di funzionamento è analogo a quello su cui si basa il radar di un aereo o il sonar di un sommergibile per localizzare un oggetto; in particolare, è possibile identificare con molta precisione la distanza, la velocità, la direzione e il verso di un oggetto. La Riflettometria nel Dominio del Tempo (TDR) viene utilizzata anche in ambito scientifico e industriale, per effettuare controlli quantitativi e qualitativi, ed interessa diverse aree di applicazione, come ad esempio il monitoraggio di sostanze all'interno di cisterne, il monitoraggio ambientale e anche la

Fig. 1
Modello generale
della linea
di trasmissione.



caratterizzazione delle linee di trasmissione. In questo articolo ci interesseremo di quest'ultima e nello specifico spiegheremo come misurare l'impedenza caratteristica di un cavo coassiale e la procedura che permette di determinare potenziali problemi di integrità del cavo stesso.

LINEE DI TRASMISSIONE

Un sistema di misura basato sulla TDR genera un gradino di tensione che viene inviato, attraverso un cavo coassiale di impedenza nota, al carico o alla linea da analizzare. L'analisi del segnale riflesso permette di dedurre la natura del carico (resistiva, induttiva, capacitiva) o la lunghezza della linea, ed eventualmente di localizzare con precisione eventuali discontinuità. Prima di procedere con la descrizione del setup di misura è importante soffermarsi sulla teoria delle linee di trasmissione. La necessità di introdurre le linee di trasmissione come formalismo matematico, dipende dal fatto che i circuiti a costanti concentrate basati sui bipoli elementari (resistenza, induttanza, capacità, ecc.) possono essere utilizzati solo quando le dimensioni dei circuiti sono molto piccole rispetto alla lunghezza d'onda (λ) del segnale che li interessa. Facciamo un esempio: alla frequenza di 50 Hz la lunghezza d'onda vale $\lambda = 6.000$ km, mentre alle microonde le dimensioni dei dispositivi sono comparabili con λ , infatti, alla frequenza di 1 GHz la lunghezza d'onda vale $\lambda = 30$ cm, mentre a 10 GHz vale $\lambda = 3$ cm. A tali frequenze le dimensioni dei dispositivi risultano comparabili con la lunghezza d'onda e la teoria dei circuiti a costanti concentrate non vale più. Tuttavia, considerando una sezione della linea di lunghezza infinitesima dx , la dimensione di tale sezione è molto piccola rispetto a λ e la teoria dei circuiti a costanti concentrate si può applicare. Analizzeremo, dunque, lo studio della trasmissione di un segnale attraverso una coppia di conduttori, che prende il nome di linea di trasmissione. Questa coppia può essere composta da due fili paralleli tra loro (doppino) e da

due conduttori coassiali separati da un dielettrico (cavo coassiale). Alla linea di trasmissione reale si approssima il circuito equivalente proposto dalla **Fig. 1**, che sostanzialmente consiste in una rete RC. In particolare, i termini stanno ad indicare:

- **R**: resistenza elettrica che tiene conto delle perdite di tensione (effetto Joule) lungo il tratto;
- **L**: induttanza in serie che tiene conto del campo magnetico concatenato;
- **C**: capacità che tiene conto del campo elettrico trasversale fra i due conduttori;
- **G**: resistenza elettrica trasversale dovuta al non perfetto isolamento fra i conduttori.

In generale, la condizione affinché un conduttore si comporti come una linea è data dalla relazione:

$$t_{r,f} < 2t_p$$

dove t_r è il tempo di salita o discesa dei segnali, t_p è il ritardo di propagazione lungo il conduttore e $2t_p$ il ritardo di propagazione di andata e ritorno, che dipende dalla lunghezza della linea.

In teoria, se la linea fosse infinita, il segnale si propaga fino a quando le perdite non la esauriscono progressivamente. Nella realtà, date le dimensioni finite della linea, il segnale incontra ad una certa distanza l'impedenza di carico Z_L . Quest'ultima, per poter assorbire tutta l'energia ricevuta, deve necessariamente avere l'impedenza caratteristica Z_0 della linea. Utilizzando il metodo dei fasori, è possibile dimostrare che la tensione e la corrente transanti lungo la linea sono espresse dalle seguenti relazioni:

$$V(x, \omega) = V_i e^{-\gamma x} + V_r e^{\gamma x}$$

$$I(x, \omega) = \frac{V_i e^{-\gamma x} - V_r e^{\gamma x}}{Z_0}$$

Z_0 è l'impedenza caratteristica della linea e rappresenta l'impedenza che la linea avrebbe se fosse infinitamente lunga. In altre parole, Z_0 è il valore dell'impedenza di carico che deve essere posto alla fine della linea di trasmissione, affinché non si abbiano riflessioni. Essa vale:

$$Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}}$$

Altri parametri importanti mediante i quali si può valutare l'entità del disadattamento del cavo sono:

- il Coefficiente di riflessione, espresso dalla for-

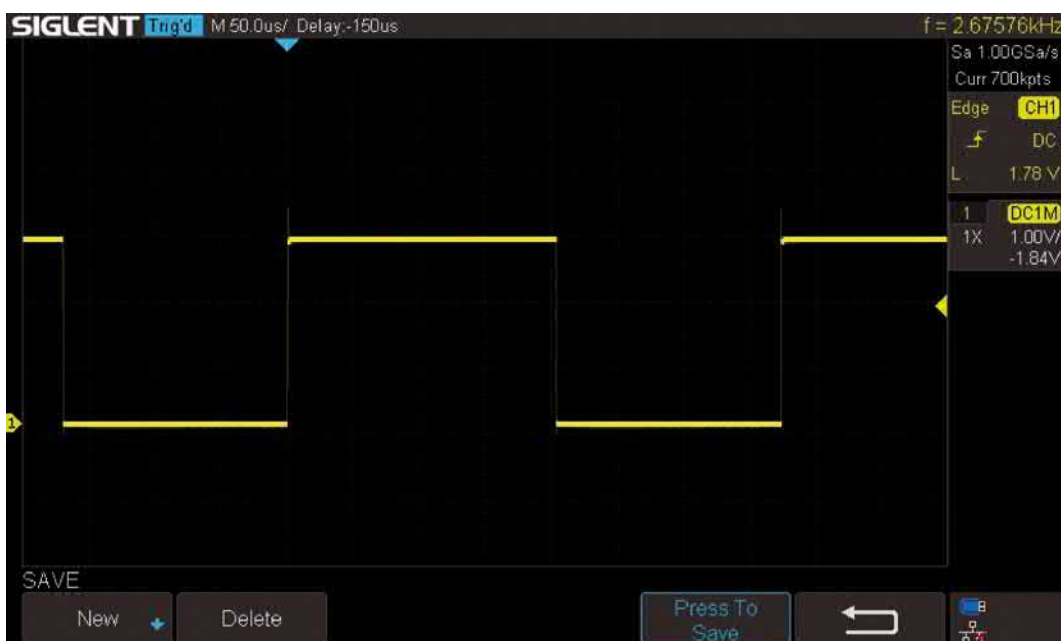


Fig. 2
Oscillogramma del segnale generato

mula qui sotto (dove Z_L è l'impedenza del carico):

$$\rho_v = \frac{Z_L - Z_o}{Z_L + Z_o}$$

- il rapporto d'onda stazionaria (**ROS**), definito come:

$$ROS = \frac{V_{max}}{V_{min}} = \frac{1 + |\rho_v|}{1 - |\rho_v|}$$

Se il carico Z_L della linea è uguale a Z_o , la linea è adattata: ciò vuol dire che il suo coefficiente di riflessione è uguale a zero e non avverrà, quindi, nessuna riflessione; la linea sarà equivalente a una linea infinitamente lunga.

GENERATORE ONDA QUADRA

Per poter effettuare le misure che vi proponiamo in queste pagine è necessario un generatore di funzioni, altrimenti detto "impulsatore".

Chi è del settore, sa benissimo che il costo di un buon generatore di funzioni da laboratorio può raggiungere cifre importanti, tuttavia, accontentandosi delle prestazioni, è possibile realizzare un semplice ed economico generatore ad onda quadra, con fronti di salita dell'ordine del nanosecondo, senza ricorrere a tecniche particolari.

Il circuito qui presentato e descritto dallo schema elettrico ne è un esempio e fornisce un segnale come quello proposto dalla **Fig. 2**, che propone la

forma d'onda rilevata con l'oscilloscopio digitale sul prototipo che abbiamo realizzato (**Fig. 3**).

Come si può osservare, il cuore del circuito è l'integrato 74AC14, contenente nel proprio package sei porte logiche inverter (NOT) di tipo Schmitt - Trigger. Questo integrato fa parte della famiglia Advanced CMOS, che è caratterizzata da tempi di attraversamento dell'ordine dei nanosecondi.

La sezione D realizza un oscillatore ad onda quadra con frequenza fissa di circa 2,2 kHz.

Per variare la frequenza generata, è necessario modificare il valore delle resistenze R1, oppure la capacità del condensatore C1, utilizzando la

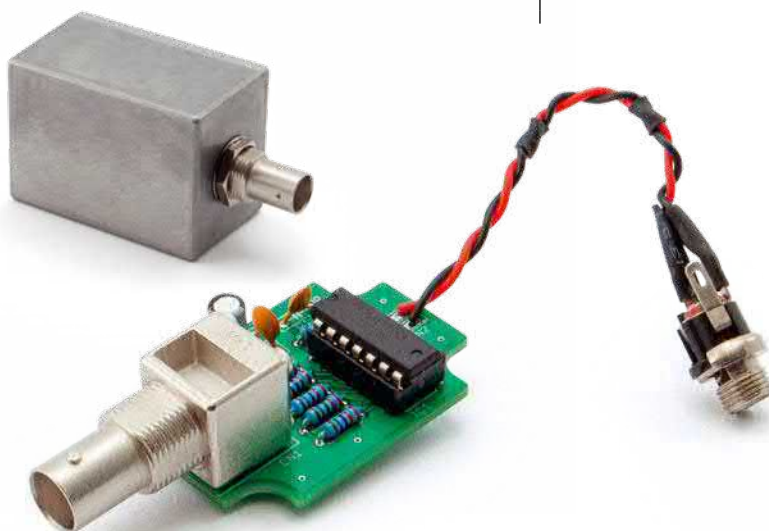
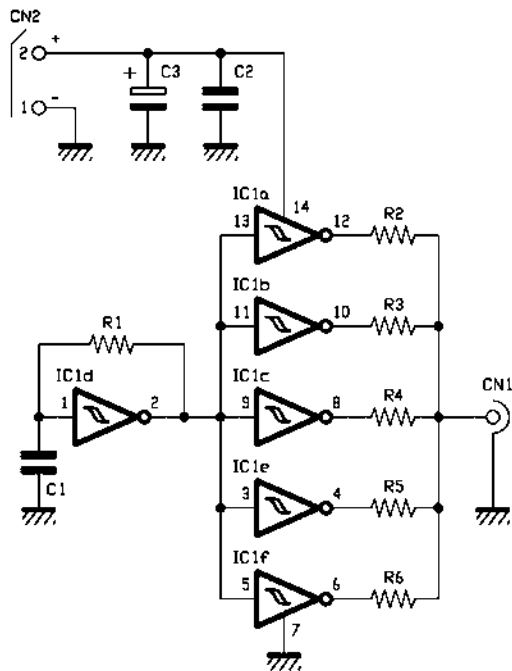
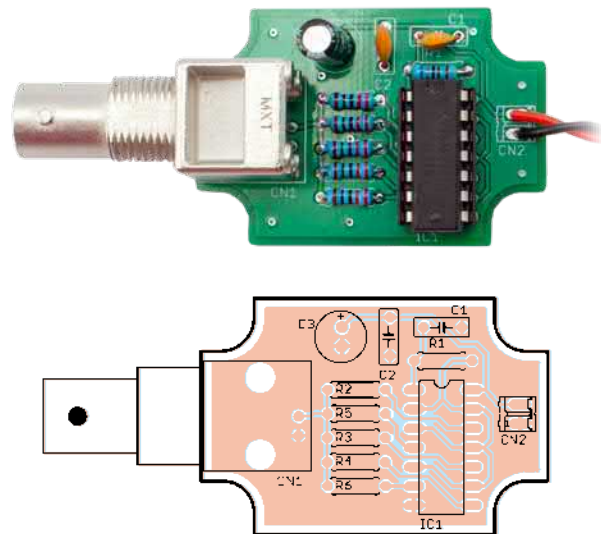


Fig. 3
Il nostro prototipo di generatore di impulsi.



Elenco Componenti:

R1: 6,8 kohm 1%
 R2, R3, R4, R5, R6: 220 ohm 1%
 C1: 47 nF ceramico
 C2: 10 nF ceramico
 C3: 10 µF 100 VL elettrolitico
 IC1: SN74AC14N



Varie:

- Zoccolo 7+7
- Presa BNC da CS
- Plug DC da pannello
- Circuito stampato S1666 (46x31 mm)

seguente formula:

$$f = \frac{0,7}{R_1 C_1}$$

Il valore della frequenza generata è sempre molto approssimativo, perché oltre alla tolleranza delle resistenze e del condensatore, c'è anche quella dell'integrato utilizzato, che può variare il valore delle soglie a seconda della casa costruttrice. Le porte A, B, C, E, F realizzano un buffer di uscita adattato a 50 ohm, attraverso le resistenze collegate in serie alle uscite degli inverter. Le principali caratteristiche elettriche del generatore sono:

- impedenza di uscita circa $Z_o = 50 \text{ ohm}$;
- segnale di uscita rettangolare di impulsi di ampiezza $V_{pp} = 5 \text{ V}$;
- periodo del segnale $T = 370 \text{ µs}$;
- durata impulso $T_{on} = 200 \text{ µs}$ (cui corrisponde un duty-cycle del 54%)
- tempo di salita di 4 ns.

IL CAVO COASSIALE

Il cavo coassiale (**Fig. 4**) è una linea di trasmissione schermata, costituita da un conduttore cilindrico di rame a sezione circolare (anima) e da uno schermo conduttore esterno, costituito da una maglia metallica. Il conduttore interno è separato da quello esterno da un dielettrico a bassa perdita, che garantisce l'isolamento tra i due conduttori. Lo schermo di metallo esterno aiuta a bloccare le interferenze. Questa tipologia di cavi è largamente usata nelle telecomunicazioni, per esempio per il trasporto del segnale audiovisivo. Sul mercato esistono diverse tipologie di cavi coassiali, prodotti in funzione della potenza e della frequenza da trasportare. I valori di impedenza principali sono due:

- 50 ohm, impiegato per le trasmissioni digitali o radioamatoriali;
- 75 ohm, utilizzato per segnale video analogico, per le antenne televisive e per le connessioni Internet via cavo.

L'impedenza caratteristica di 50 ohm è maggiormente impiegata per gli apparati trasmettenti, viceversa, il cavo con impedenza caratteristica di 75 ohm è preferito solo per gli apparati riceventi in quanto, a parità di diametro esterno, ha un'attenuazione minore di quella di un cavo a 50 ohm. In particolare, per il nostro esperimento utilizzeremo il cavo **RG-58**, impiegato in larga misura per trasferire segnali elettrici con potenze che possono raggiungere qualche centinaio di watt ed è capace di lavorare fino a frequenze di 2 GHz. Riportiamo le caratteristiche elettriche a 20 °C fornite dal costruttore (datasheet):

- Impedenza caratteristica Z_0 (Ω): (50 ± 2) ohm;
- Capacità (**C**): 100 pF/m;
- Velocità di propagazione (V_p): 66% che corrispondono a circa $1,98 \cdot 10^8$ m/s;
- Attenuazione (A_{dB}): 4,1 dB/100m @10MHz;
- Lunghezza cavo: 3 m.

MISURA IMPEDENZA CARATTERISTICA

Come accennato, la riflettometria nel dominio del tempo si basa sul calcolo dell'impedenza lungo una linea, analizzandone le varie riflessioni. In pratica, si osservano attraverso l'utilizzo di un oscilloscopio i disadattamenti di impedenza in base ai segnali riflessi. Per fare questo, occorre un generatore con tempo di salita molto breve, se possibile molto più breve del tempo di attraversamento della linea da esaminare, altrimenti l'eventuale impulso riflesso non è distinguibile. L'analisi più semplice può essere fatta utilizzando direttamente un impulso molto breve, in modo da distinguerlo dalle relative riflessioni, oppure utilizzando un segnale a gradino con il fronte di salita molto breve, in modo da vedere direttamente gli effetti delle varie riflessioni che subisce. Lo schema di principio del setup è riportato in **Fig. 5**.

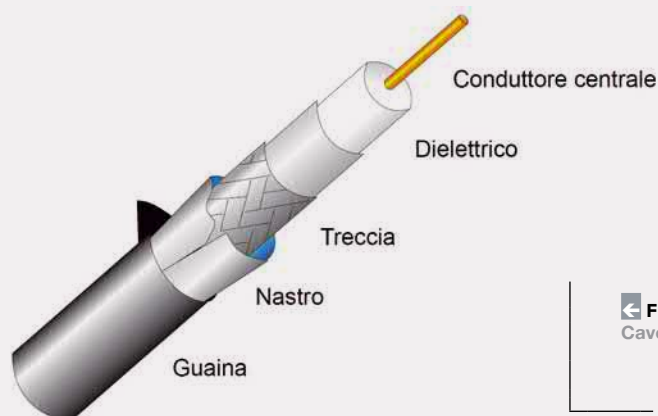


Fig. 4
Cavo Coassiale.

L'impedenza della linea è denominata Z_0 , quella della resistenza di terminazione è denominata Z_L . Di seguito vengono illustrate (**Fig. 6**) le tre forme d'onda ottenibili attraverso un segnale a gradino:

- nella prima immagine (a sinistra) è rappresentato il segnale a gradino nel caso in cui la linea è adattata, ovvero $Z_L = Z_0$;
- in centro è rappresentato il caso in cui $Z_L > Z_0$: si osserva un brusco innalzamento della tensione, dunque, la linea è disadattata;
- nell'immagine a destra, invece, è proposto il caso in cui $Z_L < Z_0$ (anche qui la linea risulta disadattata).

Prepariamo ora il setup (l'insieme di circuito e strumenti per la misura) seguendo lo schema di cablaggio proposto nella **Fig. 7**: colleghiamo il generatore ad onda quadra al cavo sotto esame e poniamo l'oscilloscopio in parallelo alla linea, utilizzando un connettore a T di tipo BNC. All'altra estremità del cavo inseriamo un trimmer multi-giri da 500 ohm.

In questo modo possiamo agevolmente modificare

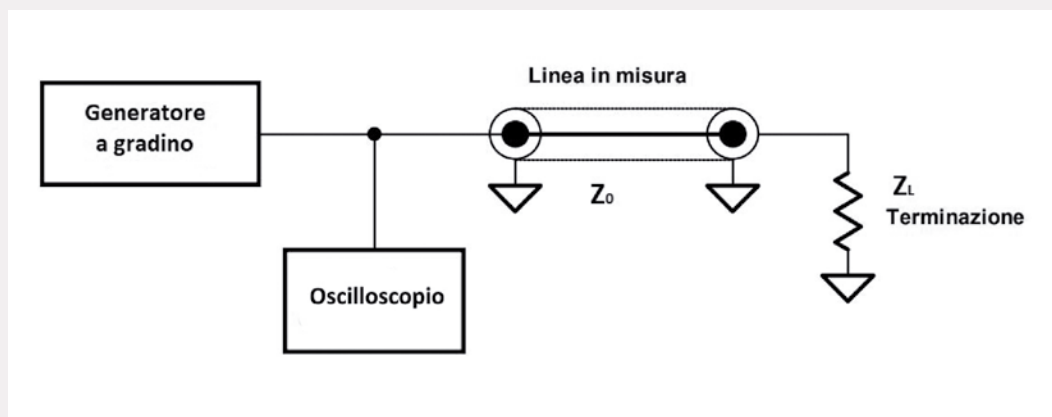


Fig. 5
Schema di principio.

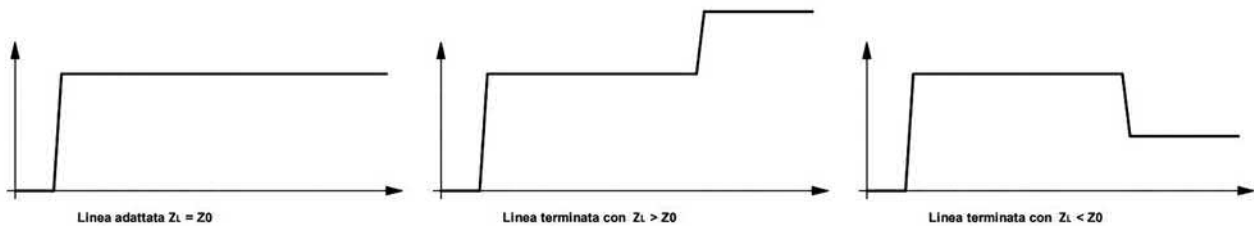


Fig. 6
Forme d'onda
ottenibili.

il valore dell'impedenza di carico, agendo sulla manopola del trimmer ed è possibile immediatamente osservare sull'oscilloscopio una variazione della forma d'onda. L'oscilloscopio servirà a valutare il segnale che viaggia lungo la linea. Possiamo così distinguere i casi dettagliati qui di seguito.

- Linea adattata, chiusa su un carico $Z_L = 50 \text{ ohm}$ (pari all'impedenza caratteristica del cavo RG58): impostiamo la scala dei tempi a 10ns/div, per avere una visualizzazione ottimale del segnale a gradino. Successivamente, ruotando il trimmer in modo da avere una resistenza da 50 ohm, sull'oscilloscopio si osserverà che il gradino (**Fig. 8**) ha ampiezza pari a circa la metà di quelli prodotti dal generatore a vuoto $V_{pp} \approx 2,5 \text{ V}$.
- Linea a circuito aperto (il carico è scollegato), la linea è disadattata: si osserverà un gradino che avrà una ampiezza pari a circa V_{pp} (**Fig. 9**).

- Linea in cortocircuito, la linea è disadattata: si osserverà un impulso che avrà ampiezza pari a $V_{pp} \approx 2,5 \text{ V}$ (**Fig. 10**).

IDENTIFICAZIONE LUNGHEZZA CAVO

Attraverso la riflettometria nel dominio del tempo, è possibile misurare la lunghezza del cavo o la distanza alla quale si è verificato un danneggiamento.

Infatti, nel caso in cui la lunghezza del cavo non è nota oppure vogliamo determinare la distanza alla quale si è verificata una rottura del cavo, è possibile farlo, misurando il tempo di ritardo Δt fra l'impulso trasmesso e l'impulso riflesso che arriva in ingresso.

Basterà impostare il banco di misura e la strumentazione come nella prova precedente, scollegando il carico (linea aperta).



Fig. 7
Il setup delle misu-
re: un BNC a T con-
sente di iniettare il
segnale all'ingresso
dell'oscilloscopio e
nel cavo testato.



◀ **Fig. 8**
Il segnale
a gradino
visualizzato
con la linea
adattata.



◀ **Fig. 9**
Segnale
tipico del cir-
cuito aperto.

Nota la velocità di propagazione v_p ($1,98 \cdot 10^8$ m/s), calcoliamo la lunghezza del cavo con la seguente formula:

$$l = v_p \frac{\Delta t}{2}$$

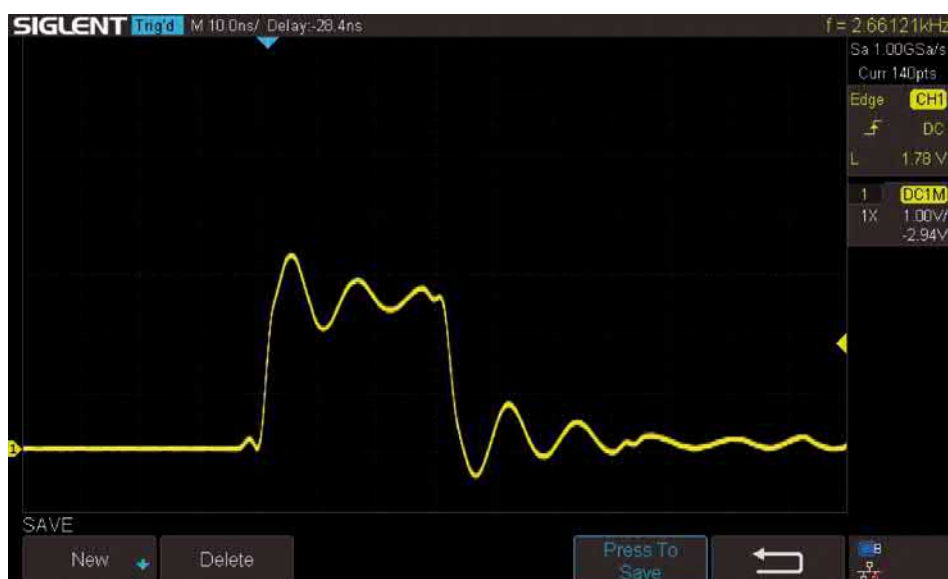
La determinazione di Δt e quindi del tempo di propagazione ($t_p = \Delta t/2$), nel caso di terminazione aperta, si può calcolare nel seguente modo: inviamo in ingresso all'estremità del cavo il segnale ad onda quadra attraverso il nostro generatore. Il segnale, si propaga lungo il cavo e quando il suo fronte di salita arriva sulla terminazione aperta, viene totalmente riflesso e ritorna sul genera-

tore in fase, sommandosi al segnale emesso dal generatore stesso, in modo da avere un brusco innalzamento della tensione, il cui valore all'incirca raddoppia. L'intervallo di tempo $\Delta t = 2t_p$ può essere determinato, misurando l'intervallo di tempo che intercorre fra il fronte di salita del segnale fornito dal generatore e l'istante in cui si verifica il brusco aumento della tensione in ingresso.

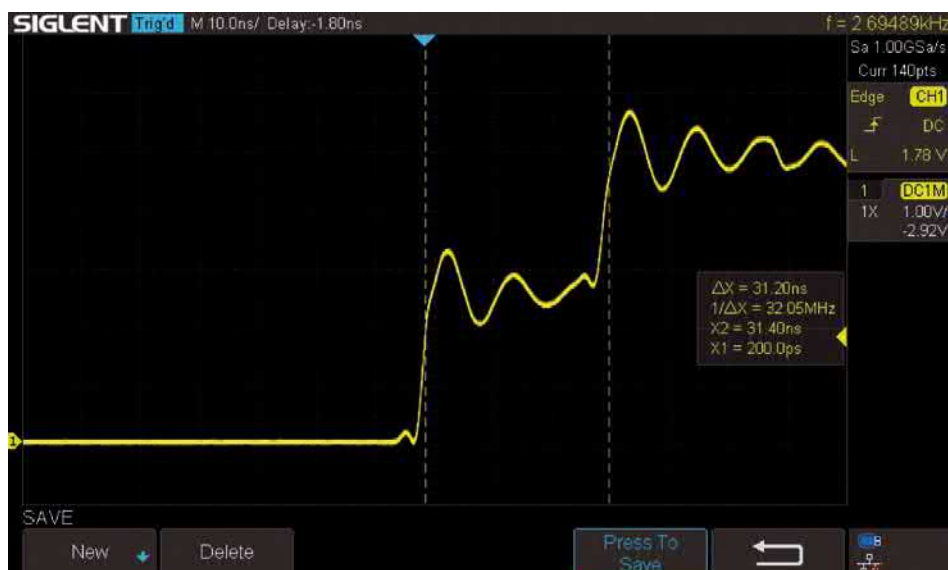
Nel nostro caso misuriamo un $\Delta t = 31,2 \cdot 10^{-9}$ s (Fig. 11). Sostituiamo i numeri nella formula e otteniamo la lunghezza del cavo in metri.

$$l = 1,98 \cdot 10^8 \frac{31,2 \cdot 10^{-9}}{2} \cong 3 \text{ m}$$

➔ **Fig. 10**
Segnale dovuto al cortocircuito della linea.



