

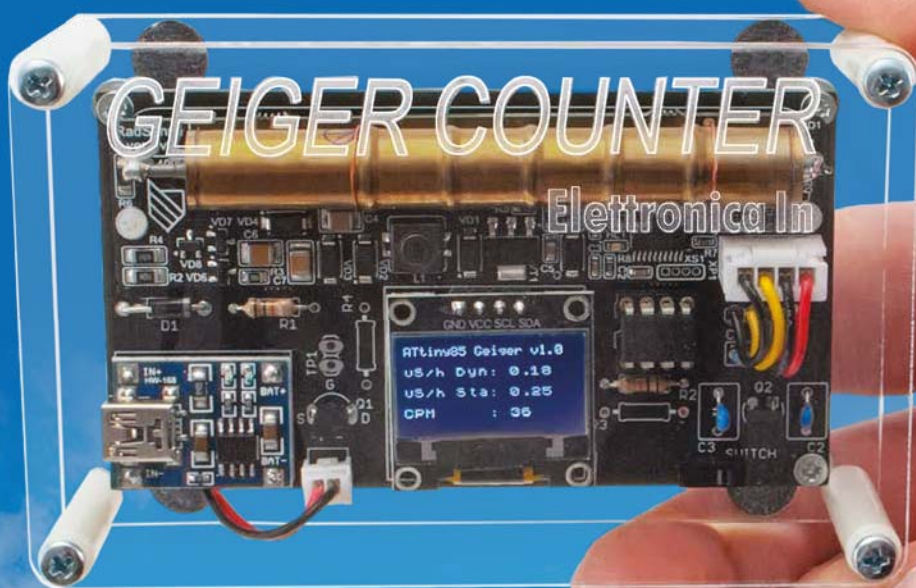
Elettronica In

www.elettronica.in.it

€ 6,90 - Anno XXVIII - n. 263 - APRILE 2022



- Realizzare un PCB con la CNC
- La stampante 3D sotto controllo
- Distributore di farmaci con micro:bit
- Deviatore Smart
- Termometri con Arduino
- Video in Time Lapse su Cloud



ATtiny85 RadSense Un contatore Geiger tascabile

Una rete neurale
con  PyTorch



Tachimetro BLE
per biciclette



CONOSCERE
E USARE



Node-RED

MAKER UNO

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

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

+

50
video
tutorial



€ 58,00

cod. MAKERKITBASE

**FUTURA
ELETTRONICA®**

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

Prezzo IVA inclusa

Caratteristiche tecniche
e vendita on-line su:

www.futurashop.it



Il lato positivo della crisi

JET, il reattore sperimentale a fusione nucleare europeo, ha raddoppiato il record di energia prodotta grazie a plasma incredibilmente caldo che ha rilasciato 59 megaJoule di energia. Sebbene non si tratti di una quantità esorbitante, i nuovi esperimenti hanno generato questa energia utilizzando solo 170 microgrammi di combustibile deuterio-trizio.

I nuovi esperimenti di JET sono necessari per spianare la strada a ITER (International Thermonuclear Experimental Reactor), che mira a diventare il primo impianto di fusione nucleare al mondo.

Molti privati stanno investendo nella fusione nucleare.

Questo reattore è attualmente in costruzione nel sud della Francia e ha lo scopo di rilasciare 10 volte più energia di quanta ne sia necessaria per innescare la fusione.

Purtroppo siamo ancora lontani dal poter considerare questo tipo di reattore un sistema per produrre energia: al momento nessun esperimento di fusione nucleare ha generato più energia di quanta sia necessaria per far funzionare il reattore stesso (il recente esperimento di JET ha un fattore di conversione (Q) di 0,35). Ma il dato importante è che JET è stato in grado di ottenere un plasma parzialmente auto-sostenuto che ha rilasciato una quantità significativa di energia di fusione con un funzionamento prolungato dell'apparato (5 secondi), un risultato che fornisce grande fiducia nell'ottenimento degli obiettivi previsti dai futuri dispositivi di fusione.

Ma non ci sono solo governi a studiare la fusione nucleare, infatti anche molti privati, forse complice la grave crisi energetico-economica che stiamo attraversando, stanno investendo in questa direzione vedendo nella fusione nucleare una fonte quasi illimitata di energia pulita.

Come scriveva Einstein "La crisi è la più grande benedizione per le persone e le nazioni, perché la crisi porta progressi" e ci auguriamo che questa crisi porti un'accelerazione nel processo e ci conduca verso un mondo più pulito e sostenibile.

ON LINE

elettronica.in.it

SI RINGRAZIA

Futura Elettronica Mouser



In copertina:
Realizziamo, partendo da un "vecchio" sensore, un moderno contatore Geiger.

Boris Landoni
 boris.landoni@elettronica.in.it



Elettronica In

Rivista mensile, anno XXVIII n. 263

APRILE 2022

EDITORIALE

Direttore Responsabile

Boris Landoni

boris.landoni@elettronica.in.it

Redazione

Stefano Garavaglia, Paolo Gaspari,
Davide Scullino, Gabriele Daghetta

redazione@elettronica.in.it

Grafica

Alessia Sfulcini

alessia.sfulcini@elettronica.in.it

Ufficio Pubblicità

Monica Premoli (0331-752668)

monica.premoli@elettronica.in.it

Ufficio Abbonamenti

Tel. 0331-752668

abbonati@elettronica.in.it

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

Tel. 0331-752668

Abbonamenti

Annuo 10 numeri Euro 49,00

Annuo 10 numeri digitale Euro 39,00

Le richieste di abbonamento vanno inviate a:

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

Tel. 0331-752668

oppure tramite il sito
<https://www.elettronica.in.it/abbonamenti/>

Distribuzione per l'Italia

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

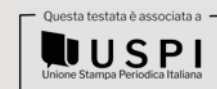
via Bettola 18 - 20092 Cinisello Balsamo (MI)

Tel. 02-660301 Fax 02-66030320

Stampa

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

Tiratura
18.000 copie



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

Prezzo di copertina Euro 6,90.

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

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

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

Impaginazione ed immagini sono realizzati in Desktop Publishing
con programmi Adobe InDesign e Adobe Photoshop per Windows.
Tutti i contenuti della Rivista sono protetti da Copyright.

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

© 2022 Futura Group srl

Futura
Group
Edizioni

Contenuti

01

RUBRICHE

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

Progetto di Copertina

ATTINY85 RADSENSE: UN CONTATORE GEIGER TASCABILE



54

Wireless

La stampante 3D sotto controllo

63



19

ARTICOLI

WIRELESS

19 Tachimetro BLE per biciclette

Basato sul sensore ad effetto Hall 49E interfacciato a un Atmega328 e programmato tramite l'IDE Microchip Studio, invia i chilometri percorsi e velocità tramite la scheda di sviluppo HELTEC WiFi kit 32 OLED.

NETWORKING

33 Motion Time Lapse in Cloud

Concludiamo l'analisi del nostro sistema basato su ESP32-CAM e Google Apps Script, vedendo il debug e l'applicazione Motion Lapse. Seconda e ultima puntata.

HI-TECH

41 Realizzare un PCB con la CNC

Impariamo a realizzare un PCB artigianale con la CNC, partendo dal disegno del circuito stampato e utilizzando software open source o free. Prima puntata.

Wireless

Tachimetro BLE per biciclette



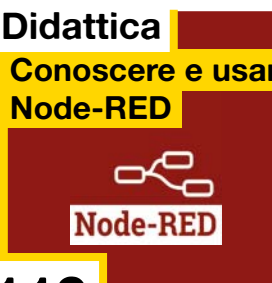
19

Networking

Motion Time Lapse in cloud



33

Hi-Tech**Realizzare un PCB con la CNC****41****Healthcare****Distributore farmaci con micro:bit****73****Networking****Deviatore Smart****87****Arduino****Termometri con Arduino****99****Didattica****Conoscere e usare Node-RED****119****Intelligenza Artificiale****Costruiamo una NN con PyTorch****109****STRUMENTAZIONE****54 ATtiny85 RadSense: un contatore Geiger tascabile**

Utilizziamo un modulo con tubo Geiger per costruire un rilevatore tascabile di radiazioni. Il tutto gestito da un piccolo ATtiny85 e alimentato con una batteria al litio ricaricabile.

WIRELESS**63 La stampante 3D sotto controllo**

Configuriamo il modulo ESP3D per ricevere le notifiche di fine filamento e di fine stampa, e per acquisire lo streaming video. Terza ed ultima puntata.

HEALTHCARE**73 Distributore farmaci con micro:bit**

Sfruttiamo l'elevata flessibilità hardware e software di questa scheda dalle piccole dimensioni ma dalle grandi potenzialità per realizzare un distributore di farmaci settimanale interconnesso.

NETWORKING**87 Deviatore Smart**

Controllato da remoto da smartphone o mediante comandi vocali utilizzando l'assistente vocale Amazon Alexa oppure Google Assistant, permette di simulare la funzione del deviatore per impianti elettrici, ma con controllo WiFi.

ARDUINO**99 Termometri con Arduino**

Misuriamo la temperatura utilizzando come sensori termistore, termoresistenza e termocoppia, per apprendere vantaggi e limiti di queste soluzioni.

INTELLIGENZA ARTIFICIALE**109 Costruiamo una NN con PyTorch**

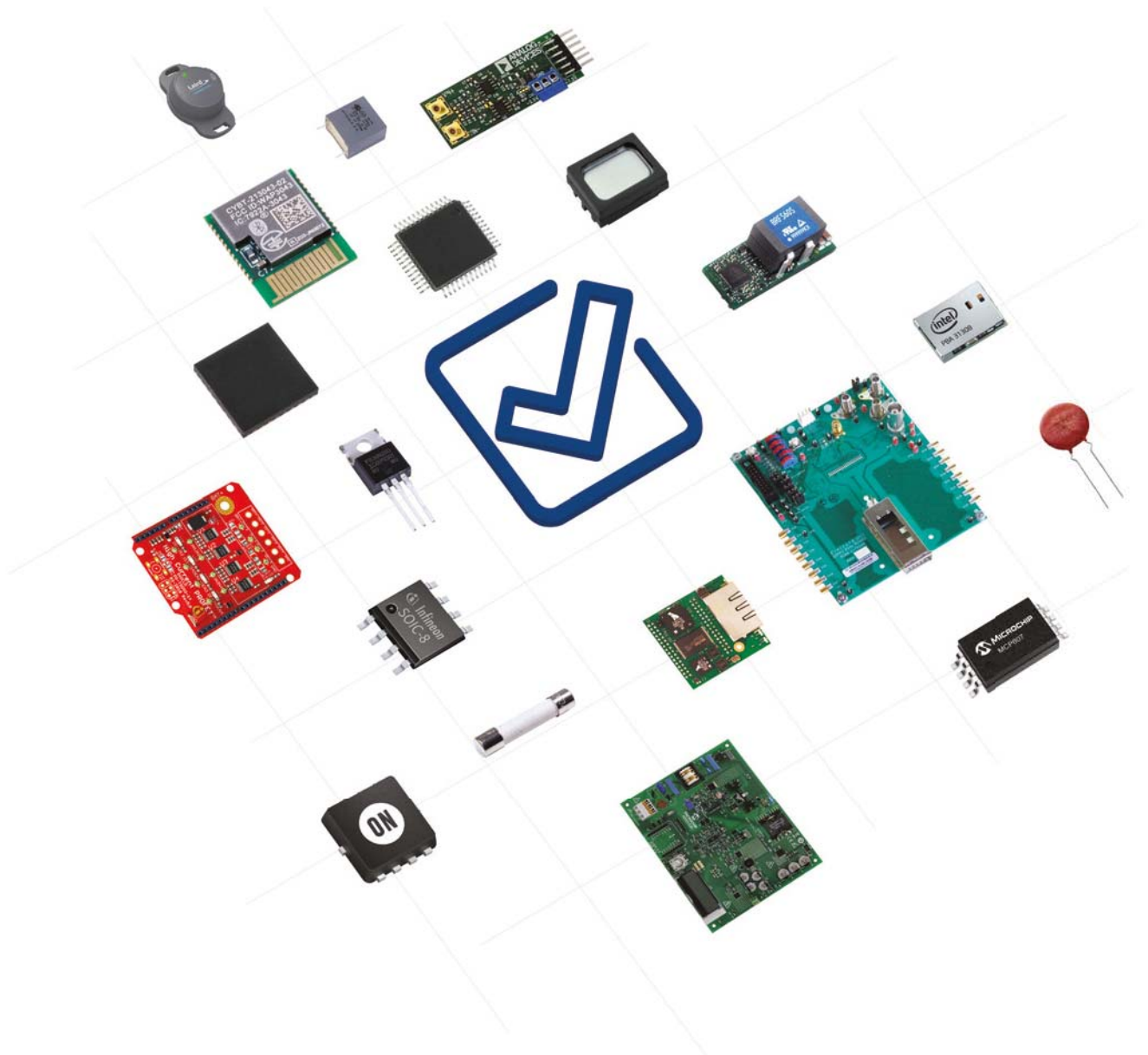
Conosciamo e utilizziamo la libreria Pytorch di Python, per poter sperimentare nuove Reti Neurali con uno strumento associato a Python.

DIDATTICA**119 Conoscere e usare Node-Red**

Impariamo ad usare Node-RED, un tool di flow-based programming orientato all'IoT ed alla connettività, originariamente sviluppato dall'IBM Emerging Technology Services team e adesso parte della JS Foundation. In questa puntata continuiamo a parlare di connettività introducendo i websockets e il protocollo MQTT. Sesta Puntata.

Garantiamo prodotti al 100% autentici

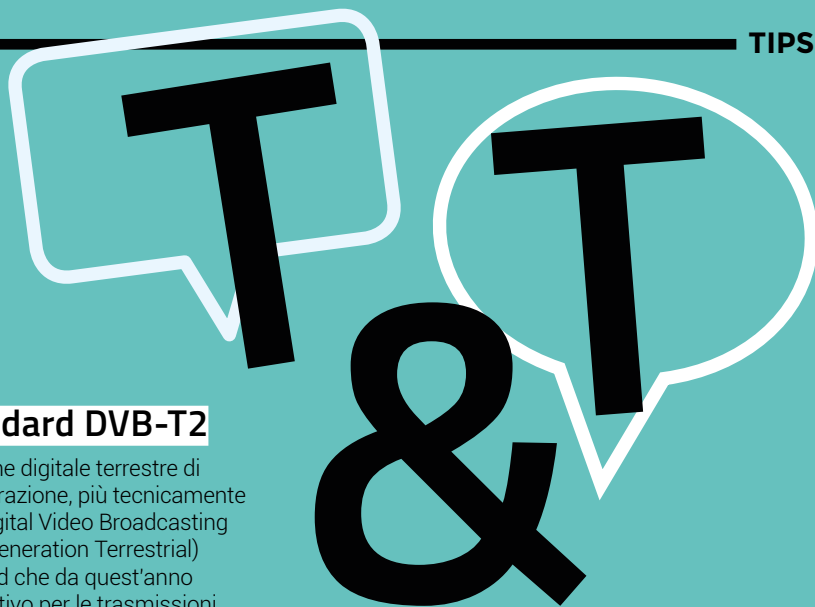
Mouser, il primo distributore autorizzato secondo la norma SAE AS6496



La più ampia selezione di componenti
elettronici a magazzino™
mouser.it/authentic-products



MOUSER
ELECTRONICS



Lo standard DVB-T2

La televisione digitale terrestre di ultima generazione, più tecnicamente DVB-T2 (Digital Video Broadcasting – Second Generation Terrestrial) è lo standard che da quest'anno diverrà effettivo per le trasmissioni sulla piattaforma digitale terrestre del consorzio europeo DVB e che sostituirà (qualcuno di voi se ne sarà già accorto) i vecchi canali TV basati sul DVB-T di prima generazione. Il passaggio al nuovo standard consentirà un miglioramento della qualità visiva e dell'alta definizione e il rilascio delle frequenze in banda 694-790 MHz (la cosiddetta "banda 700") per i servizi mobili 5G.

In Italia, per consentire un uso più efficiente dello spettro a fronte della liberazione della banda dei 700 MHz, che dovrà concludersi entro il 30 giugno 2022, è stata disposta, a partire da gennaio 2023, l'attivazione dello standard trasmissivo DVB-T2 a livello nazionale. Da tale data non sarà più possibile la ricezione delle trasmissioni televisive se non con apparecchi di nuova generazione, o abbinati ad un apposito decoder DVB-T2. Ma che cosa cambia rispetto alla prima generazione del Digitale Terrestre? Ebbene, partiamo dalla considerazione che lo scopo è liberare bande di frequenza destinate alla connettività dati e, in breve, al 5G; più bande si liberano, più larghezza di banda è disponibile, anche in prospettiva della massiccia estensione dell'IoT. Per non parlare del fatto che più bande sono disponibili, più concessioni gli Stati possono rilasciare, intascando denaro. Per mantenere lo stesso numero di canali audiovisivi (che nel Digitale Terrestre si chiamano Servizi) ed anche ampliarlo (fino a 10) comprimendo la banda, vale a dire il numero di canali RF utilizzabili, esiste solo una soluzione: cambiare la codifica di compressione, abbandonando la MPEG-2 utilizzata nel DVB-T e passando a qualcosa di più spinto, che nel caso della TV digitale DVB-T2

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.

di seconda generazione è il protocollo H.265. Lo standard H.265, noto anche come High Efficiency Video Coding (HEVC), è stato definito dal Moving Picture Experts Group (MPEG) e dal Video Coding Experts Group (VCEG). Supporta risoluzioni fino a 8192x4320 raddoppiando il rapporto di compressione dei dati rispetto all'H.264 (o MPEG-4...) a parità di qualità video. Ciò significa maggiore efficienza nello sfruttamento della banda del canale TV. Il codec H.265 è simile all'H.264 molto usato nella videosorveglianza, dato che confronta diverse parti di un fotogramma video al fine di trovare aree ridondanti, sia all'interno di ogni singolo fotogramma, sia nei fotogrammi successivi; le aree ridondanti vengono quindi sostituite con una breve descrizione invece dei pixel originali, il che consente di ridurre la quantità di dati occorrenti da far transitare nel Transport Stream. Le differenze tra H.265 e H.264 riguardano l'espansione del modello di codifica da 16x16 ad un'area fino a 64x64 pixel, oltre alla segmentazione e la previsione dei blocchi di dimensione variabile all'interno della stessa immagine, così come una maggiore previsione del vettore di movimento. Oltre a far risparmiare banda, l'H.265 richiede meno risorse di calcolo nella

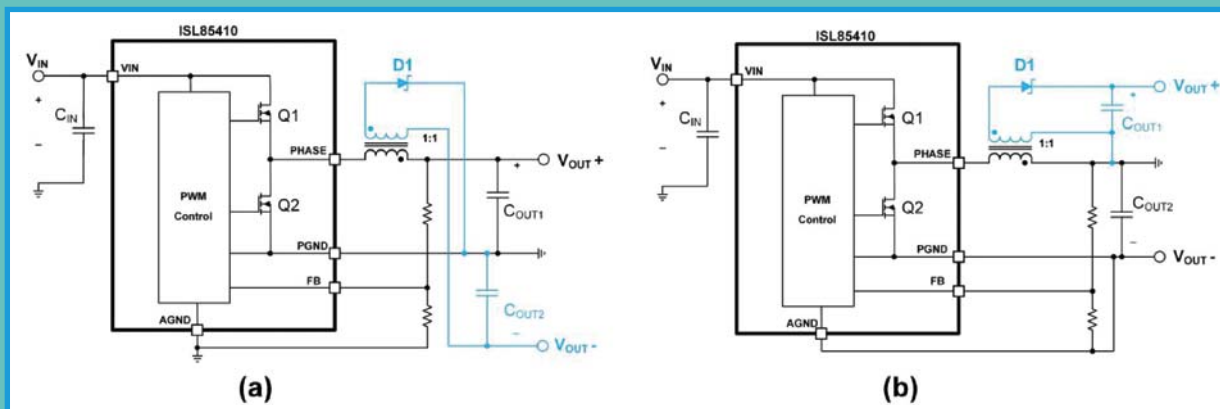
fase di decodifica.

Paragonato all'M-PEG 2 (anche noto come H.262), l'H.265 supporta una migliore qualità d'immagine e richiede 1/4 della banda occupata, con un bit-rate che scende, ad esempio, da 20 Mbps a 5 Mbps per la visione HDTV a 1.920x1.080 pixel.

Alimentazione bipolare con IC regolatore Buck

Molte applicazioni, inclusi i circuiti con amplificatori operazionali e sistemi di acquisizione dati, richiedono un'alimentazione duale del tipo $\pm 5V$ o $\pm 12V$. Un metodo comune per ottenere una tensione simmetrica a partire da una singola consiste nell'utilizzare un singolo regolatore switching insieme a un induttore accoppiato (un trasformatore) per generare una linea di tensione negativa e una linea di tensione positiva. Lo schema a inizio della pagina seguente mostra come utilizzare un convertitore buck e un convertitore buck-boost invertente per creare alimentatori duali. Come vedete, il regolatore buck ISL85410 viene prima configurato come regolatore buck che regola l'uscita positiva VOUT+, quindi l'uscita negativa VOUT- viene generata aggiungendo un ulteriore avvolgimento accoppiato. L'uscita positiva VOUT+ è regolata come in un convertitore buck, mentre l'uscita negativa VOUT-





↑ Schema semplificato dell'alimentazione bipolare utilizzando il metodo buck (a) o il metodo buck-boost invertente (b).

riflette V_{OUT+} (per semplicità si trascura la caduta di tensione diretta del diodo raddrizzatore D1) ma con polarità opposta. In fondo alla pagina vedete anche il circuito equivalente dell'alimentatore bipolare utilizzando il metodo buck durante gli intervalli di tempo di DT e $(1-D)T$; durante DT , il MOSFET Q1 del lato alto è acceso, il che porta al diodo raddrizzatore D1 polarizzato inversamente e quindi non scorre corrente nell'avvolgimento secondario. Durante $(1-D)T$, Q1 è spento e la corrente I_p fluisce attraverso il FET Q2 del lato basso. La tensione sull'avvolgimento secondario (V_s) riflette V_{OUT+} , e di conseguenza D1 viene attivato per caricare il condensatore di uscita COUT2 e fornire alimentazione al carico. Per ottenere una buona regolazione della tensione all'uscita negativa (V_{OUT-}) è opportuno configurare il convertitore in modalità CCM forzata.

Un circuito di alimentatore bipolare switching può essere implementato utilizzando l'ISL85410, un circuito integrato Renesas, del quale i parametri chiave sono: $V_{IN} = 24V$, $V_{OUT+} = 5V$, $V_{OUT-} = -5V$, I_{OUT1} e $I_{OUT2} = 100\text{ mA}$.

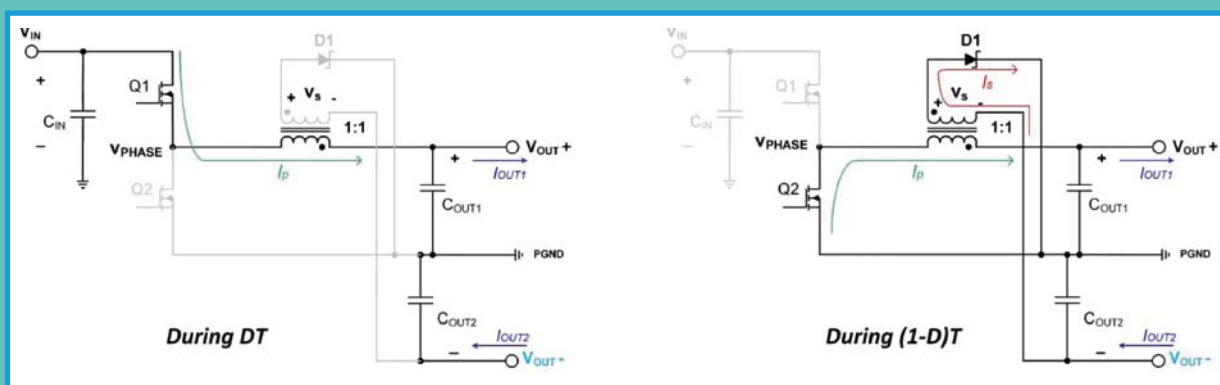
In questo caso l'integrato funziona a una frequenza di commutazione di 500 kHz. Quando il MOSFET Q2 è attivo, la corrente riflessa della corrente dell'avvolgimento secondario (I_s) fa diventare negativa la corrente primaria totale (I_p). Attraverso un'adeguata progettazione, questa corrente negativa dovrebbe essere mantenuta sufficientemente bassa da evitare di attivare il limite di corrente negativa del regolatore buck in condizioni di funzionamento normali.

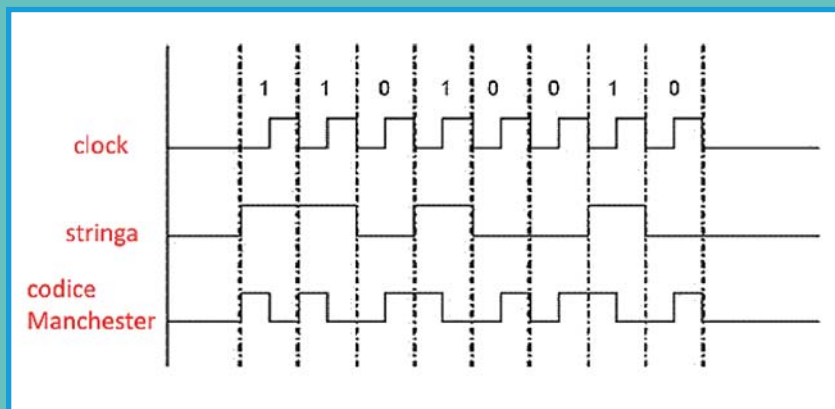
Codifica Manchester tramite microcontrollore

La codifica Manchester consiste nel costruire un bit con la commutazione del livello logico all'interno del semiperiodo corrispondente ad esso; il tutto ai fini della comunicazione seriale, che viene sincronizzata da un segnale rettangolare, che varia quindi da un livello logico alto ad uno basso e viceversa, ciclicamente e a frequenza ben determinata. Nella codifica Manchester, in pratica, in un semiperiodo dell'onda rettangolare viene sempre trasmesso un bit, il

cui valore è 1 se la tensione, nell'arco di detto semiperiodo, passa da 0 ad 1 logico, ovvero zero se nel semiperiodo si opera una transizione 1/0. La codifica Manchester è del tipo auto-sincronizzante (non necessita di un segnale di sincronismo esterno) e quindi permette un'accurata sincronizzazione dei dati seriali. La codifica Manchester fornisce un modo semplice per codificare sequenze binarie arbitrarie senza mai avere lunghi periodi di tempo privi di transizioni di clock, il che permette di prevenire la perdita della sincronizzazione del clock, oppure errori di bit causati da derive in bassa frequenza su collegamenti analogici poco equalizzati. Uno degli utilizzi più noti della codifica Manchester è nelle reti Ethernet, ma è diffusa anche nel bus DALI. Nel caso si voglia utilizzare un microcontrollore per generare dati secondo la codifica Manchester, bisogna cambiare lo stato del pin di uscita e utilizzare un timer interno che generi un' interrupt ogni $T = 1/2$ bit, ossia il tempo di mezzo bit ed è l'istante dove occorre modificare la fase del segnale come mostrato nel grafico in queste

↓ Circuito equivalente di un alimentatore bipolare con la configurazione buck.





↑ Codifica Manchester della stringa 11010010.

Web Forum e supporto tecnico

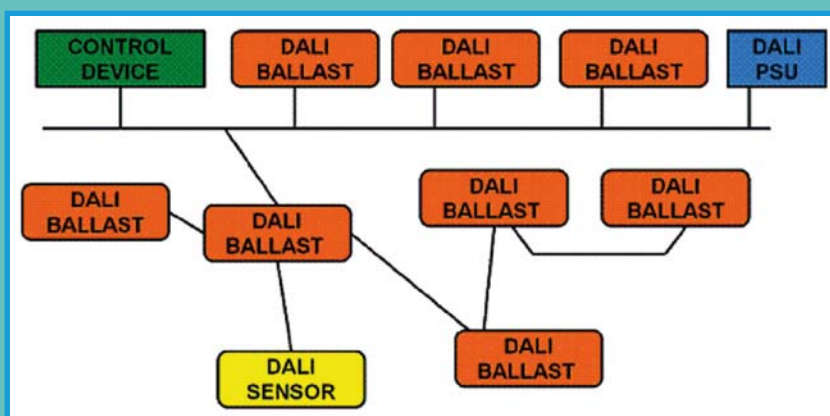
Hai un problema con uno
dei circuiti pubblicati?
Vorresti effettuare una modifica?
Accedi al nostro Web Forum
dove i nostri tecnici
(ma anche gli altri lettori)
ti aiuteranno a chiarire qualsiasi
dubbio di natura tecnica.
Collegati a:

www.elettronicaain.it/webforum

pagine. Se vogliamo mandare un bit 1, per la prima parte del bit il pin di uscita è posto ad livello alto e all'interruzione (interrupt) il predetto U/O viene posto a livello basso. Viceversa se si vuole mandare uno "0". Come risultato, sul pin di uscita del micro è presente una sequenza di uni e zeri che rappresentano la codifica Manchester della stringa di bit che si vuole generare. Il grafico in questa pagina mostra come la stringa 11010010 viene codificata nel protocollo Manchester.

Il protocollo DALI

Acronimo di Digitally Addressable Lighting Interface, il DALI è un'interfaccia digitale indirizzabile per il controllo dell'illuminazione elettrica; rientra nella specifica IEC 60929 e nasce per il controllo digitale dei reattori per lampade fluorescenti dimmerabili. Il bus DALI è seriale a due fili e prevede un'unità Master e più Slave; inoltre dev'essere alimentato tramite un alimentatore (DALI power supply) dedicato e specifico. L'unità Master invia pacchetti di dati a 16 bit codificati Manchester, cui le unità Slave (i reattori) possono rispondere con pacchetti a 8 bit codificati Manchester anch'essi. Sono previste architetture Multi - Master e due Master possono controllare uno stesso Slave. Il DALI ha 64 canali o indirizzi individuali, disponibili per i reattori che sono collegati ad una stessa linea. Ogni bus DALI può avere 16 gruppi e prevede 16 scenari. Nel Protocollo DALI non vi è alcun controllo di errore nei pacchetti trasmessi. Un sistema DALI può assumere forme diverse di topologie come Daisy Chain, a stella e Multidrop e supporta combinazioni tra due o più topologie. La figura in alto in questa pagina esemplifica una composizione di sistema DALI. Una periferica



↑ Esempio di sistema DALI a struttura libera.

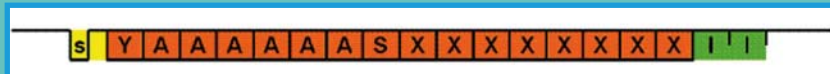
connessa a un bus DALI identifica come livello logico basso (stato attivo) una tensione minore di 6,5V, mentre lo stato alto (IDLE) corrisponde a una tensione compresa tra 9,5 e 22,5V (tipicamente 16V). L'intervallo tra i 6,5V e i 9,5V può essere considerato un three-state. La corrente massima fornita dall'alimentatore DALI è limitata a 250 mA e il tempo di risposta del limitatore deve essere inferiore ai 10 ms. Ciascuna interfaccia collegata al bus DALI può assorbire al massimo 2 mA e i connettori d'ingresso non sono polarizzati. Una periferica DALI è solitamente otticamente accoppiata al bus e può lavorare con una velocità di trasferimento dati di 1,2 kbps ($\pm 10\%$). In virtù della ridotta velocità di comunicazione (1,2 kbit/s) il cablaggio dei sistemi DALI può essere effettuato con comuni cavi elettrici a due fili, come cavi

18 AWG, di classe 1 o 2 (rigidi o flessibili). La massima distanza tra due unità è 300 metri ed è tollerata una caduta di tensione fino a 2V tra l'alimentatore DALI e ciascun componente del sistema. Per lo scambio dei dati, il protocollo prevede l'invio di un bit di start (bit 1) seguito da una serie di bit di informazione (indirizzi o dati) seguito da due bit di stop. Il bit di stop corrisponde allo stato di IDLE per un tempo di 833,33 μ s. La comunicazione avviene per Frame e può avere luogo in due direzioni:

- Forward frame; pacchetti inviati dal dispositivo di controllo alla periferica (Slave);
- Backward frame; pacchetti inviati dalla periferica al Master.

L'immagine qui in basso schematizza il Forward Frame.

↓ Struttura del Forward Frame nel protocollo DALI.



Nuove Telecamere

Full color technology



Le telecamere Full-color sono dotate di una nuova tecnologia che permette di ottenere immagini a colori anche in condizioni di scarsa luminosità, sia di giorno che di notte.

TELECAMERA FULL COLOR 2 MPX



€ 52,00

cod. FR816N

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

€ 69,00

cod. FR816VAR

TELECAMERA FULL COLOR 2 MPX - VARIFOCAL



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

TELECAMERA DOME FULL COLOR - 2 MPX

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

€ 38,00

cod. FR817



Trasformano
la notte 
in  giorno!

NB IoT 4 Click è una scheda Click Board di MikroElektronika, rivolta all'utilizzo in aree geografiche con una copertura NB-IoT inferiore al 100%, che consente implementazioni LPWAN utilizzando 2G e passa a NB-IoT quando disponibile. Include un modulo di comunicazione wireless cellulare con SIM digitale integrata (eSIM) e nasce per applicazioni IoT come Smart Metering, dispositivi indossabili intelligenti, sicurezza e tracciamento delle risorse, elettrodomestici e monitoraggio ambientale. Il cuore della scheda è il modulo di comunicazione wireless C1-RM di Cavli Wireless, un NB-IoT (Narrow Band IoT) IoT/2G a tre bande che supporta

» MikroE offre connettività dati globale anche su 2G



le bande 3, 5 e 8 si rivolge principalmente alle applicazioni di trasmissione dati a bassa potenza. Il modulo supporta anche lo standard 3GPP Release 13 e soddisfa quasi tutti i requisiti M2M grazie alle sue dimensioni ultra ridotte. I dispositivi hanno un consumo energetico in standby inferiore a 1 mA e in modalità di sospensione profonda può arrivare a 5 µA. Vengono fornite interfacce esterne, stack di protocollo e una gamma completa di servizi di trasmissione dati e SMS per soddisfare le richieste lato client.

NB IoT 4 Click è supportato da una libreria compatibile con mikroSDK e si integra nei sistemi mikroBUS. www.mikroe.com

» OpenVINO: Intel aggiorna l'AI

Da quando OpenVINO è stato lanciato, nel 2018, Intel ha consentito agli sviluppatori di accelerare le prestazioni di inferenza dell'Intelligenza Artificiale, partendo dall'edge ed estendendosi al cloud. Ora ha reso disponibile l'ultima versione della distribuzione Intel di OpenVINO Toolkit, le cui funzionalità includono una più ampia selezione di modelli di deep learning, più portabilità del dispositivo e prestazioni di inferenza più elevate con meno modifiche al codice, ma anche il rilevamento automatico dell'hardware e l'ottimizzazione automatica, in modo che gli sviluppatori di software possano ottenere prestazioni ottimali su ogni piattaforma.

Il toolkit Intel Distribution di OpenVINO, costruito sulla base di oneAPI, è una suite di strumenti per il deep learning ad alte prestazioni, mirata a risultati del mondo reale più rapidi e accurati distribuiti nella produzione su diverse piattaforme Intel, dall'edge al cloud. L'edge AI sta consentendo casi



d'uso in tutti i settori, dalla produzione e dalle applicazioni sanitarie e delle scienze della vita alla vendita al dettaglio, alla sicurezza e alla protezione; e si assiste alla crescente necessità di inferenza AI lato edge, perché riduce la latenza e la larghezza di banda e migliora le prestazioni per soddisfare le richieste di elaborazione sempre più

critiche in termini di tempo dei dispositivi e delle applicazioni Internet of Things (IoT) emergenti.

Realizzato per gli Intel Core di 12ª generazione, OpenVINO ora supporta l'architettura ibrida per offrire miglioramenti per l'inferenza ad alte prestazioni su CPU e GPU integrata.

www.intel.com

› IoT più prestante con il BLE Renesas

Renesas ha sviluppato due transceiver RF a 2,4 GHz che supportano lo standard di Near Field Communication a bassa potenza su Bluetooth Low Energy (LE). Oltre ad essere compatti, economici ed efficienti dal punto di vista energetico, i dispositivi IoT devono fornire un supporto flessibile per Bluetooth LE indipendentemente dal formato di implementazione; i dispositivi Renesas possono soddisfare questi requisiti grazie a una tecnologia di circuito di adattamento che copre un'ampia gamma di impedenza e consente al transceiver di adattarsi a una varietà di impedenze di antenne e schede senza un circuito di adattamento di impedenza esterno. Inoltre integrano una tecnologia di correzione del segnale per segnali di riferi-

mento generati localmente che utilizza un piccolo circuito per correggere automaticamente le incongruenze negli elementi del circuito e le variazioni nelle condizioni circostanti senza calibrazione. Renesas ha verificato l'efficacia di queste tecnologie su un prototipo di circuito RTX Bluetooth LE RF costruito con un processo CMOS a 22 nm, riducendo l'area del PCB a 0,84 mm² (alimentatore compreso). Ciò è stato ottenuto modificando l'architettura del ricevitore per ridurre il numero di induttori e apportando miglioramenti come un amplificatore in banda base a bassa corrente con una piccola area di montaggio e un amplificatore in classe D altamente efficiente. I transceiver assorbono appena 3,6 mW e 4,1 mW rispet-

› Potenti soluzioni Intelligent-Edge da SuperMicro

Super Micro Computer annuncia nuovi server che incorporano i più recenti processori Intel Xeon D-1700 e Intel Xeon D-2700. Le linee di server E300, 110D e 510D sono progettate esclusivamente per ambienti industriali in cui la potenza e le prestazioni dei processori Intel Xeon D sono necessarie per soddisfare gli SLA più esigenti.

Questi nuovi server basati su processori Intel Xeon D sono ideali per le soluzioni telco, industriali o intelligent edge (Edge AI). I sistemi offerti da SuperMicro hanno fino a 512 GB di DRAM, numerose opzioni di I/O e storage e

supportano PCI-E 4.0. I server Building Block Solutions offrono soluzioni ottimizzate progettate per sfruttare appieno i processori Intel Xeon D.

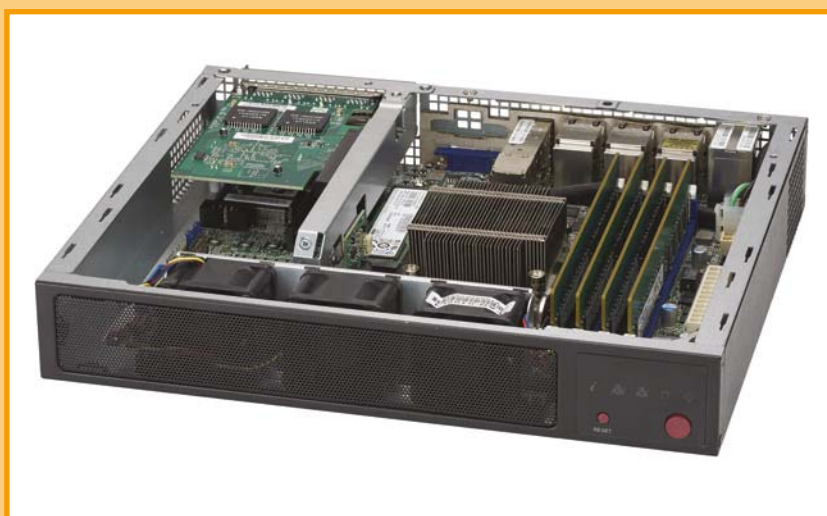
Il SYS-E300 incorpora un processore Intel Xeon D con 4, 8 o 10 core e fino a 256 GB di memoria DDR4 in un case mini-server 1U. Il modello SYS-510D offre maggiori quantità di storage e complessivamente 4 porte Ethernet da 1G e 2 da 25G, mentre i server SYS-110D contengono CPU con un massimo di 20 core e 512 GB di memoria. Ogni sistema include il processore Intel Xeon D ed è il più innovativo System-on-a-

Chip realizzato per edge con AI integrata, sicurezza, I/O avanzati e calcolo ad elevata densità. Inoltre, questi sistemi offrono un'elevata velocità di trasferimento dei dati e soddisfano i requisiti di edge essenziali richiesti dalle aziende. Le organizzazioni che forniscono servizi all'edge beneficeranno di questi nuovi sistemi per elaborare una crescente quantità di dati acquisiti a livello edge. Il server SYS-110D di Supermicro comprende un singolo processore Intel Xeon D, con un TDP fino a 125 W e uno slot PCI-E 4.0x16. Sono disponibili diverse opzioni di I/O ed è possibile installare fino a due unità SATA/U.2 da 2,5". Sono disponibili alimentatori ridondanti, a seconda che serva un'alimentazione AC o DC.

La linea di prodotti SYS-E300 contiene fino a due unità SATA. Le aree applicative più diffuse sono le applicazioni di rete come i firewall e gli Universal Customer Premises Equipment, che consolidano le esigenze di elaborazione e di rete in un fattore di forma compatto e facile da usare.

SuperMicro è specializzata nelle soluzioni IT totali ottimizzate per le infrastrutture Enterprise, Cloud, AI e 5G Telco/Edge IT, ma anche di soluzioni IT con sistemi server, Intelligenza Artificiale, storage, IoT e switch, software e servizi.

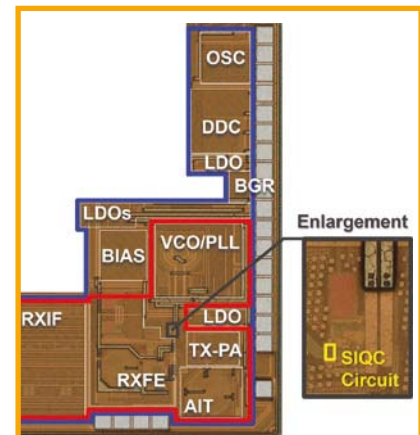
www.supermicro.com



tivamente durante la ricezione e la trasmissione. La tecnologia di adattamento dell'impedenza variabile consiste in due induttori e quattro condensatori variabili; gli induttori lato e lato ricevitore sono configurati in una disposizione concentrica e la loro induzione reciproca viene utilizzata per ridurre la perdita di segnale e tagliare la capacità parassita effettiva. I test hanno restituito un rapporto di onde stazionarie di tensione (VSWR), che indica disallineamenti di impedenza, equivalente a un massimo di 6,8 e un intervallo di impedenza variabile da

circa 25 a 300 Ω . Per elevare la precisione del segnale di riferimento per la conversione di frequenza, Renesas ha sviluppato una tecnologia di circuito di correzione di fase auto-IQ che utilizza segnali di riferimento di quattro fasi diverse per correggersi a vicenda consentendo alle differenze di fase di annullarsi a vicenda. Il rapporto di reiezione del segnale dell'immagine, cruciale per le prestazioni di ricezione, è in media di 39 dB, che soddisfa lo standard Bluetooth con un comodo margine di riserva.

www.renesas.com



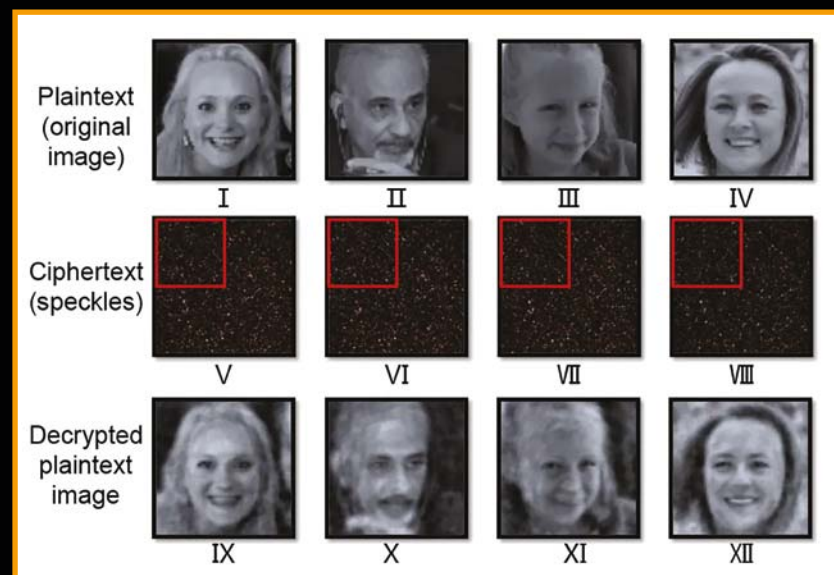
› Crittografia ottica del riconoscimento d'immagine con la tecnica a vetro smerigliato

Il riconoscimento facciale è ormai molto comune, ma ciò sta portando a crescenti preoccupazioni a causa della possibilità che i dati delle immagini possano finire nelle mani sbagliate. Questo perché i sistemi di crittografia basati su software adottati per la protezione, basati dallo scambio di chiavi tra mittente e destinatario per decifrare i dati, implicano dei vincoli, giacché se è vero che più lunga è la chiave, maggiore è la sicurezza, chiavi complesse impongono la necessità di una maggiore potenza di calcolo. Inoltre i computer quantistici di potenza sufficientemente grande sono in grado di violare i sistemi crittografici software. Pertanto, i ricercatori stanno esplorando numerose alternative ai moderni crittosistemi, tra cui spiccano i crittosistemi basati su hardware e tecnologie ottiche, le quali utilizzano la luce diffratta per convertire i dati in messaggi crittografati e promettono chiavi enormemente lunghe, velocità elevate e bassa potenza di calcolo. Tuttavia, molte tecniche ottiche sono complesse per integrarsi facilmente con altri sistemi. Ora scienziati in Cina, Hong Kong e Singapore hanno sviluppato un crittosistema ottica basata su vetro smerigliato, che trasmette i dati dell'immagine facciale

attraverso pezzi di vetro sminuzzati, ottenendo puntini che possiedono macchie di luce e oscurità sparse apparentemente casualmente che vengono poi catturate da una telecamera. Il processo di scrambling è equivalente a chiavi lunghe 17,2 miliardi di bit, che è enorme anche per un computer quantistico da attaccare, affermano i ricercatori. Per decifrare i mes-

saggi, gli scienziati hanno addestrato una rete neurale su come dovrebbe apparire una serie di 19.800 immagini facciali passate da un set di vetro smerigliato. In esperimenti preliminari, hanno scoperto che la rete neurale potrebbe decrittografare le immagini crittografate con una precisione superiore al 98%.

www.polyu.edu.hk



CATTURA L'ENERGIA DEL FUTURO

L'energia pulita che ti fa risparmiare!

Pannello solare monocristallino semirigido 12V



Pannello solare monocristallino rigido 12V



Pannelli solari monocristallini

Sono composti da singoli e sottilissimi strati di cristallo di silicio che opportunamente trattato aumenta la loro conduzione. Le celle che compongono il pannello monocristallino sono facilmente riconoscibili grazie al loro colore molto scuro. La purezza del silicio che compone le celle monocristalline permette di ottenere la massima efficienza energetica e grazie a questa caratteristica è possibile installare un numero di pannelli inferiore rispetto ai tradizionali, permettendo così di occupare superfici meno estese.

L'alta qualità del materiale utilizzato permette di avere una durata maggiore del pannello.

E per dare la giusta corrente alle tue batterie...

REGOLATORI DI CARICA PER PANNELLI FOTOVOLTAICI A 12 E 24V



cod. PWMSOLAR30
€ 39,00

30A



cod. PWMSOLAR20
€ 22,00

20A



cod. FR699
€ 108,00

15A



cod. SOLAREG10
€ 16,00

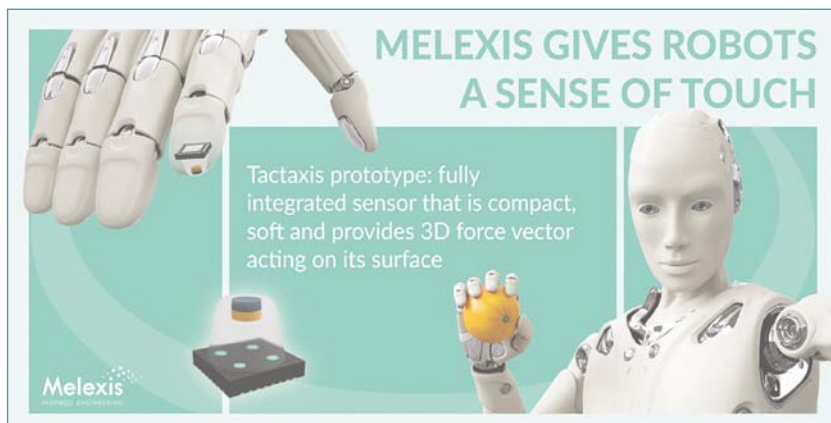
10A

Forniscono alle batterie la quantità di corrente necessaria, proteggendole da un eccesso di carica o scarica.

Da Melexis, il tatto 3D per i robot

Melexis ha presentato Tactaxis, un sensore tattile completamente integrato, compatto e morbido, che fornisce il vettore di forza 3D agente sulla sua superficie. Ciò migliora le mani e le pinze dei robot, rendendo possibili operazioni delicate come la raccolta della frutta.

L'innovativo prototipo presenta più pixel del magnetometro 3D, utilizzando la tecnologia Triaxis di Melexis; il sensore è accompagnato da un magnete incorporato in un materiale elastomerico con superficie di contatto morbida, che emula la pelle umana. La disposizione offre un'elevata sensibilità in modo che il rilevamento anche di piccole forze generi una risposta. La risoluzione della forza ottenuta è di 2,7 mN, sufficiente per distinguere la variazione di peso di una frazione di grammo (~ 0,3 grammi). Il prototipo Tactaxis è molto compatto (soli 5x5x5 mm) e quindi adatto a spazi ristretti.



L'approccio gradiometrico rende il sensore immune ai campi magnetici vaganti, il che previene potenziali errori di misurazione. È anche abbastanza robusto per far fronte a condizioni difficili (sbalzi di temperatura, ecc.).

I robot hanno bisogno del senso del tatto

per manipolare oggetti fragili. Proprio per tali applicazioni, è stato sviluppato il sensore magnetico con tecnologia Triaxis, che rappresenta una soluzione di rilevamento multiasse robusta e competitiva.

www.melexis.com

Sensore multispettrale XYZ da Ams-OSRAM

Ams-OSRAM presenta AS7343, il suo primo sensore che combina l'analisi del colore multicanale con la tecnologia XYZ per misurare il colore e l'intensità della luce visti dall'occhio umano. Quando integrato in colorimetri, spettrometri portatili e dispositivi di consumo, l'AS7343 offre maggiore produttività, flessibilità e funzionalità nelle applicazioni per la corrispondenza dei colori, il controllo dell'illuminazione e l'analisi spettrale. Il sensore è ideale per l'analisi del colore in applicazioni che richiedono misurazioni frequenti e accurate, come i gestori di serre e coltivazioni al chiuso, dove l'analisi del colore permette di ottimizzare le caratteristiche spettrali dell'illuminazione orticola per massimizzare sia la resa che l'efficienza energetica.



L'AS7343 è alloggiato in un case piatto da 3,1x2x1 mm. Funziona come uno spettrometro, misurando la distribuzione spettrale di potenza di una sorgente luminosa o luce riflessa.

È dotato di 12 canali di misurazione equidistanti nello spettro della luce visibile, insieme a un canale NIR (Near Infrared) e include la funzione XYZ i cui canali sono strettamente allineati allo spazio colore CIE 1931 XYZ, un modello standard della percezione del colore da parte dell'occhio umano. La capacità dell'AS7343 di produrre valori XYZ diretti (coordinate di colore) permette di scegliere se eseguire un'analisi spettrale completa o una scansione del colore più semplice.

www.ams-osram.com

Toshiba introduce un nuovo gate driver a MOSFET con dimensioni ridotte

Primo di una serie di sei dispositivi con supporto alla connessione back-to-back di MOSFET a canale N esterni, il gate driver a MOSFET TCK421G Toshiba controlla la tensione di gate di un MOSFET a canale N esterno in base alla tensione di ingresso. Il componente è adatto per configurare un multiplexer di potenza o un circuito di commutazione del carico dotato di blocco della corrente inversa in combinazione con una connessione back-to-back di MOSFET a canale N esterni. Il TCK421G incorpora un circuito a pompa di carica che supporta un'ampia gamma di tensioni in ingresso da 2,7 a 28,0V, e fornisce un'alimentazione stabile di 10 V ai terminali gate-source dei MOSFET esterni, agevolando la commutazione di correnti elevate. La tipica corrente di riposo in ingresso nello stato

di ON (IQon) è di soli 140µA, mentre la corrente in standby nello stato di OFF (IQoff) è di appena 0,5µA. Inoltre, il TCK421G include una funzione di blocco delle sovratensioni. Alloggiato in package chip scale WCSP6G, ha un ingombro di soli 1,2x0,8x0,35mm, che ne consente l'impiego in dispositivi densamente montati come i dispositivi indossabili e gli smartphone. Il TCK421G è il primo prodotto lanciato della serie che comprenderà sei dispositivi.

www.toshiba.com



Sensore di velocità A19360 Allegro per ADAS 3, 4 e 5



Allegro annuncia un sensore di velocità e distanza della ruota A19360 per utilizzo in applicazioni nei sistemi di assistenza alla guida (ADAS). L'innovativo sensore a magnetoresistenza GMR supporta le esigenze dei veicoli autonomi, che richiedono informazioni di rotazione delle ruote superiori per

un controllo accurato a bassa velocità. L'A19360 fornisce informazioni ad alta risoluzione generando eventi di uscita extra per ciclo magnetico con un protocollo speciale compatibile con le unità di controllo elettroniche (ECU). Include una modalità a 8 eventi per ciclo magnetico destinata alle ap-

plicazioni ADAS, fornendo un incremento per ogni ~5 mm di rotolito del pneumatico. Include anche una modalità a 4 eventi per ciclo magnetico che raddoppia il numero di uscite per ciclo magnetico (rispetto a un normale sensore di velocità della ruota).

www.allegromicro.com

Nuovi processori Xeon D da Intel

Il D-2700 e il D-1700 sono i più recenti system-on-chip (SoC) di Intel realizzati per la rete software-defined e l'edge IoT, con intelligenza artificiale e accelerazione crittografica integrate, Ethernet integrata, supporto per Intel Time Coordinated Computing (Intel TCC) e Time Sensitive Networking (TSN) e affidabilità di classe industriale. I nuovi processori Intel Xeon D estendono il calcolo con accelerazione oltre il data center principale, generando un'esperienza complessiva migliore per gli utilizzi e i carichi di lavoro chiave della rete e dell'edge. La serie Intel Xeon D include funzionalità specifiche per soddisfare i requisiti esclusivi del carico di lavoro della rete e dell'edge definiti dal software. Intel Xeon D-1700 è scalabile da 4 a 10 core, mentre lo Xeon D-2700 è scalabile da 4 a 20 core. I nuovi processori Intel Xeon D-2700 e Xeon D-1700 presentano fino a 4 canali DDR4 con velocità 3.200 MT/s, fino a 100 GbE nella capacità di throughput Ethernet e fino a 32/64 lane PCIe 4.0; inoltre hanno nuove funzionalità ottimizzate per il data cen-



ter, tra cui una cache di livello medio (L2) più ampia. Basati sull'architettura Sunny Cove Core, i nuovi Intel Xeon D offrono front-end migliorato, con una maggiore capacità e un migliore predittore di ramificazioni e beneficiano di miglioramenti nei buffer della linea di transizione (TLB), nell'esecuzione a thread singolo e nel prelettura. I nuovi Intel Xeon D sono

orientati ai dispositivi di sicurezza, router e switch aziendali, archiviazione cloud, reti wireless, inferenza AI e server edge: offrono prestazioni rivoluzionarie assistite da Intel Integrated Ethernet, Tecnologia Intel QuickAssist, estensioni Intel® Software Guard e accelerazione AI con Intel AVX-512.

www.intel.com

Controller a diodo per protezione da scarica inversa

Diodes presenta un controller a diodo ideale per la protezione contro la scarica inversa. Emulando un diodo ideale e pilotando un MOSFET a canale P, il controller DZDH0401DW fornisce l'isolamento sull'high-side necessario a bloccare la corrente inversa in circuiti fino a 40 V. Le prestazioni lo orientano ad

applicazioni come i raddrizzatori OR-ing utilizzati negli alimentatori swappabili a caldo e negli alimentatori ridondanti utilizzati nei server dei data center.

Combinando il DZDH0401DW con un MOSFET a bassa resistenza di ON, permette la riduzione della dissipazione di potenza rispetto alle so-

luzioni con diodo Schottky. Il DZDH0401DW rileva quando l'alimentazione in ingresso è scollegata portando il MOSFET in uno stato OFF e isolando così qualsiasi percorso di corrente di ritorno.

Il controller è in package SOT363 con dimensioni di 2,15x2,1x1 mm.

www.diodes.com



RZ/Five, la MPU con Core RISC-V a 64 bit da Renesas

Renesas annuncia le unità a microprocessore (MPU) per uso generale RZ/Five costruite attorno a un core CPU RISC-V a 64 bit. Il nuovo dispositivo RZ/Five impiega il core Andes AX45MP, basato sull'architettura del set di istruzioni della CPU RISC-V (ISA). L'RZ-Five espande la famiglia di MPU basate su CPU Arm

Renesas. RZ/Five è la prima MPU generica costruita attorno a un core RISC-V a 64 bit Andes; è orientata a fornire le prestazioni e le funzioni periferiche richieste dai dispositivi endpoint IoT. La sua frequenza operativa massima è di 1 GHz. Le funzioni periferiche includono il supporto per interfacce

multiple, come due canali Gigabit Ethernet, due canali USB 2.0 e due canali CAN, oltre ad un doppio modulo A/D. Viene fornito anche il supporto per il collegamento di memoria DDR esterna con funzioni di

controllo, correzione degli errori (ECC) e di sicurezza. Per RZ/Five è disponibile un pacchetto Linux verificato

(VLP) con Civil Infrastructure Platform (CIP), un Linux industriale che offre supporto di manutenzione per oltre dieci anni. L'ambiente di valutazione è costituito dalla RZ SMARC Evaluation Board Kit con una scheda modulo conforme allo standard SMARC 2.1.

www.renesas.com



Primo sensore di immagine di profondità da 0,5 Mpx da STM

STM presenta il sensore ToF VD55H1 che mappa le superfici tridimensionali misurando la distanza di oltre 0,5 megapixel. I sensori iToF, come il VD55H1, calcolano la distanza dagli oggetti misurando lo sfasamento tra il segnale riflesso e il segnale emesso. Questa è una tecnica

complementare ai sensori del tempo di volo diretto (dToF), che misurano il tempo necessario affinché i segnali trasmessi vengano riflessi sul sensore. L'esclusiva architettura dei pixel e il processo di fabbricazione di VD55H1, sfruttando la tecnologia in-

terna dei wafer impilati da 40 nm, garantiscono basso consumo e basso rumore. Dotato di un array di pixel BSI da 672 x 804, il VD55H1 opera con una frequenza di modulazione di 200 MHz con un contrasto di demodulazione superiore all'85% a 940 nm. Il funzionamento

multifrequenza, un algoritmo avanzato di decompressione della profondità, un basso rumore di fondo dei pixel e l'elevata gamma dinamica dei pixel garantiscono una precisione di profondità dell'1% e dello 0,1% sulla distanza.

www.st.com



Da Microchip, switch PCIe Gen ADAS

Le soluzioni di connettività ad alta velocità e bassa latenza per sistemi di calcolo distribuiti ed eterogenei sono un elemento fondamentale nelle applicazioni di guida autonoma di prossima generazione. In quest'ottica Microchip Technology annuncia i primi switch PCIe Gen 4 qualificati per automotive, del mercato. Queste soluzioni di switch Switchtec PFX, PSX e PAX forniscono funzionalità di interconnessione di elaborazione all'avanguardia per i sistemi avanzati di assistenza alla guida (ADAS).

Il portafoglio di switch Switchtec Gen 4 qualificato per il settore automobilistico offre la latenza più bassa e l'elevata larghezza di banda necessarie per collegare la CPU e gli acceleratori utilizzati nelle applicazioni ADAS rappresentano il culmine di diversi anni di collaborazione con i principali partner tecnologici di Microchip.

Gli switch Switchtec Gen 4 PCIe forniscono l'interconnessione ad alta velocità che supporta l'elaborazione distribuita, in tempo reale, dei dati critici per la sicurezza nelle architetture ADAS. PCIe sta emergendo come

la soluzione di interconnessione di elaborazione preferita per l'industria automobilistica per gli stessi motivi per cui è diventata pervasiva nel mercato dei data center. Fornisce una latenza ultra-bassa e scalabilità della larghezza di banda a basso consumo a CPU e dispositivi di accelerazione specializzati. La qualificazione delle soluzioni di Microchip per soddisfare le rigorose esigenze del mercato automobilistico è una pietra miliare significativa e NVIDIA ha collaborato strettamente con Microchip in tale progetto. Gli switch PCIe automobilistici di Microchip offrono flessibilità e programmabilità avanzata, consentendo la connettività SoC e GPU ad alta velocità all'interno della piattaforma NVIDIA DRIVE. L'interfaccia utente grafica (GUI) diagnostica ChipLink di Microchip fornisce ampi strumenti di debug, diagnostica, configurazione e analisi per una rapida implementazione da parte degli sviluppatori.

Gli switch PCIe Gen 4 qualificati per il settore automobilistico di Switchtec sono già in produzione e disponibili per l'acquisto

www.microchip.com





ABBONAMENTO
ANNUALE



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



Visita: www.elettronica.in.it/abbonamenti



ABBONAMENTO ANNUALE PER RIVISTA DIGITALE

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

COME ABBONARSI

→ **on-line**

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

→ **e-mail**

scrivendo a
abbonati@elettronica.in

→ **telefono**

telefonando a
+39-0331-752668

ABBONAMENTO ANNUALE PER RIVISTA **CARTACEA**

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

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

50% DI SCONTO per l'acquisto di un volume
della libreria "Elettronica In"

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



MINI TRAPANI per ogni esigenza

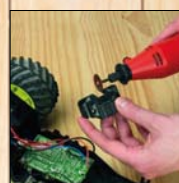
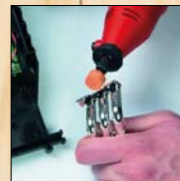


Mini Trapano con set per incisione - 162 Pezzi

cod. **VTHD0051**
€ 47,00



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



Prezzi IVA inclusa.

Mini Trapano con accessori



cod. **TP212**
€ 29,00

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

Avvitatore/trapano con torcia e impugnatura ruotabile

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



cod. **WT71880**
€ 23,90

Mini Trapano con alimentazione USB



cod. **VTSUSB5**
€ 13,90

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

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



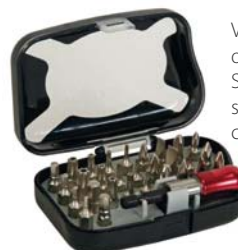
Pratica confezione comprendente 188 accessori per incidere, forare e smerigliare. Da utilizzare con i trapani della serie VTHD.

cod. **VTHDS6**
€ 22,50



cod. **VTHDS2N**
€ 5,80

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



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

cod. **VTBT17**
€ 15,00

WIRELESS

TACHIMETRO BLE PER BICICLETTE

di TIZIANO ROSSI



Basato sul sensore ad effetto Hall 49E interfacciato a un Atmega328 e programmato tramite l'IDE Microchip Studio, invia i chilometri percorsi e velocità tramite la scheda di sviluppo HELTEC WiFi kit 32 OLED.

P

er coloro che amano andare in bicicletta come passatempo, per percorrere diversi chilometri come amante delle due ruote, ma anche per chi vuol prepararsi a livello agonistico, ricorre più o meno frequentemente questa duplice domanda: quanti chilometri avrò percorso e

quale velocità massima avrò raggiunto?

A questo proposito abbiamo sviluppato un progetto che vi permette, tramite la scheda di sviluppo HELTEC WIFI kit 32 (d'ora in poi chiamata semplicemente ESP32), di visualizzare la velocità, i chilometri parziali e totali percorsi giorno dopo giorno trasmessi dalla nostra scheda tramite Bluetooth. Per ovvie ragioni il progetto può essere utilizzato per ottenere, oltre ai dati sopra riportati, la velocità periferica di tutto ciò che ha una rotazione come: ad esempio, la velocità di un motore: basta applicare un magnete per attivare il nostro sensore ad effetto Hall e il gioco è fatto.



Fig. 1
Scheda
di sviluppo
con ESP32.

Sappiamo che esistono in commercio dispositivi che permettono di ottenere tutto ciò, ma una volta acquistati dovete accettare l'impostazione della visualizzazione dei dati fatta dal costruttore, e non potete in ogni modo modificarli: si tratta di dispositivi così detti "chiusi"; mentre il nostro progetto vi permette di modificarne a vostro piacimento l'utilizzo.

