

Elettronica In

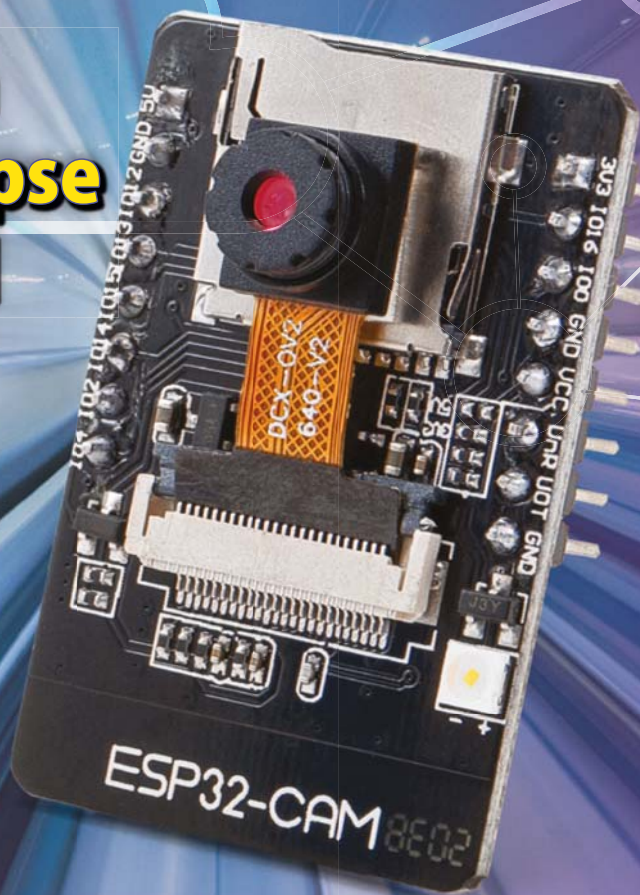
www.elettronica.in.it

€ 6,90 - Anno XXVIII - n. 262 - MARZO 2022



Video in Time Lapse su cloud

- Tracciatore di Bitcoin
- La stampante 3D sotto controllo
- Serratura IoT controllata da smartphone
- Geolocalizzazione meteo IoT su mappa
- Stazione di monitoraggio qualità dell'aria



HUB USB Full Speed isolato



Controller universale per CNC



CONOSCERE
E USARE



Node-RED

CATTURA L'ENERGIA DEL FUTURO

L'energia pulita che ti fa risparmiare!

Pannello solare monocristallino semirigido 12V



cod. SOL20M
€ 32,00

dimensioni:
570 x 350 x 2 mm

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
€ 99,00

15A



cod. SOLAREG10
€ 16,00

10A

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



Carenza

Sentiamo sempre più spesso parlare di carenza (o shortage se preferite) di materiali come conseguenza della pandemia. Tutto è iniziato con la scarsità delle materie prime, per arrivare a quello che, inevitabilmente, la "legge della domanda e dell'offerta" prevede in questi casi: un aumento dei prezzi.

In alcuni casi questi aumenti sono diretti; nel nostro settore abbiamo visto il prezzo dei chip andare alle stelle. In altri casi sono indiretti, come i costi di trasporto che si ripercuotono sui prodotti distribuiti.

La "legge della domanda e dell'offerta" prevede un aumento dei prezzi.

Tra i vari aumenti a cui siamo soggetti, ahimè, c'è anche quello della carta, dovuto a diversi fattori sia diretti che indiretti, anche se una ragione chiara purtroppo non c'è; ma cerchiamo di capire.

Il prezzo della carta sembra essere lievitato essenzialmente a causa di alcuni fattori come la riduzione obbligatoria dell'uso della plastica, che ha comportato un aumento di domanda di carta per il packaging dei prodotti. Il timido ritorno alla normalità ha fatto schizzare la domanda di cosmetici e quindi la richiesta di confezioni, di carta ovviamente. La cellulosa, la materia prima di cui è fatta, è arrivata a costare (fino ad ora) il 70 per cento in più rispetto alla fine del 2020. La luce e il gas, del cui aumento tutti abbiamo un riscontro in bolletta, rappresentano materie prime indispensabili per la realizzazione della carta.

Tutto questo ci ha portato a rivedere il prezzo di distribuzione anche della nostra rivista (invariato da 12 anni). Abbiamo "assorbito" buona parte degli aumenti rivedendo il prezzo di copertina contenendo il rincaro; rincaro applicato ovviamente alla sola rivista cartacea.

Il digitale in questi casi è sicuramente un'ottima soluzione, ma siamo sicuri che il fascino della carta rimarrà in eterno, o almeno fino alla prossima crisi economica.

ON LINE

elettronica.in.it

SI RINGRAZIA

Futura Elettronica
Microchip
Mouser



In copertina:
Realizziamo, con una ESP32CAM, un bellissimo progetto per ottenere dei video in Time Lapse memorizzando le immagini nel Cloud.

Boris Landoni
 boris.landoni@elettronica.in.it

Elettronica In

Rivista mensile, anno XXVIII n. 262

MARZO 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)

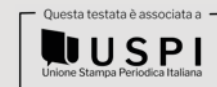
Tel. 02-660301 Fax 02-66030320

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

MOTION TIME LAPSE IN CLOUD



49

Wireless

La stampante 3D sotto controllo

27



19

ARTICOLI

NETWORKING

19 Tracciatore di Bitcoin

Utilizziamo un ESP32 per monitorare l'andamento dei Bitcoin e essere avvisati tramite Telegram di variazioni significative.

WIRELESS

27 La stampante 3D sotto controllo

Approfondiamo la preparazione dell'ambiente di sviluppo per il nostro controller remoto ESP3D. Seconda puntata.

GPS-SYSTEM

39 Meteo Net Map T-Call

Permette di geolocalizzare in un'unica mappa la posizione di più dispositivi IOT connessi in rete tramite WiFi o GSM con la possibilità di visualizzare i parametri ambientali rilevati da ciascuno di essi.

Networking

Tracciatore di Bitcoin



19

GPS-System**Meteo Net Map T-Call****39****Periferiche****HUB USB Full Speed isolato****74****Networking****Smart Gate Lock****105****Top Project****Stazione di monitoraggio qualità dell'aria****63****Hi-Tech****Controller universale per CNC****89****Didattica****Conoscere e usare Node-RED****117****NETWORKING****49 Motion Time Lapse in Cloud**

Basato su ESP32-CAM, utilizza Google Apps Script per automatizzare la registrazione su Cloud senza richiedere supporti di memorizzazione. Prima puntata.

TOP PROJECT**63 Stazione di monitoraggio qualità dell'aria**

Dotata di sensori specifici per gas e polveri, rileva anche grandezze meteorologiche e permette un'analisi approfondita dell'inquinamento atmosferico, trasmettendo su rete LoRa i valori. Seconda e ultima puntata.

PERIFERICHE**74 HUB USB Full Speed isolato**

Basato sull'integrato di Analog Device ADuM4160 permette di proteggere l'interfaccia USB del Personal Computer o di Raspberry Pi mettendo a disposizione quattro porte ad alta velocità isolate.

HI-TECH**89 Controller universale per CNC**

Presentiamo una scheda, basata su un Arduino Nano, completa di tutte le funzionalità hardware e software per pilotare macchine CNC a tre assi.

NETWORKING**105 Smart Gate Lock**

Serratura IoT controllata da smartphone o da assistente vocale Amazon Alexa e Google Assistant. Basato sul modulo WiFi ESP03 e sul server Sinric Pro, si configura tramite interfaccia web.

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

Iniziamo a scoprire 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 ci occupiamo dell'implementazione delle funzionalità di connettività usando Node-RED, spaziando dal TCP all'UDP, fino ad arrivare all'HTTP. Quinta Puntata.



Nuovi FPGA e SoC PolarFire® Low-Density

Metà della Potenza Statica delle Alternative e con Impatto Termico Minimo

I sistemi di Edge computing necessitano di dispositivi programmabili compatti con un basso consumo energetico e un impatto termico ridotto così da eliminare la ventilazione ed altri sistemi di riduzione del calore, ma allo stesso tempo devono fornire una solida potenza di calcolo. Gli FPGA e SoC PolarFire® di Microchip hanno risolto questa sfida riducendo il consumo di potenza statica del 50%.

I miglioramenti effettuati su FPGA PolarFire e SoC PolarFire superano tutti i valori di prestazioni/potenza di qualsiasi alternativa FPGA o FPGA SoC a bassa densità presenti sul mercato, con una struttura FPGA veloce, capacità di elaborazione del segnale, transceiver più capaci e l'unico processore basato su architettura RISC V® application-class rinforzato oggi disponibile con 2 megabyte (MB) di cache L2 e supporto di memoria Low Power DDR4 (LPDDR4).

Superano ogni confronto in termini di potenza, dimensioni del sistema, costi e sicurezza in svariate applicazioni, tra cui smart embedded vision e sistemi automotive, automazione industriale, comunicazioni, difesa e IoT aventi vincoli termici, in cui né la potenza né le prestazioni possono essere oggetto di compromessi.

Aspetti salienti

- Famiglie di prodotti da 25k LE per System On Chip (SoC) e 50k LE per FPGA
- Fattore di forma più piccolo, con package di 11x11 mm
- Transceiver da 12.7G, supporto multiprotocollo 10 Gb
- Potenza statica inferiore, attiva all'accensione
- Maggiore margine termico per una maggiore capacità di elaborazione
- Sicurezza e affidabilità migliori della categoria



microchip.com/lowpowerFPGAs

Il nome e logo Microchip, il logo Microchip e PolarFire sono marchi registrati di Microchip Technology Incorporated negli U.S.A. e in altri Stati. Tutti gli altri marchi appartengono ai rispettivi titolari. © 2022 Microchip Technology Inc. Tutti i diritti riservati. DS00004279A, MEC2405A-ITA-02-22

IPS: molto più di un MOSFET

Da molti anni, ossia da quando la tecnologia costruttiva dei componenti a semiconduttore ha permesso di realizzare MOSFET robusti e in grado di sopportare correnti intense e alte tensioni, i transistor ad effetto di campo a gate isolato sono diventati i dispositivi eletti per la commutazione di carichi elettrici. Proprio la loro diffusione ha fatto nascere dispositivi integrati che in un contenitore TO-220, come quello di un power-MOSFET, concentrano anche il circuito di controllo e una serie di protezioni e driver che normalmente sono implementati come circuiti a parte, per il comando e controllo. Nascono così gli IPS, ossia Intelligent Power Switch, un valido esempio dei quali è l'IPS1021 della IR (International Rectifier): internamente presenta un MOSFET low-side (ossia connesso con il source a massa in modo da funzionare da switch che chiude l'utilizzatore verso massa) con sensore e protezione da sovra-corrente, sovratemperatura, protezione ESD e diodo anti-inversione fra drain e source. Il tutto conferisce elevata affidabilità nel pilotaggio di carichi impegnativi senza richiedere i circuiti di protezione e limitazione

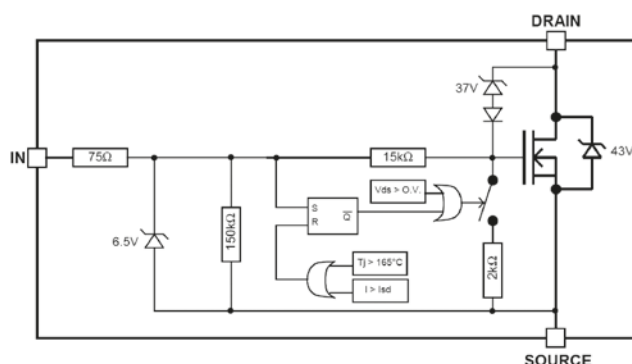
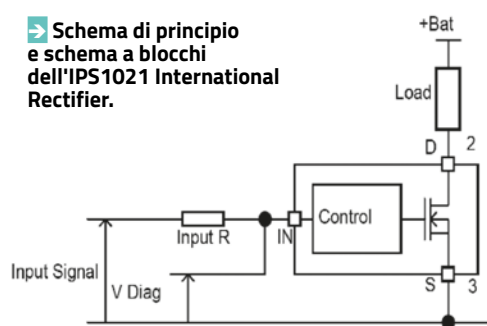
all'esterno, a vantaggio sia della riduzione di spazio sul circuito stampato (sta tutto in un TO220), sia della semplificazione del progetto, giacché le protezioni sono implementate nel componente. La protezione spegne il MOSFET di potenza quando la temperatura supera i 165°C o quando la corrente di scarico raggiunge 35A, a prescindere dalla presenza all'ingresso (il comando del gate, che avviene attraverso una resistenza in serie) del livello logico di comando (tipicamente 5V); una volta intervenuta, grazie a un flip-flop integrato inibisce il dispositivo, che si riavvia una volta che l'input è stato ciclato,

ovvero che torna a livello basso e viene riportato ad 1 logico, quindi la protezione è bistabile. Una resistenza in serie collegata all'ingresso fornisce il segnale per l'unità di protezione interna all'IPS. Quanto a piedinatura, quella del componente è simile a quella di qualsiasi MOSFET in TO-220, però il piedino a sinistra non si chiama Gate, bensì Input; il centrale è il Drain e quello a destra il Source.

Saldare le batterie

Probabilmente vi sarete imbattuti nella necessità di collegare ai dei fili delle batterie NiCd, NiMH o al litio 18650, 16430, 22650, 32650 dotate di contatti, per realizzare pacchi batterie: a fili o a la-

➔ Schema di principio e schema a blocchi dell'IPS1021 International Rectifier.



melle. Ebbene, la saldatura tradizionale a stagno (brasatura, come si chiama esattamente) non è molto consigliabile perché scaldare gli elettrodi esterni altera, se non danneggia, la chimica delle batterie suddette. Non a caso le batterie collegate in pacchi mediante lamelle e le stesse lamelle a saldare vengono applicate in fabbrica tramite una tecnica di saldatura differente, che sostanzialmente è un'elettrosaldatura; più esattamente tale tecnica è nota come saldatura a resistenza e più comunemente, puntatura. Questo nome deriva sostanzialmente dal modo in cui viene eseguita la saldatura, che è per punti; un po' come avviene nell'assemblaggio di alcuni lamierati nell'automobile.

La tecnica si basa sulla differente resistenza elettrica tra gli elettrodi, molto bassa, e quella del materiale da saldare, che è più elevata; per effetto di ciò, quando si applicano i puntali, a scaldarsi è il metallo a maggiore resistenza, che arriva a fondere. In parole povere tra i due metalli da saldare si applicano due elettrodi a punta composti da materiale a bassissima resistenza, tipicamente rame, sottoposti poi ad un breve impulso di tensione abbastanza elevata da far scorrere, per un breve istante, una corrente elettrica molto intensa e tale da surriscaldare, fino a fonderlo, il metallo da congiungere.

Nel caso delle saldature industriali, si chiudono gli elettrodi di rame sulle parti da saldare, in modo da esercitare

una certa pressione, quindi si fa scorrere in un primo momento un corrente ridotta (corrente di preriscaldamento) per un tempo abbastanza breve, poi si applica l'impulso di saldatura ad alta corrente, per un intervallo simile a quello della precedente, quindi si fa di nuovo scorrere una corrente ridotta per un tempo simile ai due precedenti in modo da far raffreddare gradualmente la giuntura ed evitare saldature fredde.

Volendo effettuare questa saldatura a livello amatoriale ci si può limitare a fornire un impulso di corrente di valore adatto a fondere le superfici dei due pezzi in contatto.