➔ **Fig. 11**
Misura del ritardo di propagazione.



CONCLUSIONI

In queste pagine abbiamo visto l'uso pratico della tecnica della riflettometria nel dominio del tempo, che è utilizzata in diversi ambiti e che, in questo articolo, è stata analizzata nell'ambito della caratterizzazione delle linee di trasmissione e nella misura dell'impedenza caratteristica di un cavo coassiale (RG58), utile anche a determinare problemi di integrità del cavo stesso. Le misure sono state effettuate realizzando un generatore ad onda quadra con tempo di salita molto breve, che pur essendo meno prestante di un impulsore, ci permette di fare le misure senza ricorrere a particolari procedure e a costi decisamente più modesti. Attraverso l'oscilloscopio si sono osservati i disa-

dattamenti di impedenza lungo la linea in base ai segnali e si è potuta così determinare la lunghezza del cavo. □



Cosa occorre?

I componenti utilizzati in questo progetto sono di facile reperibilità. L'integrato SN74AC14N (cod. SN74AC14N) è in vendita a Euro 1,60 mentre il PCB può essere realizzato con il servizio PCBPRODUCTION. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

Non farti sfuggire nulla.

Riprendi ogni attimo con le mini telecamere.



cod. FR806

€ 79,00

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

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



cod. FR834

€ 33,00

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

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



cod. FR825

€ 80,00

**TELECAMERA
A BOTTONE CON
TELECOMANDO**

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



cod. CP756

€ 19,90

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

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



cod. FR830

€ 38,00

**PENNA CON
TELECAMERA AD
ALTA RISOLUZIONE**

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

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

€ 19,50

**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

€ 34,00

**Anti flussante spray
200 ml**



cod. F02

€ 10,80

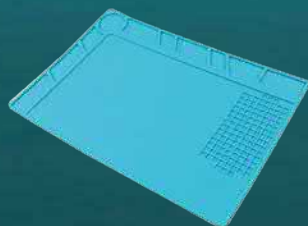
**Lacca isolante
spray 200 ml**



cod. V66

€ 11,50

**Tappeto in silicone per
saldatura 337x227 mm**



cod. AS10

€ 9,00

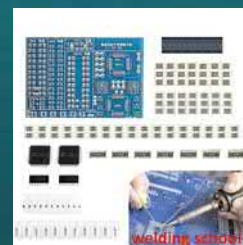
**Set di componenti per
imparare a saldare**



cod. YX286

€ 4,00

**Set di componenti SMD
per imparare a saldare**



cod. YX672

€ 5,00

SALDARE A STAGNO

di **DAVIDE SCULLINO**

Impariamo a saldare e dissaldare i componenti elettronici, da quelli più tradizionali a foro passante fino agli SMD nei package più moderni.

S

alda che ti passa... Si potrebbe parafrasare un celebre detto per rendere l'idea: imparare a saldare bene vi fa passare ogni timore che il vostro circuito, lo stiate costruendo oppure riparando, funzionerà! Ebbene sì, perché saper saldare i componenti elettronici è come un'arte e non è scontato che lo si sappia fare; anzi, come scopriremo in questo tutorial, probabilmente un po' tutti abbiamo qualcosa da imparare e prima o poi ci accorgiamo che non lo stavamo facendo correttamente.

In queste pagine vi spiegheremo come saldare e dissaldare i componenti elettronici, con varie tecniche, ciascuna delle quali si avvale di strumenti specifici usati più o meno propriamente. In origine era il saldatore o se vogliamo, c'era il saldatore a stagno: nei primi decenni dell'elettronica, infatti, i componenti

erano piuttosto primordiali e la loro saldatura richiedeva strumenti poco raffinati, ossia gli stessi in uso nell'Elettrotecnica "leggera" e quindi adatti a saldare fili, trasformatori di piccola potenza, reostati, condensatori e resistenze, ma anche bobine e capocorda, spinotti e via di seguito. Erano gli anni in cui si utilizzava come tecnica di saldatura quella che propriamente si chiama "brasatura" e che consiste nello scaldare un metallo e fondervi sopra della lega saldante a base di stagno e piombo (comunemente detta "stagno"), per unirlo ad un altro e realizzare una connessione sia elettrica che meccanica. Per favorire l'adesione e rimuovere l'ossido che, a lungo andare, interponendosi tra i metalli da saldare, avrebbe pregiudicato il contatto elettrico o addirittura staccato la saldatura, si spalmavano preventivamente le parti da saldare con della pasta flussante (talvolta chiamata pasta salda, ma impropriamente, come vedremo). Oggi il flussante è stato declinato in vari modi e viene integrato come anima nei fili di lega saldante a base di stagno, affinché si fonda durante la saldatura preparando le superfici alla migliore adesione possibile. Il flussante è una sostanza chimica che agevola la fusione della lega saldante e ne favorisce la distribuzione e l'adesione ai componenti da saldare.

LE LEGHE SALDANTI

Andiamo dunque con ordine e prima di passare agli strumenti e poi alla tecnica di saldatura (o brasatura, ora che sappiamo che è il suo vero nome) spendiamo qualche riga per spiegare cosa rende possibile la saldatura, ossia l'elemento che, fondendosi, fa più di una colla: attacca e realizza il contatto elettrico.



→ **Fig. 1**
Tipico rocchetto di
lega saldante.

Questo elemento è in realtà un composto metallurgico di due o più metalli sufficientemente duttili e malleabili, ma anche capaci di una ridotta resistenza elettrica: tipicamente è una lega di stagno (Sn) e piombo (Pb) con percentuali dei due metalli variabili tipicamente 60% Sn e 40% Pb o 63 Sn – 37 Pb. La lega saldante viene usualmente dispensata sotto forma di filamento in vari diametri (mediamente da 0,5 a 1,5 mm di diametro, ma dipende dalla destinazione) per l'elettronica civile e anche in lingotti per la saldatura industriale ad onda. All'interno del filamento viene inglobata, durante l'estrusione, della pasta flussante, che al momento della saldatura si fonderà perfezionando la brasatura. Tipiche in elettronica sono anche le leghe Sn-Pb 50/50, composte da metà di ciascuno dei metalli. Tenete presente che maggiore è la percentuale di piombo, più tenace è la saldatura ma minore diviene la conducibilità elettrica. Essendo il piombo un metallo pesante è pericoloso se penetra le falde acquifere o finisce nelle acque di mari, laghi e fiumi, da tempo sono state create leghe saldanti dette lead-free (senza piombo) o Pb-free, anche note come Rohs-compliant; le leghe tradizionali piombo-stagno (dette leaded) sono praticamente state abbandonate eccetto che in determinati ambiti, come ad esempio l'elettronica militare e aerospaziale, in virtù del fatto che la saldatura è più tenace e resiste alle forti sollecitazioni, di quanto non lo sia quella realizzata con le leghe lead-free.

Tipicamente la lega saldante Rohs è composta da stagno, argento (Ag) e rame (Cu) ed è quindi una Sn-Ag-Cu; ma esistono anche leghe di solo stagno e rame (Sn-Cu). Le percentuali dei metalli componenti sono variabili in base all'applicazione, comunque le più comuni leghe sono formate dal 90 al 99% di stagno, mediamente dal 3% di argento e dallo 0,5 allo 0,7% di rame. Quelle Sn-Cu sono tipicamente 99,3% di stagno e 0,7% di rame. Un tipico rocchetto di lega saldante è mostrato in **Fig. 1**: è il SOLD100G6SPB, composto da filo da 0,56 mm di diametro lead-free, composto da 0,3 % di argento, 0,7 % di rame, 96,5 % di stagno e 2,5 % di flussante; fonde a 217° C.

Le principali differenze pratiche tra le leghe Rohs-compatibili e leaded sono che le prime richiedono temperature di fusione maggiori e più flussante, ma aderiscono meno di quelle al piombo, ragione per cui spesso contengono più flussante (la percentuale è talvolta indicata nell'etichetta) o necessitano dell'aggiunta di pasta flussante durante la saldatura.

Inoltre le leghe Rohs-compatibili hanno un costo mediamente più alto, in virtù dei metalli più pregiati che contengono.

SALDATURA MANUALE

I componenti elettronici possono essere saldati sia manualmente che in maniera automatica; la parte che ci interessa è la prima, per eseguire la quale bisogna dotarsi di un minimo di attrezzatura a seconda del tipo di componentistica. Di base ci occorre:

- saldatore a stilo da 20÷50 W preferibilmente elettrico;
- del filo di lega saldante ed eventualmente del flussante;
- un dissaldatore o un succhiastagno;
- un tappetino per saldatura in materiale isolante, per proteggere il piano di lavoro.

Al posto del normale saldatore, meglio sarebbe usare una stazione di saldatura a stagno dotata di regolazione elettronica della temperatura; alcune stazioni dispongono di un dissaldatore a depressione, formato da uno strumento simile ad un saldatore, la cui punta è, però, cava e comunica con un tubo che porta a una pompa di depressione. Un buon esempio di saldatore a stilo è quello proposto a sinistra nella **Fig. 2**: si tratta del TP0055 della Futura Elettronica, un saldatore da 30W con elemento riscaldante di ceramica e punta da 1,2 mm; funziona a 230 Vca. A destra nella **Fig. 2** vedete invece una stazione di saldatura a stilo per utilizzo sia hobbistico che professionale: la VTSSC40N, sempre reperibile su www.futurashop.it. È utile anche disporre di trecciola dissaldante, che

è una sorta di striscia di maglia di filo di rame imbevuta di flussante che cattura lo stagno, permettendo di rimuoverlo assai bene dalle piazzole e dalle saldature intorno ai reofori e nei fori passanti metallizzati dove questi passano.

Un buon tappetino per la saldatura è l'AS10 della Futura Elettronica (www.futurashop.it): misura 337x227 mm ed è in silicone antistatico, resistente al calore fino a una temperatura di 500 °C e indeformabile. Il saldatore con cui si saldano i componenti è tipicamente a stilo, che è una punta variamente sagomata (dotata di impugnatura isolante dal calore) il cui scopo è fondere la lega saldante; la punta viene riscaldata internamente da una resistenza "corazzata" (rivestita da un involucro di acciaio inox o di materiale ceramico) cui è applicata. La punta può essere di rame rivestito (argentato) o di acciaio inossidabile; la prima si scalda più rapidamente ma dura meno, dato che è formata da un materiale più tenero e che si consuma a causa



Fig. 2
A sinistra un saldatore a stilo a 230Vca; a destra una tipica stazione saldante.



Fig. 3
Alcuni tipi comuni di punta per saldatore.

→ **Fig. 4**
Per saldare un
componente
appoggiate la punta
per saldatore su
reoforo e piazzola.



della pressione e dello sfregamento sulle zone da saldare, arrivando ad erodersi. La punta di acciaio è invece più duratura, anche se a causa dell'inerzia termica è più lenta a scaldarsi.

Negli ultimi anni hanno fatto la loro comparsa le punte per saldatore in materiale ceramico sintetizzato, che presentano elevatissima resistenza all'usura e sono elettricamente isolate.

Il saldatore per elettronica opera a temperature tipicamente comprese tra 230 e 400 °C; ha tipicamente una cordone terminante con una spina che ne permette l'alimentazione con la tensione di rete (220 Vca); talvolta dispone di un pulsante per funzionare a metà potenza, il che consente di scegliere tra due temperature di saldatura. I saldatori per elettronica hanno potenze che tipicamente spaziano da 20 a 50 watt.

La temperatura raggiunta dal saldatore viene influenzata da molte condizioni, non ultimi gli sbalzi della tensione di rete. Per effettuare saldature ad una temperatura stabile e nota, occorre utilizzare una stazione saldante, dove il controllo viene effettuato avvalendosi di un sensore collocato sulla punta.

Stazioni saldanti ne esistono di vario tipo, ma tutte funzionano secondo questo principio; la gran parte dispone di un selettore per impostare due o tre temperature o di una manopola o di un pannello a pulsanti per variare la temperatura a piacimento. Spesso lo stilo è a punte intercambiabili di varia forma (conica, a scalpello, a lama, piatta ecc.). In molte stazioni (JBC, per esempio) le punte intercam-

biabili incorporano la resistenza e lo stilo si limita a sostenerle e a fornire loro alimentazione. La **Fig. 3** propone vari tipi di punta per saldatore a stilo.

UTILIZZO DEL SALDATORE MANUALE

La saldatura o brasatura a stagno consiste nell'unire le parti (nel caso specifico si tratta dei terminali dei componenti elettronici e i fili o le piste dei circuiti stampati) che sono tipicamente di rame, mediante una certa quantità di lega saldante. Scaldato fino a fondere, lo stagno cola sulle parti e quando si raffredda le fissa l'una all'altra, stabilendo peraltro l'indispensabile collegamento elettrico. Allo scopo di migliorare la presa dello stagno, normalmente i terminali dei componenti elettronici sono stagnati in fabbrica ed è perciò che appaiono di colore grigio argenteo.

Affinché la saldatura sia corretta, occorre che le superfici da saldare non siano né umide, né ossidate; per rimuovere l'ossido dalle piste o dalle piazzole del circuito stampato si può operare in vario modo: con della comune paglietta per stoviglie, se il PCB non ha solder-resist o serigrafie sul lato saldature, ovvero con dei prodotti chimici o semplicemente della pasta flussante.

Nel caso di componenti THT (Through Hole Technology, ossia per montaggio a foro passante) la saldatura si esegue appoggiando la punta del saldatore per qualche secondo sulla pista di rame dove è già stato infilato il terminale del componente da stagnare e contemporaneamente su di esso, poi appoggiando il filo di lega saldante sul terminale o sulla pista (**Fig. 4**). Non bisogna appoggiare il filo sulla punta o quantomeno non bisogna farlo prima di aver scaldato pista e terminale, altrimenti è facile che si verifichi una "saldatura fredda": in pratica le parti sembrano unite ma non lo sono e la resistenza elettrica del contatto è troppo elevata per garantire il buon funzionamento del circuito. La saldatura fredda è molto insidiosa perché difficilmente visibile, in quanto il componente sembra saldato ma non lo è o comunque pur essendo meccanicamente stabile non c'è un buon contatto elettrico, anzi, spesso il contatto ha una resistenza elevatissima. La saldatura fredda si verifica perché la lega saldante fusa cola su parti (pista e componente) che non sono abbastanza caldi, quindi solidifica troppo rapidamente e non attecchisce come dovrebbe, oppure perché una o più parti sono ossidate.

Allo scopo va ricordato che la lega saldante ben fusa si riconosce, oltre che per l'aspetto liquido, per il colore, che diventa argento lucido; quando si



Fig. 5
A sinistra
un porta-
saldatore con
spugnetta
per pulire
la punta; a
destra una
lana metallica
con base per
pulizia della
punta del
saldatore.

raffredda e ritorna solido, lo stagno passa repentinamente dal colore lucido all'opaco. Dopo aver saldato un po' di componenti, il fluxante tende a lasciare residuo sulla punta del saldatore, che è quindi buona norma pulire periodicamente; il metodo più semplice è passarla sull'apposita spugnetta bagnata con acqua, che spesso viene fornita con il portasaldatore nell'apposito vano o con la stazione saldante; in alternativa è possibile utilizzare una "spugna metallica" come quella proposta in **Fig. 5**, reperibile su www.futurashop.it con il codice prodotto WT45239.

Per ottenere una buona saldatura e non surriscaldare i componenti, specialmente se si tratta di semiconduttori (diodi, transistor, TRIAC, ma soprattutto circuiti integrati e zoccoli) la punta del saldatore va mantenuta su piazzole e reofori giusto il tempo di veder diventare lucida la lega saldante, dopodiché va sollevata. Inoltre, laddove il saldatore lo consenta, bisogna impostare una temperatura adeguata; a riguardo va considerato che le leghe saldanti con piombo fondono a temperature mediamente di $220 \div 230$ °C, mentre quelle lead-free hanno temperature di fusione di una quindicina, anche una ventina di gradi centigradi in più. Per fare le cose bene conviene sempre consultare, quando disponibili, le schede tecniche delle leghe saldanti che si comprano, perché in base all'additivazione o composizione varia la temperatura di fusione.

Talvolta le leghe saldanti con metalli come l'argento richiedono un supplemento di fluxante, quindi prima della saldatura conviene spalmare della pasta fluxante (un ottimo esempio è la WT45238 reperibile on-line in vasetto su www.futurashop.it o la siringa Amtech, come la NC-559) sulle parti da

stagnare, così da migliorare l'aderenza. Quanto alla potenza del saldatore e alla temperatura di lavoro, considerate che la temperatura della lega saldante dipende non solo da quella impostata, ma anche e soprattutto dal rapporto tra la potenza del saldatore e la massa di lega saldante e di metallo interessati dalla saldatura; infatti un conto è la temperatura che il saldatore riesce a mantenere da solo, un altro è la quantità di calore ceduta, che dipende dalla potenza termica ed elettrica del saldatore.

Facciamo un esempio: lo stilo di una stazione saldante da 40 watt, può mantenere la temperatura impostata (tanto più quanto questa è alta) solo se si va a saldare piccoli componenti elettronici e non componenti elettromeccanici, ma anche se le piste del circuito stampato non sono particolarmente vaste e riccamente stagnate.

Anche la punta ha il suo ruolo: più vasta è la sua superficie, più calore può trasferire alla saldatura perché beneficia di una superficie più ampia. A saldature ultimate, conviene sempre ispezionare il circuito con una lente d'ingrandimento o una lampada con lente incorporata, necessari perché



Fig. 6
Un vasetto
di pasta
fluxante
WT45238.

Fig. 7
Un vasetto ed un
tubetto di pasta
saldante: quella a
sinistra è leadfree
e quella a destra è
con piombo.



quando si saldano chip con piedini molto ravvicinati è difficile riuscire a vedere ad occhio se non ci sono stati cortocircuiti, ed una lampada da tavolo per illuminare bene la zona di lavoro; ciò permetterà ad esempio di rivelare "baffi" di lega saldante che possono aver unito piazzole attigue. Siccome le saldature lasciano residui di fluxante, che ad alta temperatura possono indurire a contatto con l'aria, volendo è possibile pulire il circuito stampato con prodotti come l'anti-flussante spray commercializzato ad esempio da www.futurashop.it con il codice prodotto F02, che è un efficace detergente per qualunque tipo di componente elettrico ed elettronico: la completa evaporazione e la compatibilità con le plastiche lo rendono insostituibile nel preparare ogni circuito in modo da ottimizzarne la copertura dielettrica mediante lacca isolante quale ad esempio la V66.

Questa lacca è molto comoda ad esempio per passivare la superficie inferiore del circuito stampato e prevenire contatti accidentali con le piazzole e i reofori dei componenti.

In alternativa al rimuovi-flussante suddetto è possibile utilizzare solventi come la trielina, però quest'ultima intacca le plastiche di alcuni connettori e componenti.

Per le operazioni di "lavaggio" delle schede con l'anti-flussante e comunque per tutte quelle di saldatura è sempre utile disporre di un aspiratore di fumi o di una piccola cappa, che risucchi i vapori e i fumi dell'evaporazione del fluxante.

Un esempio è l'aspirafumi da banco per saldatura commercializzato da www.futurashop.it con il codice prodotto TP0065, dotato di filtro a carboni attivi: è dotato di un'elettroventola che aspira e getta lateralmente, lontano, fumi e aerosoli derivanti

da saldatura, dissaldatura e pulizia delle schede mediante solventi.

INTEGRATI E COMPONENTI CON MOLTI PIEDINI

La saldatura dei circuiti integrati, come dei connettori con molti piedini a passo fine, impone alcune accortezze in più di quelle appena esposte: infatti è più facile surriscaldare tali componenti perché il saldatore toccherà più volte i relativi contatti. In questi casi, specie nei circuiti integrati, è consigliabile stagnare alternativamente i piedini dei due lati, magari incrociandoli, ossia scendendo da un lato e salendo da quello opposto. Discorso analogo si applica a componenti con piedini disposti su quattro lati.

SALDARE GLI SMD

Per quanto riguarda la componentistica SMD, la tecnica di saldatura differisce in base al package con cui bisogna avere a che fare: molti elementi possono essere saldati manualmente con il classico saldatore a stagno, mentre altri richiedono una stazione ad aria calda. Esistono anche vari approcci, relativamente alla lega saldante da utilizzare, che può essere il classico filo, ma anche la pasta saldante: quest'ultima è un impasto di lega saldante e fluxante, molto malleabile a freddo tanto da poter essere spalmata sulle piazzole del circuito stampato; si trova in commercio in siringhe (come ad esempio per posizionarla con precisione su piazzole molto piccole) o in barattoli (**Fig. 7**). La pasta saldante è prodotta da varie aziende: un esempio di confezione in siringa è quella commercializzata da www.futurashop.it con codice SMDPASTASALD ed è composta dal 63% di stagno e dal 37% di piombo; è prodotta dalla

MECHANIC. Altro esempio, sempre acquistabile su www.futurashop.it, è la pasta in vasetto (codice prodotto SOLDER50G) la cui composizione è Sn 96,5%, Ag: 3%, Cu: 5%.

PER FARE PRATICA

Bene, fin qui vi abbiamo spiegato in cosa consiste e come si effettua la saldatura a stagno dei componenti elettronici, però a questo punto chi è alle prime armi, ossia sta iniziando con le sue prime saldature, si starà domandando: "da dove inizio? ...e con che cosa?". La domanda è lecita, nel senso che come quando si impara a guidare si parte da un'automobile magari dotata di doppi comandi, i primi passi nel mondo della saldatura dei componenti sarebbe meglio farli non su un circuito che poi si deve utilizzare, ma su qualcosa che non serve e che si può sacrificare.

Ecco perché esistono dei kit per l'apprendista saldatore, composti da una basetta con le piazzole e da un mucchietto di componenti assortiti per fare pratica (**Fig. 9**); kit di questo tipo si possono trovare ad esempio on-line sul sito www.futuranet.it sia per il montaggio dei componenti a foro passante, sia per gli SMD.

Il kit per i THT è il prodotto Set di componenti per imparare a saldare YX286 e contiene una basetta millefori (con piazzole a passo 2,54x2,54 mm) da 50x70 mm, per un totale di 39 elementi, ossia:

- 4 diodi 1N4007;
- 2 diodi 1N4148;
- 1 integrato DIP NE55;
- 4 condensatori in poliestere varie capacità;
- 5 condensatori ceramici di varie capacità;
- 6 condensatori elettrolitici di varie capacità;
- 7 resistori 1/4W;
- 8 resistenze 1/8W;
- 9 LED tondi;
- 1 induttanza assiale;
- 11 Transistor 8050, 8550, 9013.

Insomma, ce n'è abbastanza per fare pratica. Quanto agli SMD, potete fare riferimento al Set di componenti per imparare a saldare YX672 che contiene un PCB da 80x60 mm con punti di test ed elementi dello stesso tipo interconnessi. Il set contiene:

- resistenze formato 0805 (20 pezzi, che nel PCB sono collegati tutti in serie);
- resistenze in formato 0603 (34 pezzi, tutti in serie);
- resistenze in formato 0402 (10 pezzi, tutti in serie);

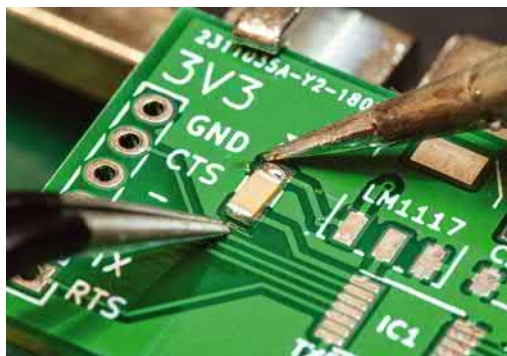


Fig. 8
Alcuni componenti SMD possono essere saldati con il comune saldatore.

- resistenze 0606 (6 pezzi, tutti in serie, con punti di prova);
- condensatori in formato 0805 (20 pezzi, tutti in serie);
- diodi 3216 (10 pezzi, tutti in serie);
- transistor SOT-23 (6 pezzi con test point);
- integrati in package SO-14 (2 pezzi, con test point);
- integrati in case LQFP44 (2 pezzo, con test point).

In questo caso sono previste piazzole di prova per verificare se le saldature sono state eseguite correttamente o se sono fredde.

DISSALDARE I COMPONENTI

Sebbene l'obiettivo di questo tutorial sia la sal-



Fig. 9
Due set per imparare a saldare componenti THT ed SMD

Fig. 10
Dissaldatura di un integrato SMD con un saldatore a punta a spatola.

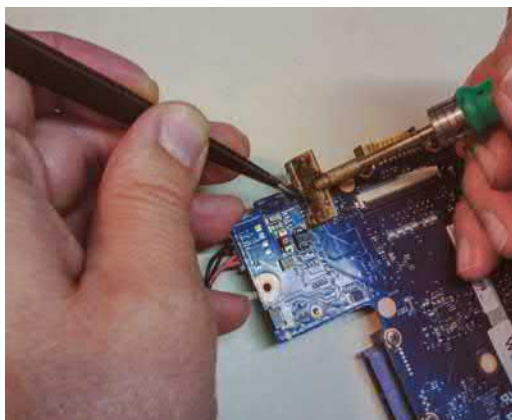


Fig. 11
Dissaldatura di un componente SMD con una pinza a becchi riscaldati.

Fig. 12
Dissaldatura di un componente con due saldatori.

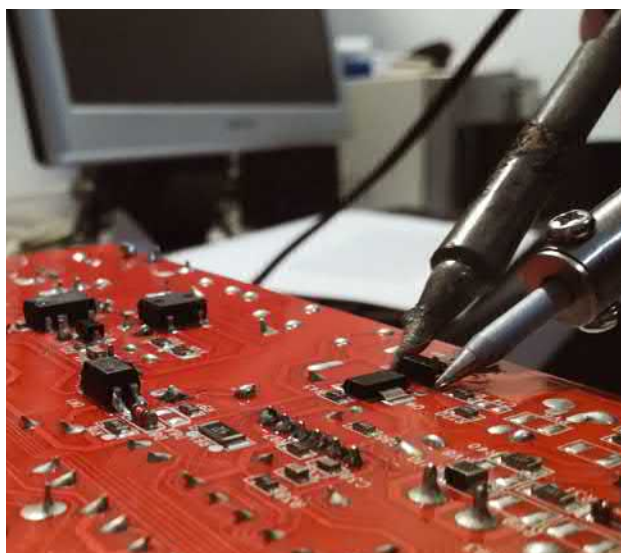
datura dei componenti elettronici, è interessante notare che il saldatore può anche essere utilizzato per dissaldare i componenti: basta appoggiare la punta sulle piazzole fino a far sciogliere la lega saldante, quindi asportare il componente del caso con delle pinzette. Ciò è fattibile per componenti a pochi terminali o, utilizzando punte piatte (a spatola) o a lama, con vari terminali in linea o ravvicinati, come i connettori o gli integrati; la **Fig. 10** mostra la dissaldatura di un chip TSSOP con lo stilo di una stazione dissaldante a punte intercambiabili, su cui è stata montata la punta a lama invece della tradizionale a stilo: si scaldano prima i terminali di un lato e poi quelli dell'altro, estraendo il componente con una pinzetta.