Trattandosi di un progetto "aperto", con semplici passi potete modificare (aggiungere/togliere) i dati di controllo come meglio vi aggrada.

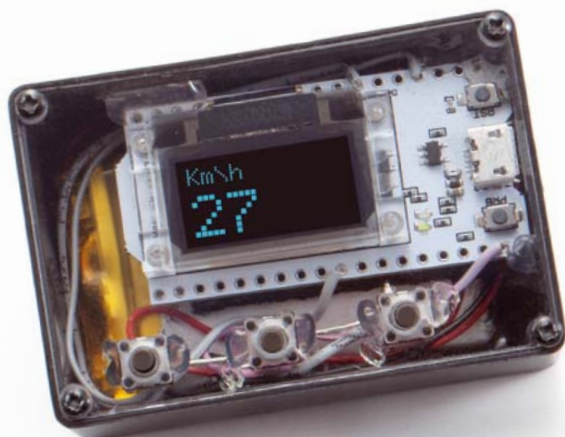
L'ESP32 (vedi **Fig. 1**) viene programmato tramite l'IDE di Arduino ed è utilizzato per controllare i dati inviati da un'ulteriore scheda che rileva la rotazione della ruota della nostra bicicletta.

Entrambe le schede sono alimentate da una batteria, per cui, in fase di utilizzo, abbiamo la necessità di avere il minor assorbimento possibile.

Per questo motivo abbiamo scelto di utilizzare moduli BLE (Bluetooth Low Energy) per la trasmissione/ricezione dati tra le due schede.

La ricarica delle batterie avviene tramite il connettore micro USB installato su entrambe le board, collegandolo a una fonte di alimentazione di 5V tramite un cavo USB, quindi possiamo tranquillamente ricaricare le batterie anche tramite un semplice powerbank.

Il circuito da installare sul manubrio, con in evidenza i tre pulsanti di impostazione.



IL DIAMETRO DELLA RUOTA

Il parametro fondamentale per il calcolo della velocità e dei chilometri percorsi è il valore della circonferenza della ruota della bicicletta.

Ma come si ottiene questo dato?

Un modo pratico, ma poco "scientifico", è quello di misurare il diametro e moltiplicarlo per 3,14 per ottenere la circonferenza. Esiste però un modo più semplice: leggere il diametro direttamente sulla spalla dello pneumatico.

I valori che troverete sono molteplici in quanto il diametro dello pneumatico può essere espresso in tre diversi standard (vedi **Fig. 2**):

- ETRTO (European Tyre and Rim Technical Organization)
- INCH (Pollici)
- FRENCH

Sulla spalla dello pneumatico sono riportati tutti e tre i parametri.

Il parametro maggiormente utilizzato è il pollice (inch), questo è il motivo per cui l'abbiamo scelto per il calcolo della circonferenza.

Il nostro progetto è stato sviluppato per selezionare e calcolare automaticamente la circonferenza, per cui la premessa che abbiamo fatto è solo per verificare che il valore restituito dal nostro programma sia congruo con i valori riportati sulla spalla dello pneumatico. Vedremo in seguito come avviene la selezione del diametro della ruota.

SCHEMA ELETTRICO DEL TRASMETTITORE

Diamo ora uno sguardo al circuito base del contachilometri, ovvero l'elettronica che misurerà la rotazione della ruota, soffermandoci su alcuni aspetti che è importante approfondire per meglio comprendere la filosofia del progetto.

Come vedete lo schema è semplice, anche grazie all'adozione di un integrato caricabatteria MCP73831, che provvede alla gestione della carica di una cella agli ioni di litio partendo dalla tensione fornita da una connessione USB (5V) e quindi da qualsiasi alimentatore con cavo dotato di connettore microUSB.

C'è poi il regolatore integrato MCP1703 (U3), che ricava i 3,3V per l'elettronica e soprattutto per l'ATmega 328 (U2).

Partiamo con il dire che lo sketch è stato sviluppato nell'IDE di Arduino per potere utilizzare la libreria chiamata "FrequencyTimer2.h" e, caricato, come vedremo più avanti, nell'IDE Microchip Studio. Dovendo essere portatile il nostro circuito è alimentato da una batteria, per cui dobbiamo fare in

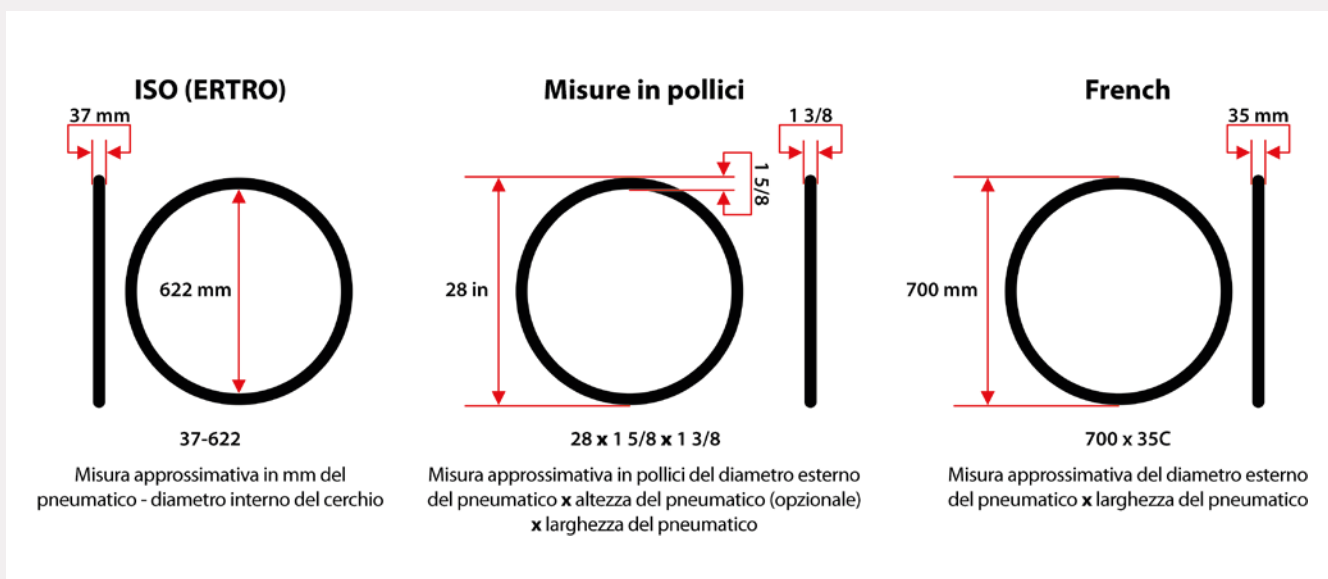


Fig. 2
Parametri
misurazione
pneumatico.

modo di ridurre il più possibile la corrente assorbita al fine di aumentare il più possibile l'autonomia. Per fare ciò dobbiamo ridurre la frequenza dell'oscillatore interno al microcontrollore da 8 MHz a 500 kHz (vedremo in seguito come): questo ci permettere di dimezzare il consumo del microcontrollore da 15mA a circa 8mA.

In secondo luogo, passati 15 minuti dal non utilizzo, spegniamo sia il sensore magnetico che il modulo Bluetooth (modalità Stand-by) tramite queste due linee di I/O:

- PIN 13 (PB1) spegne l'alimentazione del sensore magnetico ad effetto hall (on/off);
- PIN 2 (PD4) comanda il Q1 (on/off) spegne il Bluetooth.

In questo modo il consumo di corrente totale si riduce a circa 1mA.

Per riattivare tutte le funzioni del nostro PCB premete il tasto S1, uscendo dalla funzione di Stand-by.

Sul circuito sono presenti tre LED che segnalano:

- LD1 (rosso) la frequenza della pedalata;
- LD2 (verde) quando la batteria è carica;
- LD3 (rosso) che la batteria è da ricaricare.

Per ottenere i dati di velocità e km percorsi bisogna installare la scheda sulla forcella della bicicletta, ed assicurarsi che la distanza tra il magnete installato sul raggio della bicicletta (vedi **Fig. 3**) e il sensore magnetico ad effetto Hall 49E (da collegare a J4), non sia maggiore di 2 centimetri. Troppo lontano si rischia che il sensore ad effetto hall non venga attivato.

Se non utilizzate la bicicletta per un lungo periodo potete scollegare la batteria tramite lo switch SW1. Non rimane altro che caricare il firmware nel nostro microcontrollore ma, prima, approfondiamo alcuni punti del nostro sketch accennati in precedenza.

LO SKETCH

Il sensore magnetico collegato al PIN 3 di J4 genera l'impulso per il calcolo della velocità periferica. Ad ogni impulso (SALITA) corrisponde un giro completo della ruota della bicicletta, generando un interrupt che richiama la funzione "my_int", funzione che serve per ottenere il valore (conteggio) generato dal TIMER 1, per il calcolo dei dati di nostro interesse (velocità, metri percorsi, km percorsi ecc...).

Generalmente i TIMER dei microcontrollori utilizzano il clock di sistema per il conteggio degli impulsi generati nel tempo.

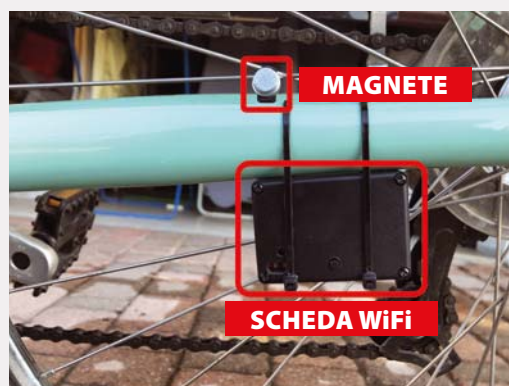
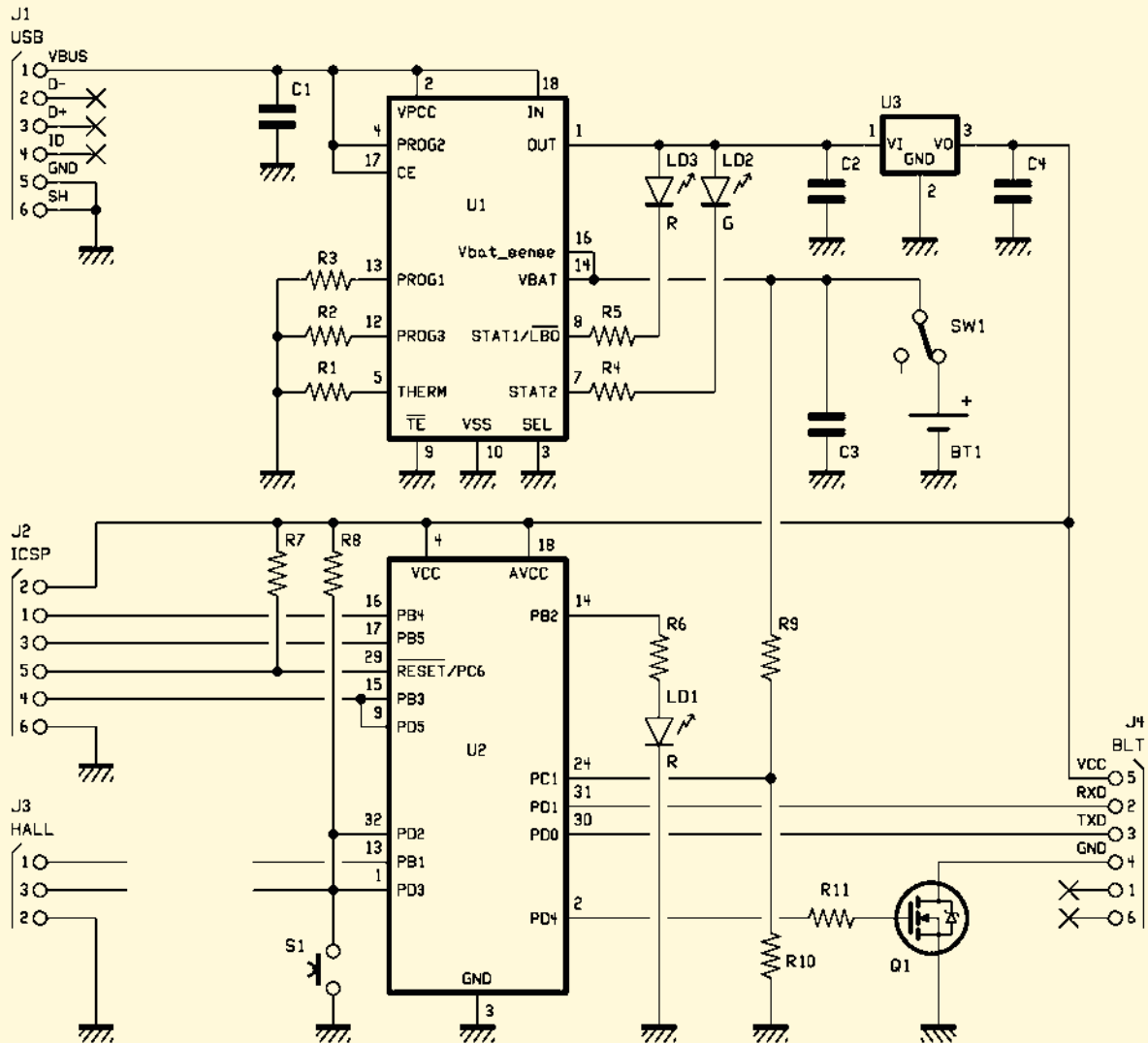


Fig. 3
Installazione
scheda.



Come anticipato in precedenza, il clock di sistema è pari a 500 kHz, ottenuto applicando un valore di "prescaler" dato dalla funzione CLKPR. Trattandosi di un TIMER a 16 bit, il valore di conteggio, pari a 65535 conteggi (FFFF esadecimale), sarebbe raggiunto in circa 130ms, per cui troppo veloce per essere utilizzato come riferimento per il calcolo della velocità periferica della ruota. Considerando che un giro completo della ruota da 28 pollici, ad una velocità di 20 Km/h, impiega circa 400ms, avremo la saturazione del valore di conteggio del TIMER 1 prima che venga completato un giro di ruota, non permettendoci di calcolare ciò che è di nostro interesse. Questo è il motivo per cui abbiamo utilizzato un clock esterno, generato dal TIMER 2, per essere collegato (PIN 9) al clock di conteggio del TIMER 1.

Utilizzando un valore di "prescaler" di 128, il TIMER 2 genera una frequenza pari a 3,9kHz. Come riportato nel datasheet dell'ATmega328AU, il clock di sistema può avere una tolleranza del 10%, tolleranza che si ripercuote sul nostro TIMER 2, per cui, per avere una lettura della velocità più reale possibile, dobbiamo avere l'effettiva frequenza generata. Se possedete un oscilloscopio vi consigliamo di verificare il valore della frequenza sul PIN 15 del microcontrollore, e riportarla nello sketch (vedi **Listato 1**) per tutte le volte che il valore è richiesto, altrimenti mantenete il valore di 3,9 KHz. La modalità stand-by viene attivata quando il sensore magnetico non genera impulsi (la ruota non gira) per almeno 15 minuti. La funzione che verrà richiamata è: "ISR (TIMER1_OVF_vect)".

In questo caso verranno spenti sia il sensore magnetico ad effetto Hall che il Bluetooth; inoltre, il microcontrollore verrà posto in modalità "Power Safe Mode": questo ci permette di metterci nella condizione di consumo minimo di corrente.

Il Bluetooth potrebbe essere messo in stand-by utilizzando la funzione (AT+SLEEP), ma abbiamo preferito togliere l'alimentazione completamente tramite il MOSFET Q1.

La funzione "wake_up" viene richiamata premendo il tasto S1 solo se la funzione stand-by è stata abilitata.

La funzione "ISR (TIMER2_COMPB_vect)" serve molto semplicemente a contare il tempo trascorso da quando si inizia a pedalare a quando si termina. In poche parole, ci dice quanto tempo abbiamo trascorso in bicicletta.

IDE MICROCHIP STUDIO

Come accennato in precedenza, per ragioni di largo

utilizzo, il programma è stato sviluppato nell'IDE di Arduino ma, utilizzando solo il microcontrollore ATmega328AU, e dovendolo programmare senza utilizzare il bootloader, lo sketch è stato caricato



Il coperchio del contenitore in plastica lavorato per dare accesso al connettore microUSB, al pulsante S1 ed all'interruttore on/off.

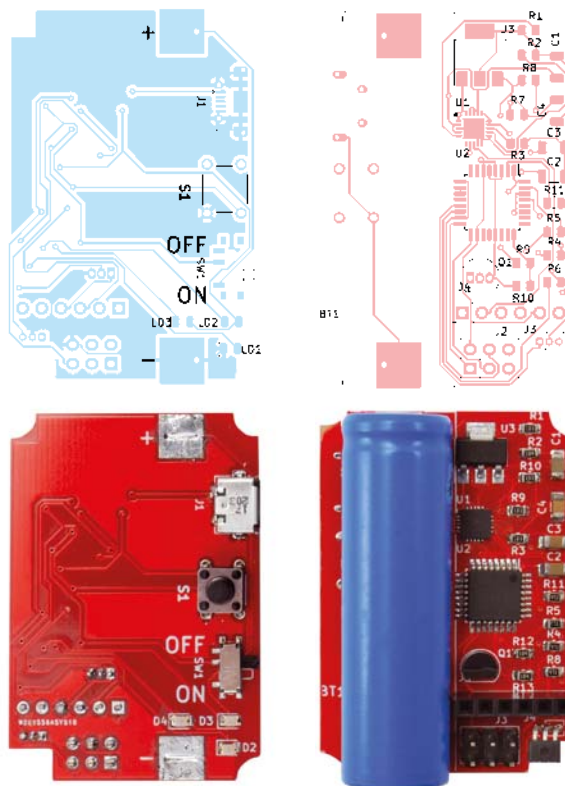
piano di MONTAGGIO

Elenco Componenti:

- R1, R3, R7, R8: 10 kohm (0805)
- R2, R9, R10: 100 kohm (0805)
- R4, R5, R6: 470 ohm (0805)
- R11: 4,7 kohm (0805)
- C1, C2, C3: 4,7 μ F (1206)
- C4: 1 μ F (1206)
- LD1, LD3: LED rosso (0805)
- LD2: LED verde (0805)
- Q1: BS170
- U1: MCP73871
- U2: ATmega328P-AU
- U3: MCP1703A
- BT1: batteria Litio 3,6 V - 800 mA
- S1: microswitch c.s.
- SW1 deviatore SMD
- J1: USB_B_Micro SMD
- J2: strip maschio 2x3 vie
- J3: strip femmina 3 vie P 1,27 mm
- J4: strip femmina 6 vie

Varie:

- circuito stampato S1584 (50x34 mm)



```

case 29:
  inch = 29;
  delay(30);

  cir = inch * 2.54 * 3.14;          //Calcolo della circonferenza della ruota espresso in centimetri
  tmr = cir / 27.77;                //27.77 = (1000mt/3600)*100 = centimetri al secondo - tempo giro ruota per percorrere 1Km/h
  n_imp = tmr * 3810;               //calcolo numero impulsi * tempo di un giro ruota (3810Hz è la frequenza generata dal TIMER 2)

break;

case 28:
  inch = 28;
  delay(30);
  cir = inch * 2.54 * 3.14;          //Calcolo della circonferenza della ruota espresso in centimetri
  tmr = cir / 27.77;                //27.77 = (1000mt/3600)*100 = centimetri al secondo - tempo giro ruota per percorrere 1Km/h
  n_imp = tmr * 3810;               //calcolo numero impulsi * tempo di un giro ruota (3810Hz è la frequenza generata dal TIMER 2)

break;

case 27:
  inch = 27.5;
  delay(30);
  cir = inch * 2.54 * 3.14;          //Calcolo della circonferenza della ruota espresso in centimetri
  tmr = cir / 27.77;                //27.77 = (1000mt/3600)*100 = centimetri al secondo - tempo giro ruota per percorrere 1Km/h
  n_imp = tmr * 3810;               //calcolo numero impulsi * tempo di un giro ruota (3810Hz è la frequenza generata dal TIMER 2)

break;

case 26:
  inch = 27;
  delay(30);
  cir = inch * 2.54 * 3.14;          //Calcolo della circonferenza della ruota espresso in centimetri
  tmr = cir / 27.77;                //27.77 = (1000mt/3600)*100 = centimetri al secondo - tempo giro ruota per percorrere 1Km/h
  n_imp = tmr * 3810;               //calcolo numero impulsi * tempo di un giro ruota (3810Hz è la frequenza generata dal TIMER 2)

break;

case 25:
  inch = 26;
  delay(30);
  cir = inch * 2.54 * 3.14;          //Calcolo della circonferenza della ruota espresso in centimetri
  tmr = cir / 27.77;                //27.77 = (1000mt/3600)*100 = centimetri al secondo - tempo giro ruota per percorrere 1Km/h
  n_imp = tmr * 3810;               //calcolo numero impulsi * tempo di un giro ruota (3810Hz è la frequenza generata dal TIMER 2)

```

Listato 1 Impostazione Frequenza TIMER 2.

nell'IDE di Microchip Studio.

Le ragioni che ci hanno portato a tale scelta sono fondamentalmente dettate dal fatto che Microchip Studio può lavorare come se l'ambiente di sviluppo fosse Arduino IDE e, soprattutto, ci permette di utilizzare il programmatore MPLAB PICKIT4. Per programmare il microcontrollore esistono diverse soluzioni, come spiegato ad esempio nel tutorial che troverete al seguente indirizzo: www.arduino.cc/en/Tutorial/BuiltInExamples/ArduinoToBreadboard, ma, per quanto ci riguarda, improponibile per microcontrollori con piedinatura TQFP montati su scheda.

Pertanto, nel nostro caso, è più funzionale utilizzare un programmatore come MPLAB PICKIT4 (programmatore della Microchip versatile e reperibile sul sito www.futurashop.it, con il codice 8200-PICKIT4).

Il programmatore MPLAB PICKIT4 ci permette di programmare una vasta gamma di microcontrollori della famiglia Microchip e microcontrollori con architettura AVR e SAM.

Come prima cosa installiamo nel nostro PC il software chiamato Microchip Studio, scaricabile dal sito di Microchip.

Una volta installato, apritelo per caricare lo sketch creato nell'IDE di Arduino; vi troverete nella pagina

di start (vedi **Fig. 4**). A questo punto dovreste fare quanto segue.

- cliccate su: "New project" e vi comparirà la finestra come in **Fig. 5**;
- cliccate su: Create project from Arduino sketch;
- in basso vi compariranno alcuni campi da compilare:
 - Name: immettete il titolo del file da creare (TACHIMETRO_bici). Automaticamente verrà compilato con lo stesso titolo il campo "Solution name"
 - Location: è l'indirizzo di dove verrà salvato il file "Name"
 - Create new solution: lasciate invariato questo campo in quanto si tratta di un nuovo file. Cliccate sul tasto OK per andare avanti.

La prossima finestra che vi comparirà è mostrata in **Fig. 6** e vi trovate le voci descritte qui di seguito.

- Sketch file: cliccate sull'icona dei tre punti per accedere alla cartella dove avete memorizzato il file Sketch che abbiamo creato con l'IDE di Arduino (file con estensione .ino). Il file selezionato verrà utilizzato per creare il nostro programma nell'IDE di Microchip Studio.
- Arduino IDE patch: si tratta del percorso dove risiede il software di Arduino.

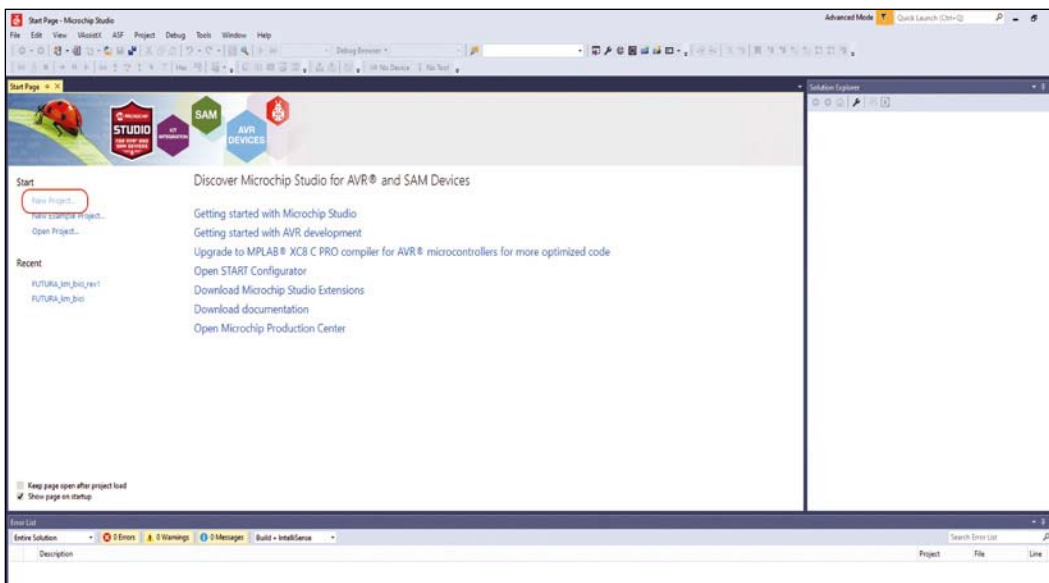


Fig. 4
Pagina
apertura
software
Microchip
Studio.

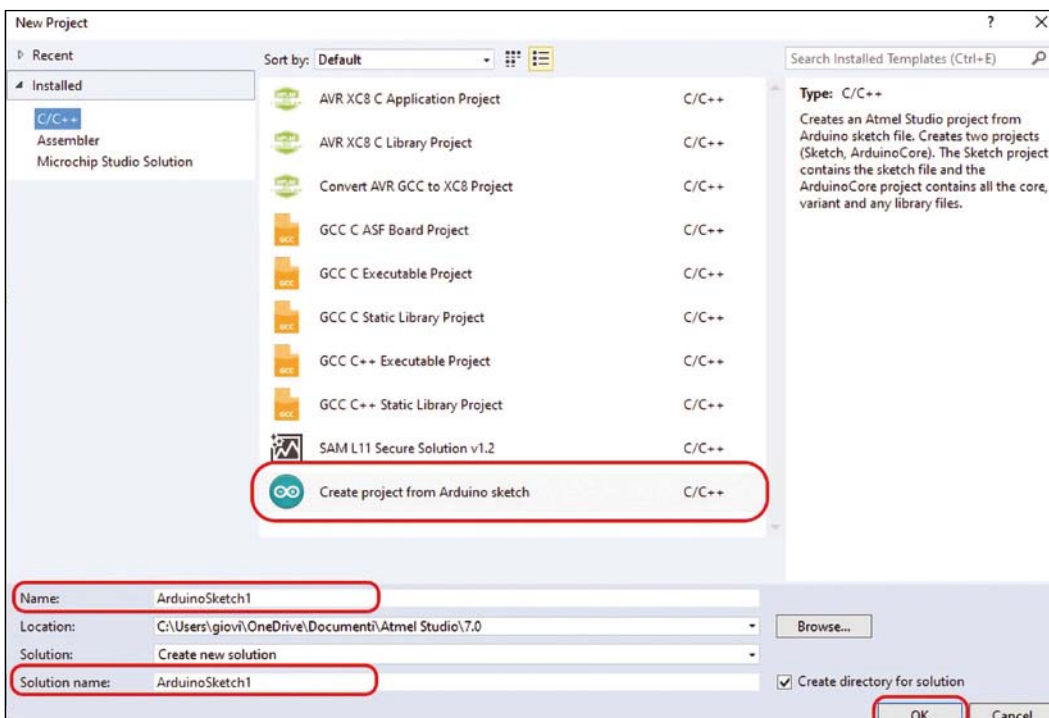


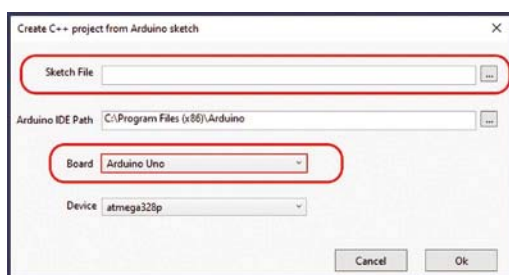
Fig. 5
Caricamento
Nuovo
Progetto.

- Board: anche se poco importante, scegliamo la scheda Arduino Uno.
- Device: ovviamente scegliamo il microcontrollore ATmega328P.

Cliccando su "OK" verrà importato il file sketch di Arduino nell'IDE di Microchip Studio. L'ambiente di Microchip Studio ha le stesse funzioni dell'IDE di Arduino, ma con maggiori peculiarità, peculiarità che vi invitiamo ad approfondire quando

avremo terminato la verifica della funzionalità del file sketch, salvato nel nuovo ambiente di sviluppo. Affinché il nostro sketch funzioni ci manca ancora un passaggio: dobbiamo caricare la libreria "FrequencyTimer2.h". Allo scopo cliccate con il tasto destro del mouse nella finestra cerchiata in rosso (vedi **Fig. 7**); si aprirà un menu a tendina dove vi comparirà: "Add Arduino Library".

Fig. 6
Caricamento
sketch Arduino.



Cliccateci sopra e, tra la lista delle librerie caricate nell'IDE di Arduino, scegliete: "FrequencyTimer2.h". A questo punto abbiamo completato tutte le operazioni che ci permettono di caricare lo sketch nel nostro microcontrollore ATmega328-AU. Per fare questo ci serviremo del programmatore MPLAB PICKit4.

CARICHIAMO IL FIRMWARE

Nella scheda è presente un connettore a 6 PIN (J3), che utilizzeremo in accoppiata con il nostro programmatore MPLAB PIKkit4 per caricare il firmware nel microcontrollore, collegando i pin come illustrato in **Fig. 8** e procedendo come segue:

- selezioniamo il tipo di microcontrollore utilizzato cliccando sull'icona posta nella barra degli strumenti, e raffigurata da un integrato cerchia-

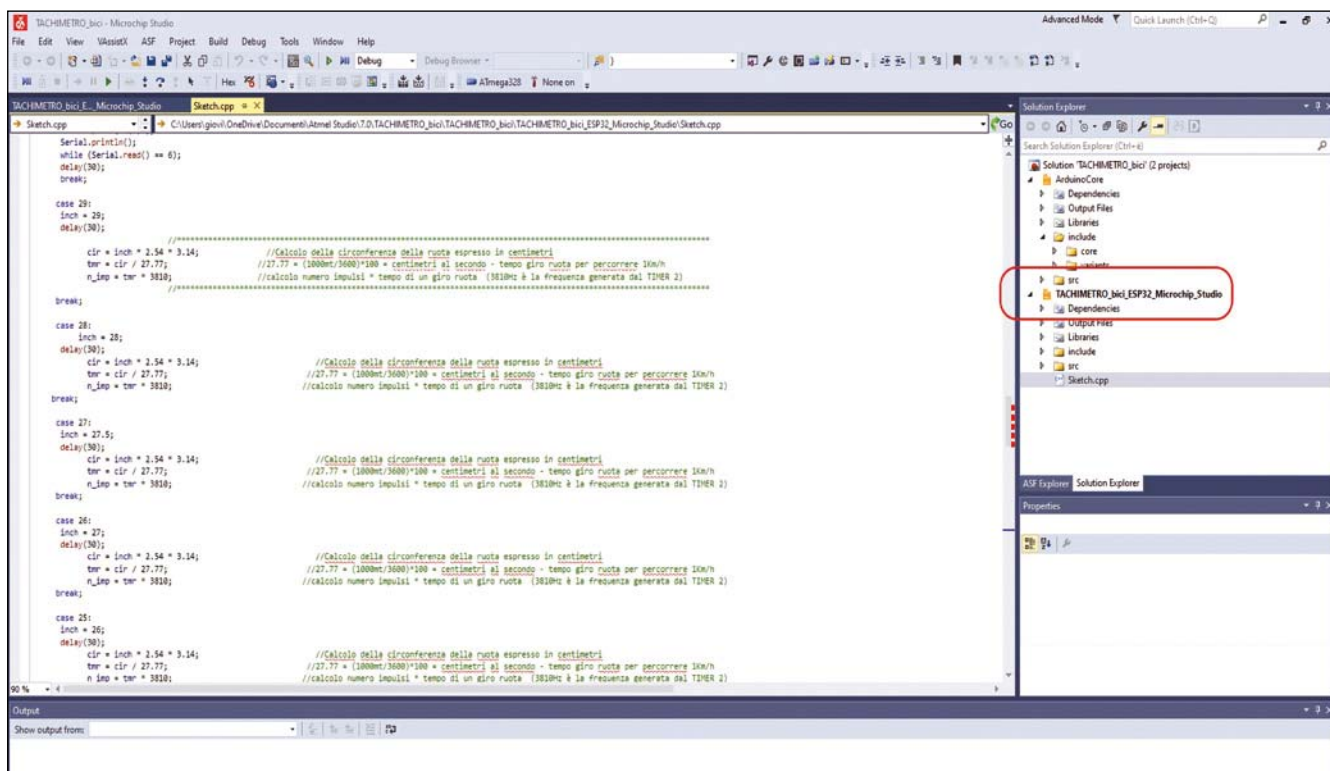
to in rosso (vedi **Fig. 9**);

- si aprirà una pagina che vi permette di scegliere il tipo di microcontrollore che stiamo utilizzando. Qualora non sia già selezionato cliccate sull'icona denominata "change device" cerchiata in verde; si aprirà un menu a tendina con una serie di microcontrollori supportati da Microchip Studio; scegliete il microcontrollore ATmega328 e cliccate su "OK". Nella stessa pagina viene riportata una lista di "programmatori" supportati (vedi lista cerchiata in viola) chiamata "Supported Tools". Cliccando sul programmatore di vostro interesse verrete indirizzati nella pagina WEB, dove potrete trovare maggiori dettagli e la documentazione a esso correlata;
- Collegate il programmatore tramite la porta seriale mini USB e selezionatelo cliccando sull'icona cerchiata in blu.

A questo punto abbiamo selezionato gli strumenti che ci servono per caricare il nostro firmware. Ora basta cliccare sull'icona rappresentata da una freccia verde (cerchiata in giallo).

Se avete selezionato correttamente tutti gli strumenti, inizierà il download del firmware nel microcontrollore. Al termine del download, nella finestra di "output" in basso (cerchiata in marrone),

Fig. 7
Caricamento
libreria
Frequency
Timer2.h.



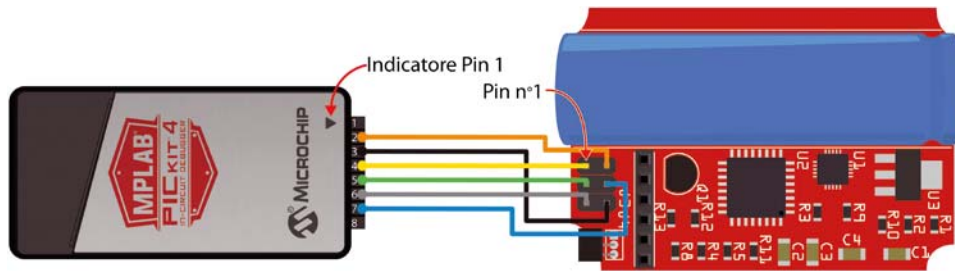


Fig. 8
Collegamento
MPLAB
PICKIT4.

vi dovete trovare la scritta "Build succeeded": significa che il download è andato a buon fine. Prima di verificare che il nostro progetto sia funzionante vediamo come configurare l'ESP32.

CONFIGURAZIONE ESP32

La scheda di sviluppo ESP32 (cod. ESP32OLED) è molto versatile e permette, come ogni scheda di sviluppo, di programmarla per svariati utilizzi, ma la cosa più importante è che la programmazione avviene tramite l'IDE di Arduino, già usato in precedenza.

Tra le periferiche integrate nel modulo ESP32 troviamo anche il Bluetooth, utile nella nostra applicazione per ricevere i dati trasmessi dalla "scheda ruota".

Per poter programmare con l'IDE di Arduino la nostra ESP32OLED dobbiamo installare come prima

MPLAB PICKIT 4					DEBUG	
Connector Pin #	Pin Name	ICSP (MCHIP)	MIPS ETAG	CORTEX SWD	AVR JTAG	AVR ISP(5W)
1	TVPP	MCLR	MCLR	MCLR	-	-
2	TVDD	VDD	VIO_REF	VTG	VTG	VTG
3	GND	GND	GND	GND	GND	GND
4	PGD	DAT	TDO	SWO	TDO	MISO
5	PGC	CLK	TCK	SWCLK	TCK	SCK
6	TAUX	AUX	-	-	RESET	RESET
7	TTDI	-	TDI	-	TDI	MOSI
8	TTMS	-	TMS	SWDIO	TMG	-

cosa la scheda.

Per tale procedura vi invitiamo a visitare il seguente indirizzo web: https://heltec.org/wifi_kit_install. Basta seguire passo dopo passo la procedura che troverete nell'Heltec Automation Docs alla sezione

Fig. 9
Selezione device
e caricamento
sketch.

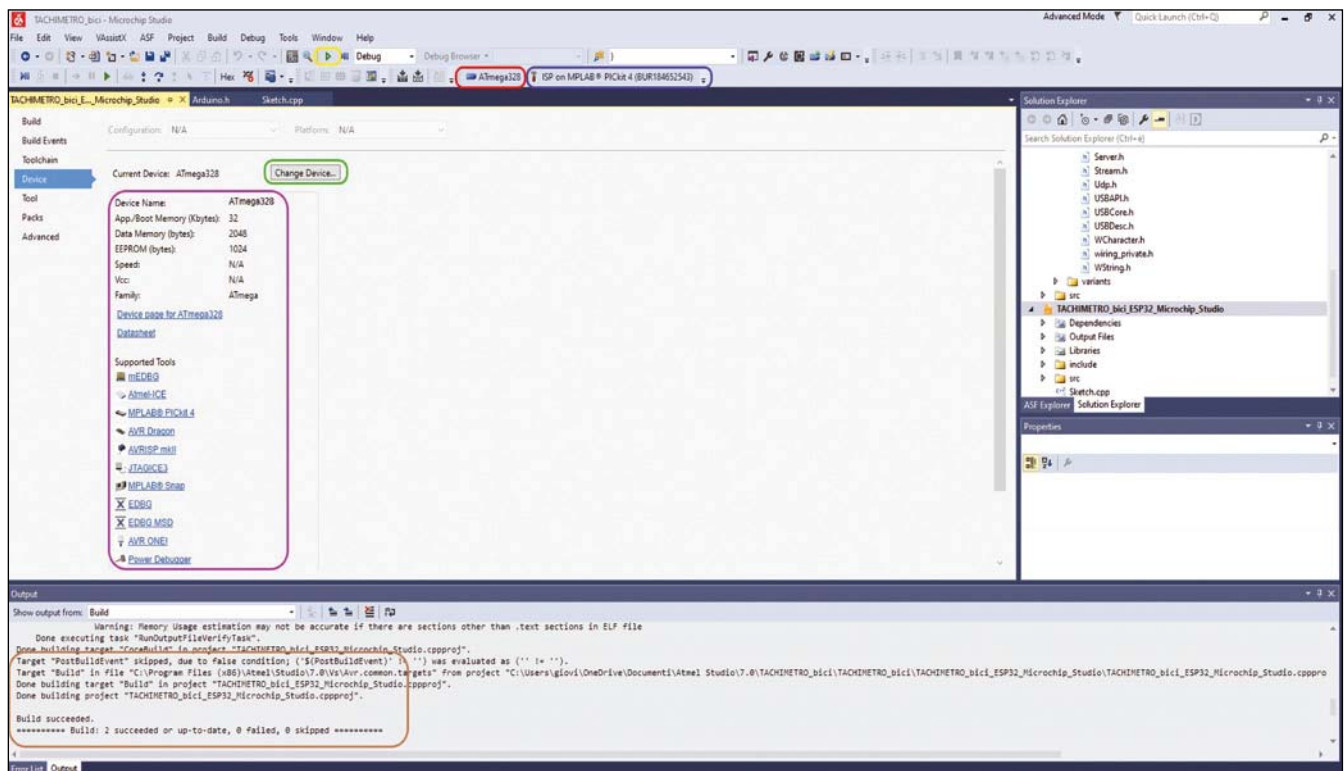
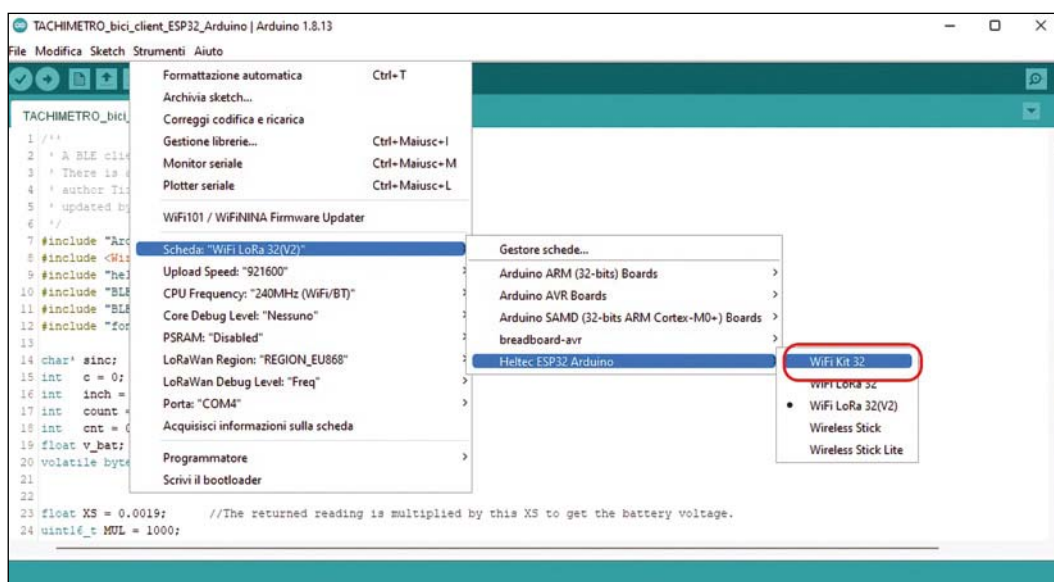


Fig. 10
Selezione scheda
su Arduino.



“Quick Start” reperibile al seguente indirizzo Web:
https://heltec-automation-docs.readthedocs.io/en/latest/esp32/quick_start.html#via-arduino-board-manager.

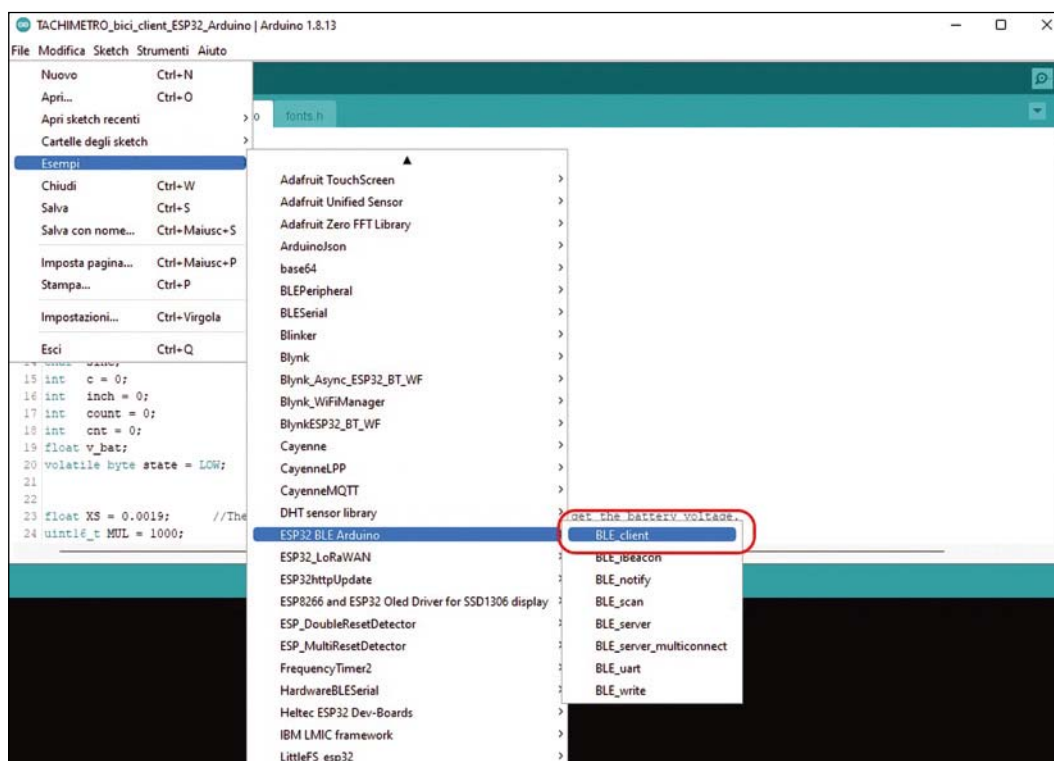
Collegata la vostra scheda di sviluppo ad una porta USB e installato quanto indicato nella sezione “Quick Start” dell’Heltec Automation Docs, potete

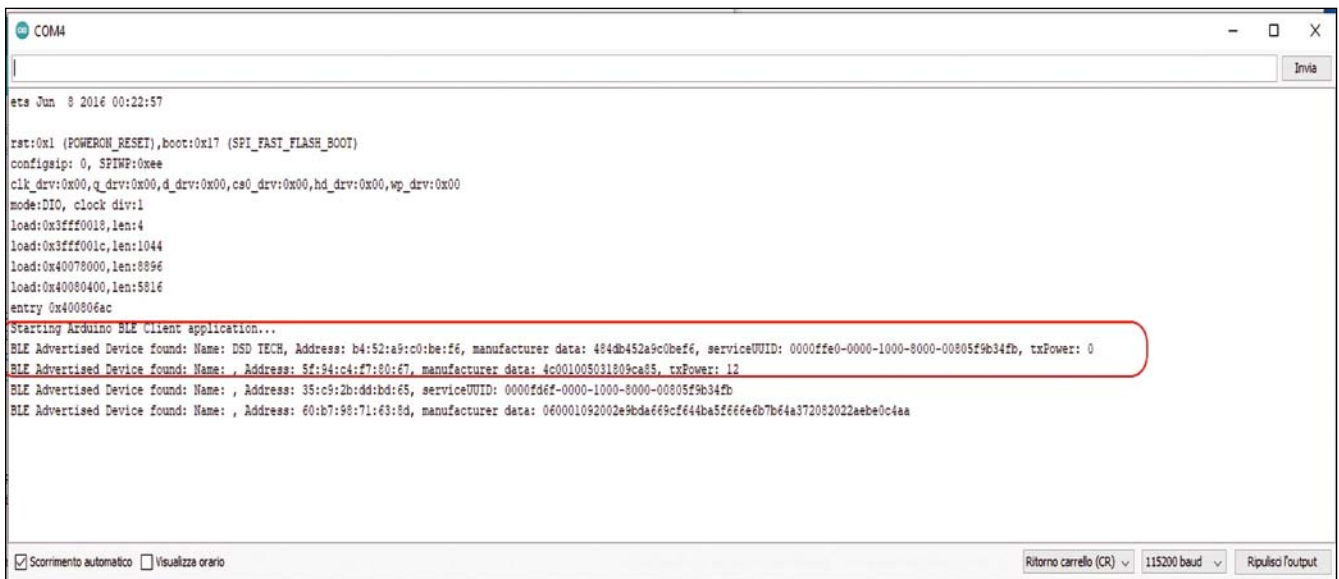
caricare uno sketch di verifica del buon funzionamento della scheda:

- Procedete a scegliere la scheda collegata alla vostra porta USB come da **Fig. 10**.
- Caricate lo sketch “BLE_client”, come da **Fig. 11**.

Una volta caricato lo sketch BLE_client nell’IDE di

Fig. 11
Selezione sketch
su Arduino.





Arduino programiamo l'ESP32 senza apportare modifiche; aprete il Serial Monitor in modo da potere leggere il service UUID (Universal Unique Identifier) del BLE a cui ci dobbiamo collegare (vedi **Fig. 12**).

Oltre al service UUID dobbiamo impostare il remote service UUID, che solitamente si differenzia di un numero. In ogni caso le sopra citate informazioni le potete verificare anche utilizzando una APP per smartphone chiamata: BLE SCANNER (vedi **Fig. 13**).

LO SKETCH PER ESP32

Dopo aver descritto lo sketch per il microcontrollore ATmega328 andiamo a parlare del codice per il modulo ESP32.

Per prima cosa dobbiamo caricare nell'IDE di Arduino lo sketch chiamato: TACHIMETRO_bici_client_ESP32_Arduino e inserire il service UUID e il client service UUID come indicato in **Fig. 14**.

Ovviamente lo sketch BLE_client ha delle funzioni che non sono state pensate per essere utilizzate per la nostra applicazione e che quindi non andremo a toccare.

Ciò che ci interessa per far comunicare il nostro ESP32 con il BLE del PCB collegato alla ruota, è nello sketch "TACHIMETRO_bici_client_ESP32_Arduino", e riguarda la parte che ci permette di collegarci come "client" ad un Bluetooth esterno. Quindi dovremo aggiungere la parte software che ci permette di ricevere i dati trasmessi dal Bluetooth HM-10 installato sulla scheda con il sensore di Hall.

I dati che verranno ricevuti dall'ESP32 sono elencati qui sotto:

- Velocità (km/h);
- Metri percorsi (Met.);
- Km Parziali percorsi (km\p);
- Km Totali percorsi;
- Tempo trascorso dall'inizio della pedalata;
- Valore della circonferenza della ruota

Fig. 12
UUID.

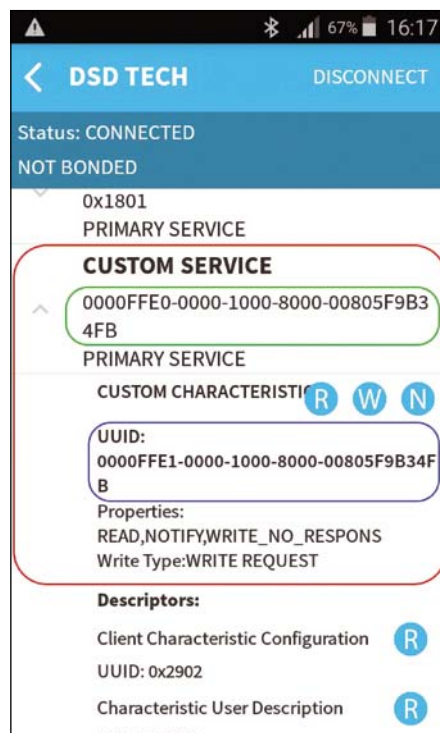


Fig. 13
App BLE SCANNER.

- Valore in Volt della batteria della scheda sulla ruota;
- Valore in Volt della batteria dell'ESP32.

Al modulo ESP32OLED abbiamo aggiunto alcuni pulsanti come mostrato in **Fig. 15**, e corrispondono a queste funzioni:

- Reset: serve a fare colloquiare la scheda con sensore di Hall alla scheda ESP32 via BLE;
- Visualizzazione dati: premendo il tasto più volte permette di visualizzare i dati sopra elencati in sequenza; mantenendolo premuto per almeno 5 secondi vi permette di entrare nella funzione per calcolare la circonferenza della ruota;
- Sleep mode: premuto in sequenza due volte mette la scheda in modalità basso consumo (stand-by); tenendolo premuto per almeno 5 secondi vi permette di entrare nella funzione della selezione del diametro ruota.

Caricate il firmware sull'ESP32 e riceverete i seguenti messaggi:

- Heltec Init!;
- Connesso (un secondo);
- Visualizzazione della velocità (Km\h).

Per semplificare la configurazione, nello sketch sono stati introdotti gli otto valori più comuni di diametro di ruote utilizzate e per selezionare quello corretto procedete come segue:

- Attivata la comunicazione tra le schede, teniamo premuto per alcuni secondi il tasto "sleep" e vi comparirà la scritta "Pronto per selezione Circonf";

- Ogni 3 secondi, in sequenza, vi comparirà il diametro da selezionare;
- Al termine di ogni visualizzazione del diametro il led (bianco) posto sulla scheda ESP32 lampeggia per 2 secondi;
- Se durante questa fase viene premuto e rilasciato il tasto "Dati" avverrà la selezione del diametro della ruota visualizzato.
- Se avete fatto tutto correttamente alla fine vi comparirà "Circonf. Selez. xxx inch"

Eseguita la sequenza sopra citata dobbiamo verificare che la selezione sia corretta.

Pedalandolo eseguite quanto di seguito riportato:

- Tenete premuto il tasto "Dati" finché non vi compare "Pronto per lettura cir.";
- Terminata la lettura vi comparirà: "Calcolo circonf.";
- Dopo un secondo verrà visualizzato il valore effettivo della circonferenza ruota;
- A questo punto abbiamo il valore della circonferenza espressa in cm calcolato dal punto di contatto con l'asfalto al punto più alto moltiplicato per 3.14.
- Confrontatelo con il dato riportato in pollici sulla spalla dello pneumatico (esempio 28 x 2.45 per ottenere il dato in cm) e se uguale, o non si discosta più del 1% / 2%, significa che il dato calcolato è corretto, per cui potete partire e ottenere i dati che ci servono.

Per concludere diamo un breve sguardo al perché abbiamo sviluppato una libreria chiamata fonts.h. Con la libreria messa a disposizione per il controllo

Fig. 14
Service UUID e
Client Service
UUID.

```
TACHIMETRO_bici_client_ESP32_Arduino  fonts.h
25
26 // The remote service we wish to connect to.
27 static BLEUUID serviceUUID("0000ffe0-0000-1000-8000-00805f9b34fb"); // HM10-11
28
29 // The characteristic of the remote service we are interested in.
30 static BLEUUID charUUID("0000ffef-0000-1000-8000-00805f9b34fb"); //HM10-11
31
32 static boolean doConnect = false;
33 static boolean connected = false;
34 static boolean doScan = false;
35 static BLERemoteCharacteristic* pRemoteCharacteristic;
36 static BLEAdvertisedDevice* myDevice;
37
38 static void notifyCallback(
39     BLERemoteCharacteristic* pBLERemoteCharacteristic,
40     uint8_t* pData,
41     size_t length,
42     bool isNotify)
43 {
44     sinc = (char*)pData;
45     Serial.print(sinc);
46     delay(10);
47 }
```

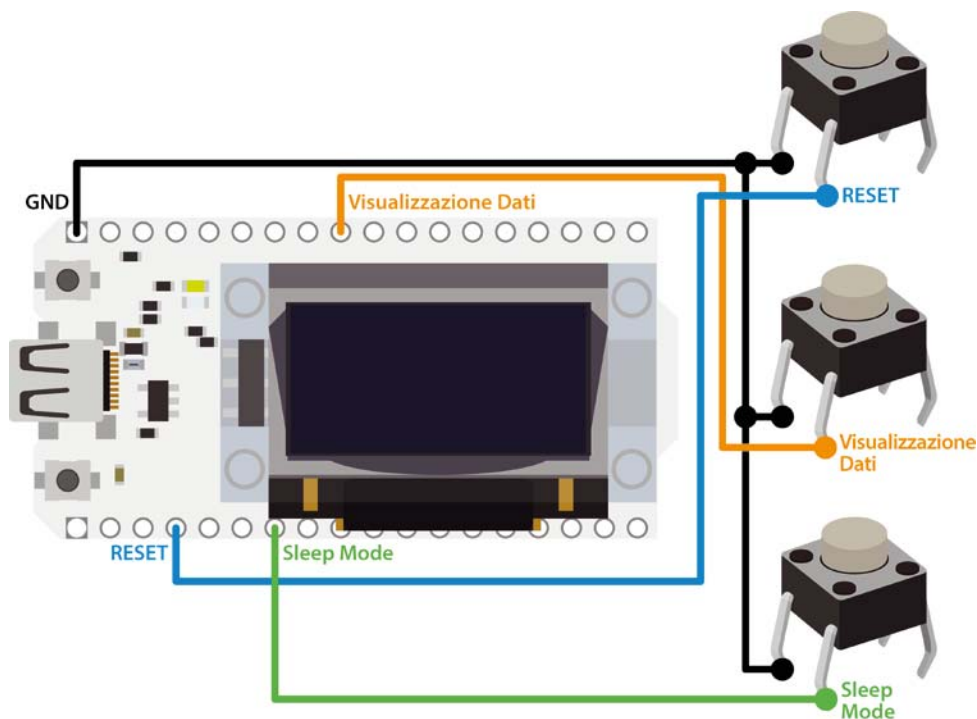



Fig. 15
Funzione
tasti.

del display OLED dell'ESP32, i caratteri di default che potete impostare hanno tre dimensioni: ArialMT_Plain 10, 16 e 24.

Durante l'uso della bicicletta la visualizzazione dei dati, diciamo, più importanti come: Velocità, Metri, Chilometri parziali, i fonts di default non sono sufficientemente grandi da essere visibili durante l'uso della bicicletta, per cui abbiamo deciso di utilizzare un font creato ad hoc e molto più grande che, come potete immaginare, si trova nella libreria fonts.h.

Mentre per gli altri dati il font utilizzato è ArialMT_Plain 24. Notate che i metri e chilometri parziali percorsi si azzerano ogni volta che la nostra scheda va in modalità "Stand-by", mentre i chilometri totali vengono mantenuti e sommati ai chilometri percorsi giorno dopo giorno.

CONCLUSIONI

Bene, termina qui la descrizione del nostro strumento combinato contachilometri + tachimetro; lo scopo di questo progetto è stato di portarvi, attraverso un'applicazione utile a molti (perché molti sono gli utenti della bicicletta...) a comprendere la logica che compone lo sketch, il funzionamento e la filosofia che c'è dietro la scheda di sviluppo ESP32, introducendovi oltretutto all'utilizzo dell'ambiente di sviluppo Microchip Studio. Partendo da qui potete approfondirne le peculiarità e le caratteristiche, per poi, in abbinamento al programmatore MPLAB Pickit4, andare a

personalizzare il firmware per il microcontrollore ATmega328 e creare nuove applicazioni; ma potete anche aggiungere/togliere dati che fanno parte del nostro sketch.

Per questo motivo abbiamo definito la nostra applicazione "aperta". Con la speranza di avervi guidato fino all'obiettivo, non ci resta che augurarvi "buone pedalate" in compagnia del vostro tachimetro e contachilometri. ■



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica.
La scheda con ESP32 e OLED (cod. ESP32OLED) è in vendita a Euro 24,00, il programmatore PICKIT4 (cod. PICKIT4) è disponibile a Euro 119,00. Le batterie al litio sono disponibili in diverse versioni sul sito www.futurashop.it. Il PCB può essere realizzato con il servizio PCBPRODUCTION all'indirizzo www.futurashop.it/produzionePCB
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>

La Presa su misura per te!

PRESA MULTIPLA SCHUKO 6 POSIZIONI CON PROTEZIONE DA SOVRATENSIONE PROTECTION



€ 21,⁵⁰

Cod. **WTPRESAM6P**

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

SET ESPANDIBILE PRESA DA RETE RADIOCOMANDATA

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

€ 22,⁹⁰
cod. **WTWRS**



Prodotti aggiuntivi al set WTWRS:

PRESA RADIOCOMANDATA ADDIZIONALE AL SET WTWRS



€ 12,⁹⁰
Cod. **RXWTWRS**

RADIOCOMANDO 4 CANALI ADDIZIONALE AL SET WTWRS



€ 9,⁰⁰
Cod. **TXWTWRS**

PRESA TIMER DIGITALE SETTIMANALE DA ESTERNO 3600W - IP44

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



€ 16,⁵⁰
Cod. **WT51301**

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



€ 11,⁹⁰
Cod. **WT40885**

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

PRESA DI RETE SCHUKO CON PROTEZIONE DA SOVRATENSIONE



€ 6,⁶⁰
Cod. **WT51270**

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

MOTION TIME LAPSE IN CLOUD

di **ANDREA CHIAPPORI**

Concludiamo l'analisi del nostro sistema basato su ESP32-CAM e Google Apps Script, vedendo il debug e l'applicazione Motion Lapse. Seconda e ultima puntata.

La tecnica time-lapse è più che collaudata, essendo stata utilizzata da parecchi anni per effettuare riprese di lunghissimi periodi con videoregistratori le cui cassette avevano durate di registrazione limitate a 240 minuti: riprendendo la scena non con continuità, ma a fotogrammi la cui periodicità si decide in base alla possibile variabilità dell'ambiente (maggiore frequenza ad esempio se si devono riprendere automobili su una strada, minore se interessa verificare lo spostamento di persone...) è possibile registrare per interi giorni cogliendo tutti i momenti rilevanti. La registrazione siffatta può essere effettuata anche con sistemi moderni, quali telecamere digitali e computer, anche microcomputer, immagazzinando i dati in storage più o meno voluminosi. Con il progetto la cui descrizione è iniziata nella prima puntata di questo articolo, vi proponiamo una solu-

zione ancora più innovativa perché consente di sfruttare la connettività wireless e Internet, per implementare un sistema di registrazione secondo la tecnica time-lapse, ma senza disporre di uno storage locale, in quanto andremo a salvare le nostre riprese direttamente su Cloud: e non su un Cloud qualsiasi, bensì su Google. Questa soluzione consentirà di realizzare la registrazione con un apparato di piccolissime dimensioni, di cui peraltro non bisogna preoccuparsi di scaricare la memoria; il nostro hardware è infatti costituito da un modulo ESP32-CAM, che abbina le funzionalità WiFi del microcontrollore ESP32.

Integriamo il Cloud utilizzando un linguaggio di scripting che Google ci mette a disposizione e con il quale potremo arricchire le nostre applicazioni in ambito domotico.

Nella prima puntata ci siamo lasciati dopo aver descritto l'hardware e la parte software che si avvale del linguaggio *Apps Script di Google* per automatizzare il tutto e vedere i video su Google Photos; abbiamo anche spiegato come, restando in ambito Google, aggiungere un file di log più dettagliato su Google Docs ed utilizzare Google Calendar come alternativa per avviare l'applicazione in maniera schedulata. Ora riprendiamo il discorso da dove lo abbiamo lasciato, ossia con una funzione di *Apps Script di Google* e del relativo ambiente integrato di sviluppo, che ci offre tante possibilità di svolgere debug e test di tutte le nostre funzioni.

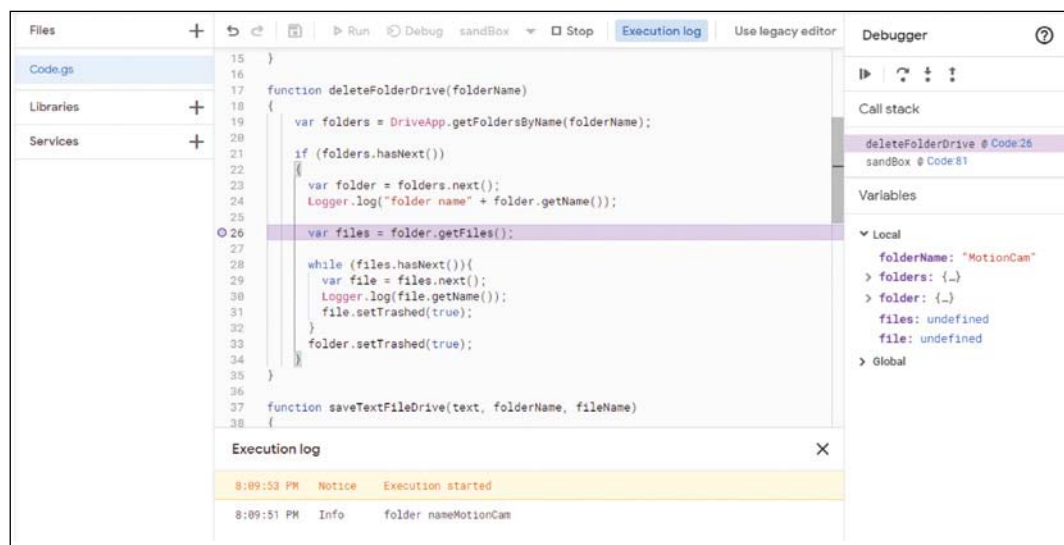
Descriveremo anche la nostra applicazione Motion-lapse, che gestirà il sistema.

Ma partiamo dunque con il debug, che avverrà direttamente on-line.