Gli apparati per questo genere di saldatura sono ormai tutti elettronici e in essi si carica una batteria di

condensatori ad un valore di tensione abbastanza elevato, quindi la batteria si chiude sul primario di un trasformatore abbassatore il cui secondario alimenta gli elettrodi di saldatura attraverso un raddrizzatore a diodi di potenza, passando da un parzializzatore d'onda o un ponte di Graetz controllato composto da SCR invece che da diodi comuni. Per scaricare i condensatori sul primario del trasformatore si può usare un contattore o un MOSFET ad alta tensione. In queste pagine trovate lo schema concettuale di un moderno dispositivo per elettrosaldatura a resistenza. Per l'elettrosaldatura al banco di batterie e simili esiste in commercio una gran quantità di dispositivi a batterie, alimentabili e ricaricabili anche da USB, dotati di puntali in rame: se ne trovano per tutte le tasche, con potenze impulsive anche di 3 o 4 kW e correnti di scarica che arrivano intorno ai 700A.

↓ Saldatura con i due puntali sulla lamina da saldare.

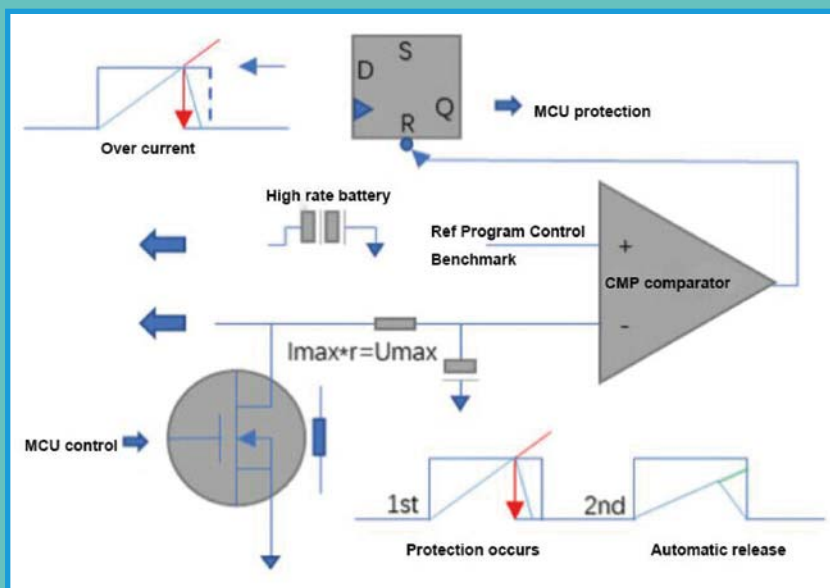


Il protocollo KNX

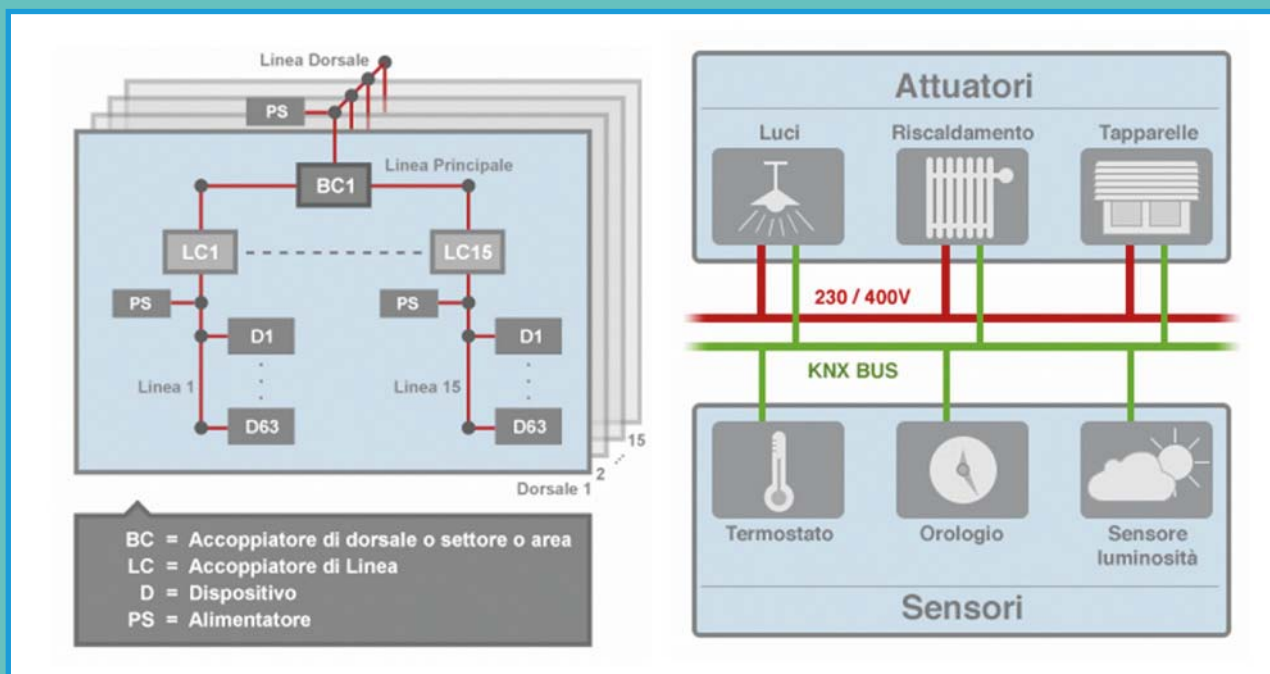
In ambito domotico e nello Smart Building si è da tempo affermato il KNX, che è un protocollo di comunicazione a scopo di controllo, monitoraggio e comando di ogni genere di utilizzatore, sensore, apparato: dal condizionamento (HVAC) agli azionamenti ed all'illuminazione. KNX può essere considerato l'erede di tre standard precedenti, ossia European Home Systems Protocol (EHS), BatiBUS, European Installation Bus (EIB o Instabus).

Contrariamente ad un impianto elettrico standard, nel KNX non c'è alcuna connessione elettrica fisica tra le unità di controllo e l'alimentazione degli utilizzatori: ad esempio un interruttore KNX non commuta direttamente la tensione sul rispettivo utilizzatore, ma si limita a inviare un comando (telegramma) digitale contenente lo stato che questa deve assumere, che verrà eseguito dall'attuatore KNX posto sul circuito di alimentazione dell'utilizzatore stesso.

Tutti i segnali, ossia comandi da dispositivi di comando ad attuatori, ovvero segnali di sensori diretti a dispositivi di elaborazione ed attuazione, feedback e quant'altro, viaggiano sotto forma di "oggetti" caratterizzati da stati logici o valori numerici; elemento comune a tutti gli elementi KNX è il bus, composto da due soli fili (la connessione avviene mediante doppino ritorto). L'intero bus funziona a 29 V e tutti i dispositivi sulla linea vengono alimentati da tale tensione,



↑ Schema a blocchi di un'elettrosaldatrice per elettronica.



↑ Schematizzazione della struttura e del funzionamento di un sistema KNX.

con la condizione che non debbano assorbire dalla linea più di 30 milliamper; per questa ragione negli attuatori si fa ampio uso di relè ad autoritenuta o bistabili, ma anche di condensatori in grado di immagazzinare energia per il momento in cui serve azionare elementi che potrebbero superare tale assorbimento.

Tutti gli elementi sul bus possono essere programmati con un solo strumento, quindi il bus KNX permette un'installazione facile e molto flessibile, e anche le successive modifiche possono essere fatte facilmente senza cambiare il cablaggio.

Sostanzialmente, un sistema KNX richiede i seguenti componenti:

- alimentatore per alimentare l'impianto, ovvero il bus a due fili;
- sensori (pulsanti, termostati, contatori di energia, ecc..) che generano comandi inviando telegrammi;
- attuatori (a relè per le luci, tapparelle, ecc..) che ricevono i telegrammi e attuano determinate azioni (gli oggetti).

Il KNX è progettato per essere indipendente da qualsiasi particolare piattaforma hardware. In modo più semplice possibile, un microcontrollore a 8 bit può realizzare la funzione desiderata, per funzioni più complesse può essere necessario un PC, secondo le esigenze di una particolare implementazione. Il

mezzo di trasmissione più comune in KNX è il doppino intrecciato, ma KNX prevede anche altri mezzi di comunicazione fisica, implementata attraverso traslatori (accoppiatori di rete KNX). La più piccola entità all'interno della topologia KNX è una **linea**, rispettivamente, un segmento di linea. Una linea può contenere un massimo di 64 dispositivi (tale soluzione è sufficiente per la maggior parte dei progetti di piccole dimensioni).

Per realizzare impianti di maggiori dimensioni è necessario dividere i dispositivi in più linee a formare un **Settore**; quest'ultimo è formato da un massimo di 15 linee interfacciate a una linea principale; l'interfacciamento delle linee o segmenti di linea con la linea principale avviene ad opera di un accoppiatore di linea. Quindi l'accoppiatore di linea (BC) permette di affacciare più linee (fino a 15) su una **linea principale**.

A sua volta la linea principale (costituente, insieme ai dispositivi di pertinenza, un'area dell'impianto) può essere interfacciata a una linea Dorsale tramite un accoppiatore di area o settore; con l'accoppiatore è possibile collegare fino a 15 aree a una linea **Dorsale**.

I dispositivi terminali KNX possono essere collegati ovunque in questa topologia; il numero massimo di dispositivi terminali KNX inseribili in

ogni sottorete è 255. Quindi dispositivi KNX possono essere numerati da 1 a 255, ricordando che non è possibile assegnare il numero 0.

Ciascun dispositivo KNX (Accoppiatori di Settore, Linea o dispositivi terminali..) deve avere un Indirizzo Individuale (o Fisico) che però dev'essere unico nell'intera topologia di rete KNX, altrimenti si crea conflitto.

Il protocollo KNX non è di libero utilizzo e chi vuole implementarlo nei propri apparati deve corrispondere delle fee (diritti in denaro) al consorzio che ne detiene la proprietà.

Web Forum e supporto tecnico

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

www.elettronica.in.it/webforum

Un'offerta sempre più grande

La più ampia selezione di semiconduttori e componenti elettronici disponibili e pronti per la spedizione™



[mouser.it](https://www.mouser.it)



Il MIUR ha finanziato il progetto OpenEYE, ideato da PaperLit, tech company del Gruppo Datrix, che in 18 mesi porterà alla creazione di una piattaforma che, grazie ad un dispositivo di nuova concezione e all'integrazione con sistemi di domotica, permetterà alle persone con deficit visivi di ascoltare contenuti scritti su carta via Google Home o Amazon Echo, grazie all'Intelligenza Artificiale. La piattaforma sarà composta dai tre elementi: OpenEye Reader, dispositivo a MCU dotato di fotocamera ad alta risoluzione che acquisisce le edizioni cartacee, elabora le immagini in modo da garantire la leggibilità delle lettere e la compressione dei dati e le trasmette via WiFi al Server OpenEye per l'analisi; quest'ultimo, dotato di tecnologie di

OpenEYE: l'AI legge per chi non vede



OCR (Optical Character Recognition) e Machine Learning che distinguono gli elementi della pagina e restituiscono un testo "logico" leggibile dagli assistenti vocali, discernendo le parti del testo (titolo, didascalie, corpo) ed il montaggio dello stesso secondo il corretto ordine di lettura (l'impaginazione editoriale posiziona il testo in maniera in-

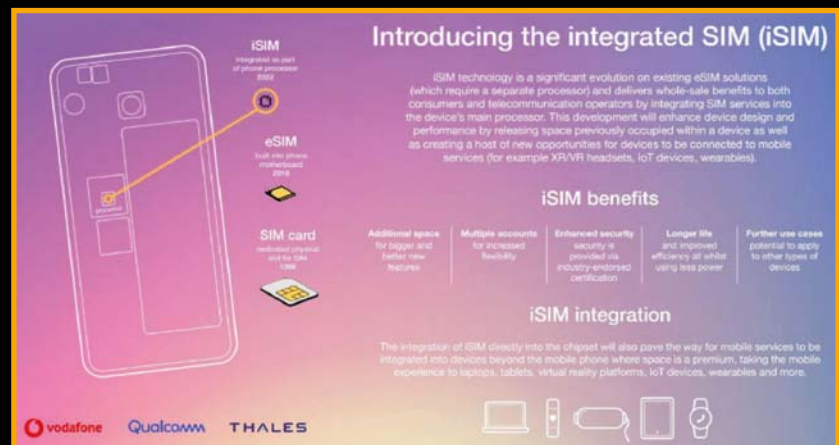
tuitiva per un essere umano ma non per una macchina). In aggiunta, è molto frequente lo spezzettamento di contenuti editoriali su più pagine non contigue tra di loro. Per il raggiungimento di questo obiettivo si farà ricorso alle più recenti ricerche in ambito machine learning ed in particolare alle tecniche di Natural Language Processing. Infine l'Assistente OpenEye, app disponibile per i servizi Google Home e Amazon Echo, riceve i comandi vocali da parte dell'utente, li invia al Reader o al Server per gestire il servizio, e riceve dal Server il testo da leggere all'utente con deficit visivi. Il progetto conterà sulla collaborazione con 3rdPlace e il Dipartimento di Ingegneria Elettronica dell'Università di Cagliari.

datrixgroup.com

iSIM, la SIM integrata nel processore

Vodafone, Qualcomm e Thales hanno dato vita ad uno smartphone dotato di iSIM (basato sulla specifica iUICC) conforme alle specifiche GSMA, che incorpora la funzionalità SIM nel processore principale del dispositivo, consentendo una maggiore integrazione del sistema, prestazioni più elevate e una maggiore capacità di memoria. Questa tecnologia è l'ultima evoluzione della tecnologia SIM in cui le "eSIM" sono integrate nei dispositivi, ma mentre eSIM richiede un chip separato, iSIM non necessita di spazio dedicato ai servizi SIM.

La tecnologia iSIM apre la strada all'integrazione dei servizi mobili nei dispositivi oltre al telefono cellulare, portando l'esperienza mobile su laptop, tablet, piattaforme di realtà virtuale, dispositivi IoT, dispositivi indossabili e altro ancora. L'iSIM consente il provisioning della SIM remota da parte dell'operatore sfruttando l'infrastruttura eSIM esistente e apre le funzionalità di connessione del ser-



vizio mobile a un host di dispositivi che in precedenza non potevano disporre di funzionalità SIM integrate. Una dimostrazione si è svolta in Europa, lavorando sull'infrastruttura esistente di Vodafone; il dispositivo è basato sul Samsung Galaxy Z Flip3 5G alimentato da una piattaforma mobile Snapdragon 888 5G, che presenta

un'unità di elaborazione sicura Qualcomm integrata, che esegue il sistema Thales iSIM ai fini della dimostrazione. Il Proof Of Concept si è svolto nei laboratori di ricerca e sviluppo della Samsung, utilizzando l'avanzata piattaforma di gestione remota di Vodafone.

www.qualcomm.com

› Machine learning nello stampaggio a iniezione

Plus10, spin-off del Fraunhofer Institute tedesco, insieme al centro di ricerche sulle materie plastiche della Germania del Sud SKZ, ha animato il progetto DarWIN, volto ad implementare l'intelligenza artificiale e in particolare il Machine Learning, nello stampaggio ad iniezione, facendo colloquiare tra loro macchine di diversi costruttori, in modo che queste potessero imparare l'una dall'altra. Un modello di apprendimento continuo che si propone di ridurre i tempi di ciclo dello stampaggio e migliorare la qualità dei prodotti, dimostrando che i modelli comportamentali di ogni singola macchina, una volta appresi, possono essere trasferiti, adattandoli, a diverse macchine e produzioni, senza intervento dell'operatore.