Per dissaldare i componenti SMD, se hanno due soli terminali (come diodi, resistenze, condensatori) è possibile ricorrere a una particolare pinza a becchi riscaldati, che può essere un attrezzo a sé

oppure parte di una stazione dissaldante; si utilizza come mostrato nella **Fig. 11**, ossia pinzando il componente dagli elettrodi, attendendo lo scioglimento della lega saldante, quindi sollevando il componente. Nella **Fig. 12** vedete un'altra tecnica che consiste nell'appoggiare sui terminali del componente la punta di un saldatore per parte, stringendo poi con i due saldatori quando la lega saldante è fusa, in modo da sollevarlo ed estrarlo.

IL SUCCHIASTAGNO

È l'attrezzo più utilizzato per dissaldare i componenti elettronici a montaggio tradizionale (i THT) con molti piedini stampati a doppia faccia con fori metallizzati, dato che il saldatore non basta, in quanto lo stagno rimarrebbe negli interstizi fra terminali e pareti dei fori. Nella sua forma più semplice, si tratta di una pompetta a mano formata da un cilindro cavo contenente uno stantuffo che si carica con una molla e premendo un pulsante ritorna a riposo, aspirando la lega saldante mediante un beccuccio metallico o di teflon comunicante con la camera della pompetta. Tale succhiastagno, che è meccanico, si usa in abbinamento al saldatore: quest'ultimo fonde lo stagno da rimuovere e il succhiastagno lo aspira. Siccome tenere in mano due attrezzi è scomodo, sono stati inventati i dissaldatori, che sono dei succhiastagno a punta metallica (in rame o, meglio ancora, acciaio inox) riscaldata come nei saldatori. In essi il cilindro comunica con la punta riscaldata da una resistenza elettrica e cava internamente (un ottimo esempio è il prodotto 8400-VTDESOL3 reperibile su www.futuranet.it). La **Fig. 13** mostra come si effettua la dissaldatura



con tale attrezzo: la punta si appoggia facendo entrare il terminale del componente nel foro e spingendo delicatamente per farla ben aderire alla saldatura, quindi si attende la fusione della stessa, allorché si preme il pulsante.

Il succhiastagno può essere una macchina a sè, sovente integrata nelle stazioni di saldatura: in tal caso si tratta di un dissaldatore come quello appena descritto, dove il risucchio avviene per la depressione generata da una pompa a vuoto e trasmessa alla cavità della punta mediante un tubo di gomma; la pompa si attiva con un interruttore a pedale o posto sullo stilo del dissaldatore. La pompa può aspirare in continuazione, diversamente da quella del dissaldatore comune, che aspira una sola volta e poi va ricaricata; per questa ragione la stazione saldante con dissaldatore è più efficace nel rimuovere anche grandi quantità di stagno. Lo stagno aspirato raffredda quasi subito e solidifica, formando all'interno dello strumento masse più o meno grandi che devono essere fermate prima di raggiungere la pompa aspirante; allo scopo, all'interno dei dissaldatori si trova un filtro a retina o formato da un tampone di cotone, in modo da lasciar passare l'aria ma da trattenere lo stagno. Col passare del tempo e con una frequenza che dipende dall'uso che si fa dell'attrezzo, il filtro va rimosso e sostituito con uno idoneo. Inoltre la punta, specie se è di rame, tende a formare incrostazioni dovute al cedimento della parete interna a causa del calore e col tempo si intasa; perciò va pulita con uno spillo o del sottile filo di ferro, oppure appositi scovolini forniti in dotazione al dissaldatore.

SALDATURA DI SMD AD ARIA CALDA

I componenti a montaggio superficiale, specie quelli più piccoli e gli integrati con piedini fittissimi o PAD sotto al corpo (per esempio QFB, LGA e BGA) devono essere saldati con una stazione ad aria calda, dopo aver stagnato o spalmato di pasta saldante le rispettive piazzole; tale operazione può essere svolta anche all'interno di un fornello specifico per queste lavorazioni. Esiste anche la possibilità di ricorrere a macchine automatiche o semiautomatiche, ma la cosa riguarda un ambito che esula dagli scopi di questo tutorial.

La stazione ad aria calda è una macchina che per saldare o dissaldare riscalda i componenti o le loro connessioni mediante un getto d'aria calda emesso da un erogatore con impugnatura termoisolante ed un ugello metallico variamente sagomato. Del getto si possono regolare, mediante comandi posti sul pannello frontale, il flusso e la temperatura.

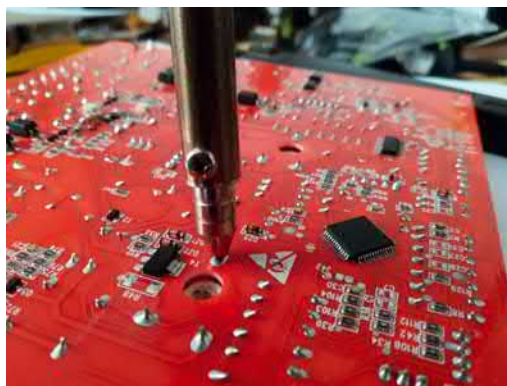


Fig. 13
Dissaldatura di un componente con il dissaldatore riscaldato.

La stazione a getto d'aria calda consta di una ventola o pompa da aria di tipo rotativo a bassa pressione (sempre comandata da un motore elettrico), che aspira aria dall'esterno (protetta da un filtro) e la soffia in un tubo che porta alla testa dell'erogatore; quest'ultimo è un ugello tubolare che può essere variamente sagomato oppure avere forma cilindrica ma essere predisposto per l'applicazione di adattatori (deflettori) in grado di dirigere l'aria in vari modi.

Un buon esempio di stazione del genere è la VTSS230 reperibile sempre su www.futuranet.it: integra un saldatore a stilo ed è quindi una stazione combinata di saldatura/dissaldatura (**Fig. 14**). La dissaldatura ad aria calda si effettua appoggiando su un piano metallico la scheda da cui staccare i componenti e dirigendo il getto caldo sui piedini di tutti i lati, passando in sequenza da uno all'altro; quando la lega saldante sta per fondersi diventa lucida, si può asportare il chip. Allo scopo bisogna, con mano ferma, afferrarlo mediante una pinzetta a molla, sollevandolo con decisione ed evitando di urtare i componenti che si trovano accanto; infatti, se l'elemento da staccare si trova in una zona del circuito stampato densamente "popolata" di componenti, è facile che anche quelli intorno siano scaldati al punto da sciogliere la lega saldante, il che vuol dire che basta urtarli per staccarli e fare un piccolo disastro, anche perché il trascinamento di un componente provoca lo spargimento dello stagno e il cortocircuito di piste e componenti vicini, poi difficile da rimuovere.

La dissaldatura riesce meglio se il circuito viene collocato sopra una piastra riscaldante, in quanto, mentre con la sola aria calda si può scaldare un lato e quindi occorre raggiungere temperature molto elevate prima di ottenere lo scioglimento della lega saldante, scaldando da sotto lo stampato si prepara lo stagno alla fusione, che può

Fig. 14
Una stazione
ad aria calda.



avvenire con una minore temperatura sul lato investito dall'aria calda. Ne beneficiano la durata dei componenti e le saldature: infatti se un lato dello stampato viene riscaldato fortemente rispetto all'altro, la differenza termica può far inarcare il circuito e deformare o staccare delle saldature, mentre con lo stampato uniformemente riscaldato questo rischio non esiste.

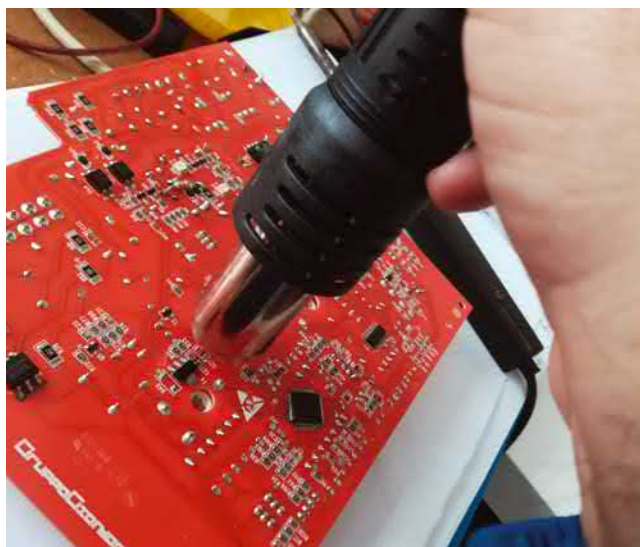
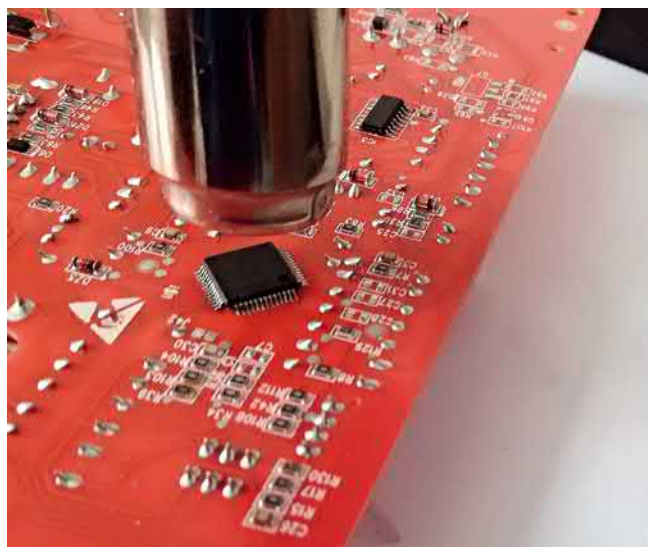
Quanto alla saldatura, si effettua semplicemente appoggiando il chip sulle rispettive piazzole, ben centrato rispetto ad esse; posizionato il componente, si può scaldarlo, badando che questa volta il getto d'aria calda deve essere abbastanza debole, altrimenti se soffia forte può spostare il chip, tanto più se questo è piccolo. Infatti, mentre quando si

tratta di dissaldare un componente si può anche incrementare la potenza del getto per aumentare rapidamente la temperatura, qui occorre cautela. La saldatura riesce meglio se prima di appoggiare il componente si depone sulle piazzole un sottile strato di fluxante: infatti questo materiale agevola la fusione e l'adesione della lega saldante attorno ai terminali del chip, impedendo che essa vada a cortocircuitare contatti attigui.

Anche nel caso della saldatura, l'operazione riesce meglio se lo stampato viene prima riscaldato mediante una piastra.

Alcune macchine a getto d'aria calda dispongono di un sistema di spegnimento ritardato che consente di raffreddare il riscaldatore quando

Fig. 15
Utilizzo della
hot air station
per saldare
o dissaldare
chip quadrati
o BGA.



l'utente spegne il macchinario; ciò comporta che se si appoggia l'erogatore, dal suo ugello uscirà ancora per un po' aria calda e quindi se si è spenta la macchina quando era a temperatura molto alta, ciò può essere pericoloso. Per questa ragione non bisogna mai appoggiare l'erogatore su un tavolo o di fronte a materiale infiammabile (solventi, alcol, carta) o deformabile col calore (polistirolo e plastiche deboli) ma, piuttosto, riporlo sul proprio supporto. Tale accorgimento vale anche per le macchine a spegnimento immediato, quando vogliamo appoggiare momentaneamente l'erogatore perché per qualche istante non ci serve e non vogliamo spegnere la macchina (ad esempio perché è di quelle in cui ogni volta che si accende bisogna reimpostare con i pulsanti la temperatura e la pressione dell'aria); in ogni caso accertatevi che dietro il sostegno dell'erogatore non vi siano oggetti infiammabili o che si possono sciogliere (Fig. 15).

SMD... AL FORNO!

Quando i componenti da saldare sono tanti e piccoli, la stazione ad aria calda può non essere la soluzione migliore; in alternativa è possibile ricorrere alla saldatura in forno, posizionando tutti i componenti sulle rispettive piazzole del circuito

stampato preventivamente spalmate di pasta saldante e inserendo il PCB lentamente e delicatamente (per non spostare i componenti) sul piano del cassetto di un fornello per saldature SMD (SMD Oven) come quello proposto dalla Fig. 16. Questo genere di forni ha un riscaldatore a resistenza elettrica o ad infrarossi gestito da micro-processore, che permette di eseguire sequenze preimpostate o definibili dall'utente, in maniera del tutto automatica. Mediamente la saldatura richiede un tempo di due o tre minuti e la temperatura impostata cresce gradualmente fino a aggiungere un picco di $250 \div 300$ °C (dipende dalla lega saldante utilizzata) per il tempo occorrente a fondere le sfere (il forno deve essere illuminato e quanto accade all'interno dovete poterlo vedere chiaramente) dopodiché torna a scendere per far raffreddare gradualmente la scheda. Se tra i componenti da saldare ce ne sono alcuni danneggiabili ad alte temperature, il consiglio è di effettuare una saldatura selettiva, saldando prima quelli più resistenti, a temperatura maggiore, poi quelli che tollerano una temperatura minore, impostando appunto una temperatura più bassa, appena sopra quella di fusione della lega saldante. In quest'ultimo caso un po' di pasta flussante agevolerà il processo di saldatura. □



Fig. 16
Fornetto per
saldatura
degli SMD.

FUTURA GROUP
IDEE IN ELETTRONICA

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

FUTURA ELETTRONICA
SHOP ON LINE

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

Elettronica In
PROFESSIONAL
Elettronica In
ELETTRONICA PER APPLENTI

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

FUTURA GROUP
PROGETTI

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

Futura Academy
CORSI DI FORMAZIONE

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

FUTURA ELETTRONICA
PROGETTAZIONE

Un team di esperti in grado di studiare, progettare e realizzare prototipi utilizzando le ultime tecnologie. Il team Futura Group è composto, oltre che dal personale dei nostri laboratori tecnici, anche da un numero elevato di collaboratori esterni, ciascuno specializzato nel proprio settore.

FUTURA ELETTRONICA
RIVENDITORI

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

FUTURA GROUP
IDEE IN ELETTRONICA

Categorie ▾

HOME

SHOP

RIVISTE

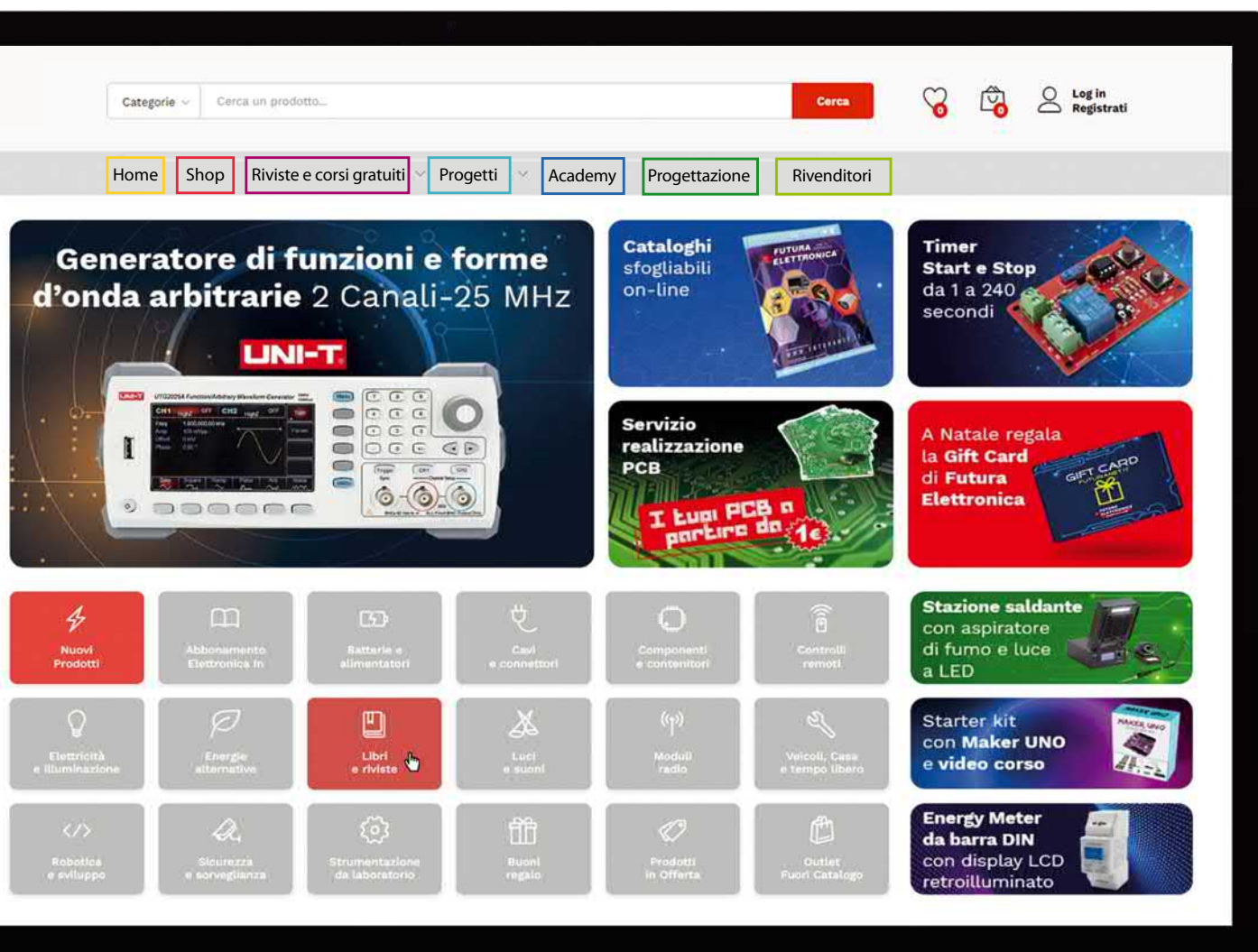
PROGETTI

ACADEMY

PROGETTAZIONE

RIVENDITORI

È on-line il NUOVO PORTALE FUTURANET.IT



NUOVE TELECAMERE

FULL COLOR

technology

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

IP CAMERA 4 MPX – FULL COLOR

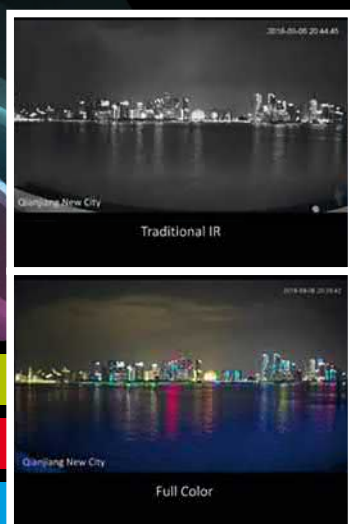
Risoluzione immagine: 4 MPX

Obiettivo: 3,6 mm

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

Grado di protezione: IP66

Alimentazione: 12 VDC



€ 52,00

cod. FR816N

**TELECAMERA FULL COLOR
2 MPX**

Risoluzione immagine: 2 MPX

Obiettivo: 3,6 mm

Portata Illuminatore: 20-30 metri

Grado di protezione: IP67

Alimentazione: 12 VDC



€ 69,00

cod. FR816VAR

**TELECAMERA FULL COLOR
2 MPX - VARIFOCAL**

Risoluzione immagine: 2 MPX

Obiettivo: varifocal 2,8-12 mm

Portata Illuminatore: 40 metri

Grado di protezione: IP67

Alimentazione: 12 VDC



€ 44,00

cod. FR817

**TELECAMERA DOME
FULL COLOR - 2 MPX**

Risoluzione immagine: 2 MPX

Obiettivo: 3,6 mm

Portata Illuminatore: 20-30 metri

Grado di protezione: IP67

Alimentazione: 12 VDC

IoTicontrollo

a cura della REDAZIONE



Una suite di dispositivi di controllo e acquisizione da remoto gestiti tramite un'app attraverso un servizio NBloT che si appoggia alla rete cellulare LTE.

Il mondo dell'IOT (Internet of Things) è vastissimo e tocca tutti gli ambiti, da quello civile e quindi la vita di tutti i giorni, a quello industriale (dove prende il nome di IIoT, ossia Industrial Internet of Things). Le tecnologie utilizzate per connettere i dispositivi sono le più svariate e ognuna è adatta ad eseguire compiti specifici, perché non c'è una sola tecnologia che possa coprire tutte le esigenze.

La parte del leone in ambito domotico la fanno il WiFi e i protocolli custom dei vari costruttori: Alexa è il classico esempio di applicazione IoT in ambito domotico.

Ma la vera Internet of Things è quella che per la connessione tra dispositivi si appoggia alle cosiddette LPWAN (Low Power Wide Area Network) cioè bande di frequenza adatte per coprire lunghe distanze, essere robuste ai disturbi e adatte per

realizzare prodotti alimentati a batteria. All'interno del mondo LPWAN (o LPWA – Low Power Wide Area, se preferite...) spiccano quelle supportate dalle bande licenziate e gestite da vari player telefonici e nello specifico la tecnologia NB-IoT (Narrow band IoT) e il CAT-M, ovvero in senso generale l'LTE. La tecnologia Narrow Band (NB-IoT) è già da tempo utilizzata per connettere contatori del gas e dell'acqua o sensori di allarme di qualsiasi natura. Caratterizzata da un consumo energetico molto basso, permette di realizzare dispositivi con durata di vita con batteria superiori ai 10 anni con una comunicazione giornaliera. La frequenza utilizzata di 800 MHz le consente di arrivare in posti dove il cellulare comune non funziona più, ad esempio negli scantinati, nel sottosuolo e in alta montagna. Sfruttando la copertura del 4G, è pressoché presente su tutto il territorio, non solo italiano ma anche europeo e mondiale. Utilizzando SIM degli operatori internazionali si può coprire tutta l'Europa con la stessa tariffa, peraltro molto contenuta.

I limiti di questa tecnologia sono legati principalmente al fatto che non si può comunicare in continuo, ma il dispositivo deve inviare pacchetti di dati e non deve veicolare filmati o altro di "pesante", per questo motivo è sufficiente l'utilizzo di bande limitate per traffico orario di dati.

La tecnologia Cat-M utilizza la rete cellulare 4G che, rispetto al Narrow Band IoT, si caratterizza per un'ampia copertura, performance stabili anche in mobilità e una capacità di trasmissione dati elevata che insieme ne fanno la tecnologia ideale per soluzioni IoT in mobilità.

Sfruttando l'NB-IoT, un'azienda italiana ha realizzato il sistema IoTcontrollo, coniando questo termine a connotare l'idea: un sistema IoT integrato ed appoggiato ad un Cloud proprietario, che consente di controllare nel senso più ampio del termine delle unità remote e acquisire parametri e impostare dispositivi locali, tramite una comoda App per smartphone; insomma, un'interpretazione particolare dell'IoT che è mirata al controllo remoto semplice e sicuro, senza i limiti che le tecnologie LoRa o SigFox o altre proprietarie imporrebbero, giacché qui ci si appoggia alla rete radiomobile 4G sfruttando la tecnologia Narrow Band IoT e più in generale l'LTE.

IOT E TECNOLOGIA LPWAN

Attraverso chip e sensori inseriti al loro interno, gli oggetti connessi sono in grado di interagire tra loro e con la realtà circostante, permettendo in questo

modo al mondo fisico di essere progressivamente digitalizzato e monitorato.

L'utilizzo delle reti a bassa potenza ed ampio raggio, ossia LPWAN, permette la connessione tra questi oggetti con basse velocità di comunicazione e scambio di poche decine o centinaia di bit nel complesso. Ricordiamo che l'accesso a queste reti, come anche a quelle implementate con i protocolli LoRa e SigFox, consente di coprire grandi distanze a patto di impiegare una banda minima, ed è per questo che a tali sistemi di comunicazione vengono appoggiati dispositivi che richiedono la trasmissione periodica di modeste quantità di dati. L'impegno orario o giornaliero delle bande assegnate non deve superare una certa percentuale (duty-cycle dei dati) e la comunicazione avviene tipicamente impiegando protocolli come l'MQTT (Message Queue Telemetry Transport) ossia un protocollo di messaggistica ISO basato su standard, o un insieme di regole, utilizzato per la comunicazione tra macchine e collocato in cima allo stack TCP/IP; per quanto riguarda il sistema IoTcontrollo, invece, viene adottato un protocollo proprietario UDP per minimizzare il numero di byte trasmessi.

I vantaggi offerti dall'utilizzo della tecnologia LPWAN NB-IoT sono:

- efficienza energetica e quindi lunga autonomia nell'alimentazione a batterie (fino a 10 anni);
- facilità di sviluppo e supporto perché basata sull'infrastruttura cellulare;
- copertura outdoor e indoor (quindi permette di accedere a dispositivi collocati in cantine e tombini, come i contatori per smart metering);
- connessione sicura e protetta, grazie alla cifratura della rete 4G;
- capacità di connessione di 100.000 dispositivi per singola cella della rete radiomobile;
- bassi costi dei componenti;
- nessuna limitazione del duty-cycle della comunicazione;
- frequenza di libero utilizzo;
- copertura praticamente mondiale.

Nel caso dei dispositivi IoTcontrollo, la tecnologia LPWAN impiegata è quella che si appoggia all'NB-IoT (vedere il riquadro dedicato in queste pagine) e quindi alla rete radiomobile 4G in banda 800 MHz. Questa tecnologia si accontenta di una banda larga appena 200 kHz per la comunicazione dati, che può essere ricavata nella parte laterale delle bande di frequenza impiegate dalle reti cellulari già esistenti, senza creare interferenze; una



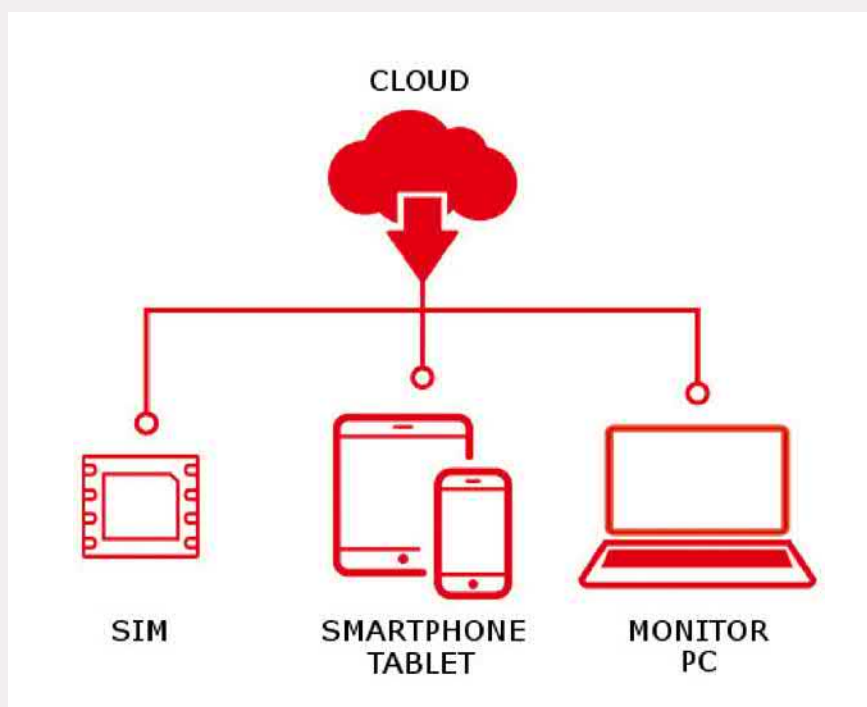


Fig. 1
Schematizzazione del sistema.

banda così ristretta (non a caso è chiamata narrow band) consente comunque, l'invio di pacchetti dati da 200 kbit, più che sufficienti per le esigenze del mondo IoT.