DEBUG ON-LINE

Una volta che abbiamo tutti i vari blocchetti per ricevere le immagini ad intervalli ed in fasce di tempo impostabili a piacere, non ci rimane che caricarle nella cartella di Drive e creare un video dalle nostre acquisizioni. Possiamo creare un progetto nuovo chiamato *MotionCam*, ed aggiungere tanti file esterni come gli esempi visti precedentemente, aggiungendo qualche funzionalità specifica. Avremo così, tanti file separati per ciascuna area d'azione; inoltre, cosa più importante, avremo la possibilità di testare le nostre funzioni singolarmente, attraverso il debugger integrato di Google, che pur essendo molto basilare si è dimostrato davvero funzionale. In questo modo non saremo costretti a far partire l'applicazione sul modulo ESP32 e ad eseguire ogni volta il deploy dello script, per verificare che la creazione del file sia andata a buon fine, ma possiamo testare semplicemente la creazione di un file usando uno script di esempio creato in locale, come per i listati già presentati nella prima puntata. Basterà selezionare una funzione tra quelle scritte nella casella di testo in cima all'editor (come la *sandbox* in **Fig. 1**) e premere il tasto *Debug*, per iniziare ad eseguire passo passo le linee del nostro codice, verificando il valore delle variabili sulla parte destra dello schermo ed esaminando le righe scritte dal modulo *Logger*, nella parte inferiore dello schermo. Allo stesso modo sarà possibile inserire breakpoint nel codice, spostarsi alle linee successive premendo sulle frecce laterali, avere informazioni su eventuali errori durante l'esecuzione del codice, o far avanzare il processo fino al successivo breakpoint, il tutto senza eseguire

Fig. 1
Fase di debug online.



Listato 1 - Utilizzo dei trigger.

```
function triggerCallback()
{
  Logger.log("trigger ok");
  GmailApp.sendEmail("kiand64@gmail.com", "subjectfromscript", "myBody");
}
function createTimeDrivenTrigger()
{
  ScriptApp.newTrigger('triggerCallback')
    .timeBased()
    .everyMinutes(5)
    .create();
}
function createEventDrivenTrigger()
{
  var doc = DocumentApp.openById("1d-TU_sWMsYsz6sFZJh1WIFK0xRlr7LotCJ4N69u95VQ");

  ScriptApp.newTrigger('triggerCallback')
    .forDocument(doc)
    .onOpen()
    .create();
}
function deleteAllTriggers()
{
  var triggers = ScriptApp.getProjectTriggers();
  for (var i = 0; i < triggers.length; i++)
  {
    ScriptApp.deleteTrigger(triggers[i]);
  }
}
```

il deploy dell'intero progetto. In caso testassimo funzioni che prevedono parametri in ingresso, basterà creare una funzione esterna di test (come ad esempio *sandbox()*) che chiamerà la funzione interessata e a quel punto imposteremo la sandbox come funzione su cui effettuare il debug. Se invece ci sentiamo più sicuri, possiamo cliccare anzichè sul pulsante *Debug*, sul pulsante *Run* per ignorare tutti i breakpoint, ma continuando a vede-

re i log nella finestra inferiore (*Execution log*). Quando effettuerete un debug per la prima volta, vi verrà chiesto da Google di confermare l'autorizzazione per quel particolare modulo, ma si può procedere tranquillamente.

UTILIZZO DEI TRIGGER

Apps Script permette anche l'uso di uno strumento per eseguire alcune nostre funzioni periodica-

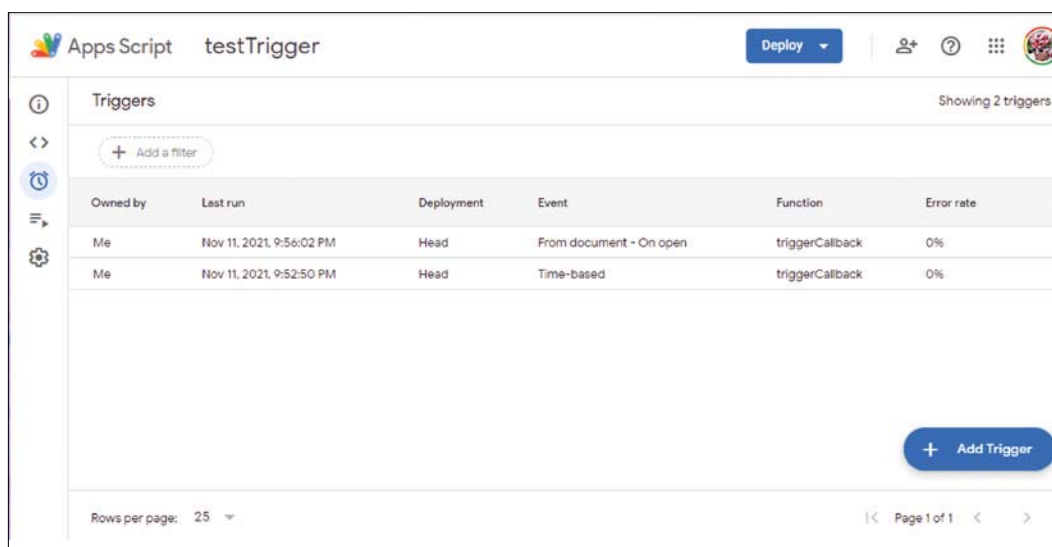
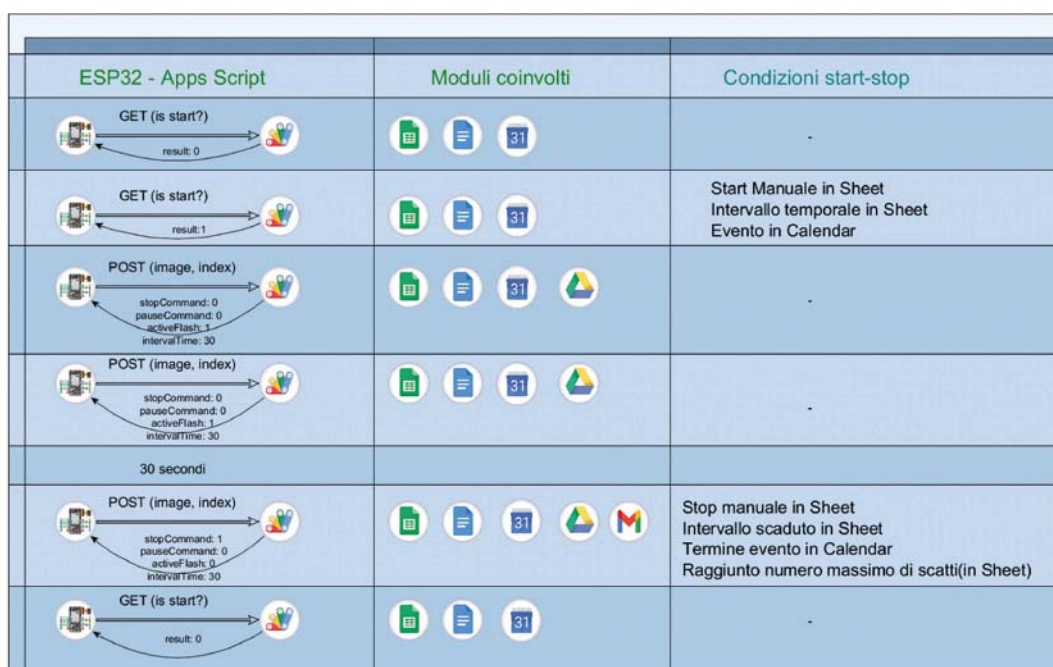


Fig. 2
Schermata
dei trigger in
Apps Script.

Fig. 3
Esempio di flusso
di un processo di
acquisizione.



mente e per questo useremo l'oggetto *ScriptApp* ed il suo metodo *newTrigger()* con il quale siamo in grado di scegliere un intervallo di tempo e la funzione che verrà richiamata periodicamente. Nel **Listato 1** basterà eseguire (in modalità *Run*) la funzione *createTimeDrivenTrigger()* che ogni cinque minuti (*everyMinutes(5)*) ci invierà una mail di test. L'uso dei trigger possono tornare utili in svariate situazioni come ad esempio per simulare un meccanismo di watch-dog o per reagire ad un evento asincrono; con lo stesso metodo *newTrigger()*, infatti, possiamo anche attivare una nostra funzione, non solo ad intervalli temporali, ma sul verificarsi di un determinato evento, come il cambiamento di un valore in Google Calendar o di una cella in un foglio elettronico o semplicemente sull'apertura del calendario o di altro documento; la funzione di trigger verrà chiamata ad ogni cambiamento generale del documento, poi saremo noi che, al suo interno, potremo controllare nelle opportune zone sensibili per la nostra applicazione. Nella nostra applicazione, ad esempio, useremo un trigger per cancellare le celle di log nel foglio elettronico; sarà un trigger sul cambiamento di tutto il foglio e nella funzione di *callback* relativa andremo a controllare se il valore della cella di nome "ClearLog" è impostato a *true* (nel caso particolare è stata usata una *checkbox* per comodità) ed in tal caso chiameremo la funzione *clearSheetLog()*. Nella funzione *createEventDrivenTrigger()*, invece, il trigger viene attivato ogni volta che apriremo un

documento Docs di Google, tra quelli presenti nel nostro spazio Cloud. Infine con la funzione *deleteAllTriggers()* che usa il metodo *deleteTrigger()* verranno cancellati tutti i trigger; i trigger possono essere ovviamente più di uno e per avere traccia di tutti quelli attivi possiamo accedere ad una pagina dedicata cliccando nell'icona dell'orologio sulla parte sinistra del browser, per accedere alla finestra come in **Fig. 2**. Da questa finestra abbiamo informazioni sui trigger, sull'ultima data di esecuzione e sulla percentuale di corretta esecuzione delle funzioni chiamate. Allo stesso modo potremmo cancellarli, modificarli e crearne di nuovi in modo manuale, cliccando sul pulsante 'Add Trigger' in basso a destra.

FIRMWARE SU ESP32

Non ci soffermeremo sul firmware per la gestione della rete WiFi e della webcam per il modulo ESP32 perchè già trattate ampiamente sui numeri della rivista, ricordiamo solamente che in fase di inizializzazione della telecamera possiamo scegliere la risoluzione delle immagini tramite la riga di codice:

```
s->set_framesize(s, FRAMESIZE_VGA);
```

sapendo che possiamo selezionare diversi valori, dal più piccolo QVGA (320x240) che potremo usare in fase di test, al più dettagliato UXGA (1600x1200) che risulterà più dettagliato anche se naturalmente più lento.

Passiamo allora a scrivere il codice per inviare le opportune richieste GET e POST alle funzioni del nostro script (*doGet()* e *doPost()*). Dal momento che *Apps Script* ci mette a disposizione uno spazio praticamente illimitato, faremo in modo che il modulo hardware esegua le operazioni più semplici, mentre cercheremo di spostare il resto della complessità lato Cloud. Useremo la libreria *HTTPRedirect* che per le GET usa una sintassi di questo tipo:

```
GET(url, host, true)
```

in cui *url* è l'indirizzo completo di richiesta, *host* è il server di google *script.google.com* ed il terzo parametro ha valore *true* se vogliamo visualizzare su debug seriale la stringa con la risposta che è comunque già presente come valore di ritorno e nel nostro caso è l'oggetto JSON. Le GET verranno utilizzate per semplici richieste mentre per l'invio dell'immagine useremo la funzione:

```
POST(url, host, payload, false)
```

in cui *url* ora è l'indirizzo completo (fornito anche in fase di deploy), ma senza nessuna stringa aggiuntiva, il *payload* è la nostra immagine ed il parametro *false* indica che non verrà visualizzata la stringa di risposta, che però possiamo sempre recuperare tramite il metodo *getResponseBody()* fornito dalla stessa libreria. L'utilizzo di questa libreria per gestire GET e POST piuttosto che la più classica *HTTPClient* è dovuta al cambiamento progettuale dei server di Google *Apps Script* che dopo qualche anno hanno incominciato a rispondere a queste richieste con un messaggio HTTP 302, ovvero richiesta di reindirizzamento ad altro server. Nello stesso messaggio è indicato l'indirizzo a cui reindirizzare, ma gestire il passaggio manualmente su un dispositivo come ESP32 non è banale, perciò abbiamo fatto ricorso a questa nuova libreria che risolve il problema; la libreria è dotata anche di stringhe di log (disabilitabili tramite una define nel *fileDebugMacros.h*) per cui siamo in grado di vedere nel monitor seriale la segnalazione 302 da parte di Google e la nostra corretta risposta. Questo reindirizzamento avviene anche per i test con *Postman*, ma in questo caso il software reindirizza le richieste al nuovo server in modo implicito senza che possiamo accorgercene. In ogni caso, sia GET sia POST ci restituiscono oggetti JSON che possiamo parserizzare più

facilmente utilizzando la libreria *ArduinoJson.h* con il metodo

```
deserializeJson(doc, jsonResult)
```

che converte la stringa di risposta (*jsonResult*) in un oggetto interno *doc* da cui poter accedere ai vari campi utilizzando una linea di codice di questo tipo

```
doc["result"]
```

in cui *result* è uno degli attributi dell'oggetto. Infine aggiungeremo al progetto anche i file per la decodifica Base64 con cui convertire l'immagine binaria in stringa.

APPLICAZIONE MOTIONLAPSE

Ci possiamo inventare ora un nostro protocollo per il trasferimento dei dati in entrambe le direzioni, perché il modulo invierà le immagini al Cloud, ma dovrà anche ricevere informazioni su quando iniziare l'acquisizione, quale intervallo usare tra un'immagine e l'altra e quando fermarsi.

In **Fig. 3** è rappresentato uno schema del flusso di acquisizione di due immagini.

Utilizzeremo le GET per effettuare richieste veloci al Cloud, mentre per l'invio della stringa contenente l'immagine, questa può avere dimensioni importanti e conviene pertanto usare le POST perché, al contrario della GET non ha limiti di dimensione.

In realtà ogni server potrebbe avere le sue regole anche per le POST e nel caso di *Apps Script* sono stati riscontrati alcuni problemi di invio già nel caso di immagini VGA (640x480) il che ha portato ad un meccanismo di spaccettamento/ricompattazione nel firmware del modulo ESP32-CAM e nella funzione *doPost()* di *Apps Script*.

Per cominciare useremo un'operazione di GET per richiedere ciclicamente al Cloud (ad esempio ad intervalli di qualche secondo) la possibilità di iniziare ad acquisire immagini; questa sarà come quella già vista negli esempi di *Postman*, ma avrà una *query string* di questo tipo *?command=isStart* dalla quale ci aspetteremo un oggetto JSON con un solo valore booleano *result* che sarà vero se è stato selezionato il comando *Start* dalla cella *Command* nel foglio elettronico, oppure se siamo all'interno dell'intervallo di tempo scelto tramite le celle *StartAt* e *EndAt* oppure ancora se abbiamo inserito un evento nel calendario Google chiamato con lo stesso nome che compare nella cella *ProjectName*.

Una volta ricevuto il consenso dal Cloud, possiamo inviare la prima immagine; quindi lato ESP32 usare-



mo la funzione `esp_camera_fb_get()` per caricare un buffer di immagine che verrà poi codificato Base64 (tramite libreria `Base64.h`) per convertirlo in una stringa di testo pronta per essere inviata al Cloud tramite richiesta POST. Tale richiesta sarà un oggetto JSON che oltre a contenere la stringa immagine avrà anche il numero sequenziale dell'immagine (`index`) ed un parametro di segnalazione `isLast` che nel caso in cui si tratti di un'immagine di grandi dimensioni varrà 0 mettendo il Cloud in condizione di attesa di ulteriori invii fino a ricevere quello in cui `isLast` sarà 1. Completata la stringa, verrà generato il `blob` contenente l'immagine che verrà salvata su Drive mentre prepariamo la risposta alla POST da restituire al modulo ESP32-CAM. Anche in questo caso la risposta sarà un oggetto JSON che questa volta conterrà più parametri tra cui `IntervalTime` per informare dopo quanti secondi poter inviare l'immagine successiva e che viene letto ogni volta dal foglio elettronico; questo ci permette tra l'altro di poterlo cambiare al volo durante l'esecuzione dell'applicazione. Altri parametri contenuti nell'oggetto JSON sono il comando `Flash` che attiva o disattiva il led sul pin 4 del modulo ed il comando di stop (`stopCommand`) che ci indicherà quando fermare l'acquisizione e viene impostato dallo script al verificarsi di uno dei seguenti eventi: la cella `Command` viene impostata manualmente con la stringa "Stop" sul foglio elettronico di Sheets, siamo usciti dalla fascia oraria di funzionamento impostata nel foglio dalla cella `EndAt`, è terminato l'evento corrispondente su Google Calendar, è stato raggiunto il numero massimo di acquisizioni impostato sul foglio tramite la cella `MaxFrames`. Essendovi diversi eventi per scatenare sia la condizione di partenza che di stop, sempre nel foglio elettronico vengono aggiunte semplici linee di log per indicare la causa della partenza, la ricezione delle singole immagini e la causa dell'eventuale stop. Un log più dettagliato è comunque creato su documento Google, prende il nome dalla cella `ProjectName` e viene creato al momento della partenza. Infine appena viene ricevuta la stringa con l'immagine, questa viene decodificata e l'immagine risultante viene inviata alla cartella Drive con lo stesso nome (`ProjectName`). Le immagini hanno un nome del tipo `file_<index>.jpg` in cui `index` è l'indice progressivo per ordinarle in base al tempo e poter creare facilmente un video finale. Al termine di tutte le acquisizioni verrà inviata una mail al proprio account con i link alla cartella Drive e al file di log. Gli stessi link vengono comunque generati sul foglio elettronico dallo stesso script in fase di start per monitorare in tempo reale il corretto funzionamento.

La scelta del tempo di intervallo tra uno scatto e l'altro non dovrebbe essere troppo breve (meno di 10 secondi) perchè, visti i vari log, accessi sul server e reindirizzamento 302, sarebbero difficilmente rispettati, ma d'altronde si tratta di un motion-lapse, quindi non dovrebbero esserci stringenti richieste di tempo; un intervallo di tempo molto ampio, invece, favorisce la regolarità dell'algoritmo, ma quasi tutti i comandi che daremo tramite foglio elettronico (ad esempio lo stop o l'attivazione del flash) avverranno con maggior latenza in quanto vengono inserite nella risposta della futura POST che però arriverà molto tempo dopo. Un altro problema che si presenta nel caso di immagini con risoluzione elevata è il tempo elevato di permanenza nella `doPost()` perchè le stringhe molto lunghe e la decodifica del `blob` con salvataggio su Drive rallentano le tempistiche; questo potrebbe facilmente bloccare il funzionamento nel modulo ESP32-CAM perchè la transazione della POST va in timeout; per questo motivo nella `doPost()` viene creata solamente la stringa, salvata nelle celle di log demandando la decodifica ed il salvataggio ad un trigger temporale eseguito nei periodi di attesa, oppure ad una nostra funzione dedicata (`decodeAllImages()`) eseguita al termine del processo direttamente dall'ambiente di sviluppo tramite pulsante `Run`.

CONCLUSIONI

Abbiamo proposto un'applicazione basata su modulo ESP32 perchè è molto comodo e può essere utile per realizzare video interessanti, ma lo scopo principale rimane quello di aver approfondito qualche aspetto su tecnologia offerta da Google che ci permette di sfruttare tanti strumenti assieme con altrettanta facilità senza bisogno di preoccuparci di spazio in memoria ed accessibile sempre ed in qualsiasi posto. □



Cosa occorre?

Il modulo ESP32-CAM (cod. ESP32-CAM) è in vendita a Euro 16,50, 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>**

Fai il pieno di energia!

€ 12,⁹⁰
Cod. UL45-6



6V-4,5AH

Batteria ricaricabile ULTRACELL al piombo senza manutenzione

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

12V-1,3AH



€ 13,⁹⁰
Cod. UL13-12



12V-7AH

€ 24,⁹⁰
Cod. UL7-12

6V-1,3AH



€ 7,⁹⁰
Cod. UL13-6



6V-3,4AH

€ 11,⁹⁰
Cod. UL34-6



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



€ 12,⁰⁰

Cod. HR03AAA800



€ 13,⁰⁰

Cod. HR6AA2100



€ 13,⁸⁰

Cod. HR14C3000

Prezzi IVA inclusa.

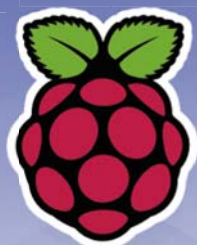
RASPBERRY PI 400!

La tastiera con il computer integrato

€ 146,00

Cod. PI400KIT-ITA

IVA inclusa



**Manuale
in italiano!**

Prezzo IVA inclusa.

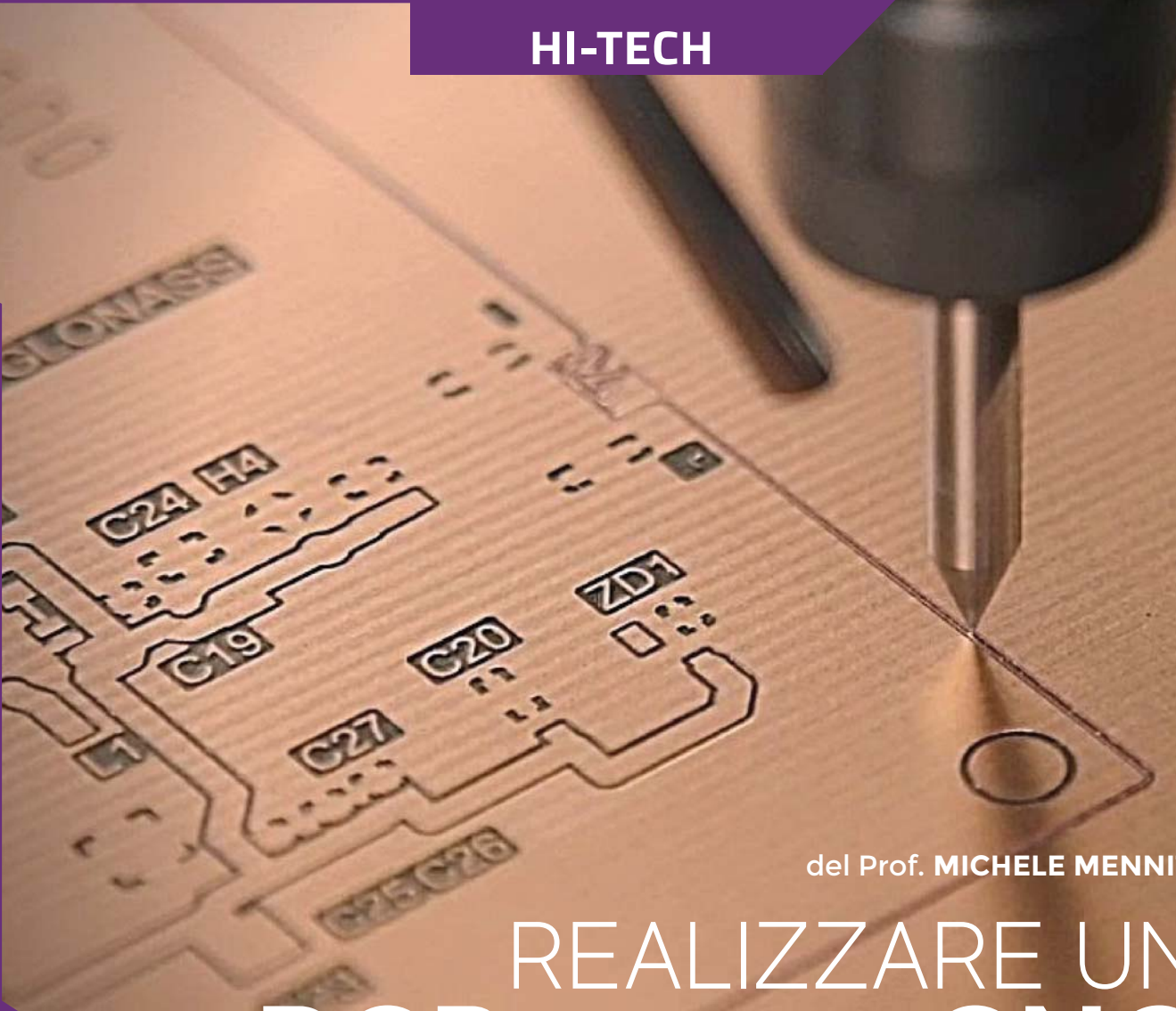
Kit completo con computer Raspberry Pi 400 accessori e manuale in Italiano.



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

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

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



del Prof. **MICHELE MENNITI**

REALIZZARE UN PCB CON LA CNC

Impariamo a realizzare un PCB artigianale con la CNC, partendo dal disegno del circuito stampato e utilizzando software open source o free. Prima puntata.

C

hiunque si occupi di elettronica pratica, per diletto o per lavoro, prima o poi si “scontra” con l’esigenza di realizzare un circuito stampato. La prototipazione si fa con le breadboard o le basette millefori, ma si tratta quasi sempre di circuiti che è difficile poter ritenere definitivi, anche se alcuni riescono a realizzare lavori

ben fatti con questi supporti. La strada maestra del prototipo, che può anche diventare definitivo, è quella del PCB (Printed Circuit Board), che in genere viene realizzato presso un service professionale, con costi che oggi sono più o meno alla portata di ogni tasca. Ma quella del service è una opzione che si può adottare per un circuito definitivo, che ha superato tutti i test e che magari deve andare in produzione, anche per piccole serie. Quando però si è in fase di progettazione, e la sperimentazione

richiede un circuito stabile con un piano di massa, e si suppone di dover apportare modifiche dopo i relativi test, allora la spesa richiesta per un PCB professionale può non essere congrua.

Fino a pochissimi anni fa, l'unico modo per realizzare un PCB artigianale era quello di fissare il disegno del circuito su una piastra di rame, mediante fotoincisione o trasferimento di toner, e poi immergere la piastra nel cloruro ferrico (o altre sostanze idonee) per eliminare il rame in eccesso; ma mentre un circuito in tecnologia SMT (montaggio superficiale) a quel punto era pronto, se parliamo di THT (montaggio a foro passante) il secondo passaggio da affrontare era quello della foratura, e qui iniziavano i problemi perché, come ben sappiamo, i fori non sono tutti delle stesse dimensioni e richiedono quindi punte di diametro diverso, per non parlare della necessità di essere precisi nel centrare ogni singolo foro; è chiaro che questo passaggio rappresentava una notevole perdita di tempo, con risultati raramente perfetti. Queste tecniche si usano ancora oggi, perché i passaggi fino all'incisione nel cloruro ferrico sono molto rapidi e danno buoni risultati, ma trattare sostanze chimiche in piccoli ambienti, come una stanza di casa o un piccolo laboratorio, può essere molto pericoloso.

L'avvento delle CNC, che ormai sono diffusissime a motivo di costi sempre più accessibili, ha sicuramente rivoluzionato il mondo della prototipazione, permettendo di realizzare circuiti tecnicamente perfetti, e senza dover maneggiare sostanze chimiche e, soprattutto, di ottenere forature di diametro corretto e perfettamente centrate.

Sul numero 262 di Elettronica In abbiamo presentato un Controller universale per macchine CNC a tre assi, ed in quell'occasione abbiamo utilizzato la meccanica CNC2018 (reperibile con il codice CNC2018 sul sito www.futurashop.it).

In queste nuove puntate useremo questa eccellente combinazione, assieme ad una serie di software open source o free, per imparare a realizzare un PCB di media difficoltà con componenti PTH; vedremo quindi tutti i passaggi necessari per ottenere il file che piloterà la CNC sia per la fase di incisione che per quella della foratura. Inoltre suggeriremo una semplice aggiunta meccanica, che permetterà di fissare al piatto la piastra da lavorare con grande facilità e precisione.

I programmi che abbiamo usato sono: Eagle 9.6.2 free, per generare i file gerber, FlatCAM v.8.5 open source (scaricabile dalla pagina <https://bitbucket.org/jpcgt/flatcam/downloads/>) che servirà per generare i file CAM in linguaggio gcode, da utilizzare con Candle v.1.1.7, sempre open source (scaricabile tramite il link: <https://github.com/Denvi/Candle>) che si occuperà di pilotare la CNC fornendo i necessari comandi al nostro Controller. Per facilitarvi le operazioni abbiamo caricato, nella sezione download del sito www.elettronica.in, sia FlatCAM che Candle, anche se consigliamo vivamente di visitare i link indicati per verificare se nel frattempo sono stati pubblicati aggiornamenti.

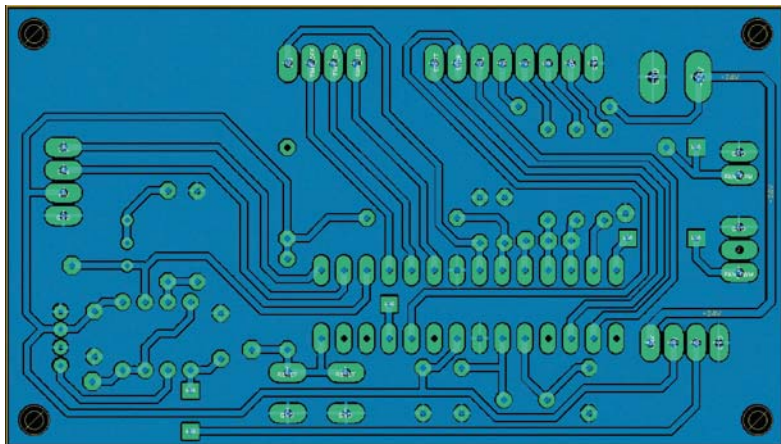
Il programma Eagle di Autodesk non è open source, quindi bisogna scaricarlo dal link <https://www.autodesk.com/eagle-download-win> e poi registrarsi sul sito per usarlo in versione free con funzioni limitate. Comunque noi daremo per scontato che lo schema elettrico ed il disegno del PCB siano già stati realizzati, quindi vedremo direttamente come generare i file gerber; ciò significa che se disponete di un altro software CAD, sarà sufficiente che siate in grado di generare i file gerber e potrete saltare direttamente al secondo passaggio, quello del CAM.

CONTROLLO DRU

Preliminarmente occorre aver disegnato lo schema elettrico del circuito ed il relativo PCB, nel caso non siate attrezzati, noi siamo in condizioni di offrire questo servizio, sarà sufficiente fornirci uno schema elettrico, anche disegnato a mano, sotto forma di file pdf, e vi restituiranno i file del PCB.

Eccetto rarissimi casi, è sempre conveniente dotare il PCB del cosiddetto "piano di massa" che, oltre ad unire tutti i punti di massa del circuito, senza necessità di disegnare le relative piste, riempirà tutti gli spazi vuoti del PCB; questa cosa sarà utilissima ai fini dell'incisione, altrimenti ci complicheremmo la vita dovendo eliminare tutto il rame in eccesso, prolungando i tempi di incisione a dismisura, oppure dovendo ricorrere all'impostazione di una serie di poligoni isolati, durante la fase di elaborazione CAM.

Fig. 1
Il disegno del PCB
usato per le prove.



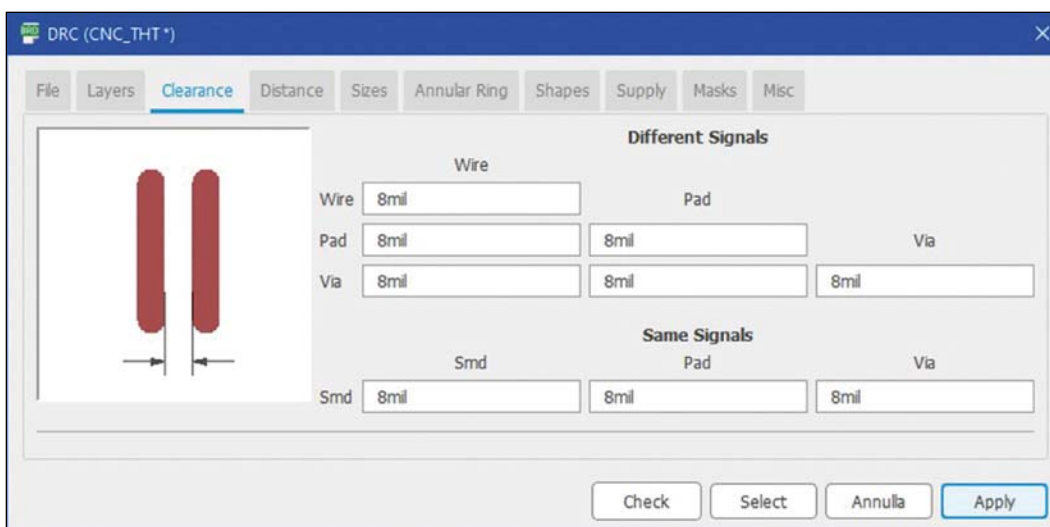


Fig. 2
Impostazione
delle
distanze.

Come esempio useremo il disegno visibile in **Fig. 1**. La prima operazione da fare sul programma CAD è quella di creare un file DRU specifico (si chiama così nel caso di Eagle) contenente le regole di progettazione, fondamentalmente le dimensioni delle piste e dei pad e le relative spaziatore, in base al tipo di PCB che si intende realizzare. Bisogna infatti considerare che le tipiche impostazioni di default dei programmi CAD vanno bene per la realizzazione del PCB presso un service professionale, ma non per una procedura artigianale.

I parametri sui quali bisogna agire fondamentalmente sono le dimensioni minime delle piste di collegamento, pad (piazzole di saldatura dei reofori dei componenti) e vias (punti di comunicazione tra faccia superiore ed inferiore del PCB) e le distanze tra queste componenti e tra di esse ed il piano di massa, ossia tutto il rame adiacente che copre gli spazi rimanenti del PCB e che è collegato a massa. Se non si adottano valori idonei potrebbe risultare difficile saldare i componenti sul PCB.

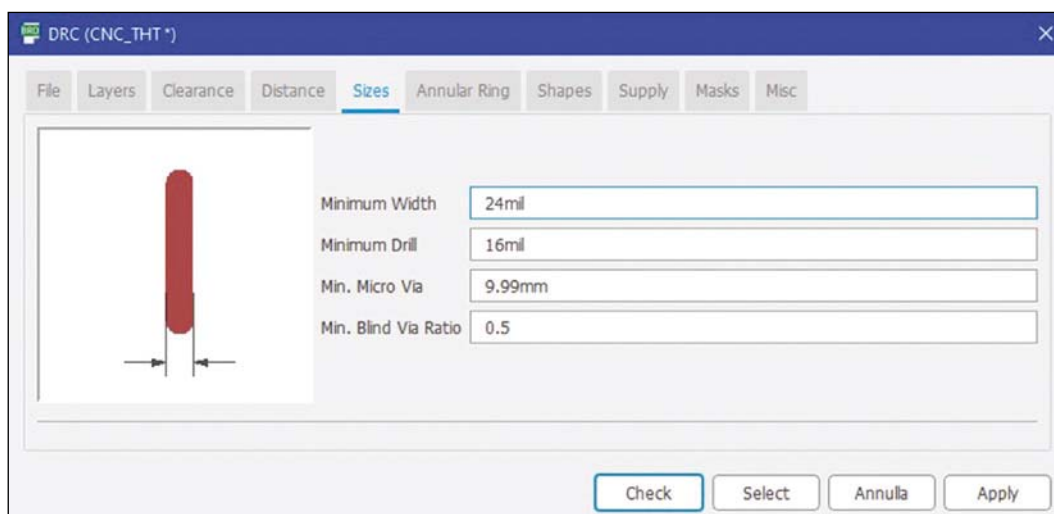
Vediamo questi parametri e che valori conviene assegnare loro:

- **Clearance:** è la spaziatura minima tra le varie componenti del disegno: piste, pad e vias. Nel nostro caso abbiamo usato un valore di 8 mils, pari a circa 0,2 mm, che è un valore standard usato anche per i service; questo significa che bisogna avere una discreta manualità col saldatore, altrimenti lo stagno può invadere piste o pad adiacenti o il piano di massa, provocando cortocircuiti. Il valore di 0,2 mm coincide con le dimensioni della punta del bulino che abbiamo usato con la nostra CNC, quindi una sola passata sarà sufficiente a rimuovere il rame in eccesso

dalla piastra ramata, durante l'incisione (**Fig. 2**). Se pensate di non avere sufficiente abilità nella saldatura, allora vi consigliamo di raddoppiare questo valore, portandolo a 16 mils; in questo caso potrete usare un bulino da 0,4 mm oppure sempre quello da 0,2 mm, col quale la CNC farà due passate adiacenti. In ogni caso, in fase di saldatura, un buon aiuto può venire dall'uso del flussante; passandolo con un pennello, prima della saldatura, si ottiene una facilitazione della saldatura e si evita che lo stagno tocchi piste o pad vicini.

- **Sizes:** sono le dimensioni minime delle piste e il diametro minimo dei fori; usando un valore troppo piccolo di larghezza delle piste, si corre il rischio che una sbavatura del taglio del bulino possa generare piste estremamente sottili o anche interrotte. Noi abbiamo usato 24 mils, pari a circa 0,6 mm, mentre il diametro dei fori è ininfluente per le nostre necessità, comunque va bene un valore di 16 mils (**Fig. 3**).
- **Annular ring:** sono le dimensioni di pad e vias; i valori standard delle librerie dei vari componenti vanno bene per un service, ma quando si opera con la CNC, facilmente il bordo esterno perde qualcosa, mentre la foratura senza metallizzazione toglie altro materiale nella parte centrale; questo può comportare una eccessiva riduzione dello spazio di saldatura attorno ai reofori, con le difficoltà che ne conseguono. Le dimensioni sono definite in percentuale rispetto al diametro interno del foro, all'interno di un valore minimo ed uno massimo. Per i pad questi valori sono settabili separatamente per le superfici superiore (Top), inferiore (Bottom) ed interna (Inner) della piastra

Fig. 3
Impostazione della
larghezza delle
piste.



ramata, ma quest'ultimo valore è ininfluente, in quanto può essere usato solo per PCB professionali multistrato, che non sono realizzabili con tecniche artigianali. Per quanto riguarda i vias le dimensioni sono uniche per le due facce (Outer), mentre per il valore interno (Inner) vale quanto detto per i pad. Anche i microvias rientrano nelle tipologie professionali. Nel nostro caso abbiamo impostato un aumento delle dimensioni del 50% rispetto a quello previsto dalle librerie dei componenti, con un range di valori tra 10 e 40 mils (**Fig. 4**).

Nella sezione download del sito www.elettronica.in.it troverete il file CNC_THT.dru che andrà copiato nella cartella "dru" di Eagle. Una volta aperto il file del disegno del PCB, bisogna usare il comando menu "Tools – DRC...", cliccare sul pulsante "Load..."

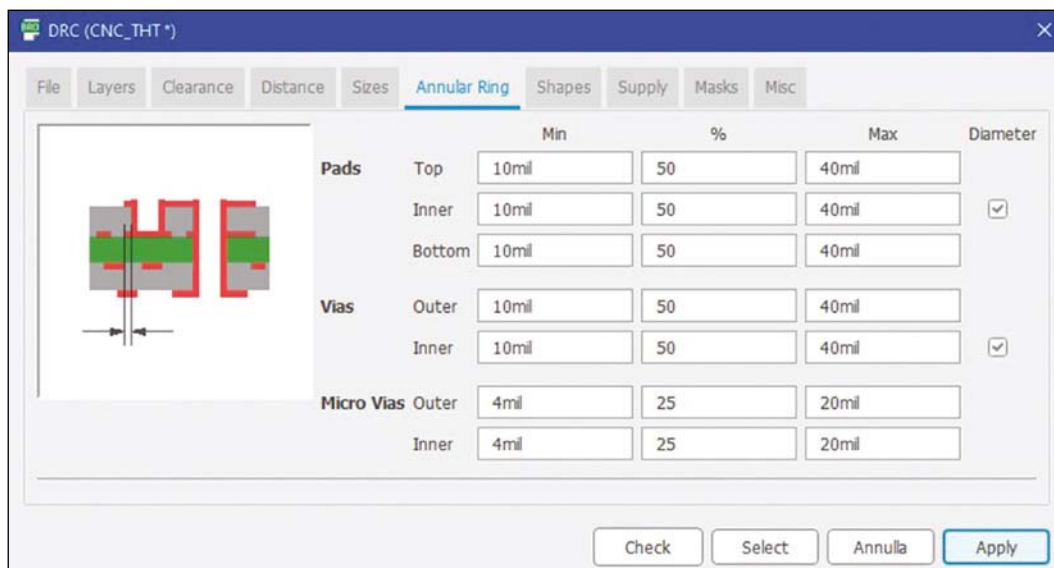
e caricare il file. Cliccando prima su "Apply" e poi su "Check", Eagle applicherà le impostazioni e poi verificherà che il disegno abbia caratteristiche corrispondenti a quelle previste nel DRU, in caso negativo si aprirà una maschera con tutti gli errori riscontrati. Nella **Fig. 5** abbiamo volutamente applicato un DRU per la CNC ad un disegno di un PCB realizzato per la stampa presso un service.

Il risultato è quello di ben 387 errori e 30 avvisi.

Nello specifico abbiamo:

- 31 Clearance: significa che in 31 casi non sono state rispettate le distanze minime tra le piste;
- 1 Drill Size: significa che in un caso le dimensioni di un foro sono inferiori a quelle minime previste;
- 5 Overlap: significa che ci sono 5 problemi di sovrapposizione di piste, dovuti al fatto che le impostazioni del DRU utilizzate aumentano le

Fig. 4
Impostazione delle
dimensioni di pad
e vias.



dimensioni dei PAD e li sovrappongono tra loro; in realtà questo errore uscirebbe anche con il DRU corretto, infatti li abbiamo provocati volontariamente;

- 350 Width: significa che in 350 punti le piste non rispettano le dimensioni minime previste dal DRU;
- 30 Wire Stub: sono ripassi ripetuti in alcuni punti delle piste, succede facilmente quando si cancella una pista per ricrearla, ma non ci si accorge di non averla cancellata completamente, specialmente all'interno dei PAD.

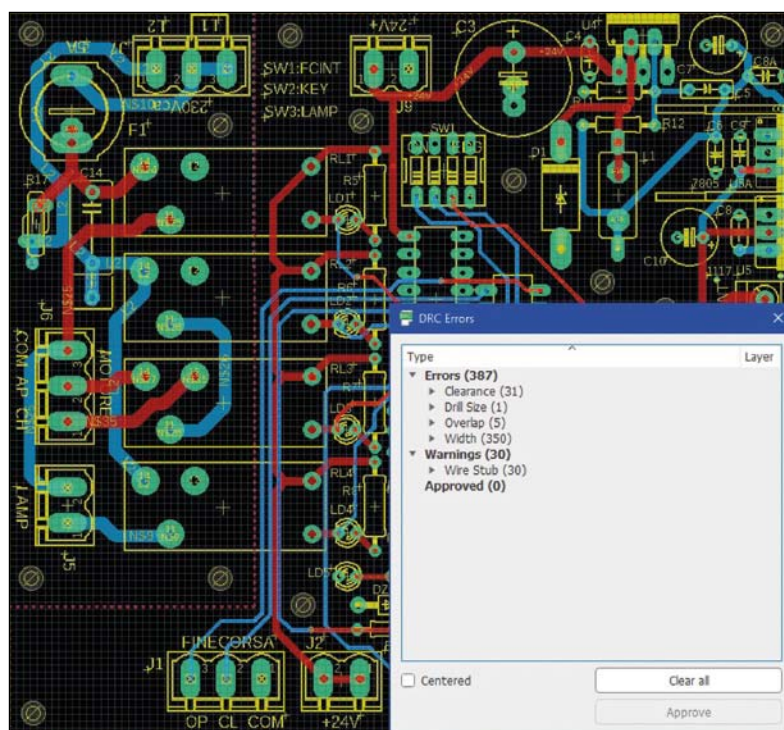
Eagle ci aiuta a trovare e rimuovere gli errori, in quanto basta selezionarli singolarmente e si aprirà un riquadro evidenziatore, che indicherà il punto o tratto interessato dall'errore selezionato. La **Fig. 6** mostra appunto l'indicazione di un errore di clearance, la pista che passa tra i due PAD è troppo vicina e non rispetta i parametri del DRU; quindi va ridotta o spostata fino ad eliminare l'errore. In alcuni casi potreste decidere che l'errore indicato dal controllo DRU sia ininfluente ai fini della realizzazione del vostro prototipo, in tal caso potreste selezionare quell'errore e cliccare sul pulsante "Approve", in modo da eliminarlo dall'elenco, ma questa opzione va usata solo se sapete esattamente cosa state facendo, altrimenti potreste avere poi problemi nell'incisione e, probabilmente, nel funzionamento del circuito dopo il montaggio dei componenti.

GENERAZIONE FILE GERBER

Una volta rimossi (o approvati) tutti gli eventuali errori indicati dal DRU, si può passare alla fase della creazione dei file gerber, operazione piuttosto semplice con Eagle.

Di seguito l'elenco dei passaggi:

- cliccare su CAM Processor (**Fig. 7**);
- selezionare TOP COPPER per controllare l'antepagina della faccia superiore del PCB (utile per circuiti SMT o THT a doppia faccia);
- selezionare BOTTOM COPPER per controllare l'antepagina della faccia inferiore del PCB (utile per monofaccia THT);
- attivare "Board shape" per ottenere un bordo attorno al PCB (sarà utile in fase di taglio finale del PCB);
- attivare "Cutouts" per ottenere le cerchiature interne dei fori grandi; NON toccare la sezione "Advanced", che serve fondamentalmente per il mirror dei PCB THT, la tratteremo usando il FlatCAM;
- cliccare su Process Job, istantaneamente



Eagle crea, nel percorso selezionato, la cartella "CAMOutputs" con tutti i file necessari per la lavorazione del PCB da parte di un service. Potete lasciare la cartella dove è stata creata o spostarla in un percorso più comodo, per i prossimi passaggi.

Fig. 5
Errori segnalati dal controllo DRU.

GENERAZIONE FILE CAM DELLE PISTE

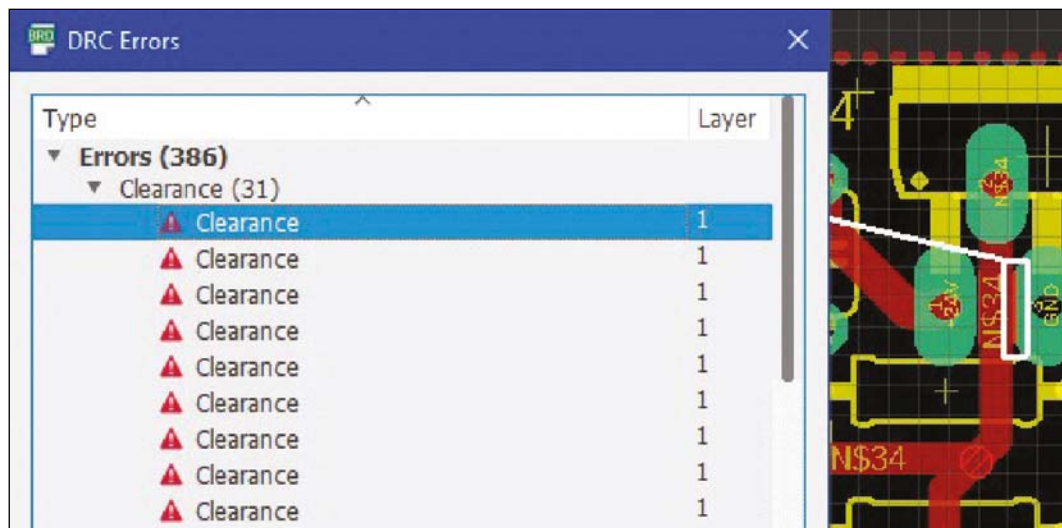
Dopo aver installato FlatCAM occorre eseguirlo e, la sola prima volta che lo si usa, bisogna effettuare questi semplici passaggi:

- posizionarsi nella riga vuota che si trova alla base della shell del programma (**Fig. 8**);
- se la shell non è visibile, eseguire il comando "Tool – Command line";
- scrivere: **get_sys excellon_zeros** e premere INVIO sulla tastiera;
- scrivere: **set_sys excellon_zeros T** e premere INVIO sulla tastiera;

Queste due righe di codice servono per poter vedere i fori in modo chiaro, nel momento in cui caricherete un file di tipo "drill", ne parleremo più avanti. Questa operazione modifica definitivamente le opzioni del programma, quindi non andrà più eseguita in futuro. A questo punto la shell si può anche chiudere per lasciare tutto lo spazio possibile al riquadro principale, è sufficiente eseguire di nuovo il comando "Tool – Command line".

Eseguire il comando "File – Open Gerber ..." e

Fig. 6
Localizzazione di
un errore DRU.



cercare il file da aprire nella cartella "CAMOutputs/
GerberFiles", poiché stiamo lavorando con un circuit-
to THT monofaccia, ci interessa solo quella inferiore
(file "copper_bottom.gbr"), nel caso dei circuiti SMT

ci servirebbe il file "copper_top.gbr".

La **Fig. 9** mostra il tipo di immagine che si vedrà nel
riquadro centrale, dopo il caricamento del file.

Aprire la scheda "**Options**" e impostare il campo

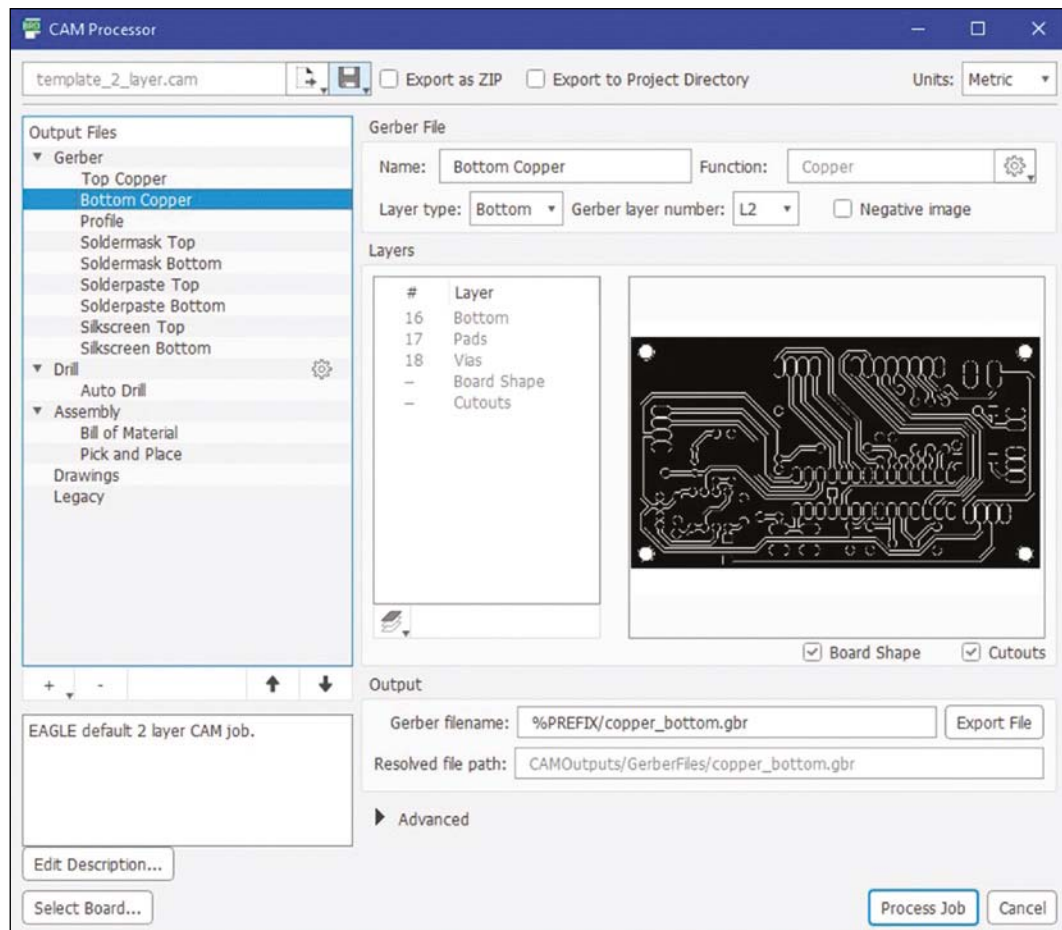


Fig. 7
Il CAM Processor
di Eagle.

"Units" su mm.

Poiché il disegno del PCB è sempre realizzato dal lato dei componenti, così com'è l'incisione risulterebbe rovesciata, quindi bisogna effettuare l'inversione a specchio (mirror). Occorre eseguire il comando "Tool – Double-Sided PCB Tool" e impostare questi parametri (Fig. 10):

- Bottom layer = copper_bottom.gbr;
- Mirror axis = Y;
- Axis location = Box;
- Point/Box = copper_bottom.gbr.

Cliccare su "Mirror Object", il PCB sarà ora visibile direttamente dal lato saldature (faccia inferiore).

Nota bene: Naturalmente la scheda Tool non andrà modificata nel caso in cui dovessimo lavorare con la faccia superiore (circuiti SMT).

Aprire la scheda "Project" e fare doppio clic sul progetto "copper_bottom.gbr", si aprirà la scheda "Selected", bisogna impostare tutte le seguenti opzioni:

Plot options:

- Attivare le caselle "Plot" e "Solid".

Isolation routing:

- Tool dia: qui bisogna inserire il diametro della punta del bulino che useremo per l'incisione. Il DRU che abbiamo usato noi prevede una distanza tra le piste, o tra le piste ed il piano di massa, di 8 mils (circa 0,2 mm) quindi occorre scegliere un bulino con diametro uguale o sottomultiplo di tale valore; noi disponiamo di bulini da 0,1 mm e 0,2 mm. Se adottiamo il bulino da 0,2 mm la CNC farà un solo passaggio incidendo esattamente

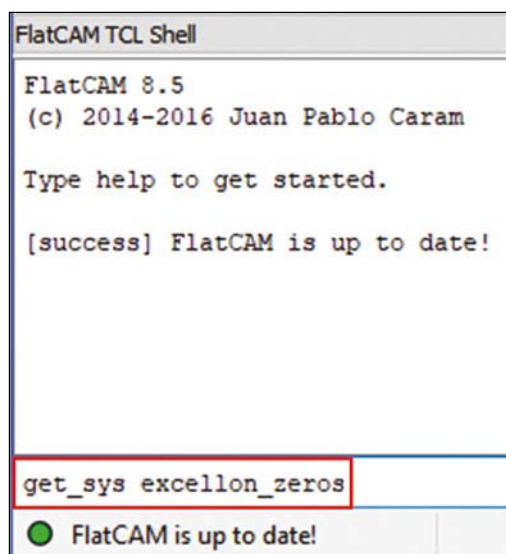


Fig. 8
Il riquadro della linea di comando.

lo spazio previsto dal DRU; se invece usiamo il bulino da 0,1 mm la CNC farà automaticamente due passaggi affiancati per incidere lo stesso spessore; impostiamo "0.2".

- Width: serve per aumentare l'isolamento delle piste; ha significato usare questo campo quando non si può intervenire a monte sul DRU applicato al PCB, ma nel nostro caso si imposta a 1 (singola passata). Se si imposta un numero di passate maggiore il FlatCAM crea un file grbl per ogni passata, quindi occorre incidere il PCB più volte; ad ogni nuova passata il bulino viene spostato di una frazione pari al valore dato al successivo campo Pass overlap. Per esempio se

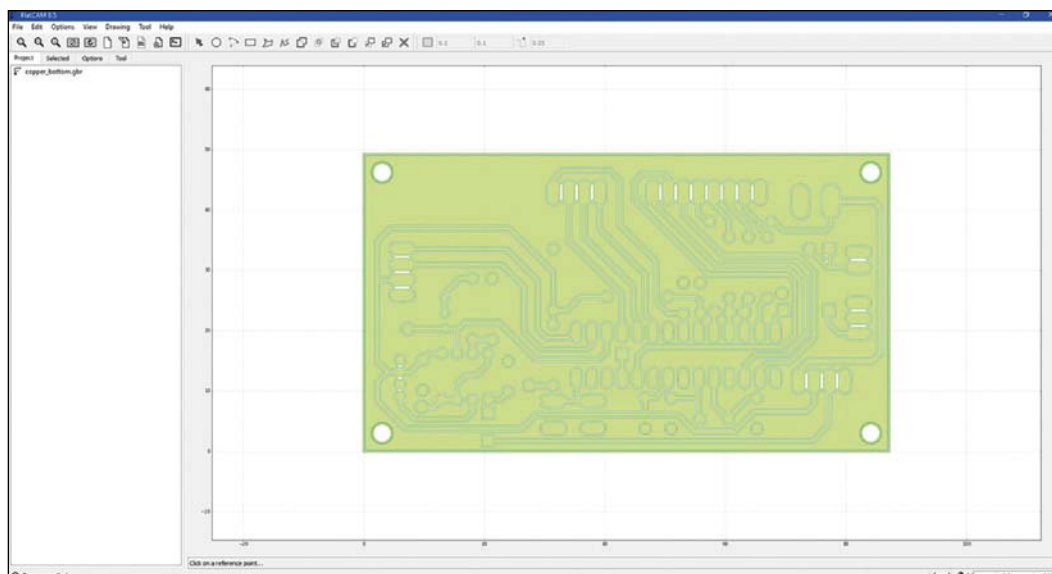


Fig. 9
Immagine del file gerber.

Fig. 10
Il mirror del PCB.

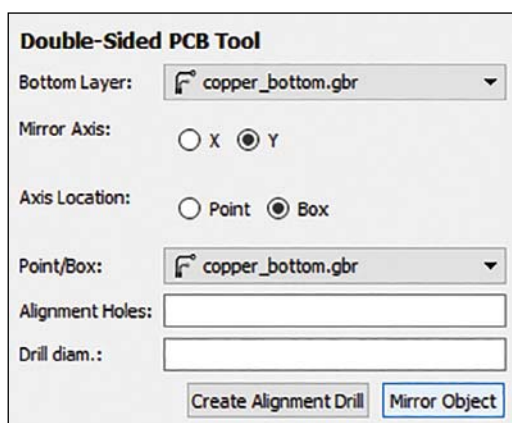


Fig. 11
Le linee di incisione.

l'isolamento previsto dal DRU è di 0,2 mm, ma noi vogliamo aumentarlo a 0,3 mm, disponendo del solo bulino da 0,2 mm, impostiamo questo campo a 2 e il campo successivo (Pass overlap) a 0,5 (50%), in modo che la seconda passata sia sovrapposta per il 50% alla prima, di conseguenza la seconda incisione sarà di 0,1 mm.

- Pass overlap: questo campo viene usato solo se Width è maggiore di 1, in caso contrario viene ignorato.
- Combine passes: può essere utile attivare questa opzione, quando Width > 1, in quando il FlatCAM

genera un unico file invece di un numero di file pari al valore di Width.

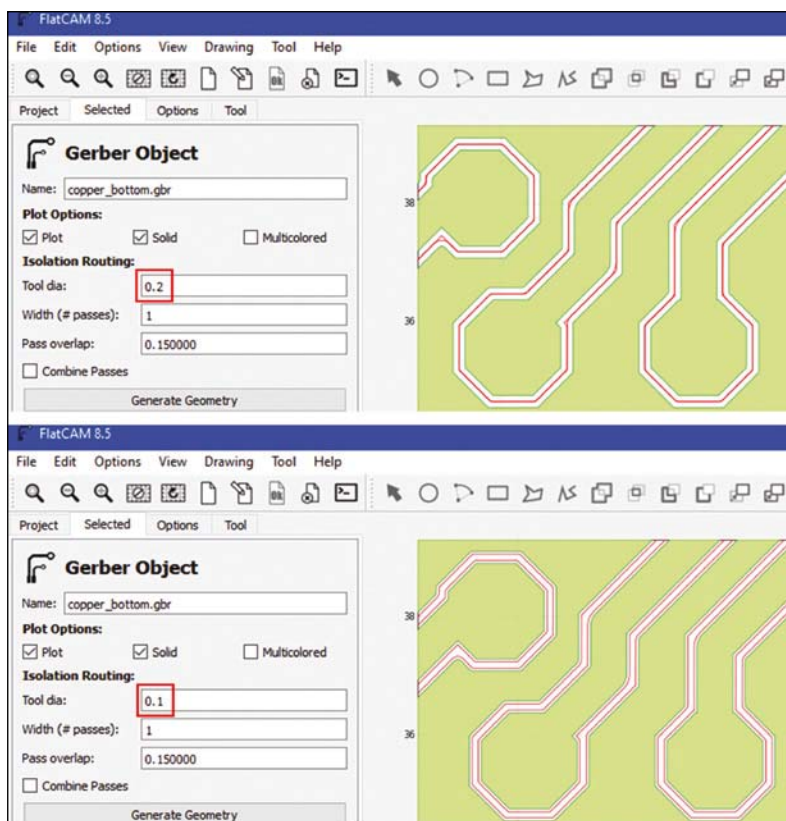
- Cliccare su "Generate Geometry", apparirà una linea rossa all'interno delle tracce di isolamento, ad indicare che una sola passata sarà sufficiente per eliminare il rame in eccesso; nella **Fig. 11** abbiamo ingrandito l'immagine, e simulato sia l'uso del bulino da 0,2 mm (sopra) che quello da 0,1 mm (sotto) in modo da rendere evidente il fatto che nel secondo caso la CNC dovrà fare due passate per garantire l'isolamento previsto dal DRU.

Nel caso vi accorgiate di qualche errore dopo aver eseguito l'operazione di generazione, occorre eseguire questi passaggi:

- Aprire la scheda "Project";
- Selezionare il progetto "copper_bottom.gbr_iso" appena creato dal programma;
- Premere "Canc" sulla tastiera per eliminarlo;
- Fare doppio clic sul progetto "copper_bottom.gbr";
- Ripetere la sequenza di operazioni precedenti.

Se invece è tutto a posto aprite la scheda "Project" e fate doppio clic sul progetto "copper_bottom.gbr_iso", si aprirà nuovamente la scheda "Selected", e bisognerà impostare i seguenti valori:

- Plot: attivare la casella;
- Cut z: profondità dell'incisione in millimetri; di norma lo spessore del rame è di 0,035 mm, quindi occorre inserire un valore maggiore, considerando anche tolleranze di movimento dello spindle, in genere va bene da -0.06 a -0.08;
- Travel z: è il sollevamento del bulino durante gli spostamenti, quando non incide, si possono impostare 1,5 mm;
- Feed rate: velocità di incisione in mm/min, un valore basso impiega molto tempo ma garantisce grande affidabilità, un valore alto genera una incisione veloce ma con qualche rischio per il bulino o di perdita di passi; noi abbiamo trovato un buon compromesso con un valore compreso tra 50 e 80;
- Tool dia: 0.2 (uguale al valore inserito in precedenza);
- Spindle speed: velocità massima di rotazione del mandrino; questo valore serve per la gestione del segnale PWM generato dal controller della CNC, il nostro modello prevede questa possibilità, e dal datasheet dello spindle ricaviamo che a 36 V può raggiungere 9000 RPM, quindi inseriamo tale valore. Se il controller da Voi usato invece ha la semplice commutazione a relè, questo campo



può essere ignorato e restare vuoto. (Fig. 12).

Cliccare su "Generate"; le linee rosse saranno sostituite da linee blu che copriranno esattamente tutto lo spazio di isolamento, in pratica sarà possibile controllare quanto rame sarà eliminato con l'incisione. Inoltre si vedrà una complessa serie di linee semi-trasparenti che rappresenteranno il percorso che farà il bulino durante la CNC. Questa immagine si può ingrandire ed osservare attentamente, per verificare che il risultato sia quello atteso, in caso contrario si potranno prendere i necessari provvedimenti, visto che qui siamo ancora a livello di generazione dei file e possiamo sempre iniziare tutto daccapo (Fig. 13).

Nel caso vi accorgiate di qualche errore dopo aver eseguito l'operazione di generazione, occorre eseguire questi passaggi:

- Aprire la scheda "Project";
- Selezionare il progetto "copper_bottom.gbr_iso_cnc" appena creato dal programma;
- Premere "Canc" sulla tastiera per eliminarlo;
- Fare doppio clic sul progetto "copper_bottom.gbr_iso";
- Ripetere la sequenza di operazioni precedenti.

Se invece è tutto a posto aprite la scheda "Project" e fate doppio clic sul progetto "copper_bottom.gbr_iso_cnc", si aprirà nuovamente la scheda "Selected", bisogna attivare l'opzione "Plot" e inserire nel campo "Tool dia" lo stesso valore usato nelle maschere precedenti (nel nostro caso 0.2), quindi cliccare su "Update Plot", attendere alcuni secondi, anche se apparentemente non succede nulla, poi cliccare su

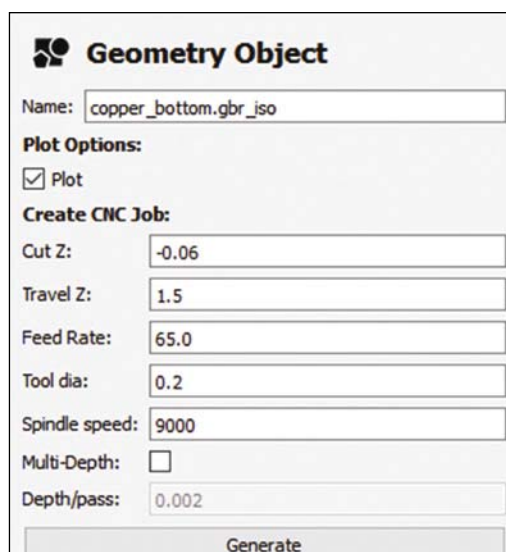


Fig. 12
Le impostazioni
per lo spindle

Export G-Code e salvare il file come "Nome_PCB.nc", in cui Nome sarà il nome del vostro progetto; non è obbligatorio dare un nome specifico, ma se lo fate potrete creare una cartella contenente tutti i vostri file, per usi futuri; l'unica cosa indispensabile è la desinenza ".nc", altrimenti il file non sarà visto dal software Candle.