Per testare gli strumenti sviluppati è stato scelto come partner Sumitomo (SHI) Demag, produttore nippo-tedesco di macchine per stampaggio plastiche a iniezione AI-ready. Questa fase è servita a validare la capacità dei modelli di adattare in modo automatico i parametri di processo a variazioni esogene, come - ad esempio - le caratteristiche delle materie prime o le condizioni ambientali. È stata anche utile per dimostrare che i pattern operativi e i modelli comportamentali sono trasferibili a macchine di pari tonnellaggio e tecnologia. I ricercatori di plus10 e SKZ hanno eseguito una serie di test pilota di intelligenza artificiale su macchine Demag Sumitomo (SHI) sfruttando le capacità di connessione in tempo reale tra

› Geolocalizzazione con Cloud IoT Explorer di Tencent

Semtech e Tencent Cloud hanno annunciato un accordo per offrire i servizi di geolocalizzazione LoRa Cloud in Cina. Tencent Cloud IoT Explorer e connettere immediatamente i dispositivi IoT al cloud utilizzando LoRa Edge. Lanciata nel 2020, LoRa Edge (LR1110) è una piattaforma software-defined altamente versatile che fornisce funzionalità di geolocalizzazione interna ed esterna; suddivide il workload per calcolare la posizione tra il dispositivo e LoRa Cloud, estendendo l'autonomia dei dispositivi IoT a mesi o anni, a vantaggio delle applicazioni IoT industriali, agricole, dei trasporti e della logistica. Inoltre LoRa Edge

è compatibile con lo standard LoRaWAN, nonché pienamente compatibile con i requisiti normativi globali per le comunicazioni wireless in Cina, Stati Uniti ed Europa. Il servizio di geolocalizzazione IoT di Tencent Cloud ha supportato capacità di posizionamento indoor e outdoor diversificate. Se combinato con il programma LLink Mini di Tencent, agli utenti viene offerta un'esperienza semplice e conveniente. La piattaforma device-to-Cloud LoRa di Semtech è una soluzione a lungo raggio e bassa potenza adottata a livello globale per applicazioni IoT, che consente lo sviluppo e l'implementazione rapida di reti IoT, gateway, sensori,

prodotti a modulo e bassissimo consumo, efficienti in termini di costi e Servizi IoT in tutto il mondo. I dispositivi LoRa di Semtech forniscono il livello di comunicazione per lo standard LoRaWAN, che è mantenuto da LoRa Alliance, un'alleanza IoT aperta per applicazioni LPWAN (Low Power Wide Area Network) che è stata utilizzata per implementare reti IoT in oltre 100 paesi. Semtech è un membro fondatore della LoRa Alliance.

Tencent Cloud IoT Explorer è una piattaforma PaaS (Platform As A Service) per Internet delle cose per applicazioni di interconnessione industriale e smart life. Fornisce servizi intelligenti per dispositivi completi per produttori di apparecchiature, fornitori di soluzioni e sviluppatori di applicazioni in vari settori basati sull'Internet delle cose. La piattaforma offre enormi capacità di connessione e gestione dei dispositivi e capacità di sviluppo di applicazioni basate sul programma Tencent LLink Mini, sfruttando i prodotti Tencent Cloud e le capacità di Intelligenza Artificiale, in modo da migliorare l'efficienza delle apparecchiature intelligenti nelle industrie tradizionali e ridurre i costi di sviluppo, funzionamento e manutenzione degli utenti.

Per saperne di più su come LoRa abilita l'IoT, visita il sito LoRa di Semtech.

www.semtech.com



le presse. Per raggiungere lo scopo, il team di ricerca ha implementato un sistema di Deep Learning da diversi produttori di macchinari. Il programma di intelligenza artificiale DarWIN si è concentrato specificamente sulla raccolta di comportamenti di elaborazione dettagliati e trasferibili da più modelli di macchine per lo stampaggio a iniezione che utilizzano dati macchina ad alta frequenza. Attraverso la trasferibilità di modelli di apprendimento automatico pre-addestrati, il progetto DarWIN dimostra che le singole macchine di diversi produttori hanno la capacità di imparare l'una dall'altra.

Questa capacità di adattarsi alle condizioni prevalenti, ad esempio le caratteristiche delle materie prime e le condizioni ambientali, dimostra che i modelli di elaborazione e i modelli comportamentali sono trasferibili a macchine di dimensioni e tecnologia simili, indipendentemente dal produttore.



Il progetto DarWIN è stato finanziato dal Ministero Federale dell'Istruzione e della Ricerca tedesco.

www.sumitomo-shi-demag.eu

› IoT contro i roghi boschivi

La certezza di poter fermare un incendio e limitarne i danni dipende dalla possibilità di identificarlo più rapidamente possibile; sfortunatamente l'estensione delle zone boschive e l'assenza di insediamenti umani rendono ciò piuttosto difficile. Dryad ha affrontato il problema sviluppando nodi sensori avanzati e utilizzando una tecnologia di rete altamente distribuita. Progettata per superare una serie di vincoli tipici degli ambienti forestali, come la mancanza di alimentazione e di connettività, la soluzione Silvanet di Dryad è costituita da tre componenti hardware, da una rete RF proprietaria che estende la portata del protocollo di comunicazione LoRaWAN e da una piattaforma di analisi cloud.



Il nodo Silvanet Wildfire Sensor, alimentato da un pannello fotovoltaico, rileva le concentrazioni di vari gas: è basato su un microcontrollore che integra supporto LoRaWAN e funzioni di elaborazione ad apprendimento automatico. Più Wilfire Sensor si collegano ai gateway Silvanet Mesh alimentati a energia solare; i gateway si interconnettono formando una rete di reti a stella, che insieme possono coprire anche le foreste più ampie.

Questi due elementi creano una struttura "Internet of Trees" (Internet degli alberi) che si affaccia sul web tramite il gateway Silvanet Border, posizionabile ai margini della foresta, fungendo da gateway LoRa e implementando la connettività 4G o 2G/GPRS, laddove



necessario. La piattaforma cloud Silvanet può raggruppare sensori e gateway sia per area geografica che per caratteristiche.

L'ambiente forestale rappresenta una sfida enorme a livello di implementazione. Ad esempio, per alimentare i sensori, il sistema deve utilizzare supercondensatori anziché le classiche batterie ricaricabili, potenzialmente tossiche e infiammabili.

All'hardware si affianca un ambiente firmware di analisi dei dati caratterizzato da vincoli di alimentazione, aggiornabile over-the-air e conforme agli standard che riguardano LoRaWAN e la sicurezza IoT.

Il tutto deve essere ottenuto con apparecchiature funzionanti a bassa potenza, alimentate da pannelli solari spesso in ombra.

Oltre a questo, anche le comunicazioni RF rappresentano una sfida poiché ogni albero è un ostacolo che assorbe potenzialmente le onde radio.

L'infrastruttura Silvanet può monitorare altri aspetti come l'umidità del suolo, la crescita degli alberi, il flusso di linfa e persino le attività illegali di disboscamento, oltre a consentire il monitoraggio della qualità dell'aria e dell'acqua su vaste aree.

www.dryad.net

MINI TRAPANI per ogni esigenza

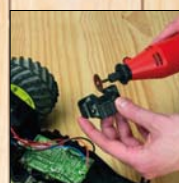
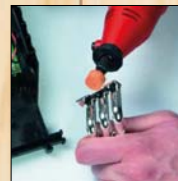


Mini Trapano con set per incisione - 162 Pezzi

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

Le MCU AURIX Infineon fanno l'auto elettrica e digitale

Al CES 2022, Infineon ha lanciato una nuova generazione della propria famiglia di microcontroller (MCU) AURIX; si tratta della serie TC4x, che promuove le tendenze dell'eMobility, dei sistemi avanzati di assistenza alla guida (ADAS), delle architetture elettriche-elettroniche (E/E) del settore automobilistico e delle applicazioni di intelligenza artificiale (AI) a prezzi accessibili. Inoltre, la sua scalabilità consente di avere un'architettura software comune, che si traduce in risparmi significativi sui costi della piattaforma. Infineon AURIX è diventata l'architettura automobilistica di prima scelta per applicazioni ad alta crescita e critiche per la sicurezza come powertrain, sicurezza, guida assistita e autonoma, nonché controllo di dominio o di zona. Data la crescente complessità dei veicoli e l'abilitazione delle implementazioni dell'IA, è stata posta un'attenzione significativa sull'ecosistema



AURIX TC4x per garantire un rapido time-to-market e facilità d'uso; per questo una delle principali case automobilistiche europee disporrà di circa 35 MCU AURIX a bordo di una nuova vettura che sarà prodotta a partire dalla metà di questo decennio. La famiglia AURIX TC4x offre connettività, sicurezza e pro-

tezione avanzate; le sue nuove funzionalità SOTA (Software Over the Air) aiutano a soddisfare le richieste delle case automobilistiche per una connessione car-to-cloud veloce e sicura, consentendo aggiornamenti, diagnosi e analisi durante l'utilizzo del veicolo.

www.infineon.com

Renesas a quota 1 miliardo di MCU



Renesas ha superato il miliardo di MCU a 32 bit della famiglia RX, basata sul core CPU RX proprietario. Le MCU RX offrono prestazioni CPU elevate grazie alla loro architettura per sistemi embedded. Attualmente sono disponibili quattro serie di MCU RX (RX100, RX200, RX600 e RX700 di fascia alta) che offrono un'ampia gamma di prestazioni, memoria integrata e una varietà di funzioni e pacchetti periferici.

Dopo l'introduzione dell'ultimo core RXv3 nel 2018, Renesas sta anche lavorando allo sviluppo del prossimo core con l'obiettivo di superare le prestazioni del core RXv3. Per facilitare l'implementazione degli MCU RX, Renesas offre un configuratore intelligente e un generatore di codice, in risposta alla crescente necessità di codice basso e nullo per ridurre il carico di sviluppo dei sistemi embedded.

www.renesas.com

Wireless ad alte prestazioni per l'IoT

Murata presenta il suo nuovo modulo Type 1XA dedicato alla progettazione di sistemi IoT. Basato sul SoC AIROC 54591 di Infineon/Cypress, supporta la connettività Wi-Fi 5 dual-band 2x2 e wireless BLUETOOTH 5.2 dual-mode. Supporta inoltre la funzionalità Real Simultaneous Dual Band (RSDB) che garantisce connessioni simultanee in modalità 1x1 su entrambe le bande 2,4 GHz e 5 GHz. Questa soluzione di Murata e Infineon completamente certificata, offre la migliore connettività AIROC della categoria per applicazioni quali gateway, altoparlanti intelligenti, stampanti, telecamere IP e altri sistemi IoT di alta fascia. Murata Type 1XA WLAN+BLUETOOTH Combo Module supporta lo standard Wi-Fi 802.11a/b/g/n/ac 2x2 MIMO RSDB e il Bluetooth 5.2 BR/EDR/LE con data-rate PHY (a livello fisico) fino

a 866Mbps su Wi-Fi e fino a 3Mbps su Bluetooth. La sezione WLAN supporta l'interfaccia PCIe v3.0 Gen 1 mentre la parte Bluetooth supporta un'interfaccia UART high-speed four-wire e il PCM per lo stream audio. In Wi-Fi, supporta modulazioni da MCS0 a MCS9 (fino a 256 QAM) nei canali 20MHz, 40MHz e 80MHz, per data-rate fino a 866Mbps.

Il modulo Murata Type 1XA WLAN+Bluetooth Combo è disponibile in package LGA da 11,4x8,9x1,4 mm.

www.murata.com



Fotorelé Toshiba ad alta tensione 1,5kV per BMS



Toshiba lancia un fotorelé normalmente aperto (NO) 1-Form-A destinato al sistema di gestione della batteria (BMS) delle auto elettriche. Il TLX9160T è il primo dispositivo Toshiba con una tensione di tenuta (V_{off}) di 1.500 V, in grado di funzionare da una tensione di alimentazione (V_{DD}) fino

a 1.000 V, quindi compatibile con tutte le batterie di trazione. Alloggiato in un contenitore SO16L-T modificato, realizzato con un materiale in resina IEC60664-1 I, che ha un indice di tracciamento comparativo (CTI) superiore a 600, il TLX9160T presenta una distanza di dispersione

di almeno 5 mm, tensione di isolamento (BVS) di 5kVrms (min.) e distanze di fuga e di sicurezza di 8 mm. La massima corrente diretta (IF) è 30 mA mentre la corrente in stato On (ION) è di 50 mA con una corrente in stato Off (IOFF) di soli 100 nA. Qualificato AEC-Q101.

www.toshiba.com

Power Converter radiation-tolerant da Microchip

I progettisti di sistemi aerospaziali non possono accontentarsi dei tradizionali convertitori di potenza ibridi; per questo Microchip offre convertitori CC-CC di qualità aerospaziale basati su componenti discreti, che ora includono dispositivi radiation-tolerant da 50 W. I recenti SA50-28 con ingresso a 28 V sono facili da personalizzare in fatto di tensione, corrente, ecc. La famiglia di converter switching SA50-28 di Microchip è l'unica disponibile sul mercato, con ingresso a 28 V, tollerante alle radiazioni, che si basa su componenti discreti con costruzione a montaggio superficiale e processi di assemblaggio non ibridi. Offrendo più capacità rispetto ai convertitori di potenza alternativi standard e di qualità spaziale, un singolo dispositivo SA50-28 con parametri personalizzati elimina i problemi di volume, peso e complessità dell'utilizzo di soluzioni ibride con i loro dispositivi multipli e i circuiti circostanti. La famiglia SA50-28, qualificata secondo QML, spazia da 20V a 40V con nove tensioni d'uscita standard da 3,3V, 5V, 12V, 15V e 28V in configurazioni a uscita



singola e tripla, con una potenza massima di 50W. I dispositivi possono essere adattati alle esigenze di alimentazione di un sistema in un tempo relativamente breve con costi aggiuntivi minimi rispetto ai prodotti di conversione di potenza in stile ibrido. Altre caratteristiche includono alta efficienza, basso rumore di uscita, controllo di inibizione dell'uscita, protezione da sovracorrente, sincronizzazione esterna e funzionamento

a piena potenza da -55°C a $+85^{\circ}\text{C}$ con derating lineare a $+125^{\circ}\text{C}$. La famiglia SA50-28 fa parte della linea crescente di convertitori di potenza standard non ibridi di qualità spaziale di Microchip che consentono ai progettisti di utilizzare componenti disponibili in commercio, i cui circuiti hanno una comprovata esperienza in operazioni spaziali.

www.microchip.com

Cardea e Siemens Healthineers per nuovi lab-on-chip

Cardea Bio collaborerà con il Center for Innovation in Diagnostics (CID) di Siemens Healthineers allo sviluppo applicazioni di biosensori in tempo reale basati sulla BPU (Biosignal Processing Unit) di Cardea.

Lo scopo è arrivare a un test immunologico SARS-CoV-2 di nuova generazione e, in

caso di successo, espandere la collaborazione ad altre aree che potrebbero includere la misurazione simultanea dell'antigene e il rilevamento dell'RNA senza amplificazione per dimostrare la potenza combinata della sensibilità a livello di PCR e la velocità dei test immunologici in un unico test.

La BPU è un chip a semiconduttore capace di tradurre flussi in tempo reale di segnali multiomici, in informazioni digitali. Scopo di Cardea è democratizzare l'accesso ai biosensori e arrivare all'Internet della Biologia.

www.cardeabio.com



Semtech offre compatibilità SDVoE ai sistemi LivePremier di Analog Way



Semtech ha annunciato che Analog Way ha integrato la sua tecnologia BlueRiver nelle sue nuove schede di input/output compatibili con Software Defined Video Over Ethernet (SDVoE) rendendole interoperabili con qualsiasi extender e sistema di controllo SDVoE.