L'AMBIENTE IOTICONTROLLO

L'ecosistema IoTicontocontrollo è composto sostanzialmente da tre attori: gli end-device, che vengono collocati dove serve controllare processi o acquisire dati, il Cloud dedicato e proprietario di IoTicontocontrollo, che gestisce l'intero progetto ed è interfacciato tramite rete cellulare e protocollo NB-IoT, ed infine l'App installata su smartphone, la quale si abbina ad uno o più end-device e interagisce attraverso il Cloud (**Fig. 1**).

Il bello del sistema è che i dispositivi IoT si connettono sfruttando una SIM virtuale (vale a dire una SIM on chip di un gestore che, nello specifico, è 1NCE) fornita da IoTicontocontrollo insieme ad essi, che offre connettività illimitata (per l'IoT) e che non bisogna preoccuparsi di ricaricare.

Al momento l'offerta dell'azienda (ioticontocontrollo.it) conta sei prodotti, ossia altrettanti end-device IoT, ciascuno rivolto a un proprio campo applicativo:

- Security IoT Box;
- Power Control Box;
- Smoke Detector;
- Plug Meter;
- Motion IoT Sensor;
- Door Alarm.

Tutti sono autoalimentati, a batteria ricaricabile o pila di lunga durata, a seconda della configurazione. Tutti dispongono di una connettività dati che supporta la connessione permanente e l'utilizzo del canale dati per un massimo di 24 trasmissioni al giorno e, considerata la modesta quantità di dati scambiati, il traffico è illimitato.

I device infatti dispongono di connettività garantita ad un costo fisso che non grava sull'utente, per un totale di 500 MB, che per un dispositivo IoT bastano e avanzano.

Facciamo dunque una carrellata sui dispositivi di IoTicontocontrollo.

Security IoT Box

Questo dispositivo viene incontro alle esigenze di sicurezza e si può installare abbinato a porte, cancelli e varchi in generale, oppure macchinari; rileva semplicemente il movimento grazie ad un accelerometro incorporato, che invia le informazioni sulle accelerazioni rilevate lungo più assi e quando la soglia di spostamento viene superata trasmette un messaggio di allarme al Cloud e da esso all'App, che vi è connessa. Il modulo Security IoT Box è proposto in **Fig. 2**.

Il dispositivo è alimentato attraverso una batteria che gli consente un'autonomia di 18 mesi (alla temperatura operativa di 20°C, trasmettendo 1 evento al giorno, con copertura cellulare ottimale. Ha una temperatura di esercizio compresa fra -10

Fig. 2
Security IoT Box:
notate il QR-code
sul corpo, che serve
per l'abbinamento.



e +50°C, quindi funziona in ogni ambiente, ed è impermeabile con grado di protezione IP65, tanto da poter funzionare all'aperto.

Misura 146 x 86 x 19,5 mm e pesa intorno ai 100 grammi. È possibile, sempre tramite App, impostare un tempo di attivazione, vale a dire un intervallo trascorso il quale il dispositivo deve osservare lo stato dell'accelerometro e inviare l'allarme; dall'App è possibile fare una programma-

zione settimanale di quando debba essere attivo l'accelerometro, così da rilevare solo nei periodi in cui occorre farlo, ignorando quelli restanti (per esempio ignorare spostamenti e vibrazioni quando rientrano nella normale attività).

All'accensione del dispositivo è previsto un intervallo di quiescenza (un tempo di attivazione) pari a un massimo 5 minuti, che serve a dare il tempo al dispositivo stesso di connettersi alla rete LPWA e ad essere associato all'App. Questo modo di funzionamento vale per tutti i dispositivi descritti in questo articolo.

Il dispositivo non richiede SIM tradizionale perché ha una SIM on chip di un operatore virtuale il cui utilizzo non richiede alcuna operazione da parte dell'utente, giacché l'abbonamento corrispondente è associato da IoTicentro al dispositivo stesso.

Power Control Box

Questo dispositivo di IoTicentro (**Fig. 3**) nasce per svolgere principalmente il monitoraggio della presenza della tensione di rete a 230 Vca su una linea elettrica e la commutazione di un carico o di un ingresso di allarme o d'altro genere; allo scopo dispone di quattro morsetti a vite, due dei quali siglati FN ~ destinati a connettere la linea da monitorare e i due restanti che riportano il contatto pulito di un relé che permette di gestire un utilizzatore alimentato dalla rete elettrica di distribuzione a 230 Vca o altro. I contatti disponibili sono C e NA (sono resi accessibili dall'esterno tramite i morsetti con il simbolo del contatto) ed il relé può essere comandato a stato stabile tramite l'App da smartphone.

Quando viene a mancare la linea elettrica su F ed N viene generato un avviso diretto all'App dello smartphone associato; il messaggio può essere inviato perché il dispositivo ha una batteria al litio di backup da 300 mAh. Il Power Control Box si rivela quindi utile per avvisare se salta la corrente in casa o in altro luogo servito dalla rete elettrica a 230Vca, ad esempio per intervenire con tempestività se vi sono utilizzatori critici (per esempio elettromedicali) che funzionano con la rete.

Il contatto consente di commutare l'alimentazione di linee elettriche funzionanti a tensione di 230 Vca e potenza fino a 300 VA su carichi sia resistivi che misti; in assenza della tensione di rete il relé ricade e comunque non è più controllabile da remoto.

Le dimensioni del dispositivo sono contenute (appena 50x80x30 mm) e tali da consentirne l'introduzione in una scatola da incasso per impianti elettrici in formato 503 e semplificarne il cablaggio;

Fig. 3
Power
Control Box.



il peso è ridotto ad appena 80 grammi. Non richiede SIM tradizionale perché ha una SIM on-chip.

Smoke Detector

Questo dispositivo NB-IoT (lo vedete in **Fig. 4**) è sostanzialmente un sensore di fumo per ambienti civili e industriali, da collocare a soffitto per rilevare la presenza di fumo di vario genere e quindi l'insorgere di incendi o l'invasione di un locale da parte di fumi tossici.

È alimentato da una batteria interna al litio da 3 volt, tipo CR17450 che garantisce un'autonomia operativa di 18 mesi a 20 °C di temperatura ambiente e con una trasmissione al giorno; le sue dimensioni sono di 92 mm di diametro (infatti è tondo) x 35 mm. Il peso di è 100 grammi.

Motion IoT Sensor

Questo dispositivo è un sensore di movimento basato sulla combinazione della tecnica P.I.R. (Passive Infrared Radar) e del radar a microonde a 2,4 GHz; è connesso tramite rete cellulare e protocollo IoT ed è utilizzabile per rilevare presenza e spostamento di persone e animali, ma anche di automobili. Lo vedete in **Fig. 5**.

Ogni volta che rileva qualcosa nel suo campo di copertura fornisce un allarme trasmesso al Cloud IoTicontrollo e raggiunge l'App cui è associato. La combinazione di due tecnologie offre la massima sicurezza del rilevamento, perché con il dispositivo è possibile rilevare anche corpi che non emettono particolare calore, condizione, quest'ultima, determinante per il rilevamento tramite infrarossi passivi.

Il dispositivo è autoalimentato mediante due batterie al litio da 3,6V, che assicurano un'autonomia di 12 mesi a temperatura ambiente di 20 °C e con un massimo di un evento al giorno.

La temperatura di esercizio è tra -10 e +50°C e il tempo di prima attivazione fino a 5 minuti.

Le dimensioni sono 47x81x134 mm e il peso di 170 g.

Plug Meter

Questo dispositivo è sostanzialmente un Energy Meter connesso al Cloud di IoTicontrollo, inserito in una presa passante per la rete 230 Vca (**Fig. 6**) in grado di trasferire una potenza massima di 200 VA. Permette di controllare da remoto e comodamente tramite l'App, i consumi del carico collegato alla presa. L'installazione è semplice, perché basta staccare la spina dell'utilizzatore da monitorare e introdurla nel Plug Meter e quindi infilare quest'ultimo nella presa.



Fig. 4
Dispositivo Smoke Detector.

Il dispositivo funziona fra 10 e +50°C, ha un tempo di prima attivazione fino a 5 minuti (intervallo necessario ad aggancio rete e associazione all'App IoTicontrollo) misura 73x83x60 mm e pesa 160 grammi; come gli altri, funziona tramite una SIM on chip. È autoalimentato tramite la tensione di rete prelevata dalla connessione tra la spina e la presa passante, perché per forza di cose va installato in presenza di una rete elettrica.

Door Alarm

L'ultimo dispositivo dell'offerta di IoTicontrollo che andiamo ad analizzare in questo articolo è il



Fig. 5
Dispositivo Motion IoT Sensor.

Definita per la prima volta dal 3GPP (3rd Generation Partnership Project) nelle specifiche della release 13, la NB-IoT è la tecnologia di rete mobile più adatta per le applicazioni IoT che richiedono copertura geografica eccezionalmente vasta e impongono un consumo energetico estremamente ridotto. Queste applicazioni sono tipicamente caratterizzate da basse velocità di trasferimento dei dati e tempi di reazione (latenza) dell'ordine di qualche secondo, come lo Smart Metering, il controllo remoto di contenitori per la raccolta differenziata rifiuti, l'agricoltura smart, le Smart Bike e via di seguito. Le prime attivazioni delle reti sono iniziate alla fine del 2017 e l'avvio commerciale su larga scala in vari paesi nel mondo è iniziato nel corso del 2018. Un servizio NB-IoT può essere attivato all'interno di una portante LTE, nella Guard Band LTE o come soluzione autonoma. In futuro, la flessibilità delle unità radio 5G NR faciliterà anche il funzionamento dell' NB-IoT nelle bande assegnate al 5G. Sintetizzando, la tecnologia NB-IoT può essere definita come rete geografica LPWAN cellulare funzionante in direzione downlink con una larghezza di banda del canale di 180 kHz (un PRB) e una spaziatura tra le sottoportanti di 15 kHz, mentre in direzione uplink tipicamente con trasmissione a sottoportante singola da 3,75 kHz o 15 kHz. Per soddisfare il requisito di perdita massima di accoppiamento (MCL) che è 20 dB, una

rete NB-IoT prevede l'utilizzo di tecniche di miglioramento della copertura quali il power boosting in downlink o la ripetizione subframe sia in uplink sia in downlink. L'implementazione di dispositivi che utilizzano la comunicazione NB-IoT impone il dover affrontare una serie di problematiche. Una durata della batteria di oltre 10 anni richiede l'utilizzo di funzionalità specifiche per il risparmio energetico, quali PSM, eDRX o RAI (Release Assistance Indication). Inoltre, l'introduzione di reti LTE-M e NB-IoT richiede la definizione di un'architettura core ottimizzata che consenta, ad esempio, la trasmissione di dati degli utenti attraverso il piano di controllo e la trasmissione anticipata dei dati (EDT, Early Data Transmission). Una rete NB-IoT riutilizza ampiamente funzionalità e tecniche familiari già utilizzate nelle reti LTE, pertanto, la gran parte delle applicazioni di test è simile a quelle per effettuare le prove su un normale modem LTE. Tuttavia, le caratteristiche specifiche di un'applicazione LPWAN richiedono particolare attenzione: ad esempio, l'ottimizzazione e l'analisi dettagliata del consumo energetico in configurazioni e modelli operativi diversi sono altamente consigliate per assicurare una durata della batteria superiore a 10 anni. Il funzionamento in condizioni di copertura al chiuso richiede che le prestazioni della sezione RF e dell'antenna siano molto elevate.

	Rel. 12	Rel. 13	Rel. 14	Rel. 15	Rel. 16
Low power consumption	Power saving mode (PSM) of up to 413 days	Extended DRX (eDRX) of up to 175 min	Release assistance indicator (RAI) in AS	Wake-up signal (WUS) in idle mode	WUS for groups of UEs
Low power and low latency and efficiency		Optimized EPS architecture option for CIoT/MIoT		Early data transmission (EDT) for UL	Early data transmission (EDT) for DL
NB-IoT based on NB-LTE and NB-CIoT and further improvements		Cat NB1 (FDD) - 1 HARQ - 1000 bit TBS	Cat NB2 (FDD) - 2 HARQs - 2536 bit TBS	TDD support	5G NR coexistence
		Power classes - class 5: 23 dBm - class 3: 20 dBm	Power class - class 6: 14 dBm	Small cell support	Improved scheduling
Coverage enhancements		Coverage enhancement level CE level 0/1/2		NPRACH range enhancements	

modulo per porte, ossia il Door Alarm (**Fig. 7**): è un dispositivo da applicare a porte e finestre che rileva l'apertura e la segnala inviando un messaggio, attraverso il Cloud, all'App. Possiamo considerarlo un allarme per porte (o anche finestre) integrato

e connesso in ottica IoT. Per il funzionamento, il dispositivo va fissato all'anta di cui rilevare l'apertura, tramite l'apposita staffa fornita in dotazione; l'apertura viene rilevata grazie a un sensore di movimento (accelerometro MEMS) e non tramite relé

Reed come spesso avviene nei sensori per porte e finestre. L'alimentazione del modulo è affidata a una batteria LiPo interna, quindi ricaricabile, che fornisce un'autonomia di 12 mesi a temperatura ambiente di 20 °C e con un massimo di un evento al giorno. Il dispositivo funziona a temperatura ambiente compresa fra -10 e +50°C, è destinato ad un uso all'interno ed ha un tempo di prima attivazione di massimo 5 minuti. Le dimensioni sono di 47x70x19 mm e il peso di 60 g.

LA SIM

I prodotti di IoTicontrollo sono tipicamente forniti di apposita SIM, in versione plastica oppure on-chip, in base alla tipologia di impiego, condizioni ed esigenze operative. La gran parte dei dispositivi integra una SIM on-chip, vale a dire che ha memorizzati i dati che si troverebbero nella tradizionale SIM telefonica (che tecnicamente parlando è una Smart Card) inseriti in fabbrica, durante la registrazione del singolo dispositivo. La SIM virtuale o SIM On Chip è una tecnologia che permette di registrare alla rete telefonica dell'operatore di riferimento un determinato apparato GSM/UMTS/4G senza dover necessariamente avere e far leggere una SIM-Card; ha il grande pregio di poter supportare più abbonamenti su un unico apparato senza dover commutare alcuna SIM. Nel caso di IoTicontrollo l'operatore è fisso.

La connessione radiomobile dei dispositivi IoTicontrollo è teoricamente a traffico illimitato, nel senso che non bisogna curarsi di ricaricarla né di sottoscrivere alcun abbonamento, giacché viene gestita dall'azienda IoTicontrollo; del resto il traffico dati nei dispositivi IoT è irrisorio, se paragonato ai piani dati minimi che qualsiasi operatore offrirebbe. Attualmente l'operatore cui IoTicontrollo si appoggia è 1NCE (o Vodafone), specializzato in connessioni dati dedicate all'IoT. La IoT SIM Chip Industrial viene fornita da 1NCE e integrata nei dispositivi di IoTicontrollo (è associata all'offerta Flat Rate 1NCE IoT). La SIM è di tipo dati e non ha associato un numero telefonico che è possibile chiamare. Il piano dati IoT Flat Rate offre connettività in oltre 100 paesi del mondo per un traffico complessivo di 500 Mbyte.

LA APP PER SMARTPHONE

La App fornita a corredo si chiama IOTICONTROLLO, è disponibile su PlayStore di Android o App Store di Apple ed è compatibile con i sistemi operativi Android dal 5.1.1 in poi e iOS versione 9 o superiore. Eseguite l'applicazione e seguite il tutorial per



Fig. 6
Dispositivo Plug
Meter.

registrarvi (vi verrà richiesto di inserire un indirizzo e-mail valido, al quale verrà inviato un link su cui cliccare per confermare la registrazione).

Una volta installata, l'App propone una schermata base alla quale verranno aggiunte tante sezioni quanti saranno i dispositivi abbinati (la **Fig. 8** propone un esempio con installati il Motion IoT Sensor e il Security Box); l'abbinamento di ciascun dispositivo si esegue, in base alla tipologia, premendo il pulsantino di cui è dotato o sollecitando un evento (ad esempio muovendolo).

Per esempio per attivare (quindi registrare al Cloud di IoTicontrollo) e associare all'App un Door Alarm basta premere il pulsante sul suo corpo ed entro un tempo massimo di pochi minuti sullo schermo dello smartphone apparirà la relativa sezione con il nome del dispositivo. Ogni sezione riporta anche delle impostazioni specifiche per il sensore correlato: per esempio la sensibilità. Oppure degli indicatori (Plug Meter). Sempre, in alto a sinistra



Fig. 7
Dispositivo Door
Alarm.

Fig. 8
Schermata dell'App
IoTiccontrollo con in-
stallati due device.

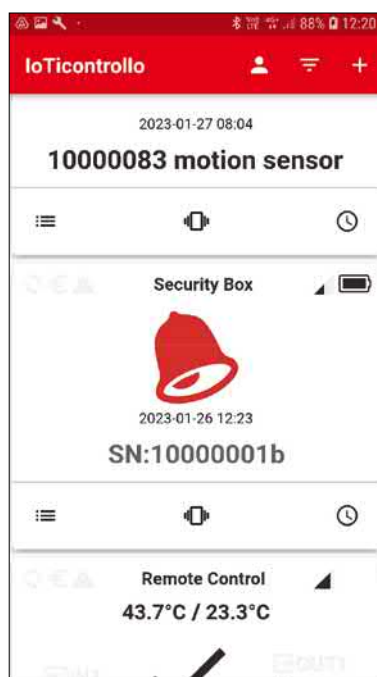


Fig. 9
Schermata
dell'App
IoTiccontrollo
relativa
a sensibilità
dei sensori
ed eventi.

è presente l'icona di menu che consente di aprire altre schermate specifiche come, appunto, quella della sensibilità del sensore o degli eventi rilevati (Fig. 9).

Altro esempio: per attivare (quindi registrare al Cloud di IoTiccontrollo) e associare all'App un Security IOT box basta premere il pulsante sul suo corpo ed entro pochi minuti sullo schermo dello

smartphone apparirà la prima comunicazione nella relativa sezione con il nome del dispositivo.

Riepilogando, i passaggi da eseguire per l'associazione di un dispositivo, ad App installata, sono: premere il pulsante +, allorché si attiverà la telecamera per inquadrare il codice QR riportato sul dispositivo; acquisito il codice QR, vi apparirà la schermata di conferma che il dispositivo è stato riconosciuto. A questo punto occorre premere su "Continua" e nella schermata principale comparirà il dispositivo associato in attesa della prima comunicazione; questa avverrà premendo il pulsantino sul dispositivo IoTiccontrollo desiderato.

Alla prima alimentazione o dopo la sostituzione delle pile o batterie (ovvero all'inserzione nella presa di corrente, per il Plug Meter) il dispositivo inizia la connessione; bisogna quindi attendere il termine della connessione, evidenziato dal fatto che il LED si spegnerà.

A questo punto occorre premere il pulsante e il LED si accenderà; il dispositivo IoT eseguirà la prima trasmissione di attivazione: sull'app comparirà l'icona dell'avvenuta attivazione del dispositivo.

Ora il dispositivo è pronto per essere utilizzato. Nell'app sono visualizzate altre informazioni come l'intensità del segnale e la carica della batteria.

È possibile modificare il nome del dispositivo, premendo sulla scritta per modificarlo.

Periodicamente il dispositivo invierà una trasmissione di sicurezza per far capire che è operativo (sullo schermo, nella relativa sezione, apparirà l'icona con la spunta): questa non determinerà alcun evento locale o segnalazione, ma verrà gestita in modo trasparente all'utente.

In caso di assenza prolungata di trasmissioni, l'utente riceverà una notifica che segnala l'anomalia. Bene, con questo concludiamo la carrellata sui prodotti IoTiccontrollo, nati per portare i vantaggi dell'IoT nella vita di tutti i giorni. □

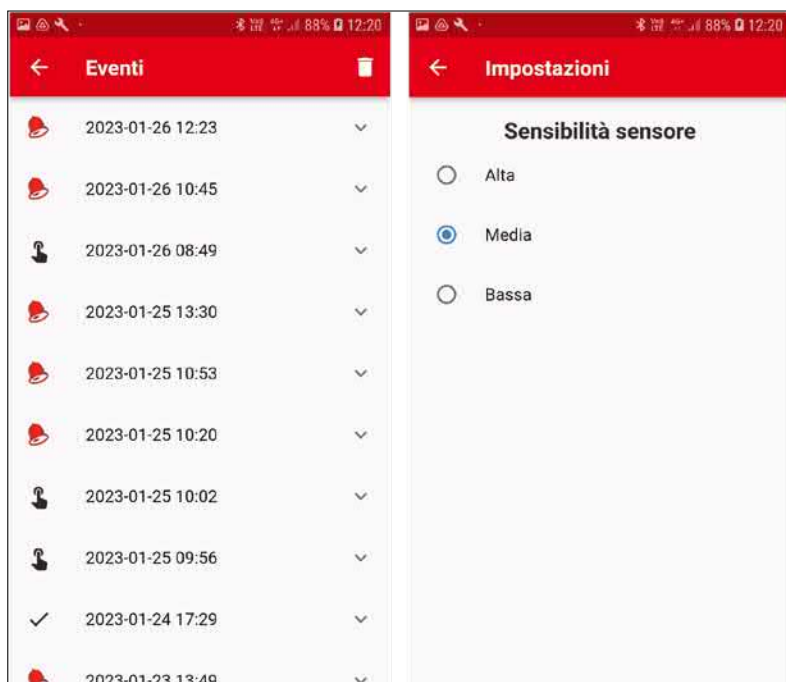


Cosa occorre?

I dispositivi IoTiccontrollo sono distribuiti da Futura Elettronica. Attualmente quelli disponibili sono il Security IoT Box (cod. SEC3NV000) in vendita a Euro 119,00, il Power Control (cod. PWC3NV000) a Euro 119,00 e lo Smoke Detector (cod. SMK3NV000) a Euro 149,00. 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**





IMPARIAMO A UTILIZZARE **EAGLE**

di VINCENZO GERMANO e DAVIDE MADDALENA

1

Cominciamo il nostro viaggio alla scoperta di un potente strumento di progettazione elettronica, che permette di coprire l'intero ciclo di sviluppo di una scheda elettronica. Prima Puntata.

Oggigiorno, con la grande diffusione dell'Open Hardware e di Arduino, sempre più spesso capita di realizzare interi progetti su breadboard o su millefori; ma una volta ottenuto un prototipo funzionante bisogna creare un PCB (Printed Circuit Board - circuito stampato). Per poter ottenere i file necessari alla realizzazione di un PCB a livello industriale serve uno strumento di ausilio allo sviluppo del progetto che aiuti in questo percorso, per l'appunto un CAD (Computer-Aided Design). Il livello di complessità del PCB si

rispecchia anche nella scelta dello strumento CAD da utilizzare, che può essere più o meno complesso e sofisticato. Ad oggi esistono molte soluzioni software nel mondo Open Source: due esempi sono KiCad e TinyCAD, che offrono svariate funzionalità, e che vogliono competere con tool o suite professionali, realizzati da aziende che sviluppano software CAD totalmente orientati alla progettazione di schede elettroniche e che generalmente vengono indicati come ECAD (Electronic CAD) o EDA (Electronic Design Automation).

Alcuni esempi di aziende specializzate nella produzione di questa categoria di software sono ad esempio Altium (produttrice di Altium Designer), Cadence (produttrice di OrCAD) e Mentor Graphics. Oltre a questi tool estremamente professionali ma anche molto costosi (le licenze hanno costi dell'ordine di migliaia di euro) ne esiste uno in particolare che, tra le altre caratteristiche, permette di avere flessibilità nella scelta delle licenze (da Freeware, mensile, annuale fino ad una licenza triennale), di affrontare senza la progettazione elettronica schede complesse e permette di avere oltre alle funzionalità già offerte, espandibilità delle stesse o aggiunta di altre, mediante "script". Questo tool è Autodesk EAGLE (acronimo di Easily Applicable Graphical Layout Editor), che si dimostra perfettamente adeguato sia alle esigenze dell'hobbista che vuole cimentarsi nella progettazione elettronica per uso privato, che dell'azienda con la necessità di progettare i propri prodotti o per eventuali schede interne. Dal Giugno 2016, a partire dalla versione 8.0.0, EAGLE viene acquisito da Autodesk, che opera un sostanziale cambiamento del software fino al giugno 2020, quando non viene più proposto come prodotto a sè stante (standalone) ed entra a far parte della suite Autodesk Fusion 360. L'ultima versione standalone di EAGLE è la 9.6.2, oggetto di questo corso.

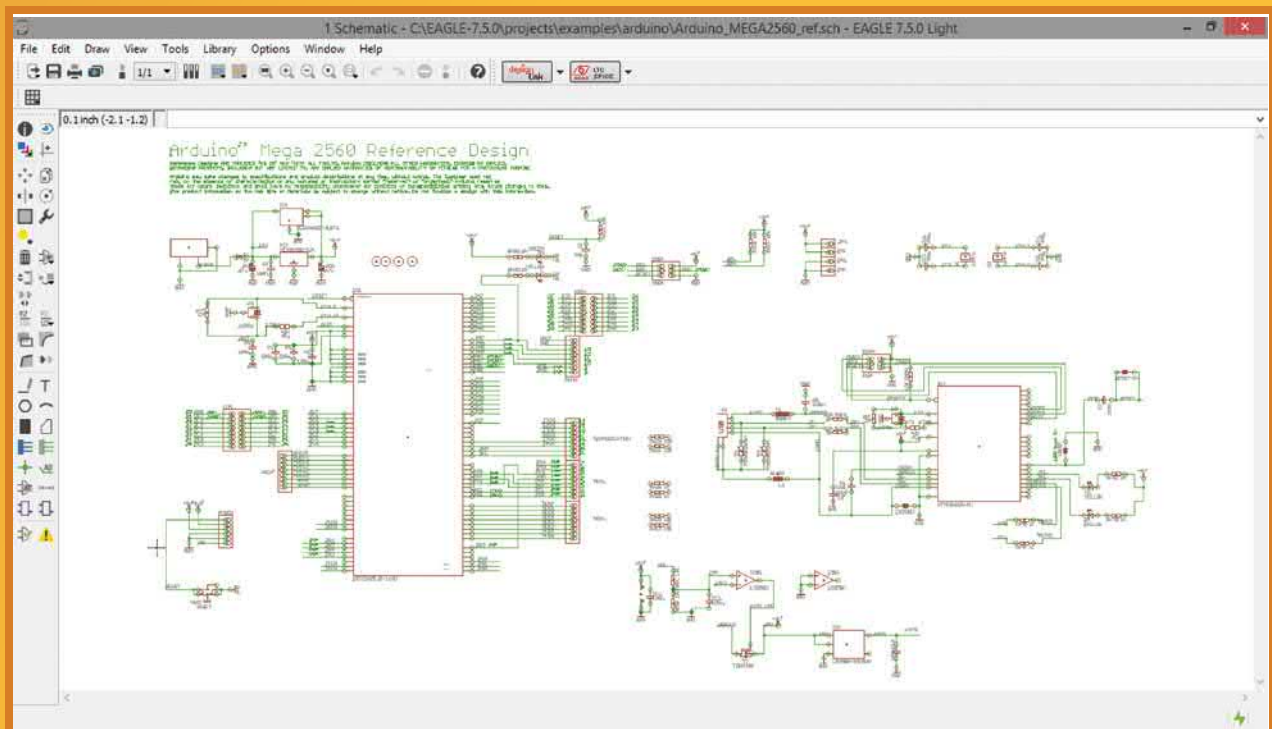
Quest'ottimo strumento si è sempre più affermato nell'ambito dell'Open Hardware, anche trainato dal "fenomeno" Arduino.