Arrivati a questo punto, se il vostro progetto prevede componenti SMD, e quindi non ci sono fori, potete chiudere FlatCAM e passare direttamente alla fase dell'incisione, altrimenti bisogna proseguire con il passaggio successivo di questo programma.

GENERAZIONE FILE CAM DEI FORI

Eseguire il comando "File – Open Excellon..." e aprire il file "drill_1_16.xln", presente nella cartel-

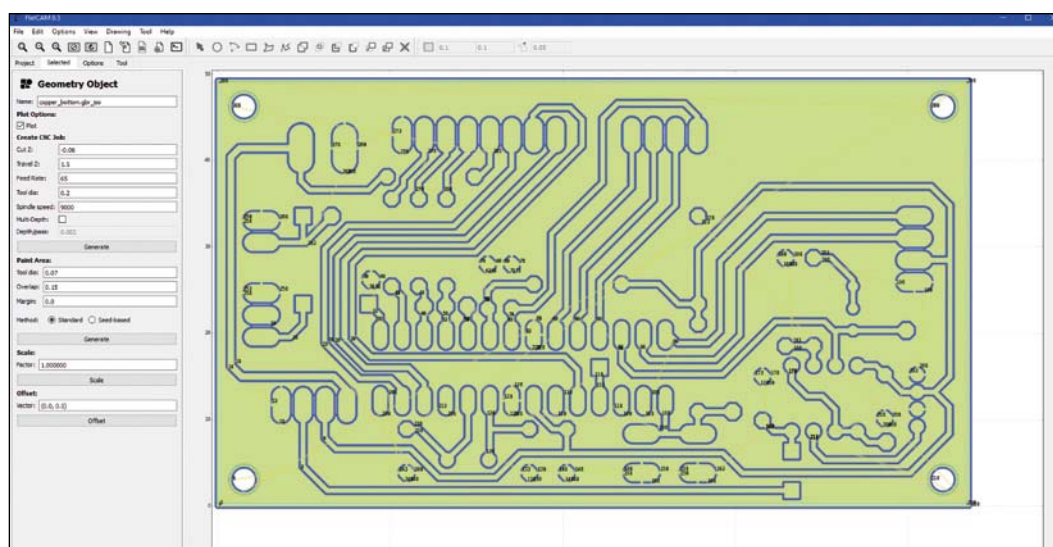
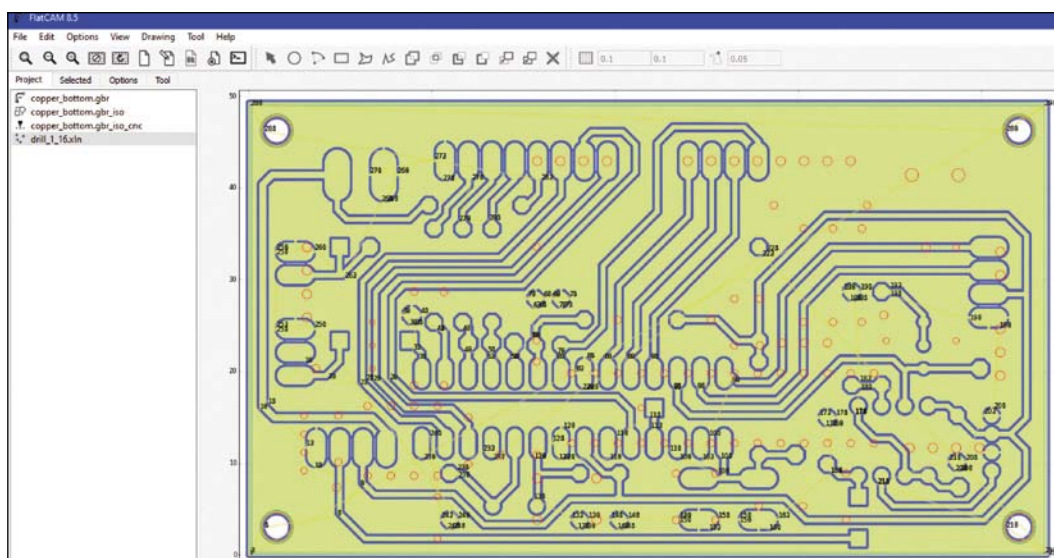


Fig. 13
Anteprima
dello spazio
di isolamento.

Fig. 14
Esempio di fori
disallineati.



la "CAMOutputs/DrillFiles. Se in precedenza avete eseguito il mirror del PCB, vedrete che i fori (rappresentati da cerchietti rossi) non combaceranno con i sottostanti pad o vias (**Fig. 14**);

Bisogna eseguire il comando "Tool – Double-Sided PCB Tool" e impostare questi parametri:

- Bottom layer = drill_1_16.xln;
- Mirror axis = Y;
- Axis location = Box;
- Point/Box = copper_bottom.gbr.

Cliccare su "Mirror Object", istantaneamente i fori risulteranno perfettamente centrati rispetto ai sottostanti pad e vias (**Fig. 15**).

Nel caso vi accorgiate di qualche errore dopo aver

eseguito l'operazione di mirror, occorre eseguire questi passaggi:

- Aprire la scheda "Project";
- Eliminare il progetto "drill_1_16.xln" premendo "Canc" sulla tastiera;
- Eseguire il comando "File – Open Excellon..." e aprire il file "drill_1_16.xln"
- Ripetere la sequenza di operazioni precedenti.

Se invece è tutto a posto aprite la scheda "Project" e fare doppio clic sul progetto "drill_1_16.xln", si aprirà nuovamente la scheda "Selected", e bisognerà impostare i seguenti valori:

Plot options

- Plot: attivare la casella;

Fig. 15
Allineamento dei
fori.

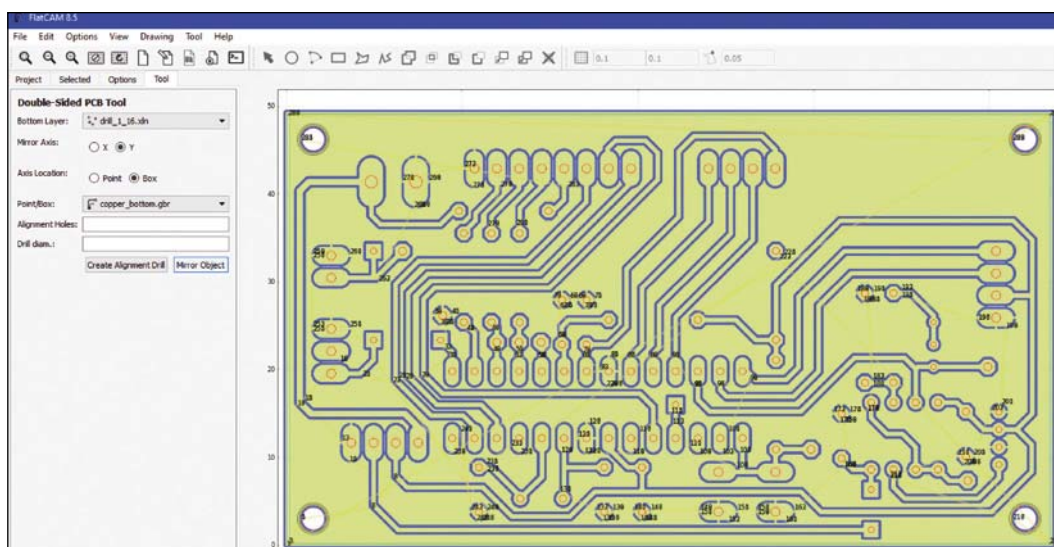
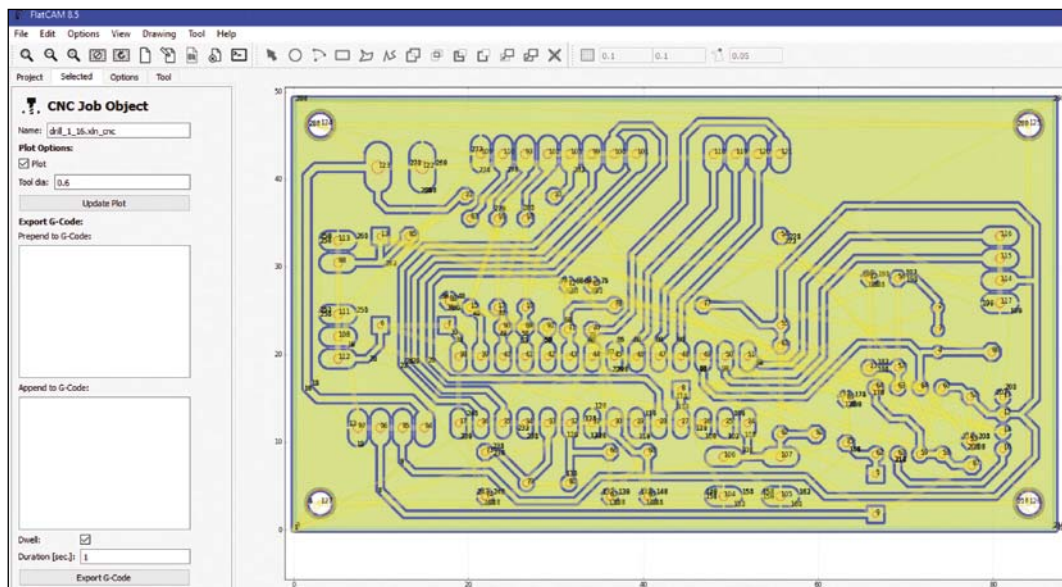


Fig. 18
Anteprima del
percorso della
sequenza di
foratura.



per la nostra CNC impostiamo 9000, se invece disponete di un altro modello di mandrino, dovete far riferimento al suo datasheet.

Nella **Fig. 16** abbiamo riportato la sezione appena descritta con i valori impostati. Cliccare su "Generate", tutti i fori saranno numerati in base alla sequenza di foratura (**Fig. 17**); se ciò non dovesse verificarsi, totalmente o solo per alcuni fori, significa che non sono stati selezionati correttamente gli ID dei tools. Nel caso vi accorgiate di qualche errore dopo aver eseguito quest'ultima operazione, occorre eseguire questi passaggi:

- Aprire la scheda "Project";
- Selezionare il progetto "drill_1_16.xln_cnc" appena creato dal programma;
- Premere "Canc" sulla tastiera per eliminarlo;
- Ripetere la sequenza di operazioni precedenti.

Se invece è tutto a posto aprite la scheda "Project" e fare doppio clic sul progetto "drill_1_16.xln_cnc", si aprirà nuovamente la scheda "Selected", e bisognerà impostare il seguente valore:

- Tool dia: 0.6 (diametro punta iniziale di default);
- ed eseguire i seguenti passaggi:
- Cliccare su Update Plot, l'immagine mostrerà ora anche il percorso delle forature (**Fig. 18**);
- Cliccare su Export G-Code;
- Salvare con Nome_FORI.nc, in cui Nome sarà il nome del vostro progetto, come spiegato in precedenza per il file delle piste; anche in questo caso è indispensabile usare la desinenza ".nc", altrimenti il file non sarà visto dal software Candle.

Con quest'ultimo passaggio abbiamo completato le operazioni CAM ed ora disponiamo di due file in formato grbl, con i quali andremo a lavorare la nostra piastra di rame, togliendo prima il rame in eccesso e praticando poi i necessari fori per i componenti.

Nota bene: è importante salvare il progetto FlatCAM, magari nella stessa cartella dei file grbl, in modo da poterlo riaprire per effettuare rapide modifiche a qualche parametro, senza dover rifare tutti i passaggi visti finora. Il comando da usare è "File – Save Project As ...", fornendo percorso e nome (in genere lo stesso usato per i due file grbl) tipo "Nome_prj" senza desinenza, in cui Nome sarà sempre il nome usato in precedenza. Nella prossima puntata passeremo alla parte pratica, con l'approfondimento dello studio del programma Candle e la realizzazione del nostro primo PCB. □

Cosa occorre?

Il materiale elettronico utilizzato in questo progetto è disponibile presso Futura Elettronica.

La meccanica della CNC (cod. CNC2018) è disponibile a Euro 179,00, l'alimentatore switching 350W/36V (cod. MW35036) a Euro 39,00, lo spindle (cod. ELEMANDER11) è in vendita a Euro 85,00, i motori utilizzati (cod. STEPMOT03) costano ciascuno Euro 18,00, la scheda controller (cod. GRBLCONTROLLER) è disponibile a Euro 89,00 mentre la scheda offline con display (cod. CNCCTRL) costa Euro 44,00.

Gli accessori per le macchine CNC sono disponibili sul sito bit.ly/CNCFE. 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>**

I tuoi PCB a partire da

1€

Con il servizio PCB di Futura Elettronica, potrai realizzare facilmente il circuito stampato del tuo progetto!

Vai su www.futurashop.it, inserisci nella ricerca il codice "PCBPRODUCTION" e carica il tuo Gerber scegliendo le dimensioni e le caratteristiche preferite.

Prezzi IVA inclusa.



cod. PCBPRODUCTION

COLORE SERIGRAFIA



BIANCA



NERA

La serigrafia nera è disponibile solamente per PCB bianchi e gialli.

COLORE SOLDER



VERDE

ROSSO

GIALLO

BLU

BIANCO

NERO

SPESSORE (mm)

mm



0.2

0.4

0.6

0.8

1.0

1.2

1.6

2.0

2.4

2.8

3.2

NUMERO DI LAYERS

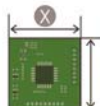
1 Layer

2 Layers

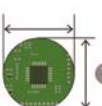
4 Layers

6 Layers

DIMENSIONI (cm)



Single Size



Pannel Size

- 5x5 - 5x10 - 5x15 - 5x20 - 5x25 - 5x30
- 10x10 - 10x15 - 10x20 - 10x25 - 10x30
- 15x15 - 15x20 - 15x25 - 15x30
- 20x20 - 20x25 - 20x30
- 25x25 - 25x30
- 30x30

ECCO QUALCHE ESEMPIO:

PCB con 1 layer
da 5x5 cm
spessore 1,6 mm
solder verde
serigrafia bianca
€ 1 cad.

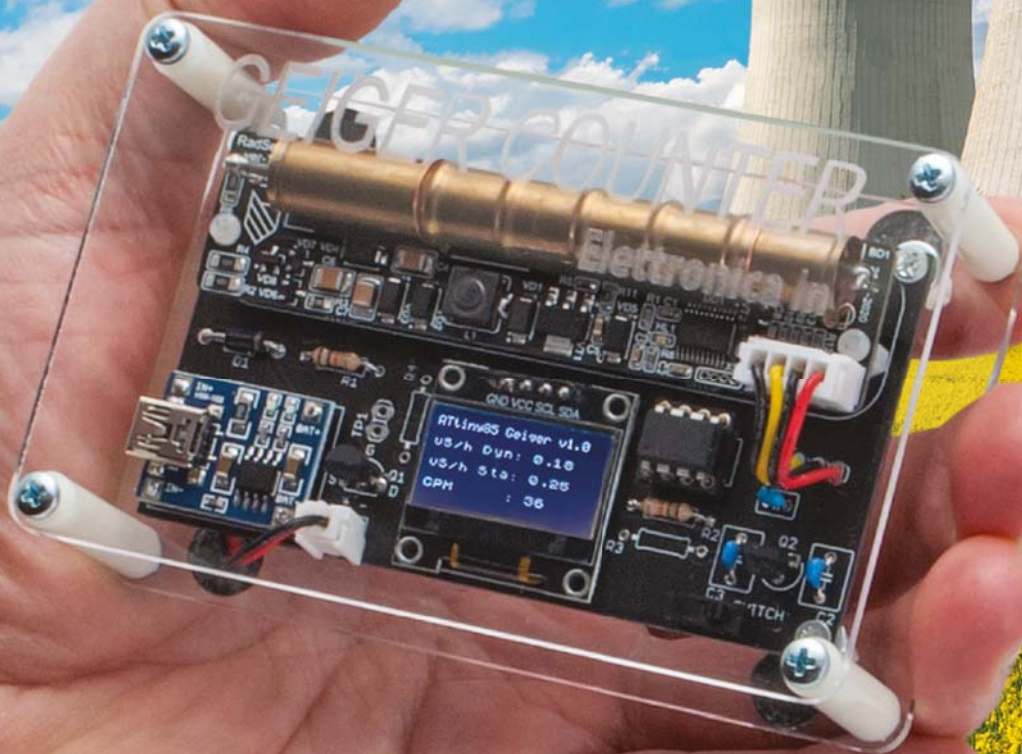
PCB da 2 layer
da 10x10 cm
spessore 1,6 mm
solder verde
serigrafia bianca
€ 6 cad.

www.futurashop.it

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

**FUTURA
ELETTRONICA®**

ATTINY85 RADSENSE: UN CONTATORE GEIGER TASCABILE



Utilizziamo un modulo con tubo Geiger per costruire un rilevatore tascabile di radiazioni. Il tutto gestito da un piccolo ATtiny85 e alimentato con una batteria al litio ricaricabile.

di GUGLIELMO BRAGUGLIA

S

i sente (troppo) spesso parlare di nucleare e sappiamo tutti quali conseguenze può portare un'esposizione prolungata alle invisibili radiazioni. Non sono molti i modi per scoprire se la zona in cui viviamo è "infetta", ma tra questi il contatore Geiger Muller è sicuramente tra i più conosciuti.

Partendo da queste considerazioni abbiamo scelto di realizzare un contatore Geiger «tascabile» dalle ridottissime dimensioni, (soli 9cm x 5cm) in grado di funzionare con una semplice batteria LiPo per un'intera giornata e dotato di carica batteria integrato, utilizzabile anche per la sua alimentazione. Il progetto sfrutta un potente, ma estremamente compatto, modulo, che effettua fisicamente la rilevazione delle "radiazioni" e la loro "conta".

Il sensore, chiamato RadSense, è dotato di un piccolo LED che lampeggia proporzionalmente agli impulsi ricevuti, quindi è semplice capire che stia lavorando correttamente.

Il RadSense è connesso ad un piccolo ATtiny85 che svolge il compito di gestire l'acquisizione dei dati (via bus I2C) e la loro visualizzazione su un display OLED da 0.96".

Assieme a questi due componenti di base viene utilizzato un modulo per la ricarica delle batterie LiPo, una circuiteria per permettere l'utilizzo del "contatore" anche durante la ricarica della batteria ed un regolatore di tensione per avere i 3,3V, necessari a far funzionare tutta l'elettronica.

Il cuore del progetto è certamente il modulo RadSense, visibile in **Fig. 1** e le cui caratteristiche sono riportate nella **Tabella 1**. Tale modulo monta di base un tubo Geiger SBM 20-1, ma è in grado di montare anche tubi differenti quali J305, M4011, STS-5 ed altri. A bordo ha tutta l'elettronica necessaria per generare l'alta tensione (400 V) partendo dalla tensione di alimentazione di 3,3V e una MCU STM32 che svolge i compiti più gravosi di acquisizione dei dati, gestione dell'elettronica (è in grado di spegnere ed accendere il convertitore ad alta tensione, ai fini di risparmio energetico, attraverso comandi software) e colloquio con una MCU esterna via bus I2C. Ha dimensioni estremamente ridotte, solo 89 x 21 x 16 mm ed è facilmente fissabile sul circuito stampato su cui è basato il nostro progetto.

A completamento dell'hardware, su GitHub, è possibile scaricare la libreria, direttamente installabile nell'IDE di Arduino, per la sua gestione attraverso semplici chiamate a funzioni (<https://github.com/climateguard/RadSens>).

Il firmware presente a bordo di detto modulo, e che gira

→ **Fig. 1**
Il modulo
RadSense.



all'interno del STM32, permette di avere già pronte le seguenti misurazioni recuperabili via I2C:

- Intensità delle radiazioni ($\mu\text{R}/\text{h}$) con l'utilizzo di un algoritmo per il calcolo della media su lunghi periodi (500 secondi).
- Intensità delle radiazioni ($\mu\text{R}/\text{h}$) con l'utilizzo di un algoritmo dinamico per una rilevazione più immediata di radiazioni locali.
- Numero degli impulsi ricevuti; valore da utilizzare tra due letture per poter approssimare il valore cps (*count per second*) o cpm (*count per minute*).

Dato che molti lettori non sono attrezzati per lavorare con componentistica SMD e dato che comun-

que, causa le dimensioni del modulo Geiger, vi era lo spazio necessario per posizionare i componenti tradizionali, si è scelto di utilizzare tutti componenti "through-hole" facilmente montabili e saldabili da qualsiasi appassionato.

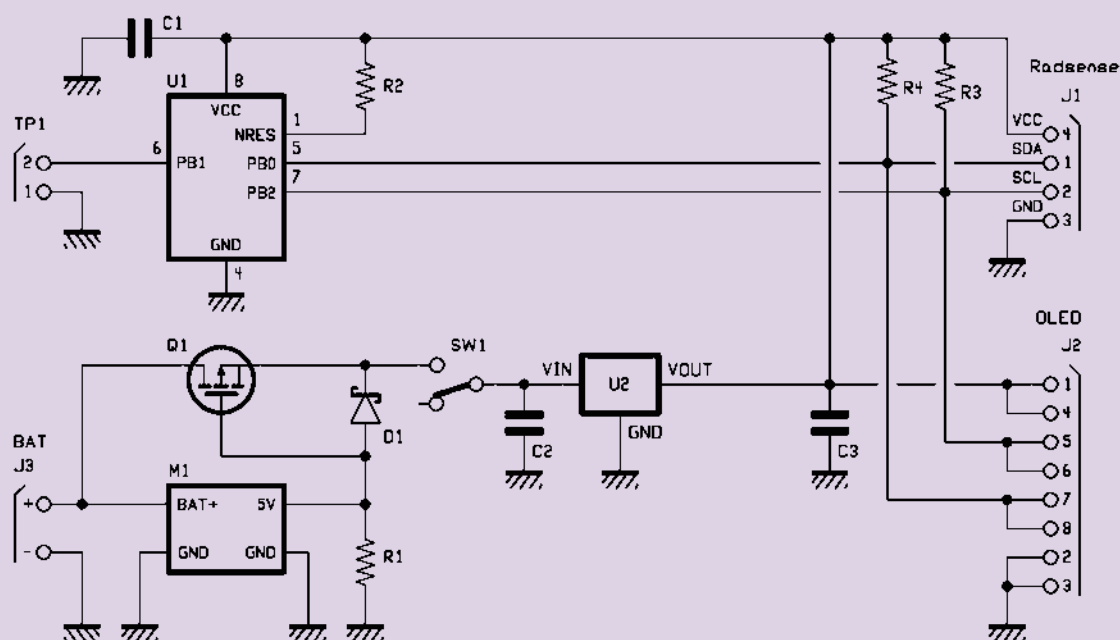
LO SCHEMA ELETTRICO

Analizziamo lo schema elettrico che, come potete vedere nella pagina seguente, è molto semplice. Partiamo dalla sezione di alimentazione e in particolare dal modulo carica batterie LiPo (codice 1ALITIOBATT disponibile su sito www.futurashop.it). Il circuito è in grado di fornire fino ad 1A ed è dotato di un connettore USB che sfrutteremo sia per ricaricare la batteria che, volendo, per alimentare il nostro circuito. Per ottenere quest'ultima possibilità, è presente un MOSFET (Q1) che, quando si fornisce alimentazione tramite la porta USB, isola la batteria dal resto del nostro circuito e utilizza i

↓ **Tabella 1**
Caratteristiche del modulo RadSense.

CARATTERISTICA	VALORE
Tubo Geiger-Muller	SBM20-1
Intervallo radiazioni accettato in $\mu\text{R}/\text{h}$	Da 14.4 a 144000.0
Rumore generato dal circuito stesso	Non più di un impulso al secondo
Sensibilità alla radiazione gamma (sorgente 37 Cs), imp / μR	Da 160 a 75
Tensione di alimentazione del modulo e corrente	Da 3.0V a 3.5V, assorbimento max. 50 mA
Interfaccia e clock	Bus I2C, clock 400 KHz
Indirizzo I2C del modulo	0x66
Temperatura di lavoro	Da -20°C a $+60^{\circ}\text{C}$
Dimensioni in mm	89 x 21 x 13.5

schema ELETTRICO



5V come tensione d'ingresso per il regolatore U2, un MCP1702-330, che provvede a generare i 3,3V necessari. Quando la USB non è connessa, Q1 invia la tensione della batteria al detto regolatore. I pin PB0 e PB2 del ATtiny85 (U1) sono utilizzati come bus I2C (tramite la classica libreria *Wire.h*), mentre il pin PB1 è utilizzato semplicemente come "test point". Se tutto funziona regolarmente tale pin cambia stato con periodo di un secondo, se invece, in fase di avvio del programma, viene rilevata un'anomalia, tale periodo cambia (se non si riesce ad inizializzare la libreria del sensore RadSense il programma si blocca e su detto pin viene generato un segnale che cambia di stato ogni 100 msec.). Per il modulo OLED abbiamo previsto due file di pin, difatti, purtroppo, tali schermi OLED esistono con due piedinature diverse (con i pin GND e VCC invertiti) ed in questo modo è possibile montare sul circuito entrambi i tipi.

Come vedete dal piano di montaggio, la fila superiore di piazzole per il display OLED da 0.96" prevede la piedinatura: GND, VCC, SCL, SDA, mentre la fila inferiore prevede: VCC, GND, SCL, SDA.

Per terminare segnaliamo che le resistenze R3 ed R4, con funzione di pull-up sui segnali SCL ed SDA del bus I2C, sono state previste ma non montate

dato che i moduli OLED da 0.96" già le incorporano. Come batteria LiPo si suggerisce di utilizzare una batteria con capacità di 800/1000mA che, dato l'esiguo assorbimento del circuito (circa 70/80 mA), ne garantirà il funzionamento per un'intera giornata lavorativa. Prestare solo attenzione alla polarità del connettore: come visibile dal piano di montaggio il positivo è a sinistra ed il negativo è a destra sul connettore JST-PH. Come il resto dei componenti anche il cuore del progetto, il microcontrollore ATtiny85 è in formato DIP8 ed è stato quindi montato su zoccolo così da permettere la facile estrazione e reinserimento sia per la sua programmazione iniziale che per l'applicazione di eventuali futuri aggiornamenti del codice.

IL CODICE

Passiamo ora alla parte "software" del nostro circuito, ovvero quello che gira all'interno del ATtiny85 e che ne controlla tutte le funzionalità. Come "core" per la programmazione del ATtiny85, attraverso l'IDE di Arduino, si è utilizzato quello di Spence Konde "ATtinyCore" che si installa attraverso il "Board Manager" del IDE dopo aver aggiunto, dal menu "Preferenze", il seguente link:
http://drazzy.com/package_drazzy.com_index.json

▼ Listato 1

```
#include <Wire.h>
#include <radSens1v2.h>
#include <SSD1306Ascii.h>
#include <SSD1306AsciiWire.h>
#define VCC_MIN      2900 // minimum voltage on Vcc in milliVolt
#define TP_PIN       1    // Test Point Pin
#define TP_PIN_TIME   1000 // Normal TP interval
#define OLED_I2C      0x3C // OLED I2C address
#define OLED_TIME     1000 // OLED display period
#define ROW_DYN       2    // OLED row for radIntensyDyanmic
#define ROW_STA       4    // OLED row for radIntensyStatic
#define ROW_CPM       6    // OLED row for CPM
```

Oltre a questo occorre installare, nella cartella delle librerie, la libreria per la gestione del RadSense (<https://github.com/climateguard/RadSens>) ed una libreria, particolarmente “leggera” (sotto il profilo utilizzo di risorse), per la gestione del OLED, la “SSD1306Ascii”, installabile attraverso il “Library Manager” del IDE o, comunque, scaricabile dal link: <https://github.com/greiman/SSD1306Ascii>

Il programma inizia includendo tutte le librerie necessarie e prosegue con la definizione di una serie di costanti (*#define*) che, con nomi mnemonici, ci permettono una più chiara e leggibile scrittura del codice come visibile nel **Listato 1**.

Vengono poi istanziate le classi necessarie e definite le variabili globali (**Listato 2**).

Terminata questa fase “dichiarativa”, il programma prosegue con una serie di funzioni che vengono richiamate durante l’esecuzione. L’unica degna di nota e di essere mostrata, è quella che implementa il cosiddetto “Secret Voltmeter” (**Listato 3**): possibilità che è presente nelle MCU della vecchia

famiglia Atmel AVR, ovvero la capacità di leggere e conoscere da programma la tensione di alimentazione presente sul pin VCC.

Tale funzione, se chiamata, restituisce il valore della tensione di alimentazione in mVolt.

Nel programma tale valore viene confrontato con un valore fisso, determinato dalla definizione:

```
#define VCC_MIN 2900 // minimum VCC voltage in mV
```

Se il dato letto scende sotto tale valore (modificabile da programmatore secondo le esigenze), il programma si blocca e chiede all’operatore di ricaricare la batteria. Il resto del codice è molto semplice e non viene qui riportato, ma è ovviamente scaricabile per intero dal sito di Elettronica In.

In pratica esso, con una determinata periodicità (fissata a 30 secondi), legge le informazioni dal modulo RadSense e le visualizza sul display OLED. I valori di radioattività vengono visualizzati sul display OLED in $\mu\text{Sv/h}$ (*micro Sievert/ora*) con la

▼ Listato 2

```
SSD1306AsciiWire oled;

ClimateGuard_RadSens1v2 radSens ( RS_DEFAULT_I2C_ADDRESS );

uint8_t      fFirstReading   = true; // first 15 sec waiting for CMP
uint8_t      rad_Status      = 0;    // radSense return status
uint32_t     radNumberOfPulses = 0;   // the number of registered impulses
uint32_t     lstNumberOfPulses = 0;   // last number of registered impulses read from radSense
uint32_t     numberOfPulsesMin = 0;   // the number of impulses per minute (CPM)
uint32_t     radLastMillis    = 0;    // last time RadSense was read
uint32_t     pulseLastMillis  = 0;    // last time CMP was calculated
uint32_t     oledMillis       = 0;    // last time OLED was updated
float        radIntensyDyanmic = 0.0; // radiation in uR/h with a dynamic range of calculation time
float        radIntensyStatic  = 0.0; // radiation in uR/h with averaging algorithm (for long time period)
uint8_t      tp_Status       = LOW;   // TP (Test Point) initial status
uint32_t     tpLastMillis     = 0;    // TP last change status time
```

Listato 3

```
uint32_t readVcc() {
    uint8_t low;
    uint8_t high;
    uint32_t result;
    // Read 1.1V reference against AVcc
    // set the reference to Vcc and the measurement to the internal 1.1V reference
    #if defined(__AVR_ATmega32U4__) || defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__)
        ADMUX = _BV ( REFS0 ) | _BV ( MUX4 ) | _BV ( MUX3 ) | _BV ( MUX2 ) | _BV ( MUX1 );
    #elif defined (__AVR_ATtiny24__) || defined(__AVR_ATtiny44__) || defined(__AVR_ATtiny84__)
        ADMUX = _BV ( MUX5 ) | _BV ( MUX0 );
    #elif defined (__AVR_ATtiny25__) || defined(__AVR_ATtiny45__) || defined(__AVR_ATtiny85__)
        ADMUX = _BV ( MUX3 ) | _BV ( MUX2 );
    #else
        ADMUX = _BV ( REFS0 ) | _BV ( MUX3 ) | _BV ( MUX2 ) | _BV ( MUX1 );
    #endif
    delay ( 2 ); // Wait for Vref to settle
    ADCSRA |= _BV ( ADSC ); // Start conversion
    while ( bit_is_set ( ADCSRA, ADSC ) ); // measuring
    low = ADCL; // must read ADCL first - it then locks ADCH
    high = ADCH; // unlocks both
    result = ( high << 8 ) | low;
    result = 1125300L / result; // Calculate Vcc (in mV); 1125300 = 1.1*1023*1000
    return result; // Vcc in millivolts
}
```

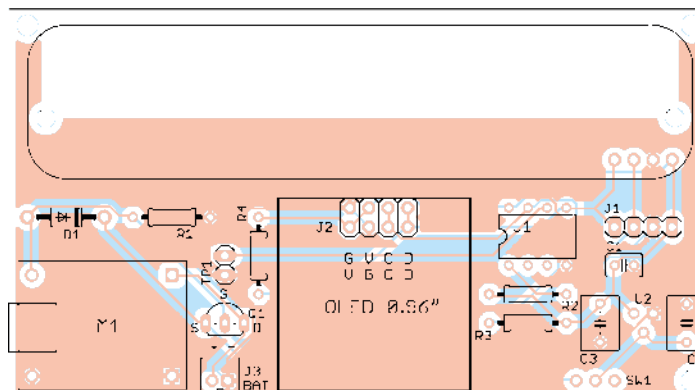
piano di MONTAGGIO

Elenco Componenti:

R1, R2: 10 kohm
 R3, R4: 4,7 kohm
 C1: 0,1 µF multistrato
 C2, C3: 1 µF multistrato
 D1: 1N5818
 Q1: VP3203
 U1: ATTINY85-20P
 U2: MCP1700
 M1: LiPo charger
 SW1: deviatore da c.s.
 TP1: pin strip 2 poli maschio
 J1: pin strip 4 poli maschio
 J2: pin strip 2x4 poli maschio
 J3: JST 2 poli P-2 mm

Varie:

-modulo RadSense v1.2
 -distanziale HEX 6 mm M/F M2 Nylon
 -vite M2x4 Nylon
 -dado M2 Nylon
 -Circuito stampato S1603 (93x52 mm)



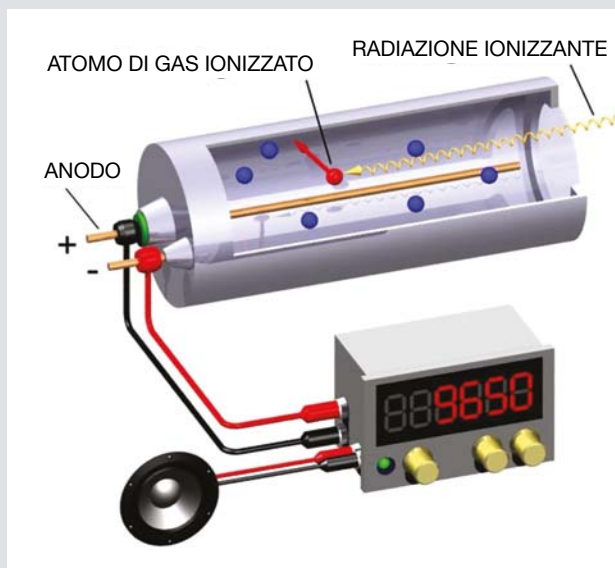
Il contatore Geiger è un apparecchio che permette di misurare le radiazioni ionizzanti ed in particolari le emissioni beta e gamma (in alcuni particolari modelli, anche gli alfa).

Esso è costituito essenzialmente da un tubo metallico (quelli che misurano anche le emissioni alfa hanno una "apertura" di mica, dato che i raggi alfa non attraversano il metallo) contenente una particolare miscela di gas, a bassa pressione, che ha la peculiarità di diventare, per un breve periodo, conduttore se attraversato dalle suddette radiazioni ionizzanti. Il tubo ha al suo interno, al centro, un sottile filo metallico e tra di esso e la superficie del tubo viene creata una differenza di potenziale di alcune centinaia di Volt (il valore dipende dalle caratteristiche del tubo e, in funzione del modello, può andare da circa 400V a circa 1000V). Quando una radiazione ionizzante attraversa il tubo si ha un "impulso elettrico" (dovuto alla temporanea ionizzazione del gas) e, maggiore sarà il numero di particelle che attraversano il tubo per unità di tempo, maggiore sarà in numero di impulsi che potranno essere "contati" nello stesso periodo di tempo.

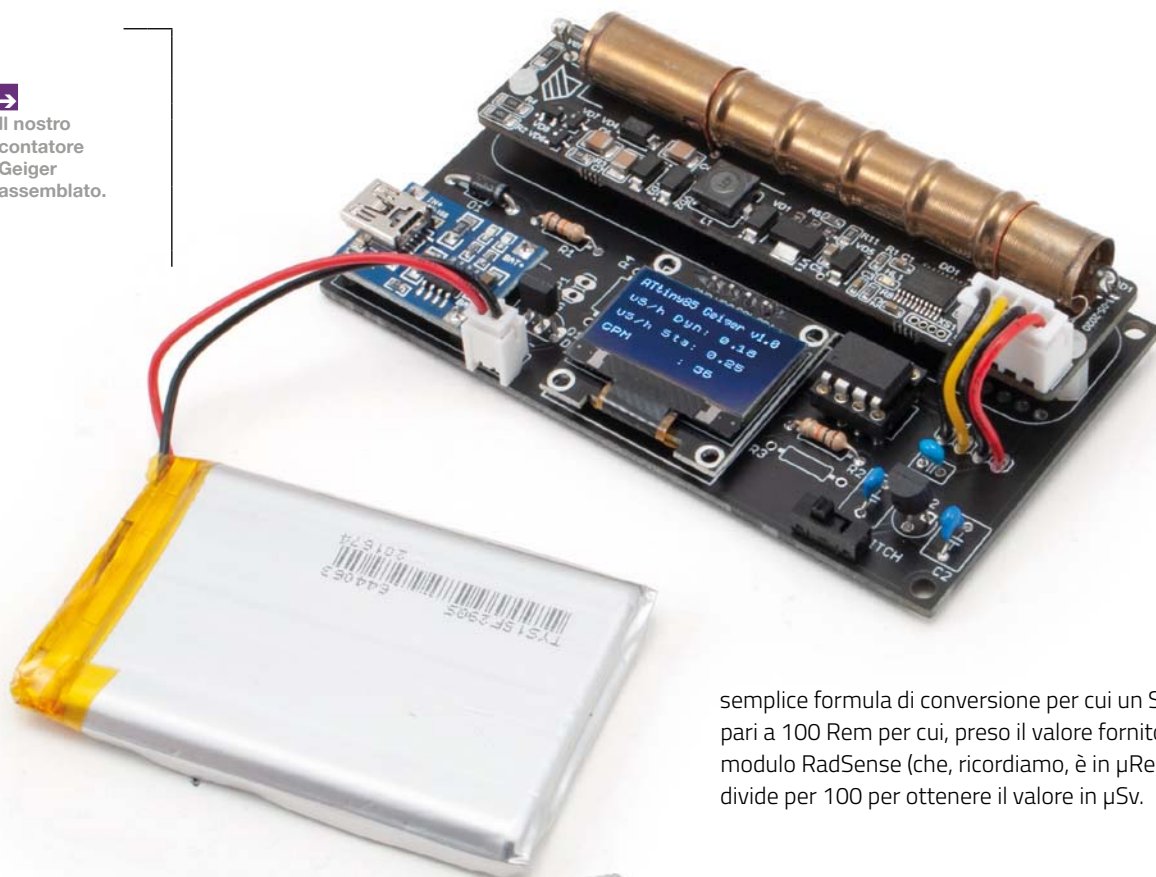
Come abbiamo detto, ogni tubo Geiger funziona solo se alimentato con la giusta tensione, quindi un tubo costruito per una tensione di lavoro di 400 volt, se alimentato con 200 volt non si innescerà mai, mentre, se alimentato con 600 volt, una volta innescato non si disinnescerà più. Nel datasheet fornito dal costruttore è normalmente riportata la tensione minima e massima che si può/deve applicare. Ogni tipo di tubo deve inoltre essere calibrato per permettere la corretta conversione del numero di "impulsi" rilevati per unità di tempo in "Sievert" (o meglio nei

sui sotto multipli), ovvero nell'unità di misura della "dose equivalente" o "dose efficace" di radiazioni.

Una "dose" compresa tra i 0.01 $\mu\text{S/ora}$ ed i 0.50 $\mu\text{S/ora}$ e da considerarsi normale, tra i 0.50 $\mu\text{S/ora}$ ed i 2.00 $\mu\text{S/ora}$ è da considerarsi alta ed oltre i 2.00 $\mu\text{S/ora}$ è da considerarsi pericolosa. Ovviamente i valori indicati sono validi per un'esposizione continuativa e non per una breve esposizione occasionale.



Il nostro
contatore
Geiger
assemblato.



semplice formula di conversione per cui un Sv è pari a 100 Rem per cui, preso il valore fornito dal modulo RadSense (che, ricordiamo, è in μRem) lo si divide per 100 per ottenere il valore in μSv .


```

Board: "ATtiny25/45/85 (No bootloader)"
Chip: "ATtiny85"
Clock Source (Only set on bootloader): "8 MHz (internal)"
Timer 1 Clock: "CPU (CPU frequency)"
LTO (1.6.11+ only): "Enabled"
millis()/micros(): "Enabled"
Save EEPROM (only set on bootloader): "EEPROM retained"
B.O.D. Level (Only set on bootloader): "B.O.D. Disabled (saves power)"
Port: "/dev/cu.URT1"
Get Board Info

```

Fig. 2
Impostazioni dell'IDE di Arduino per il ATtiny85.

COSTRUZIONE

Come già accennato più volte il circuito è stato realizzato con componenti discreti e quindi facilmente saldabile. Prestate attenzione al circuito del sensore che deve essere fissato tramite viti al PCB madre e deve essere collegato a questo tramite un connettore 4 vie. Il modulo caricabatteria viene saldato direttamente sul PCB tramite le sue quattro piazzole.

Il circuito montato può essere inserito all'interno di un piccolo contenitore plastico che può essere stampato in 3D o realizzato tramite taglio laser. Durante l'uso evitate di toccare il tubo Geiger, non tanto per l'alta tensione (circa 400V, ma la corrente fornibile è talmente bassa che praticamente non la si sente), quanto perché toccando il tubo si falsano tutte le misure.

PROGRAMMARE L'ATTINY

Per caricare il firmware all'interno del microcontrollore potete utilizzare l'IDE Arduino.

Per farlo dovrete selezionare la corretta "board" e le corrette impostazioni, per semplificare l'operazione trovate nella **Fig. 2** i parametri esatti.

Il microcontrollore ATtiny85 deve essere programmato tramite un programmatore collegato tramite ISP, programmatore che, volendo, può anche essere realizzato utilizzando una scheda Arduino (AVR) e l'apposito programma "Arduino as ISP" ricordando i collegamenti con l'ATtiny (**Fig. 3**)

Per impostare correttamente i FUSE del ATtiny85 (clock interno a 8MHz, BOD disabilitato, ...), connessa la MCU al programmatore, occorre selezionare, dagli strumenti dell'IDE, l'opzione per "caricare il bootloader". Dato che, come visibile, abbiamo scelto la "board" senza bootloader, in realtà nella MCU non verrà caricato alcunché, ma verranno invece correttamente programmati i FUSE.

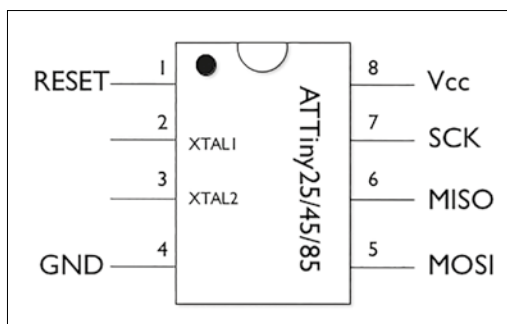


Fig. 3
Collegamenti per la programmazione SPI su ATtiny

CONCLUSIONI

Eccoci giunti alla conclusione di questo progetto. Il sensore RadSense si è rivelato estremamente semplice da utilizzare, grazie principalmente alla sua elettronica che risolve molti problemi. Resta a voi prendere spunto da questo progetto per poter realizzare anche qualcosa di più complesso, ad esempio un sistema in grado di inviare notifiche in caso in cui la radiazione superi certi livelli. Sperando di non averne mai bisogno...



Cosa occorre?

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

Il sensore RadSense (cod. RADSENSE) è disponibile a Euro 110,00, il microcontrollore ATtiny85 (cod. ATTINY85) è in vendita a Euro 4,50, il display OLED (cod. SMALLOLED) costa Euro 10,00, il caricabatterie (cod. 1ALITIOBATT) è disponibile a Euro 3,90, al batteria al litio da 1A (cod. PLIB1000-2MM) è in vendita a Euro 6,90. Il PCB può essere realizzato utilizzando il servizio di stampa PCB (cod. PCBPRODUCTION) direttamente dal sito www.futurashop.it. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

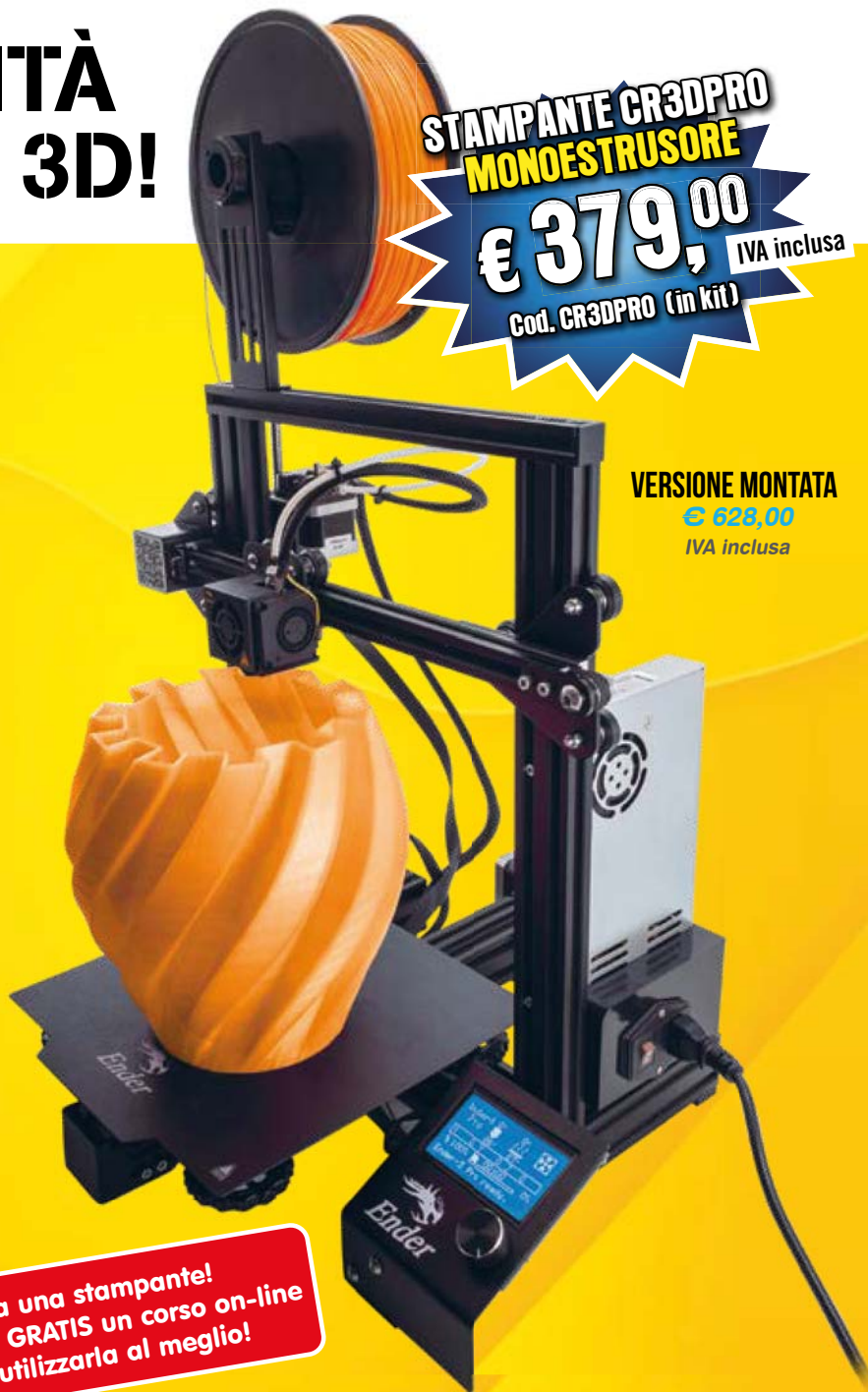
Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 - <http://www.futurashop.it>

ULTIME NOVITÀ DAL MONDO 3D!

CR3DPRO

**STAMPANTE 3D PER STAMPE
FINO A 22 x 22 x 25 cm**

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



VERSIONE MONTATA
€ 628,00
IVA inclusa

Acquista una stampante!
Potrai seguire GRATIS un corso on-line
su come utilizzarla al meglio!



VERSIONE MONTATA
€ 1148,00
IVA inclusa

CRCP3IN1

**MACCHINA 3 IN 1:
STAMPANTE 3D,
INCISORE LASER
E FRESATRICE CNC**

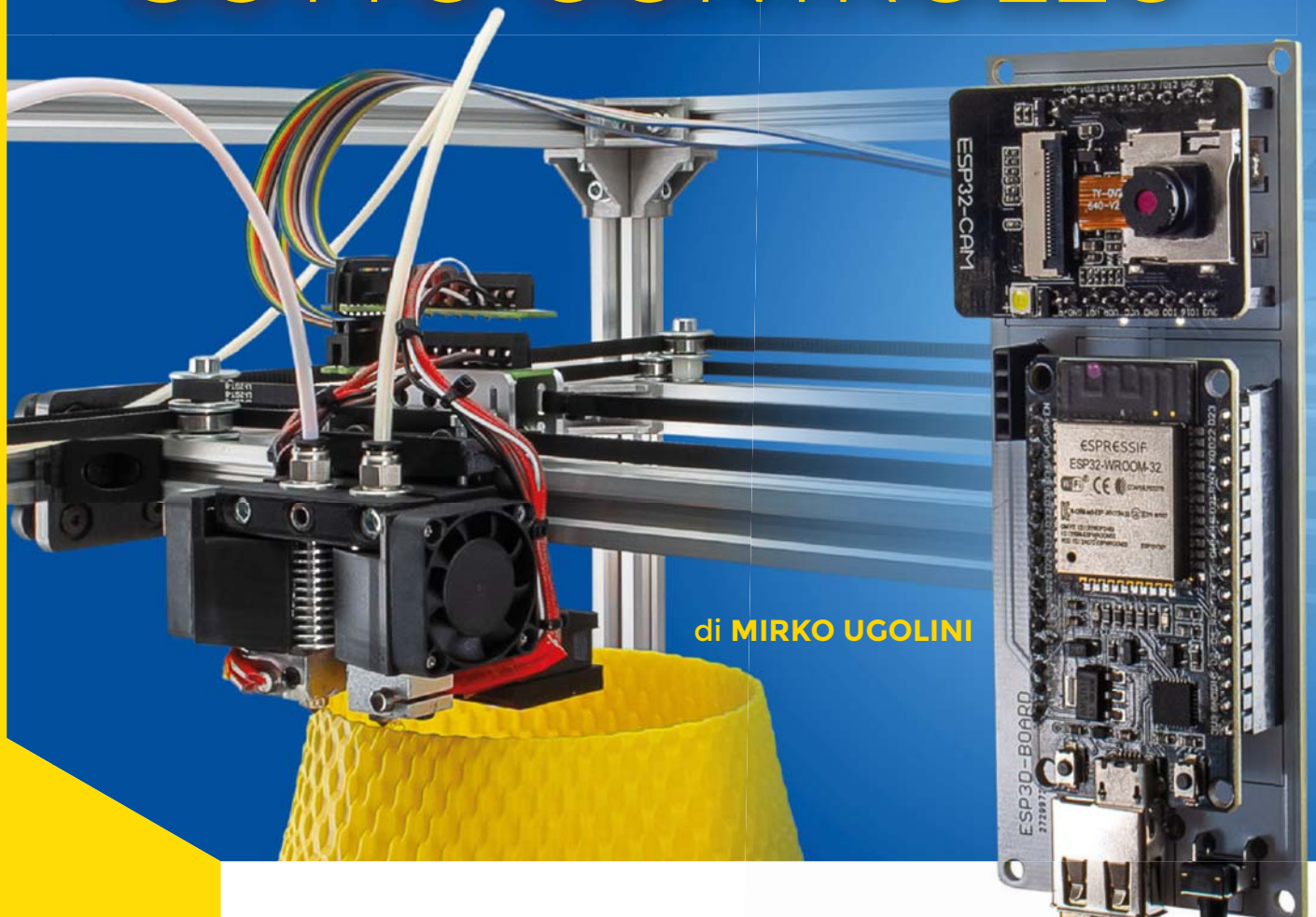
**CRCP3IN1
MACCHINA 3 IN 1**

€ 899,00 IVA inclusa
Cod. CRCP3IN1 (in kit)

- ✓ Telaio in profilati di alluminio V-slot da 40 x 40 mm
- ✓ Scocca in alluminio a struttura aperta
- ✓ 5 motori passo-passo Nema17 (200 step)
- ✓ Spostamento asse Z: doppia vite trapezia Ø8 mm a 3 principi in acciaio INOX
- ✓ Struttura con profilati in alluminio 40 x 40 mm
- ✓ Schermo grafico touch da 4,3"
- ✓ Firmware Marlin open source
- ✓ Connettività SD - USB
- ✓ Alimentazione 200-240 V / 3.4 A
- ✓ Output 24 VDC - 270 W
- ✓ Dimensioni telaio (xyz): 400 x 450 x 650 mm
- ✓ Peso 10 Kg

WIRELESS

LA STAMPANTE 3D SOTTO CONTROLLO



di MIRKO UGOLINI

Configuriamo il modulo ESP3D per ricevere le notifiche di fine filamento e di fine stampa, e per acquisire lo streaming video. Terza ed ultima puntata.

Nel primo articolo (rivista 261), abbiamo introdotto il dispositivo ESP3D che ci permette di controllare da remoto la nostra stampante 3D. Abbiamo spiegato a grandi linee il funzionamento e vi abbiamo messi in condizione di utilizzarlo immediatamente fornendovi i file binari da caricare nei due moduli che compongono ESP3D senza neppure aprire l'IDE di Arduino.

Nel secondo articolo (rivista 262) vi abbiamo spiegato nel dettaglio come preparare l'ambiente di sviluppo per poter

compilare e debuggare sia il codice del firmware da caricare sul modulo ESP32-DEV che il corrispettivo codice Javascript e HTML che compone l'interfaccia grafica del nostro progetto, dando particolare enfasi alla parte relativa alla interfaccia grafica.

Nella puntata corrente vogliamo invece indicarvi come configurare il modulo ESP3D per utilizzare la funzionalità di notifica di fine filamento e di fine stampa, configurare lo streaming video acquisito dalla camera ESP32CAM e configurare i Moduli SonOff per controllare l'illuminazione e l'alimentazione della stampante.

CONFIGURAZIONE

Per poter utilizzare le funzionalità di notifica sul fine filamento è necessario apportare delle semplici modifiche al codice del firmware installato nella nostra Stampante 3D. Al termine delle modifiche dovrete ricompilare il firmware Marlin e caricarlo sulla Board della Stampante. Se non siete pratici relativamente a questa procedura, vi preghiamo di seguire il tutorial presente sui precedenti articoli apparsi sulla rivista.

Iniziamo dunque a modificare il codice del firmware Marlin. Apriamo la cartella **Marlin** all'interno della cartella **Marlin-1.1.x** individuando il file **Marlin.ino** ed apriamolo con l'IDE di Arduino.

In **Fig. 1** vediamo come si presenta la porzione di codice dove è definita la funzione per la gestione di

fine filamento. Notiamo che sono presenti diverse informazioni.

Come prima cosa è possibile abilitare o meno l'opzione di monitoraggio di fine filamento. Nel nostro caso vediamo che la riga **930** non è commentata, quindi l'opzione attiva.

Alla riga **932** vediamo che il numero di sensori sono 2, (nel caso siano presenti due estrusori). Poi sono presenti alla riga **933** e **934** due parametri di configurazione che sono peculiari del tipo di sensore utilizzato. Indicano se la logica deve essere invertita o meno e il fatto che debba essere utilizzato il resistore di pull-up. Se impiegate i sensori **YB785175MM** lasciate la selezione come la trovate.

Alla riga **935** invece abbiamo una riga di codice molto interessante.

Viene definito il GCODE da eseguire nel caso in cui venga individuata la fine di uno dei due filamenti. Di default viene inviato semplicemente il comando **M600**.

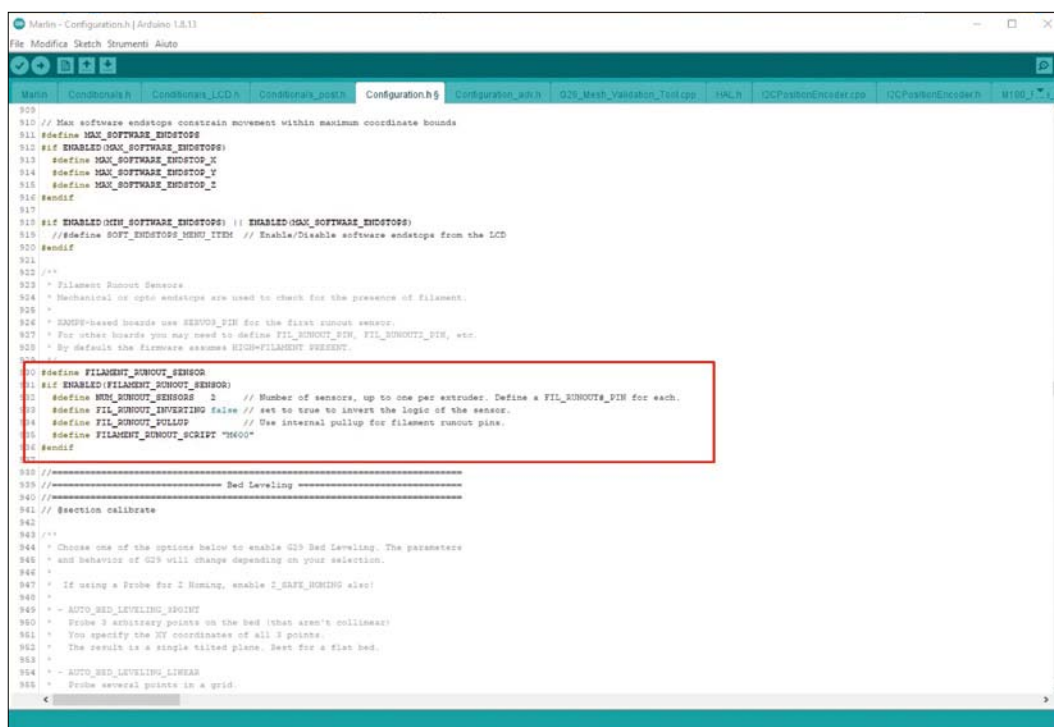
```
#define FILAMENT_RUNOUT_SCRIPT "M600"
```

Al seguente URL troverete tutte le info relative al comando **M600** di Marlin

<https://marlinfw.org/docs/gcode/M600.html>

A noi serve allertare anche il modulo ESP32 che sta in ascolto sulla seriale, e fare in modo che venga inviato un messaggio di notifica all'utente per avvi-

Fig. 1
Porzione del codice che definisce la gestione di fine filamento nel codice Marlin.



```
929 // Max software endstops constrain movement within maximum coordinate bounds
930 #define MAX_SOFTWARE_ENDSTOPS
931 #if ENABLED(MAX_SOFTWARE_ENDSTOPS)
932 #define MAX_SOFTWARE_ENDSTOP_X
933 #define MAX_SOFTWARE_ENDSTOP_Y
934 #define MAX_SOFTWARE_ENDSTOP_Z
935 #endif
936
937 #if ENABLED(MAX_SOFTWARE_ENDSTOPS) || ENABLED(MAX_SOFTWARE_ENDSTOPS)
938 // #define SOFT_ENDSTOPS_MENU_ITEM // Enable/Disable software endstops from the LCD
939 #endif
940
941 /**
942  * Filament Runout Sensor
943  * Mechanical or opto endstops are used to check for the presence of filament.
944  *
945  * RUMPF-based boards use SENSOR_PIN for the first runout sensor
946  * For other boards you may need to define FIL_RUNOUT_PIN, FIL_RUNOUT2_PIN, etc.
947  * By default the firmware assumes HIGH=FILAMENT PRESENT.
948  */
949
950 #define FILAMENT_RUNOUT_SENSOR
951 #if ENABLED(FILAMENT_RUNOUT_SENSOR)
952 #define NUM_RUNOUT_SENSORS 2 // Number of sensors, up to one per extruder. Define a FIL_RUNOUTx_PIN for each.
953 #define FIL_RUNOUT_INVERTING false // set to true to invert the logic of the sensor.
954 #define FIL_RUNOUT_PULLUP // Use internal pullup for filament runout pins.
955 #define FILAMENT_RUNOUT_SCRIPT "M600"
956 #endif
957
958 //===== Bed Leveling =====
959 // Section calibrate
960
961 /**
962  * Choose one of the options below to enable G29 Bed Leveling. The parameters
963  * and behavior of G29 will change depending on your selection.
964  *
965  * If using a Probe for Z Homing, enable Z_SAFE_HOMING also!
966  */
967
968 * - AUTO_BED_LEVELING_3POINT
969 * Probe 3 arbitrary points on the bed (that aren't collinear)
970 * You specify the XY coordinates of all 3 points.
971 * The result is a single tilted plane. Best for a flat bed.
972
973 * - AUTO_BED_LEVELING_LINEAR
974 * Probe several points in a grid.
```

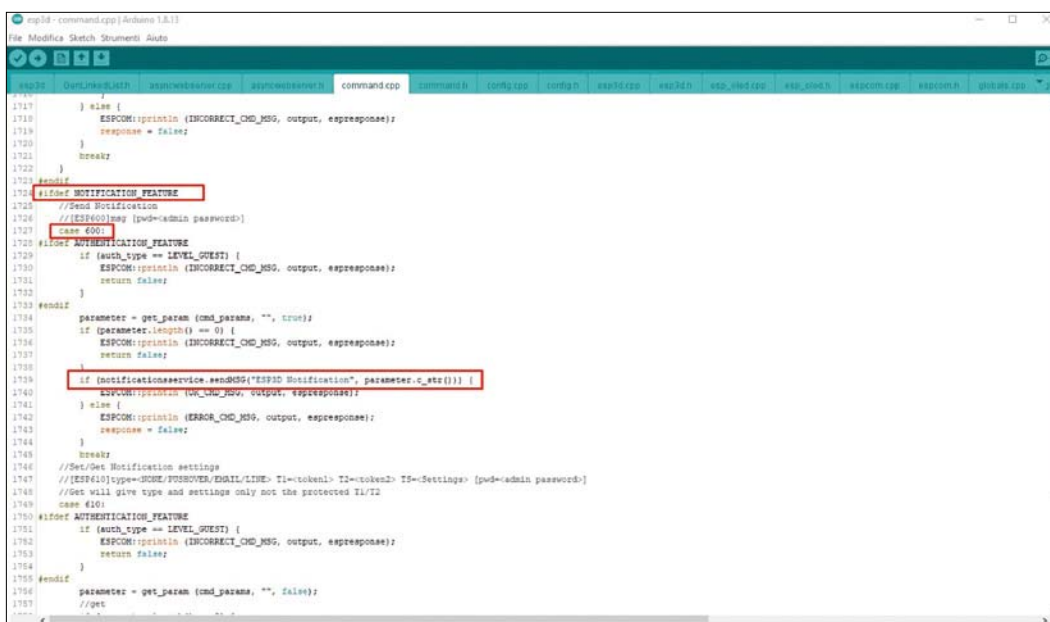


Fig. 2
Porzione di codice del modulo ESP32 dove si definisce la gestione delle notifiche.

sarlo che uno dei due filamenti è terminato. Quindi modificheremo la riga 935 in questo modo:

```
#define FILAMENT_RUNOUT_SCRIPT "M600\nM118 P0
[ESP600] Filamento Terminato"
```

Come vedete abbiamo aggiunto la stringa `\n` "nuova linea" e il comando **M118 P0** che serve ad inviare i dati sulla seriale dove **P0** è il numero della porta seriale. Poi abbiamo aggiunto la stringa **[ESP600]** **Filamento Terminato**, in cui la prima parte **[ESP600]** va intesa come una estensione specifica del comando M600 ma destinata al modulo ESP3D. Questo comando non lo troverete nella documentazione di Marlin perché è peculiare di ESP3D. Infatti questo comando tra parentesi quadre, viene parsato (interpretato) dal codice che gira su ESP3D e che sta in ascolto sulla seriale.

Quando riceve questo comando sa che deve gestire un messaggio di notifica, e quindi estrae la seconda parte dopo **[ESP600]**, il testo **"Filamento Terminato"** e lo invia tramite il meccanismo di notifica impostato (secondo le nostre impostazioni) su ESP3D. Le info sul comando M118 le potete trovare al seguente url:

<https://marlinfw.org/docs/gcode/M118.html>

In **Fig. 2** possiamo vedere la parte di codice che gira su ESP3D deputata appunto alla gestione dei messaggi di notifica. Alla riga **1724** viene verificato se la funzionalità di notifica è attiva e nel caso lo fosse, alla riga **1734** viene estratto il messaggio da notificare ed alla riga **1739** viene richiamata la funzione deputata all'invio del messaggio.