I sistemi di presentazione multi-schermo 4K/8K modulari e scalabili di Analog Way e i processori per video wall sono orientati allo staging di fascia alta e all'integrazione di sistema premium. La piattaforma BlueRiver di Semtech apre l'Ethernet alla distribuzione del segnale AV, cambiando per sempre le dinamiche del

settore Pro AV e consentendo agli installatori e agli OEM di sostituire gli switch a matrice AV proprietari con reti Ethernet standard che offrono un rapporto prezzo/prestazioni notevolmente migliore, flessibilità e scalabilità. La piattaforma BlueRiver offre un singolo SoC programmabile utilizzabile per sostituire i tradizionali extender AV/KVM, matrici, controller per video wall e processori windowing con una semplice rete di trasmettitori, ricevitori ed Ethernet standard interruttori. I chipset BlueRiver sono alla base della tecnologia SDVoE che offre streaming 4K non compresso a latenza zero su Ethernet standard, con supporto esclusivo per HDMI 2.0a e HDR; integrano anche il ridimensionamento video ad alta fedeltà, il windowing e il downmixing dell'audio.

www.semtech.com

Nuovi moduli di ricarica wireless di ROHM

ROHM ha sviluppato moduli caricabatteria wireless dotati di antenna integrata su PCB: BP3621 (trasmettitore) e BP3622 (ricevitore). Consentono di aggiungere funzionalità di alimentazione wireless a dispositivi più piccoli come smart tag/card o periferiche per PC. I moduli ROHM di ricarica wireless operano a 13,56MHz e consentono ai progettisti di aggiungere facilmente funzionalità di alimentazione wireless a piccoli dispositivi. Misurando solo da 20 a 30 mmq ciascuno, il BP3621 e il BP3622 incorporano un layout dell'antenna (bobina) ottimizzato. Consentono la configurazione di un sistema di ricarica wireless compatto che utilizza la banda di frequenza 13,56MHz per fornire fino a 200mW. La struttura della scheda posteriore completamente piatta facilita il



montaggio in dispositivi sottili e compatti, contribuendo a una maggiore flessibilità nella progettazione dello chassis rispetto alle soluzioni convenzionali. L'antenna integrata supporta la comunicazione dati bidirezionale e il funzionamento del tag NFC Forum Type 3, ampliandone l'applicabilità.

www.rohm.com

Il buffer DAQ TI decuplica la larghezza di banda

Texas Instruments presenta l'amplificatore buffer ad alta impedenza di ingresso BUF802, con ben 3 GHz di larghezza di banda. La larghezza di banda raggiunta dal BUF802 era precedentemente possibile solo utilizzando circuiti integrati specifici dell'applicazione (ASIC) con aumento dei tempi, della complessità e dei costi di progettazione del sistema. Le precedenti alternative alle implementazioni di progettazione basate su ASIC richiedevano dozzine di componenti discreti come FET, diodi di protezione e transistor e non garantivano la stessa larghezza di banda degli ASIC, limitando il throughput del segnale delle applicazioni di acquisizione dati. Il BUF802 fornisce un'alternativa a chip singolo agli ASIC o alle implementazioni basate su amplificatori con ingresso FET, integrando le caratteristiche dei componenti discreti fornendo al contempo una larghezza di banda 10 volte più ampia rispetto agli amplificatori con ingresso FET, eguagliando le prestazioni degli ASIC personalizzati.

Il flessibile BUF802 è il primo buffer del settore a consentire la regolazione della corrente di riposo

per una gamma di requisiti di larghezza di banda e oscillazione del segnale, da 100 MHz a 3 GHz a 1 Vpp e fino a 2 GHz a 2 Vpp. Questo ampio intervallo di regolazione per larghezza di banda e oscillazione del segnale consente ai progettisti di ridimensionare facilmente i loro progetti front-end su più applicazioni di acquisizione dati, riducendo i costi del sistema e la riprogettazione. Le modalità funzionali integrate consentono agli ingegneri di utilizzare il BUF802 come buffer autonomo o in un circuito composito con un amplificatore di precisione come l'OPA140. Come buffer autonomo, il BUF802 può aiutare a ottenere un'elevata impedenza di ingresso e velocità di variazione elevate in applicazioni che possono tollerare offset di 100 mV o in cui la catena del segnale è accoppiata in c.a. In un loop composito, il nuovo buffer può raggiungere un'elevata precisione CC e una larghezza di banda di 3 GHz in applicazioni che richiedono una deriva di offset massima di 1 $\mu\text{V}/^\circ\text{C}$. Il BUF802 è disponibile in package VQFN (very thin no-lead) a 16 pin da 3x3 mm.

www.ti.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/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"

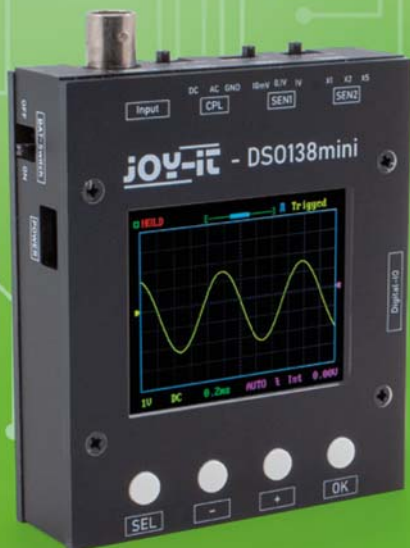
La possibilità di **LEGGERE GRATUITAMENTE**
la versione digitale di ELETTRONICA IN



Ora i componenti non hanno più segreti!

DISPOSITIVI PROFESSIONALI PER MISURE E ANALISI PRECISE

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



COD. DSO138MINI
€ 59,90

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

TESTER PER COMPONENTI ELETTRONICI



COD. JT-LCRT7
€ 29,00

Dotato di display LCD da 3.5" a colori, ricevitore a infrarossi e batteria ricaricabile integrata da 3,7 volt 350 mAh. Permette di misurare diodi, diodi zener, resistori, condensatori, l'ESR (Resistenza Equivalente Serie) di un condensatore, induttori, tiristori, triac, FET, transistor bipolar (indicando se è NPN o PNP) e batterie (max. 4,5 V). Funziona anche con condensatori elettrolitici polarizzati e non polarizzati in entrambi gli orientamenti. Grazie al ricevitore a infrarossi integrato possono essere decodificati anche i segnali a infrarossi e visualizzati sul display, in forma d'onda e codice esadecimale. Le misurazioni vengono avviate automaticamente premendo semplicemente un pulsante.

Prezzi IVA inclusa.

Analizzatore di componenti DCA55



€ 73,00
cod. DCA55

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

Analizzatore di semiconduttori DCA75



€ 148,00
cod. DCA75

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

Misuratore di induttanza, capacità, resistenza e impedenza LCR45

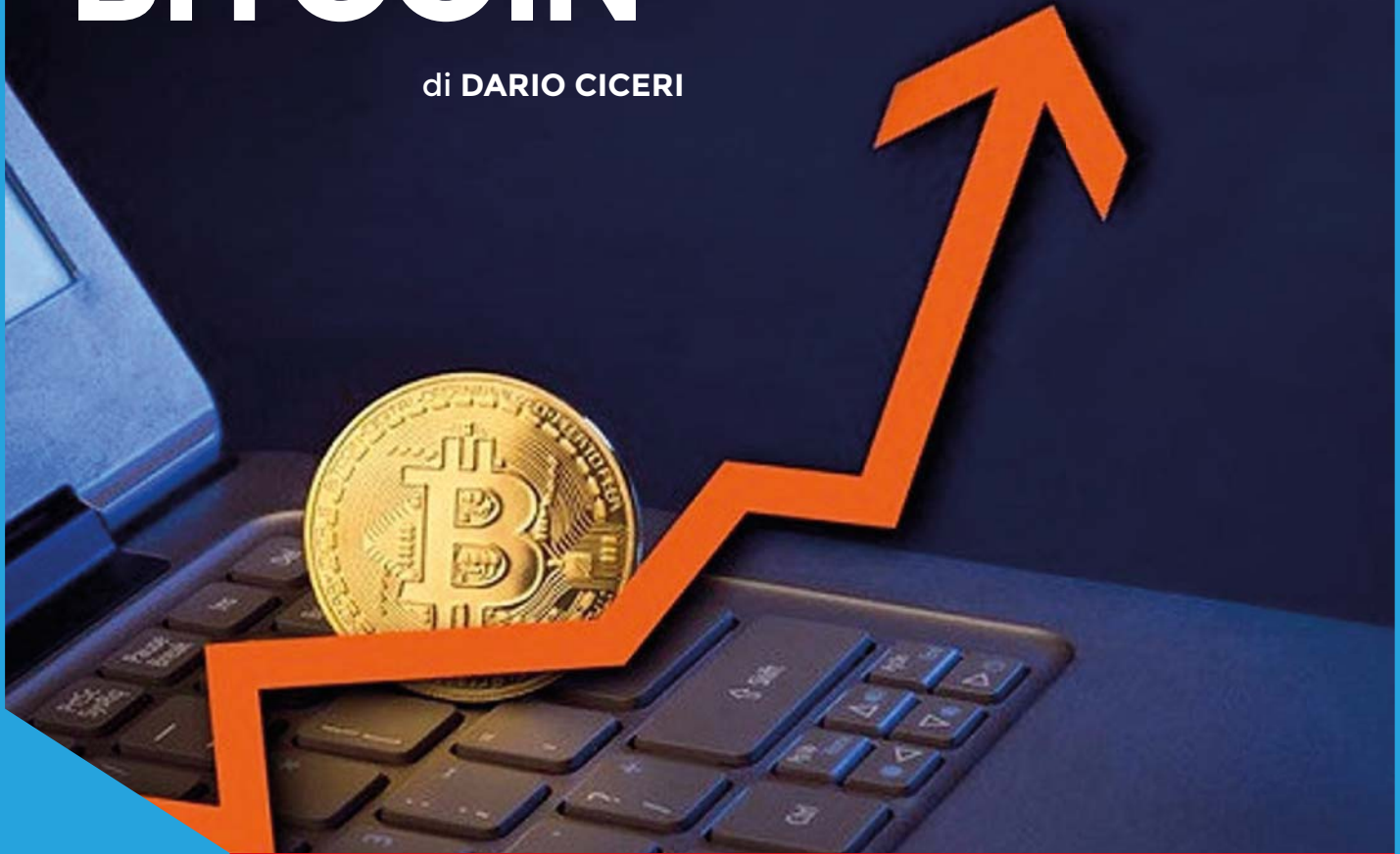
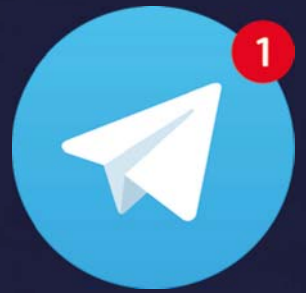


€ 129,00
cod. LCR45

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

TRACCIATORE DI BITCOIN

di DARIO CICERI



Utilizziamo un ESP32 per monitorare l'andamento dei Bitcoin e essere avvisati tramite Telegram di variazioni significative.

G

razie ai progressi in campo tecnologico e all'uso di una crittografia sempre più avanzata, l'uso delle criptovalute, dal 2009 ad oggi, è diventato sempre più diffuso portando a fenomeni come quello del Bitcoin che nel mese di ottobre dell'anno scorso è salito da 30 a 60 mila dollari in appena 90 giorni.

Le criptovalute sono state create per fornire un metodo di pagamento alternativo per le transazioni online. Ad oggi, non sono ancora in molti ad utilizzarle e sono attualmente troppo volatili per essere considerate dei metodi di pagamento solidi. Per questo motivo spesso sono utilizzate per fare speculazione più che come metodo di pagamento e monitorare costantemente il valore di una criptovaluta non è facile e può portare via parecchio tempo: attualmente non è

Fig. 1
Scheda Heltec WiFi
ESP32OLED.

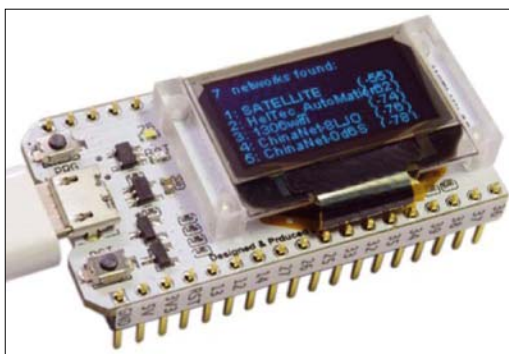
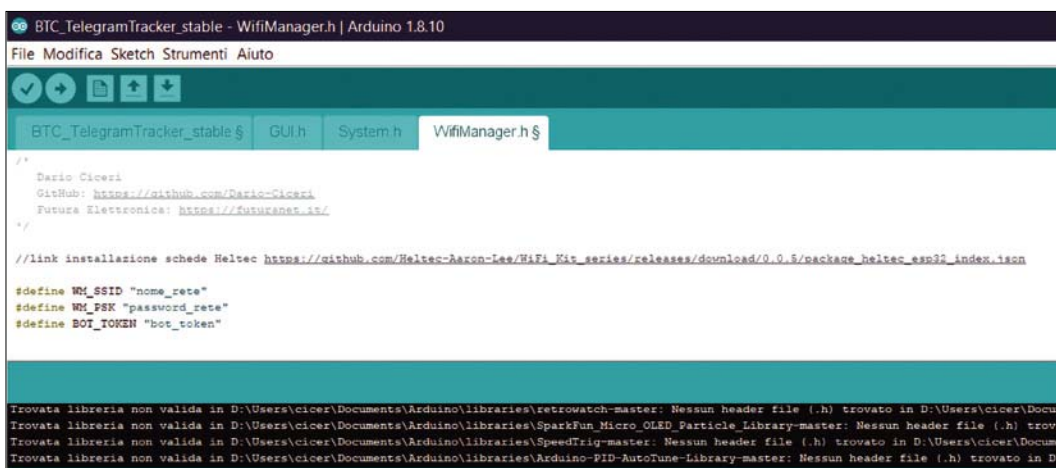


Fig. 2
The Botfather di
Telegram.



Fig. 3
Scheda
WifiManager.h.



possibile prevedere facilmente quando sarà il momento migliore per vendere o comprare criptovalute, così abbiamo creato uno strumento in grado di tracciare i valori di una criptovaluta, informarci quando raggiunge un determinato target, avvisarci dopo un determinato periodo di tempo oppure tenerci aggiornati ogni volta che il valore cambia in positivo o in negativo.

DESCRIZIONE DEL SISTEMA

Il sistema è basato su una scheda di sviluppo a 32bit dalle dimensioni ridotte (circa 52x25x10mm), che utilizza il chip ESP32 **Fig. 1**. Dispone di Bluetooth e WiFi, schermo OLED 0.96" e una memoria SRAM interna da 520kB, perfetta per gestire tutti i dati di cui abbiamo bisogno. Lo schermo OLED, permette di visualizzare in tempo reale il valore odierno e del giorno precedente della criptovaluta, insieme alla differenza percentuale tra i due, così da aiutarci a capire se è il momento di vendere o comprare. La scheda, dispone inoltre di diversi pin di input e output che possiamo utilizzare, ad esempio, per attivare una periferica in base ad un evento come potrebbe essere l'aumentare o il diminuire del valore della criptovaluta.

Per questo progetto abbiamo deciso di monitorare i valori senza controllare alcuna periferica esterna in base agli eventi, tuttavia modificare il codice è piuttosto semplice e vedremo più avanti come fare. Rispetto alle altre schede in commercio, quella che vi proponiamo possiede non solo un display OLED da 0.96" di dimensioni davvero ridotte, ma anche un circuito integrato per la ricarica di batterie ai polimeri di litio, il che la rende ottima per progetti IoT e per monitorare le criptovalute ovunque, anche fuori casa. La scheda

può essere alimentata da 3.3 a 7 volt e dispone di funzioni integrate per ridurre al minimo il consumo di dati e batteria che possono essere abilitate via firmware.

CREARE UN BOT TELEGRAM

Per ricevere notifiche e inviare comandi dobbiamo creare una chatbot su Telegram con la quale potremo interagire e gestire l'intero sistema. Prima di procedere con il firmware dunque, apriamo Telegram e avviamo una chat con @BotFather (Fig. 2), premiamo su `/newbot`, seguiamo le istruzioni per creare il nostro bot e copiamo il token che ci verrà fornito.