Infatti, quanto appena affermato è confermato e riscontrabile, semplicemente notando che gli schemi elettrici dei PCB ufficiali delle schede Arduino vengono rilasciati come file EAGLE. Anche rinomati siti web che realizzano e vendono prodotti DIY (Do It Yourself - fai da te), rilasciano i file dei loro progetti (librerie e schematici) nello stesso formato ECAD, prevedendo e rendendo disponibili intere librerie EAGLE dedicate. Tutto ciò ci fa capire la diffusione, compatibilità e usabilità di EAGLE.

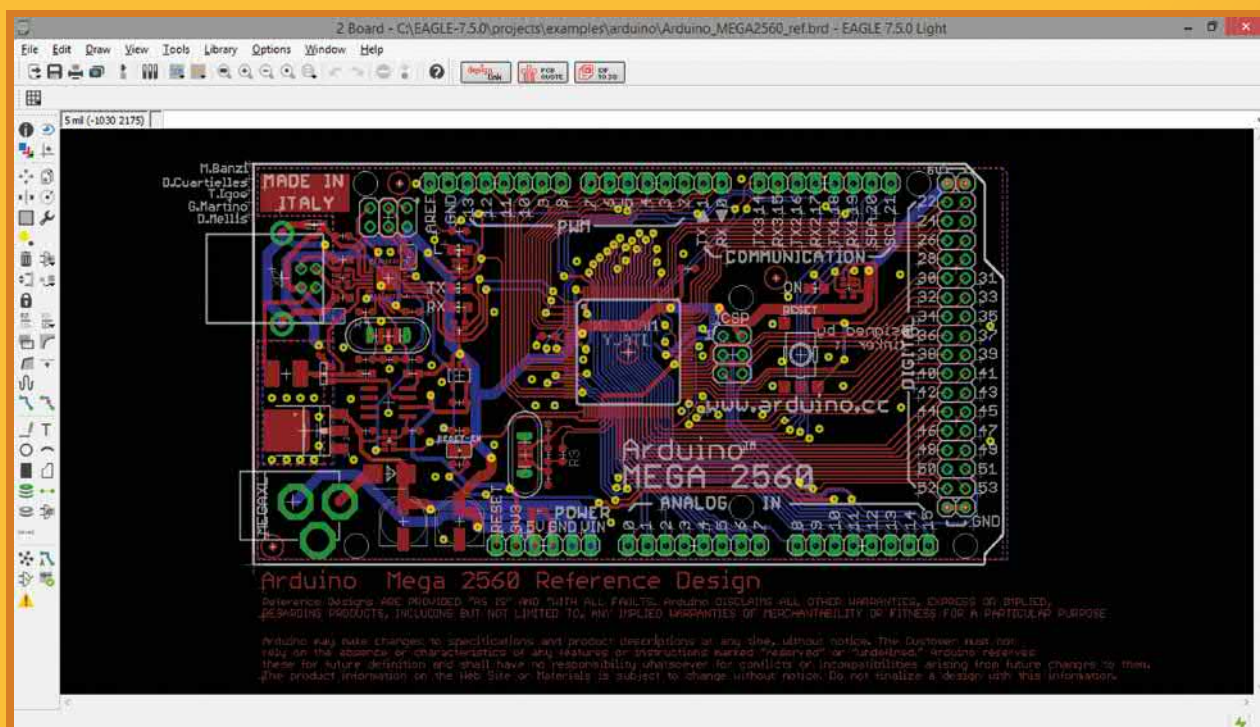
Perciò, come si è ben compreso, questa e le prossime puntate del corso, si pongono come obiettivo quello di avvicinare gli utenti al tool Autodesk EAGLE, presentando e spiegando l'intero processo di sviluppo di un circuito stampato, utilizzando gli strumenti messi a disposizione dal tool; partendo dall'idea del circuito elettronico fino ad arrivare alla generazione dei file Gerber per la realizzazione dei PCB. Il tutto grazie all'ausilio della progettazione di una scheda orientata all'Internet of Things (IoT), che approfondiremo e impareremo a conoscere durante il corso, in perfetto stile "Open Hardware".

INTRODUZIONE AD EAGLE

EAGLE è un tool EDA pensato per il disegno elettronico, identificabile anche come IDE (Integrated Development Enviroment), cioè costituito da una serie di tool software racchiusi all'interno di un "Control Panel" (pannello di controllo – abbreviato da ora in poi come CP), che permette la



↑ Fig. 1 Esempio di schematico di un Arduino Mega realizzato con Schematic.



↑ Fig. 2 Esempio di layout realizzato con Board di una Arduino Mega.

gestione dei file di progetto e l'avvio dei singoli programmi da un'unica interfaccia centralizzata. All'interno del CP, i due principali programmi che è possibile lanciare e su cui ci concentreremo durante il corso sono:

- **Schematic:** l'editor degli schemi elettrici, che integra l'editor per le librerie dei componenti; ne viene riportata in Fig. 1 un'immagine d'esempio in cui è presente uno schema elettrico di una Arduino Mega.
- **Board:** l'editor dei circuiti stampati, che gestisce fino a 16 layer (strati), integra un tool per la generazione dei gerber e un editor per le librerie dei footprint; in Fig. 2 è riportata un'immagine del layout di una Arduino Mega.

EAGLE è impostato in modo da agevolare l'utilizzo di un certo ciclo di sviluppo, che parte con la realizzazione dello schema e termina con la generazione dei file Gerber, perciò l'accesso ad uno step successivo avviene solo dopo il completamento di quello precedente, in modo da minimizzare il più possibile la generazione di errori durante il progetto (onerosi da individuare/recuperare nelle fasi finali). Tuttavia, rispetto ad altri ECAD, è meno vincolante riguardo ai vari passaggi da effettuare, perché una volta generata la prima versione di board, questa viene legata direttamente agli schemi elettrici e da questo momento in poi EAGLE mantiene aperte le due finestre, in modo tale da riversare in tempo reale ogni cambiamento che viene effettuato sugli schemi anche sulla board e viceversa (senza passare da

alcun file di "netlist" come avviene per altri tool). Ripercorrendo brevemente i vari step di progetto, possiamo affermare che il primo passo dello sviluppo di un circuito stampato (dopo la creazione del progetto stesso) è il disegno dello schema elettrico; quindi il punto di partenza è, senza ombra di dubbio, una libreria di componenti esistente o la creazione dei componenti necessari da centralizzare in una libreria. Considerate che la creazione di componenti in Schematic non necessita per forza della definizione contestuale del footprint, ma sicuramente può essere una buona norma definirli in una fase iniziale come vedremo nelle prossime puntate del corso. In ogni caso una volta definito un file di libreria, con estensione "*.lbr" come riportato nella Tabella 1, abbiamo tutto il necessario per procedere con lo schema elettrico. Questo processo produce il file con estensione "*.sch" (Tabella 1), che costituisce il file di lavoro di Schematic. A questo punto, tale file viene preso come input da Board, che lo utilizza come punto di partenza per la generazione del file con estensione "*.brd", ossia quello di lavoro di Board stesso. Da questo momento in poi si può procedere con il piazzamento componenti e lo sbroglio del PCB, che produrrà come output tutti i file necessari alla fabbricazione della scheda stessa, compresi i Gerber. Il passo finale di tutto questo processo è l'ispezione dei file di output generati, utilizzando un "Gerber View". Tale step, sebbene opzionale, è di fondamentale importanza per assicurare che non ci siano stati errori durante le fasi

TYPE	WINDOW	NAME
Board	Layout Editor	*.brd
Schematic	Schematic Editor	*.sch
Library	Library Editor	*.lbr
Script File	Text Editor	*.scr
User Language Program	Text Editor	*.ulp
Any text file	Text Editor	*.*

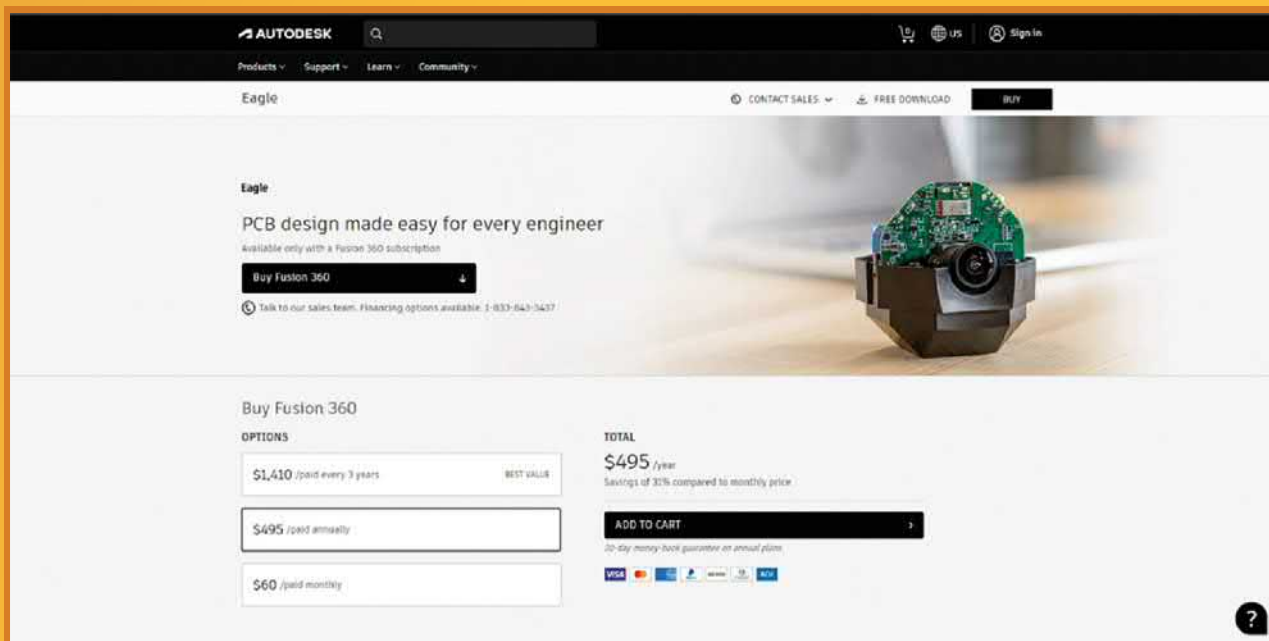
↑ **Tabella 1** Estensioni dei file

precedenti e che potrebbero portare alla produzione di un pezzo difettoso o non conforme a quanto atteso. EAGLE, oltre all'esportazione dei file Gerber offre inoltre la possibilità di esportare un modello tridimensionale della scheda; in questo modo si può disporre, oltre ad una vista 3D dello stampato (rendering), di una "matematica" utile, per esempio, a verificare eventuali ingombri nel creare una meccanica che contiene lo stampato. Per completezza, nella **Tabella 1** vengono anche riportate le due estensioni "*.scr" e "*.ulp", i cui file non sono altro che eseguibili con funzionalità avanzate, ovvero file contenenti delle istruzioni in un particolare linguaggio di programmazione utilizzato da EAGLE, che tuttavia non saranno analizzati durante il corso in quanto appartenenti ad un uso esperto del programma. Chiaramente alcuni passi appena visti possono subire delle alterazioni derivanti da errori che vengono corretti tornando allo step precedente, effettuando le modifiche

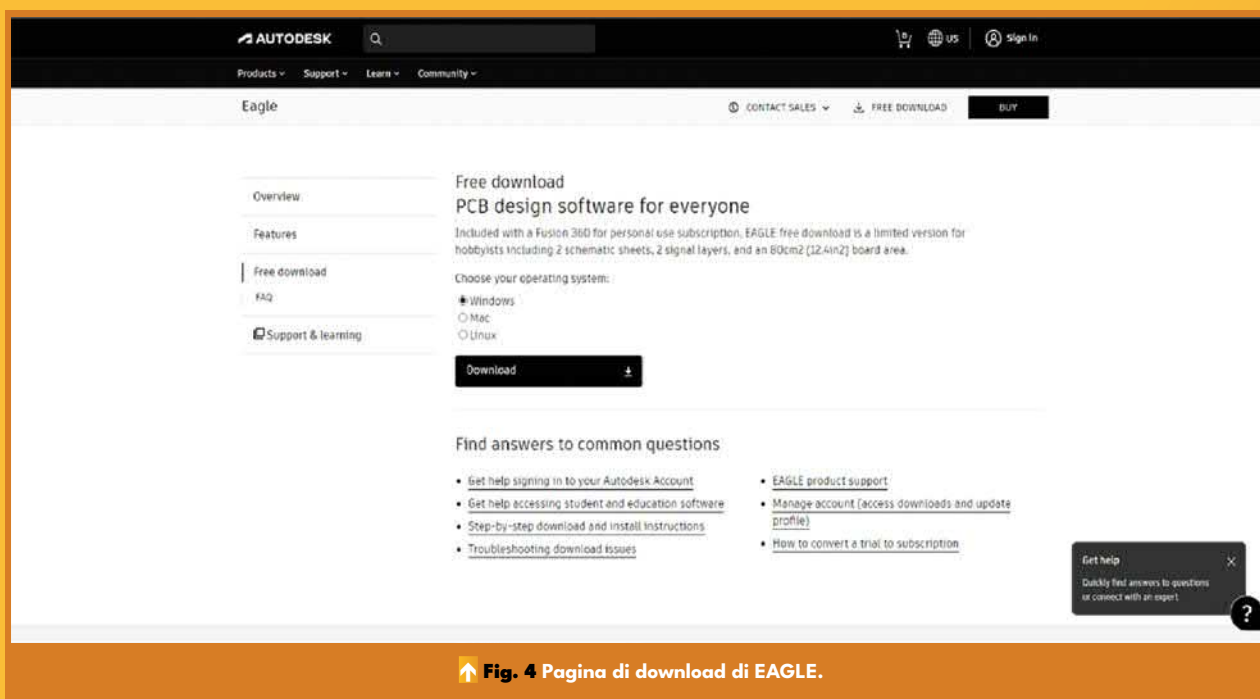
necessarie e rigenerando i file di ingresso corretti per lo step successivo. Questo processo, generalmente viene denominato backward annotation e permette di mantenere un'assoluta coerenza tra tutti i passi di progetto, in maniera che non si possano verificare situazioni di inconsistenza tra le varie fasi della progettazione, come ad esempio tra lo sviluppo dello schematico e lo sbroglio del PCB.

LE TIPOLOGIE DI LICENZE

A questo punto, prima di intraprendere il nostro "viaggio" alla scoperta di questa suite, è necessario precisare che EAGLE non è un software propriamente "freeware" ma è un prodotto commerciale, che può essere utilizzato nella versione freeware (non commerciale) solo per fini di studio e/o valutazione, quindi non per scopi di lucro. Questo significa che nel caso in cui si voglia utilizzare il tool per la realizzazione di progetti per i quali si percepisce una qualsiasi forma di vantaggio economico, sia esso diretto e/o indiretto (come indicato nelle sue condizioni di utilizzo) è necessario comprare una licenza adeguata alle proprie esigenze. Per una breve panoramica riguardo la tipologia delle licenze, fate riferimento alla **Fig. 3**, che riguarda la pagina web del Licensing, proponente le caratteristiche delle licenze acquistabili, che, come anticipato, non saranno direttamente come programma EAGLE ma come Fusion 360. La licenza libera (EAGLE free download) è limitata all'uso dilettantistico; le licenze commerciali si distinguono solo per la durata di validità (1 mese, 1 anno, 3 anni) e non hanno limitazioni di progetto.



↑ **Fig. 3** Tipologie di licenze acquistabili di Fusion 360.



↑ Fig. 4 Pagina di download di EAGLE.

La versione che verrà analizzata durante questo corso è la Freeware, ovvero l'ultima versione EAGLE disponibile e utilizzabile gratuitamente per applicazioni come ad esempio progetti personali. Nonostante le sue limitazioni, permette di realizzare circuiti di una discreta complessità, a patto che vengano rispettati i seguenti vincoli:

- dimensioni del PCB non superiori a 100 x 80 mm;
- PCB con al massimo due layer, quindi niente PCB multistrato ma soltanto "doppia faccia";
- possibilità di creare con lo schema solo due "fogli" per ogni progetto.

EAGLE è disponibile in versione 64 bit ammesso che vengano rispecchiati i seguenti requisiti minimi di sistema:

- almeno 3 MB di memoria libera;
- circa 700 MB di spazio libero su disco;
- minima risoluzione grafica 1024 x 768 pixel;
- un mouse con almeno 3 pulsanti.

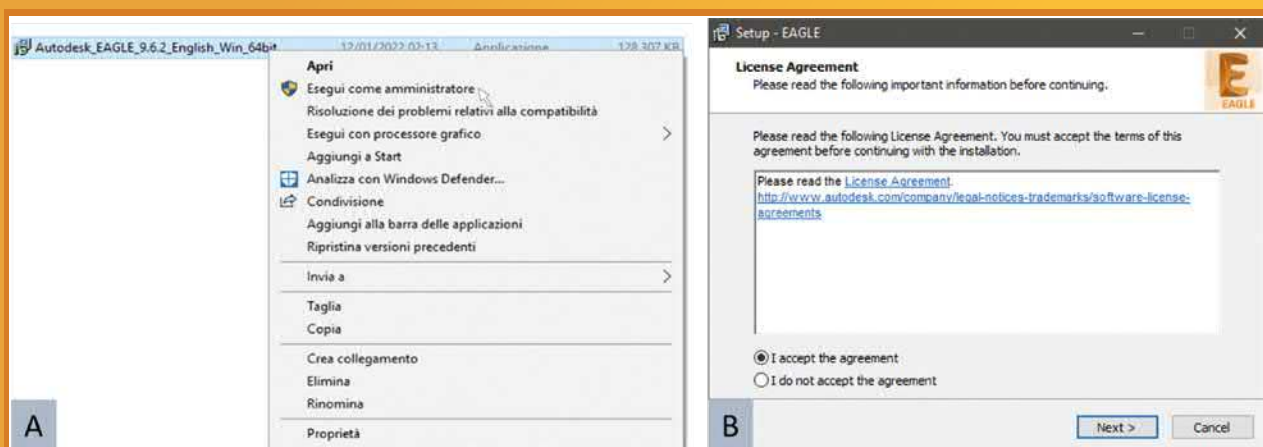
Quindi per riassumere brevemente, mediante EAGLE è possibile realizzare schemi elettrici, distinte componenti o comunemente chiamate BOM (Bill of Materials) e PCB; in aggiunta all'interno di EAGLE è possibile utilizzare un gestore integrato per le librerie dei componenti, completo di editor sia per gli schematici che per i footprint. Oltre a quanto offerto direttamente dal tool vi è la possibilità di espandere e potenziare tutte le sue funzionalità mediante l'ausilio di "script" scaricabili direttamente dal sito ufficiale, oppure generandone di nuovi mediante il linguaggio di programmazione ULP (User Language Programs).

DOWNLOAD E INSTALLAZIONE

Passiamo alla preparazione dell'ambiente di lavoro, descrivendo il download e l'installazione di EAGLE; si può notare che rispetto ad alcuni tool concorrenti è disponibile una versione adatta a tutti i principali sistemi operativi: Windows, Mac e Linux; anche se durante questo corso faremo riferimento solo alla versione per Windows, vale quanto esposto anche per tutte le altre.

EAGLE è scaricabile dal sito ufficiale: www.autodesk.com/products/eagle/free-download. Come mostrato nella Fig. 4, nell'apposita sezione "Download" del sito sono presenti le ultime versioni rilasciate per tutti i sistemi operativi; alla stesura di questo corso, l'ultima versione scaricabile era la 9.6.2, perciò faremo riferimento a tale versione.

A questo punto non resta che selezionare la versione per il proprio sistema operativo e cliccare sul pulsante "Download". Il download partirà in automatico ed a seconda delle impostazioni del browser, scaricherà la versione corrispondente (ad esempio: *Autodesk_EAGLE_9.6.2_English_Win_64bit.exe*). Una volta completato il download possiamo lanciare l'eseguibile, in modo da avviare la procedura di installazione che vediamo insieme nei prossimi passi. Avviate il programma di installazione con i diritti di amministratore cliccando col tasto destro del mouse sull'icona e selezionando "Esegui come amministratore" (Fig. 5A). Nella schermata successiva, riportata in Fig. 5B selezionate "I accept the agreement" e cliccare sul pulsante "Next". Selezionate il percorso dove installare il software (ad esempio "C:\Program Files\EAGLE 9.6.2") e cliccate su "Next".



↑ Fig. 5 Avvio e accettazione termini licenza.

Selezionate se volete creare un collegamento del software su desktop e cliccate su "Next" (Fig. 6).

A questo punto è tutto pronto: si può lanciare l'installazione cliccando su "Install"; al termine del processo d'installazione si potrà scegliere se avviare direttamente EAGLE (lasciare la spunta su "Launch EAGLE") oppure no, quindi cliccare "Finish" (Fig. 7). Con i passi appena elencati si è completata l'installazione di EAGLE. Come si può notare in nessuno dei passi precedenti è stata richiesta un'opzione di utilizzo licenza, questo perché come anticipato, Autodesk EAGLE viene scaricato già come "free" (gratuito) per usi non commerciali. Tuttavia al primo avvio del software partirà la procedura di log-in o registrazione utente.

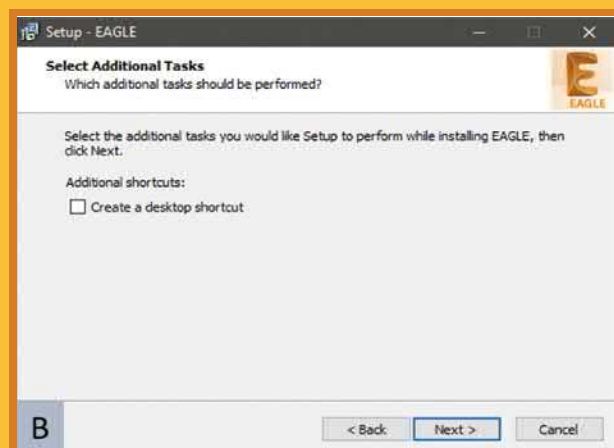
LOGIN UTENTE

Per poter essere utilizzato EAGLE richiede che l'utente disponga di un "account" Autodesk ovvero di essere registrato e successivamente essere "loggato" (almeno la prima volta). La creazione e l'utilizzo dell'account sono gratuiti. Vediamo insieme i passi necessari per il primo avvio e la creazione dell'account. Nel caso disponiate già di un account Autodesk dovete inserire l'indirizzo e-mail usato per la registrazione (Fig. 8A) e cliccare su "AVANTI" e saltare tale procedura effettuando direttamente il login. Altrimenti dovete cliccare su "CREA UN ACCOUNT" e, nel form on-line che appare, inserire nome, cognome, indirizzo e-mail (da confermare ulteriormente) e password desiderata (usare almeno 8 caratteri dei quali almeno un carattere maiuscolo, un carattere speciale ed almeno un numero) per la registrazione. Spuntate inoltre "Accetto le Condizioni d'uso di Autodesk e l'Informativa sulla privacy" quindi cliccate su "CREA ACCOUNT" (Fig. 8B). Verrà inviata una e-mail con richiesta verifica all'indirizzo e-mail usato per la registrazione; una volta confermato l'account cliccate su "CONTINUA" (Fig. 9A). A questo punto tornate alla pagina di login, ma con un'utenza Autodesk verificata, dunque inserite lo username e cliccate

su "AVANTI" (Fig. 9B). Infine, inserite la password e cliccate su "ACCEDE". Ora che abbiamo completato l'installazione del nostro ambiente di progettazione, possiamo lanciarlo e cominciare ad analizzare la sua interfaccia.

IL "CONTROL PANEL" DI EAGLE

Come ampiamente anticipato EAGLE è un IDE, perciò dispone di una serie di tool integrati all'interno di una singola interfaccia utente, in questo caso nota come "Control Panel", che appare all'avvio del programma ed è il centro di controllo dell'intero programma, ovvero tutti i file specifici vengono gestiti al suo interno e in aggiunta si possono configurare le impostazioni di base, centralizzando il tutto in un'unica interfaccia. Si presenta simile ai "File Manager" utilizzati da un'ampia varietà di applicazioni e sistemi operativi in cui ogni file viene visualizzato in una "vista ad albero" per mezzo di una piccola icona, come è possibile riscontrare in Fig. 10.



↑ Fig. 6 Percorso d'installazione e shortcut





Fig. 7 Fine installazione e avvio programma.

Nella parte destra (information window) è presente la visualizzazione delle informazioni riguardo i file selezionati, quindi se un oggetto è selezionato vengono mostrate altre informazioni come l'anteprima di uno schematico o un componente, come vedremo nelle puntate successive di questo corso. È sufficiente fare clic su vari file/cartelle in modo da sperimentare la facilità e le potenzialità del CP. Nella parte sinistra (tree view window) è presente la vista ad albero mediante la quale è possibile navigare all'interno dei file di progetto, offrendo una rapida panoramica delle librerie, della documentazione, delle regole di progettazione, ULP, file di script, lavori CAM e progetti (tutti questi file possono far parte di un progetto così come schematici e layout).

Nella struttura ad albero, un menu di scelta rapida si apre cliccando con il mouse su una voce desiderata; ciò consente a seconda dell'oggetto selezionato, di svolgere una serie di azioni, come ad esempio rinominare, copiare, stampare

ma anche aprire file PDF (che sarà aperto con l'applicazione predefinita).

Nella parte alta (barra dei menu) sono presenti i classici menu a tendina di Windows, dai quali si possono operare le impostazioni della suite EAGLE (Fig. 11). Infine nella parte bassa (barra di stato) vengono mostrati tutti i feedback di programma.

Il Control Panel supporta anche la funzione del Drag&Drop, che può essere utilizzata anche tra diversi programmi, permettendo ad esempio di copiare i file, spostare, o creare collegamenti, semplicemente trascinandoli.