Andiamo ora a verificare il codice presente su ESP3D relativo alla gestione del comando **ESP600**. In **Fig. 3** notiamo anche che avendo definito due distinti sensori di termine filo dovremo avere anche due pin assegnati. Infatti nella nostra scheda i pin definiti per tale scopo e peculiari per la scheda **3Drag** li troveremo dentro il file **pins_3DRAG.h**.

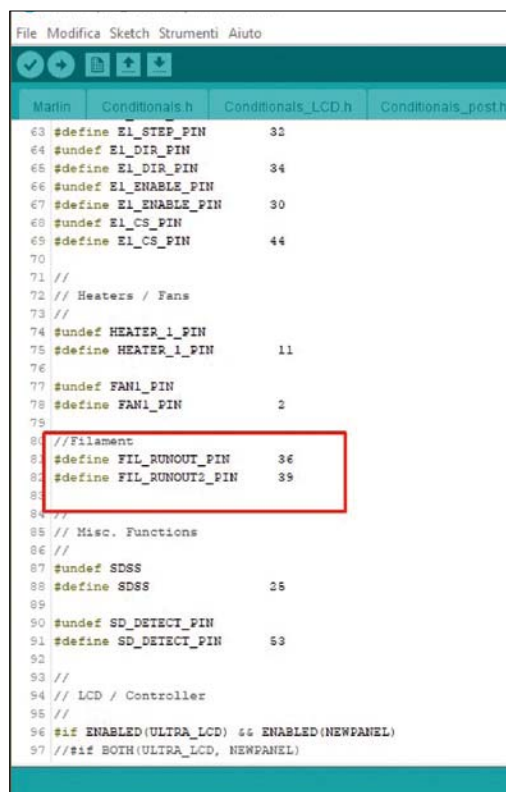


Fig. 3
Codice Marlin dove si dichiarano i PIN destinati ai sensori di fine filamento.

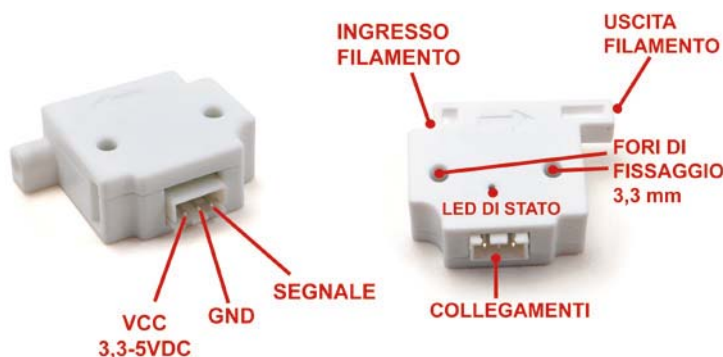


Fig. 4
Sensori di fine
filamento e loro
pinout.

I pin interessanti sono il **36** e **39**.

In **Fig. 4** vediamo come si presentano i sensori di fine filamento che abbiamo adottato.

Il codice Futura Elettronica dei sensori impiegati è **YB785175MM**.

Il funzionamento è molto semplice. Il filamento, prima di essere inserito nell'estrusore deve essere fatto passare attraverso il corpo del sensore e nella parte centrale il filamento stesso viene in contatto con la linguetta di uno microswitch. Quindi se il filamento è presente all'interno del corpo del sensore il microswitch viene mantenuto chiuso, nel momento in cui viene a mancare il filamento, il microswitch si aprirà notificando l'evento alla motherboard. E' presente anche un led che indica lo stato del microswitch.

Attualmente sono state implementate tre modalità di notifica configurabili su ESP3D:

NOME	NOTE	GRATUITO
Line	https://line.me	Si
Pushover	https://pushover.net	No
Email	SMTP "per esempio Gmail"	Si

NOTIFICA DI FINE STAMPA

Possiamo utilizzare il sistema di notifica anche per essere avvisati quando la stampa è terminata.

Per fare questo è sufficiente inviare il seguente comando sulla seriale **"M118 P0 [ESP600] Stampa Terminata"** al termine della stampa.

Bene, questa cosa può essere fatta in due modi diversi: aprire il singolo file gcode del nostro modello da stampare ed accodare la stringa indicata, oppure un modo molto più pratico è quello automatizzare questo processo facendo fare l'accodamento al software di Slicing che utilizziamo.

Per esempio nel caso di **Simplify3D**, se selezioniamo il nostro processo di stampa es. **"Process1"** e poi clicchiamo su pulsante **"Edit Process Settings"** si

aprirà la schermata di **Fig. 5**, ora andiamo a selezionare il Tab **Scripts** e poi il Tab **"End Script"**.

Bene, quella parte di codice verrà sempre appesa al termine della sequenza di comandi Gcode che andranno a costituire il file Gcode che salveremo sulla SD o che verrà inviata tramite USB alla nostra stampante 3D. Come potete vedere, ci sono dei comandi di utility che servono a spegnere gli estrusori, il piatto, fare risalire l'estrusore di 5 mm, posizionarsi sulla Home X e Y, disabilitare i motori e poi per ultimo, l'invio del comando tramite seriale verso ESP3D del comando ESP600 ed il relativo testo di notifica da inviare all'utente.

CONFIGURAZIONE LINE

Per prima cosa è necessario installare l'App Line per Android o per IOS utilizzando le indicazioni proposte per esempio tramite numero di telefono, poi andare nelle impostazioni dell'App selezionare la voce **Account**, impostare l'email e la password, terminata la configurazione dell'account andate sul sito <https://notify-bot.line.me/my/> ed effettuate il login. Al termine del login sarà possibile generare i Token (**Fig. 6**).

Cliccate sul pulsante e nella pagina successiva che si aprirà digitate il nome dell'applicazione per esempio ESP3D e selezionate la chat **"1-on-1 chat LINE Notify"** e poi cliccate sul pulsante **"Generate Token"**. Si aprirà un popup con all'interno il Token generato, copiatelo nel blocco note prima di chiudere il popup altrimenti non riuscirete a recuperarlo e dovrete ripartire da capo.

In ogni caso potrete generare quanti token desiderate ed eliminare quelli non usati.

Per configurare ESP3D ad usare LINE come sistema di notifica è sufficiente selezionare la voce di menu **"Line"** ed inserire il valore acquisito del Token dalla configurazione nel rispettivo campo **Token1**.

Notare che nel caso di LINE è sufficiente un solo Token (**T1**) per configurare il sistema.

CONFIGURAZIONE PUSHOVER

Visitate il sito <https://pushover.net>, create un account e scaricate sul vostro dispositivo mobile la relativa App.

Sempre sul portale di PushOver loggatevi con le vostre credenziali, vi verrà mostrata la pagina di **Fig. 7** in cui è presente il **Token1** che dovremo utilizzare per inviare messaggi. In basso è necessario creare una Applicazione.

Noi l'abbiamo denominata ESP3D ma potreste scegliere qualunque nome vi aggradi. Dopo aver creato l'applicazione, apprendete il dettaglio come

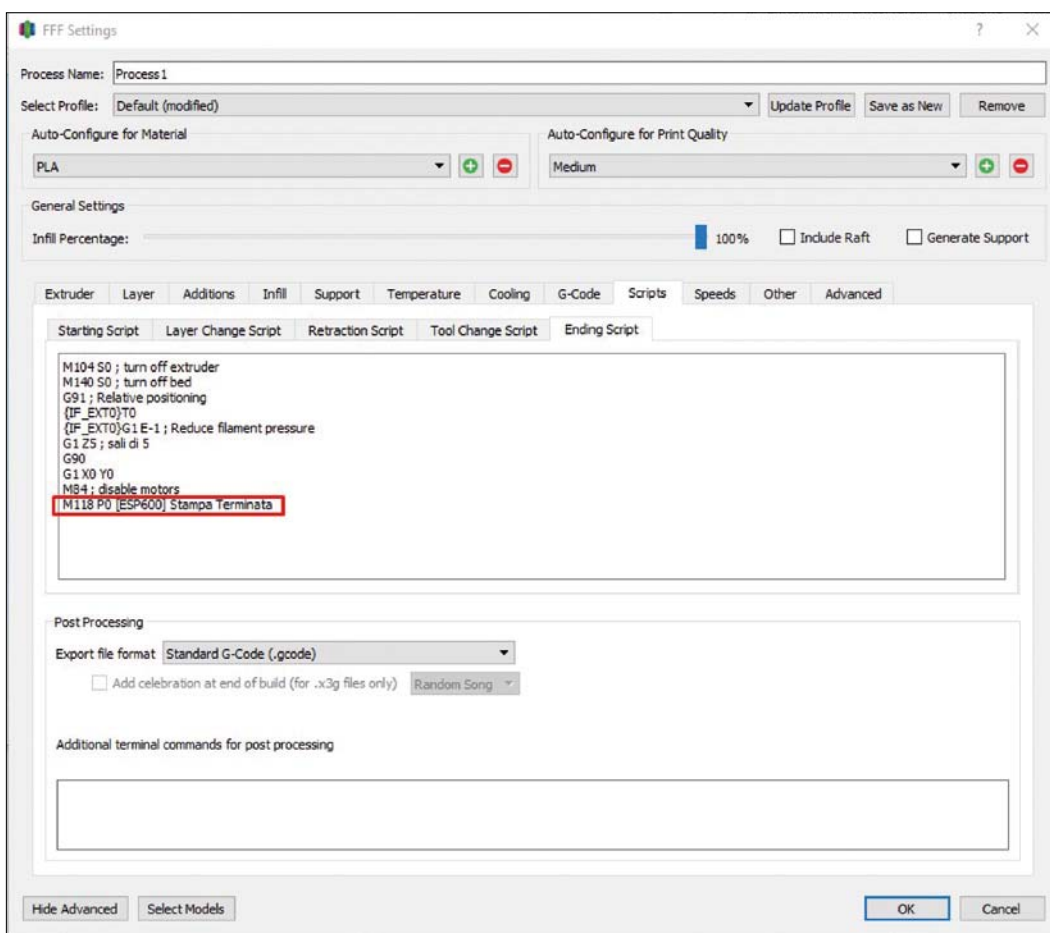


Fig. 5
Configurazione
dello script di
fine stampa
nel caso di
Simplify3D.

in **Fig. 8** potrete accedere al **Token2** specifico per l'applicazione.

Per configurare ESP3D ad usare Pushover come sistema di notifica è sufficiente selezionare la voce di menu **"Pushover"** (**Fig. 9**) ed inserire il valore acquisiti dei due Token dalla configurazione nei rispettivi campi, **Token1** e **Token2**

Per testare l'invio di un messaggio di notifica digitate il seguente testo (chiaramente escluse le virgolette) **"[ESP600] Test Messaggio Pushover"** nella input box di **Fig. 10** e cliccate su "Send"

CONFIGURAZIONE EMAIL

Per utilizzare l'email come sistema di notifica servono in pratica tre parametri: **T1** sarà la username dell'account Email del mittente, **T2** la rispettiva password e infine **TS** che definisce il destinatario del messaggio, il server e porta da utilizzare. E' fortemente sconsigliato l'utilizzo del vostro account email principale come account per l'invio. Createvi quindi un account separato per invio dei messaggi perché è facilissimo creare dei messaggi che potrebbero essere considerati spam dal server

e finire bannati per un eventuale errore. Nel caso utilizzate Gmail come server per l'invio, è

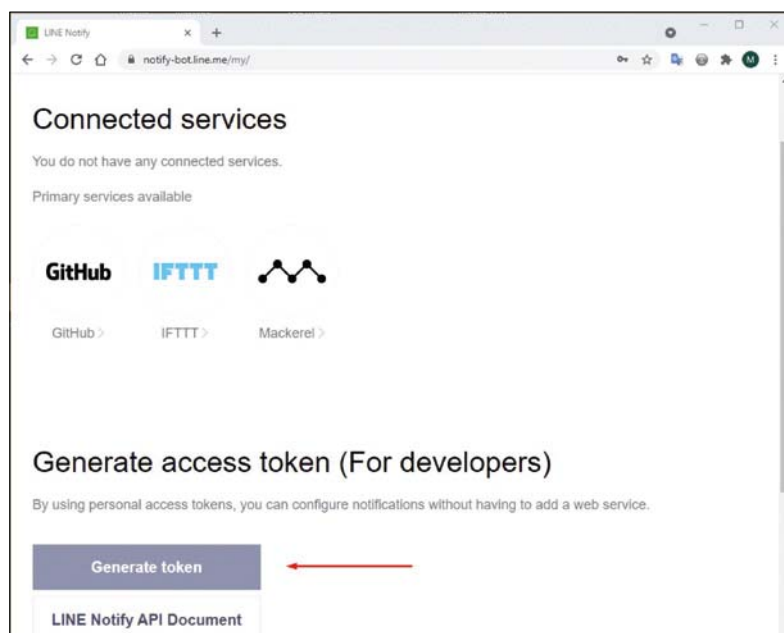
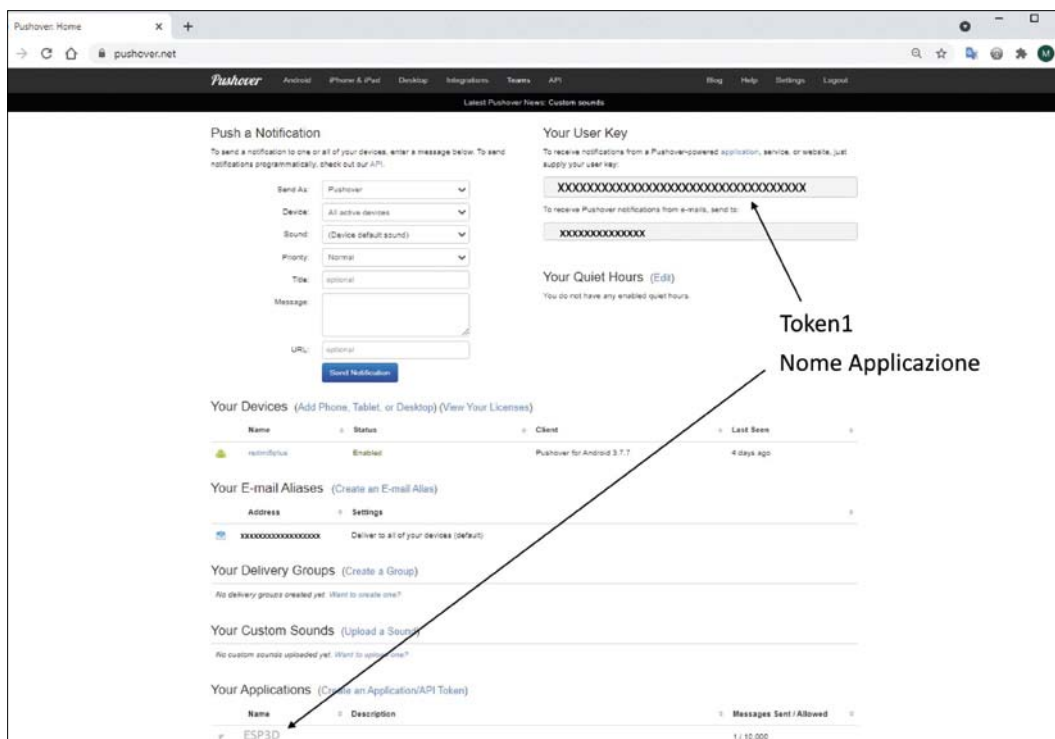


Fig. 6
Dove generare i
token per la notifica
nel caso di Lime.



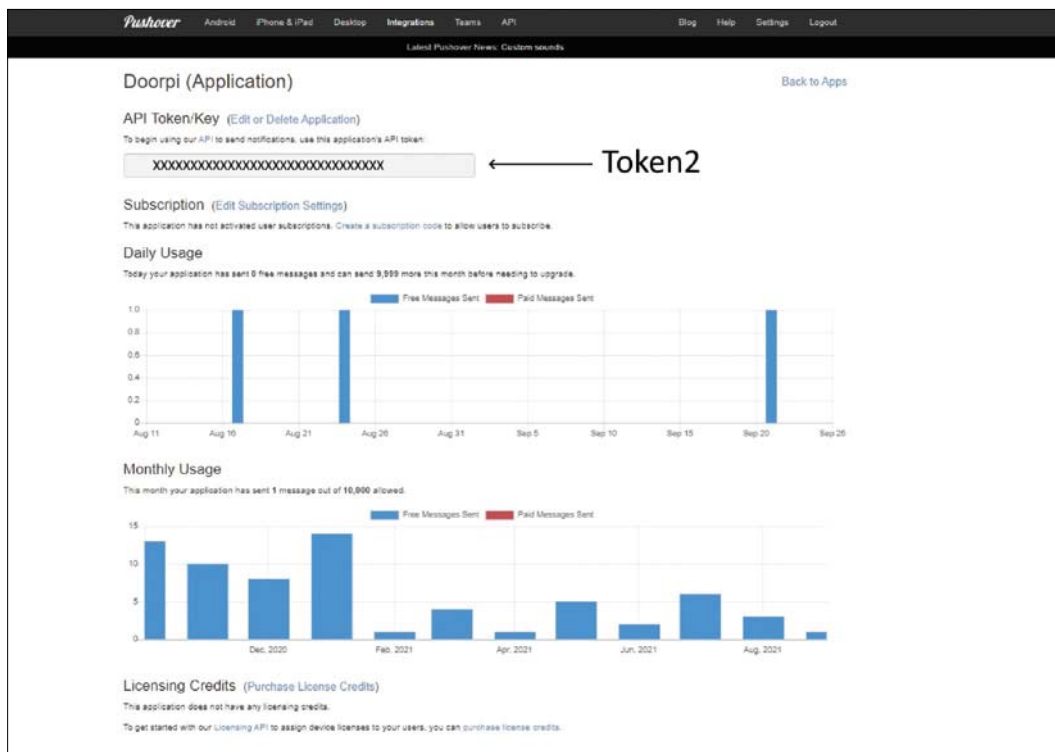
Fig. 7
Schermata della
configurazione di
PushOver per
Token1.



necessario abilitare una opzione visitando il seguente url: <https://myaccount.google.com/lesssecureapps> per permettere applicazioni non sicure di utilizzare l'account.

Per configurare ESP3D ad usare Email come sistema di notifica è sufficiente selezionare la voce di menu "Email" (Fig. 9) ed inserire il l'email dell'account Email nella input box **Token1**, la password

Fig. 8
Schermata della
configurazione di
PushOver per Token2.



dell'account nella input box **Token2**, il destinatario dei messaggi ed il server da utilizzare nella input box **"Notifications Settings"**.

Per esempio il valore da inserire potrebbero essere

- **Token1:** *email-esp3d@dominio.com*
- **Token2:** *password-email-esp3d*
- **Notification Settings:** *email-destinatario-messaggio@gmail.com#smtp.gmail.com:465*

Per testare l'invio di un messaggio di notifica digitate il seguente testo (chiaramente escluse le virgolette) **"[ESP600] Test Messaggio Email"** nella input box di **Fig. 10** e cliccate su "Send"

QUALE SISTEMA DI NOTIFICA PREFERIRE?

Avrete notato che tra i sistemi di notifica utilizzabili, PushOver non è gratuito.

Ebbene in effetti non lo è ma il costo è veramente esiguo. Cosa solo **4,99\$ (una tantum)** circa **4,26€** al cambio attuale. Si possono inviare fino a 10.000 notifiche senza dover pagare nulla.

Le notifiche sono inviate in modo veloce ed è possibile riceverle nella App dedicata. La configurazione è molto semplice e intuitiva. Quindi ci sentiamo di consigliarvela.

CONFIGURAZIONE CAMERA

Per poter utilizzare la camera è necessario aprire il menu ad Hamburger che si trova in alto a destra della finestra e cliccare sulla opzione **"Preferences"** poi selezionare (**Fig. 11**) la spunta su **"Show camera Panel"**. Al termine, o si clicca sul pulsante "Save" in calce alla pagina oppure se si chiude il popup con la crocetta viene chiesta la conferma di salvataggio dei dati.

Il valore dell'IP e quindi dell'URL che viene impiegato per visualizzare lo streaming video in MJPEG viene acquisito in tempo reale dalla scheda ESP32CAM e viene riportato come indicazione in alto a sinistra del video.

Come certamente saprete non è possibile osservare lo streaming video della camera ESP32Cam da più di un client alla volta, quindi se lo state osservando dalla interfaccia della del pannello di controllo di ESP3D e provate ad aprire una nuova pagina del browser allo stesso url indicato, non vedrete nulla. Questa è una limitazione del modulo ESP32CAM. Come avrete notato la risoluzione del video visualizzato in ESP3d è fissa ad una particolare valore e anche tutti i parametri sono stati settati per ottimizzare la visualizzazione.

Ma questo non toglie che possiate settare questi parametri secondo le vostre particolari esigenze.

Fig. 9
Posizione in cui si definisce la modalità di notifica nella GUI di ESP3D.

Fig. 10
Dove si può testare l'invio di un comando nella GUI di ESP3D.

Per fare ciò lasciate alimentato il modulo ESP3D, ma uscite dal Tab della visualizzazione della camera, in modo che le risorse dello streaming vengano rilasciate. Poi aprite una nuova finestra nel browser e digitate la url che avevate sulla schermata della camera. Supponendo che la url sia *http://192.168.1.170:81/* allora aprite il browser alla pagina *http://192.168.1.170* Omettendo quindi la porta 81 dalla url. Nella pagina si aprirà (**Fig. 12**) la visualizzazione dello streaming

Fig. 11
Dove si definisce se il TAB Camera debba essere presente sulla GUI di ESP3D.



Fig. 12
Configurazione
dei parametri
della Camera
ESP32CAM.

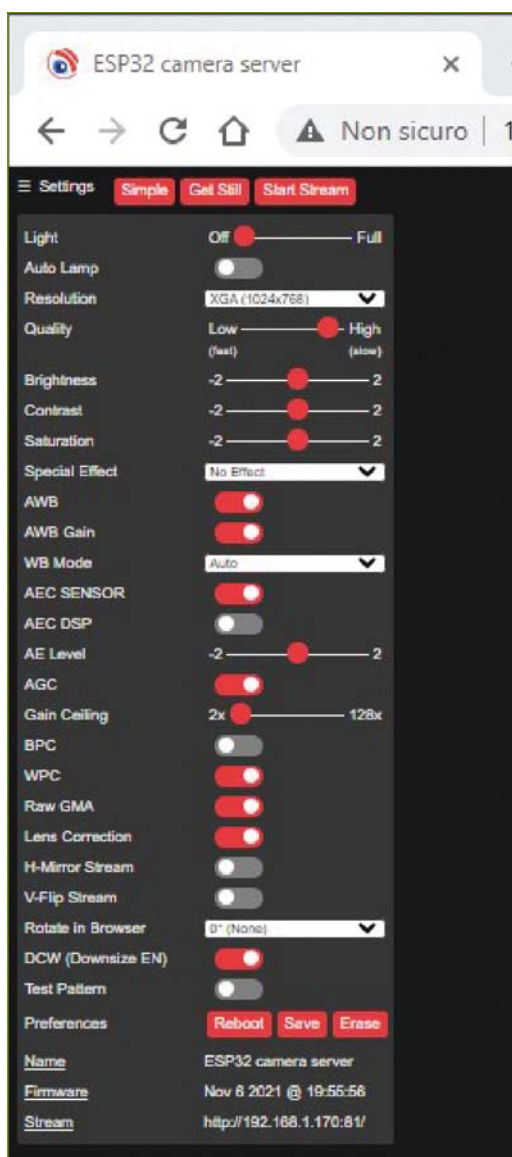


Fig. 13
Posizione in cui
si definisce la
tipologia di modulo
SONOFF e i relativi
indirizzi IP.

video, ma in questo caso avrete anche la possibilità di configurare (cliccando sul pulsante "Full") a vostro piacimento i parametri del video. Quando avrete trovato il giusto mix tra risoluzione, qualità del video,

contrasto etc. chiudete questa pagina del browser e riaprite il Tab camera della applicazione web ESP3D, troverete lo streaming video con le impostazioni che avete configurato precedentemente.

Le configurazioni vengono salvate nello spazio SPIFF nel modulo ESP32CAM in formato Json e richiamate al momento di visualizzare lo streaming.

CONFIGURARE MODULI SONOFF

Per configurare i moduli SonOff è necessario aprire il Tab **ESP3D**, scorrere la lista delle impostazioni fino alla fine e individuare la sezione relativa a SonOff, come in **Fig. 13**.

È possibile notare che la selezione della tipologia di modulo Sonoff prevede la possibilità di utilizzare i moduli nella modalità **DIY** e i moduli sui quali è stato aggiornato il firmware con **Tasmota**. Se dalla combo box si seleziona "**None**" sul Tab di controllo della camera sarà presente solo il pulsante di interruzione della stampa, mentre non saranno visibili i pulsanti relativi ai due moduli SonOff.

Nelle rispettive Texbox è possibile inserire gli indirizzi IP del modulo impiegato per il controllo della Illuminazione e per il controllo della alimentazione della stampante. Anche in questo caso, se non si inserisce un valore per l'indirizzo IP il rispettivo pulsante non sarà visibile nel pannello di controllo della stampante.

CONCLUSIONI

Bene, ora che sapete anche come configurare il modulo ESP3D per la notifica di fine filamento, non ci resta che augurarvi buona sperimentazione e buone stampe 3D. □



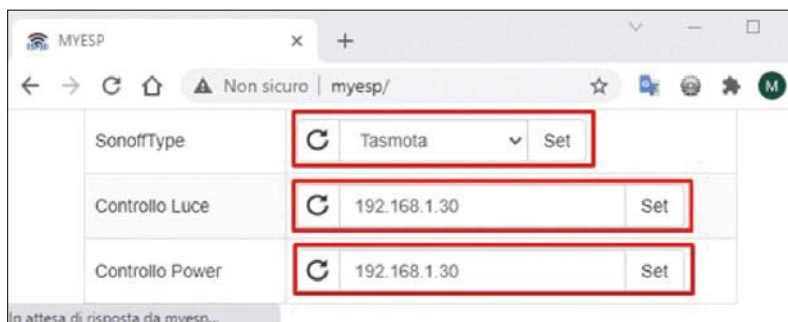
Cosa occorre?

Il materiale principale utilizzato in questo articolo è disponibile presso Futura Elettronica. La board di controllo per stampanti 3D dual extruder completa di driver (cod. FT1370M) è disponibile a Euro 119,00, la board ESP32 con telecamera (cod. ESP32-CAM) è in vendita a Euro 16,50, il sensore fine filo per filamenti da 1,75mm (cod. YB785175MM) costa Euro 9,00.

I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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



Costruisci un vero robot!

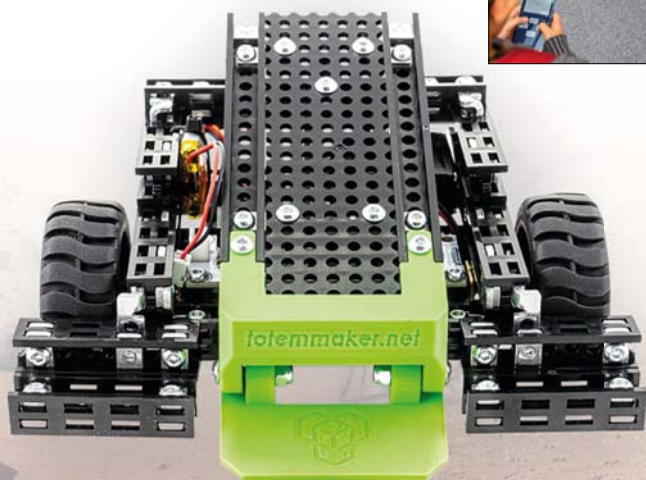


Mini robot da combattimento controllato da App

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

€ 79,⁵⁰

Cod. MINITROOPER



**CONTROLLABILI
TRAMITE
APP!**

Rover con braccio robotico e pinza controllato da App

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



€ 69,⁰⁰

Cod. GRIPPERBOT

Assemblando i robot imparerai:

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

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

Prezzi IVA inclusa

**FUTURA
ELETTRONICA®**

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

Caratteristiche tecniche
e vendita on-line su:

www.futurashop.it



IMPARA L'ELETTRONICA FACILMENTE CON TOTEM MINI LAB



€ 112,00
cod. TOTEMMINILAB

Piccolo ma potente laboratorio di elettronica ideale per sperimentare e apprendere le basi dell'elettronica. Pienamente compatibile con l'ambiente di programmazione Arduino, Totem Mini Lab comprende una varietà di strumenti di misura e blocchi elettronici pronti all'uso per realizzare esperimenti. Pensato per essere utilizzato insieme al sistema di costruzione Totem consente agli utenti di costruire solidi banchi da lavoro con breadboard integrate. Inoltre, il laboratorio è compatibile con schede di espansione I/O, sensori e audio che possono espandere ulteriormente le capacità del Mini Lab. L'utilizzo di Totem Mini Lab è semplice e divertente, ideale sia per ragazzi sia per adulti!

COMPLETALO CON LE SCHEDE DI ESPANSIONE!



ESPANSIONE AUDIO

Con questo pannello sarà possibile avere disponibili e facilmente accessibili i seguenti componenti: amplificatore audio in classe AB con 1 W di potenza, altoparlante integrato e alimentazione a 12 volt, VU-meter con barra a 10 LED, generatore di funzioni in grado di generare segnali fino a 1 MHz.

€ 45,00
Cod. TOTEMAUDIOSIDE



ESPANSIONE I/O

Contiene componenti elettronici che solitamente sono difficili da inserire sulle breadboard. Con questo pannello sarà possibile avere disponibili e facilmente accessibili i seguenti componenti: 3 interruttori a pulsante con indicatori LED, 3 potenziometri da 10 kOhm, 1 encoder rotativo, 3 pulsanti, 1 LED RGB, 1 presa Jack audio da 3,5 mm, 1 Relè da 10 A / 250 V, con LED di stato.

€ 37,00
Cod. TOTEMIOSIDE



ESPANSIONE SENSORI

Con questo pannello sarà possibile avere disponibili e facilmente accessibili i seguenti componenti: un microfono preamplificato, un display OLED 128x64 con interfaccia I²C, un sensore umidità DHT11, un termistore NTC da 10 kohm, un buzzer e un piccolo driver per motori DC.

€ 39,00
Cod. TOTEMSENSORSIDE

DISTRIBUTORE FARMACI CON micro:bit

dell'Ing. VINCENZO GERMANO
e MARCONATALE PARISE



Sfruttiamo l'elevata flessibilità hardware e software di questa scheda dalle piccole dimensioni ma dalle grandi potenzialità per realizzare un distributore di farmaci settimanale interconnesso.

N

ella vita di tutti i giorni, in cui siamo presi da impegni di vario genere, la tecnologia ci ha fornito validi strumenti per poter semplificare molte attività giornaliere, si pensi banalmente a quante applicazioni di questo tipo possano essere utilizzate all'interno di uno smartphone.

Su questo filone, vi illustriamo la realizzazione di un distributore di farmaci settimanale, semplice da implementare grazie alle caratteristiche del micro:bit e facile da utilizzare grazie all'interfaccia utente semplificata mediante display OLED (Organic Light Emitting Diode) sul dispositivo e app per Android. Verranno illustrati tutti i vari passaggi implementativi, a partire dalla descrizione della componentistica hardware, per passare all'analisi software e scrittura del codice necessario per la realizzazione delle funzioni d'interesse e poi concludere con l'applicazione Android (che rendere il tutto il più flessibile e permette una configurazione totale del sistema).

DESCRIZIONE HARDWARE

Come per altre applicazioni presentate, il cuore del progetto è la scheda micro:bit, che, con l'ausilio di alcune *board* aggiuntive, permette una gestione settimanale preconfigurata di un distributore di farmaci automatico. Per ottenere una gestione semplice ed immediata, si è pensato di rendere disponibile all'utilizzatore finale un'applicazione Android per impostare settimanalmente tutto il sistema. Il funzionamento che ne consegue è snello e versatile anche per i meno esperti.

L'idea è quella di definire l'intera settimana da applicazione Android, dopodiché il sistema micro:bit procede in automatico.

Come vedremo successivamente, l'interfaccia dell'applicazione dà modo di immettere il nome e la tipologia del farmaco, il giorno e l'ora in cui è da assumere ed in fine il box corrispondente del particolare farmaco; il tutto per sette giorni. Effettuata questa prima fase di caricamento/impostazione del sistema, il micro:bit attende il primo giorno d'interesse e all'ora esatta, muove i vari box fino a rendere disponibile quello interessato, accende il led del box corrispondente, attiva una segnalazione acustica mediante un buzzer e mostra sul display OLED un messaggio periodico che esorta all'assunzione del farmaco. Com'è facile immaginare abbiamo implementato una prima versione di sistema, che tuttavia può essere presa in considerazione per evoluzioni fu-

ture a seconda delle necessità, come ad esempio più farmaci nell'arco della giornata a differenti ore, oppure lo stesso farmaco nell'arco della giornata in momenti differenti e via dicendo.

Per una maggiore comprensione del sistema, riportiamo in **Fig. 1** un diagramma a blocchi nel quale è immediatamente visibile il micro:bit al centro di tutto il sistema, che gestisce prevalentemente attuazioni come i LED (Light Emitting Diode), il display OLED e il driver per la movimentazione dei vari box attraverso un motore stepper; infine, mediante Bluetooth comunica con lo smartphone. Da tale schema viene anche messo in risalto la tipologia di comunicazione o protocollo utilizzato per lo scambio di dati tra i vari blocchi, che a breve analizzeremo nel dettaglio mano a mano che affrontiamo le funzionalità delle schede d'interesse.

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

Compresi i macro-aspetti di sistema, passiamo all'analisi delle caratteristiche principali delle schede utilizzate (per semplicità si tralasceranno i LED e il Buzzer) per poi passare ai vari dettagli d'implementazione software e applicazione Android.

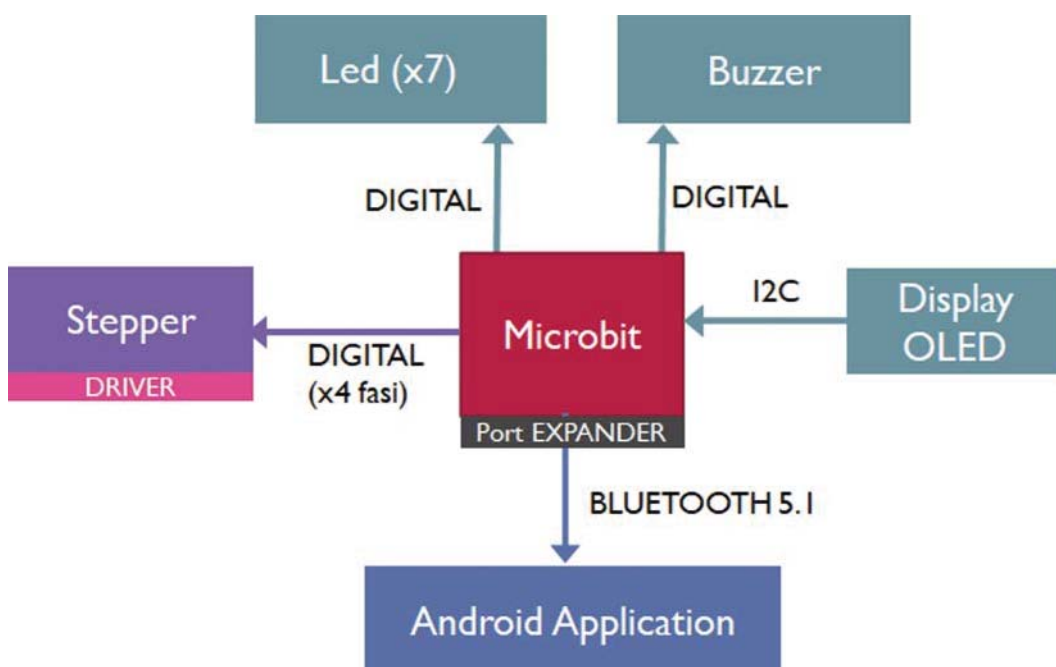


Fig. 1
Schema a blocchi
Distributore di
farmaci.

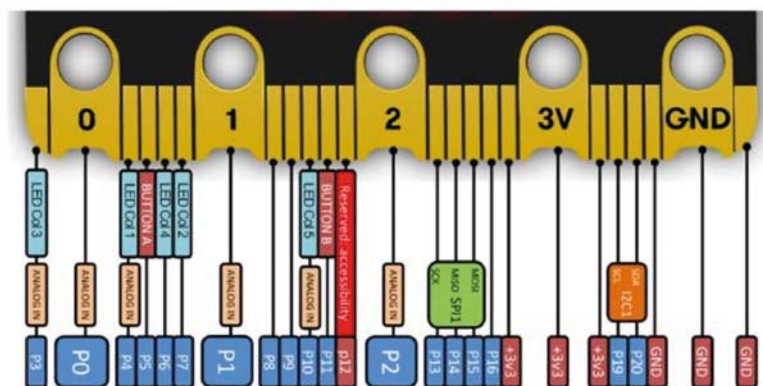
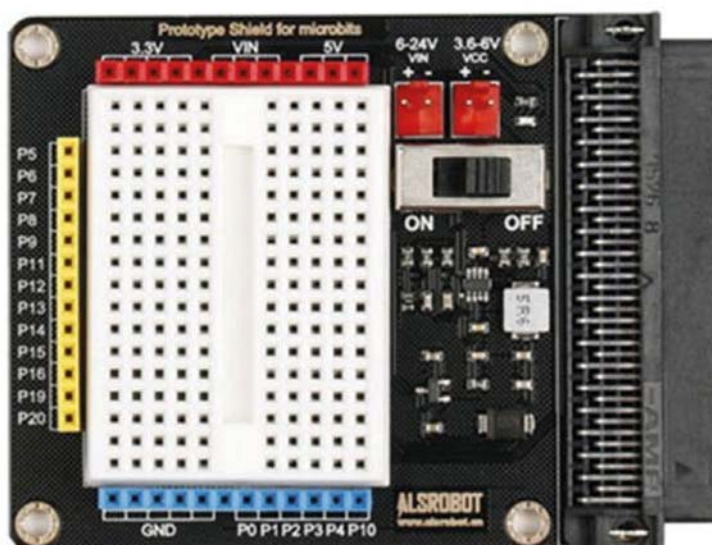
LA SCHEDA D'ESPANSIONE

Per permettere un facile utilizzo del micro:bit e sfruttare a pieno tutte le sue caratteristiche di potenza computazionale, flessibilità sui protocolli di comunicazione grazie all'astrazione hardware delle librerie di programmazione ma anche e soprattutto i pin di input/output, è molto utile l'utilizzo di una scheda d'espansione che permetta una facile interconnessione verso il "mondo esterno". Purtroppo, in contrapposizione con i suoi tanti vantaggi, il connettore è un elemento un po' troppo vincolante rispetto altre schede "Open" tipo Arduino, Raspberry Pi e simili, che facilmente si riescono a connettere con board aggiuntive mediante *pin header* (con passo 2,54mm e singolo pin da 1mm). Pur rimanendo molto compatto (riportato in **Fig. 2** con anche visibili le funzionalità dei vari pin) non è standard e quindi è doveroso parlare di una delle schede d'espansione che agevola le connessioni. Tra le varie soluzioni disponibili, quella più compatta e flessibile è sicuramente un "Port Expander" MICROBITPROTO (acquistabile sullo store futurashop.it) in grado di rendere di facile utilizzo tutte le porte di input/output della board. Come mostrato in **Fig. 2**, è visibile il TOP della scheda MICROBITPROTO e si comprende rapidamente come il connettore del micro:bit possa essere reso usabile/accessibile mediante pin header colorati in vario modo, equivalentemente a quanto accade con Arduino, Raspberry Pi o altro.

Oltre l'immediata possibilità di riportare la connessione in un ambito "standard", ci sono vari vantaggi aggiuntivi, come la possibilità di essere alimentata tramite i due connettori rossi, con tensioni variabili fino a 24V o fino a 6V e poter sfruttare uno dei due regolatori stabilizzati di tensione da 5V o 3.3V; oppure la possibilità della scheda di essere alimentata direttamente tramite la porta USB di micro:bit. Riguardo i pin header con passo 2,54mm, si hanno le seguenti funzionalità:

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

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



collegamenti fisici magari complessi da collegare direttamente al micro:bit.

IL DISPLAY OLED

Come analizzato nel diagramma a blocchi di **Fig. 1**, uno dei componenti "gestiti" da micro:bit è il display OLED (Organic Light-Emitting Diode), ovvero diodo organico ad emissione di luce, che a differenza di un comune display LCD (Liquid Crystal Display), non necessita di una fonte luminosa esterna per essere visibile ma è direttamente in grado di emettere luce propria; ne consegue che possono essere realizzati display dalle ridottissime dimensioni e peso, come in questo caso circa 29 x 27 x 5 (mm) e solo 3 grammi.

Questo, oltre ad altre caratteristiche, lo rende un ottimo sostituto al display LCD.

Può essere acquistato con due risoluzioni/dimensioni differenti, 128x64 e 128x32, angolo di visione circa 180°, dotato di un buon contrasto; naturalmente la risoluzione da 32 avrà la parte

Fig. 2
Port Expander
e pinout.

→ Fig. 3
Display OLED.



del display dimezzata. In più, viene prodotto con differenti colorazioni, sia modelli monocromatici (maggiormente diffusi ed economici), che modelli bicromatici. Questi ultimi, a differenza dei primi che presentano una sola colorazione, hanno una striscia ben definita nella parte alta del display di un colore differente rispetto alla sottostante (i colori generalmente sono giallo e blu). Posso essere utili in svariate applicazioni, come ad esempio nel caso in cui si vogliano differenziare le informazioni da mostrare nella parte alta - si pensi alla data, allo stato di una batteria e via dicendo - per lasciare la parte inferiore ai contenuti della particolare applicazione. Tuttavia, per il nostro scopo abbiamo selezionato un display monocromatico 128x64 di colore blu (cod. OLED096C).

Nella maggior parte dei casi, questi display OLED utilizzano il driver SSD1306 on-board per l'interfacciamento, che mette a disposizione due protocolli di comunicazione possibili, SPI (Serial Peripheral Interface) e I2C (Inter Integrated Circuit), nel nostro caso I2C; con tensione di alimentazione dai 3,3V ai 5V circa. Perciò tolti i pin di alimenta-

↓ Fig. 4
Stepper Driver e motore stepper.



zione e zero volt (GND), si riesce a pilotarlo agevolmente soltanto mediante gli altri due pin dell'I2C, SDA (Serial Data) ed SCL (Serial Clock). Questa scheda, come le altre che vedremo, ha anche dei comodi fori di fissaggio per una facile installazione mediante delle viti.

DRIVER STEPPER E MOTORE

Passando oltre, per la movimentazione dei box e quindi per permettere l'accesso ai farmaci, tra i vari moduli e motori disponibili sul mercato la scelta è ricaduta sul sistema SM2003DB visibile in **Fig. 4**, acquistabile sullo store *futurashop.it*. Questo perché permette una soluzione tutta in uno, ovvero una scheda per la gestione del motore ed in più il motore associato; essendo poi uno stepper (passo-passo) e non un DC, si ha la possibilità di ruotare i vari box dei farmaci in modo preciso e direttamente legato agli step/passi di rotazione impostati. Ma vediamo qualche dettaglio in più. Come appena anticipato, il set acquistabile è composto da una scheda di controllo basata su un driver 7 canali (Darlington NPN) per motori passo-passo mediante il chip ULN2003 e da un motore unipolare stepper di tipo 28BYJ-48. Questa tipologia di motori non sono altro che dei dispositivi elettromagnetici che si muovono con passi discreti ben definiti; internamente vi sono varie bobine con un albero centrale su cui sono agganciati una serie di magneti, le bobine vengono alternativamente eccitate con correnti che generano campi magnetici, i quali attraggono o respingono i magneti sull'albero, facendo ruotare il motore. Dotato di cinque fili e quattro bobine, ha le connessioni centrali di quest'ultime tutte interconnesse tra loro e utilizzate come alimentazione e il nome unipolare deriva proprio dal fatto che la potenza arriva sempre su questo polo. Considerando la scheda, l'unico integrato presente è l'ULN2003A dedicato alla gestione del motore (per maggiori dettagli si faccia riferimento al datasheet), poi troviamo 4 diodi LED con le corrispondenti resistenze di polarizzazione, che corrispondono alle quattro fasi (A - B - C - D) del motore passo-passo e vari connettori. Per poter utilizzare lo stepper, le quattro fasi di pilotaggio verso la scheda devono essere collegate ai pin-header J1; mentre attraverso il connettore J4 a 5 poli si deve collegare il motore alla scheda facendo accortezza alla polarità di connessione. Infine, il pin-header J3 serve per l'alimentazione, ovvero, se si vuole alimentare il motore a 5V si dovrà mon-

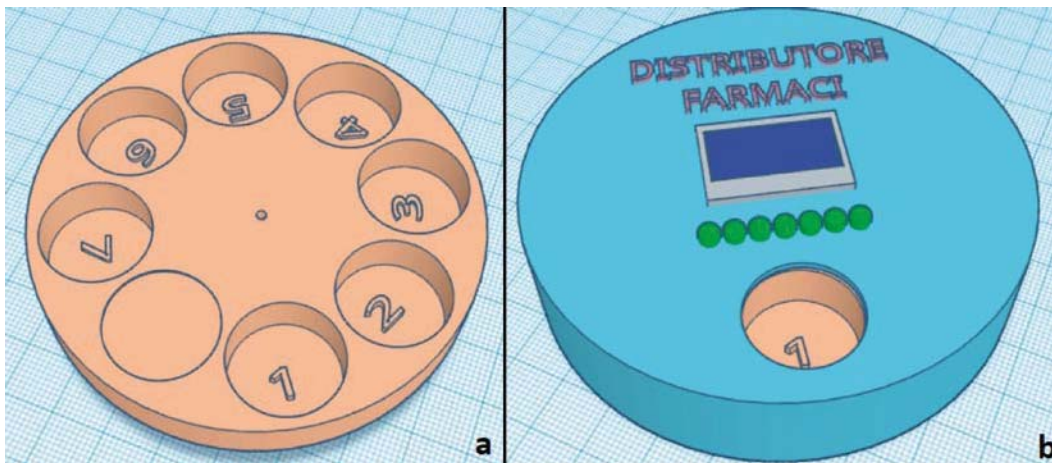


Fig. 5
Una possibile
implementazione
meccanica

tare il jumper affinché la stessa tensione arrivi all'integrato U1 e al motore. Dato che un motore di questo tipo ha un assorbimento rilevante, potrebbe essere preferibile fornire l'alimentazione da una scheda apposita e non dalla scheda controllo. Si tenga presente che questa tipologia di motori può anche essere acquistata singolarmente in due versioni, a 5V e a 12V, a seconda della necessità; forniscono entrambe una buona coppia anche in stato di stop ma hanno un assorbimento importante da tener presente. In ogni caso, acquistando il set direttamente da futurashop.it si ha tutto il necessario per il progetto.

LA MECCANICA

Facciamo una breve trattazione dal punto di vista della meccanica per illustrare quale potrebbe essere una possibile implementazione pratica, sulla base della quale procedere con tutte le prossime considerazioni. Nella **Fig. 5a** viene riportata un'idea di come potrebbero essere gestiti i box o scompartimenti all'interno dei quali inserire i farmaci; come si può vedere, sono stati inseriti dei numeri interni per facilitare l'associazione con i giorni della settimana. Il contorno circolare tra l'uno e il sette indica il box pieno o "dummy box" che viene preso come riferimento per la posizione di inattività del distributore. In **Fig. 5b** viene riportato il distributore di farmaci completo, con a vista il display OLED, i 7 LED (in figura di colore verde) e l'unico foro circolare accessibile a seconda dell'erogazione d'interesse (in figura viene effettuata l'erogazione del lunedì).

DESCRIZIONE SETUP

Prima di passare all'implementazione software su micro:bit e alla presentazione dell'app Android,

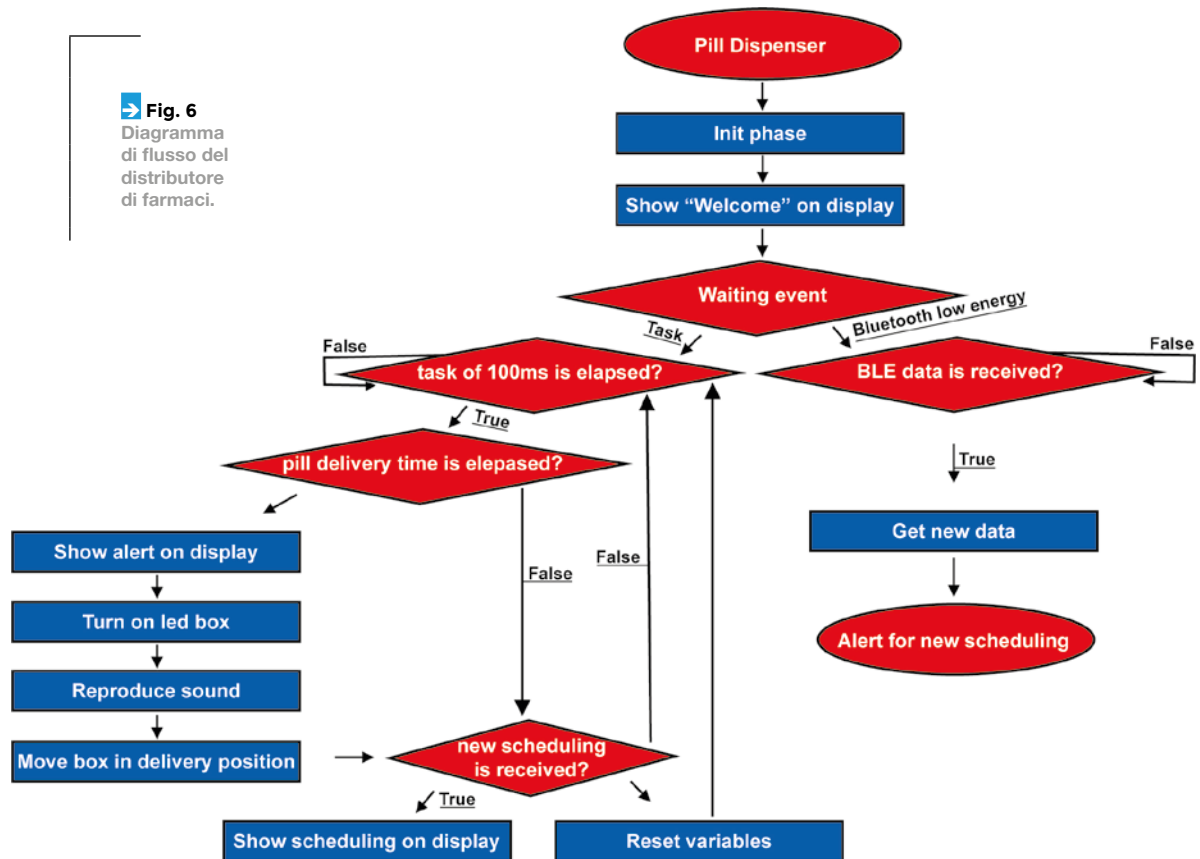
e dopo aver illustrato le caratteristiche di ogni componente del progetto, vediamo brevemente le interconnessioni fisiche presentate a livello logico nel diagramma a blocchi di **Fig. 1** e il setup di sistema tra il micro:bit e le varie periferiche. Ripartiamo dal considerare il *port expander*, con riferimento all'immagine di **Fig. 2**, abbiamo scelto di utilizzare la tensione di alimentazione V1, il riferimento G ed i connettori gialli per le periferiche. Per quanto riguarda le tensioni di alimentazione è importante impostare la linea V1 su 5V, collegando i due jumper verdi come mostrato in figura. Per semplicità i pin d'interesse destinati alle periferiche vengono riportati in **Tabella 1** ed esplicitate le varie connessioni tra schede attraverso i loro pin-header, ma vediamoli insieme.

Le connessioni principali sono quelle che vanno da micro:bit al display OLED e alla scheda del motore passo-passo. Il primo collegamento prevede alimentazione, zero volt (GND) e protocollo di comunicazione I2C; mentre la seconda connessione oltre alle alimentazioni ha le quattro fasi per il pilotaggio dello stepper, perciò:

- **Vin e GND:** che rappresentano il positivo di alimentazione e massa
- **SCK e SDA:** necessari per la comunicazione (particolare attenzione per SDA che dovrà essere collegato in pull-up con una resistenza di 1KΩ)
- **Phase 1, 2, 3, 4:** solo le quattro fasi per la gestione motore
- **3V:** non collegato

Infine, per quanto riguarda il collegamento tra micro:bit e LED, basta prevedere una resistenza da 1kΩ tra essi su ogni connessione per avere una corretta gestione; mentre il buzzer è collegabile direttamente.

Fig. 6
Diagramma
di flusso del
distributore
di farmaci.



Per i meno esperti, come vedremo successivamente, le librerie software utilizzate ci astrarranno da tutti questi dettagli e impostazioni, permettendoci di semplificare enormemente lo sviluppo software e rendendo la sua configurazione immediata, illustrando le funzioni più importanti per implementare la lettura dei dati d'interesse.

SVILUPPO SOFTWARE

Ora che il setup del sistema è completo, possiamo procedere con l'attività di sviluppo software così da "animarlo". Una considerazione è doverosa prima di procedere oltre, ovvero, per permettere un corretto funzionamento di tutto quello che verrà implementato a breve, bisogna avere accortezza nel montaggio iniziale dell'alberino (o mozzo) dello stepper rispetto l'alloggiamento dei box dedicati ai farmaci, questo perché il software parte dal presupposto che l'apertura di accesso del sistema (per prelevare i farmaci) sia centrata in una fase iniziale con il "box chiuso" (identificato in seguito come dummy), cioè quello che non permette l'inserimento o il prelievo di nessun farmaco. Tutto ciò garantisce un ritorno in una posizione

Tabella 1

Connessioni fisiche tra i pin header delle varie schede.

PIN	FUNZIONI
P0	Led Box 1
P1	Led Box 2
P2	Led Box 3
P3	Led Box 4
P4	Led Box 5
P6	Led Box 6
P7	Led Box 7
P8	Buzzer
P13	Phase 1 Motor
P14	Phase 2 Motor
P15	Phase 3 Motor
P16	Phase 4 Motor
P19	I2C SCL Display
P20	I2C SDA Display



dummy quando il sistema non deve erogare farmaci e tutela l'utente evitando un accesso continuo ai box.

Il caricamento dei farmaci è stato pensato come manuale, ovvero dovrà essere cura dell'utente/utilizzatore inserire il farmaco in uno dei sette box mostrati in **Fig. 5** e successivamente tramite l'applicazione Android (descritta di seguito), definire le pianificazioni in base al box associato al particolare farmaco.

DIAGRAMMA DI FLUSSO

Prima di addentrarci nello sviluppo del codice è importante schematizzare le logiche di funzionamento del sistema completo di erogazione farmaci e lo facciamo tramite un diagramma di flusso mostrato in **Fig. 6**, che consentirà di definire ad alto livello le macro funzioni implementate.

Il primo blocco è rappresentato dalla fase di inizializzazione delle variabili e dei registri che riguardano i componenti display OLED, led, motore stepper e tutte le variabili necessarie per lo sviluppo.

L'algoritmo dovrà operare in base ad eventi generati dallo scambio dati via Bluetooth ed un timer che periodicamente (ogni 100 millisecondi) aggiornerà lo stato delle periferiche.

Per la funzione Bluetooth è necessario valutare se il dispositivo è accoppiato o meno; nell'eventualità attendere la ricezione di nuovi dati così da acquisire le informazioni relative alle nuove pianificazioni trasmesse dall'applicazione Android tramite uno smartphone ed aggiornare il flag rappresentativo di "nuovo dato ricevuto".

Il secondo evento viene generato ogni 100 millisecondi (successivamente anche indicati come *ms*) con lo scopo di analizzare i "nuovi dati ricevuti" via bluetooth e quindi gestire tutti i componenti coinvolti nel sistema. Alla scadenza del timer di 100ms, il primo controllo verifica lo stato della pillola in programmazione, qualora si avesse corrispondenza con la data di erogazione allora verranno aggiornate tutte le informazioni da mostrare su display, verrà riprodotto un suono, verrà acceso il led corrispondenza al box del farmaco e il motore stepper avrà il compito di ruotare il box nella posizione di consegna.

Nel caso in cui la pianificazione della pillola non venisse aggiornata, l'algoritmo verifica la ricezione di un nuovo dato Bluetooth così da mostrare a video la prossima data di assunzione farmaco.

Se nessuna delle condizioni precedenti viene rispettata allora si attende semplicemente il successivo evento di timer per analizzare i contenuti.

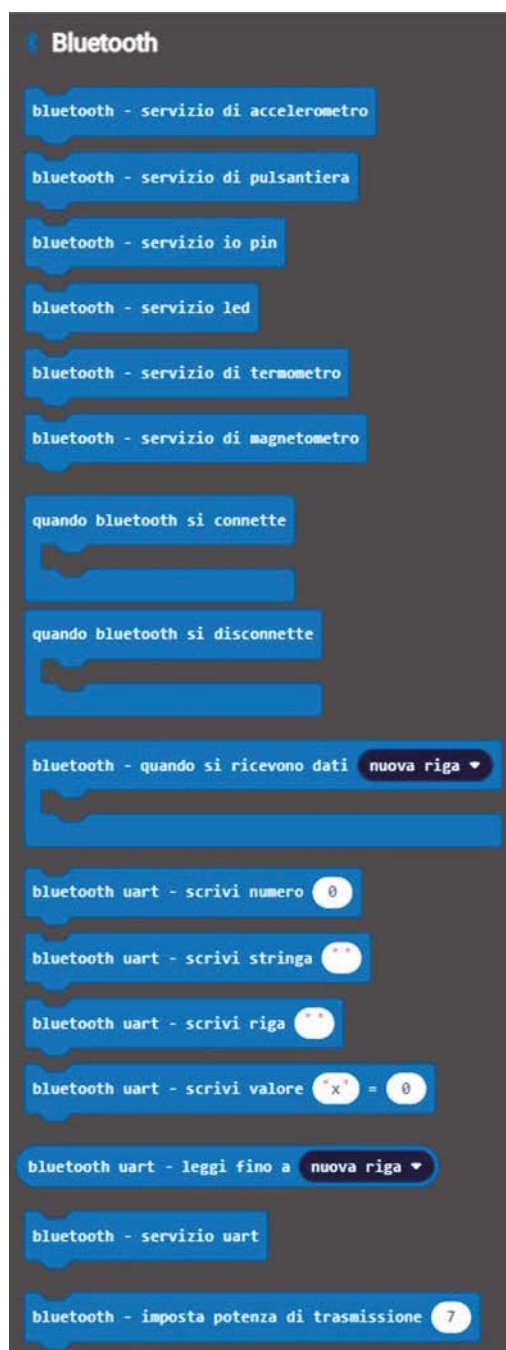


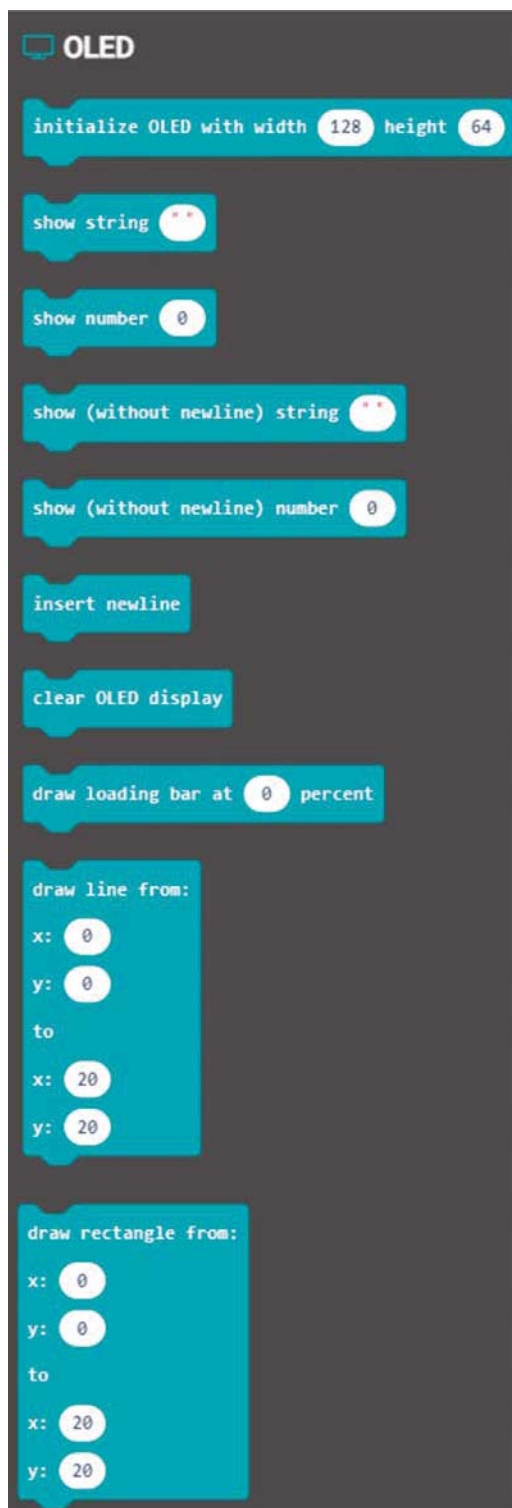
Fig. 7
Libreria
Bluetooth.

Sulla base del diagramma di flusso definito in **Fig. 6**, possiamo implementare il codice utilizzando la piattaforma di sviluppo *Make Code Block* e definire le librerie necessarie per la gestione dei componenti coinvolti.

DESCRIZIONE LIBRERIE

Le librerie che andremo ad importare sono relative al motore stepper, al Display OLED e al protocollo

Fig. 8
Libreria Display
OLED.



Bluetooth. Per aggiungerle all'interno del nostro progetto è necessario selezionare "Estensioni" e digitare le parole chiave "Bluetooth" e "ssd1306" per il display OLED, facendo attenzione che la versione sia la 2.0.9 tramite il tasto approfondisci e "stepper-motor" per il motore. La libreria Bluetooth presenta innumerevoli fun-

zioni ma per il nostro progetto utilizzeremo solo quanto segue:

- *Quando bluetooth si connette* – per verificare quando il dispositivo è connesso
- *Quando bluetooth si disconnette* – per verificare quando il dispositivo è disconnesso
- *Quando si ricevono dati nuova riga* – per leggere i dati ricevuti in base al carattere iniziale.
- *Bluetooth uart – leggi fino a nuova riga* – per leggere i dati ricevuti via bluetooth fino al carattere definito.
- *Bluetooth – servizio uart* – per inizializzare la periferica bluetooth

La libreria OLED verrà utilizzata per gestire il display, di cui utilizzeremo:

- *Initialize OLED with width 128 height 64* – per inizializzare il display con riferimento alla risoluzione
- *Clear OLED display* – per ripulire il contenuto del display
- *Show String* – per scrivere una stringa su display
- *Insert newline* – per aggiungere una linea vuota su display

Ed infine la libreria stepper rende disponibili le seguenti funzioni:

- *Imposta Motor* – per associare le quattro fasi ai pin del micro:bit
- *Move Motor 0 steps anti-clockwise* – per movimentare il motore in senso anti-orario per un numero di steps definiti
- *Move Motor 0 steps clockwise* – per movimentare il motore in senso orario per un numero di steps definiti
- *Motor set delay between steps to 0 ms* – consente di cambiare la velocità di rotazione in base al delay tra i vari steps.

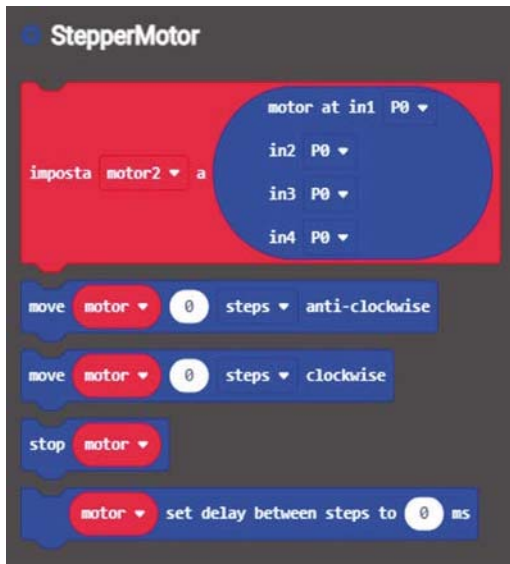
IMPLEMENTAZIONE SOFTWARE

Dopo la breve descrizione delle librerie passiamo allo sviluppo del codice tramite la piattaforma Make Code Bock. Il codice verrà reso disponibile in formato compilato (con estensione ".hex") direttamente su sito elettronica.in.it, pertanto descriviamo unicamente le funzioni principali in linea con il diagramma di flusso presentato in **Fig. 6**.