COMPILARE IL FIRMWARE PER LA SCHEDA ESP32

Una volta configurata la chatbot su Telegram possiamo dedicarci al firmware dell'ESP32 (scaricabile dal sito della rivista). Il codice è strutturato in maniera tale da poter modificare le sezioni più importanti senza doverle ricercare all'interno dell'intero listato.

Aprendo il file .INO all'interno dell'IDE troverete quattro schede come mostrato in Fig. 3. Iniziamo da quella chiamata *WifiManager.h* dove possiamo impostare le credenziali per la rete WiFi e il token della chatbot. Prima di programmare la scheda dovremo installare all'interno dell'IDE la versione di ESP32 che stiamo utilizzando, ovvero la Heltec WiFi Kit 32. Per fare questo da Impostazioni (Fig. 4), selezioniamo "URL aggiuntive per il Gestore schede" come mostrato in Fig. 5 e scriviamo l'URL da cui il nostro IDE potrà recuperare le informazioni: https://github.com/Heltec-Aaron-Lee/WiFi_Kit_series/releases/download/0.0.5/package_heltec_esp32_index.json

Dal *Gestore schede* dell'IDE (Fig. 6), cerchiamo Heltec e installiamo la versione 0.0.5 (Fig. 7), successivamente selezioniamo la scheda WiFi Kit 32 tramite la voce *Strumenti* -> *Scheda*.

Una volta che abbiamo installato e configurato il sistema, possiamo caricare il codice e aprire la chatbot su Telegram che verrà utilizzata come interfaccia uomo-macchina.

Avviata la chat e inviato il comando `/start`, la scheda invierà la lista dei comandi che potremo usare per interagire con essa (Fig. 8):

- `/info` - informazioni
- `/auto` - attiva/disattiva modalità automatica
- `/notifiche` - attiva/disattiva robot
- `/percentuale` - invia un messaggio in base alla differenza % tra ieri ed oggi

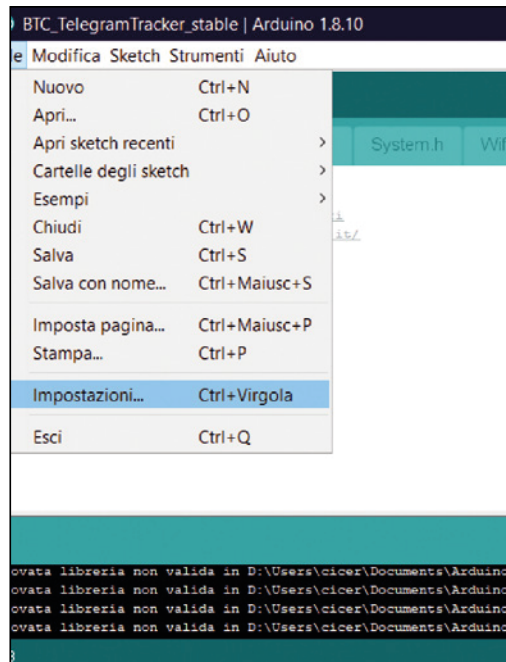


Fig. 4
Impostazioni
Arduino IDE.

- `/minuti` - invia un messaggio ogni 'x' minuti
- `/watchdog` - attiva/disattiva se inviare una notifica dopo il riavvio

Il comando `/percentuale` attiva una particolare funzione grazie alla quale verremo avvisati dalla chatbot se ci sarà una variazione pari alla percentuale indicata con questa istruzione.

Il comando `/watchdog` (in inglese, cane da guardia) attiva una funzione speciale che ci permette di sapere se la scheda è stata resettata o ha perso l'alimentazione, così da permetterci, al riavvio, di riprendere la funzione di ricerca in caso

Fig. 5
URL aggiuntive per
il Gestore Schede.

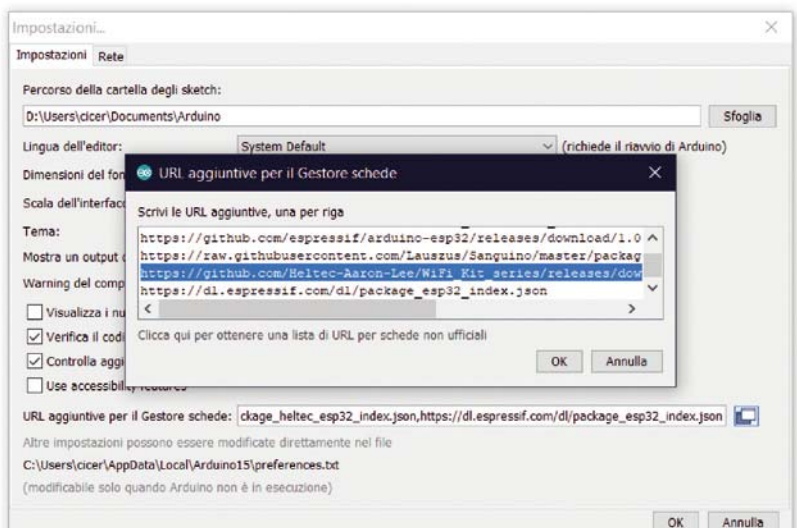
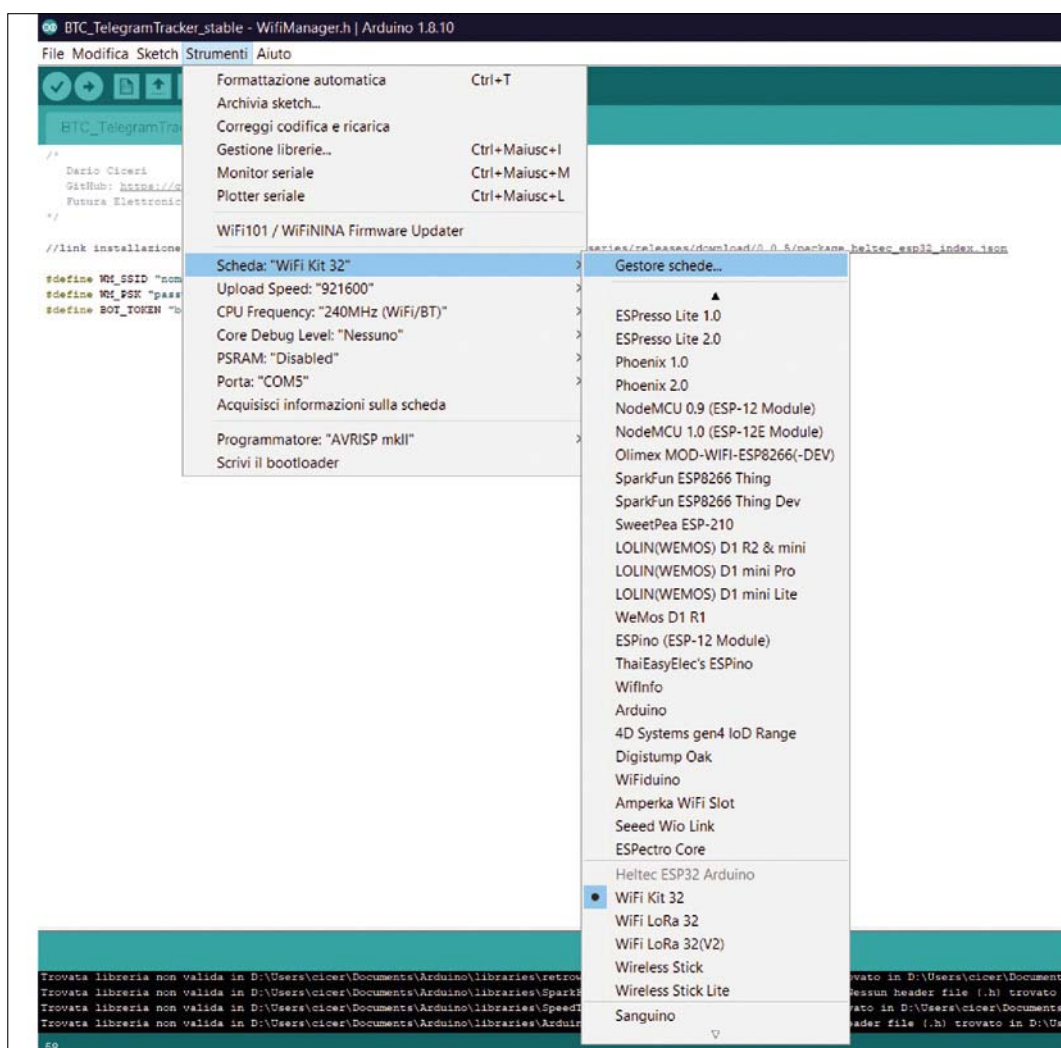


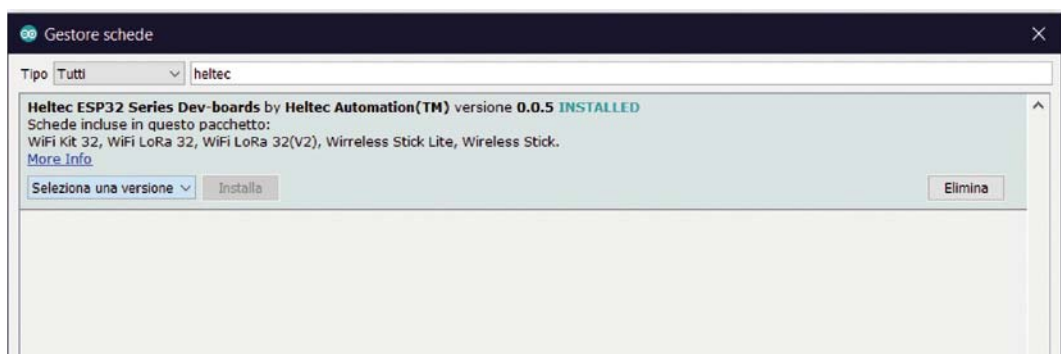
Fig. 6
Il Gestore Schede di
Arduino IDE.



avessimo inviato precedentemente il comando */percentuale*. I comandi possono essere eseguiti uno alla volta e per alcuni è necessario terminare il processo in esecuzione prima di poter fare altre richieste, non è possibile ad esempio richiedere il comando */minuti* mentre */percentuale* è

ancora in esecuzione e in chat vi verrà chiesto se volete terminare il processo in corso o ignorare il comando che avete appena inviato, in altri casi alcuni comandi possono essere eseguiti in modo istantaneo e non interrompono alcuna funzione in corso.

Fig. 7
Installazione della
scheda Heltec.



▼ Listato 1

```
void sysCallbackMainRoutine() {  
[...]  
cmd_list = "-----\n";  
cmd_list += "Lista dei comandi:\n";  
cmd_list += "/info - informazioni\n";  
[...]  
cmd_list += "-----\n";  
[...]  
}
```

COME MODIFICARE IL CODICE

Ipotizziamo di voler aggiungere altri comandi per eseguire delle funzioni personalizzate, le cose da fare sono principalmente due e andranno svolte all'interno della scheda *System.h*. La prima è creare il comando e aggiungerlo insieme agli altri comandi all'interno della funzione *handleNewMessages*.

```
void handleNewMessages(int numNewMessages)  
{  
[...]  
Serial.println();  
[...]  
}
```

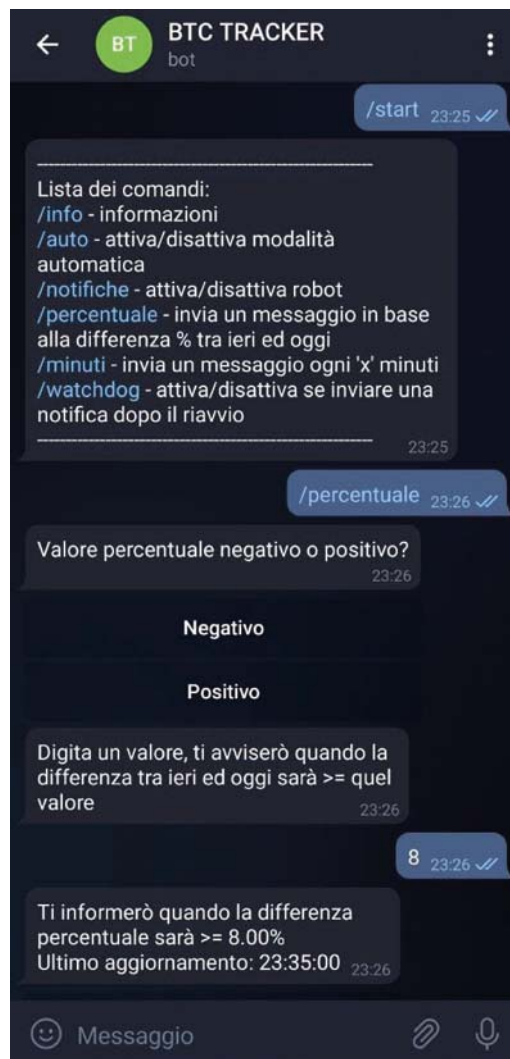
All'interno della funzione *handleNewMessages*, dopo l'ultimo comando per la seriale *Serial.println()*; riportato nel listato appena sopra, possiamo aggiungere il nuovo comando utilizzando la porzione di codice riportata nel listato qui di seguito.

```
if (text == "/nuovo_comando" && !callCheckChange)  
{  
//istruzioni da eseguire  
}
```

Come vediamo viene creata un'espressione di controllo che verifica se il testo letto in chat corrisponde al nome del nuovo comando e viene controllato un flag chiamato *callCheckChange* che indica se il comando */percentuale* è ancora in esecuzione. Questo controllo va aggiunto ad ogni nuovo comando per impedirne l'esecuzione di una nuova funzione mentre il comando */percentuale* è attivo e rischiare quindi di interrompere il suo corretto funzionamento. Se il testo in chat corrisponde al nome del nuovo comando e il valore del flag *callCheckChange* è false, allora vengono eseguite le istruzioni all'interno dell'espressione condizionale. La seconda cosa da fare è aggiornare la lista dei comandi inviata alla chat all'interno della funzio-

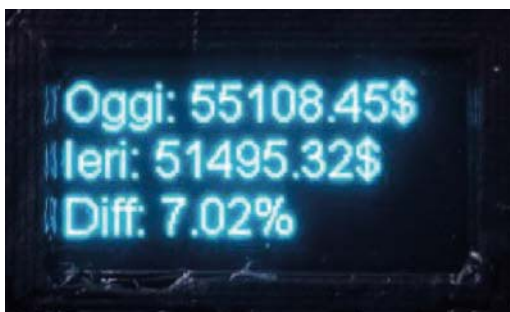
ne *sysCallbackMainRoutine* che possiamo trovare all'interno della scheda *System.h* (**Listato 1**). Per aggiungere il nuovo comando basta inserire la seguente riga di codice:

```
cmd_list += "/nuovo_comando - descrizione del nuovo  
comando\n";
```



◀ **Fig. 8**
Chatbot su
Telegram.

→ Fig. 9
Interfaccia su
schermo OLED.



prima dell'ultima istruzione

```
cmd_list += "-----\n";
```

Infine basta compilare, caricare il codice sulla scheda e verificare che tutto funzioni correttamente.