Facciamo una breve carrellata sulle cartelle visibili all'interno del Control Panel:

- **Documentation:** consente l'accesso diretto ai tutorial EAGLE e ai manuali, disponibili in diverse lingue, inoltre si possono trovare i file UPDATE/README e file di documentazione di alcuni ULP.
- **Libraries:** permette di visualizzare il contenuto delle librerie, fornendo una rapida panoramica della componentistica disponibile. Nel campo Description è possibile vedere una breve descrizione del contenuto e selezionandola si possono visualizzare maggiori informazioni nella parte destra del Control Panel ma approfondiremo l'argomento nelle prossime puntate.
- **Design Blocks:** consistono in parti di schemi, layout o entrambi che possono essere usati, per esempio, come riferimento alla progettazione o sotto parti di circuito da riutilizzare nei progetti.
- **Design Rules:** possono essere definite in EAGLE delle speciali regole di progettazione (Design Rules) per sbrogliare le schede elettroniche, tali regole possono essere salvate come set di dati in file speciali con estensione "*.dru". Perciò questa cartella contiene tutte le configurazioni predefinite e quelle create dall'utente, che possono essere richiamate e utilizzate all'occorrenza.

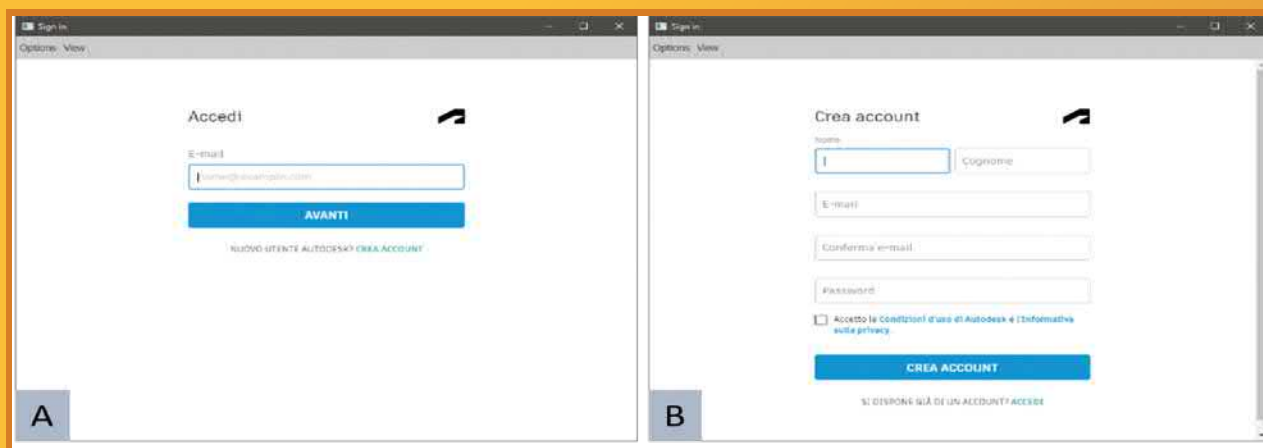


Fig. 8 Login e registrazione

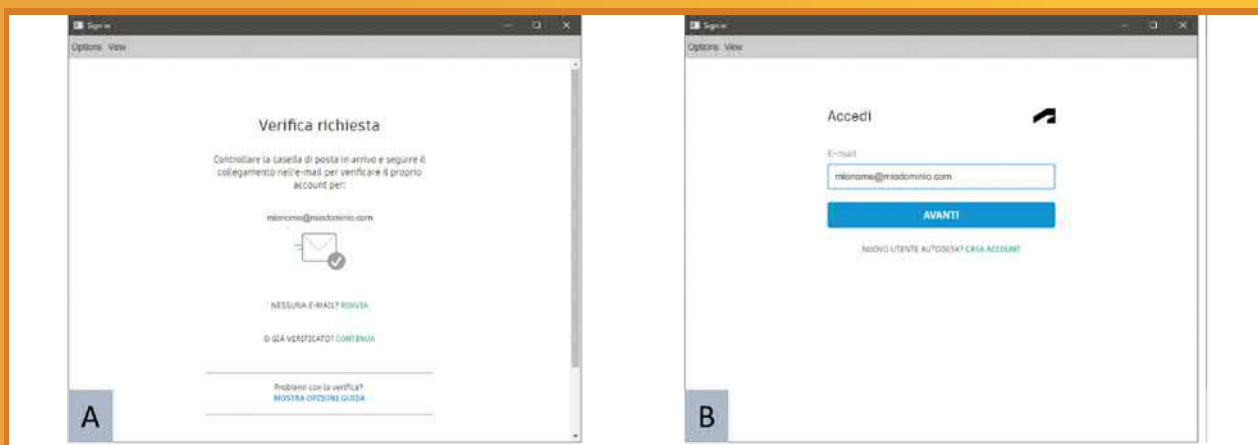


Fig. 9 Verifica e-mail e ultime fasi di accesso

za; se non viene selezionata nessuna configurazione in particolare, EAGLE carica quella predefinita.

- **User Language Programs, Script and CAM Jobs:** consentono di visualizzare il contenuto delle cartelle ULP, SCR e CAM, che contengono diversi programmi per l'utente (*.ulp), file di script (*.scr) e file CAM (*.cam). Tutti questi file estendono le funzionalità di EAGLE ed è possibile scaricarne di nuovi da internet o costruirsi di propri, dipendentemente dalla necessità. In ogni caso, come già anticipato, selezionando i singoli file se ne può vederne la descrizione. Per maggiori dettagli si faccia riferimento al sito ufficiale di EAGLE o

ai vari manuali utente.

- **SPICE Models:** da alcune edizioni a questa parte la suite EAGLE si interfaccia con LTSPICE per la simulazione di reti elettriche/elettroniche via modelli SPICE.
- **Projects:** permette di navigare all'interno dei progetti e aprire/gestire sia i file degli schematici che delle board.

Navigando all'interno di tali cartelle ci si può dedurre che l'installazione di EAGLE è già ben corredata di svariato materiale, da manuali a file di esempio o anche a script e file ULP per poter espanderne le funzionalità ma l'aspetto più lampante è l'enorme libreria messa a disposizione.

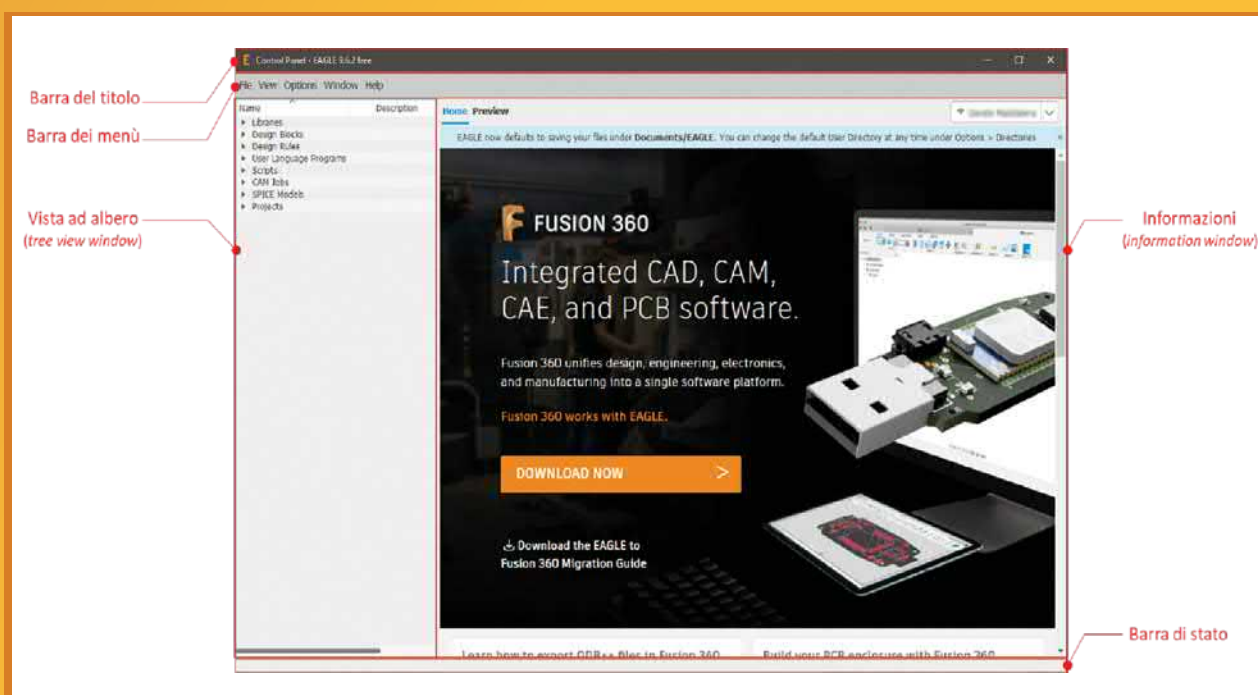


Fig. 10 Il Contol Panel di EAGLE



I COMPONENTI IN EAGLE

Il primo passo da affrontare per la realizzazione di un nuovo progetto è gettare le basi per poter lavorare sia su uno schema che su un layout.

Generalizzando, se vediamo un PCB come una serie di oggetti montati su delle piazzole connesse tra loro ci rendiamo conto che i componenti o le piazzole hanno un certo ingombro ed occupano una posizione ben determinata dunque ogni componente ha bisogno di essere disegnato a seconda della tipologia, caratteristiche meccaniche, elettriche, e via discorrendo. Oltretutto le connessioni non sono eseguite a caso ma secondo un determinato schema.

Dunque gli stessi componenti necessitano di un modello semplificato che permetta sia la schematizzazione "su carta" che l'identificazione del componente.

Chiamiamo quest'ultimo Simbolo (Symbol) ed un suo corrispettivo modello su layout Impronta (Footprint). Sia simboli che footprint possono avere forme diverse a seconda della tipologia di componente (resistore, condensatore e altro), della tipologia di montaggio (foro passante, montaggio superficiale e così via) o della tecnologia.

In uno schema possiamo trovare lo stesso simbolo più di una volta ma con caratteristiche "meccaniche" diverse dunque con footprint diversi.

Dunque è sempre più chiaro come sia necessario avere una "collezione" di tutti i possibili simboli; questo concetto non è proprio di EAGLE ma si estende in generale e questa collezione prende il nome di libreria di componenti. Abbiamo quindi introdotto tre termini che ci accompagneranno per

tutto il nostro corso:

- **COMPONENTE:** in EAGLE prende il nome di **DEVICE** e copre qualsiasi tipologia di componente fisico (resistore, condensatore, ecc.) oppure semplicemente virtuale (power symbol, ground symbol, cartiglio, ecc.);
- **SIMBOLO:** in EAGLE prende il nome di **SYMBOL** e rappresenta un disegno che schematizza qualsiasi componente su schema elettrico;
- **IMPRONTA:** identificato come **FOOTPRINT** e rappresenta nient'altro che il componente stesso ma su layout, dunque è l'insieme degli elementi:
 - Ingombro geometrico del componente con eventuali zone di rispetto.
 - Profilo geometrico dei pin.
 - Profilo dei fori per viti o perni.
 - Campo testo per il riferimento del componente.
 - Eventuale serigrafia del perimetro del componente.
 - Eventuale campo/i testo per indicazioni testuali particolari.

Andiamo quindi avanti nella creazione di una libreria di componenti, detta **LIBRARY**, per poter procedere alla creazione del nostro progetto.

CONFIGURAZIONE DEL CONTROL PANEL

Prima di tutto organizziamo il nostro lavoro configurando il Control Panel (CP) che essendo a tutti gli effetti una classica interfaccia utente, contiene una barra dei menu. Dunque ritroviamo i menu File, View, Options, Window e Help. Senza dilungarsi troppo, le voci del menu verranno descritte all'occorrenza; in questa sezione è interessante descrivere la voce per la configurazione dell'ambiente: il menù Options. In Fig. 11 è possibile osservare le finestre di dialogo selezionabili nel menu Options:

- Directories;
- Backup/Locking;
- User Interface;
- Proxy settings;
- Data Collection and Use.

Con riferimento alla Fig. 12, descriviamo le relative finestre di dialogo.

A.Options → Backup/Locking: EAGLE permette un backup del progetto fino a 9 livelli (0 per disabilitare il backup). Si può scegliere l'intervallo di tempo dopo il quale è possibile eseguire ogni backup.

Una volta raggiunto il massimo livello di backup il successivo andrà a sovrascrivere il primo salvato. Spuntando "Automatically save project file" verrà salvato il progetto quando si esce da EAGLE.

B.Options → User Interface: Permette di personalizzare l'aspetto generale oltre che delle finestre di editor layout, schematic e library.

C.Options → Proxy Settings: Permette di impostare le opzioni di server proxy.

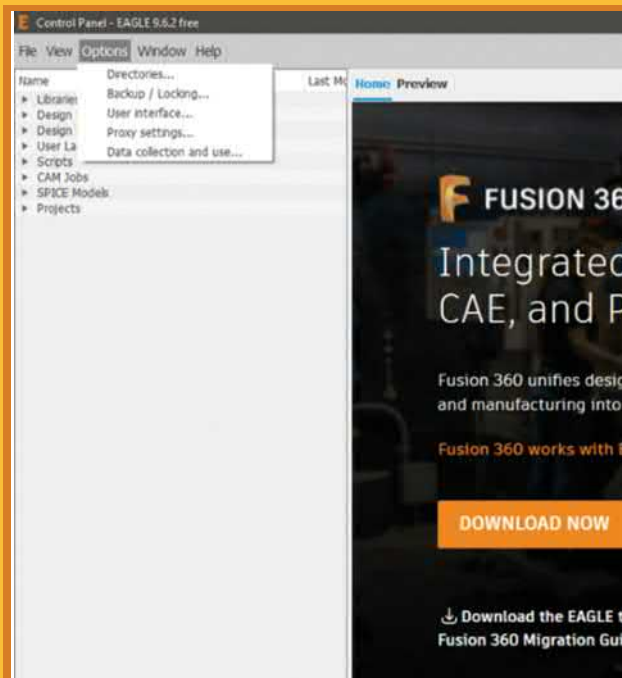


Fig. 11 Interfaccia principale del Control Panel

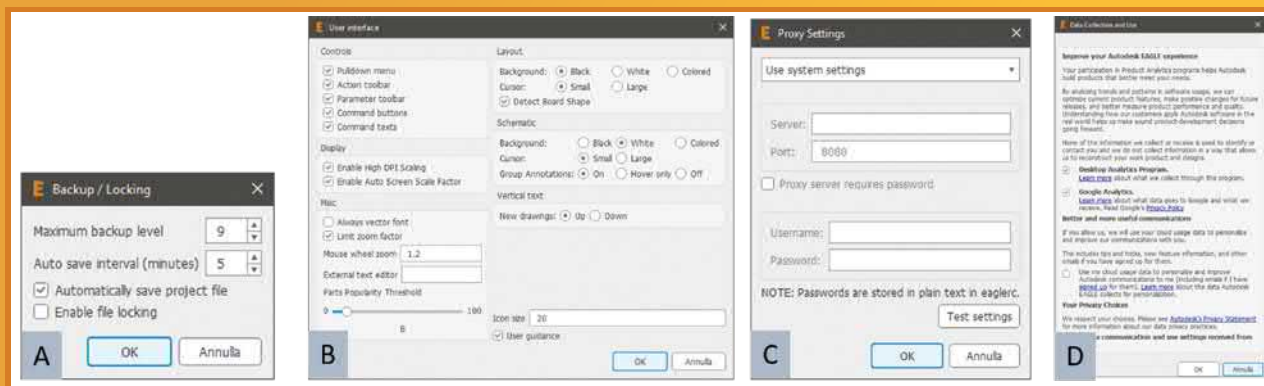


Fig. 12 Le finestre di dialogo del CP

D.Options → Data Collection and Use: Permette di configurare le impostazioni di privacy.

Tutte le finestre di dialogo elencate riguardano una specifica configurazione dell'ambiente EAGLE; fa eccezione Options → Directories che volutamente non è stata inserita nella tabella perché richiede una descrizione più dettagliata.

FINESTRA DIRECTORIES

La finestra di dialogo Directories, come da Fig. 13, è usata per definire i percorsi in cui cercare i file; è sicuramente una finestra di configurazione dell'ambiente EAGLE ma, ad eccezione delle altre, ha un legame stretto con il progetto perché attraverso questa finestra si va a puntare ai file di libreria nonché di schematici e layout. Da notare che le voci che troviamo in tale finestra sono le stesse che troviamo all'interno della vista ad albero del CP. Infatti se espandiamo una delle voci della vista ad albero (e.g.: Projects) il contenuto rispecchia il contenuto del percorso della voce corrispondente più eventualmente una sottocartella examples se è spuntato il controllo "Include EAGLE examples". Per aggiungere uno o più percorsi a cui puntare è sufficiente aggiungerli all'interno della casella di testo a destra della voce separati da ";" (punto e virgola). Questa operazione può essere ovviamente fatta a mano scrivendo per intero o incollando il percorso di riferimento nella casella di testo oppure è sufficiente posizionarsi all'interno della casella di testo e selezionare il pulsante "Browse..." dunque apparirà una classica finestra di dialogo e una volta selezionata la cartella di file desiderata apparirà, già separata da ";", all'interno della casella di testo. Questa operazione vale per tutte le voci della finestra. Qualora si volesse tornare ad una condizione di default (la stessa mostrata nella figura) è sufficiente selezionare il pulsante "Set to defaults". Ovviamente il pulsante "Ok" salva le modifiche mentre "Annulla" le tralascia. All'interno delle varie caselle di testo appaiono due stringhe interessanti:

- "\$HOME": rappresenta il percorso locale dei Documenti.
- "\$EAGLEDIR": rappresenta il percorso d'installazione di EAGLE.

Dunque abbiamo ora gli strumenti necessari e sufficienti per poter costruire un progetto partendo da zero. Per farlo verranno seguiti una serie di step, da qui in avanti, solo a scopo didattico e che quindi non rappresentano l'unica modalità per realizzare un progetto in EAGLE.

GESTIONE DELLE LIBRERIE

Partiamo ora con la configurazione e successiva creazione di una libreria di componenti. Per cominciare creiamo una cartella dal nome "NordIOT" in un percorso a scelta (e.g.: sul desktop). Creiamo poi due sottocartelle "Libraries" e "Project"; torniamo poi nella finestra di dialogo "Directories" e posizioniamoci all'interno della casella "Libraries" dunque selezioniamo "Browse..."

Attraverso la finestra di dialogo puntiamo al percorso "Libraries" all'interno di "NordIOT" quindi selezioniamo "Seleziona cartella".

Come mostrato nella Fig. 14A, il percorso selezionato appare in coda nella casella di testo separato da ";". Selezioniamo "OK" e torniamo sulla finestra del Control Panel; espandendo la voce "Libraries" (la prima della vista ad albero) troviamo il percorso appena inserito come evidenziato nella Fig. 14B.

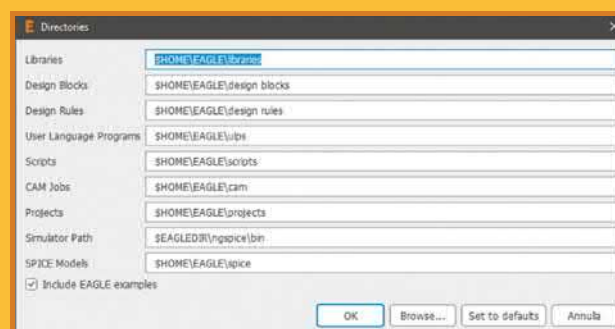


Fig. 13 Le finestre Directories

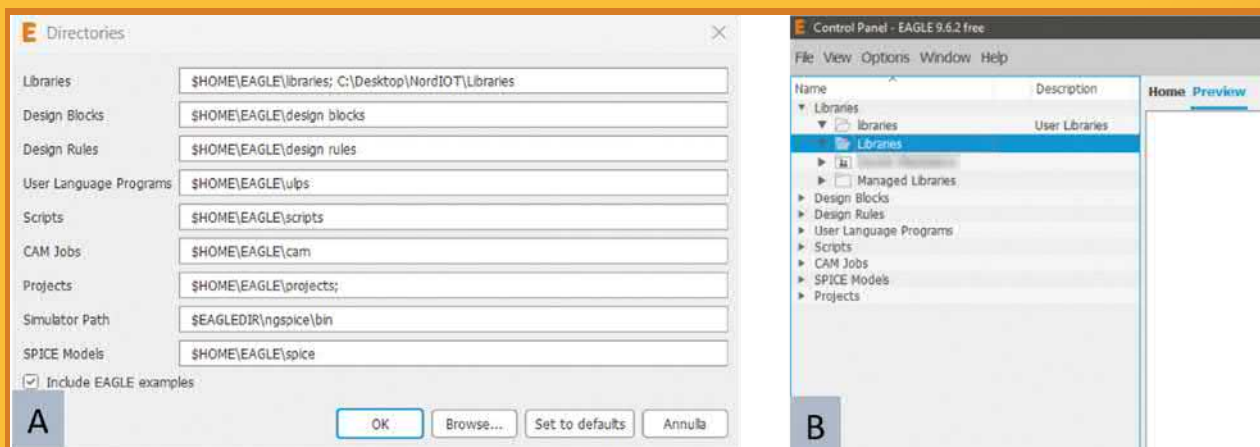


Fig. 14 Configurazione percorso di libreria

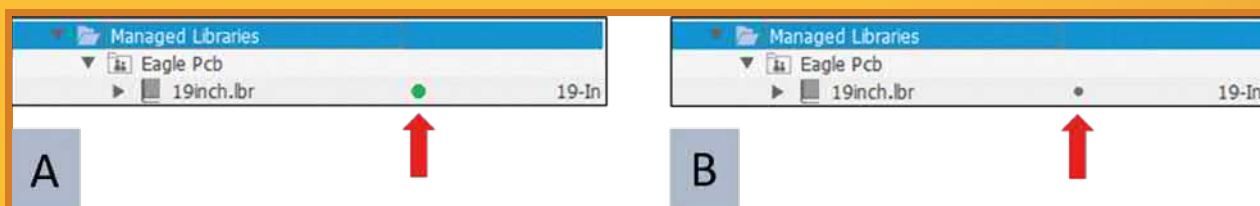


Fig. 15 Attivazione/disattivazione libreria

Oltre a quanto inserito ci sono diversi altri percorsi, in particolare:

- **libraries:** fa riferimento al percorso "\$HOME\EAGLE\libraries".
- **<username> registrato:** è il percorso sfocato e fa riferimento ad un percorso gestito su library.io, un sito web realizzato per condividere, collaborare e salvare le librerie EAGLE ed i modelli 3D.
- **Managed Libraries:** sono le librerie fornite con EAGLE, anche queste sono gestite su library.io e compilate con molta cura e servizi aggiuntivi direttamente per l'utente EAGLE.

Per il momento focalizziamo l'attenzione sul percorso Library della cartella NordIOT e vediamo come disabilitare l'utilizzo delle librerie da non usare nel progetto; selezioniamo la voce libraries all'interno della finestra ad albero, con il tasto destro del mouse apparirà il popup menu relativo alla libreria selezionata e selezionando "Use none" all'interno del menu, tutte le librerie all'interno delle sottocartelle verranno disabilitate all'uso. Ripetere questo passo per tutte le librerie di non interesse eccetto per Libraries. Questo passo è fondamentale nella gestione della componentistica da utilizzare, perché come vedremo nelle puntate successive, per inserire un componente nel layout e ancor prima nello schema, si userà il comando "Add" che andrà ad indirizzare tutte le librerie abilitate all'uso cioè seguite da un "pallino

no verde" (Fig. 15A) anziché un "pallino grigio" (Fig. 15B), ovvero disattivate.

In realtà, anche se ci stiamo focalizzando sulle librerie, il menu contestuale appena visto si applica a tutti i tipi di cartella (non esclusivamente alle librerie) mentre per i singoli file (e.g.: la libreria "19inch.lbr" dell'immagine precedente) il menu contestuale cambia.

Vedremo all'occorrenza le tipologie di menu contestuale; per ora descriviamo brevemente le voci del menu contestuale soprariportato:

- **New Folder:** questo comando permette di creare una sottocartella all'interno della cartella selezionata. Ciò accade sia nella finestra ad albero sia fisicamente nella directory corrispondente.
- **Edit Description:** questo comando permette di inserire una descrizione "in formato html" della cartella selezionata.
- **Use all:** abilita all'uso tutti i file all'interno della cartella e delle sottocartelle.
- **Use none:** disabilita all'uso tutti i file all'interno della cartella e delle sottocartelle.
- **Search in folder:** questo comando abilita la barra di ricerca in fondo alla finestra del control panel e permette di ricercare un oggetto all'interno della cartella.

Una volta descritto l'essenziale per gestire una libreria (o più in generale un file) all'interno del Control Panel, proce-

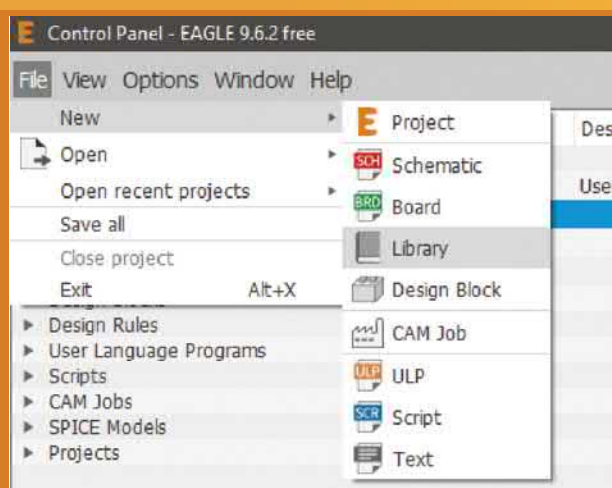


Fig. 16 Creazione nuova libreria

diamo con la creazione vera e propria della nostra libreria componenti.

CREAZIONE DELLE LIBRERIE

Le librerie in EAGLE sono dei veri e propri file con estensione lbr. Dunque, tornando al precedente esempio, esisterà un file 19inch.lbr all'interno delle cartelle di EAGLE che rappresenta la libreria suddetta.