Il primo blocco riguarda l'inizializzazione delle variabili e delle periferiche come mostrato in

Fig. 10 e quindi:

- Inizializziamo la periferica Bluetooth tramite il blocco *bluetooth – servizio uart*



- Disabilitiamo la matrice led del micro:bit tramite la funzione *abilita LED falso* in quanto avremo necessità di utilizzare un maggior numero di pin per i LED associati ad ogni box come definito nella **Tabella 1**
- *Initialize OLED with width 128 height 64* inizializza il display che abbiamo deciso di utilizzare con risoluzione 128x64
- La variabile *data_received* che rappresenta il flag utilizzato per verificare quando un nuovo blocco dati bluetooth è ricevuto
- La variabile *statusIsChanged* verifica quando lo stato della pianificazione viene aggiornata
- La variabile *indexCounterAlarm* è un contatore che consente di temporizzare il tempo di erogazione pillola
- La variabile *limitAlarm* impone il limite temporale rispettivamente per caricare ed avvisare l'utente di prelevare il farmaco
- *displayString* è la stringa che dovrà essere mostrata su display OLED
- *oldString* è la stringa tampone che evita di scrivere le stesse informazioni su display OLED
- *boxNumber*, *timeData* sono le variabili che verranno utilizzate per riconoscere il box e la data di erogazione del farmaco in base al dato ricevuto via bluetooth
- *Motor* è l'inizializzazione dello stepper in base il pinout definito in **Tabella 1**
- *Chiamata initLed* consente di inizializzare il segnale di PWM per ogni LED associato ad un box
- *Chiamata showDataDisplay* consente di mostrare le informazioni su display OLED (in questo caso la stringa "Hello from Distributore di farmaci!")

A questo punto introduciamo l'implementazio-

ne delle logiche relative ad acquisizione dei dati Bluetooth e gestione delle periferiche in corrispondenza di una nuova programmazione oppure erogazione del farmaco; tuttavia in questa fase è opportuno descrivere come verranno impacchettati i dati bluetooth trasmessi, interpretati poi dal micro:bit.

Il *master* è identificato da un smartphone (ma approfondiremo meglio nel capitolo successivo), al momento di una nuova pianificazione invierà delle stringhe con il seguente formato:

#AgendasX;

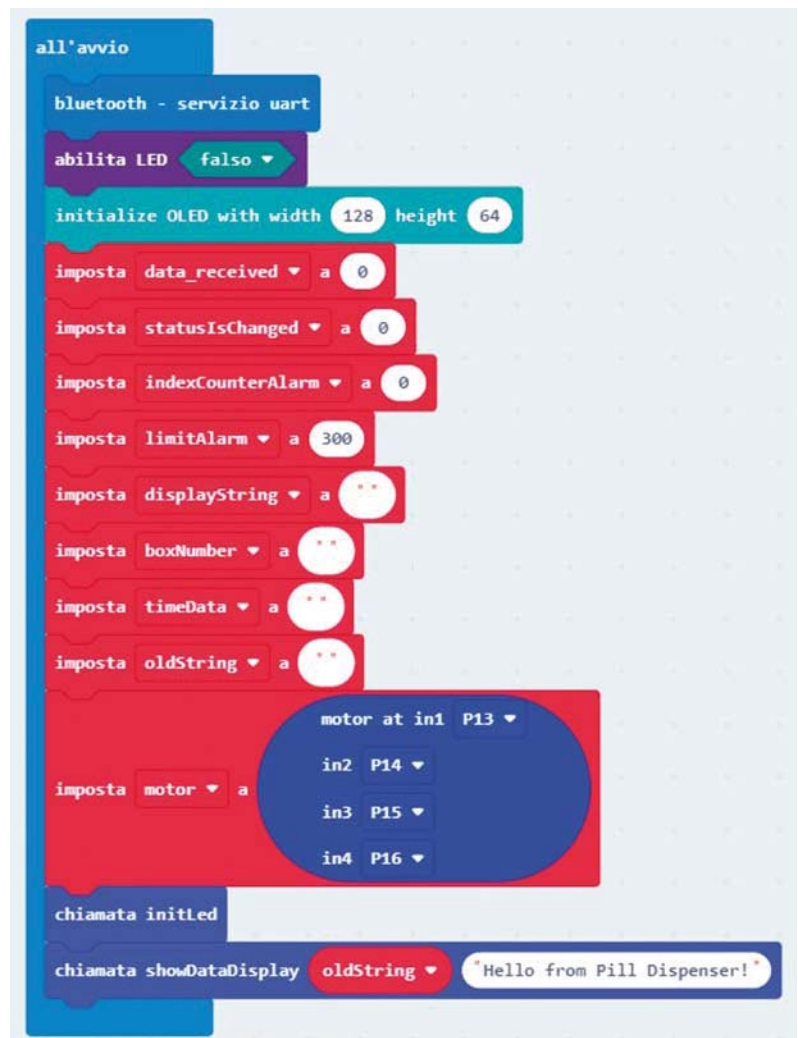
dove *X* rappresenta il numero delle programmazioni, e successivamente la programmazione:

#B1TddmmyyyyhhmmS0;

B indicherà il box assegnato, *T* verrà utilizzata per definire la data di erogazione con formato giorno, mese, anno, ora e minuti ed infine *S* indicherà lo

Fig. 9
Libreria
Motore
Stepper.

Fig. 10
Inizializzazione
Implementazione
Software.



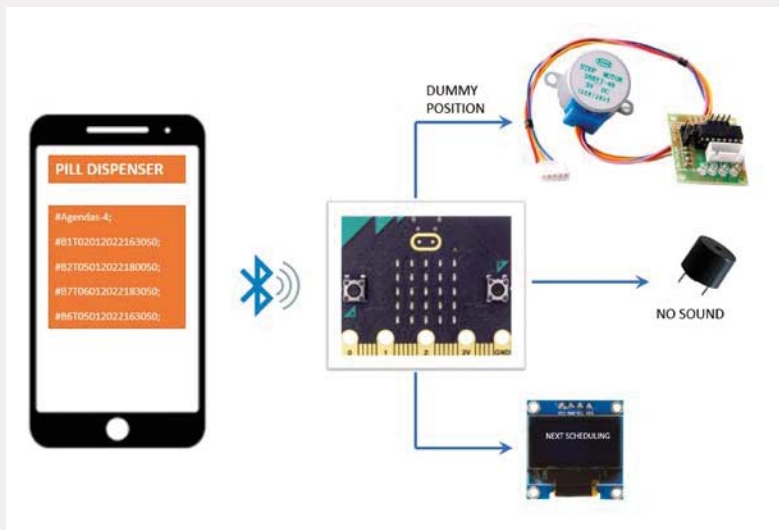


Fig. 11
Scambio
dati App-
micro:bit
standby.

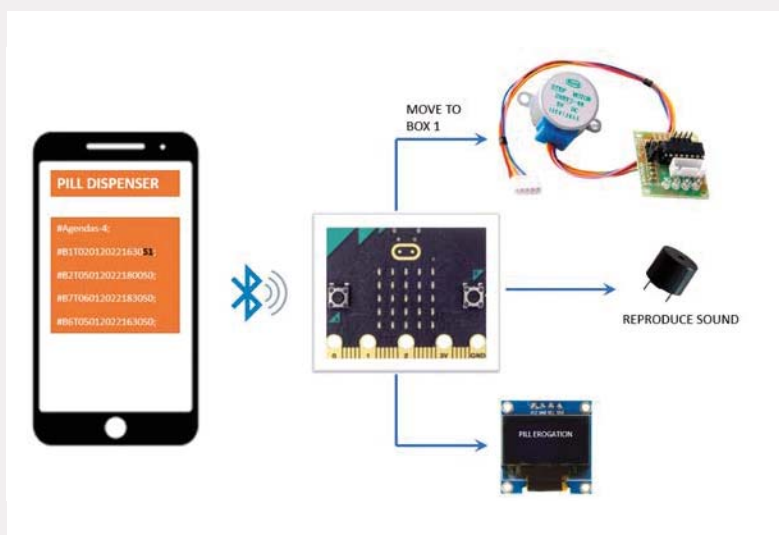


Fig. 12
Scambio
dati App-
micro:bit
erogazione.

stato della programmazione.
Un esempio di programmazioni può essere:

```
#Agendas-4;
#B1T020120221630S1;
#B2T050120221800S0;
#B7T060120221830S0;
#B6T050120221630S0;
```

Quando uno stato di programmazione passerà dallo stato **S0** allo stato **S1** allora il micro:bit dovrà essere in grado di pilotare i carichi per garantire la corretta erogazione del farmaco. Se volessimo schematizzare la comunicazione tra il micro:bit e lo smartphone allora bisognerà tenere in considerazione il flusso dati di **Fig. 11**.

Com'è facile intuire, in una fase iniziale lo smartphone fornisce tutte le informazioni relative alle pianificazioni decise dall'utente ed in base

a quest'ultime il micro:bit avrà solo il compito di informare l'utente su quella che sarà la successiva erogazione. A questo punto rimane in attesa di nuovi aggiornamenti da parte dell'applicazione Android.

La **Fig. 12** mostra come lo stato della pianificazione cambi da S0 a S1 con conseguente trasmissione dati al micro:bit che dovrà gestire i carichi e quindi movimentare il box verso la posizione di erogazione, riprodurre il suono per avvisare l'utente, abilitare il led corrispondente al box ed infine mostrare a video le informazioni sulla pillola. Considerando il flusso dati definito, implementiamo la porzione di codice in **Fig. 13**. Attendiamo la ricezione del carattere '#' e leggiamo il dato fino al carattere ';' così da definire il contenuto della variabile *stringDataBle*. Associamo al vettore *agendaList* il dato ripulito dei caratteri non necessari attraverso la funzione *getAgendaData*. Considerando la sequenza dati sopra definita sappiamo che la prima stringa ricevuta conterrà anche il numero di programmazioni, pertanto andiamo ad estrapolare il valore con la funzione *getNumAgendas*. Se questo valore risulterà maggiore di zero allora proseguiamo con l'analisi altrimenti il vettore *agendaList* verrà re-inizializzato e si rimarrà in attesa del successivo pacchetto dati. Qualora il numero delle programmazioni fosse uguale alla lunghezza dell'array *agendaList* meno 1, allora l'acquisizione delle programmazioni potrà ritenersi conclusa ed il flag di *data_received* impostato ad 1; il primo elemento del vettore *agendaList* rappresentato da '#AgendasX;' verrà rimosso in quanto non più utile per la successiva analisi.

A questo punto sono presenti una o più pianificazioni e quindi possiamo procedere con la gestione delle periferiche in linea con il ramo di destra del diagramma di flusso di **Fig. 6**.

Per assolvere tale scopo, utilizziamo la funzione nativa di micro:bit *every 100 ms* che consente di temporizzare il nostro programma con un periodo di chiamata di 100 millisecondi.

Per semplicità decidiamo di sezionare l'immagine con i due principali eventi ma ricordiamo che il codice deve essere definito all'interno del blocco *every 100 ms*.

Con questa implementazione andiamo ad analizzare gli eventi di:

- un nuovo dato ricevuto via bluetooth analizzato in precedenza con la variabile *data_received*
 - cambiamento di stato della programmazione rappresentato dalla variabile *statusIsChanged*
- Decidiamo di implementare inizialmente la sezio-

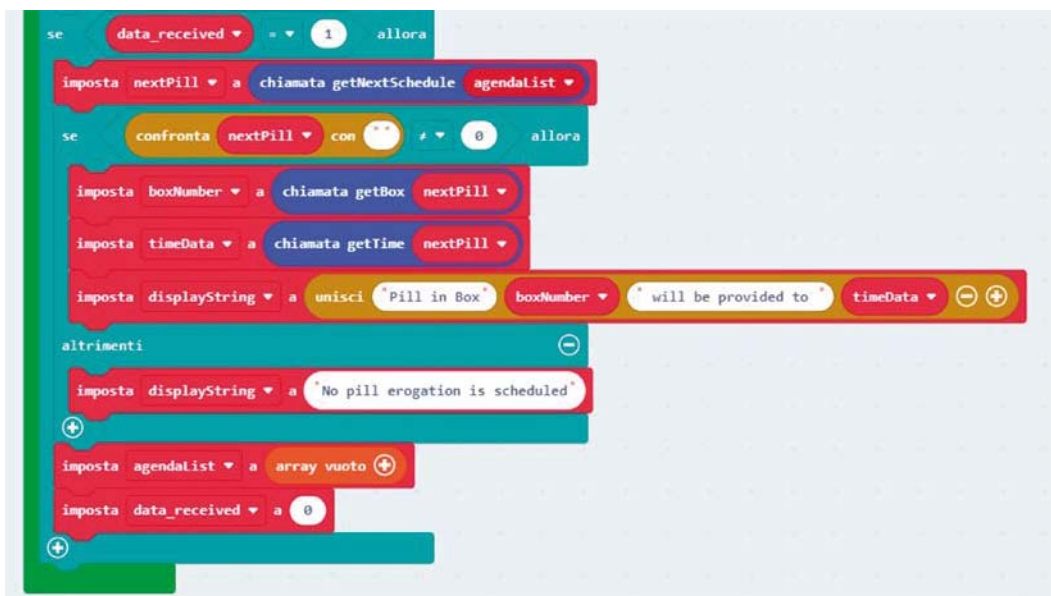


Fig. 13
Implementazione
acquisizione
dati BLE



Fig. 14
Gestione
Periferiche
Erogazione
Farmaco.

Fig. 15
Gestione
Periferiche
Ricezione Dato BLE



ne relativa ad un nuovo data bluetooth ricevuto, visibile in **Fig. 15**, all'interno del quale definiamo la variabile *nextPill* che rappresenterà la successiva programmazione tramite la funzione *getNextSchedule*.

Verificato che il contenuto di *nextPill* sia non vuoto, allora estrapiamo le informazioni relative al numero del box *boxNumber* tramite la funzione *getBox*, alla data di erogazione *timeData* tramite la funzione *getTime* ed infine prepariamo la stringa da mostrare su display OLED.

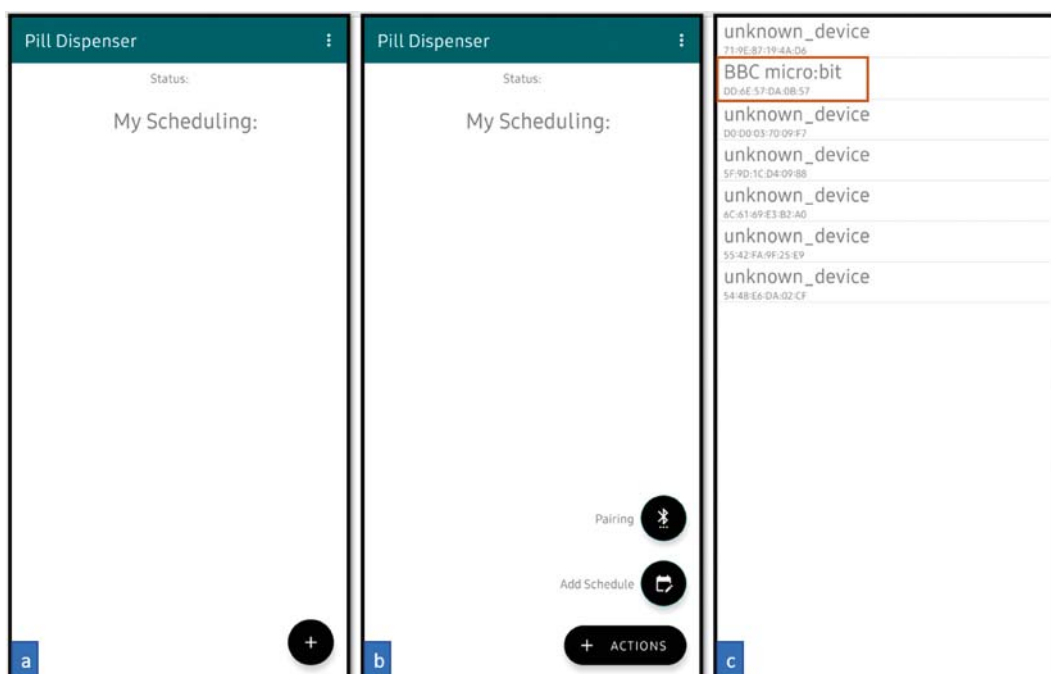
Al termine re-inizializziamo il vettore *agendaList* e

il flag *data_received*.

Terminata questa fase, il sistema rimarrà in attesa del cambiamento di stato della programmazione, pertanto spostiamo la nostra attenzione sull'implementazione del codice relativo alla gestione dei carichi quando è richiesta l'erogazione del farmaco come mostrato in **Fig. 14**.

A tal fine valutiamo inizialmente se lo stato della programmazione è cambiato tramite la funzione *alarmNextPill* (che accetta le variabili *boxNumber* e *agendaList*) e verifica lo stato del box candidato all'erogazione; quindi mostriamo la data della

Fig. 16
Introduzione
Applicazione
Android.



successiva erogazione tramite la funzione *showdataDisplay*.

Quando lo stato di erogazione associato al box cambia (*statusIsChanged* = 1) allora vengono eseguite le seguenti funzioni:

- *StepperMoving* che posizionerà il corrispondente box nella posizione di delivery
- *turnOnLed* che abiliterà il led corrispondente al box
- *reproduceSound* che tramite buzzer avvertirà l'utente di recuperare la pillola
- *showDataDisplay* verrà mostrato su display un countdown di 30 secondi

Terminati i 30 secondi:

- il contenitore andrà nella posizione di dummy tramite la funzione *StepperDummyPosition*
- verrà interrotto la riproduzione del suono tramite la funzione *stopSound*
- disabilitati tutti i led tramite *turnOffLed*
- re-inizializzate le variabili *statusIsChanged* e *indexCounterAlarm*

Come anticipato in precedenza, in una fase iniziale sarà cura dell'utente il caricamento del box e successivamente l'allineamento con l'applicazione Android in base ai box dedicati ai farmaci e non è stata prevista una fase di caricamento automatico



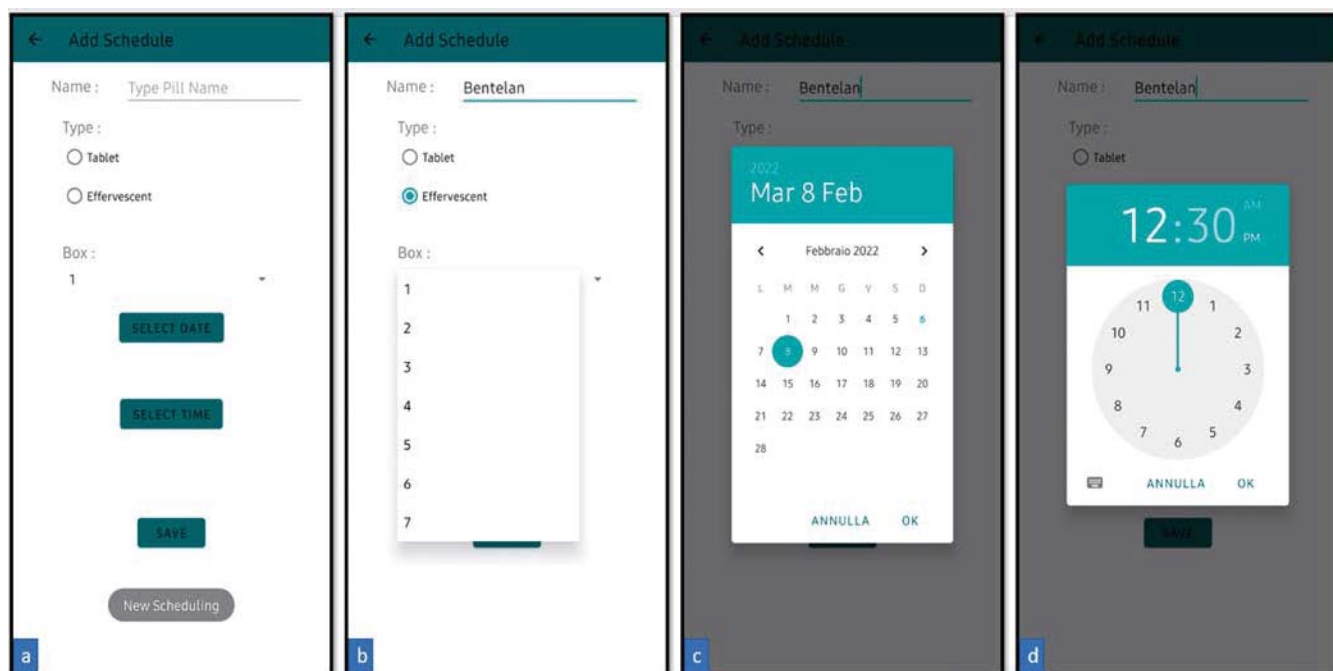
Il nostro prototipo con il case stampato in 3D.

dei farmaci ma potrebbe essere una futura implementazione lasciata come spunto per i più esperti. Infine, evitiamo di analizzare ulteriori funzioni visto che il codice sorgente verrà reso disponibile direttamente on-line per chi volesse approfondire meglio il suo funzionamento.

APPLICAZIONE ANDROID

Terminata l'implementazione del codice lato micro:bit, valutiamo il progetto in tutta la sua interezza descrivendo in questa sezione l'applicazione Android che avrà il compito di svolgere la funzione

Fig. 17 Creazione Pianificazione Applicazione Android.



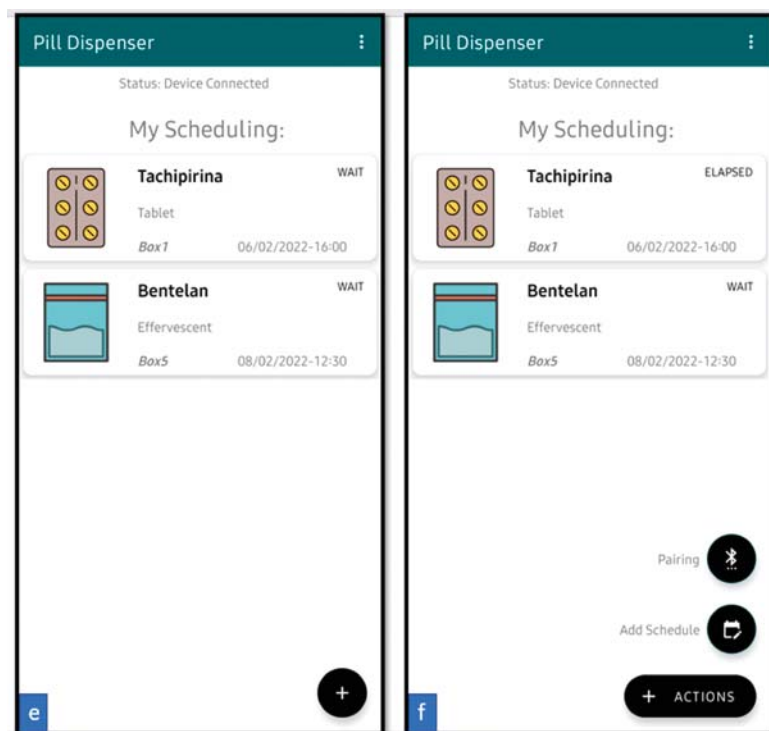


Fig. 18
Gestione
Schede di
Pianificazione
Applicazione
Android.

di master e trasmettere tutte le informazioni relative alle pianificazioni. L'aspetto "estetico" dell'applicazione viene mostrato in **Fig. 16a**, mediante il tasto **ACTIONS** (**Fig. 16b**) è possibile aggiungere una nuova pianificazione *Add Schedule* oppure accoppiare il dispositivo con il micro:bit mediante la *Pairing*. La pressione del tasto di *Pairing* consente l'accoppiamento con il micro:bit come mostrato in **Fig. 16c**, con conseguente aggiornamento dello stato di accoppiamento attraverso l'etichetta *Status* ed aggiornamento del display OLED connesso al micro:bit.

A questo punto è possibile procedere con la creazione delle pianificazioni attraverso il tasto *Add Schedule* che porterà l'applicazione su un nuovo layout come mostrato in **Fig. 17a**.

L'utente ha la possibilità di associare un nome alla pillola, sceglierne la tipologia – se effervescente o compressa – associare il box dove è stata caricata ed infine definire la data e l'ora di erogazione. Le **Fig. 17b**, **Fig. 17c** e **Fig. 17d** mostra i vari contenitori che l'applicazione mette a disposizione per l'inserimento dei dati utili alla creazione della pianificazione. Infine la pressione del tasto *Save* raccoglie le informazioni e riporta l'applicazione nella schermata principale come mostrato in **Fig. 18e**.

Ogni scheda rappresenta una pianificazione con tutte le informazioni create dall'utente, che potrà:

- rimuovere tramite l'operazione di swipe verso sinistra;
- aggiungere come visto in precedenza;
- aggiornare automaticamente dall'applicazione quando il farmaco dovrà essere erogato e quindi in corrispondenza dell'allineamento tra la data di sistema e quella inserita dall'utente.

Tutte queste operazioni generano un evento di invio dati bluetooth verso il micro:bit che dovrà gestire i carichi in base alle informazioni ricevute. La **Fig. 18f** mostra come lo stato della pianificazione cambi da *WAIT* ad *ELAPSED* con conseguente trasmissione dati bluetooth verso il micro:bit che abilita il corrispondente LED del box associato, pilota il motore stepper per portare il box nella posizione di delivery ed emette una segnalazione sonora per avvisare l'utente.

CONCLUSIONI

Arriviamo a termine di questo progetto di distributore di farmaci, analizzato a tutti i livelli, hardware e configurazione del micro:bit, scelta delle board, implementazione del software, definizione di logiche di controllo e trasmissione di dati via Bluetooth ed infine alla configurazione dell'app Android dedicata, grazie alla quale si è in grado di effettuare un setup del sistema e il caricamento dei farmaci settimanali. Come anticipato in precedenza può essere considerata come una prima implementazione di un sistema intelligente da evolvere e modificare a seconda delle proprie necessità, espandendolo per la gestione di più farmaci nell'arco della giornata o lo stesso farmaco assunto più volte nella stessa giornata e così via. □

Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili da Futura Elettronica. La board micro:bit V2 è disponibile con lo starterkit (cod. MICROBITQSKIT) a Euro 45,00, lo stepper motor con driver (cod. SM2003DB) è in vendita a Euro 5,50, il display OLED (cod. OLED096C) costa Euro 11,00, la protoboard (cod. MICROBITPROTO) è disponibile a Euro 15,00. 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>

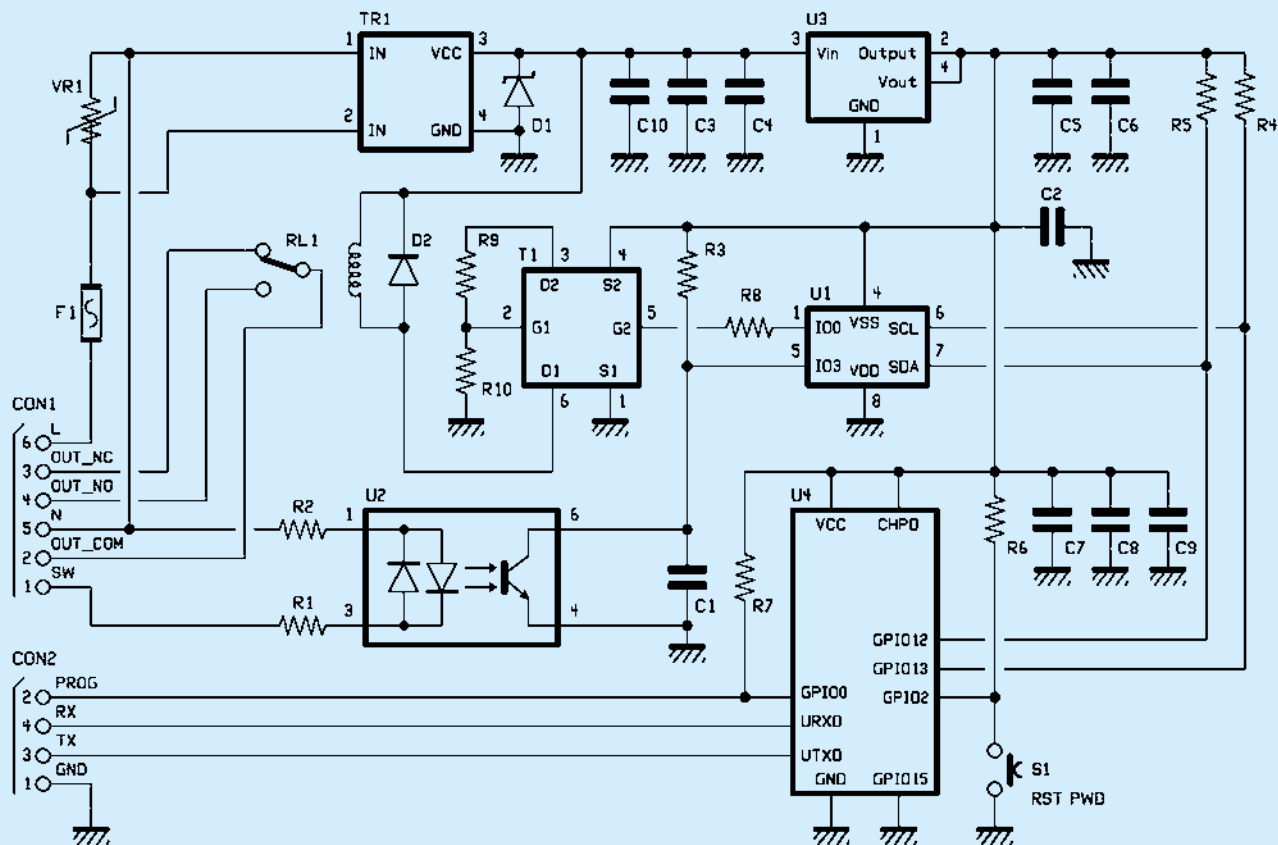
DEVIATORE SMART

di MAURIZIO ŠKERLIČ



Controllato da remoto da smartphone o mediante comandi vocali utilizzando l'assistente vocale Amazon Alexa oppure Google Assistant, permette di simulare la funzione del deviatore per impianti elettrici, ma con controllo WiFi.

Rieccoci assieme con un nuovo modulo domotico da affiancare a quelli già pubblicati: dopo lo Smart Dimmer (fascicolo n° 259) lo Smart switch (fascicolo n° 260), il Double smart switch (fascicolo n° 261) e lo Smart gate lock (fascicolo n° 262) in questo numero vogliamo proporvi il Deviatore smart. Si tratta sostanzialmente del classico deviatore per impianti elettrici, ma implementato attraverso un relé comandato, mediante l'apposita elettronica, tramite comandi ricevuti attraverso il WiFi. Ricordiamo che se un punto luce viene azionato da una sola posizione si utilizza un interruttore, mentre se lo vogliamo comandare da due posizioni dobbiamo sostituire l'interruttore con due



deviatori. Nel caso il numero di posizioni di comando aumentano ulteriormente, tra i due deviatori dobbiamo aggiungere degli invertitori. Quindi, per completare l'assortimento e la scelta dei nostri dispositivi IoT, ci è sembrato doveroso, pubblicare anche il progetto di un Deviatore smart.

Questo dispositivo permette infatti di simulare un classico deviatore per accendere un punto luce da più posizioni, con il vantaggio di poterlo comandare anche con l'assistente vocale, Amazon Alexa o Google assistant, oppure tramite app con lo smartphone o tablet. Come tutti i suoi fratelli, con cui condivide lo stesso software, è un dispositivo compatto con interfaccia WiFi e dotato di un relé elettromeccanico con contatti puliti molto versatile capace di commutare carichi che assorbano fino a 5 A.

SCHEMA ELETTRICO

Incominciamo partendo dal modulo ESP03, siglato U4 basato sul microprocessore Espressif ESP8266 con architettura RISC a 32 bit, clock a 80 MHz ed equipaggiato con una memoria Flash da ben 1MB. Il dispositivo viene ali-

mentato da un alimentatore switching da 5V - 1A, siglato TR1, la cui tensione d'uscita viene abbassata e stabilizzata a 3,3 V dal regolatore di tensione NPC1117, siglato U3. La tensione a 5 V è filtrata dal condensatore C10 mentre quella a 3,3V dai condensatori C8 e C9 indispensabili per evitare reset o crash di sistema casuali dell'ESP8266. I piedini 10 e 11 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 U1. Il piedino 1 di U1 è configurato come porta output mentre il piedino 5 è configurato come porta input. Quest'ultimo è mantenuto a livello logico alto dalla resistenza R3 e il segnale, proveniente dal collettore del transistor presente nel fotoaccoppiatore U2, viene filtrato dal condensatore C1. Il fotoaccoppiatore U2 grazie al doppio diodo LED e le due resistenze R1 e R2, permette di disaccoppiare la tensione alternata a 230 V proveniente dal pulsante collegato al piedino 1 del connettore CON1. Le resistenze R4 e R5 usate come resistenze di pullup mantengono alto rispettivamente le

linee SCL e SDA del segnale I²C. Il pulsante S1 collegato al piedino GPIO2 del modulo ESP03, mantenuto ad alto livello dalla resistenza R6, serve a reimpostare la password predefinita di sistema "123456789", in caso dimenticaste quella memorizzata. Il piedino 1 di U1 pilota un relé elettromeccanico mediante il doppio MOSFET, che nello schema elettrico siglato con T1. Il diodo D2 serve a sopprimere le extratensioni di segno opposto generate dalla bobina quando il relé viene diseccitato e serve ad evitare che la giunzione gate-source del T1 si guasti per breakdown elettrico. Il carico viene collegato tra i piedini 2 (COM), 3 (NC) e 4 (NO) mentre la tensione di rete neutro (N) e fase (L) viene collegata rispettivamente ai piedini 5 e 6 di CON1. Il circuito è protetto da 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. I contatti dello scambio del relé funzionano quindi come quelli di un deviatore manuale.

REALIZZAZIONE PRATICA

Il nostro dispositivo si realizza su un circuito stampato a doppia faccia, con componenti prevalentemente SMD. Per chi fosse interessato a realizzare da solo lo stampato, facciamo presente che sul nostro sito web www.elettronica.in trovate disponibili per il download i file Gerber stampati a doppia faccia ottenibile per fotoincisione (nonché il firmware da caricare nel microcontrollore). Notate che in realtà i PCB sono due, il più piccolo dei quali ospita il modulo ESP03.

Per il montaggio vi occorre un saldatore, preferibilmente a punta fine, oltre a del filo di lega saldante di diametro minimo (0,5 mm dovrebbe andare bene) e della pasta flussante a media densità, a una pinzetta per posizionare i componenti SMD e una lente d'ingrandimento per controllare sia il posizionamento dei componenti, sia la buona riuscita delle saldature e l'assenza, a fine montaggio, di "baffi" di stagno che potrebbero cortocircuitare piazzole che di loro dovrebbero rimanere separate. Ciò detto, iniziate spalmando con un pennelli-

piano di MONTAGGIO

Elenco Componenti:

R1, R2: 200 kohm 300 V (0805)	VR1: V430CH8
R3: 470 kohm (0402)	TR1: RAC05-05
R4, R5: 2,2 kohm (0603)	F1: F 4 A
R6, R7: 10 kohm (0603)	U1: PCA9536TK
R8: 120 ohm (0402)	U2: TLP184
R9: 10 ohm (0402)	U3: NCP1117ST33T3G
R10: 10 kohm (0402)	U4: ESP-03
C1: 1 µF 6,3 V (0603)	RL1: PCH 105D2H-000
C2, C3, C6, C7: 100 nF (0402)	CON1: Morsettiera 6 poli P - 5 mm
C4, C5: 10 µF 6,3 V (0402)	CON2: DIP-1x4 P - 2 mm
C8, C9: 100 µF 6,3 V (1206)	S1: pulsante SMD 4,6x2,8mm
C10: 220 µF 6,3 V (1206)	
D1: SMF5.0A	Varie:
D2: LL4148 (SOD123FL)	-stampato S1596 (41x37,5 mm)
T1: BSD235CH	

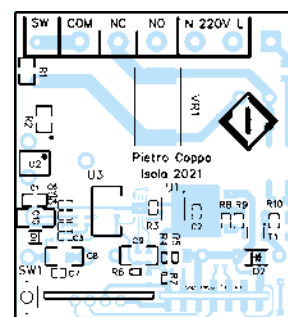
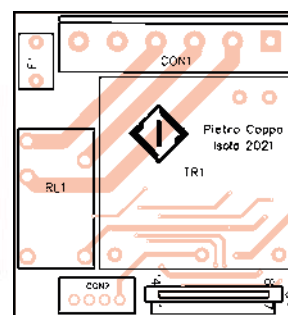


Fig. 1
Montaggio del
modulo ESP03 in
posizione verticale
sul PCB.



no un po' di pasta flussante sulle piazzole del PCB che ospitano i componenti, posizionate questi ultimi rispettando, se prevista, la polarità desumibile dal piano di montaggio visibile in queste pagine, quindi saldate i terminali o pad cercando di tenere la punta del saldatore su ciascuno giusto il tempo di veder sciogliersi e diventare brillante la lega saldante. Procedete con i componenti SMD e poi completate il circuito con le morsettiere, il modulo alimen-

tatore switching AC/DC ed il relé. Su un lato, in corrispondenza delle rispettive piazzole, saldate in verticale il circuito stampato contenente il modulo ESP03, facendo in modo che il modulo stesso sia rivolto all'esterno del PCB, come si vede in **Fig. 1**.

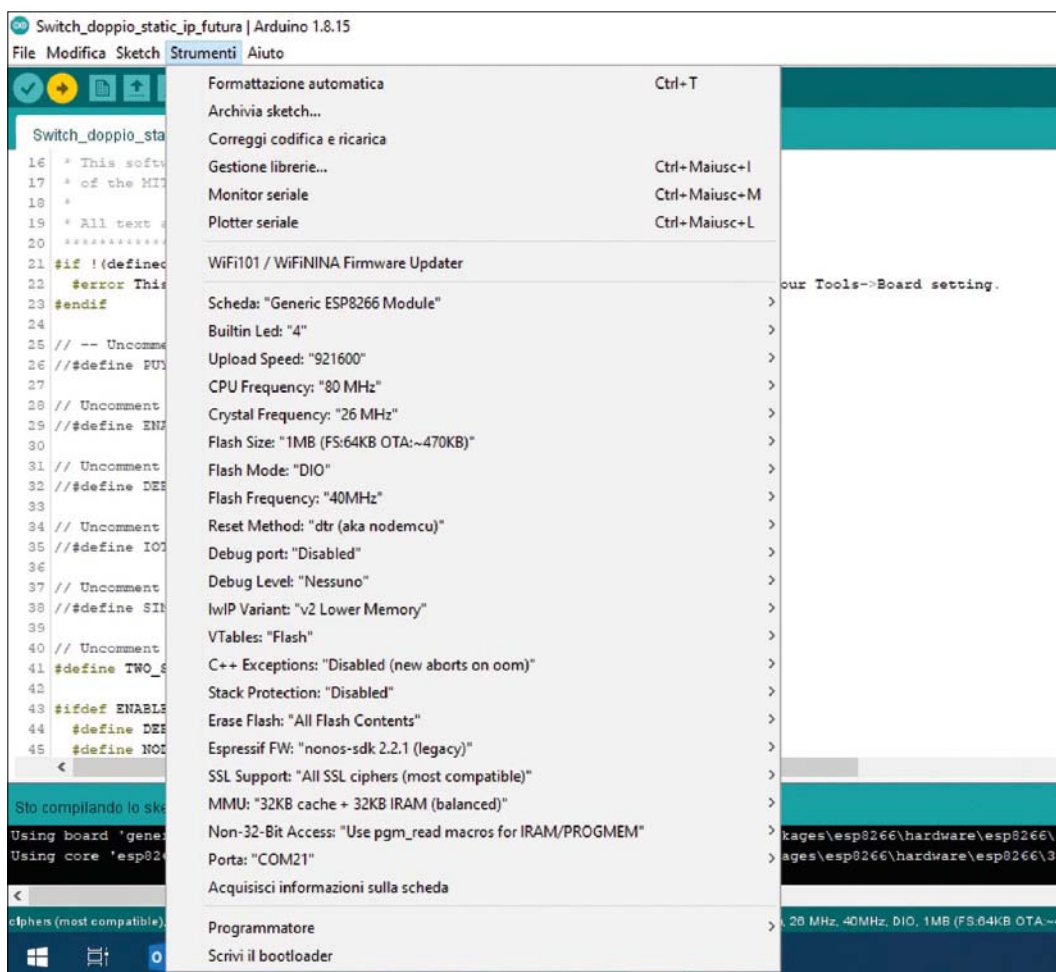
Una volta terminato il montaggio pulite bene il PCB con alcol isopropilico o in alternativa con un altro solvente, in modo da togliere ogni residuo di flussante.

PROGRAMMAZIONE E COLLAUDO

Poiché la configurazione e il collaudo del circuito sono identiche a quelle degli altri moduli IoT come lo Smart Dimmer presentato nel fascicolo n° 259 della nostra rivista, per non essere ripetitivi spiegheremo solo i passaggi più importanti, rimandando a tale progetto eventuali approfondimenti.

Per coloro cui interessa, in **Fig. 2** riportiamo la configurazione di programmazione del modulo

Fig. 2
Impostazioni di
programmazione
del modulo
nell'ambiente di
sviluppo Arduino
IDE.



ESP03 usata nell'ambiente di sviluppo Arduino IDE in fase di progettazione. 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 corretto collegamento del pulsante sul connettore CON2 fate riferimento allo schema di cablaggio proposto nella **Fig. 3**. Ricordatevi di collegare 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 il Deviatore smart in modalità di programmazione premete il pulsante *Prog*, collegato al pin 3 del connettore CON2, prima di allacciarlo alla corrente elettrica, rilasciandolo dopo un secondo.

Lo sketch è stato compilato utilizzando il package core ESP8266 versione 2.7.4 e le librerie "lotwebconf" versione 3.2.0, "SinricPro" versione 2.9.10, "SparkFun_PCA9536_Arduino_Library" versione 1.2.2 e "ArduinoJson" versione 6.18.5. Fate attenzione che la versione del package core dell'ESP8266 all'interno dell'IDE di Arduino sia aggiornato alla 2.7.4, altrimenti potrete riscontrare problemi di funzionamento con la libreria lotWebConf. Al momento della stesura dell'articolo la versione del package core dell'ESP8266 è la v. 3.0.2; ebbene, dobbiamo farvi notare che con questa versione abbiamo riscontrato problemi riguardanti la rete WiFi, che probabilmente con gli aggiornamenti futuri verranno risolti. All'inizio dello sketch alcune righe del programma sono commentate e 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"`. Se volete attivare il debug della libreria Sinricpro togliete il commento davanti a `"#define ENABLE_DEBUG"` se volete visualizzare i commenti dello schetch togliete il commento davanti a `"#define DEBUG_SKETCH"`. La libreria lotwebconf, per impostazione predefinita fa visualizzare sulla seriale le chiavi del vostro account IoT in chiaro. Per motivi di sicurezza vi consigliamo di disabilitare tale opzione aggiungere all'inizio del file *lotWebConfSetitings.h*, che si trova nella cartella *SRC*, la seguente riga di testo `"#define IOTWEBCONF_DEBUG_DISABLED"`.

UTILIZZO DEL DISPOSITIVO

Bene, visto anche questo andiamo a spiegare le impostazioni del Deviatore smart, che si effettuano da browser puntando alla pagina web corrispondente.

Per consentirvi di accedere alla pagina web di configurazione del dispositivo (la vedete nella **Fig. 4**) vi verrà richiesto il nome utente e la password. Come nome utente inserite "admin" e come password "123456789". Facciamo presente che negli smartphone e tablet, una volta scelta la rete SSID, il browser non si avvierà in automatico ma bisognerà cliccare sul pulsante accedi alla rete, oppure immettere nella barra degli indirizzi del browser l'indirizzo IP 192.168.4.1. Ai meno esperti, consigliamo di effettuare questa configurazione mediante PC. Nella pagina iniziale viene visualizzato l'indirizzo fisico ossia l'indirizzo MAC (Media Access

CARATTERISTICHE TECNICHE

- Carico massimo 900 W
- Adatto per carichi resistivi o induttivi
- Configurazione tramite interfaccia web
- Protezione con password
- Reset password tramite pulsante
- Montaggio in scatola da incasso o supporti Vimar, Gewiss, BTicino
- Timer spegnimento automatico
- Controllo locale da pulsante
- Controllo remoto mediante App
- Sincronizzazione stato di apertura/chiusura
- Controllo vocale mediante Amazon Alexa e Google Home
- Collegamento in rete da router WiFi
- Protezione termica e da sovratensioni
- Dimensioni 36 x 43 x 21 mm

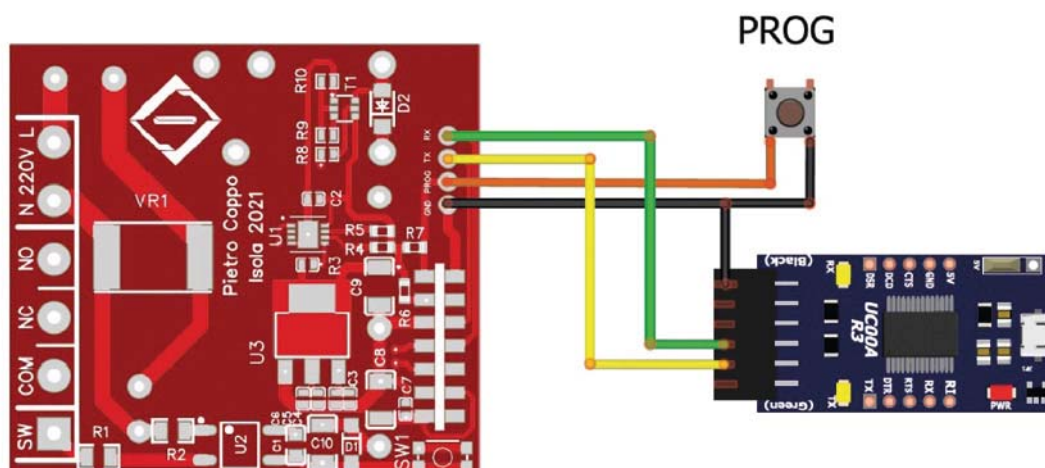


Fig. 3
Collegamento del programmatore seriale al Deviatore smart.

Fig. 4
Home page del
Deviatore smart.

Control) del Deviatore smart. L'indirizzo MAC è un numero a 48 bit composto da 6 numeri esadecimali da 8 bit separati dai due punti. I primi 24 bit dell'indirizzo MAC rappresentano il numero ID del produttore dell'adattatore di rete. I secondi 24 bit rappresentano il numero di serie assegnato all'adattatore di rete dal produttore. Prendete nota di questo numero perché vi servirà in seguito. Cliccando sul pulsante Configuration della home-page accederete alla sezione "System configuration" dalla quale potrete modificare il

nome del dispositivo, modificare la password di sistema, inserire il nome e la password della rete SSID della rete WiFi di casa vostra e lo "Startup delay" (le relative schermate sono proposte nella **Fig. 5**); quest'ultimo è il tempo in secondi per il quale, ad ogni accensione o riavvio, il Deviatore smart diventa un punto di accesso per consentirci di raggiungere la home page. Nella sezione "Device IP" selezionando la casella di spunta "Enable static IP" potete disabilitare l'assegnazione dell'IP dinamico mediante il DHCP server e impostare quello statico. Nella casella "IP address" scrivete il numero IP che volete assegnare al Deviatore smart, nella casella "Subnet mask" inserite l'IP della maschera di sottorete della vostra rete, nella casella "Gateway" inserite l'IP address del router WiFi cui punterete e nella casella "DNS" scrivete il numero IP del server DNS (Domain Name System) del vostro provider oppure l'IP del vostro router. In alternativa potete utilizzare il numero IP di un servizio DNS pubblico come quello di Google, numero primario 8.8.8.8 e numero secondario 8.8.4.4, oppure quello di CloudFlare numero prima-

Fig. 5
Schermata della
pagina web di
configurazione.

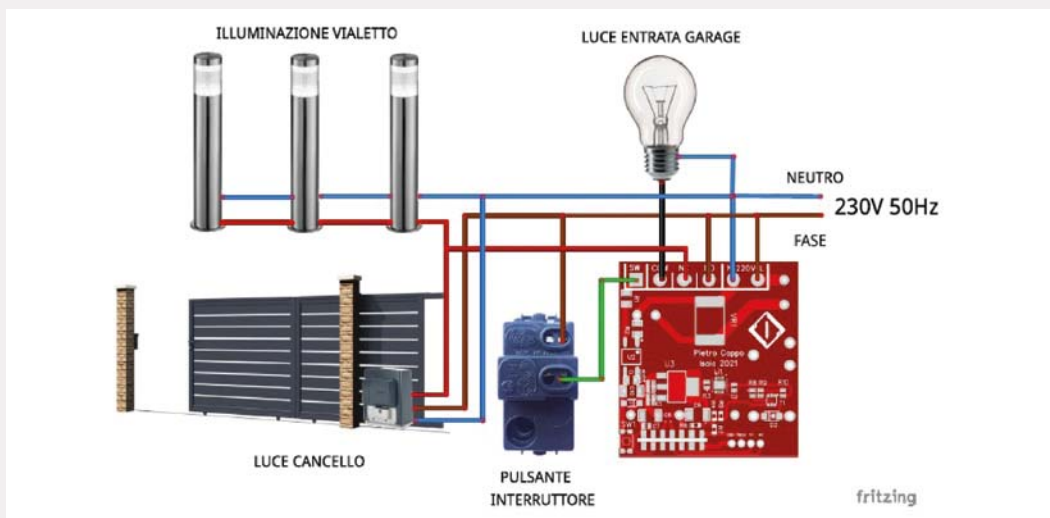


Fig. 6
Schema di cablaggio del Deviatore smart. All'apertura del cancello si accendono tutte le luci mentre con l'interruttore si accende solo la luce all'entrata del garage.

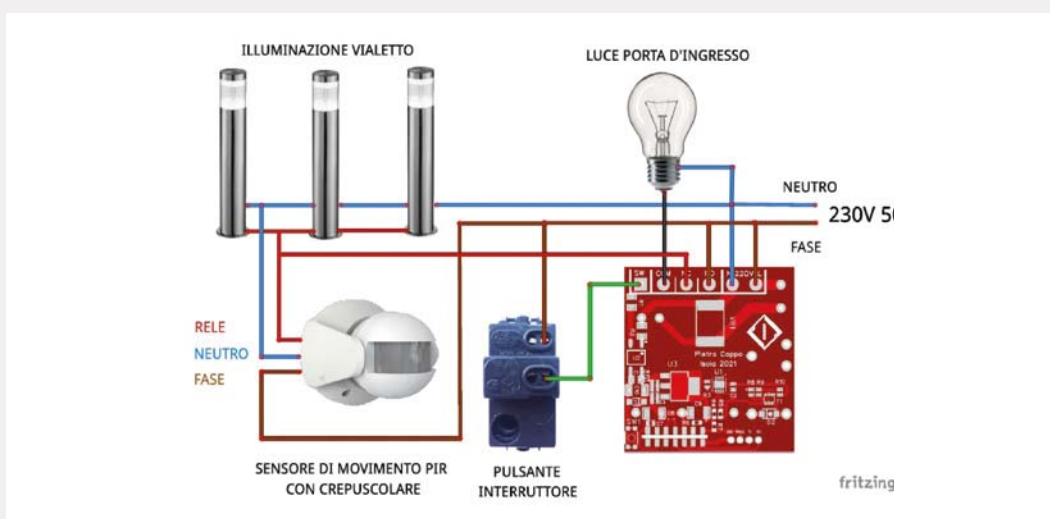


Fig. 7
Schema di cablaggio del Deviatore smart. La luce della porta d'ingresso può essere accesa con l'interruttore oppure mediante il sensore PIR assieme alle luci del vialetto.

rio 1.1.1.1 e numero secondario 1.0.0.1, che tra i due risulta essere pure il servizio DNS pubblico gratuito con un ping medio migliore. Concludiamo le impostazioni con la sezione Settings, che contiene quanto segue. La casella di spunta "Turn on SW1 at startup", permette di attivare l'accensione del carico all'avvio. La casella di spunta "Toggle switch" permette di definire se al Deviatore smart collegherete un interruttore oppure un pulsante. In "Timer" potete impostare il valore in secondi della durata dell'attivazione del relé in uscita. Il valore è impostabile da 0 a 600 secondi che corrispondono a 10 minuti. Se impostate il valore a zero secondi il "Timer" verrà disabilitato e il relé rimarrà attivo fino al comando di disattivazione. La casella di spunta "Enable external pushbut-

ton" vi permette di abilitare o disabilitare la regolazione mediante il pulsante esterno. 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 "Enable web control" vi permette di abilitare il controllo della regolazione mediante pagina web. La casella di spunta "Reboot device on save", abilita il riavvio del dispositivo dopo il salvataggio della configurazione con il tasto "Apply". Terminata la configurazione di sistema salvatela cliccando sul pulsante Apply e verificate che il dispositivo si connetta correttamente alla rete wireless. In **Fig. 6**, **Fig. 7**, **Fig. 8** e **Fig. 9** trovate alcuni schemi esemplificativi di collegamento del Deviatore smart che si ap-

Fig. 8
Schema di cablaggio del Deviatore smart abbinato allo Smart dimmer pubblicato sul fascicolo n° 259. La luce della porta d'ingresso può essere accesa con l'interruttore oppure regolata assieme alle luci del vialetto con lo "Smart dimmer".

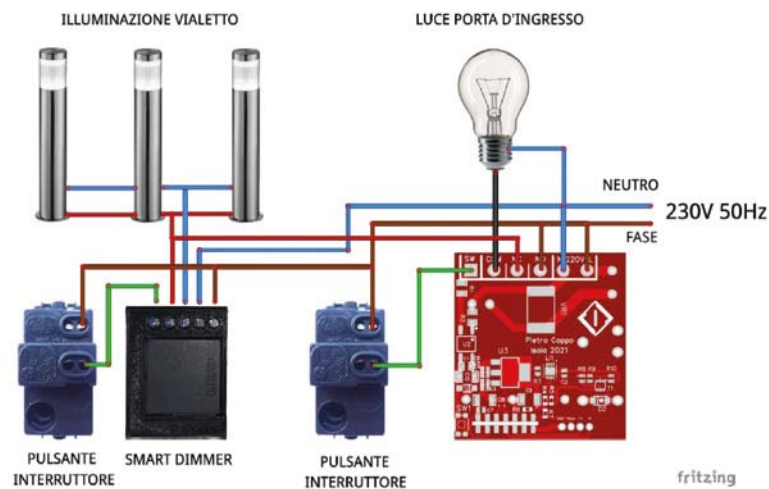
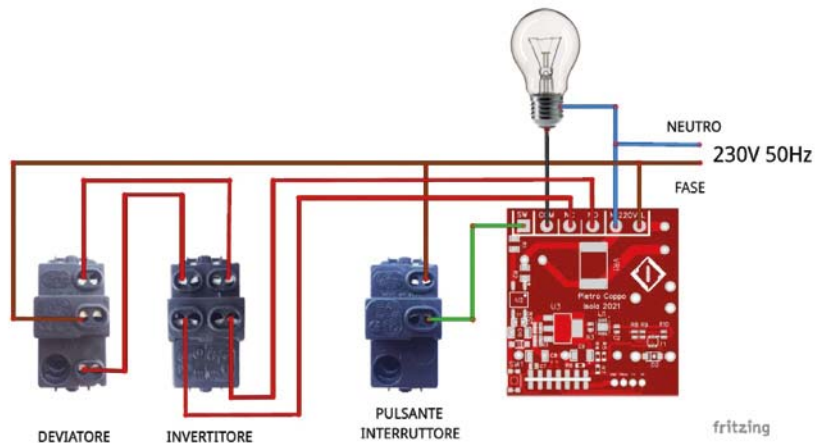


Fig. 9
Schema di cablaggio del Deviatore smart nell'impianto domestico per accendere un punto luce da più posizioni.



Il prototipo visto dalle due facce del PCB: notare l'ESP03 montato in verticale.



plicano alla descrizione dell'utilizzo che stiamo facendo.

IMPOSTAZIONI IN SINRIC PRO

Per funzionare con l'assistente vocale, il Deviatore smart necessita delle credenziali IoT per Sinric pro e per Amazon Alexa. Dal vostro browser, aprite la pagina web <https://sinric.pro> dove vi registrate e fate il login, quindi dal menu dispositivi cliccate sul pulsante "Aggiungi dispositivo": nella sezione "Informazioni sul dispositivo" ricordatevi di selezionare, nel menu a tendina "Tipo di dispositivo" la voce "Switch".

Ricordiamo che quando entrate per la prima volta in Sinric pro prima di aggiungere un dispositivo nel menu "Stanze" dovete aggiungere una nuova stanza, altrimenti il programma non vi permetterà di salvare il nuovo dispositivo (**Fig. 10**). In <https://sinric.pro>, al nostro dispositivo abbiamo assegnato il nome "Luce corridoio".

Dal menu "Credenziali" della pagina web Sinric.pro copiate le chiavi "App key" e "App Secret", mentre dal menu "Dispositivi" copiate l'ID del dispositivo: le chiavi dovete inserirle nella pagina di configurazione del Deviatore smart alla sezione "IoT credential", rispetti-

vamente nei campi "Api Key", "Api Secrets" e "Device Id". A questo punto nel vostro smartphone o tablet dovete installare l'App Amazon Alexa e, una volta entrati, dal menu home installate la skill Sinric Pro. Spostatevi nel menu dispositivi e aggiungetene uno nuovo; nella schermata successiva scegliete come dispositivo "altro" e successivamente cliccate sul tasto "trova dispositivi", quindi attendete la rilevazione del Deviatore smart. Completate la configurazione del vostro sistema seguendo le istruzioni a schermo.

CONTROLLO VOCALE

Se sul vostro smartphone installate l'App Google Home, il Deviatore smart potrà essere comandato anche con l'assistente vocale Google Assistant.

Di seguito riportiamo alcuni comandi riconosciuti dall'assistente vocale. Supponiamo che in <https://sinric.pro> al nostro dispositivo abbiamo assegnato il nome "Luce corridoio".

Per attivare il relé pronunciate:

"Alexa, accendi luce corridoio" oppure "Hey Google, accendi luce corridoio".

Per disattivare il relé pronunciate:

"Alexa, spegni luce corridoio" oppure "Hey Google, spegni luce corridoio".

Fig. 10
Impostazioni del "Nuovo dispositivo" in Sinric Pro. Nella casella "Tipo di dispositivo" scegliete "Switch".





Fig. 11
Contenitori e frutti
stampati in 3D per
placche Vimar,
Gewiss e BTicino.

INTEGRAZIONE NELL'IMPIANTO ELETTRICO

Tra i file scaricabili dal sito web www.elettronica.it, nella cartella STL, trovate i file per la stampa 3D dei contenitori e dei frutti per i supporti Vimar Plana, Gewiss System, Gewiss Playbus e BTicino living, visibile in **Fig. 11**, che faciliteranno l'installazione del Deviatore smart nel vostro impianto elettrico domestico. Fate attenzione al corretto orientamento dei coperchi dei contenitori, in modo che il forellino sia posizionato sopra al pulsante S1, per poterlo premere. 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 0,2 mm e, della versione del software di slicing, attivare l'utilizzo del riempimento a linee nel riempimento a griglia ad alta densità, altrimenti le pareti del contenitore, spesse appena 1,5 mm, non verranno riempite 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 3D, può usufruire del servizio di stampa 3D messo a disposizione sul nostro sito all'indirizzo <https://www.futurashop.it/index.php?route=stampa3d/stampa3d> e chiedere un preventivo.

CONCLUSIONI

Con questo siamo arrivati al termine, sperando di avervi dato tutte le informazioni che vi serviranno a realizzare e installare nel vostro impianto elettrico il Deviatore smart, gestibile e comandabile da remoto con lo smartphone, mediante app e pagina web, oppure con l'assistente vocale Amazon Alexa e Google assistant. Ricordiamo che il progetto Deviatore smart fa parte di una serie di moduli e attuatori domotici progettati e realizzati dagli alunni dell'istituto tecnico Pietro Coppo Isola Slovenia. □



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 PCB (spessore 0,8mm) possono essere realizzati utilizzando il servizio PCBPRODUCTION disponibile sul sito www.futurashop.it. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

Where are you?

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



LOCALIZZATORE GPS/GSM/GPRS CON ASCOLTO AMBIENTALE

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



NEW



cod. FR763

€78,00



cod. TK102

€54,00

MICRO LOCALIZZATORE SATELLITARE GPRS/GSM

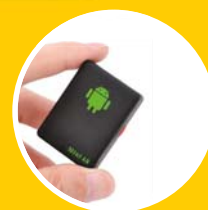
Tutto sotto controllo con il nuovo micro localizzatore GPRS/GSM. Per attivarlo basta inserire una SIM di qualsiasi operatore; inviando un SMS potrai attivare l'ascolto ambientale oppure ricevere un link con le coordinate che indicano la posizione in tempo reale del localizzatore. Dispone di batteria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

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



cod. A8TRACK

€14,00



LOCALIZZATORE TASCABILE GPS/GSM/GPRS

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

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

LIVELLO 2



€ 42,90

Cod. **ARDUKITLEVEL2**

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

LIVELLO 1



€ 32,90

Cod. **ARDUKITLEVEL1**

Disponibili anche....

SET PER ARDUINO UNO REV3

€ 65,00

Cod. **ARDUKITV6**

**IL PIÙ
COMPLETO!**



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

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

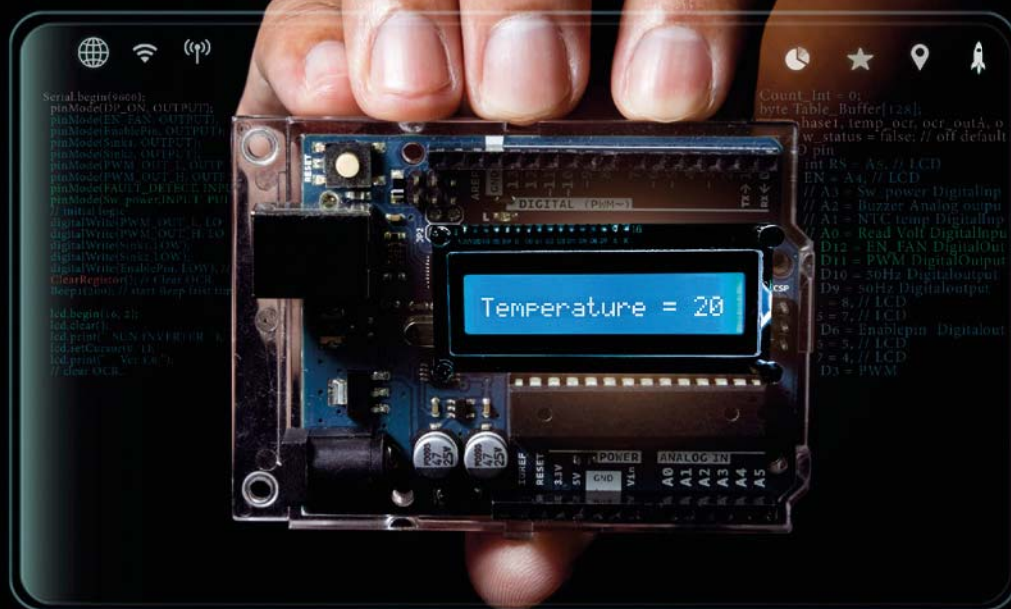
Contiene il necessario per utilizzare Arduino Uno Rev 3; per iniziare basta connettersi a un computer tramite il cavo USB o alimentarla con un trasformatore AC/DC o una batteria.



€ 25,50

Cod. **COMB06G**

Prezzi IVA inclusa



TERMOMETRI CON ARDUINO

dell' Ing. MIRCO SEGATELLO

Misuriamo la temperatura utilizzando come sensori termistori, termoresistenze e termocoppie, per apprendere vantaggi e limiti di queste soluzioni.