INSERIRE ICONE E LOGHI PERSONALIZZATI

Possiamo sfruttare lo schermo OLED anche per mostrare delle icone che ci aiutino a capire se ci sono novità importanti, oppure per inserire un'immagine di avvio personalizzata. Le dimensioni massime di questo display sono 128x64 pixel e sono sufficienti a mostrare piccole immagini e icone. Per aggiungere icone e/o loghi personalizzati possiamo utilizzare diversi strumenti online per convertire un'immagine in una porzione di codice che possa essere interpretato dalla scheda. Il formato richiesto dall'ESP32 per poter gestire correttamente le immagini è del tipo *xbm*. All'interno della sezione *GUI.h* del nostro codice possiamo aggiungere le immagini che vogliamo specificando le dimensioni dell'immagine nelle righe di codice:

```
#define width 128
#define height 64
```

e creare un array contenente i dati dell'immagine che avete convertito:

```
const unsigned char nome_immagine[] PROGMEM =
{dati dell'immagine};
```



→ Fig. 10
Esempio di splash
screen.

infine possiamo utilizzare il seguente comando per mostrare l'immagine sul display:

```
Heltec.display->drawXbm
    (0, 0, width, height, nome_immagine);
Heltec.display->display();
```

Andando in ordine, i primi due numeri sono rispettivamente la posizione X e la posizione Y dell'angolo superiore sinistro dell'immagine da visualizzare, gli altri due valori sono le costanti che abbiamo dichiarato, ovvero la larghezza e l'altezza dell'immagine, mentre l'ultimo dato è l'array contenente i dati necessari per mostrare l'immagine correttamente.

Una volta compilato e caricato il codice sulla scheda, il risultato finale dovrebbe essere di questo tipo **Fig. 10**.

CONCLUSIONI

Questo codice utilizza le API di CoinDesk che potete trovare al sito <https://www.coindesk.com/price/bitcoin> e che ci permettono di leggere i valori di Bitcoin utilizzando la libreria *ArduinoJson*. Con qualche modifica è possibile riutilizzare l'intero codice per monitorare altre criptovalute o altri valori in tempo reale sfruttando le funzioni già presenti per prelevare i dati e gestire la chatbot. Il codice del progetto è scaricabile dal sito www.elettronica.in.it e potete modificarlo per aggiungere altre funzioni, per monitorare altre valute o per attivare dei pin in caso di variazioni significative. □



Cosa occorre?

La board con ESP32 (cod. ESP32OLED) è disponibile presso Futura Elettronica a Euro 24,00, la batteria per rendere il modulo autonomo (cod. PLIB1100) è disponibile a Euro 7,90. 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>

HDMI
↓
VGA

€ 11,50

Cod. **BOXHDMITOVGA**

CONVERTITORE DA HDMI A VGA 1080P PLUG AND PLAY

Permette di convertire un segnale digitale HDMI (compatibile con HDMI1.3) in un segnale VGA (SUB-D) ad alta definizione con audio in uscita. Con questo convertitore, il computer, il lettore video portatile, le console di gioco e alcuni altri dispositivi video con uscita HDMI possono connettersi a schermi HD di grandi dimensioni oppure a proiettori HD con ingresso VGA.

CONVERTITORE DA AV A HDMI PLUG AND PLAY

AV
↓
HDMI

€ 14,00

Cod. **BOXAVTOHDMI**

Permette di convertire un segnale audio/video CVBS (PAL e NTSC) in ingresso in un segnale HDMI in uscita. Con questo convertitore, il computer, il lettore video portatile, le console di gioco e alcuni altri dispositivi video con uscita CVBS possono connettersi a schermi con ingresso HDMI. L'uscita HDMI supporta una risoluzione fino a 1080P 60Hz.

Convertitori di segnale

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

CONVERTITORE DA HDMI AD AV PLUG AND PLAY

Permette di convertire un segnale HDMI in ingresso in un segnale video composito CVBS (PAL e NTSC) in uscita. Con questo convertitore, il computer, il lettore video portatile, le console di gioco e alcuni altri dispositivi video con uscita HDMI possono connettersi a schermi con ingresso CVBS. L'ingresso HDMI supporta una risoluzione fino a 1920x1080 a 60 Hz.

VGA
↓
HDMI

€ 11,50

Cod. **BOXVGATOHDMI**

CONVERTITORE DA VGA A HDMI PLUG AND PLAY

Permette di convertire un segnale audio e video VGA (SUB-D) in un segnale HDMI (tipo A) ad alta definizione con audio. Con questo convertitore, il computer, il lettore video portatile, le console di gioco e alcuni altri dispositivi video con uscita VGA possono connettersi a schermi HD di grandi dimensioni oppure a proiettori HD con ingresso HDMI. L'ingresso VGA supporta video fino a 1920x1080 a 60 Hz, l'uscita HDMI può supportare video fino a 1080P.

HDMI
↓
AV

€ 11,00

Cod. **BOXHDMITOAV**

**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.



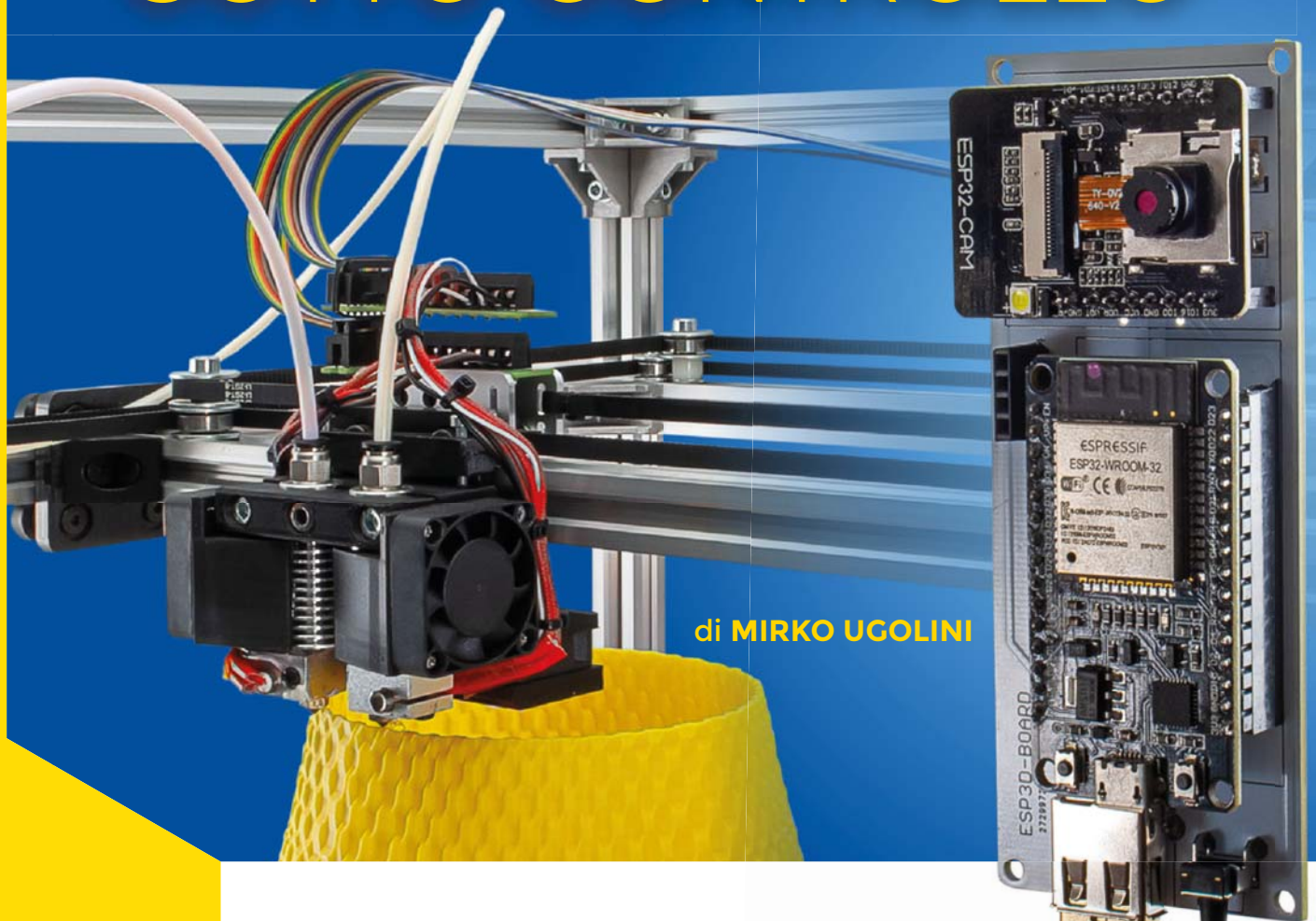
€ 25,50

Cod. **COMB06G**

Prezzi IVA inclusa

WIRELESS

LA STAMPANTE 3D SOTTO CONTROLLO

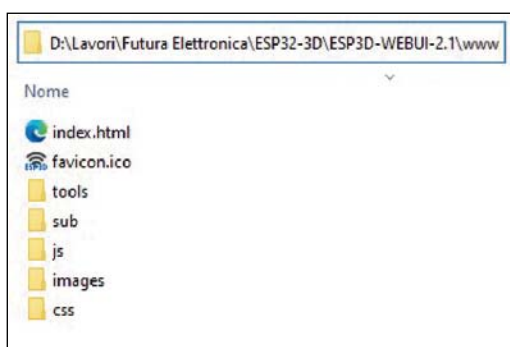


di MIRKO UGOLINI

Approfondiamo la preparazione dell'ambiente di sviluppo per il nostro controller remoto ESP3D. Seconda puntata.

Nella pratica quotidiana, chi utilizza una stampante 3D può, al pari di una stampante su carta, aver bisogno di gestirla da remoto; il dispositivo ESP3D che vi abbiamo introdotto nella prima puntata di questo articolo permette proprio di fare ciò. Dopo avervelo presentato descrivendo l'hardware, ne abbiamo spiegato a grandi linee il funzionamento e vi abbiamo messo in condizione di utilizzarlo immediatamente fornendovi i file binari da caricare nei due moduli che compongono ESP3D senza neppure aprire

Fig. 1
La cartella
con il sorgente.



l'IDE di Arduino. In questa puntata vogliamo invece spiegarvi nel dettaglio come preparare l'ambiente di sviluppo per poter compilare e debuggare sia il firmware da caricare sul modulo ESP32-DEV, sia il rispettivo codice Javascript e HTML che compone l'interfaccia grafica del nostro progetto. Analizzeremo, inoltre, l'interfaccia grafica. Per chi non fosse intenzionato a cimentarsi in alcuna modifica, questo articolo è comunque importante per capire come siano state implementate alcune soluzioni tecniche sia dal punto di vista progettuale che da quello di sviluppo e debugging del codice.

INTERFACCIA GRAFICA

Il cuore di tutto il dispositivo è il modulo ESP32-Dev il quale implementa due funzionalità: un server Web che esporrà l'interfaccia grafica ed una componente che si occupa di creare un canale di comunicazione tra la stampante 3D (tramite UART) e i controlli grafici presenti nella pagina Web.

Le pagine Web vengono realizzate tramite codice HTML (Hyper Text Markup Language) che definisce gli elementi che compongono la pagina: pulsanti, checkbox, radio button, input box, immagini e via di seguito.

Per realizzare una pagina web con aspetto gradevole e professionale è necessario affiancare al codice HTML, uno o più fogli di stile CSS (Cascading Style Sheets).

Ma non è sufficiente abbellire la pagina, perché è necessario darle vita tramite il linguaggio di programmazione vero e proprio (ricordiamo che HTML in senso stretto non lo è). Il linguaggio di programmazione che verrà utilizzato per animare la nostra interfaccia è, naturalmente, JS (Javascript).

Come potete vedere nella **Fig. 1**, abbiamo la cartella *www* che contiene tutto il sorgente dell'interfaccia grafica di ESP3D, abbiamo la pagina *index.html* che rappresenta la Homepage (HP) della Dashboard. Poi notiamo la presenza delle cartelle *sub*, *css* e *js*.

La cartella *sub* contiene le pagine web di tutte le varie componenti della Dashboard: per esempio il blocchetto che definisce la gestione degli estrusori, il controllo della temperatura, la visualizzazione dei comandi inviati e ricevuti dalla stampante, la pagina che visualizza lo stream della camera video ecc. Tutte queste funzionalità sono state ripartite su vari file per evitare una struttura monolitica del progetto e quindi consentire di gestire in modo più semplice le modifiche sulle diverse parti che compongono la Dashboard.

La cartella *css* contiene i fogli di stile che vengono impiegati nelle varie pagine HTML e che contribuiscono a dare al progetto un aspetto gradevole, funzionale e professionale.

La cartella *js*, infine, contiene invece il codice in linguaggio Javascript che rende viva la Dashboard e la fa comunicare con la parte server che gira sul modulo ESP32-DEV.

Normalmente nei siti Web, molti dei file CSS e JS vengono scaricati da repository on-line, questo per ottimizzare il funzionamento e beneficiare di alcuni meccanismi di caching.

Anche nel mondo dei controllori ESP32 alcune volte ci si avvale di questa soluzione, ma nel caso attuale si è preferito fare in modo che tutto sia contenuto nello spazio di memorizzazione del microcontrollore ed in particolare nella porzione di memoria SPIFF che è l'analogo (con le dovute limitazioni) di una cartella di un server Web.

Qual è il motivo di questa scelta? Ebbene, ce ne sono due: in primis la sicurezza che i file siano stabili nel tempo e non subiscano variazioni inaspettate se non tramite una nostra esplicita modifica; poi il fatto che di quei fogli di stile ne abbiamo necessità anche quando il dispositivo non è ancora connesso ad internet e quindi non saremmo in grado di scaricarli da un server esterno. Pensate al Wizard di configurazione iniziale in cui andiamo a settare i parametri della connessione WiFi. Anche la gradevolezza di questa pagina la dobbiamo ai fogli di stile CSS.

Bene, detto questo potreste pensare che sia sufficiente caricare il contenuto della cartella *www* di **Fig. 1** nello spazio di memorizzazione del modulo ESP32-DEV per ottenere il risultato desiderato. Purtroppo non è così semplice. ESP32 è un dispositivo meraviglioso che ci sorprende sempre per la sua versatilità, ma non fa miracoli e non ha certamente le risorse computazionali di un PC o di una Raspberry Pi.

Una limitazione particolare la dobbiamo ricercare nel fatto che il server Web che gira su ESP32 non

è in grado di servire contemporaneamente molte richieste, non solo provenienti da client concorrenti (più utenti che accedono alla stessa pagina), ma è limitato anche nel fornire alla sola pagina tutte le risorse che compongono la stessa e che vengono richieste in modo quasi contemporaneo nella fase di caricamento. Una pagina Web è composta da più immagini, icone, fogli di stile CSS, script JS da caricare per comporre la pagina e farla funzionare. Normalmente il caricamento delle vari componenti avviene autonomamente in tempi diversi e molto ravvicinati; dobbiamo quindi trovare un escamotage per superare queste limitazioni.

L'uovo di Colombo, in questo caso, consiste nell'impacchettare tutti i file che ci servono in un unico file compresso con estensione .gz.

Questa compressione non può esser fatta con strumenti quali Winzip, nel senso che non possiamo usare un normale programma di compressione, ma dobbiamo utilizzare un Tool che normalmente utilizzano i grafici. Viene in nostro aiuto un "software" che si chiama Gulp, che permette di automatizzare una serie di processi che altrimenti dovremmo eseguire a mano, con molta fatica e con la possibilità di commettere degli errori difficoltosi da scovare. Soprattutto se questi passaggi debbano essere ripetuti diverse volte in seguito alla modifica dei sorgenti nella fase di debugging.

Tramite Gulp invece, è possibile creare un file che descrive in modo abbastanza semplice tutti questi passaggi. Una volta creato lo script (che noi abbiamo realizzato per voi) e lanciato, questo ci permetterà di ottenere quello che ci eravamo proposti. Un singolo file che contiene tutta l'interfaccia Web e che deve essere scaricato una sola volta dal server web del dispositivo ESP32-DEV, ed il browser avrà a disposizione tutte le risorse necessarie senza dover ogni volta richiederle al server Web.