Facendo riferimento alla Fig. 16, vediamo come creare una nuova libreria, quindi un nuovo file lbr, a partire dal Control Panel attraverso i passi: 1) Selezionare il menu File, 2) Selezionare il sottomenu New, 3) Selezionare il comando Library. Apparirà la finestra di nuova libreria senza titolo presentata nella Fig. 17. Notiamo che il nuovo file viene temporaneamente salvato all'interno della cartella EAGLE nei documenti personali (la famosa \$HOME\EAGLE\libraries) dunque diventerà una libreria utente. La finestra si compone di quattro listbox sulla sinistra ed una finestra proprietà più a destra. Più in basso a ciascuna listbox appaiono i pulsanti "Add Device...", "Add Footprint", "Import 3D Package" e "Add Symbol...". Notiamo come tre delle quattro listbox con pulsante "Add" richiamano i concetti accennati ad inizio puntata riguardo il componente. Il package 3D è stato introdotto più recentemente e verrà approfondito all'occorrenza. La prima cosa da fare è salvare la nostra nuova libreria nella cartella NordIOT (ricordiamo che le librerie utente sono state disabilitate). L'operazione è semplice dato che la finestra è dotata di un classico menu: dal menu File selezionare il comando "Save as..." quindi riportarsi all'interno della cartella NordIOT\Libraries. Per il salvataggio possiamo utilizzare il nome predefinito oppure usarne uno a piacere. In linea generale si consiglia di nominare le librerie in modo da raggruppare una categoria di componenti (ad esempio Passives.lbr per i componenti passivi). A questo punto abbiamo creato la nostra prima libreria di componenti che, essendo vuota, andrà popolata con almeno

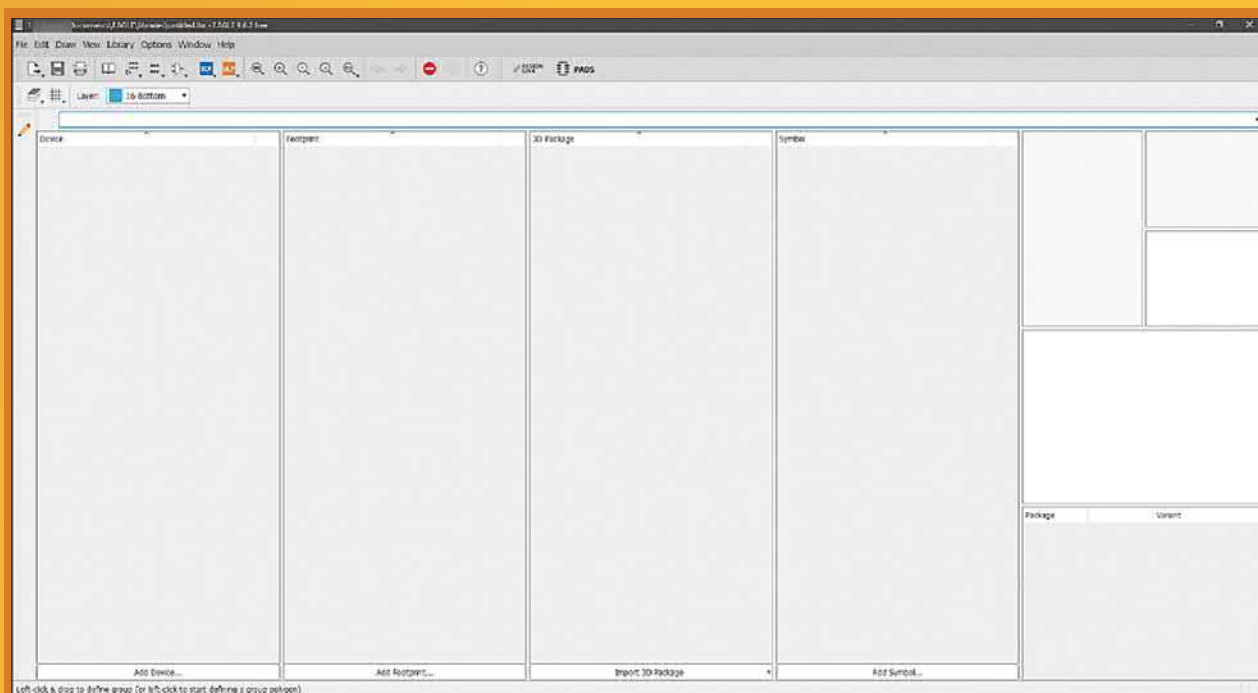


Fig. 17 La nuova libreria vuota

un componente. Come esempio scegliamo una resistenza dunque andremo prima a creare un simbolo che identifica la resistenza poi un footprint d'esempio che ci serviranno per generare il nostro primo componente semplice.

CREAZIONE DEI COMPONENTI: EDITOR DEI SIMBOLI

Per generare un nuovo simbolo selezioniamo "Add symbol..." in basso alla listbox Symbol. Comparirà una finestra di input dove sarà possibile inserire il nome del nuovo simbolo.

Per semplicità chiamiamo questo nuovo simbolo RESISTOR_SYM e confermiamo cliccando su "OK". È bene notare che in questa finestra, ma anche in quelle relative al device e footprint, è possibile usare solo caratteri maiuscoli e caratteri speciali, eventuali caratteri minuscoli vengono automaticamente commutati nel corrispettivo maiuscolo. Selezionando OK viene richiesta una conferma sulla creazione del simbolo dunque confermiamo. Apparirà un ambiente per il disegno dei simboli molto simile all'ambiente Schematic per il disegno degli schemi elettrici. Prima di introdurre gli strumenti messi a disposizione impostiamo il piano di lavoro (indicato con il numero 3 in Fig. 18) selezionando lo strumento Grid (indicato con 1 in Fig. 18) oppure digitiamo il comando GRID da riga di comando (indicata con 2 in Fig. 19). Apparirà la finestra della griglia mostrata in Fig. 19A.

Le impostazioni riportate sono predefinite (le stesse che si ottengono selezionando il pulsante Default in basso a sinistra). Non è consigliato modificare queste impostazioni ai fini di utilizzare i simboli all'interno degli schemi però non è certamente proibito; si consiglia, qualora queste impostazioni di griglia vengono modificate, di prendere nota delle modifiche e riportarle all'interno delle impostazioni dello schema al fine di evitare che i componenti non siano in griglia con i collegamenti nello schema. Per il momento limitiamoci solo a modificare le impostazioni di visualizzazione quindi lo stile della griglia e l'unità di misura passando da pollici (inch) a millimetri (mm). Il risultato finale è mostrato in Fig. 19B. Notiamo che cambiando l'unità di misura automaticamente cambia secondo l'equivalenza 1

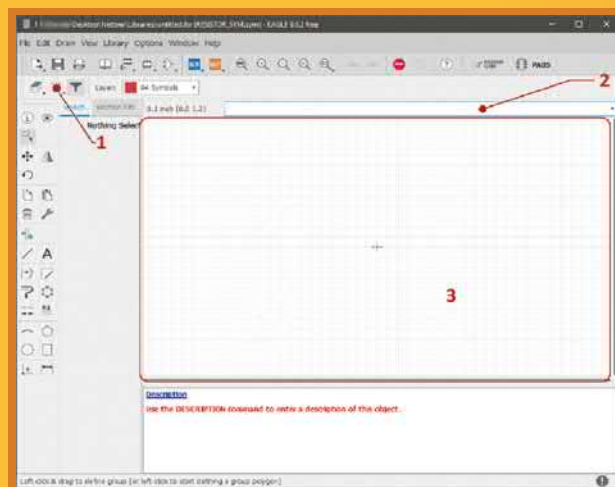


Fig. 18 Editor di simboli in libreria

inch = 2,54 mm. Selezionando OK notiamo che il foglio di lavoro cambia aspetto. Andiamo brevemente a descrivere gli strumenti di lavoro per il disegno di un simbolo; questi strumenti sono riportati nella Fig. 19C.

Disegniamo dunque il corpo del simbolo come in Fig. 20A utilizzando lo strumento Line. Per terminare l'operazione di disegno è sufficiente selezionare il tasto ESC. Notiamo che i punti di disegno sono ancorati nei punti di griglia con passo 2,54mm. È possibile disegnare con una griglia più fitta tenendo premuto il pulsante ALT; in questo caso il passo sarà 0,254mm. Il prossimo passo è l'inserimento dei pin di collegamento: nel caso di un resistore ne abbiamo due che possiamo inserire come effettuato in Fig. 20B.

Per ruotare i pin è sufficiente premere il tasto destro del mouse. I punti di collegamento con le connessioni future nello schematico (o note con il termine "net") sono cerchiati ed identificate da un'etichetta (nel nostro caso "io 0") che sta ad identificare direzione e livello di interscambio del pin. Ultime operazioni per la creazione di un simbolo sono

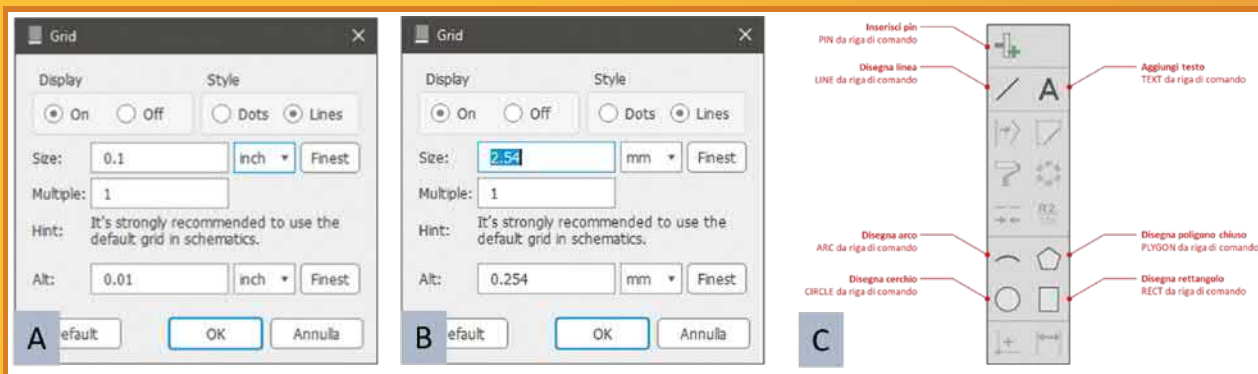


Fig. 19 Griglia e impostazioni di editor

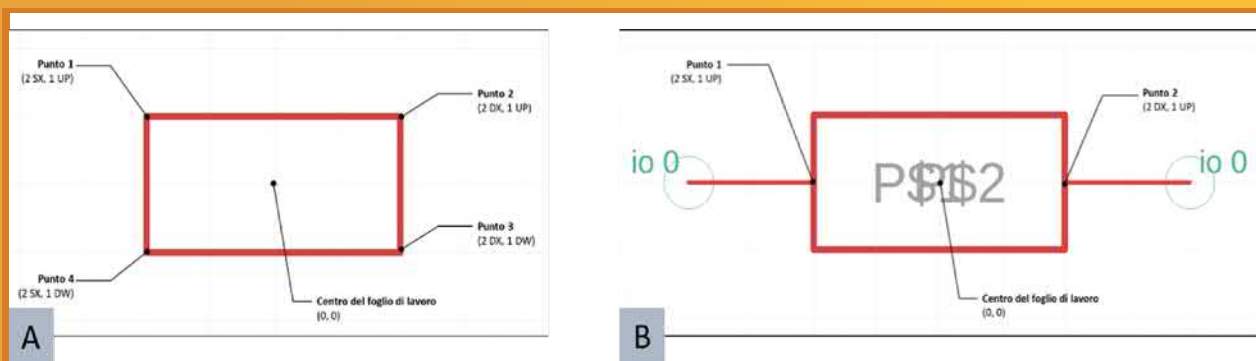


Fig. 20 Disegniamo il componente

l'aggiunta dei campi di testo, in particolare i campi VALUE e NAME. Il primo identifica il componente (e.g.: il valore di resistenza) mentre il secondo darà un riferimento al componente (e.g.: R1). Per inserirli bisogna cambiare layer; infatti finora abbiamo lavorato nel layer n° 94 Symbols ma esistono diversi layer di lavoro (questo è valido sia a livello di schematic che board) ognuno con una propria funzione e significato. Posizioniamoci dunque nel layer 96 Values selezionandolo dalla casella a discesa in alto a sinistra come mostrato nella Fig. 21A. Una volta selezionato il layer 96 selezioniamo lo strumento aggiungi testo; appare una casella di testo come da Fig. 21B, quindi editiamo all'interno ">VALUE" e selezioniamo OK. Posizioniamo il testo sotto il componente come mostrato in Fig. 22A. Per completare l'operazione selezioniamo due volte il tasto escape. Ripetiamo l'operazione aggiungi testo però selezionando il layer 95 e inserendo il testo ">NAME". Il risultato apparirà come in Fig. 22B.

Possiamo ritenere conclusa la creazione del nostro primo simbolo; salviamo il lavoro.

Come nella maggior parte dei programmi a finestra, tra l'altro dotati di menu, per salvare è sufficiente selezionare la combinazione dei tasti CTRL+S oppure utilizzare lo

strumento a forma di floppy-disk nella barra strumenti o anche la voce "Save..." nel menu "File".

Una volta salvato, per ritornare nella libreria è sufficiente chiudere la finestra però, al fine di introdurre qualche nozione in più su EAGLE, soffermiamoci qualche istante sul simbolo appena creato; i due pin sono stati inseriti con direzione io (input/output), il che presuppone siano di tipologia logica/digitale. Questo aspetto non è generalmente un problema ma potrebbe portare a problematiche di tipo DRC (design rules check) nel momento in cui si andrà a compilare uno schema. Infatti un pin "io" concettualmente potrà essere collegato con un altro (o più d'uno) pin di "io", input ("in") o output "out". Dunque è necessario modificare la direzione di tali pin e, per farlo, è sufficiente posizionarsi col puntatore sul cerchio del pin e selezionare il tasto destro del mouse; apparirà il menu contestuale di Fig. 23A.

Selezioniamo la voce "Properties" dunque apparirà la finestra di Fig. 23B. In questa finestra vengono descritte tutte le proprietà del pin appena selezionato; quella di nostro interesse è "direction" dove possiamo scambiare "io" con "pas" (passive). Di seguito vengono brevemente descritte queste proprietà.

- **Name:** rappresenta il nome del pin.
- **Position:** è la posizione del pin rispetto all'origine.
- **Angle:** è possibile posizionare il pin solo a 0°, 90°, 180° o 270° rispetto alla posizione inserita. Valori diversi non sono consentiti.
- **Direction e Swap Level:** insieme formano l'etichetta che abbiamo visto sul pin (e.g.: "io 0"). Queste due voci meritano un approfondimento.
- **Length:** rappresenta la lunghezza del pin. È possibile selezionare tra point (0 passo di griglia), short (1 passo di griglia), middle (2 passi di griglia) e long (3 passi di griglia).
- **Function:** rappresenta la funzione associata a tale pin generalmente se si dà un significato logico al pin. È possibile selezionare tra none (nessuna funzione), dot (negato, conferisce il classico cerchietto all'estremità

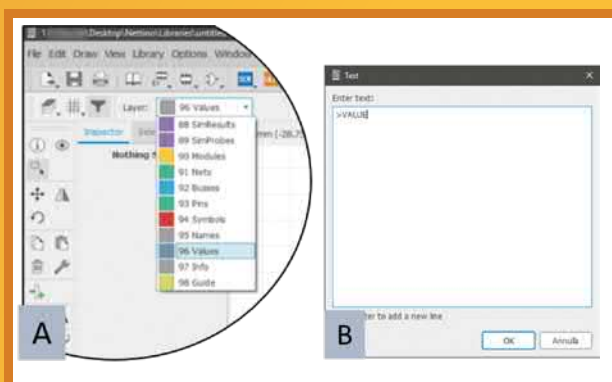


Fig. 21 Cambio di layer e inserimento testo

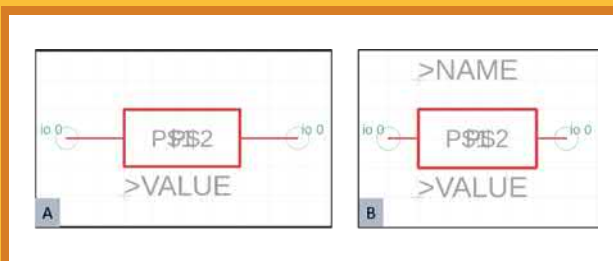


Fig. 22 Inseriamo i due campi VALUE e NAME

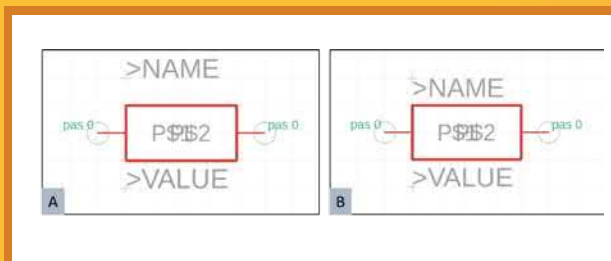


Fig. 24 Impostazione dei pin e completamento componente

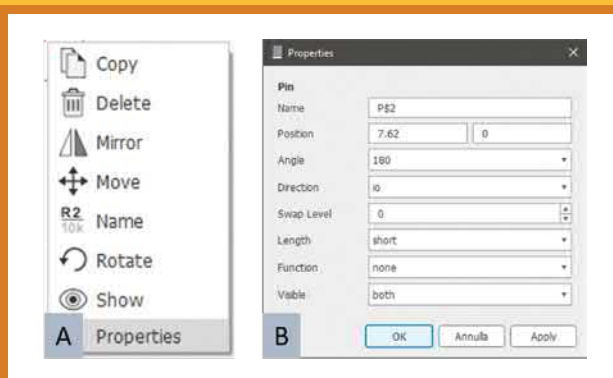


Fig. 23 Proprietà del componente

del pin), clk (clock, conferisce il simbolo di clock all'estremità del pin) e dotclk.

- **Visible:** indica la visibilità del pin. Approfondiremo tale voce nel momento della creazione del simbolo.

Selezionando Ok nella finestra e ripetendo lo stesso procedimento per l'altro pin il risultato è riportato in Fig. 24A. In ultimo riposizioniamo il campo ">NAME" che darà il riferimento al nostro componente (e.g.: R1) in modo che non sia troppo distante dal corpo del componente. Per farlo è sufficiente entrare nelle proprietà del campo: il procedimento è identico a quanto fatto per il pin. Nel campo "Align"

selezioniamo "top-left" dunque applichiamo le modifiche e ritorniamo al nostro foglio di lavoro ottenendo il risultato di Fig. 24B. A tal punto possiamo ritenere conclusa la creazione del simbolo dunque salviamo e ritorniamo nella finestra della libreria mediante il tasto indicato in Fig. 25A. Siamo tornati nella finestra dei contenuti della libreria che, dopo aver creato il nostro primo simbolo, appare come nella Fig. 25B.

CREARE I COMPONENTI: EDITOR DEI FOOTPRINT

Passiamo ora alla creazione del nostro primo footprint selezionando "Add Footprint..." proprio sotto la listbox Footprint. Come nel caso del simbolo appare una finestra di input dove inseriremo il nome del nostro footprint. Per semplicità chiamiamolo R0603_PKG. Selezionando OK verrà richiesta una conferma sulla creazione; confermando si apre la finestra di disegnazione footprint come mostrato in Fig. 26A. La finestra è del tutto simile a quella per la creazione dei simboli e, come vedremo, alla finestra per la disegnazione dei layout. Ovviamente trattandosi di un footprint ci saranno degli strumenti relativi ai footprint come l'inserimento dei pad sia THT che SMD e l'inserimento dei fori come è possibile vedere nella casella degli strumenti. Tuttavia rimangono le stesse considerazioni per la finestra di creazione dei simboli soprattutto per quanto riguarda l'impostazione della griglia che impostiamo semplicemente in millimetri (mm) e la lista dei layer. La lista degli strumenti principali viene riportata in Fig. 26B.

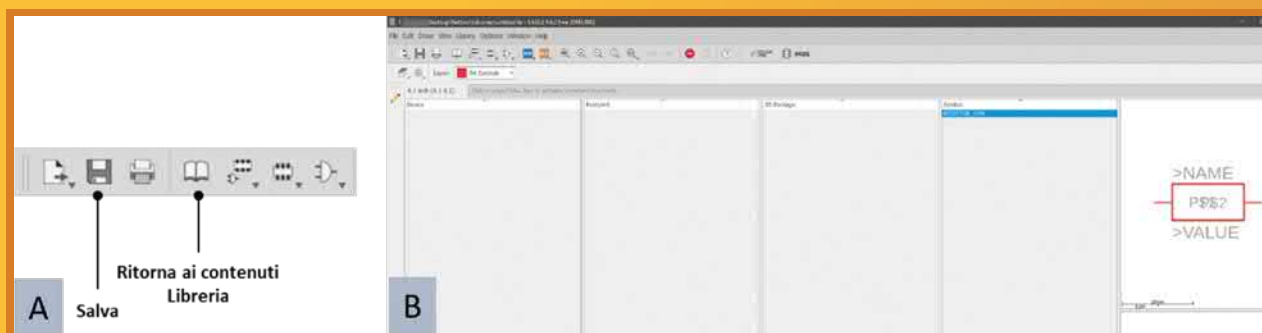


Fig. 25 Salvataggio e visualizzazione dell'intera libreria

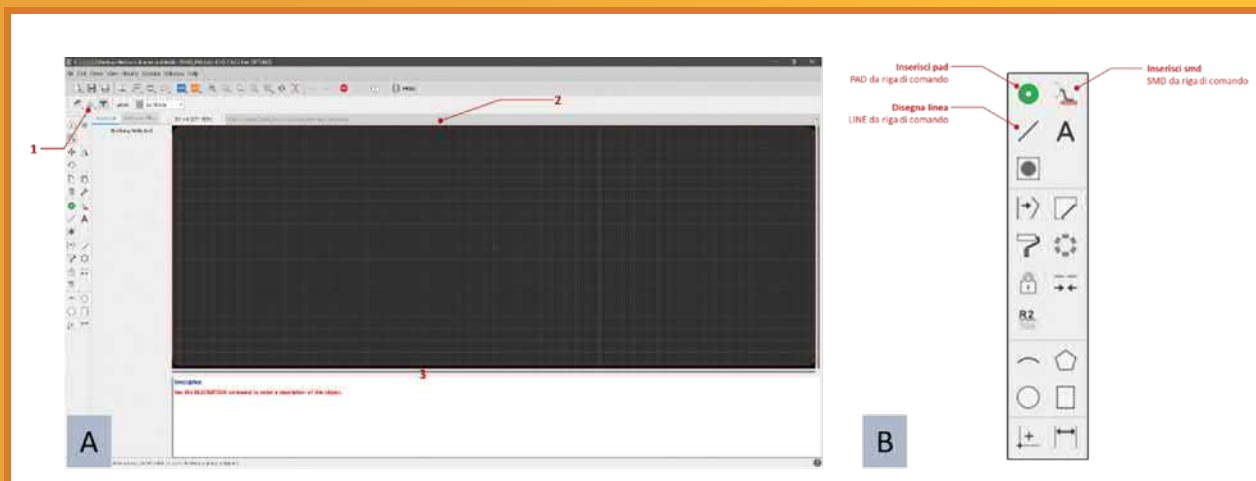


Fig. 26 Editor per la realizzazione dei Footprint e lista strumenti disponibili

La prima cosa che facciamo è inserire le piazzole del resistore selezionando lo strumento "SMD" oppure digitando "SMD" da riga di comando. Dunque posizioniamo la prima piazzola a sinistra e la seconda a destra dell'origine per esempio a due punti di griglia ciascuna. A differenza dei simboli un footprint ha bisogno di rispettare dei vincoli dimensionali perciò le piazzole dovranno avere delle dimensioni e posizioni particolari del componente. La scelta dei due punti di griglia è solo per comodità; infatti se clicchiamo col tasto destro sul primo pad (quello di sinistra) e selezioniamo "Properties" appare la finestra di proprietà del pad come mostrato in Fig. 27A.

Ovviamente, avendo impostato la griglia in mm, dimensioni e posizione sono in mm. Nella Fig. 27B è riportato un esempio di "recommended footprint" tratto da datasheet VISHAY per tre principali case (0603, 0805 e 1206) di resistori. Come vediamo ogni pad è largo 0,5mm e alto 0,9mm; inoltre la distanza tra i bordi dei due pad è 1,0mm quindi la distanza centro-centro dei pad è 1,5mm oppure, se voglia-

mo, il centro di ogni pad dista 0,75mm dall'origine posta al centro del componente. Trasliamo queste informazioni all'interno della finestra proprietà del primo pad facendo attenzione che, essendo alla sinistra dell'origine, come "Position" avrà un valore negativo (ovvero con segno meno "-"). Nella Fig. 28A viene rappresentato un confronto tra il pad modificato (a sinistra) e quello "standard" non modificato. Selezioniamo OK e ripetiamo la stessa operazione per il pad a destra facendo attenzione che la posizione avrà valore positivo. Nella finestra proprietà del pad (smd) vengono descritte tutte le proprietà di seguito brevemente descritte.

- **Name:** rappresenta il nome del pin.
- **Position:** è la posizione del pin rispetto all'origine.
- **Angle:** ogni pad può essere disposto con qualsiasi angolazione.
- **Smd Size:** rappresenta le dimensioni del pad nel formato larghezza per altezza (L x H).
- **Layer:** indica il layer di rame del pad. È possibile selezionare solo i due layer conduttivi più esterni (Top e Bottom) o, nel caso siano previsti, più interni.
- **Roundness:** rappresenta l'arrotondamento, espresso in percentuale, dei vertici del pad.
- **Locked:** indica se il pin viene bloccato da modifica (generalmente lo spostamento).
- **Thermals:** indica se il pad presenterà o meno il thermal relief quando affogato in un piano di massa.
- **Stop e Cream:** indicano l'inclusione dei layer 29 tStop e 31 tCream.

Come fatto per un simbolo aggiungiamo ancora i campi di testo >NAME e >VALUE sotto i rispettivi layer 25 tNames e 27 tValues; in questo modo il nostro footprint sarà corredato di reference come da schema. Si potrebbero ancora aggiungere la serigrafia (silkscreen) e il disegno del componente rispettivamente nei layer 21 tPlace e 51 tDocu.

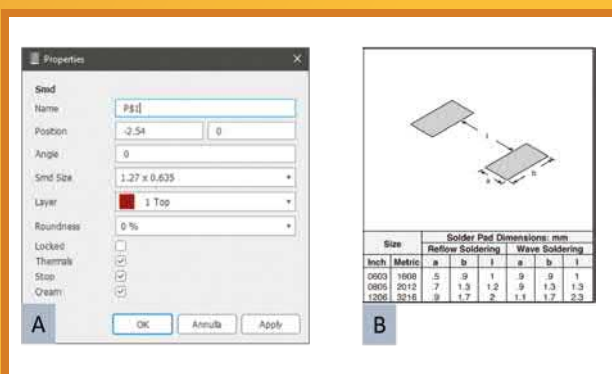
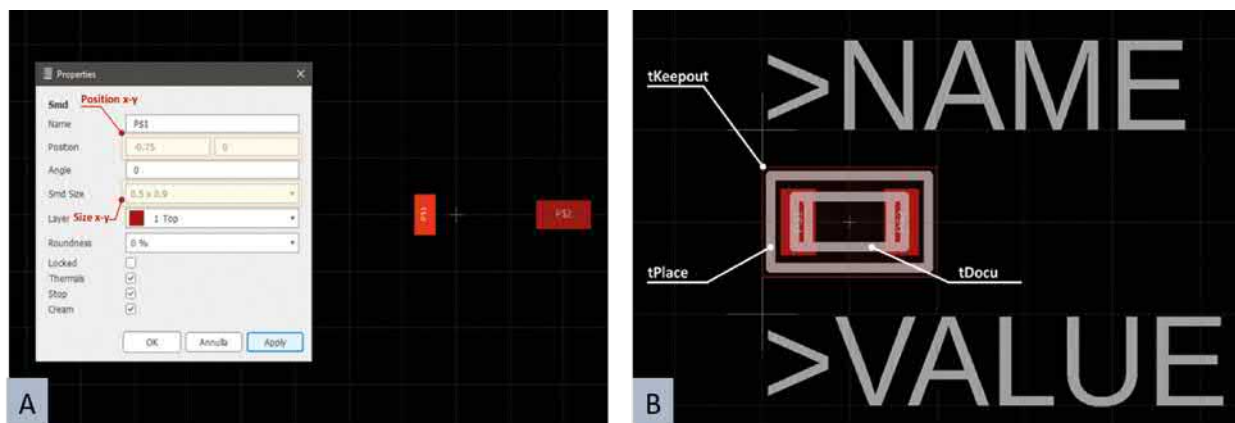


Fig. 27 Editor per la realizzazione dei Footprint e lista strumenti disponibili



↑ Fig. 28 Impostazioni delle proprietà e dei layer del componente

La differenza sostanziale tra questi ultimi due è che il primo rappresenta una parte grafica visualizzabile direttamente sullo stampato mentre la seconda è semplicemente di documentazione quindi visibile per esempio sulle viste topografiche del layout. Per creare questi ultimi due oggetti è consigliabile l'uso di linee.