Da anni, ai classici misuratori di temperatura analogici basati su sonde di vario genere, amplificatori operazionali, convertitori A/D e display, si preferiscono diverse soluzioni che sfruttano una logica programmabile, ovvero, nello specifico, microcontrollori opportunamente programmati con routine firmware che implementano l'acquisizione, l'eventuale condizionamento del segnale e la sua disponibilità per l'elaborazione da parte di dispositivi terzi o visualizzazione locale. In questo articolo ve ne daremo un esempio, perché tratteremo la misura di temperature con microcontrollori prendendo a riferimento la popolare scheda Arduino, ma lo stesso ragionamento può essere applicato a schede basate, per esempio, su moduli ESP, che offrono la possibilità di interfacciare via WiFi il sistema al mondo del web, ad esempio per caricare i dati su un sito. Prima di

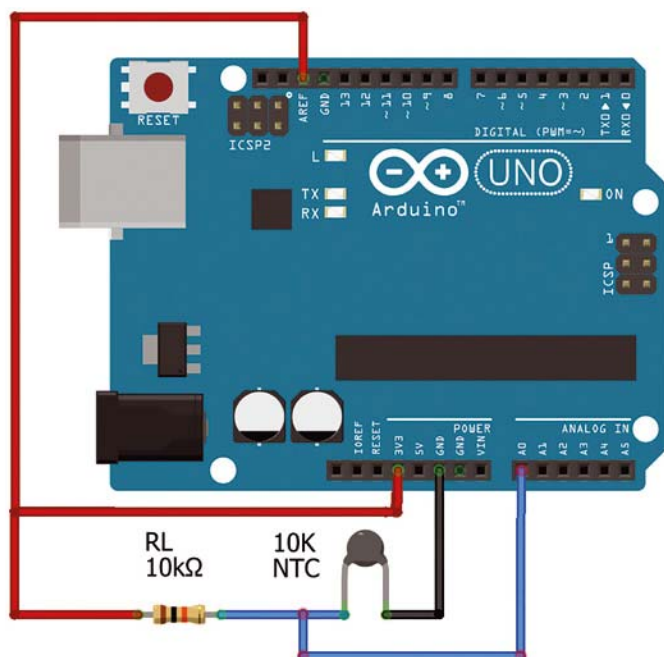
Fig. 1
Un esempio di
termistore.



entrare nel dettaglio precisiamo che esistono essenzialmente due categorie di dispositivi idonei alla misura di temperatura: trasduttori e sensori integrati.

I trasduttori sono componenti passivi (quindi privi di alimentazione esterna) che variano una loro grandezza elettrica caratteristica in funzione della temperatura, che quindi trasducono una grandezza fisica in un'altra; da qui il termine "trasduttori". I sensori sono componenti elettronici più complessi che si basano su un trasduttore ed integrano una componentistica che permette di semplificare il loro interfacciamento, ad esempio a un microcontrollore o microprocessore. In quest'ultima categoria troviamo i componenti LM35, MCP9700, TMP36 che forniscono in uscita una tensione proporzionale alla temperatura, decisamente semplice da gestire con un ingresso analogico. Nella stessa categoria troviamo i trasduttori DHT11, DHT22 e DS18B20 che forniscono in uscita la grandezza rilevata in forma digitale ed utilizzano una sola linea

Fig. 2
Schema di
montaggio
della
resistenza
NTC.



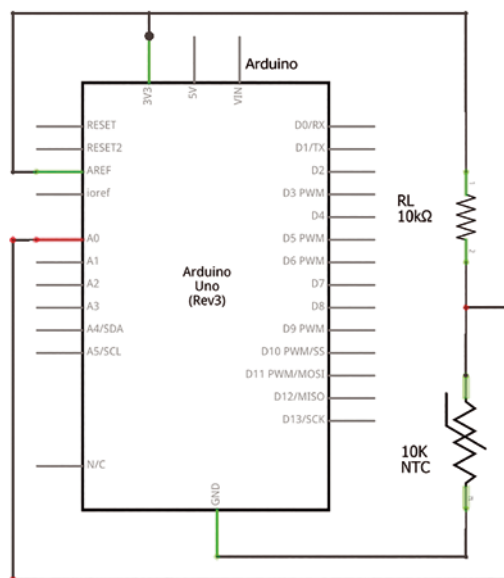
di comunicazione con la logica programmabile. In questo articolo ci concentriamo però sull'uso dei trasduttori, prendendo in considerazione elementi discreti come i termistori, le termoresistenze e le termocoppie, descrivendo il loro utilizzo pratico e i loro pro e contro.

I tre elementi in questione sono tutti trasduttori di temperatura, ciascuno dei quali ha specificità che sono tali da indicarlo e farlo preferire agli altri per determinate applicazioni: per esempio le termocoppie si utilizzano spesso per la misura di altissime temperature, laddove né i termistori, né le termoresistenze riuscirebbero a lavorare; è il caso della misura della temperatura nelle colate di vetro e di metalli. In quest'ultimo caso, per esempio nella metallurgia dell'acciaio, conoscere la temperatura consente di determinare, con l'aiuto del Diagramma ferro-carbonio, il tenore di carbonio dell'acciaio che si desidera ottenere. Ma vediamo in ordine una alla volta.

IL TERMISTORE

Questo componente è sostanzialmente un resistore che viene realizzato con materiale semiconduttore, il cui valore resistivo dipende (nel senso che ne è alterato) in modo più o meno significativo dalla temperatura; il componente in questione si presenta tipicamente come mostrato nella **Fig. 1**, che ne mostra una versione molto diffusa, tuttavia

Fig. 3
Schema di
collegamento
della
resistenza
NTC.



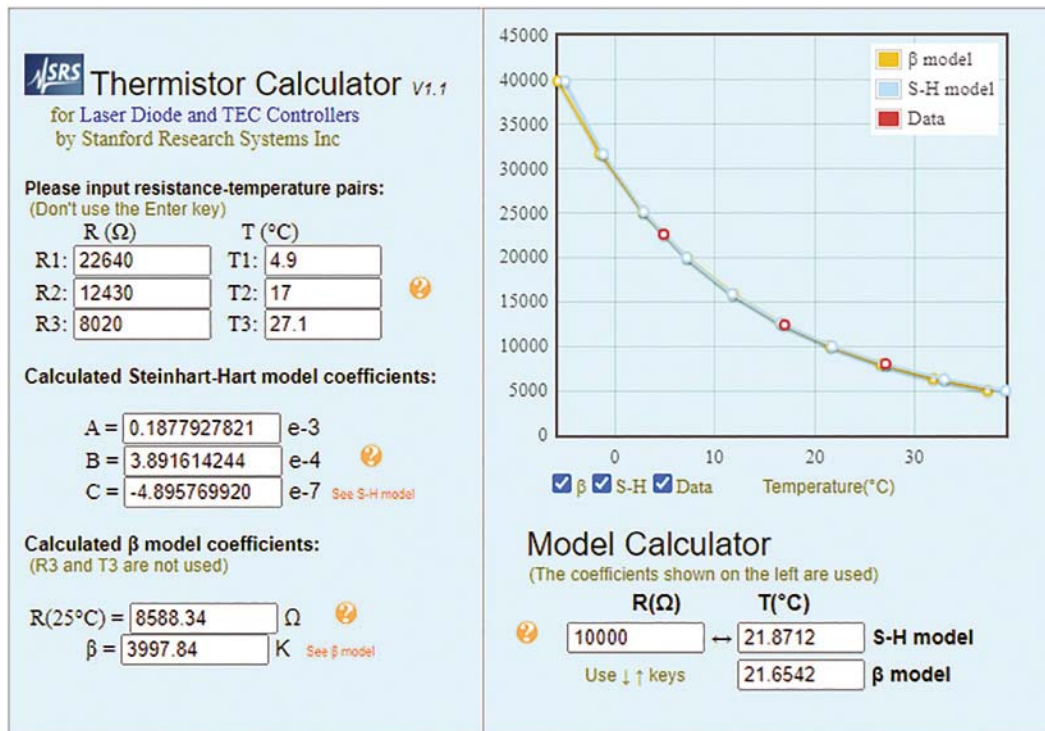


Fig. 4
Misura
sperimentale
dei parametri
R0 e B.

può essere incapsulato in vari contenitori e si può presentare sotto forma di disco rivestito da resina, eventualmente con bande colorate che indicano, secondo il Codice dei colori, il valore resistivo alla temperatura di riferimento (tipicamente 25 °C). Di termistori ne esistono di due tipi, ossia NTC e PTC, a seconda di come la temperatura influenza il loro valore resistivo: nel primo caso il termistore ha un coefficiente di temperatura negativo e nel secondo, coefficiente positivo. Se la resistenza diminuisce all'aumentare della temperatura, il componente è appunto l'NTC (acronimo di Negative Temperature Coefficient) mentre se la resistenza aumenta all'aumentare della temperatura, si tratta di un termistore PTC (acronimo di Positive Temperature Coefficient).

I termistori PTC sono spesso utilizzati in ambito elettronico sotto forma di fusibili auto-ripristinanti, sfruttando la loro proprietà di manifestare un aumento di resistenza con la temperatura che avviene in modo molto repentino, interrompendo quasi il circuito qualora la corrente dovesse aumentare oltre un certo limite, tale che per effetto joule riscalda rapidamente il componente. Anche la scheda Arduino ne integra uno per proteggere la porta USB del PC da eventuali cortocircuiti. Per le nostre prove abbiamo usato un termistore con rivestimento in resina epossidica di resistenza nominale 10 kohm commercializzato dalla Futura Elettronica (www.futurashop.it) con

codice NTC10K0. Le NTC presentano una caratteristica non lineare pertanto l'espressione che lega il valore resistivo alla temperatura è un'espressione complessa, la più importante è quella descritta da Steinhart-Hart:

$$\frac{1}{T} = a + b \cdot \ln(R_T) + c \cdot \ln^3(R_T)$$

In essa T è la temperatura (in kelvin), RT è la resistenza della NTC e a, b, c sono tre opportuni coefficienti. Vista la complessità di questa equazione si preferisce spesso utilizzare l'equazione seguente:

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{B} \ln\left(\frac{R_T}{R_0}\right)$$

nella quale T0 è la temperatura di riferimento (298,15 K) e R0 è resistenza alla temperatura di riferimento; nel nostro caso R0 vale 10 kohm ed il parametro B viene indicato in 4100 K. Per utilizzare una NTC è sufficiente creare un partitore di tensione con una resistenza nota (RL) in modo da ottenere una tensione variabile con la temperatura, facilmente misurabile con una scheda Arduino; alimenteremo il tutto con la tensione di 3,3V più precisa e meno soggetta a rumore di quella fornita dalla USB, come indicato in **Fig. 2** e **Fig. 3**.

Di solito possiamo utilizzare i parametri (R0 e B) forniti dal produttore ma se vogliamo ottenere la massima precisione è opportuno ricavarli misuran-

Listato 1 - Sketch per la misura di temperatura con una NTC

```
// Misura di temperatura con termistore NTC
// NTC connessa con un partitore verso massa
// Il partitore è alimentato a 3,3V connesso a Aref

// valore resistivo della NTC alla temperatura di riferimento (25°C)
#define Ro 8588
// temperatura di riferimento in Kelvin (273.15 + 25)
#define To 298.15
// numero di campionamento per il calcolo della media
#define NUMSAMPLES 10
// Coefficiente beta (normalmente 3000-5000)
#define BCOEFFICIENT 3998
// valore della resistenza serie del partitore
#define RL 10000.0

int i; float somma, media, RT, temperatura;

void setup(void) {
  Serial.begin(115200);
  analogReference(EXTERNAL);
}

void loop(void) {
  // campiona NUMSAMPLES valori
  somma=0;
  for (i=0; i< NUMSAMPLES; i++) {
    somma += analogRead(A0);
    delay(100);
  }

  // calcolo la media dei valori letti
  media = somma/NUMSAMPLES;
  Serial.print("\nValore medio dei dati letti ");
  Serial.println(media,1);

  // calcolo il valore della resistenza della NTC
  RT = RL*media/(1023-media);
  Serial.print("Resistenza termistore ");
  Serial.println(RT,0);

  // calcolo la temperatura
  temperatura = 1.0 / (1.0/To + (1.0/BCOEFFICIENT) * log(RT/Ro));

  // conversione da Kelvin a Centigradi
  temperatura -= 273.15;
  Serial.print("Temperatura ");
  Serial.print(temperatura,1);
  Serial.println(" °C");
}
```

do R_T a due differenti temperature scelte all'interno del range per il quale verrà usato il termometro. Ovviamente queste misure dovranno essere eseguite con la massima attenzione ed usando un termometro di buona precisione. Sono molti i servizi online che ci possono aiutare in questa fase, noi abbiamo usato www.thinksrs.com che ha fornito i risultati visibili in **Fig. 4**. È opportuno conoscere con buona precisione anche la resistenza del partitore R_L .

A questo punto passiamo dalla teoria alla pratica e scriviamo lo sketch da far girare sulla scheda Arduino, esso consta essenzialmente di tre parti; la prima prevede di leggere il valore fornito dall'ADC

facendo la media di 10 valori per ottenere una lettura più "pulita", nella seconda si ricava la misura di R_T a partire dal valore fornito dall'ADC (N_{ADC}) e per ultimo si determina la temperatura con l'equazione precedentemente menzionata. Per la misura del valore R_T è stata usata l'equazione

$$R_T = R_L \cdot \frac{N_{ADC}}{1023 - N_{ADC}}$$

ovvero Arduino è usato come ohmmetro, lo sketch è riportato nel **Listato 1**.

In **Fig. 5** potete vedere la schermata del Serial monitor con in esecuzione lo sketch ed il valore di temperatura misurato. Da diverse prove esegui-

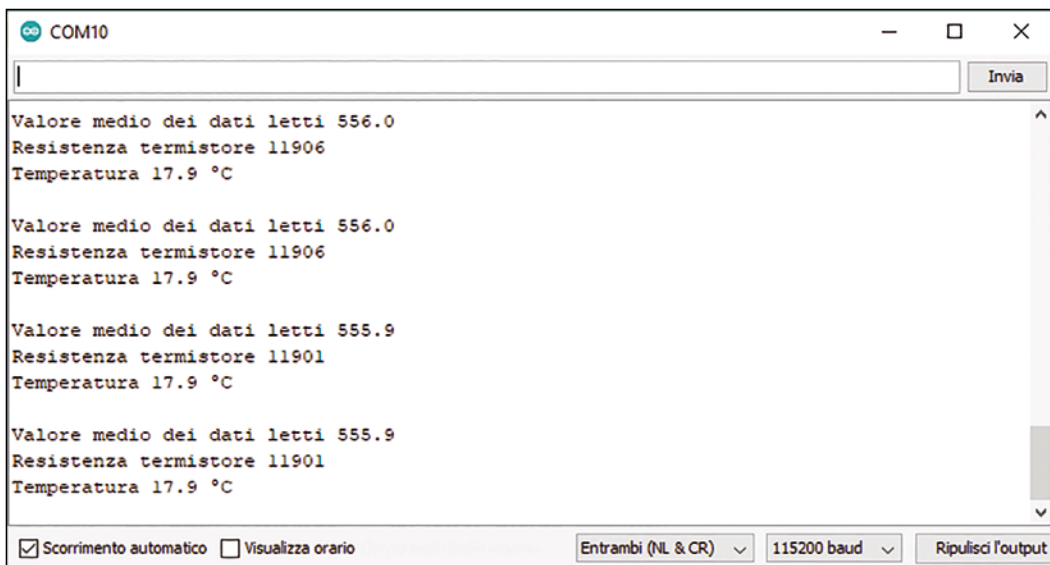


Fig. 5
Serial monitor con in esecuzione lo sketch del Listato 1.

te possiamo dire che con le dovute calibrazioni la precisione ottenuta è molto buona arrivando a misurare in modo significativo anche i decimi di grado. Per andare oltre occorre considerare i limiti del convertitore di Arduino, la variabilità dei parametri della NTC e della resistenza del partitore, oltre ovviamente all'auto-riscaldamento della NTC dovuto al passaggio della corrente del partitore.

LA TERMORESISTENZA

Si tratta di un resistore realizzato con materiale metallico il cui valore resistivo dipende in modo proporzionale dalla temperatura e che si presenta come in **Fig. 6**. Le più comuni sono quelle realizzare con il platino, che assicura un'ottima linearità e un'elevata sensibilità, in questo caso prendono il nome di PT100 o PT1000 a seconda del valore resistivo a 0°C. La resistenza è legata alla temperatura da un'equazione lineare:

$$R_T = R_0 (1 + \alpha \cdot T)$$

dove R_0 è il valore resistivo a 0°C e α è il coefficiente che determina la sensibilità e dipende dal materiale usato, nel caso del platino vale 0,00385 1/°C. Per il nostro articolo abbiamo usato una PT100, commercializzata dalla Futura Elettronica con il codice prodotto 1606-PT100.

Le termoresistenze sono spesso usate in ambito industriale per le loro buone prestazioni e la loro robustezza e pertanto in commercio esistono appositi circuiti di condizionamento specifici; comunque anche con una semplice scheda Arduino si possono realizzare dei buoni termometri a patto di usare alcuni accorgimenti. Calcolatrice alla

mano una PT100 presenta 100Ω a 0°C e 138,5Ω a 100°C ben lontani dalle variazioni di un termistore, occorre poi fare attenzione all'auto-riscaldamento pertanto è necessario limitare al massimo la corrente usata per la misura. Il circuito è lo stesso di prima (potremmo omettere il filo che porta i 3,3V ad Aref), ovvero il partitore di tensione, con una R_L che questa volta sarà di 1kΩ. La termoresistenza in nostro possesso arriva fino a 450°C però i fili di collegamento possono sopportare al massimo 200°C, pertanto considerando un range di misura tra 0°C e 200°C e la tensione fornita dal partitore andrà da 0,3V a 0,5V, useremo pertanto la tensione interna di 1,1V come riferimento per l'ADC. La formula che determina la resistenza della termoresistenza (R_T) noto il valore fornito dall'ADC è leggermente diversa da prima perché il partitore è alimentato da una V_{CC} di 3,3V mentre l'ADC ha una V_{ref} di 1,1V:

$$R_T = \frac{V_{ref} \cdot N}{1023 \cdot V_{CC} - N \cdot V_{ref}}$$

Nota la R_T è possibile determinare la temperatura con la formula inversa:



Fig. 6
Termoresistenza PT100.

Listato 2 - Sketch per la misura di temperatura con una PT100.

```
// Misura di temperatura con termoresistenza PT100
// PT100 connessa con un partitore verso massa
// Il partitore è alimentato a 3,3V

// valore resistivo della PT100 alla temperatura di riferimento (0°C)
#define R0 100.0
// numero di campionamento per il calcolo della media
#define NUMSAMPLES 20
// Coefficiente alfa
#define alfa 0.00385
// valore della resistenza serie del partitore
#define RL 995.0
// valore di tensione che alimenta il partitore
#define Vcc 3.293
// valore di riferimento per l'ADC
#define Vref 1.1

int i; float somma, media, RT, temperatura;

void setup(void) {
  Serial.begin(115200);
  analogReference(INTERNAL);
}

void loop(void) {
  // campiona NUMSAMPLES valori
  somma=0;
  for (i=0; i< NUMSAMPLES; i++) {
    somma += analogRead(A0);
    delay(100);
  }

  // calcolo la media dei valori letti
  media = somma/NUMSAMPLES;
  Serial.print("\nValore medio dei dati letti ");
  Serial.println(media,1);

  // calcolo il valore della resistenza della PTC
  RT = RL*Vref*media/(Vcc*1023.0-media*Vref);
  Serial.print("Resistenza termistore ");
  Serial.println(RT,1);

  // calcolo la temperatura
  temperatura = (RT - R0) / (alfa*R0);

  Serial.print("Temperatura ");
  Serial.print(temperatura,1);
  Serial.println(" °C");
}
```

$$T = \frac{R_T - R_0}{\alpha \cdot R_0}$$

Nel **Listato 2** proponiamo le righe di codice che implementano le formule precedentemente ricavate. L'esigua escursione di tensione non permette di ottenere elevate risoluzioni, pur avendo stampato i decimali (**Fig. 7**) questi risultano poco significativi, inoltre visto il piccolo valore resistivo della PT100 anche i collegamenti sulla breadboard posso influenzare la misura. Rispetto a prima abbiamo però il vantaggio di misurare temperature più elevate,

inoltre il contenitore metallico stagno della PT100 permette un suo utilizzo anche in ambienti critici.

LA TERMOCOPPIA

Si tratta di un sensore di temperatura formato assemblando due elementi metallici di diversa natura, ovvero due differenti metalli, che devono avere potenziale di contatto (dovuto alla struttura elettronica periferica dei rispettivi atomi) abbastanza diverso; il punto di unione, chiamato "giunto caldo", va applicato al punto in cui compiere la misura, mentre le due estremità libere, chiamate



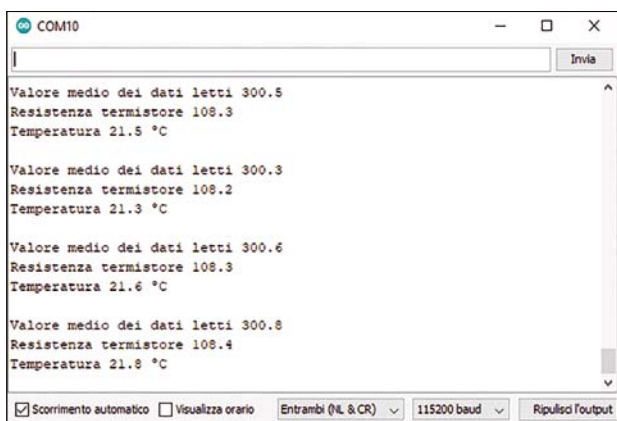


Fig. 7
Serial
Monitor con
in esecuzione
lo sketch del
Listato 2.

“giunto freddo”, sono mantenute ad una temperatura nota (**Fig. 8**). Ai capi delle estremità libere sarà presente una tensione proporzionale alla differenza di temperatura tra il giunto caldo ed il giunto freddo, questo avviene grazie all’effetto Seebeck (dal nome del fisico che studiò il fenomeno). Seebeck (ma ancor prima Alessandro Volta) studiando gli effetti termoelettrici nei metalli si era accorto che in un circuito elettrico chiuso, realizzato con fili metallici di diversa natura e sottoposti a una differenza di temperatura, si veniva a creare una circolazione di corrente.

Realizzare una termocoppia è semplice ed economico ed essendo realizzata in metallo è anche adatta all’impiego in ambito industriale per la misura di temperatura oltre i 1000 °C. La tensione in uscita dipende dalla differenza di temperatura tra i due giunti pertanto occorre compensare la temperatura sul giunto freddo che a sua volta potrebbe essere influenzato dalla temperatura del giunto caldo vista la trasmissibilità termica dei metalli. A questo si aggiunge il fatto che l’effetto Seebeck varia con la temperatura rendendo la

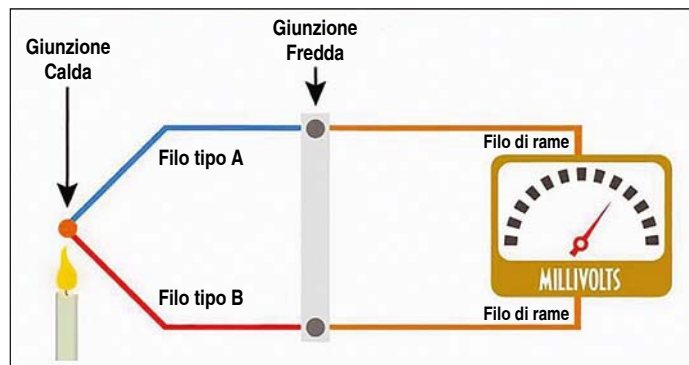


Fig. 8
Rappresentazione
di una termocoppia.

tensione di uscita non perfettamente lineare con la temperatura, eventualmente da linearizzare via software conoscendo in dettaglio la caratteristica di trasferimento della termocoppia. Insomma, avete capito che non sarà possibile ottenere una grande precisione nella misura. A seconda

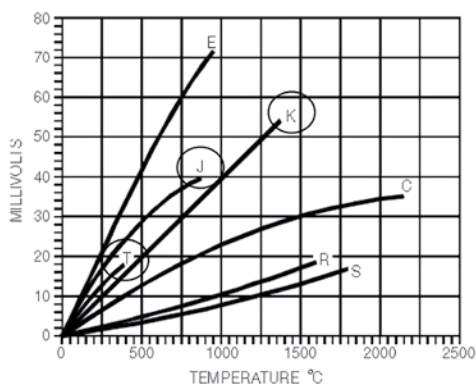


Fig. 9
Caratteristica
di trasferimento
delle
principali
termocoppie
commerciali.

Tabella 1
Caratteristiche delle principali termocoppie commerciali.

TYPE	T-WIRE	T+WIRE	TEMP RANGE (°C)	SENSIVITY (μV/°C)
K	Alumel	Chromel	-270 to +1372	41.276 (0°C to +1000°C)
J	Constantan	Iron	-210 to +1200	57.953 (0°C to +1000°C)
N	Nisil	Nicrosil	-270 to +1300	36.256 (0°C to +1000°C)
S	Platinum	Platinum / Rhodium	-50 to +1768	9.587 (0°C to +1000°C)
T	Constantan	Copper	-270 to +400	52.18 (0°C to +400°C)
E	Constantan	Chromel	-270 to +1000	76.373 (0°C to +1000°C)
R	Platinum	Platinum / Rhodium	-50 to +1768	10.506 (0°C to +1000°C)



Fig. 10
Tipica
termocoppia
economica
fornita con i
multimetri.

Fig. 11
Termocoppia
di codice
1606-MODMAX6675.



Fig. 12
Termocoppia di codice
8220-VMA339.



dei materiali usati è possibile realizzare diverse termocoppie con differenti range di temperatura e linearità, come visibile in **Fig. 9** e nella **Tabella 1**. Nella **Fig. 10** potete vedere una comune termocoppia generalmente fornita assieme ai multimetri che dispongono della misura di temperatura. Per le nostre prove abbiamo utilizzato due termocoppie di tipo K distribuite dalla Futura Elettronica, con codice prodotto 1606-MODMAX6675 (**Fig. 11**) e 8220-VMA339 (**Fig. 12**) entrambe adatte a misure da 0°C e 600°C.

Le termocoppie di tipo K sono le più comuni perché sufficientemente lineari e con una buona sensibilità. Per queste ultime la tensione di uscita è di appena 4mV a 100°C per cui necessitano di un opportuno circuito di condizionamento che amplifichi il debole segnale prodotto e compensi l'influenza della temperatura ambiente. Entrambi i prodotti indicati vengono già forniti di opportuno circuito di interfaccia basato sull'inte-

grato della Maxim MAX6675 che tramite interfaccia digitale SPI fornisce il valore di temperatura già calibrato con una risoluzione di 12 bit assicurando una risoluzione di un quarto di grado, la precisione è invece di circa 1,5°C.

Per il collegamento fisico dei componenti fate riferimento allo schema riportato nella **Fig. 13**, assicuratevi che il filo rosso della termocoppia venga collegato al morsetto contrassegnato con il segno più e di conseguenza il filo blu al morsetto contrassegnato con un meno.

Avviamo l'IDE di Arduino e tramite il gestore delle librerie scarichiamo la libreria MAX6675 Library by Adafruit (**Fig. 14**); lo sketch usato nella nostra prova è riportato nel **Listato 3**. Poche righe di codice sono sufficienti per visualizzare su Serial Monitor la temperatura, dobbiamo però accettare il fatto di non poter raggiungere elevate precisioni, nel nostro caso a temperatura ambiente vengono misurati ben 2°C in più.

↓ Listato 3 - Sketch per la misura di temperatura con termocoppia e MAX6675

```
// Misura di temperatura con termocoppia e MAX6675
// MAX6675 Library by Adafruit

#include "max6675.h"

int thermoSO = 8;
int thermoCS = 9;
int thermoCLK = 10;

MAX6675 thermocouple(thermoCLK, thermoCS, thermoSO);

void setup() {
  Serial.begin(115200);
  Serial.println("MAX6675 test");
  // attende avvio MAX6675
  delay(500);
}

void loop() {
  Serial.print("T= ");
  Serial.print(thermocouple.readCelsius(),1);
  Serial.println(" °C");
  // massimo 4 letture al secondo!
  delay(1000);
}
```

Fig. 13

Collegamenti elettrici tra la termocoppia ed Arduino.

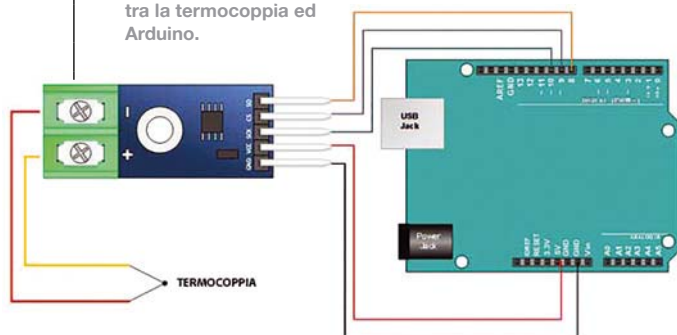


Tabella 2

Riassunto delle principali caratteristiche dei sensori usati nell'articolo.

SENSORE	RANGE	SENSIBILITÀ	PRECISIONE	COSTO EURO
Termistore	-40 a +150°C	alta	alta	1
Termoresistenza	-50 a +450°C	media	media	4
Termocoppia	0 a 600°C	bassa	bassa	8

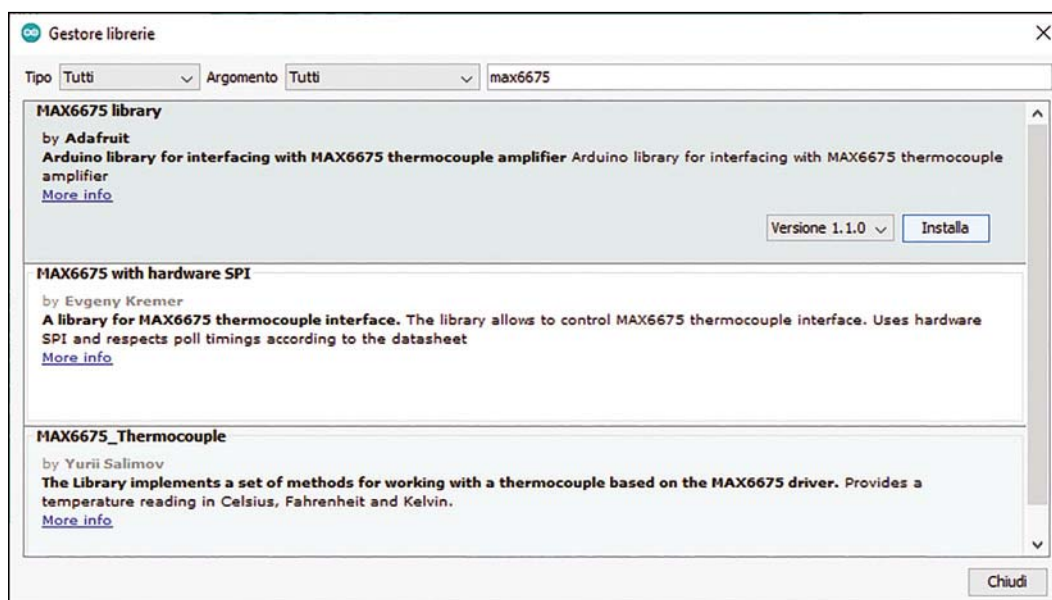


Fig. 14
Installazione della libreria.

CONCLUSIONI

Per verificare la bontà delle soluzioni proposte abbiamo testato il termistore a diverse temperature ambiente ottenendo sempre buoni risultati. La PT100 e la termocoppia sono state testate immergendole in acqua bollente ottenendo misure molto prossime ai 100°C per la precisione 102,5°C per la PT100 e 98,5°C per la termocoppia. In **Tabella 2** riassumiamo le principali caratteristiche dei sensori utilizzati nell'articolo, da cui possiamo trarre alcuni importanti conclusioni. Il termistore ha il range di misura più contenuto ma un'alta precisione, ideale per misure di temperatura ambiente come in stazioni meteo o termostati casalinghi. La termoresistenza ha un range più esteso, una buona precisione, e viene spesso usata in ambito industriale in abbinamento con opportuno circuito di interfaccia. La termocoppia è l'unica che non permette misure di temperature negative, in compenso può essere usata per temperature più elevate inoltre viene già fornita di un dedicato circuito di interfaccia, senza

necessariamente incidere eccessivamente sul prezzo di vendita. Tra i file disponibili con questo numero della rivista trovate un codice che utilizza un display LCD per visualizzare la temperatura, utile per un utilizzo stand-alone. A voi la scelta del sistema di misura che ritenete più idoneo per la vostra applicazione. □



Cosa occorre?

I sensori utilizzati in questo articolo sono disponibili presso Futura Elettronica. La termoresistenza da 10KOhm (cod. NTC10K0) è in vendita a Euro 0,70, il sensore PT100 (cod. PT100) è disponibile a Euro 4,50, la breakout con termocoppia (cod. MODMAX6675) costa Euro 8,00, la sonda di temperatura (cod. VMA339) è disponibile a Euro 19,00. 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>

Purificatore d'aria a raggi ultravioletti

VERSIONE
DA PARETE

€179,00

Cod. MICROUVAIR-P

Alimentazione: 12 Vcc
Potenza assorbita: @ 3.6 W
Durata sorgente UV: @20000 ore
Capacità sterilizzazione: 60 m³/h
Dimensioni: altezza 25 cm x 10.5 cm x 7 cm

È un sistema di sterilizzazione dell'aria che assicura l'eliminazione di virus e batteri.

Riduce il rischio di contaminazione delle persone all'interno degli uffici.

Rende gli ambienti batteriologicamente sani eliminando anche le spore.

È in grado di effettuare un trattamento di disinfezione dell'aria in ambienti fino a 60 m³.

Può funzionare continuativamente anche in presenza di persone.

Abbatte il 99% della carica batterica aerea dopo circa un'ora di funzionamento.



MICROUVAIR è un sistema di sanificazione aria con ventilazione forzata a ciclo chiuso basato su led UVC. L'aria in ingresso viene purificata attraverso un filtro ad alta densità (30 PPI), quindi viene sterilizzata tramite luce ultravioletta emessa da led UVC in grado di effettuare la massima azione germicida. L'aria sanificata viene quindi filtrata ulteriormente da un filtro meccanico ad alta densità e quindi rimessa nell'ambiente.



VERSIONE
DA TAVOLO

€199,00

Cod. MICROUVAIR-T



Prezzi IVA inclusa.

SANIFICATORE UVC PER OGGETTI

Sanificatore automatico con LED UVC, particolarmente indicato in tutte le situazioni dove si rende necessario igienizzare, sanificare e comunque disinfettare oggetti potenzialmente depositari di particelle infettive (virus, batteri, ecc.) come mascherine chirurgiche, utensili, occhiali, chiavi, ecc. È costituito da un contenitore in plexiglass con un coperchio incernierato all'interno del quale si trova una strip a LED UV-C con quattro LED SMD UV-C tipo 3535 da 5 mW ciascuno con lunghezza d'onda della radiazione emessa nella banda dei 265 e i 278 nm, gamma ideale per sterilizzare e sanificare superfici e oggetti. Il set contiene tutte le parti in plexiglass, le parti stampate in 3D, l'elettronica e l'alimentatore.



€80,00

Cod. UVCSAN



COSTRUIAMO UNA NN CON PyTorch

dell'Ing. **DANIELE DENARO**



Conosciamo e utilizziamo la libreria Pytorch di Python, per poter sperimentare nuove Reti Neurali con uno strumento associato a Python.

Nel recente passato, in vari articoli, abbiamo utilizzato Reti Neurali (NN) già configurate ed addestrate per eseguire particolari compiti: per esempio una rete YOLO per identificare gli oggetti in una fotografia o in un fotogramma della webcam, così da realizzare un antifurto sensibile alle sole figure umane, piuttosto che una trappola fotografica per alcuni tipi di animali selvatici. Poi abbiamo utilizzato una Rete Neurale Ricorrente (RNN) definita ed addestrata da GOOGLE per riconoscere il parlato. Una rete chiamata WaveNet, anch'essa addestrata da GOOGLE, è stata invece usata per la sintesi vocale. Infine, la rete OpenFace ci è servita per far riconoscere i volti a una scheda Raspberry Pi. Abbiamo dato, in questo modo, una serie di capacità "intelligenti" a Raspberry Pi o a un qualunque altro nostro computer di media potenza. Ma se ora volessimo svincolarci da strutture già preconfezionate da parte di ricercatori o aziende

e sperimentare in proprio? Ebbene, in alcuni articoli di circa un anno fa, abbiamo affrontato il problema degli ambienti di sviluppo per Reti Neurali e per il Deep Learning come Tensorflow e Keras, i quali sono sicuramente software potenti, ma un po' complessi. Nel frattempo si sta affermando come ambiente molto popolare, la libreria Pytorch di Python, la quale ha "dalla sua" una maggiore semplicità d'uso e una completa integrazione in Python, che a sua volta si sta affermando come linguaggio di sviluppo in questi ambienti di intelligenza artificiale e non solo in questi. A questo punto eleggiamo Python con Pytorch come linguaggio di sviluppo e proviamo a descrivere in modo semplice l'approccio alla costruzione di strutture di NN. In questa sede non ci è possibile fare un corso "accelerato" sul linguaggio Python, per cui lo daremo per acquisito (anche se in forma non approfondita). Invece descriveremo le principali caratteristiche e funzioni di Pytorch, partendo da strutture elementari. Innanzi tutto installiamo Pytorch, impartendo il comando seguente:

```
pip3 install torch
```

Notate che bisogna comunque aver già installato la libreria "numpy" di Python, perché Pytorch la richiede. Esiste anche la versione che può utilizzare CUDA, per esempio su Jetson NANO o su Windows con scheda CUDA. In questo caso si può fare riferimento al sito <https://pytorch.org/get-started/locally/>, dove vengono illustrati i comandi specifici di installazione.

Come integrazione a Pytorch esistono altre importanti librerie come "torchvision" e "torchaudio" che possono essere installate con lo stesso comando.

IL TENSORE: QUESTO SCONOSCIUTO

Sia Tensorflow che Pytorch pongono in primo piano l'oggetto tensore (e addirittura Tensorflow ce l'ha nel nome!). Già questo termine può spaventare, se poi ci aggiungiamo matrici e calcolo

matriciale, il panico prende il sopravvento. Cerchiamo di demistificare e facciamo anche un po di chiarezza sulle NN. Storicamente le NN nascono come imitazione dei circuiti cerebrali. Di reti neurali si comincia a parlare già negli anni 40 del 900 con McCulloch che è uno psichiatra. Ma anche riferendoci alle sole reti ad addestramento supervisionato (da cui originano le strutture deep-learning), sia i primi "percettroni" di F. Rosenblatt che le prime reti multistrato di "David E. Rumelhart, James L. McClelland", sono strutture ideate da psicologi. Il loro libro "PDP. Microstruttura dei processi cognitivi" (considerato l'inizio del connessionismo: 1986), è stato appunto scritto da psicologi. Ma poiché aprivano un interessante campo nel calcolo, ci si sono fiondati gli informatici e i matematici.

Risultato: maggiore pulizia formale ma anche maggiore astrusità.

All'origine c'è il neurone, come mostrato nella Fig. 1, che propone a confronto quello naturale e quello artificiale. In una rete multistrato, i neuroni di uno strato sono connessi con tutti quelli dello strato successivo, come proposto nella Fig. 2.

Se teniamo presente che :

$$\begin{aligned} y_0 &= w_{00}x_0 + w_{01}x_1 + \dots + w_{0n}x_n \\ y_1 &= w_{10}x_0 + w_{11}x_1 + \dots + w_{1n}x_n \\ &\dots \\ y_m &= w_{m0}x_0 + w_{m1}x_1 + \dots + w_{mn}x_n \end{aligned}$$

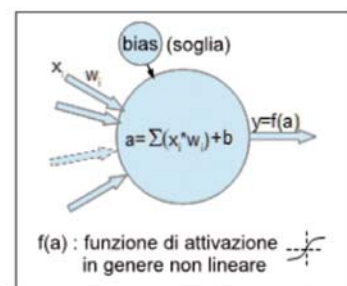
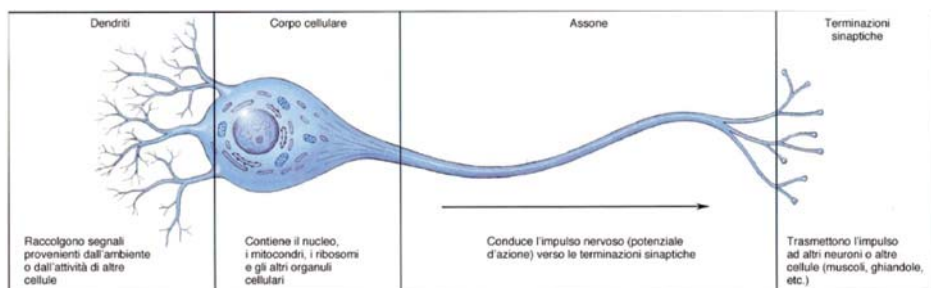
corrisponde a:

$$\begin{bmatrix} y_0 \\ y_1 \\ \dots \\ y_m \end{bmatrix} = \begin{bmatrix} w_{00} & w_{01} & \dots & w_{0n} \\ w_{10} & w_{11} & \dots & w_{1n} \\ \dots & \dots & \dots & \dots \\ w_{m0} & w_{m1} & \dots & w_{mn} \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix}$$

e, più sinteticamente, a $Y_m = W_{m \times n} \cdot X_n$ allora i valori di uscita formano il vettore Y, W è una matrice e X è il vettore di ingresso.

Ma sono tutti chiamati **tensori**.

Fig. 1
Neurone
naturale e
neurone
artificiale
(schematico).



Questo perché in algebra lineare si era partiti a distinguere le due strutture principali: vettore come elemento a una dimensione e matrice come elemento a due dimensioni. Ma poi con il crescere delle dimensioni utilizzate nei calcoli di fisica e ingegneria, si è finito per avere matrici multidimensionali, e poiché le operazioni definite in algebra lineare sono le stesse per qualunque dimensione dei soggetti, si usa spesso la parola *tensor* per definire un elemento di qualunque dimensione. Notate che le soglie (bias) sono state omesse per semplicità ma potete pensare che sono aggiunte semplicemente al vettore di uscita. Inoltre ai valori del vettore di uscita va applicata la funzione di attivazione (non rappresentata).

Tornando alle nostre reti multistrato, per costruire un software che le elabori, possiamo utilizzare il concetto di neurone definendo per esempio una classe (usando la programmazione ad oggetti) che contenga tutti i pesi a lui afferenti con l'indicazione del neurone di provenienza. Quindi si tratterà di effettuare loop di propagazione del segnale per tutti i neuroni di uno strato e per tutti gli strati. Oppure, possiamo utilizzare le efficienti librerie di calcolo matriciale presenti nei principali linguaggi e trasferire il segnale da uno strato al successivo applicando un prodotto matriciale (o più propriamente tensoriale) tra i pesi di collegamento fra uno strato e l'altro. Il primo approccio è stato utilizzato, per esempio, nella libreria NN per Arduino https://github.com/open-electronics/Artificial_Intelligence (directory Arduino). Infatti il calcolo matriciale non è proprio la funzionalità tipica di un hardware limitato come Arduino. Questo approccio è inoltre più vicino a quelli adottati dalla **"Neuromorphic Computation"** o **"Neuromorphic AI"** (vedi riquadro dedicato). Il secondo approccio è quello più comune, per i motivi detti e soprattutto perché può contare sulle potenti schede grafiche *multicore*, come CUDA, nate proprio per il calcolo matriciale distribuito, utilizzato nelle elaborazioni classiche delle immagini. Per chiarezza dobbiamo però precisare che l'elaborazione distribuita delle schede grafiche riguarda l'ottimizzazione del calcolo matriciale, e non riguarda una distribuzione dei neuroni tra i *core*, come invece si cerca di fare nei computer neuromorfici.

IL GRADIENTE

Pytorch assegna una variabile predefinita per tutti i tensori che costituiscono la struttura di una NN. Questa variabile conterrà il valore del **gradiente** per ogni elemento del tensore. Questi valori

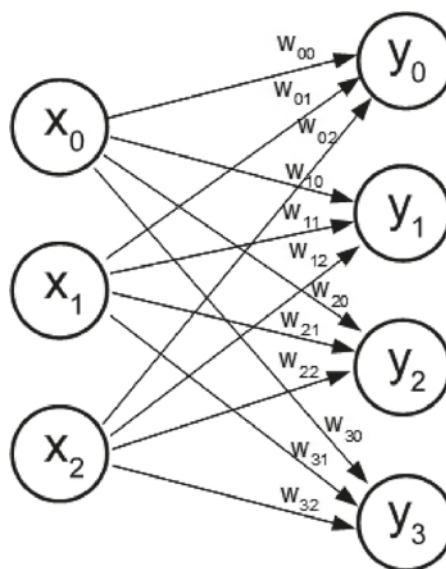


Fig. 2
Strato
di una NN.

vengono calcolati nella fase di "back-propagation". Cerchiamo di spiegare cosa è il gradiente e perché è fondamentale nelle reti "feed-forward", ovvero nelle reti ad apprendimento supervisionato, caratteristiche del deep-learning.

Una rete ad apprendimento supervisionato, può essere pensata come una black-box con numerose manopole di regolazione. Queste manopole di regolazione (una per ogni peso, ovvero parametro), saranno ruotate in modo da diminuire l'errore calcolato in quel punto ad ogni presentazione di un esempio. Il trucco sta nel poter calcolare il contributo all'errore per ogni parametro. Questo può essere fatto utilizzando la regola matematica della catena delle derivate. Senza entrare nei particolari, Pytorch può calcolare automaticamente il gradiente di una concatenazione di operazioni come quelle fatte in una rete "feed-forward". L'unica cosa che ha bisogno di conoscere è la successione degli strati. Questa caratteristica di Pytorch è chiamata **"Autograd"**. Il *gradiente* (dell'errore) relativo ad un parametro della rete, quindi, non è altro che la pendenza del valore del parametro in funzione dell'errore complessivo. Per aggiustare la rete basta applicare una piccola variazione al valore del parametro in direzione contraria alla pendenza e vedere come va (**Fig. 3**).

DEFINIRE UNA NN CON PYTORCH

La documentazione ufficiale di Pytorch la potete trovare in Rete, all'indirizzo <https://pytorch.org/docs/stable/index.html>. Il modulo principale della nostra libreria Pytorch è **"torch.nn"**.

Questo modulo contiene gli elementi necessari per costruire ed utilizzare una rete neurale (NN).

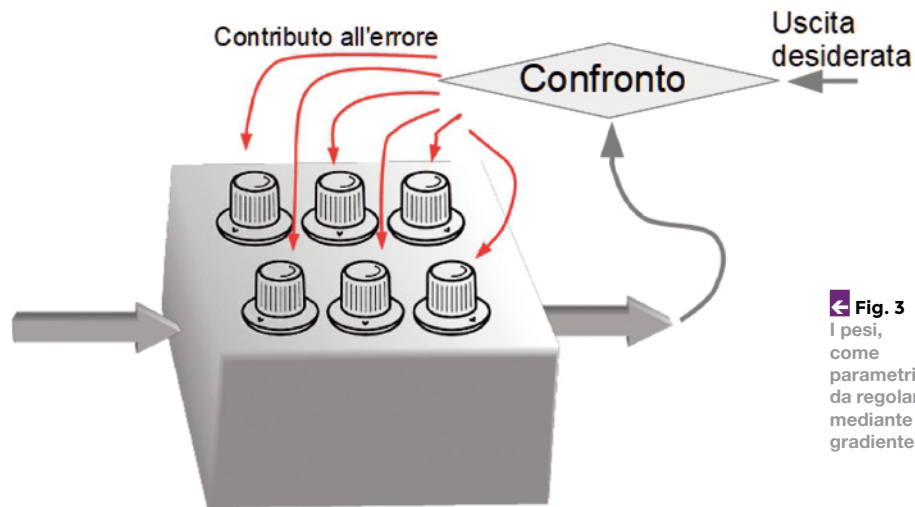


Fig. 3
I pesi,
come
parametri
da regolare
mediante il
gradiente.

La base della libreria `torch.nn` che, da questo momento chiameremo in Python semplicemente `nn` (attraverso il comando `import torch.nn as nn`) è l'oggetto (di tipo `Container`) **Module** → **modello**. Il generico **modello** corrisponde alla **rete neurale** ed è assimilabile ad una funzione che, dato un determinato input, restituisce un output, secondo la relazione seguente:

```
y = net(x) #dove net è un Module
```

Un `Module` può contenere a sua volta più `Module`, per esempio collegati sequenzialmente. Il `Module` più semplice è: **Linear** e corrisponde ad un semplice layer di una NN (**Fig. 4**). Per esempio `y = nn.Linear(3, 2)` corrisponde a un layer con 3 input e 2 output:

```
monoLayer = nn.Linear(3, 2, bias=False)
for p in monoLayer.named_parameters():
    print(p)
```

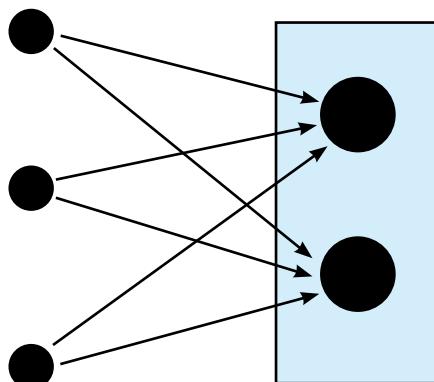


Fig. 4
Singolo layer o
monostrato.

che dà in uscita:

```
('weight', Parameter containing:
  tensor([[ 0.5479,  0.5177, -0.0769],
          [ 0.1744,  0.3426,  0.5109]], requires_
grad=True))
```

Includendo anche i bias e visualizzando meglio la struttura:

```
monoLayer = nn.Linear(3, 2, bias=True)
print("weights: ")
for i in monoLayer.weight[0]:
    print(" %.3f " %(i.item()), end='')
print()
for i in monoLayer.weight[1]:
    print(" %.3f " %(i.item()), end='')
print("\nbias")
for i in monoLayer.bias:
    print(" %.3f " %(i.item()), end='')
```

che dà in uscita:

```
weights:
-0.334 -0.382  0.431
-0.414  0.072 -0.474
biases
0.043  0.079
```

UTILIZZO DELLA RETE (FORWARD)

Il forward viene effettuato chiamando semplicemente la rete passandogli il parametro di input

```
x = th.tensor([1, -2, 3], dtype=th.float)
out = monoLayer(x)
print("out: ", end='')
for i in out:
```

▼ Listato 1

```
monoLayer = nn.Linear(3, 2, bias=True)      #creiamo il layer con pesi casuali
th.save(monoLayer, netfile)                 #salviamolo così da poterlo utilizzare sempre uguale
savedlayer = th.load(netfile)               #lo ricarichiamo (e gli cambiamo nome per chiarezza)
x = th.tensor([1, 2, 0], dtype=th.float)    #creiamo un input con tre valori
target = th.tensor([0.5, 0.5], dtype=th.float) #creiamo una uscita desiderata con due valori
#definiamo una modalità di correzione dei pesi con un certo learning rate ed un momento
optimizer = th.optim.SGD(savedlayer.parameters(), lr= 0.05, momentum=0.2)
#applicata ai parametri della rete (pesi e bias)
loss = nn.MSELoss()                        #definiamo la funzione loss
                                           #(funzione di valutazione dell'errore)

for ep in range(0, 15):
    y = savedlayer(x)                      #forward della rete
    print("y = %.3f %.3f" % (y[0].item(), y[1].item())) #visualizzazione risposta
    err = loss(y, target)                  #calcoliamo l'errore
    savedlayer.zero_grad()                #azzeriamo eventuali accumuli del gradiente
    err.backward()                        #appliciamo la propagazione del gradiente
    optimizer.step()                      #appliciamo la correzione ai pesi
```

```
print(" %.3f " % (i.item()), end='')
```

che dà in uscita:

```
out:  -1.683  2.426
```

ADDESTRAMENTO (TRAINING)

Continuando nel nostro esempio essenziale, proviamo ad addestrare la nostra monolayer a rispondere con un output (0.5, 0.5) ad un input del tipo (1, 2, 0); il relativo codice è proposto nel **Listato 1**. Spieghiamo alcune funzioni e notazioni.

- Import: *import torch as th* e *import torch.nn as nn*
- Per salvare la rete (con tutti i parametri) esiste la funzione base *th.save(rete, filerete)* (in genere tipo *.pth)
- Per caricare la rete salvata esiste la funzione *rete = th.load(filerete)*

Dopo poche ripetizioni (ne bastano 7) l'errore va a 0 e l'uscita è quella desiderata. Ma questo esempio didattico è una struttura troppo elementare (in pratica un perceptrone di Rosenblatt).

Per avere comportamenti sofisticati, ovvero risposte diversificate a stimoli particolari, bisogna avere più strati. Anche molti strati (deep-learning). In questo caso la libreria Pytorch prevede come standard l'**addestramento a batch** (o meglio a **mini-batch**). In sostanza si tratta di applicare una serie di esempi tutti insieme, ovvero a lotti. Ad ogni esempio viene retro-propagato il gradiente per tutti i parametri, ma essi non vengono aggiustati. Si sfrutta la predisposizione all'accumulo dei gradienti in Pytorch. Solo alla fine del lotto di esempi viene lanciato l'aggiornamento dei parametri. Ecco

perché all'inizio o alla fine del lotto, dopo l'aggiornamento, va attivato l'azzeramento dei gradienti (*modulo.zero_grad()*). Questo tipo di addestramento è più efficace ed è lo standard per Pytorch.

Questo significa che **i dati di input e quelli di uscita desiderata hanno sempre una dimensione in più** (quella che elenca gli esempi del lotto).

Per esempio se l'input di una rete è ad una dimensione (vettore), deve diventare a 2 dimensioni. Per diminuire o aumentare le dimensioni di un tensore al volo, si possono utilizzare le funzioni **squeeze** e **unsqueeze**. La proprietà *shape* di un tensore restituisce le sue dimensioni.

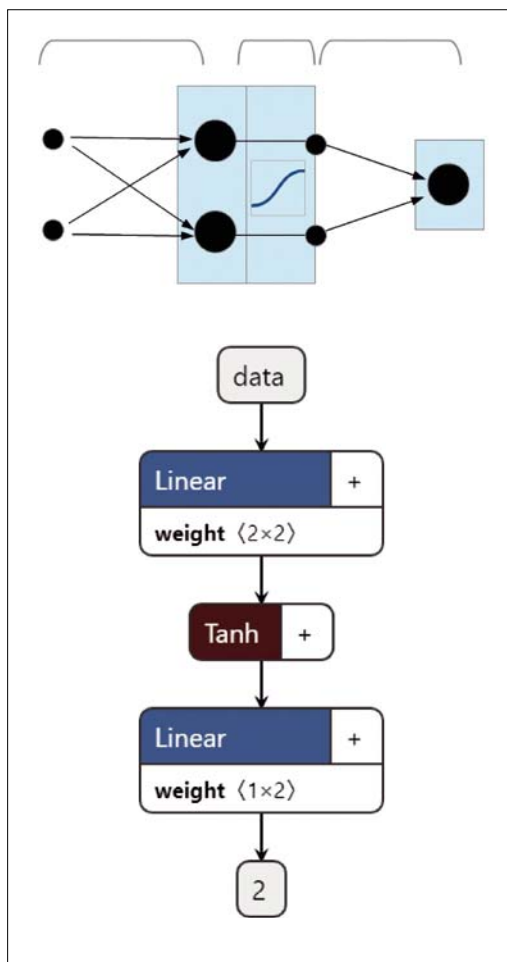
```
>>> ta = th.rand(4)
>>> print(ta.shape, ta)
torch.Size([4]) tensor([0.1716, 0.0211, 0.1625, 0.7430])
>>> tb = ta.unsqueeze(0)
>>> print(tb.shape, tb)
torch.Size([1, 4]) tensor([[0.1716, 0.0211, 0.1625, 0.7430]])
>>> tb = tb.squeeze()
>>> print(tb.shape, tb)
torch.Size([4]) tensor([0.1716, 0.0211, 0.1625, 0.7430])
```

RETE MULTISTRATO (MULTI-LAYER)

Una rete neurale multistrato, può essere definita all'interno di un *Module* che, come abbiamo già visto, può fare da contenitore per altri moduli; allo scopo è sufficiente elencare sequenzialmente i moduli, che quindi corrisponderanno agli strati della rete. Ciò si può ottenere definendo una sotto-classe di **Module** o, più semplicemente, utilizzando un modulo predisposto allo scopo, che si chiama **Sequential**. Qui di seguito proponiamo un esempio di rete multistrato definita secondo quanto appena spiegato e con l'utilizzo di *Sequential*.



Fig. 5
Schema della rete
(in alto) e come
viene visualizzata
da Netron
(in basso).

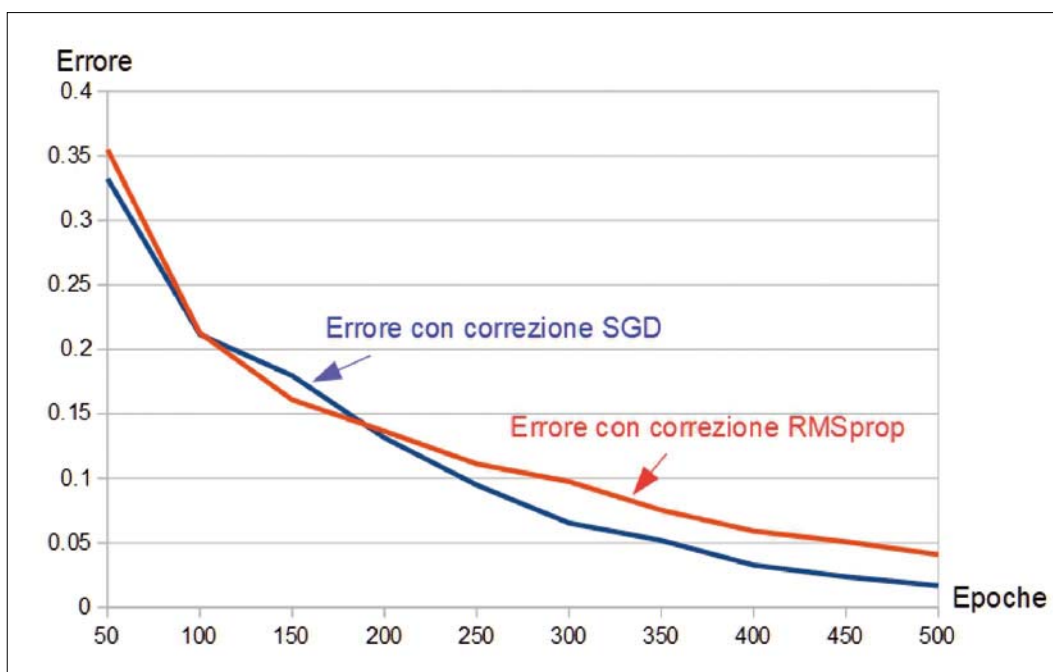


```
class Xornet(nn.Module):
    def __init__(self):
        super(Xornet, self).__init__()
        self.L0 = nn.Linear(2, 2, bias=False)
        self.L1 = nn.Linear(2, 1, bias=False)
    def forward(self, x):
        y = self.L0(x)
        y = nn.functional.tanh(y)
        y = self.L1(y)
        return y
xornet = Xornet()

#oppure più semplicemente:
xornet = nn.Sequential(
    nn.Linear(2, 2, bias=False),
    nn.Tanh(),
    nn.Linear(2, 1, bias=False)
)
```

La rete definita nell'esempio è la classica rete per la simulazione della funzione XOR ed è composta da due strati con l'interposizione della funzione di attivazione Tangente iperbolica (Tanh). Una volta salvata su file la sua struttura può essere visualizzata da un software grafico open source on-line reperibile all'indirizzo Internet <https://netron.app/>. Lo stesso può essere scaricato anche dall'indirizzo <https://github.com/lutzroeder/netron>. Netron non legge solo i file rete Pytorch ma anche

Fig. 6
Addestramento
della rete xornet.



▼ Listato 2

```
def dataSet(batch):  
    xs = th.tensor([[0., 0.], [0., 1.], [1., 0.], [1., 1.]]) # valori di input  
    ytrg = th.tensor([[0.], [1.], [1.], [0.]]) # corrispondenti valori di output corretti  
    databatch = [] # lista python degli esempi di un lotto  
    for i in range(0, batch):  
        r = random.randint(0, 3) # carica la lista con "batch" valori  
        # valori casuali fra i possibili 4  
        databatch.append((xs[r], ytrg[r])) # carica la lista con la coppia x, y (tensori)  
    return databatch # restituisce la lista con le coppie di valori
```

altri formati come lo standard ONNX, Tensorflow ecc. Nella **Fig. 5** vedete come la rete viene visualizzata da Netron. A questo punto potrebbe venire da domandarsi quanti sono i parametri di una rete neurale; ebbene, possiamo saperlo facilmente utilizzando questo loop:

```
params = sum(p.numel())  
for p in xornet.parameters():  
    if p.requires_grad:
```

In questo caso il risultato sarà 6.

Nel caso volessimo utilizzare la definizione più dettagliata usando la sottoclasse di Module abbiamo bisogno di definire due funzioni:

- `__init__` che chiama il super iniziatore e poi definisce come elementi della classe i vari layer
- `forward` che dettaglia la sequenzialità delle operazioni per l'esecuzione della rete.

Attenzione, però, perché la funzione `forward` non va mai chiamata direttamente: sarà infatti eseguita automaticamente, insieme ad altre operazioni, con l'esecuzione della rete:

```
out = xornet(x)
```

Vediamo adesso di spiegare come potrebbe essere svolta la procedura di addestramento per questa nostra semplicissima rete neurale d'esempio.

Ebbene, siccome utilizzeremo un addestramento a lotti (modalità mini-batch) dobbiamo definire una funzione che ci restituisca una lista in Python degli esempi per ciascuno dei lotti. Tale funzione è proposta nel **Listato 2**. A questo punto andiamo a creare il "loop" per implementare un certo numero di ripetizioni (ossia di epoche), come proposto nel **Listato 3**. L'andamento dell'addestramento si può vedere in **Fig. 6**. Questo è il funzionamento della rete alla fine dell'addestramento `y= xornet(x)`:

```
(0 0) -> 0.00 rounded 0 correct: 0  
(0 1) -> 0.87 rounded 1 correct: 1  
(1 0) -> 0.88 rounded 1 correct: 1  
(1 1) -> 0.16 rounded 0 correct: 0
```

In Pytorch, il tipo di correzione può essere scelto fra diversi tipi predisposti e comuni nel settore

▼ Listato 3

```
lossXor = nn.MSELoss() # tipo di calcolo dell'errore : errore quadratico medio  
# tipi alternativi di aggiustamento dei parametri in base al gradiente  
optim = th.optim.SGD(xornet.parameters(), lr=0.01, momentum=0.01)  
# oppure:  
# optim = th.optim.RMSprop(xornet.parameters(), lr=0.01, alpha=0.99)  
epochs = 500 # numero di ripetizioni  
batch = 16 # numero di esempi per lotto (batch)  
mloss = 0 # azzera accumulo errore  
# (indicatore da visualizzare per controllo addestramento)  
step = 50 # step di visualizzazione  
for i in range(epochs+1):  
    for x, yt in dataSet(batch): # estrae gli esempi dal lotto  
        y = xornet(x) # esegue la rete  
        loss = lossXor(y, yt) # calcola errore  
        mloss += loss.item() # accumula errore  
        loss.backward() # back-propagation (accumulo gradiente)  
    optim.step() # aggiusta i parametri a fine lotto utilizzando i gradienti accumulati  
    xornet.zero_grad() # azzera gradienti accumulati (pronti per un nuovo lotto)  
    if (i % step == 0) and (i != 0): # stampa periodica di controllo  
        print("Epoch: {:4d} Loss: {:.6f} ".format(i, mloss/(batch*step)))  
        mloss = 0
```

Benché l'intelligenza artificiale "connessionista" sia nata con l'intento di simulare i processi cerebrali ed in contrapposizione con l'intelligenza artificiale classica (basata su logica programmata), il successo delle reti ad addestramento supervisionato, basato sulla "back-propagation", ha finito per far dimenticare le assunzioni delle origini: avvicinarsi quanto più possibile alle strutture cerebrali. Il risultato è stato, insieme agli indiscutibili successi, un proliferare di potenti sistemi di calcolo matriciale dal consumo di centinaia di watt. Si stima che il cervello umano con miliardi di neuroni e decine di miliardi di connessioni consumi circa 20 W. Una struttura di rete ad addestramento supervisionato attuale che sia equivalente in termini di nodi e di link consumerebbe probabilmente una decina di MW.

Ma non è solo un problema di energia, è soprattutto un problema di struttura. Noi oggi simuliamo di tutto, anche le attività intelligenti, con computer che adottano la tradizionale architettura von Neumann definita nel secolo scorso. In questa architettura i dati vengono elaborati secondo un "clock" centrale. Anche utilizzando strutture parallele, la sincronizzazione è comunque centralizzata. E' possibile simulare architetture "data driven", dove ogni nodo si attiva e produce output nel momento in cui riceve gli input, ma probabilmente le migliori performance si otterranno creando strutture veramente distribuite.

Ma veniamo all'aspetto principale: il modello.