WEBSOCKET

Normalmente l'interazione tra la pagina Web caricata sul browser ed il server Web (nel nostro caso ESP32-DEV) che la fornisce al Web client avviene tramite i metodi POST e GET. Questo si traduce in un ricaricamento della pagina a seguito di ogni azione eseguita sui controlli presenti sulla pagina WEB. Pensate per esempio alla modifica del setup della temperatura dell'estrusore o del piatto riscaldato: ricorrendo alla tecnica tradizionale, ogni volta che effettuiamo una modifica dei valori dovremmo compiere un POST (invio) dei dati ed un ricaricamento della pagina per ricevere i valori aggiornati. Questa procedura, oltre ad essere poco efficiente

è anche poco gradevole dal punto di vista della UX (User Experience) e poco pratica.

Per risolvere tale problema è stata adottata la tecnologia dei Websocket. In pratica con essa si instaura un canale di comunicazione bidirezionale tra la pagina Web ed il server Web (ESP32-DEV) che permette da un lato di inviare i dati dei controlli dalla pagina Web al server (e di conseguenza alla stampante 3D) senza dover ricaricare la pagina e dall'altro di ottenere gli aggiornamenti dei controlli, (pensiamo alla renderizzazione grafica della temperatura dell'estrusore in tempo reale) con i dati provenienti dal server Web (e di conseguenza dalla stampante 3D).

GUIDA ALLA COMPILAZIONE CON GULP DELL'INTERFACCIA WEB

Vediamo ora quali sono le varie fasi da svolgere per l'installazione di Gulp. Per generare il file compresso **index.html.gz** che conterrà tutti i file html, css e js che definiscono l'interfaccia di ESD3D è necessario installare Nodejs, Gulp ed alcuni specifici add on. Posizionamici quindi nella cartella (ESP3D-WEBUI-2.1) che contiene il nostro progetto sorgente della interfaccia grafica e poi:

- a) installiamo Node.js;
- b) installiamo Gulp CLI;
- c) installiamo Gulp;
- d) installiamo gli AddOns;
- e) lanciamo Gulp.

INSTALLAZIONE DI NODE.JS

Andiamo quindi ad installare Node.js: è possibile scaricare il pacchetto di installazione di Node.js dall'URL <https://nodejs.org/en/download/>. Se stiamo usando Windows 10 scarichiamo il pacchetto a 64 bit (**Fig. 2**). Terminata l'installazione apriamo il prompt dei comandi nella cartella ESP3D-WEBUI-2.1.

Per fare questo velocemente, facciamo doppio clic sul file "openDosHere.bat". Poi nella finestra a riga di comando che si sarà aperta digitiamo:

```
node -v
```

e in risposta dovremmo ricevere a video il numero di versione di Node.js appena installato.

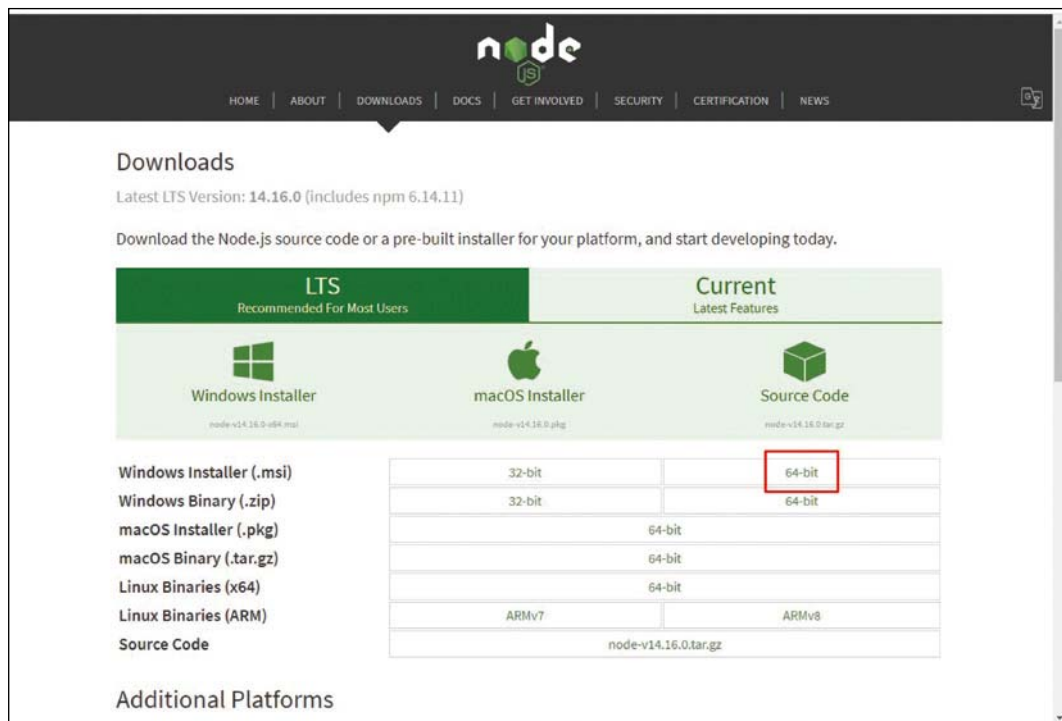
INSTALLAZIONE DI GULP

Sempre nella stessa finestra della riga di comando, digitiamo:

```
npm install --global gulp-cli
```



Fig. 2
Il pacchetto Node.js
per Windows 10.



seguito dal comando:

```
npm install --global gulp
```

Poi installiamo tutte le dipendenze tramite il comando:

```
npm install
```

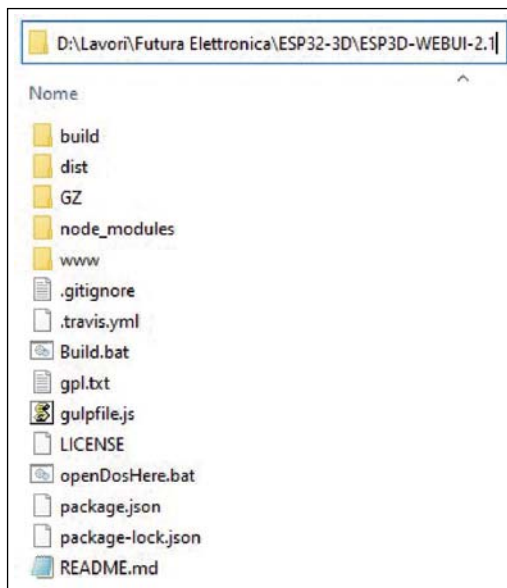


Fig. 3
Contenuto
della cartella.

Al termine verifichiamo sempre da prompt DOS se le installazioni siano andate a buon fine.

La cartella dell'interfaccia web avrà ora il contenuto mostrato nella **Fig. 3**.

Il file *gulpfile.js* è il file Gulp personalizzato per il nostro progetto, ed è lì che è specificato come eseguire la compressione dei singoli file, rimuovere spazi indesiderati ed inutili, bypassare certe sezioni nel codice, tutto al fine di ottenere un singolo file compresso. Mentre *Build.bat* è il file BAT DOS che dovremo lanciare per mandare in esecuzione lo script *gulpfile.js*.

La cartella *node_modules* contiene tutti i plug-in che sono stati scaricati per fare funzionare lo script di Gulp.

La cartella *www* contiene tutto il codice sorgente che possiamo personalizzare, html, css, js.

La cartella *build* al termine del processo conterrà i file che dovremo caricare nel filesystem SPIFF del modulo ESP32DEV, l'abbiamo già popolata con tre file statici che non devono essere rimossi, mentre il file più importante (*index.html.gz*) verrà inserito al termine della esecuzione dello script Gulp e nel caso fosse già presente, sarà rimosso nel primo passo della esecuzione dello stesso script.

Facciamo quindi doppio clic sul file *Build.bat*; nel caso in cui ci siano degli errori nei file *.js* ci verrà indicata la posizione dell'errore nei rispettivi file. Correggiamo gli eventuali errori e lanciamo nuova-

mente lo script Gulp finché otterremo una schermata (**Fig. 4**) priva di notifiche di errore. Al termine, se tutto è andato per il verso giusto dovremmo trovare nella cartella build il file *index.html.gz* che contiene al suo interno tutta l'interfaccia Web del modulo ESP3D. Quindi al termine della procedura di configurazione andremo a selezionare i file contenuti nella cartella build e li caricheremo tramite l'interfaccia web del Wizard nello spazio SPIFF del modulo ESP32DEV come abbiamo visto nella prima puntata.

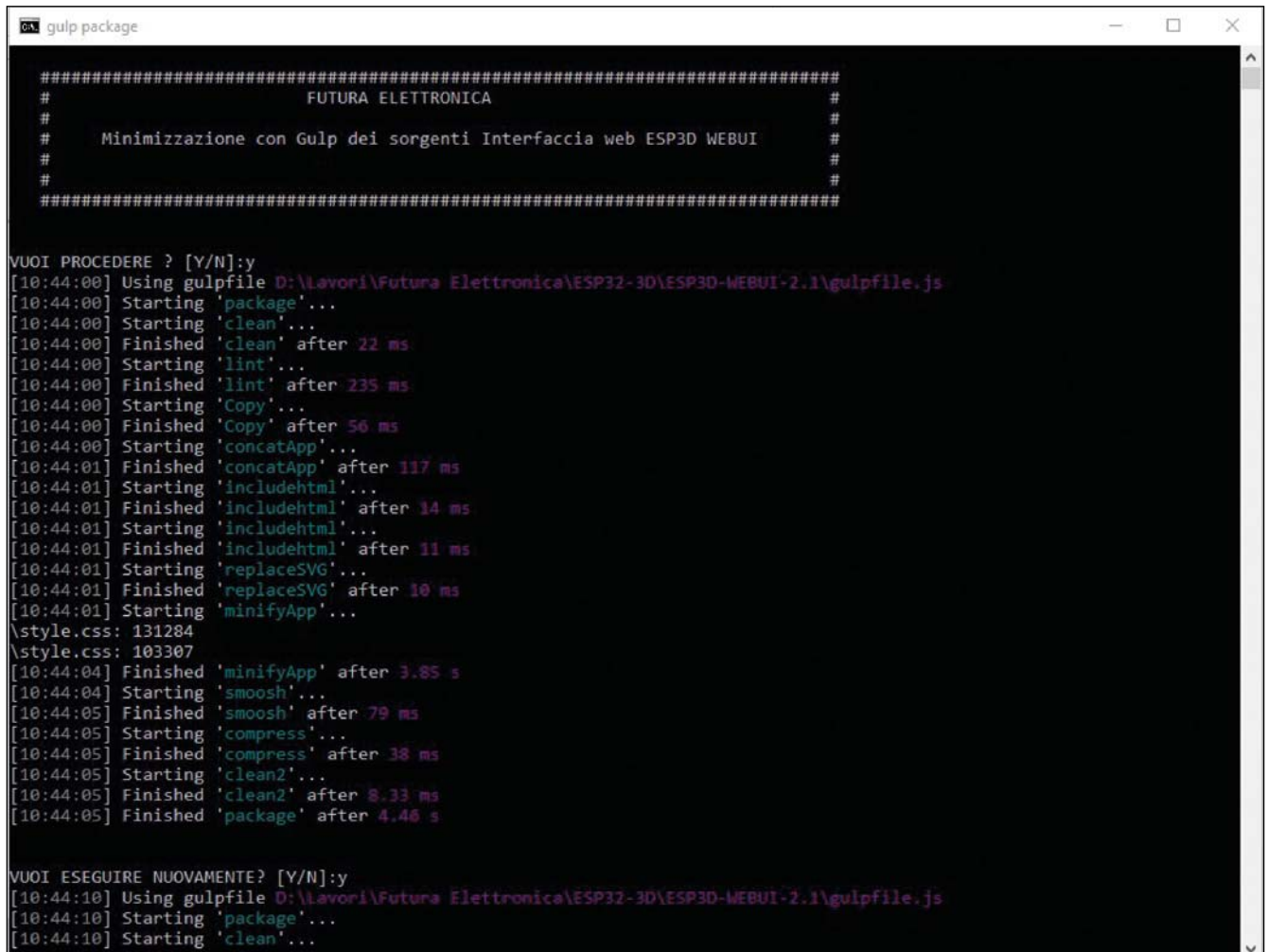
Nella **Fig. 5**, vediamo come funziona la comunicazione tra le varie componenti software e hardware nel caso di utilizzo del solo modulo ESP3D collegato alla stampante3D; i numeri rappresentano i passaggi descritti qui di seguito. Al punto (1) apriamo il browser con url *http://myesp* oppure usando *http://xxx.xxx.xxx.xxx* dove dovremo inserire l'indirizzo IP del Modulo ESP3D. Se abbiamo collegato il mo-

dulo alla stampante tramite il collegamento seriale, l'indirizzo IP acquisito dal modulo ESP3D apparirà nel display della nostra stampante 3D.

Al punto (2) il web server interno al modulo ESP3D risponde alla richiesta del browser andando a pescare il file compresso *index.html.gz* presente nell'area di memorizzazione SPIFF che contiene tutte le risorse di cui il browser ha necessità per rendere la DashBoard di controllo e quindi il browser visualizza l'interfaccia web con tutti i controlli. L'utente clicca con il mouse sul controllo di spostamento dell'estrusore nella posizione Home (3). La richiesta viene inviata tramite websocket al server Web ESP3D che apre e mantiene il canale Websocket aperto.

La richiesta viene elaborata dal codice C++ all'interno del modulo ESP3D per decidere cosa fare (4). Viene inviato un comando tramite la porta USART alla Motherboard della stampante 3D (5).

Fig. 4
Schermata di Gulp.



```
#####
#                                     #
#           FUTURA ELETTRONICA       #
#                                     #
#   Minimizzazione con Gulp dei sorgenti Interfaccia web ESP3D WEBUI   #
#                                     #
#####

VUOI PROCEDERE ? [Y/N]:y
[10:44:00] Using gulpfile D:\lavori\Futura Elettronica\ESP32-3D\ESP3D-WEBUI-2.1\gulpfile.js
[10:44:00] Starting 'package'...
[10:44:00] Starting 'clean'...
[10:44:00] Finished 'clean' after 22 ms
[10:44:00] Starting 'lint'...
[10:44:00] Finished 'lint' after 235 ms
[10:44:00] Starting 'copy'...
[10:44:00] Finished 'copy' after 56 ms
[10:44:00] Starting 'concatApp'...
[10:44:01] Finished 'concatApp' after 117 ms
[10:44:01] Starting 'includehtml'...
[10:44:01] Finished 'includehtml' after 14 ms
[10:44:01] Starting 'includehtml'...
[10:44:01] Finished 'includehtml' after 11 ms
[10:44:01] Starting 'replaceSVG'...
[10:44:01] Finished 'replaceSVG' after 10 ms
[10:44:01] Starting 'minifyApp'...
\style.css: 131284
\style.css: 103307
[10:44:04] Finished 'minifyApp' after 3.85 s
[10:44:04] Starting 'smoosh'...
[10:44:05] Finished 'smoosh' after 79 ms
[10:44:05] Starting 'compress'...
[10:44:05] Finished 'compress' after 38 ms
[10:44:05] Starting 'clean2'...
[10:44:05] Finished 'clean2' after 8.33 ms
[10:44:05] Finished 'package' after 4.46 s

VUOI ESEGUIRE NUOVAMENTE? [Y/N]:y
[10:44:10] Using gulpfile D:\lavori\Futura Elettronica\ESP32-3D\ESP3D-WEBUI-2.1\gulpfile.js
[10:44:10] Starting 'package'...
[10:44:10] Starting 'clean'...
```