Poniamo una particolare attenzione ad alcuni tipi di layer:

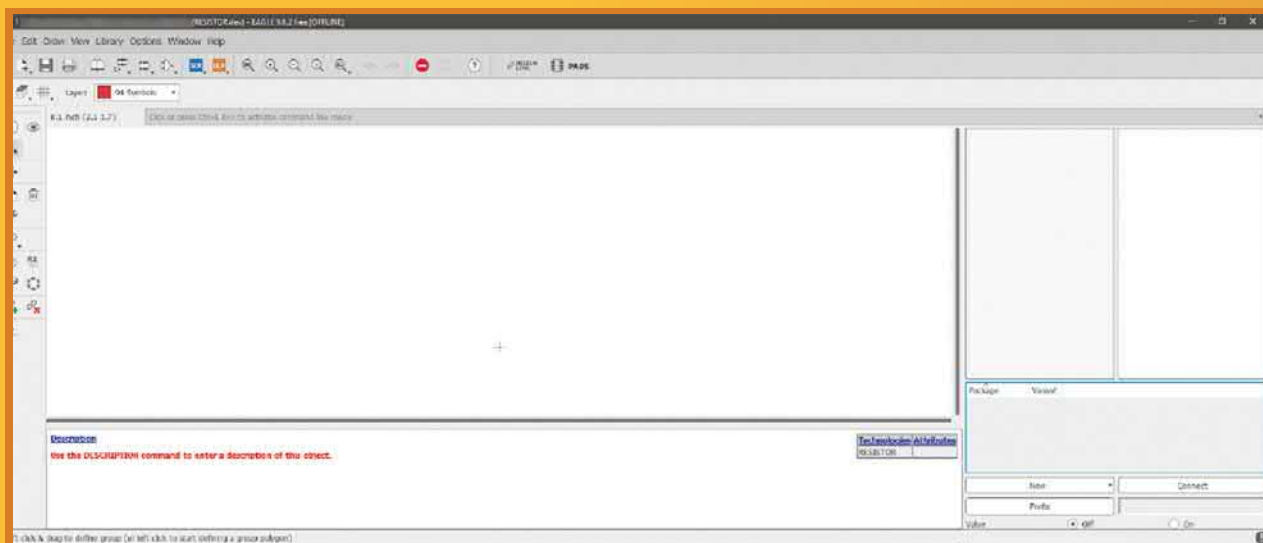
- 29 tStop: rappresenta l'area di apertura del solder resist.
- 31 tCream: è l'area in cui viene deposta la pasta salda.
- 39 tKeepout: è l'area di pulizia da altri componenti.
- 41 tRestrict: rappresenta l'area di pulizia da altri conduttori.

I primi due sono tipici dei pad, mentre gli altri riguardano il componente. La Fig. 28B mostra il footprint completo

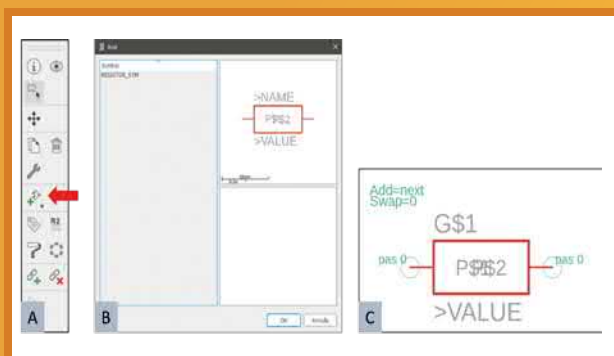
di silkscreen, ingombro topografico e keepout. Possiamo ritenere conclusa la disegnatione del nostro footprint dunque salviamo il lavoro e ritorniamo alla libreria componenti come fatto in precedenza e possiamo notare come anche nella colonna footprint sia presente un'occorrenza "R0603_PKG".

CREARE COMPONENTI: EDITOR DEI COMPONENTI

Ci rimane un ultimo passaggio, ovvero la realizzazione di un componente che collega il simbolo al footprint e che permetta il suo utilizzo all'interno di schematici e board. Sotto la prima listbox, selezioniamo "Add Device..." e nominiamo il device come RESISTOR. Al termine dell'operazione apparirà la finestra di editor device (Fig. 29).



↑ Fig. 29 Editor dei componenti



↑ Fig. 30 L'aggiunta del Symbol

Un componente si crea per associazione tra un simbolo ed uno o più footprint e la procedura è immediata, ovvero, partiamo dal fissare il prefisso del componente selezionando il tasto "Prefix" in basso a destra della finestra e si aprirà una finestra aggiuntiva di inserimento. Questo può essere composto da qualunque stringa e rappresenta la prima parte del riferimento (reference) componente all'interno dello schema elettrico. Nel caso specifico inseriamo il prefisso "R" trattandosi di una resistenza e confermiamo con il tasto OK. Attiviamo la visualizzazione del campo value selezionando "On" sempre in basso a destra nella finestra principale dell'editor. A questo punto selezioniamo "Add part" nella barra degli strumenti a sinistra come da Fig. 30A. Apparirà la finestra "Add" contenente l'elenco dei simboli della nostra libreria. Nel nostro esempio ci sarà solo il simbolo RESISTOR_SYM. Selezioniamo tale simbolo e premiamo OK. Posizioniamo il simbolo con il suo centro coincidente con l'origine dell'area di lavoro come da Figura 31C. Una volta posizionato selezionare per due volte ESC. Nella

finestra principale dell'editor, in basso a destra selezionare il pulsante "New" quindi "Add local package". Apparirà la finestra "Create new package variant for ..." dove sarà presente l'elenco dei footprint della nostra libreria.

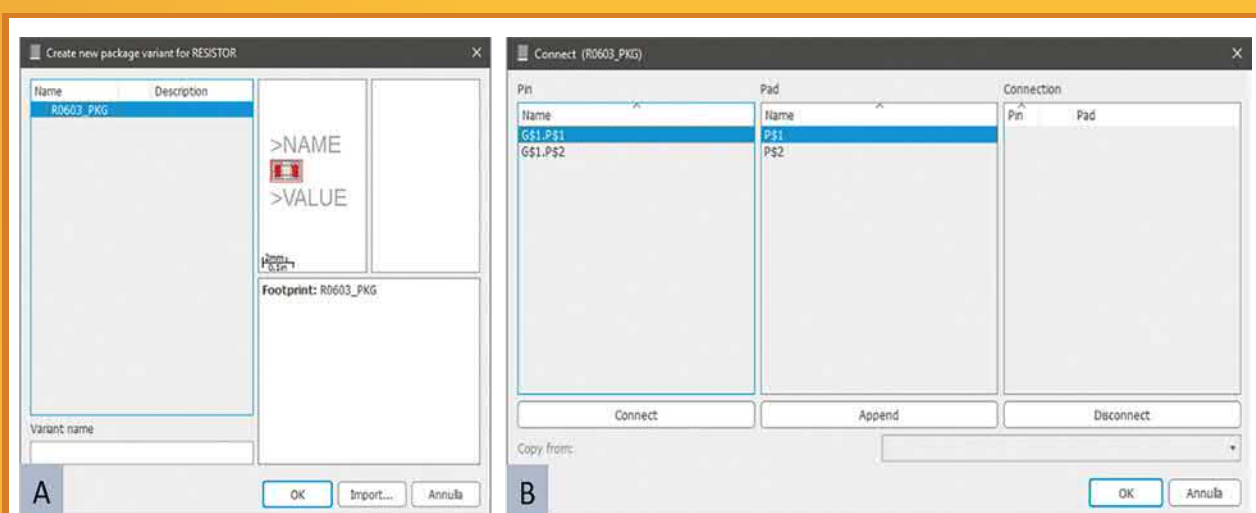
Nel nostro esempio ci sarà solo il footprint R0603_PKG (come da Fig. 31A); selezionare tale footprint e premere Ok. Questo passo può essere ripetuto tante volte quanti sono i footprint che vogliamo associare (ad esempio R0805_PKG, R1206_PKG, ecc.). Ancora una volta vi ritroverete nella finestra principale dell'editor, a questo punto premete sul pulsante "Connect" in basso a destra, in modo tale che apparirà la finestra "Connect", all'interno della quale distinguiamo i pin del simbolo a sinistra ed i pad del footprint a destra. Selezioniamo, per ciascun pin, un pad da associare, quindi selezioniamo "Connect". Quando tutti i pin e i pad di nostro interesse saranno associati, clicchiamo Ok (Fig. 31B).

A questo punto possiamo ritenere conclusa la creazione del componente: salviamo e torniamo alla libreria. Nel caso in cui si fossero creati altri footprint per il nostro esempio la procedura di associazione può essere ripetuta. Inoltre, quando si andrà ad associare i pin con i pad, non è necessario ripetere tutti i punti, ma si può selezionare il primo componente associato nella combo-box "Copy from:".

IMPORTAZIONE COMPONENTI

È chiaro che creare una libreria da zero potrebbe rappresentare un lavoro molto oneroso, soprattutto per quanto riguarda componenti semplici. Questi componenti possono essere direttamente importati da altre librerie, per esempio quella di EAGLE; per farlo è necessario che la libreria di destinazione, ovvero la finestra editor delle librerie sia aperta; di seguito spieghiamo come copiare un componente da un'altra libreria.

Per prima cosa andiamo sul control panel e posizioniamoci



↑ Fig. 31 Connessione tra simbolo e footprint



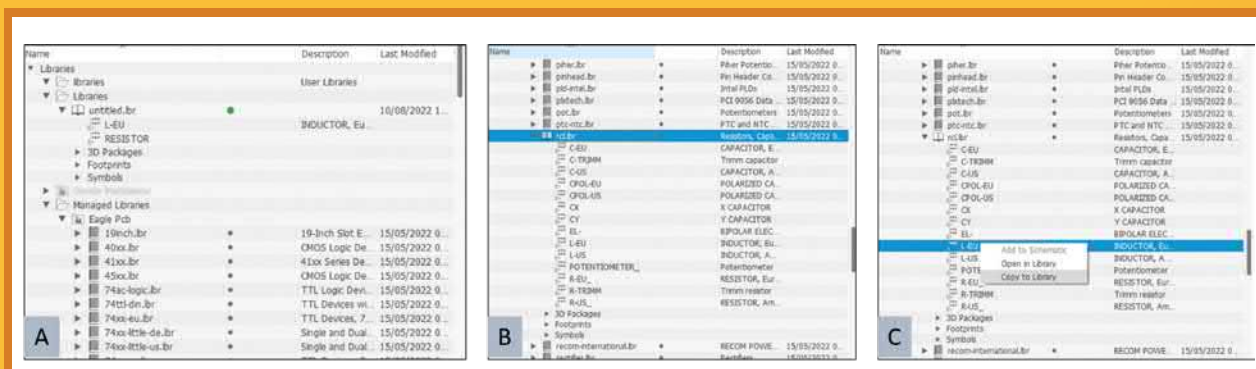


Fig. 32 Popolare la libreria di interesse da altre librerie

nella finestra ad albero espandendo la libreria "Managed Libraries\EAGLE PCB" (Fig. 32A); nell'elenco individuiamo la libreria rcl.lbr e la espandiamo (Fig. 32B). Clicchiamo con il tasto destro sul componente "L-EU" e selezioniamo "copy to library" (Fig. 32C), si aprirà la finestra editor dei componenti; salviamo e torniamo alla nostra libreria. Ritroveremo il componente copiato più tutte le dipendenze (simboli, footprint ed eventualmente 3D) che si porta dietro. Analogamente è possibile copiare altri componenti, corredando il componente di diversi footprint in una volta sola. Non è obbligatorio copiare l'intero componente dalla libreria, ma è possibile copiare solo i simboli o solo i footprint. La lista seguente elenca librerie interessanti per l'importazione dei componenti a partire da "Managed Libraries\EAGLE PCB":

- Passivi: rcl.lbr.
- Potenzimetri: pot.lbr.
- Package: ic-packages.lbr, smd-ipc.lbr (solo footprint).
- Semiconduttori: semicon-smd-ipc.lbr, transistor-fet.lbr.
- Simboli di alimentazione: supply1.lbr, supply2.lbr
- Cartigli: frames.lbr
- LED: led.lbr

A questo punto vi lasciamo come esercizio la realizzazione della libreria di progetto NordIOT (proposta in Fig. 33) che sarà disponibile per il download sul sito www.futuranet.it al termine del corso, tenendo presente che combinando tutte le nozioni precedenti è possibile importare una libreria scaricata da internet all'interno del Control Panel per poi copiare all'interno della libreria di progetto solamente tutti i componenti d'interesse; oppure partire da schematici EAGLE presenti online, esportare la libreria corrispondente per poi utilizzarla nel proprio progetto che approfondiremo nelle puntate successive.

CONCLUSIONI

Siamo quindi giunti al termine di questa prima puntata; abbiamo installato e introdotto l'ambiente, descritto il work-flow, l'interfaccia e alcune funzionalità saliente del CP per poi passare all'argomento fondamentale del ciclo di

sviluppo di un progetto elettronico, ovvero la realizzazione delle librerie di componenti. Dalla prossima puntata inizieremo ad occuparci della realizzazione degli schematici con l'ausilio del progetto pratico NordIOT che ci accompagnerà in tutte le prossime puntate e ci porterà alla realizzazione di una scheda wireless mediante il modulo U-blox Nora-b106 con core della Nordic nrf5340.



Fig. 33 Libreria del progetto NordIOT

TrinamiX sviluppa l'antifurto biometrico

Continental e trinamiX hanno svelato Driver Identification Display, un sistema antifurto biometrico basato sul riconoscimento facciale. La soluzione è dotata di una fotocamera integrata nel cruscotto per l'identificazione del volto del conducente. Simile al riconoscimento facciale utilizzato sugli smartphone di oggi, la nuova tecnologia significa che un veicolo può essere avviato solo dopo che la persona al posto di guida è stata riconosciuta. La soluzione migliora anche altre funzionalità come i pagamenti digitali, che vengono elaborati utilizzando il display dell'auto, per il rifornimento, il parcheggio, il pagamento dei pedaggi o il noleggio dell'auto. Oltre a ciò, la telecamera integrata monitora l'attenzione del guidatore per evitare, ad esempio, che si addormenti al volante. Driver Identification Display offre tre funzioni chiave in un unico pacchetto: autenticazione affidabile del conducente e possibilità di effettuare transazioni di pagamento digitali sicure tramite il display del veicolo, ad esempio nei garage, ai caselli autostradali o negli app store; la telecamera, installata in modo invisibile nel sistema, viene utilizzata anche per monitorare l'attenzione del guidatore, cosa che in futuro sarà resa obbligatoria sulle automobili vendute in Europa.

La soluzione di sicurezza utilizza l'autenticazione facciale integrata basata su una tecnologia sviluppata e brevettata da trinamiX, la quale combina il riconoscimento facciale per verificare l'identità dell'utente con un nuovo tipo di rilevamento della vivacità; è in grado di riconoscere la vera pelle umana per prevenire lo spoofing del sistema con fotografie o maschere 3D realistiche del proprietario o conducente del veicolo. L'autenticazione facciale di trinamiX è stata certificata secondo i più elevati standard di sicurezza biometrica.

www.continental.com



Emirates testa un motore GE90 con carburante SAF

Emirates ha eseguito con successo una serie di test per l'impiego di solo carburante sostenibile per aviazione (SAF, Sustainable Aviation Fuel) nei suoi velivoli; il primo passo è stato un test a terra alimentando uno dei motori General Electric GE90 di un Boeing 777-300ER il 100% di SAF.

La compagnia aerea mira ora ad effettuare un volo di prova utilizzando il 100% di SAF in entrambi i motori. Per il momento, per ragioni di sicurezza Emirates ha effettuato il suo primo volo dimostrativo su un Boeing 777-300ER, alimentando uno solo dei motori turboventola con carburante SAF; il possente Boeing è decollato dall'aeroporto internazionale di Dubai (DXB) al comando del capitano Fali Vajifdar e dal capitano Khalid Nasser Akram, volando per più di un'ora sulla costa di Dubai.

L'equipaggio della cabina di pilotaggio era accompagnato da Adel Al Redha, Chief Operating Officer di Emirates, e dal Capitano Hassan Hammadi, Divisional Senior Vice President, Emirates Flight Operations.

Il volo dimostrativo alimentato da SAF ha un significato particolare per gli Emirati Arabi Uniti, che hanno dichiarato il 2023 Anno della sostenibilità, che metterà in mostra l'impegno del Paese arabo nel cercare soluzioni innovative a sfide come la ricerca di nuove forme di energia, la lotta ai cambiamenti climatici e altre questioni legate alla sostenibilità.

Il volo dimostrativo di Emirates, il primo in Medio Oriente e Nord Africa ad essere alimentato al 100% da SAF, supporta sforzi più ampi per ridurre le emissioni di anidride carbonica nel ciclo di vita di un aereo, che coinvolgono anche l'industria. I voli contribuiranno anche a perfezionare il playbook per le future dimostrazioni SAF e supporteranno la futura certificazione del SAF come carburante esclusivo per la propulsione aerea civile; attualmente il SAF è approvato per l'uso su tutti gli aerei, ma solo in miscele fino al 50% con carburante per aerei convenzionale.



Emirates ha collaborato con i partner GE Aerospace, Boeing, Honeywell, Neste e Virent per procurarsi e sviluppare una miscela di SAF che riproduce da vicino le proprietà del carburante per aerei convenzionale. Per ogni rapporto di miscelazione sono state effettuate misurazioni delle proprietà chimiche e fisiche del carburante. Dopo numerosi test di laboratorio e prove rigorose, sono arrivati a un rapporto di miscelazione che rispecchiava le qualità del carburante per aerei. Sono state miscelate diciotto tonnellate di SAF, composto da HEFA-SPK fornito da Neste (esteri

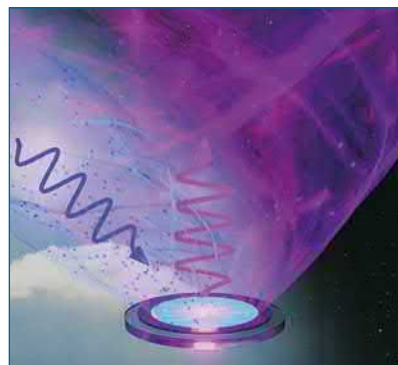
e acidi grassi idrotrattati e cherosene paraffinico sintetico) e HDO-SAK da Virent (cherosene aromatico sintetico idrodeossigenato).

Il volo di prova dimostra ulteriormente la compatibilità del SAF appositamente miscelato come fonte di carburante sicura e affidabile.

Il promettente risultato di questa iniziativa si aggiunge anche al corpus di dati e ricerche del settore sulle miscele SAF in proporzioni più elevate, aprendo la strada alla standardizzazione e alla futura approvazione del 100% di SAF in sostituzione del carburante per aerei.

Il SAF è stato testato anche nel funzionamento dell'unità di alimentazione ausiliaria 331-500 (APU) che è il sistema che fornisce l'alimentazione a terra a motori spenti, l'energia per l'avviamento dei motori, la climatizzazione e il backup di emergenza in volo.

www.emirates.com



Fotoni dal visibile all'ultravioletto

La luce ultravioletta (UV) ha fotoni ad energia più elevata rispetto alla luce visibile e per sfruttare tale caratteristica i ricercatori del Tokyo Tech hanno sviluppato un materiale a stato solido in grado di convertire in modo stabile ed efficiente i fotoni della luce visibile solare in fotoni UV.

La luce solare contiene luce UV ad alta energia con una lunghezza d'onda inferiore a 400 nm, che può essere ampiamente utilizzata, ad esempio, per la fotopolimerizzazione per formare una resina e l'attivazione di fotocatalizzatori per guidare reazioni che generano idrogeno verde o idrocarburi utili (combustibili, zuccheri, olefine, eccetera). Quest'ultima è spesso chiamata "fotosintesi artificiale". La reazione fotocatalitica mediante luce UV per uccidere efficacemente virus e batteri è un'altra importante applicazione. Però solo il 4% circa della luce solare terrestre rientra nella gamma UV, il che lascia una grande porzione dello spettro solare non sfruttata per questi scopi. La conversione dei fotoni avverrebbe tramite un processo chiamato "annichilazione tripletta-tripletta" o TTA. Fino ad ora si utilizzavano solventi organici che richiedevano che la soluzione venisse prima deossigenata e poi sigillata in un contenitore ermetico; i materiali mancavano di fotostabilità in presenza di ossigeno e non riuscivano a funzionare efficacemente con luce incidente. Tali problemi sembrano stati risolti al Tokyo Tech, con una pellicola solida fatta di un sensibilizzatore (cioè un cromoforo molecolare in grado di assorbire fotoni di lunghezza d'onda maggiore) e un'elevata quantità di un annihilatore (una molecola organica che riceve la tripletta di energia eccitata dal sensibilizzatore).

www.titech.ac.jp/english/news/



In servizio il bus a fuel-cell di Toyota

La prefettura di Fukuoka, la Commercial Japan Partnership Technologies Corporation (CJPT) e la Kyushu Railway Company (JR Kyushu) hanno stipulato un accordo per testare un autobus elettrico a celle a combustibile sulla BRT Hikoboshi Line. Questa iniziativa permetterà di verificare l'utilizzo commerciale del trasporto pubblico basato su idrogeno e fuel-cell, attraverso l'impiego su un percorso di linea in parallelo all'uso degli autobus a motore endotermico.

L'autobus a fuel-cell è basato sul bus Toyota Coaster, da 20 posti con 14 posti effettivi; questo automezzo per il trasporto di persone è lungo 6,99 m, largo 2,1 m e alto 2,63 m. La propulsione è a

motori elettrici alimentati dall'elettricità sviluppata a partire dall'idrogeno puro da un sistema a fuel-cell di seconda generazione derivato dalla Toyota Mirai FCEV. Il progetto si inquadra in un programma di elettrificazione dei trasporti locali, che dovrebbe coinvolgere anche le merci, con l'adozione di camion elettrici leggeri a fuel-cell da parte degli operatori logistici, nonché di macchine operatrici urbane, per esempio i camion per la raccolta dei rifiuti urbani (raccolta rifiuti) e ambulanze. Il programma comprende anche lo sviluppo di infrastrutture progettate per opzioni di mobilità a fuel-cell.

global.toyota/en/newsroom

Elettricità dalle rotaie

Il Virginia Tech Center for Vehicle Systems and Safety (CvESS) e il Railway Technologies Laboratory vogliono sfruttare l'energia creata dai treni in movimento e trasformarla in elettricità utilizzabile. La ricerca è stata finanziata dal Dipartimento dei trasporti degli Stati Uniti attraverso il Centro di trasporto universitario sul miglioramento della sostenibilità e della durabilità delle infrastrutture di trasporto ferroviario.

Circa il 76 per cento delle miglia totali sui binari ferroviari americani si trova in aree rurali prive dell'elettricità necessaria per il funzionamento di sistemi ferroviari in-

telligenti come come apparecchiature di sicurezza lungo la strada, comunicazioni wireless e monitor per lo stato dei binari. Ecco quindi l'idea di un generatore di elettricità integrato nella traversina, ovvero che sostituisce una traversina: le staffe imbullonate sono flessibili ed applicate a una barra di metallo, la quale sormonta una molla, cosicché quando le ruote del treno passano sopra la rotaia, il peso spinge verso il basso azionando una serie di ingranaggi. Gli ingranaggi fanno ruotare l'albero di un generatore, creando elettricità, che può quindi essere immagazzinata in una batteria.

I ricercatori hanno testato il dispositivo su una linea della Norfolk-Southern per determinare quanta energia può essere generata e come ottimizzare il design del dispositivo. Per ogni ruota del treno che passa, si generano da 15 a 20 watt; il passaggio di un treno con 200 vagoni, quindi almeno 800 ruote, produrrebbe 1,6 kilowatt, più che sufficienti ad alimentare infrastrutture e sistemi locali di sicurezza e monitoraggio delle linee.

vtx.vt.edu/articles



In orbita il satellite Inmarsat 6 F2

Il satellite per comunicazioni più avanzato del mondo è decollato nella notte del 17 febbraio scorso a bordo di un razzo vettore Falcon 9 di SpaceX, per arricchire la flotta satellitare di Inmarsat. L'I-6 F2 ha spiccato il volo dallo Space Launch Complex 40 presso la Cape Canaveral Space Force Station (USA) quindi si è separato dal razzo circa 32 minuti dopo e tramite la propulsione elettrica si è diretto verso la sua posizione orbitale geostazionaria sull'Oceano Atlantico. In linea con ogni missione Falcon 9, il primo stadio del razzo è atterrato con successo poco dopo il lancio, in modo da essere riutilizzato. Il booster ha precedentemente supportato un volo con equipaggio verso la Stazione Spaziale Internazionale e il lancio di un satellite GPS 3. L'I-6 F2 entrerà in servizio nel 2024 dopo rigorosi test in orbita. Il satellite è identico all'Inmarsat-6 F1 (I-6 F2) che dovrebbe iniziare le operazioni da uno slot geostazionario sull'Oceano Indiano all'inizio di quest'anno, dopo il suo lancio nel dicembre 2021 da parte della giapponese Mitsubishi Heavy Industries. Entrambi i satelliti, costruiti da Airbus, sono dotati di un carico utile ibrido in banda Ka e banda L per fornire principalmente servizi di connettività mobile nei mercati marittimo, aeronautico e governativo. La capacità in banda Ka dell'I-6 F1 mirerà alla domanda di connettività ad alta velocità nel sud-est asiatico, mentre i segnali in banda Ka dell'I-6 F2 si rivolgeranno agli hotspot sull'Oceano Atlantico. Inmarsat conta oggi otto satelliti.

www.inmarsat.com



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.

RICARICA I TUOI DISPOSITIVI OVUNQUE TI TROVI!

PANNELLO SOLARE PORTATILE DA 60W



PANNELLO SOLARE MONOCRISTALLINO DA 60W

ad alta efficienza (16,85%), pieghevole e portatile. Dispone di uscita DC max 20,9 volt 2,87A, uscita USB-A da 5V 3A, uscita USB-QC: 5V 3A / 9V 2A / 12V 1,5A. Esternamente è realizzato in nylon e dispone di due robuste maniglie con potenti magneti che assicurano una salda chiusura. Ideale per il campeggio, l'escursionismo e altre attività all'aperto. Dimensioni: 530 x 440 x 30 mm (chiuso), 880 x 530 x 10 mm (aperto), peso: 2,9 g.



cod. FOLDINGPAN60

€ 135,00



USCITA PRESA DC

(5,5x2,1 mm): max 20,9 volt – 2,87 A

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

USCITA USB-A
5V 3A

Prezzo IVA inclusa.

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

È DISPONIBILE IL NUOVO E-BOOK Le tecnologie RFID e NFC



A SOLO
€ **7,90**
cod. EBOOK231

Prezzo IVA inclusa.

L'e-book "Le tecnologie RFID e NFC" approfondisce le tematiche delle moderne soluzioni basate su sistemi wireless e contactless, sia da un punto di vista teorico-didattico che da un punto di vista pratico-progettuale, ed è rivolto a chiunque voglia entrare in questa branca dell'Elettronica Digitale.