Le reti supervisionate sono sostanzialmente degli algoritmi che estraggono e modellano le caratteristiche statistiche di un particolare compito. Assomigliano al funzionamento cerebrale? Neanche un poco. I risultati sembrano coincidere, ma probabilmente perché anche il cervello estrae statistiche; ma, secondo quanto risulta ai neurologi (al momento), non c'è assolutamente traccia di circuiti di back-propagation. Quindi come fa?

Questo è un altro scopo del "Neuromorphing Computing": capire meglio il funzionamento del cervello. Questo risultato non solo

ci permetterebbe di avere sistemi di intelligenza artificiale più efficienti e potenti, ma ci aiuterebbe anche a capire noi stessi. Visti i presupposti, non stupisce quindi che sia l'Unione Europea che gli USA abbiano stanziato ingenti fondi per approfondire le strutture neuromorfiche. L'EU ha dato il via nel 2013 al programma decennale "Human Brain Project", che ha dimensioni enormi in termini di risorse economiche ed umane (<https://www.humanbrainproject.eu/en/>).

Mentre gli USA hanno dato il via nel 2013 ad un progetto con un nome simile "The BRAIN Initiative" che però ha un focus più biologico (<https://braininitiative.nih.gov/>).

Non si sa molto di cosa stia facendo la Cina, ma senz'altro sta esplorando intensamente anche questa strada dell'intelligenza artificiale. Infatti in rete potete trovare alcuni progetti di neuromorphing computing.

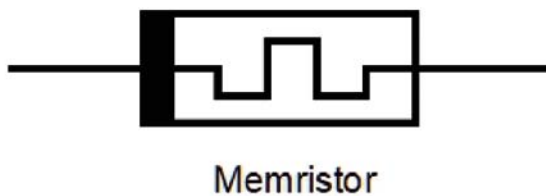
In ogni caso c'è molta carne sul fuoco in tutto il mondo, anche se non ha la stessa rilevanza delle strutture deep-learning che stanno ottenendo molti successi in numerosi campi.

Sono ancora attività di ricerca di base che cercano di affrontare i due aspetti principali di questa tematica:

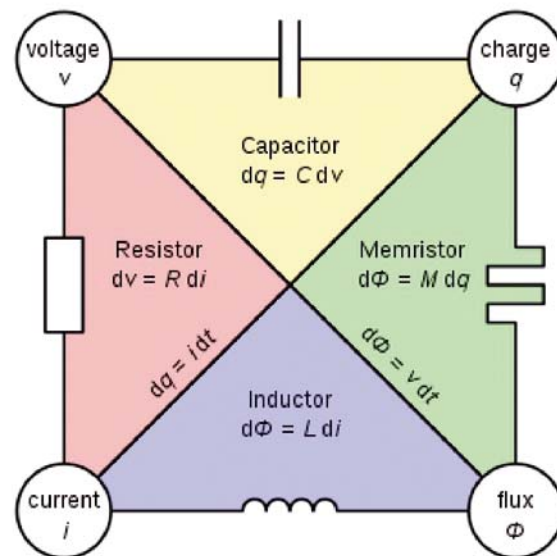
- l'hardware da utilizzare, digitale o analogico;
- la modalità dell'apprendimento.

Stabilito che le strutture devono essere asincrone, o se volete "data driven", come simulare i neuroni? Possiamo utilizzare batterie di microprocessori (anche minimali) per simulare uno o più neuroni. Oppure possiamo pensare di tornare all'analogico, per cui l'attivazione del neurone è un valore dato direttamente dalla composizione degli input. Ed in quest'ultimo caso, utilizziamo, per esempio, un transistor mosfet?

Oppure passiamo a componenti più innovativi e di pochissimo assorbimento come i "memristor"? Un memristor è un componente passivo che memorizza la sua resistenza (modificabile).



 **Fig. A**
Simbolo del memristor e sua simmetria con gli altri componenti passivi.



Può essere realizzato con tecnologie diverse ma sembra l'ideale per simulare le sinapsi.

In realtà l'argomento più impegnativo è la modalità di apprendimento. Le reti deep-learning si basano sulla modalità back-propagation, ma questa non è esclusiva delle reti supervisionate con esempi esterni. Ci sono strutture come le reti competitive generative "Generative adversarial network (GAN)", o le reti "Autoencoder" che imparano dalla loro stessa produzione, ma sempre tramite back-propagation. Poi c'è tutta la tematica del "Reinforcement Learning" che può usare la back-propagation in modo parziale. Ma come abbiamo detto non c'è niente di simile alla back-propagation in biologia.

Come addestrare allora le sinapsi? Prima della back-propagation, si sono immaginate altre modalità di automodifica delle sinapsi. Per esempio secondo la regola di Hebb (psicologo 1950): «se un neurone A è abbastanza vicino ad un neurone B da contribuire ripetutamente e in maniera duratura alla sua eccitazione, allora ha luogo in entrambi i neuroni un processo di crescita o di cambiamento metabolico tale per cui l'efficacia di A nell'eccitare B viene accresciuta»

Questa regola applicata ai neuroni artificiali non ha portato a significative strutture capaci di imparare compiti complessi ma solo ad alcuni esempi di memorie associative come quelle proposte da Hopfield (fisico 1982).

Un'altra struttura interessante che non utilizza la back-propagation è quella delle "Mappe di Kohonen" o "self-organizing map (SOM)" (1983). Questa struttura simula aree cerebrali in cui i neuroni sono connessi anche con i loro vicini.

Quelli più vicini hanno connessioni eccitatorie, mentre quelli più lontani hanno connessioni inibitorie (secondo una distribuzione detta a cappello messicano). Questa situazione fa sì che ci sia un neurone che vince su tutti, quello con le connessioni di entrata più simili all'input. Con la simulazione, in realtà le

connessioni laterali non sono implementate e si fa vincere semplicemente il nodo con i pesi più simili all'input. L'aggiornamento avviene aumentando la somiglianza dei pesi con l'input per il vincitore e per i suoi vicini prossimi. Il risultato è una distribuzione dei nodi in un piano, con specializzazione dei nodi a rispondere ad input particolari ma con una distribuzione topologica.

Le reti con nodi Radial Basis Function (RBF), sono simili alle SOM in quanto i nodi si specializzano a rispondere a particolari input, ma non forniscono una informazione topologica in quanto non hanno una struttura geometrica. Ambedue sono comunque tipi di reti **non supervisionate**. Ovvero si auto-organizzano in base ad alcune caratteristiche del solo input.

Esistono già dei prodotti commerciali con struttura neuromorfica? INTEL ha recentemente annunciato il chip *Loihi* con 128 core collegati in un reticolo bidimensionale (maggiori dettagli sono su www.intel.com/content/www/us/en/research/neuromorphic-computing.html).

Ogni core può simulare fino a 8198 neuroni e può gestire fino a 128K pesi. È facilmente scalabile a migliaia di core e può gestire la comunicazione fra i neuroni in modo asincrono (SNN: spike neural network).

Ovvero i neuroni "sparano" il segnale verso gli altri, quando sono pronti, per esempio quando è superata una soglia.

Si tratta in realtà di una struttura parallela completamente programmabile anche nel comportamento dei singoli neuroni e nel tipo di apprendimento. Si tratta quindi di un approccio digitale al "neuromorphing computing".

Anche IBM ha un chip simile: *TrueNorth*.

<https://research.ibm.com/articles/brain-chip.shtml>.

Si tratta sempre di una struttura digitale parallela a basso consumo e scalabile. Fa parte del più generale progetto SyNAPSE per lo studio di computer neuromorfici.

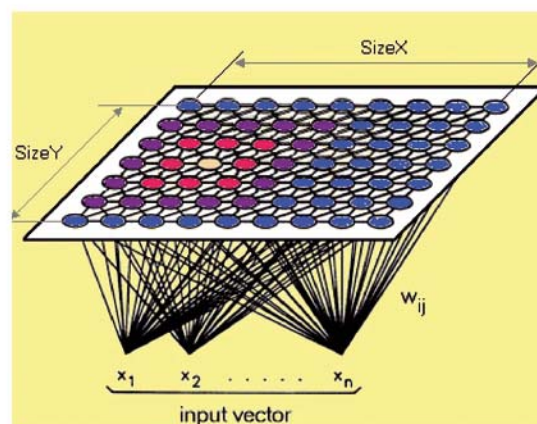
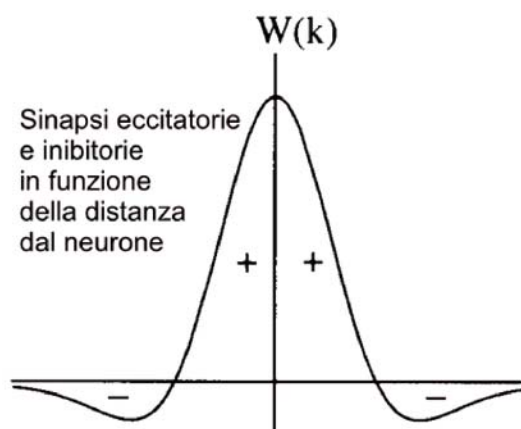


Fig. B - Sinapsi laterali in alcune aree cerebrali e simulazione del loro comportamento con SOM

Tabella 1
Comandi base Pytorch.

Creazione di un tensore monodimensionale: `x = torch.tensor([n0, n1, ...])`

Creazione di un tensore bidimensionale: `x = torch.tensor([[n00, n01, ...], [n10, n11, ...], ...])`

Ma si possono creare anche tensori con valore 0 o con valori uguali qualunque, o vuoti

Visualizzazione della forma di un tensore : `print(x.shape)`

Un tensore può essere derivato da un array numpy : `x = torch.from_numpy(...)` (viceversa: `n = x.numpy()`)

Modifica della struttura: `x = torch.squeeze(x)` riduce di una dimensione , `x = torch.unsqueeze(x)` aumenta dimensione

Tipi di strati (layer) :

- Lineare : `L = torch.nn.Linear(x, y, ...)` (dimensione input = x, dimensione output = y)
- Convolutivo : `L = torch.nn.Conv2d(x, y, kernel, ...)`
- Polling : `L = torch.nn.MaxPool2d(kernel, ...)`

Al layer può essere aggiunta una funzione di attivazione:

- Sigmoidale : `F = torch.nn.Sigmoid()`
- Tangente iperbolica : `F = torch.nn.Tanh()`
- ReLU : `F = torch.nn.ReLU()`

Composizione di strati (rete complessiva):

a) Mediante la definizione del modulo sequenziale : `NN = torch.nn.Sequential( ..., ..., ...)`

b) Mediante la definizione di una sottoclasse di Module :

```
class Miaclasse(torch.nn.Module):  
    .....  
    NN = Miaclasse()
```

Calcolo dell'errore nell'addestramento (loss function):

- Errore quadratico medio : `lf = torch.nn.MSELoss()`
- Cross entropia : `lf = torch.nn.CrossEntropyLoss()`

Aggiustamento dei pesi (funzioni di ottimizzazione):

- Algoritmo della discesa stocastica del gradiente (con momento): `opt = torch.optim.SGD(...)`
- Algoritmo RMSprop: `opt = torch.optim.RMSprop(...)`
- Algoritmo Adam: `opt = torch.optim.Adam(...)`

Azzeramento dei gradienti accumulati in tutta la rete NN: `NN.zero_grad()`

dell'addestramento supervisionato. Questi tipi si trovano nel modulo *torch.optim*. Nell'esempio precedente abbiamo utilizzato quello più comune che è SGD. Ma abbiamo anche provato l'addestramento con il metodo di correzione RMSprop. In ambedue, il parametro *lr* definisce il tasso di modifica (learning rate). Quelle utilizzate e descritte sono solo una piccola parte delle numerosissime funzionalità di PyTorch, di cui nella **Tabella 1** riportiamo i comandi principali e la loro spiegazione. Speriamo, in futuro, di proporre ulteriori esempi di utilizzo ed ulteriori approfondimenti riguardanti Pytorch. □



Cosa occorre?

La board ideale per sperimentare con PyTorch è Raspberry Pi. Raspberry Pi 4 Tipo B con 4GB di memoria (cod. RPI4-4GB) è disponibile da Futura Elettronica a Euro 72,00, la stessa versione con 8GB di memoria (cod. RPI4-8GB) è in vendita a Euro 99,00. Per Raspberry Pi 4 Tipo B sono disponibili anche diversi starter kit già pronti all'utilizzo disponibili alla pagina bit.ly/RPIStarterKit. 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>





CONOSCERE E USARE

di FRANCESCO FICILI

6

Node-RED

Impariamo ad usare Node-RED, un tool di flow-based programming orientato all'IoT ed alla connettività, originariamente sviluppato dall'IBM Emerging Technology Services team e adesso parte della JS Foundation. In questa puntata continuiamo a parlare di connettività introducendo i websockets e il protocollo MQTT. Sesta Puntata.

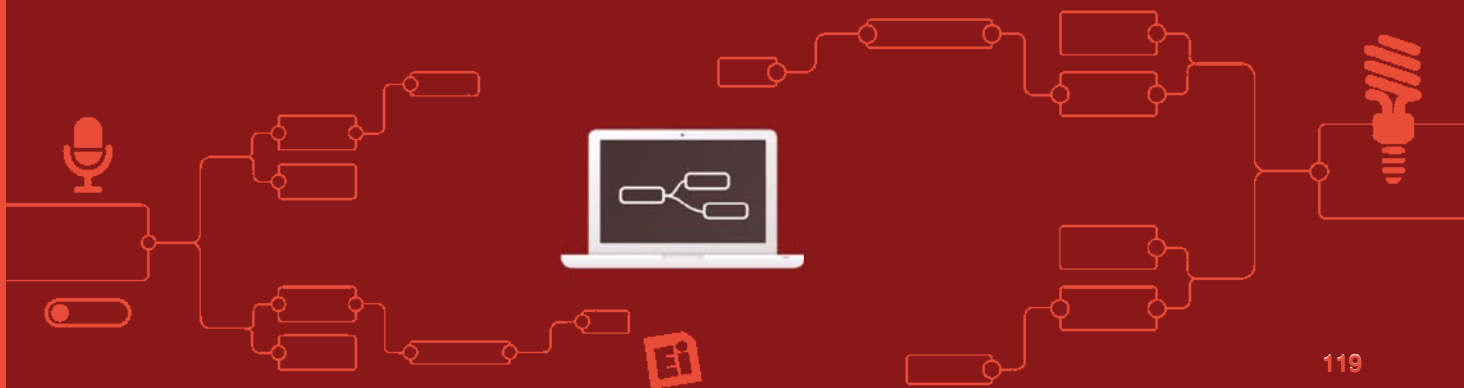
Come abbiamo potuto vedere nella scorsa puntata, Node-RED è un tool con una forte predisposizione allo sviluppo di applicazioni connesse. Infatti, tra i nodi presenti di default esiste un'ampia sezione dedicata alla connettività, identificata dal gruppo di nodi Network. Nella scorsa puntata ci siamo addentrati nello studio dei nodi per la gestione di comunicazioni via UDP, TCP e http. Tuttavia, in tempi più recenti, l'avvento delle applicazioni IoT ha visto sempre di più l'affermazione di tecnologie di comunicazione specializzate, capaci di soddisfare i requisiti specifici e di superare le limitazioni imposte da quella classe di dispositivi che sono al centro del mondo dell'internet degli oggetti, ossia i dispositivi embedded. Tali tecnologie frequentemente continuano ad appoggiarsi su TCP/IP come livello di trasporto, ma offrono una implementazione a livello applicativo (Livello 7 OSI) più consona a quelle che sono le esi-

genze dei dispositivi embedded di quanto non faccia un protocollo come http, che è più pensato per il world wide web e la comunicazione tra browser e server di quanto non lo sia per la comunicazione machine-to-machine.

Ci sono diverse tecnologie di comunicazione che stanno cercando di imporsi nel panorama IoT in questi anni; in particolare, in questa puntata, noi ci concentreremo su due meccanismi: i websockets ed il protocollo MQTT.

WEBSOCKET

Una delle caratteristiche del protocollo http che abbiamo analizzato nella scorsa puntata è quella di ridurre il numero di connessioni aperte a quelle strettamente necessarie. Infatti in una tipica transazione dati http il client invia la richiesta, che viene servita dal server con una appropriata risposta, dopodiché la connessione tra client e server viene



generalmente chiusa e per essere nuovamente aperta il client deve necessariamente inviare una nuova richiesta. Questo comportamento è ideale per il World Wide Web, in cui le pagine dei siti spesso contengono dei collegamenti a pagine ospitate da altri server, diminuendo così il numero di connessioni attive a quelle effettivamente necessarie. Tuttavia questa tipologia di funzionamento cosiddetto connection-less introduce delle latenze nello scambio dati e può risultare meno adatto ad applicazioni in cui la responsività sia importante. Per ovviare a questo limite sono state sviluppate tecnologie alternative, come appunto i Websockets. Il protocollo websocket è una tecnologia web che permette di implementare canali di comunicazione full-duplex usando una singola connessione, che viene mantenuta aperta, tipicamente basata sul protocollo TCP. Questa tecnologia facilita l'interazione tra client e server, facilitando la realizzazione di applicazioni in cui la componente real-time riveste una certa importanza.

Come anche il protocollo http, il protocollo websocket si posiziona al livello 7 della pila protocollare OSI, quindi è anch'esso un protocollo di livello applicazione.

WEBSOCKET IN NODE-RED

Node-RED offre nativamente due nodi per la gestione della comunicazione tramite websocket, rappresentati in Fig. 1.

- **websocket in:** questo nodo permette di ricevere messaggi tramite protocollo websocket. Una volta che il nodo ha ricevuto il messaggio lo invia su *msg.payload*.
- **websocket out:** questo nodo permette di inviare messaggi tramite protocollo websocket. Il nodo invia il nodo contenuto in *msg.payload* di qualsiasi messaggio in ricezione.

UN ESEMPIO DI UTILIZZO DEI WEBSOCKET

Vediamo adesso un semplice esempio di utilizzo del protocollo websocket con Node-RED. Ciò che ci proponiamo di realizzare è una semplice comunicazione machine-to-machine usando il protocollo websocket. Come fatto in precedenza per gli esempi con TCP ed UDP, useremo una Raspberry Pi e il nostro PC per ospitare le due istanze di Node-RED che dovranno comunicare via websocket. Oltre alla Raspberry Pi e al PC ci occorrerà anche un server esterno che faccia da mediatore nello scambio dati. A questo server la Raspberry Pi si collegherà per inviare i dati, mentre il PC si metterà in

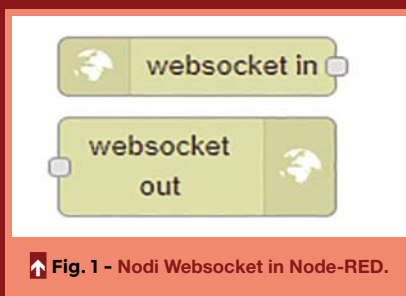


Fig. 1 - Nodi Websocket in Node-RED.

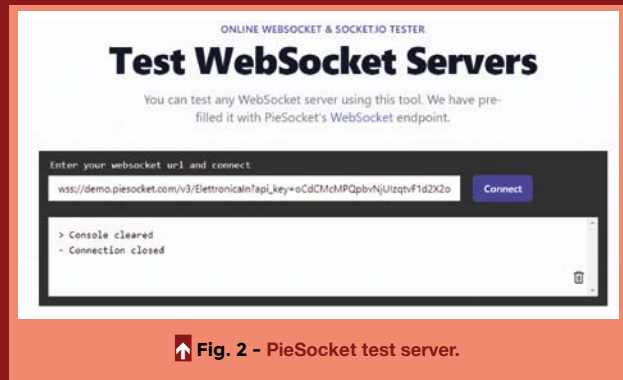


Fig. 2 - PieSocket test server.

ascolto per riceverli. Come server di test abbiamo deciso di utilizzare un servizio offerto da PieSocket, che è possibile trovare a questo indirizzo: <https://www.piesocket.com/websocket-tester#>.

In Fig. 2 è possibile vedere la schermata in cui è rappresentata la console di controllo del server di test. Nella barra della URL, dopo la parte fissa *www://demo.piesocket.com/v3/* è possibile inserire un path interno che verrà utilizzato per lo scambio dati (nel nostro esempio */Eletronicaln*), e successivamente è presente l'hash della API key utilizzata per la connessione. Copiate l'intera stringa e salvatela da qualche parte, ci servirà in seguito.

Partiamo dalla creazione del flow sulla Raspberry Pi, ossia quello che trasmette il dato. Preleviamo dalla palette il solito nodo *Inject* ed un nodo *Websocket out* e posizioniamoli sul workspace. Possiamo configurare il nodo inject per inviare una qualsiasi stringa (nel nostro esempio la stringa "Ciao da Raspberry!!!"). Dopo aver fatto questa semplice operazione apriamo la finestra delle proprietà del nodo websocket out, che possiamo vedere rappresentata in Fig. 3.

A questo selezioniamo dal dropdown menu Type la voce "Connect To" e clicchiamo sull'icona a forma di matita per configurare il websocket client. La configurazione del websocket client è riportata in Fig. 4.

Qui è sufficiente copiare nel campo URL la stringa precedentemente salvata (che contiene l'indirizzo del server, il path interno e l'API key). Salviamo e chiudiamo. Per evitare che il nodo venga rappresentato con un'icona eccessivamente lunga nel workspace possiamo anche dare un nome al nodo websocket out (nell'esempio lo abbiamo rinominato *Websocket Test*). A questo punto non resta altro da fare che connettere l'uscita del nodo inject all'ingresso del nodo websocket out, e il flow del nostro nodo trasmettitore è completo.

Passiamo adesso al nodo ricevitore, implementato sul PC (ovviamente sarebbe anche possibile sostituire questo nodo con una seconda Raspberry Pi). I nodi che ci occorrono in questo flow sono un nodo websocket in ed un nodo debug. La configurazione del nodo websocket in (Fig. 5) è del tutto simile a quella dal nodo websocket out, con la differenza che

questa volta il campo type dovrà essere settato al valore "Listen on", in quanto il nodo è in ascolto.

Anche in questo caso andrà creato un configuration node (per il listener questa volta), la cui finestra delle proprietà è riportata in Fig. 6.

A questo punto non ci resta che collegare l'output del nodo websocket in al nodo debug e anche il flow del nodo ricevitore è completo. Ora possiamo testare la nostra implementazione, cliccando sul pulsante integrato del nodo inject per inviare il contenuto del suo campo payload dal nodo trasmettitore al nodo ricevitore via websocket. Se abbiamo fatto tutto correttamente, dal lato del ricevitore vedremo comparire sulla finestra di debug le stringhe rappresentate in Fig. 7.

MQTT

Il protocollo MQTT (Message Queue Telemetry Transport) è un protocollo di messaggistica, originariamente creato da Andy Stanford-Clark e Arlen Nipper nel 1999. MQTT è un protocollo estremamente leggero e che utilizza un pattern di tipo pubblicazione-sottoscrizione (publish-subscribe o PubSub), ossia non esiste una connessione diretta tra i nodi che si scambiano un messaggio, ma piuttosto esistono una serie di nodi client che pubblicano su o richiedono dati da

una terza parte, denominata normalmente broker.

Come http e websocket, MQTT è un protocollo di livello applicazione e sfrutta un livello trasporto basato su TCP/IP. Per via delle sue caratteristiche, il protocollo MQTT è molto utilizzato in ambito IoT, in quanto è molto leggero e veloce, e quindi adatto ad essere implementato anche in sistemi embedded con risorse limitate, ancorché su smartphone e server. Ne è stata sviluppata anche una variante, denominata MQTT-SN, destinata a sistemi embedded che per il networking non fanno uso di un livello trasporto basato su TCP/IP, ma su protocolli alternativi come 802.15.4 o ZigBee. Recentemente alcune importanti aziende hanno affermato di fare uso di MQTT nelle loro implementazioni, ed uno degli esempi più rilevanti è costituito da Facebook Messenger, che ha fatto uso di MQTT per aumentare le performance in termini di velocità e occupazione di banda.

IL PATTERN PUBLISH-SUBSCRIBE

Come abbiamo già accennato in precedenza, il protocollo MQTT è basato su un modello protocollare di tipo pubblicazione-sottoscrizione. In sostanza vengono definite due entità della rete di telecomunicazione: un certo numero di client e un message broker. I client sono tipicamente i dispositivi terminali della rete, ossia i reali produttori e consumatori dei

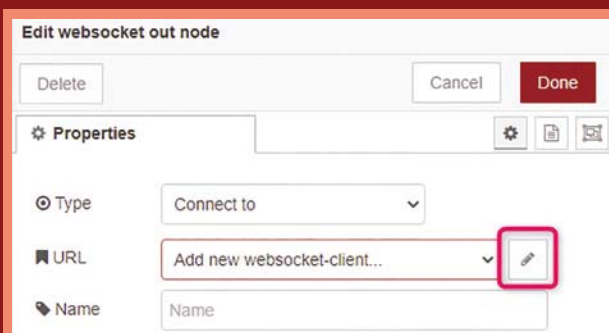


Fig. 3 - Finestra proprietà del nodo websocket out.

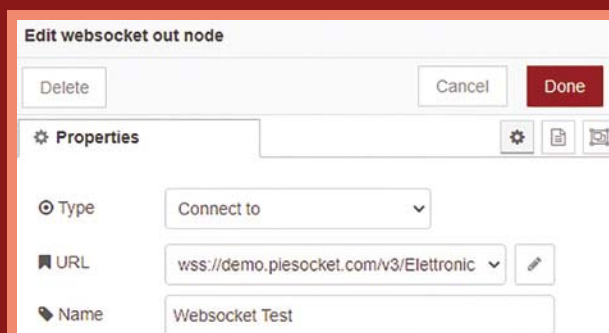


Fig. 5 - Configurazione del nodo websocket in.

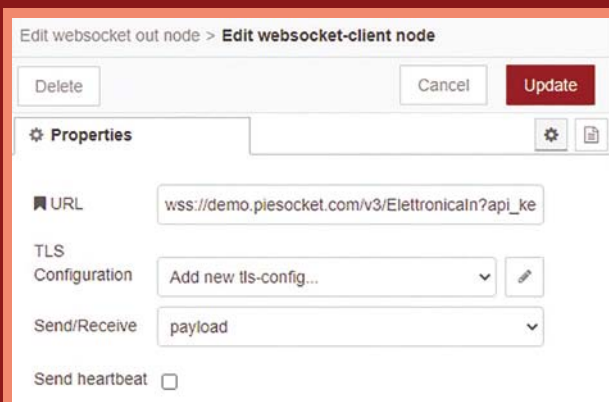


Fig. 4 - Configurazione del websocket client node.

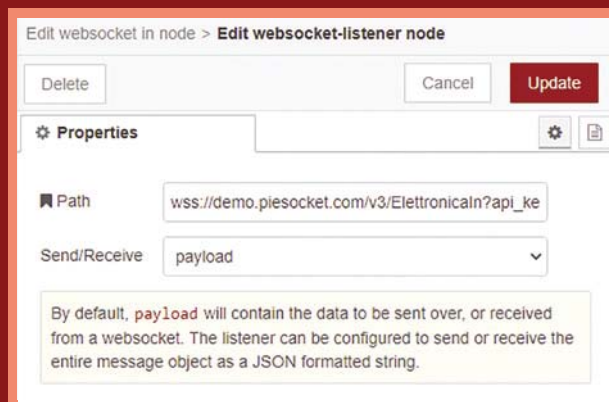
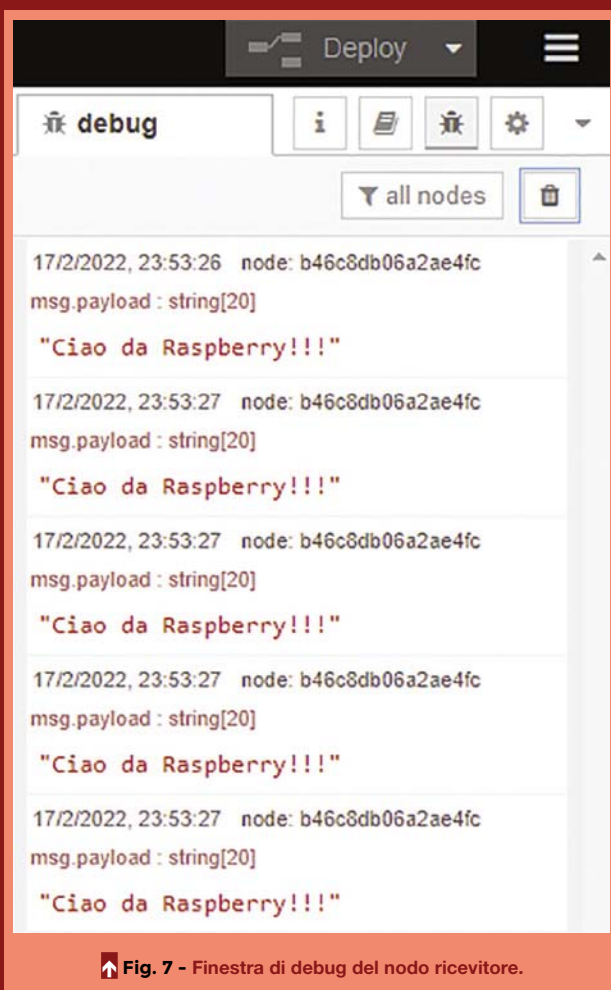


Fig. 6 - Finestra proprietà del configuration node websocket-listener.

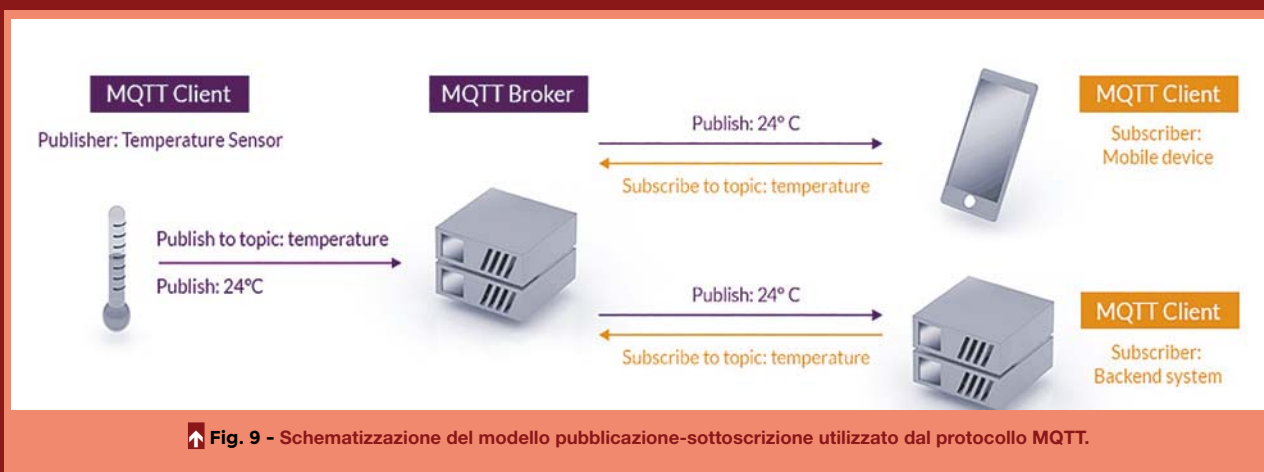


messaggi (tipicamente nodi contenenti sensori o attuatori), mentre il broker non è altro che un server dedicato che riceve tutti i messaggi dai vari client che intendono comunicare e li inoltra ai vari client che invece sono interessati a riceverli. Quindi, usando il protocollo MQTT, se un ipotetico nodo A volesse comunicare un messaggio su uno specifico argomento (comunemente detto topic) ad un ipotetico nodo B, non sarebbe necessario stabilire una connessione diretta e sincrona tra i due nodi (come avviene invece in http tra client e server), ma il nodo A pubblica semplicemente il messaggio con il topic specifico (ad esempio la lettura di un determinato sensore) sul message broker (che ha il compito specifico di ricevere tali messaggi) e sarà poi il broker, in maniera asincrona rispetto al client che trasmette, ad inviare lo specifico messaggio al client interessato (in questo caso il nodo B). Si intuisce anche che il nodo interessato (ossia sottoscrittore) ad uno specifico topic non debba essere necessariamente un client singolo, ma il broker potrebbe inoltrare il messaggio anche ad un gruppo di nodi sottoscrittori lo stesso topic (quindi è nativamente supportato anche il broadcasting). La Fig. 9 schematizza il modello di comunicazione publisher-subscriber.

PROTOCOLLO MQTT

Il protocollo MQTT è stato sin da subito pensato per la comunicazione machine-to-machine, quindi ha una serie di caratteristiche che lo rendono intrinsecamente adatto ad essere utilizzato in ambito IoT. Infatti si tratta di un protocollo estremamente leggero e poco verboso rispetto ad altri protocolli di rete come ad esempio l'http, e quindi è molto indicato per essere utilizzato in dispositivi con poche risorse in termini di memoria, capacità di calcolo e banda. Ciononostante il protocollo è in grado di garantire anche un livello di servizio affidabile, quando quest'ultimo sia una caratteristica significativa dell'applicazione e si sia disposti a sacrificare banda trasmissiva per ottenerla. Infatti in MQTT sono previsti tre livelli di qualità del servizio o QoS:

- **QoS 0 o At most once:** questa è la più bassa qualità del



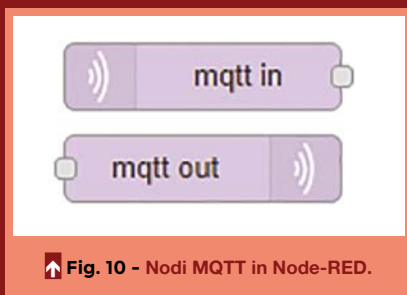


Fig. 10 - Nodi MQTT in Node-RED.

servizio prevista dal protocollo. Il messaggio viene trasmesso una singola volta, senza garanzia di consegna. Il destinatario del messaggio non conferma la ricezione ed il messaggio non è memorizzato e ritrasmesso dal mittente. Questo livello del servizio è spesso denominato “fire and forget” e sostanzialmente offre le stesse garanzie di consegna del protocollo TCP sottostante.

- **QoS 1 o At least once:** questo livello del servizio garantisce la consegna del messaggio almeno una volta al nodo destinatario. Infatti il mittente continua ad inviare il messaggio finché non riceve una conferma di ricezione da parte del destinatario. Tuttavia, con questa metodologia, esiste la possibilità che il messaggio venga ricevuto dal destinatario più di una volta.
- **QoS 2 o Exactly once:** questo è il più alto livello di servizio raggiungibile in MQTT: il messaggio viene consegnato esattamente una volta al destinatario. Tramite uno scambio protocollare più complesso tra nodo trasmettitore e nodo ricevitore, viene confermata la ricezione e c'è la certezza che lo specifico messaggio sia stato ricevuto una sola volta.

Il protocollo dispone inoltre di una serie di messaggi “speciali”; questi messaggi possono essere utilizzati per attuare delle politiche di gestione della rete al verificarsi di determinate condizioni. Essi sono:

- **Birth:** questo messaggio viene pubblicato quando viene stabilita la connessione.
- **Close:** questo messaggio viene pubblicato prima che la connessione venga chiusa normalmente (ad esempio perché un nodo viene ri-deployato o spento).
- **Will:** questo messaggio viene pubblicato su un nodo specifico nel caso in cui un nodo perda inaspettatamente la connessione.

Infine il protocollo MQTT supporta anche un certo livello di security infatti è anche possibile trasmettere username e password tramite i suoi pacchetti ed è possibile cifrare le connessioni tramite l'uso di SSL, sebbene questa caratteristica appesantisca il protocollo.

MQTT IN NODE-RED

Node-RED dispone nativamente di due nodi per la gestione della comunicazione tramite protocollo MQTT, illustrati in Fig.10.

- **mqtt in:** questo nodo permette di connettersi ad un broker MQTT e di effettuare il subscribe su un topic specifico.
- **mqtt out:** questo nodo permette di connettersi ad un broker MQTT e di pubblicare su un topic specifico.

Inoltre è presente un configuration node in cui l'utente può impostare la configurazione del broker.

In particolare è possibile configurare una serie di opzioni per le seguenti categorie:

- **Connection:** permette di configurare l'indirizzo IP del broker, la porta alla quale connettersi, se usare o meno la security ed anche la specifica versione del protocollo da utilizzare.
- **Security:** se si intende utilizzare una connessione sicura, in questa sezione è possibile configurare username e password da utilizzare.
- **Messages:** in questa sezione è possibile configurare i messaggi di birth, close e will (in termini di topic e payload da utilizzare, QoS e politica di retain).

In Fig.11 è riportata a schermata di configurazione del nodo configuration mqtt-broker (tab connection).

MQTT BROKER IN NODE-RED

Come si è potuto intuire dalle sezioni precedenti, per poter far funzionare una applicazione MQTT con Node-RED ci occorre comunque un Broker. Esistono diverse implementazioni di Broker MQTT disponibili ad oggi sul mercato, incluse soluzioni free estremamente performanti. Un esempio abbastanza noto è costituito da Mosquitto, un broker MQTT sviluppato in collaborazione con la eclipse foundation. Può essere scaricato al seguente indirizzo web:

<https://mosquitto.org>

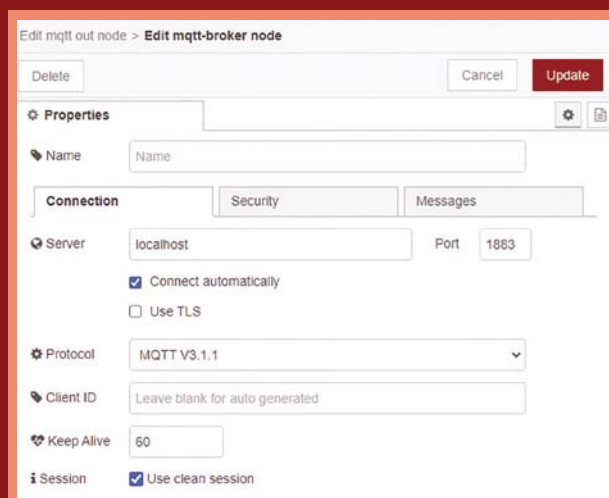


Fig. 11 - Schermata di configurazione del nodo mqtt-broker.



Fig. 12 - Nodo Aedes.

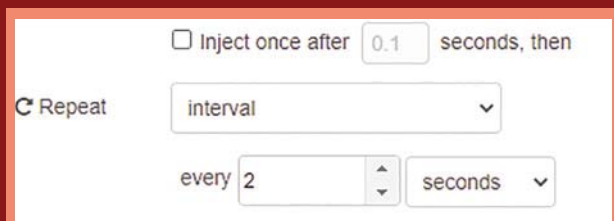


Fig. 13 - Configurazione della periodicità del nodo Inject.

Esiste in diverse versioni per differenti sistemi operativi, come Linux, Mac e Windows. Mosquitto è un broker MQTT estremamente performante ed è sicuramente la soluzione più indicata per implementazioni con molti nodi e con un traffico dati importante. Esistono tuttavia anche implementazioni più "light", sottoforma di nodi Node-RED che possono essere facilmente installate direttamente dalla palette di Node-RED, e possono essere utilizzate nelle situazioni più semplici o anche a scopo di test. Una valida implementazione di questo tipo è costituita da Aedes, un semplice broker MQTT installabile all'interno di Node-RED sottoforma di nodo (Fig. 12). Maggiori informazioni possono essere reperite al seguente indirizzo web:

<https://flows.nodered.org/node/node-red-contrib-aedes>

Per installarlo basta cercare la parola chiave "Aedes" tramite la palette di Node-RED ed installare il nodo. L'utilizzo è semplicissimo, sarà sufficiente trasportare il nodo "Aedes Broker" dalla palette al workspace ed eseguire un deploy, al termine del quale il broker sarà già funzionante.

UN ESEMPIO DI UTILIZZO DI MQTT

Passiamo adesso all'esempio pratico di questa puntata. Ciò che ci proponiamo di realizzare è un semplice sistema di comunicazione basato su MQTT in cui un nodo publisher (che implementeremo con una istanza di Node-RED su PC) invia periodicamente lo stato di un LED and un nodo subscriber che implementeremo con una Raspberry Pi che lo attua utilizzando un GPIO. Questo stato vogliamo che si alterni tra True e False alla frequenza di 1Hz; in sostanza vogliamo far lampeggiare il LED sulla nostra Raspberry Pi via MQTT. Inoltre decidiamo che il broker sarà implementato tramite il nodo Aedes sul nodo subscriber, ossia sulla Raspberry Pi (occorre quindi conoscere l'indirizzo IP di questo dispositivo sulla rete locale). Partiamo dall'implementazione del nodo publisher, che nel nostro esempio sarà implementato trami-

te una istanza di Node-RED basata su PC. Se non ricordate come installare Node-RED su PC (sia esso Linux, Mac o Windows), potete consultare la sezione "running locally" della getting started guide di Node-RED al seguente indirizzo:

<https://nodered.org/docs/getting-started/local>

Avere una installazione locale, se non avete a disposizione molte Raspberry Pi, può essere un ottimo sistema per eseguire i vostri test di connettività, l'importante è che il PC e la Raspberry Pi stiano sulla stessa rete locale. Il nodo subscriber ha il compito di inviare periodicamente il nuovo stato del LED, alla frequenza di 1Hz. Se ricordate, proprio all'inizio del corso Node-RED abbiamo visto come far lampeggiare un LED alla frequenza desiderata, e ciò che ci serve è un nodo *Inject* ed un nodo *Trigger*. Inseriamo sul nostro workspace anche un nodo *mqtt out*, che ci servirà in seguito. Configuriamo il nodo Inject per inviare il suo payload ad un intervallo di 2 secondi, come illustrato in Fig. 13, mentre il nodo trigger va configurato per inviare prima il valore booleano true, aspettare per 1 secondo e poi inviare il valore false, come illustrato in Fig. 14. Ora non ci resta che configurare il nodo mqtt out. Clicchiamo due volte sul nodo stesso per aprire la sua finestra delle proprietà ed impostiamo:

- **Server:** clicchiamo sull'icona a forma di matita per aprire la configurazione del broker. Nel campo Server dobbiamo inserire l'indirizzo IP della Raspberry Pi, che conterrà il broker MQTT. Lasciamo il resto della configurazioni ai valori di default e clicchiamo su Update.
- **Topic:** Led

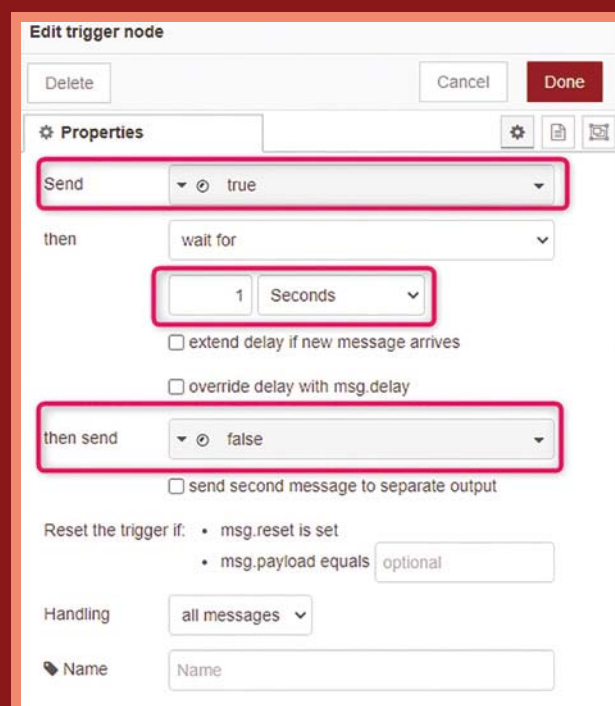


Fig. 14 - Configurazione del nodo trigger.

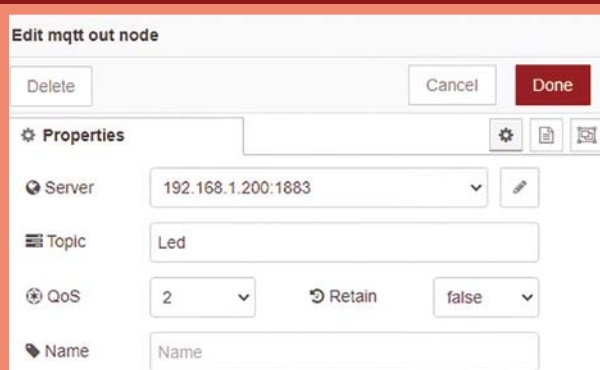


Fig. 15 - Configurazione del nodo mqtt out.



Fig. 16 - Flow che implementa il nodo publisher.

- QoS: 2
- Retain: false

Se tutto è stato fatto correttamente, dovremmo vedere una finestra delle proprietà come quella di Fig. 15 (naturalmente l'indirizzo del server sarà uguale all'indirizzo della Raspberry Pi nella rete locale, nel nostro esempio è 192.168.1.200). A questo punto connettiamo tra di loro i nodi come illustrato in Fig. 16.

L'implementazione del nodo subscriber è ancora più semplice. Ci occorre un nodo Aedes, un nodo mqtt in ed un nodo rpi gpio out. Collochiamo questi tre nodi sul workspace ed iniziamo a configurare il nodo mqtt in. Anche in questo caso va configurato il configuration node mqtt broker, ma stavolta possiamo comodamente inserire "localhost" sul campo

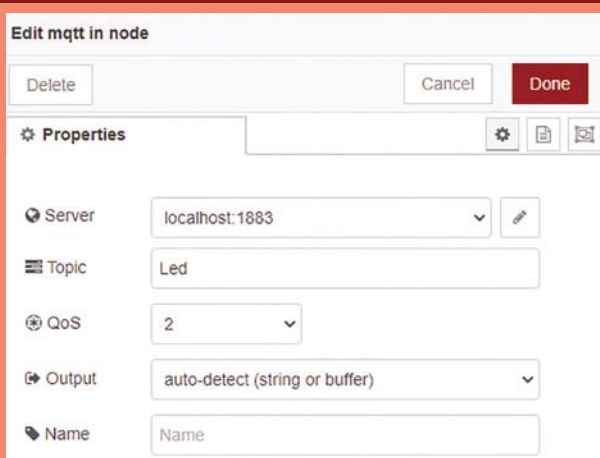


Fig. 17 - Finestra delle proprietà del nodo mqtt in.

server, in quanto il broker MQTT è presente sullo stesso dispositivo (più precisamente sulla stessa istanza di Node-RED). Il resto delle configurazioni sono identiche a quelle viste per il broker del nodo publisher, quindi in questo caso dovrete ritrovarvi con una finestra delle proprietà identica a quella di Fig. 17.

A questo punto passiamo alla configurazione del nodo rpi gpio out; in questo caso vogliamo utilizzare il pin 12 (GPIO 18) per il controllo del LED, quindi impostiamo questo GPIO come digital output ed inizializziamolo a 0 come stato logico, come abbiamo visto nelle prime puntate, in cui abbiamo imparato a gestire il GPIO del Raspberry Pi tramite Node-RED. Il nodo Aedes non ha necessità di particolari configurazioni, ci basterà posizionarlo da qualche parte nel workspace. A questo punto colleghiamo la porta di uscita del nodo mqtt in alla porta di ingresso del nodo rpi gpio out ed eseguiamo il deploy sia del nodo publisher che del nodo subscriber. Una volta che i due nodi saranno in esecuzione vedremo lampeggiare il LED collegato sulla Raspberry Pi alla frequenza impostata.

Come hardware per l'esempio pratico di questa puntata possiamo usare lo shield Raspberry Pi I/O (cod. FT1060), distribuito da Futura Elettronica, una cui immagine è visibile in Fig. 18.

Questo semplice esempio mostra quanto sia agevole utilizzare il protocollo MQTT per la comunicazione machine-to-machine e come quest'ultimo sia una scelta perfetta per le applicazioni IoT basate su Node-RED e Raspberry Pi.

CONCLUSIONI

Siamo arrivati alla conclusione di questa puntata in cui abbiamo visto come gestire connettività via Websockets ed MQTT in Node-RED. Questi due protocolli costituiscono due potenti strumenti per la gestione della comunicazione, permettendoci di scambiare molto agevolmente informazioni tra più dispositivi o tra dispositivi e servers.

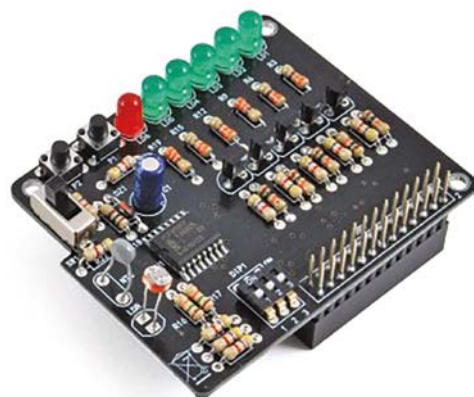


Fig. 18 - Shield Raspberry Pi I/O di Futura Elettronica.

I laser Terahertz compatti sono realtà

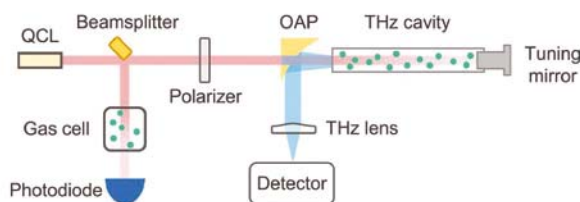
La radiazione terahertz è notoriamente difficile da sfruttare praticamente a causa della sua fisica peculiare, tanto che le sorgenti sintonizzabili nella gamma da 0,3 a 3 terahertz dello spettro elettromagnetico sono rimaste inarrivabili ai ricercatori.

Le sorgenti elettroniche di radiazioni tendono a diventare sempre più deboli man mano che si sale di frequenza, poiché le loro proprietà materiali limitano le prestazioni dei dispositivi quando ci si avvicina alla regione dei terahertz. Quindi, mentre la regione delle microonde è ben apprezzabile, quando si superano i 100 gigahertz (0,1 THz), la potenza crolla. Inoltre le sorgenti di radiazioni ottiche (ad esempio i laser a infrarossi) peggiorano man mano che si diminuisce la frequenza, sempre a causa delle proprietà del materiale.

Alla ricerca di una fonte utilizzabile in questa gamma dello spettro, un team della Harvard John A. Paulson School of Engineering and Applied Sciences (SEAS) ha sviluppato un laser terahertz compatto, operativo a temperatura ambiente e ampiamente sintonizzabile. I ricercatori hanno utilizzato una pompa laser a cascata quantistica (QCL) con un laser molecolare al fluoruro di metile, composto che reagisce fortemente con i campi ottici, assorbendo gli infrarossi ed emettendo radiazioni terahertz. Il QCL funziona anche in una cavità laser molto più compatta rispetto ai precedenti laser ad anidride carbonica. Nei test, il team ha operato su 120 frequenze individuali comprese tra 0,25 e 1,3 THz. Nel prototipo, un wafer di silicio placcato in oro è stato utilizzato come divisore di raggio per riflettere una piccola porzione del raggio della pompa nella cella a gas di riferimento, mentre il resto è entrato nella cavità terahertz.

L'interesse per i laser terahertz pompati otticamente abbraccia quasi 40 anni e negli anni '80 ha lavorato alla realizzazione di laser molecolari compatti, giacché gli esistenti laser erano troppo ingombranti. Esistono anche possibilità nel telerilevamento, perché le molecole hanno firme distintive nella regione dei terahertz che consentono di discriminare specie, tossine, agenti chimici, cose che potrebbero essere una minaccia o semplicemente di monitorare cosa sta succedendo in un impianto chimico. Bisogna però superare il limite che la radiazione terahertz può essere limitata dall'assorbimento del vapore acqueo atmosferico.

www.seas.harvard.edu/news



JET porta la fusione

Il Tokamak Joint European Torus (JET) installato vicino a Oxford, nel Regno Unito, ha recentemente battuto, raddoppiandolo, il record di energia prodotta da fusione nucleare: il mese scorso ha sviluppato un picco di energia pari a 59 megajoule! Il precedente record di energia da fusione nucleare svolto su JET risaliva al 1997 ed aveva portato a 21,7 megajoule. I risultati ottenuti saranno essenziali per calibrare il funzionamento di ITER, il colosso della fusione sperimentale, in via di completamento nel sud della Francia.

Il Tokamak (acronimo russo per "camera toroidale con spire magnetiche") è un reattore per la fusione nucleare sostenuta la cui ideazione risale agli studi Sacharov e Tamm nel 1950. Al suo interno, il plasma generato dalla fusione viene sostenuto a bassa pressione e tenuto lontano dalle pareti, che altrimenti fonderebbero per l'altissima temperatura sviluppata, dell'ordine di milioni di gradi centigradi; il sostentamento è operato da un potente campo elettromagnetico generato da elettromagneti a superconduttori.

JET è il più grande e potente Tokamak operativo al mondo ed è parte di una campagna sperimentale dedicata progettata da EUROfusion per testare oltre due decenni di progressi nella fusione e preparare in modo ottimale l'avvio del progetto internazionale ITER.

Durante l'esperimento, JET ha prodotto un totale di 59 Megajoule di energia termica dalla fusione in un periodo di 5 secondi; la corrispondente potenza media ottenuta dalla fusione (cioè l'energia al secondo) è stata di circa 11

megawatt (59 megajoule diviso 5 secondi). Il precedente record energetico di un esperimento di fusione, ottenuto da JET nel 1997, era di 22 megajoule di energia termica. La potenza di picco di 16 MW raggiunta brevemente nel 1997 non è stata superata, perché l'esperimento di febbraio è stato focalizzato sull'energia di fusione sostenuta.

Il record e i dati scientifici di questi esperimenti cruciali sono un importante impulso

nucleare a 59 MJoule

fusione come mezzo sicuro, efficiente e a basse emissioni di carbonio per affrontare la crisi energetica globale.

Questo risultato è frutto di anni di preparazione da parte del team di ricercatori di EUROfusion in tutta Europa ed ha permesso di acquisire conoscenze riguardo alla fattibilità dell'energia da fusione nucleare sicura. Se è stato possibile mantenere la fusione per cinque secondi, lo si può fare per cinque minuti e poi per cinque ore mentre aumentiamo le nostre operazioni nelle macchine future. I risultati sono il supporto per una decisione precoce per una centrale elettrica DEMO europea, poiché la fusione è necessaria per la decarbonizzazione a lungo termine del nostro approvvigionamento energetico. La fusione, il processo che alimenta stelle come il nostro sole, promette una fonte di elettricità pulita quasi illimitata a lungo termine, utilizzando piccole quantità di materia che possono essere reperite in tutto il mondo a basso costo. Il processo di fusione riunisce atomi di elementi leggeri come l'idrogeno ad alte temperature per formare elio e rilasciare un'enorme energia sotto forma di calore. Fusion è intrinsecamente sicuro in quanto non può avviare un processo di fuga.

L'esperimento di fusione JET permette di creare plasmi che raggiungono temperature di 150 milioni di gradi Celsius, 10 volte più di quanto si raggiunge nel nucleo del sole. JET è l'unico tokamak operativo al mondo in grado di utilizzare la stessa miscela di combustibili deuterio-trizio (DT) prevista per quei dispositivi.

www.euro-fusion.org

per ITER, la versione più ampia e avanzata di JET.

ITER è un progetto di ricerca sulla fusione nucleare sostenuto da scienziati di sette stati membri: Cina, Unione Europea, India, Giappone, Corea del Sud, Russia e USA.

Con l'aumento delle pressioni per affrontare gli effetti dei cambiamenti climatici attraverso la decarbonizzazione della produzione di energia, questo successo rappresenta un importante passo avanti nella tabella di marcia della



Namibia: idrogeno dall'ex roccaforte dei diamanti

In Namibia è in costruzione un gigantesco parco eolico e solare finalizzato alla produzione di idrogeno verde, parte del quale andrà a beneficio della Germania. Nel parco nazionale di Tsau Khaeb, vicino alla città portuale di Lüderitz nel sud-ovest dello stato africano della Namibia, Hyphen Hydrogen Energy, società di sviluppo progetti nella capitale namibiana Windhoek, sta realizzando un progetto insieme ad Enertrag di Daurthal nello stato federale del Brandeburgo per EUR 9,4 miliardi di dollari, per impianti eolici e solari con una potenza installata di 5 GW. Il progetto prevede inoltre elettrolizzatori con una capacità di 3 GW, un impianto di dissalazione dell'acqua di mare e un impianto di produzione di ammoniaca da azoto e idrogeno. Il piccolo porto nella città di Lüderitz verrà ampliato in modo da poter esportare idrogeno e ammoniaca. Quando l'impianto sarà completamente terminato, produrrà 300.000 tonnellate di idrogeno ogni anno, pari a 10 miliardi di kWh di energia. Il progetto è uno dei più importanti che il Namibia Investment Promotion and Development Board (NIPDB) abbia lanciato fino ad oggi. Si stima che 1 kg di idrogeno dalla Namibia alla fine costerà tra 1,50 e 2 euro e se si considera che al momento, 1 kg di idrogeno prodotto dal gas naturale costa quasi 10 euro alla stazione di rifornimento, la convenienza è chiara.

La Germania è urgentemente dipendente dalle importazioni di idrogeno se vuole mantenere la sua promessa alla Commissione europea di far funzionare centrali elettriche a gas naturale almeno in parte con idrogeno a medio termine.

L'ammoniaca verde può essere trasportata in forma liquida ad una pressione di circa 10 bar. Una volta arrivata a destinazione, potrebbe essere bruciata come gas per generare elettricità e calore, separata nuovamente in idrogeno e azoto per immettere l'idrogeno nella rete del gas naturale o utilizzata direttamente per generare elettricità o per produrre fertilizzanti. Oggi, 200 milioni di tonnellate di ammoniaca all'anno vengono trasformate in fertilizzanti; l'idrogeno necessario è prodotto dal gas naturale, mentre l'ammoniaca verde no. L'idrogeno può essere utilizzato anche in forma liquida ad una temperatura di Trasporto a 253 °C senza pressione. Nonostante il miglior isolamento, parte dell'idrogeno si trasforma sempre in gas durante il viaggio, ad esempio dalla Namibia all'Europa. Questo può essere utilizzato per alimentare i motori della nave. Quando arriva a destinazione, l'idrogeno liquido viene riconvertito in gas e distribuito tramite tubazioni. Un rigassificatore è già operativo a Zeebrugge, in Belgio, ma per il gas naturale. Potrebbe essere convertito in idrogeno in pochi anni.

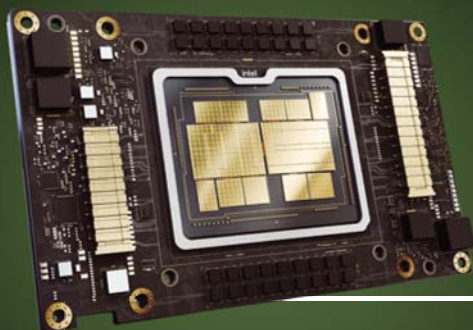
<https://hyphenafrika.com/>

Ponte vecchio, tecnologia nuova da Intel

Intel ha creato il processore su cui sarà basato il supercomputer Aurora, progettato per superare la barriera dell'exaflop: un miliardo di miliardi di operazioni in virgola mobile al secondo. Il processore, chiamato Ponte Vecchio, è un SoC che combina più riquadri di calcolo (realizzati con il processo N5 di TSMC) cache, rete e memoria, o "chiplet". I chip del package sono realizzati mediante diverse tecnologie di processo, nell'ottica dell'integrazione eterogenea. In Ponte Vecchio, Intel ha racchiuso 3.100 millimetri

quadrati di silicio, quasi pari a quattro GPU Nvidia A100, in un ingombro di 2.330 mm². Ogni sistema Ponte Vecchio è in grado di eseguire più di 45 trilioni di operazioni in virgola mobile a 32 bit al secondo (teraflop). Quattro di questi sistemi si adattano a due CPU Sapphire Rapids in un sistema di elaborazione completo. Questi saranno combinati per un totale superiore a 54.000 Ponte Vecchio e 18.000 Sapphire Rapids per formare Aurora, una macchina che punta a 2 exaflop. Ogni processore Ponte Vecchio è in realtà due set di chiplet speculari legati utilizzando la tecnologia di integrazione 2D di Intel Co-EMIB, che forma un ponte di interconnessioni ad alta densità tra due pile 3D di chiplet; il ponte è un pezzo di silicio incorporato nel substrato organico e collega anche la memoria HBM e il chiplet Xe Link I/O al chiplet più grande su cui sono impilati altri.

www.intel.com



Imbarco aereo sicuro? Ci pensa il computer

Un recente studio basato sulle risorse di supercalcolo presso l'Argonne National Laboratory del DOE degli Stati Uniti, ha analizzato e rivisto le procedure di imbarco attuate durante la pandemia da COVID-19 per ridurre il rischio di contagio. Molte compagnie aeree hanno introdotto procedure di imbarco e di volo riviste nel tentativo di ridurre al minimo il contagio tra viaggiatori.

I ricercatori hanno utilizzato la macchina Theta ospitata presso l'ALCF, per valutare e confrontare l'efficacia, i vantaggi e gli svantaggi delle diverse procedure di imbarco, in 16.000 scenari possibili.

È stato riscontrato che un minor numero di posti a sedere incoraggia una maggiore distribuzione dei passeggeri in tutto l'aereo, diminuendo la vicinanza; al contrario, più zone avvicinano i

passeggeri seduti durante il processo di imbarco.

Gli scenari a zona singola hanno ridotto al minimo il rischio di trasmissione del COVID-19 tra i passeggeri e la procedura che ha ridotto notevolmente l'esposizione è stata una politica di posti assegnati combinata con l'imbarco casuale. L'imbarco back-to-front, dimostratosi la politica più rischiosa nella maggior parte degli scenari, ha quasi raddoppiato il tasso di esposizione rispetto all'imbarco casuale dei posti assegnati e ha aumentato l'esposizione di circa il 50% rispetto alle procedure standard di imbarco pre-pandemia. Il computer ha anche concluso che vietare di riporre i bagagli nelle cappelliere può limitare l'esposizione.

www.alcf.anl.gov/news/



Due pale inclinabili per 6,2MW: così è il nuovo eolico off-shore

La tecnologia eolica ha enormi potenzialità ma pone dei limiti, che essenzialmente sono nell'incostanza dei venti e nell'impatto delle installazioni, entrambi mitigati dall'utilizzo dell'eolico off-shore. La società britannica Petrofac ha siglato un accordo con Seawind Ocean Technology, per lo sviluppo e la produzione di rivoluzionarie turbine eoliche galleggianti a due pale da 6,2 MW. Tra le sue caratteristiche principali spicca il design a due pale integrate in una struttura galleggiante in cemento, che permette l'installazione sia in zone di acque profonde che in quelle basse, ma anche in quelle zone con condizioni meteorologiche avverse, grazie alla capacità di resistere a venti fino a 70 m/s (250 km/h) e raffiche fino a 90 m/s (325 km/h). Ogni turbina Seawind 6-126 ha una potenza installata di 6,2 MW, un rotore del diametro di 126 metri capace di ruotare a 20,8 giri/min, un rapporto velocità massima di 137 metri al secondo (m/s) e opera a una profondità oltre 50 m, con velocità di 3,5-25 m/s (12,5-90 km/h). Il rotore a due pale consente di aumentare la stabilità del rotore elevando l'efficienza e il raccordo oscillante che disaccoppia l'albero del rotore, protegge la turbina da carichi che potrebbero comprometterne l'integrità, aiutato anche da un sistema di guida attivo che permetta di sfruttare le alte velocità del vento senza danneggiarne la struttura. A favore del generatore c'è anche il fatto che l'assemblaggio può essere fatto nel porto, quindi non saranno necessarie grandi navi per il suo avviamento in alto mare. Il dispiegamento nelle acque europee è previsto nel primo trimestre 2024.

<https://seawindtechnology.com>

OSCILLOSCOPIO DA LABORATORIO 2 CANALI-100 MHz

cod. DS02C10
€ 298,00



CARATTERISTICHE TECNICHE OSCILLOSCOPIO

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

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



cod. DS02D10
€ 358,00

CARATTERISTICHE TECNICHE GENERATORE FORME D'ONDA

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

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

La TV cambia? Non cambiarla!

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

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



Cod. GS240T2
€ 26,90

**NUOVO
MODELLO**

DVBT/T2
TERRESTRIAL

**H.265
HEVC**

**FULL HD
1080**

USB

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

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



Prezzi IVA inclusa.

Il tuo telecomando TV è rotto?

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

€ 5,90
OGNI PEZZO

**PRONTI
ALL'USO!**



Cod. TVIR-SAMSUNG



Cod. TVIR-SONY



Cod. TVIR-PHILIPS



Cod. TVIR-LG



Cod. TVIR-PANASONIC

DA PROGRAMMARE

Per comandare fino a due dispositivi audio/video.

€ 9,00
Cod. TVFREEDOM2



DISPONIBILE ANCHE 1 in 1
Cod. TVFREEDOM1

€ 8,50

**FUTURA
ELETTRONICA®**

www.futurashop.it

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

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