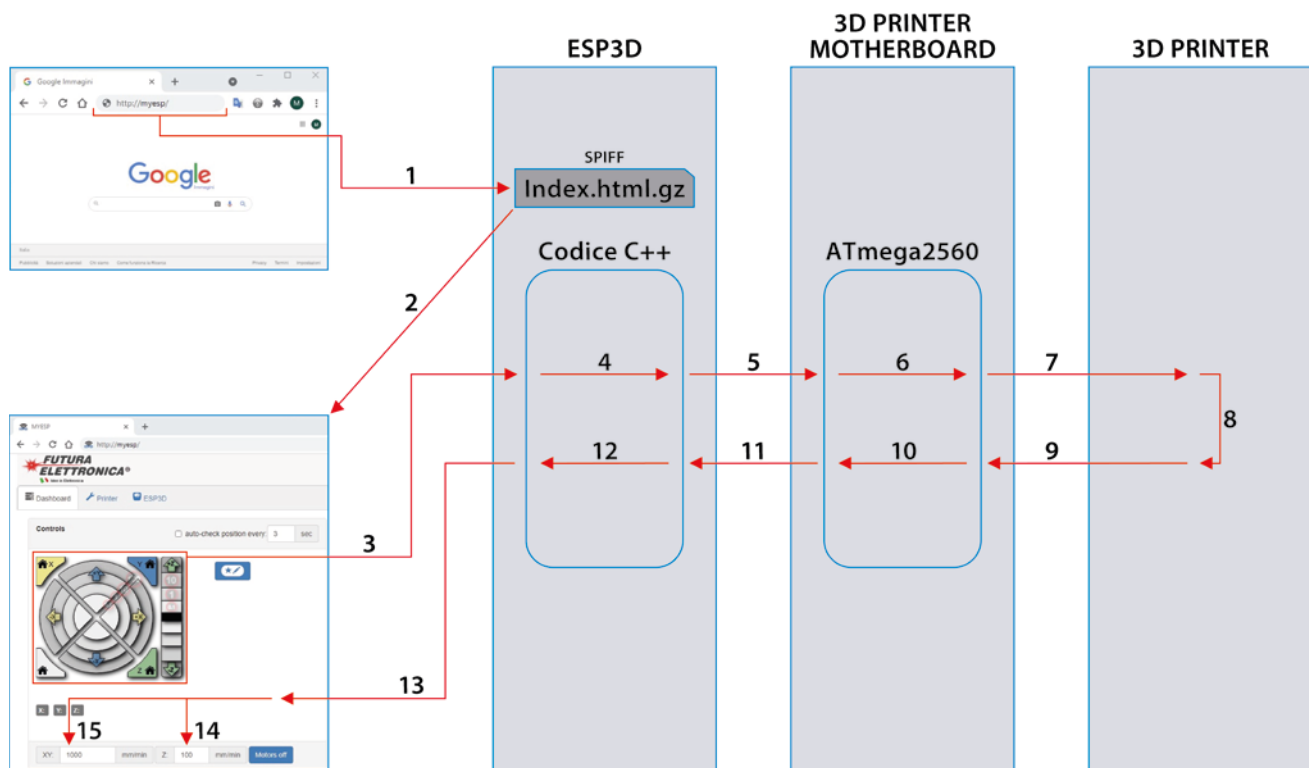


Fig. 5
Procedimento
di comunica-
zione.

Il processore ATmega2556 elabora la richiesta (6) ed invia i comandi di movimento alla stampante 3D (7). La stampante 3D esegue i comandi di movimento ricevuti (8).

Viene quindi ritornato un comando al processore ATmega2556 (9) e quest'ultimo elabora la risposta (10).

Al punto 11 il comando corrispondente viene trasferito tramite USART al modulo ESP3D. Il codice C++ del modulo ESP3D elabora la risposta (12) e tramite il canale Websocket vengono inviati dei dati all'interfaccia WEB del browser (13).

Ai punti 14 e 15 il codice Javascript presente sul browser elabora i dati ricevuti dal Websocket ed aggiorna le due text box che riportano la posizione X,Y,Z dell'Estrusore.

Tutto questo processo avviene senza ricaricamenti della pagina HTM visualizzata dal browser.

E non è tutto.

In virtù del canale Websocket che si è instaurato, eventuali modifiche al posizionamento dell'estrusore di stampa dovute ad esempio all'inizio della fase di stampa verrebbero visualizzate aggiornate senza intervento dell'utente.

Vi rendete sicuramente conto della estrema utilità di questa tecnica.

GESTIRE LE MODIFICHE AL CODICE

Bene, abbiamo visto come generare il file *index.html.gz* partendo dai sorgenti del nostro progetto. Ma come testiamo ogni piccola modifica sul codice HTML? Facendo le modifiche al codice HTML, CSS, JS, rigenerando il file compresso con Gulp per poi caricarlo sul Modulo ESP32-DEV, testare, modificare e procedere per cicli di debugging successivi finché non riusciamo ad ottenere il risultato desiderato? Certo, nessuno ce lo vieta, ma sarebbe una follia. Anche perché debuggare il codice js compresso è veramente frustrante, se non quasi impossibile. Vediamo quindi di trovare una soluzione anche a questo problema...

Dopotutto sono i problemi che ci fanno imparare qualcosa: se filasse tutto liscio, che gusto ci sarebbe!?

Supponiamo, per il momento, che il codice della parte Arduino (ESP32-DEV) sia già consolidato. Possiamo quindi usare il modulo ESP32-DEV per la comunicazione tra la stampante 3D ed il codice JS che gira sul Browser, ma invece di usare il modulo ESP32-DEV per ospitare la pagina *index.html.gz* compressa, è meglio usare un normale web server e lavorare direttamente sui sorgenti del progetto (quindi la cartella *www*) non sul file compresso.

Per fare questo potremmo usare la versione personale del famoso web server di Microsoft, IISExpress che abbiamo tutti nel nostro PC Windows10. Ma c'è una soluzione migliore. Useremo Node.js! Questo ci permetterà di evitare le complesse configurazioni che sarebbero richieste su IISExpress e inoltre abbiamo già installato Node.js per usare Gulp e quindi una parte del lavoro è già fatto. Avremo bisogno di due server Node.js, un server Web vero e proprio ed un server Proxy. Il server Web Node.js conterrà tutto il sorgente HTML,CSS,JS (cartella *www*) del progetto e lo renderà disponibile per un cliente Web (il nostro browser) il server starà in ascolto sulla porta 8080. Il Server Proxy Node.js invece è un semplice Proxy (intermediario) che starà in ascolto sulla porta 82 e ridirigerà il traffico nei due sensi alla porta 80 del server web su cui gira ESP32-DEV. Noi non interagiranno direttamente con il Proxy, ma sarà il codice JS nelle varie pagine che lo chiameranno in causa. Quindi lo scopo del Proxy è quello di fare comunicare la pagina Web della Dashboard con il dispositivo ESP32-DEV tramite websocket.

In modo tale di comunicare in effetti con la stampante3D. Mentre il server Web Node.js è impiegato unicamente per ospitare tutto il sorgente web ed avere quindi la possibilità di poter editare direttamente il sorgente, html, js, css e vederne la modifica in modo quasi immediato sul Browser. Inoltre non avendo il codice minimizzato in un unico file, il procedimento di debugging sarà molto più agevole. Istanziare dei server con Node.js è un gioco da ragazzi. È sufficiente (previa installazione di Node.js) creare un semplice file di testo ed il gioco è fatto. Vediamo i vari passi nel dettaglio. Nelle cartella NodeJS del progetto abbiamo due sotto cartelle, una si chiama *HTTP_Server*, ed una *HTTP_Proxy*. Nella cartella avremo tutta la cartella *www* del sorgente, poi avremo un file *server.js* che è il file che definisce il funzionamento del server web ed un file *run_server.bat* che dovremo usare per lanciare, quando pronti, il server Web Node.js. Il file *server.js* necessita per funzionare di una serie di moduli software di Node.js da "installare" localmente.

```

1  /*
2  3  Il traffico sulla porta 82 locale viene ridiretto verso il dispositivo ESP32 sulla porta 80
4
5  Esempio:
6  Server Nodejs      Server Esp32 Fisico
7  192.168.1.232:82   ->>  192.168.1.188:80
8
9  */
10
11 var fs = require('fs');
12 var express = require('express');
13 var cors = require('cors');
14 var ip = require("ip");
15
16
17
18 // Porta su cui sta in ascolto il Proxy
19 var origin_port = 82;
20
21
22 // Indirizzo del dispositivo fisico ESP32
23 var target_url = "http://192.168.1.188";
24
25
26
27
28
29 var js_content = "var esp32_address = \"http://\" + ip.address() + ":" + origin_port + "\";";
30 try {
31   var server_dir = __dirname.replace("HTTP_Proxy", "HTTP_Server");
32   const data = fs.writeFileSync(server_dir + '/www/js/Proxy_Address.js', js_content)
33 } catch (err) {
34   console.error(err)
35 }
36
37
38
39 console.log ("HTTP proxy up and running");
40 console.log ("listening on port: " + origin_port );
41 console.log ("Target url: " + target_url );
42
43
44 var proxy = require('http-proxy-middleware');
45
46 var app = express();
47
48 app.use(cors());
49
50
51 var proxyOpts = {
52   target: target_url,
53   onProxyReq: function onProxyReq(proxyReq, req, res) {
54     // Log outbound request to remote target
55     console.log('---> ', req.method, req.path, '-->', proxyReq.baseUrl + proxyReq.path);
56   },
57   onError: function onError(err, req, res) {

```

 **Fig. 6**
Impostazione dell'IP.

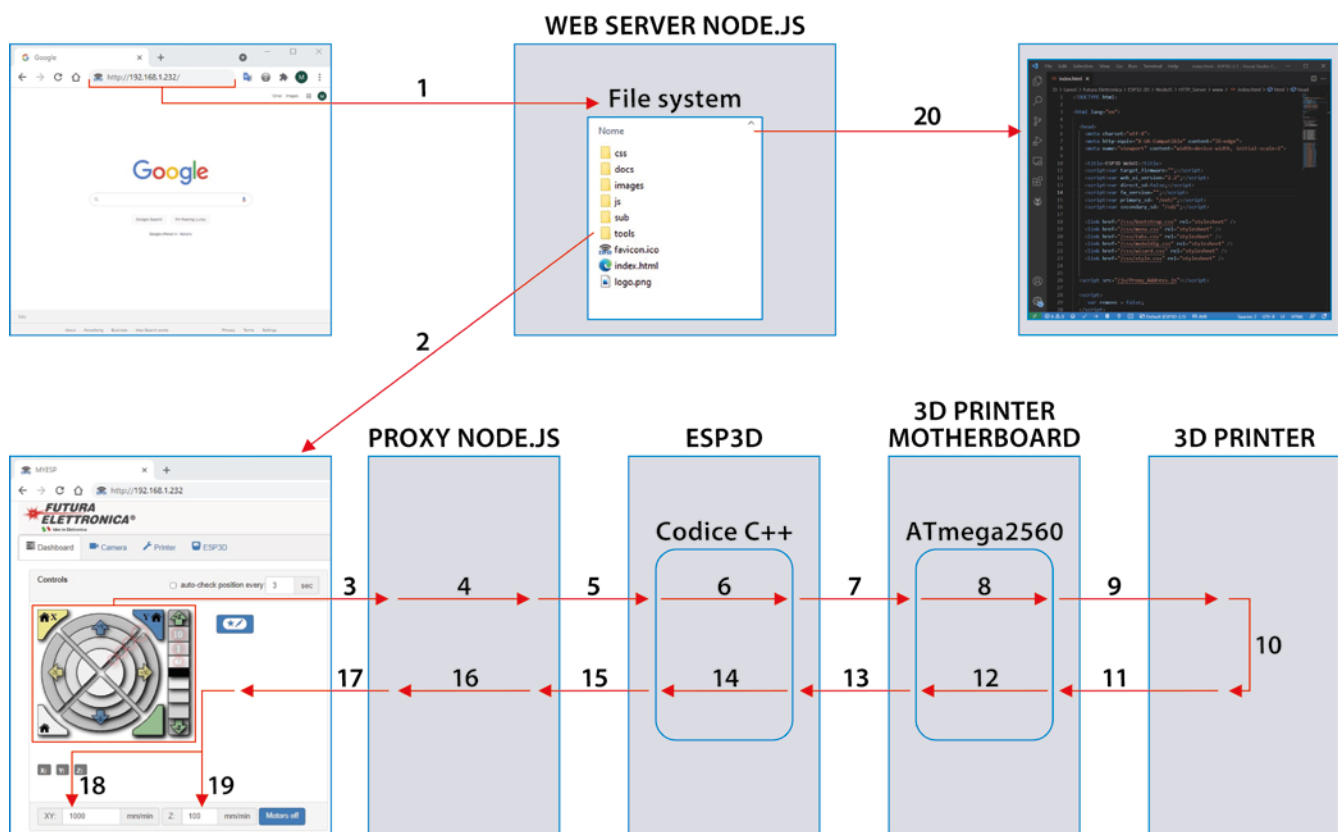


Fig. 7
Schema di
test e debug.

Questi file saranno posizionati nella cartella *node_modules* che attualmente è non presente o vuota. Per popolare tale cartella con i file necessari è sufficiente lanciare dalla cartella *HTTP_Server* il file *get_node_modules.bat* il quale provvede a scaricare dai repository di node.js tutti i file necessari. Ci vorrà un po' di tempo quindi pazientate. Analogamente, nella cartella *HTTP_Proxy*, abbiamo il file *proxy.js* che è il file che definisce il funzionamento del Proxy web e un file *run_proxy.bat* che dovremo usare per lanciare, quando pronti il Proxy Web. Il file *proxy.js* necessita per funzionare di una serie di moduli software di Node.js da "installare" localmente. Questi file saranno posizionati nella cartella *node_modules* che attualmente è non presente o vuota. Per popolare tale cartella con i file aggiornati e necessari è sufficiente lanciare dalla cartella *HTTP_Proxy* il file *get_node_modules.bat* il quale provvede a scaricare dai repository di node.js tutti i file necessari. Anche in questo caso ci vorrà un po' di tempo.

CONFIGURAZIONE DEL PROXY

Abbiamo cercato di ridurre al minimo le configurazioni da apportare ai file sia del server web Node.js

che del Server Proxy Node.js. L'unica cosa che è necessario impostare sul proxy è l'indirizzo IP del modulo fisico ESP32-DEV sul quale gira il motore di ESP3D. In **Fig. 6** (nella pagina precedente) potete vedere dove impostare tale valore nel file *Proxy.js*. Nel nostro caso il valore dell'indirizzo IP è 192.168.1.188.

Ricordiamo che se abbiamo collegato il modulo ESP3D alla stampante 3D, dovremmo vedere l'IP acquisito dal modulo ESP3D nel display della stampante 3D. Altrimenti è necessario utilizzare dei software di scanning di rete quali "Advanced IP scanner".

Naturalmente la creazione del file *index.html.gz* tramite Gulp deve essere eseguita solamente al termine del processo di test attuato tramite il web server Node.js. Quando siamo sicuri di aver terminato la fase di modifiche del codice e tutto funziona in modo corretto.

Nella **Fig. 7**, vediamo come utilizzare i due server Node.js ed il modulo ESP32-DEV per testare e debuggare le modifiche della interfaccia Web: al punto (1) apriamo il browser all'url *http://xxx.xxx.xxx.xxx* che corrisponde all'IP del server Node.js. Siccome il server Node.js gira sul nostro PC.

Nella schermata in cui viene mandato in esecuzione il server Node.js viene riportato il valore dell'indirizzo IP che ci interessa.

Al punto (2) il web server Node.js risponde alla richiesta del browser andando a prelevare nel file system, tutte le risorse di cui il browser necessita per renderizzare la DashBoard di controllo e quindi il browser visualizza l'interfaccia web con tutti i controlli.

L'utente clicca con il mouse sul controllo di spostamento dell'estrusore nella posizione Home ed il codice JS dirige la richiesta al Proxy Node.js (3).

Ai punti 4 e 5 il server Proxy Node.js elabora la richiesta e ridirige la richiesta al modulo ESP3D, il quale apre e mantiene il canale Websocket aperto.

Al punto (6) la richiesta viene elaborata dal codice

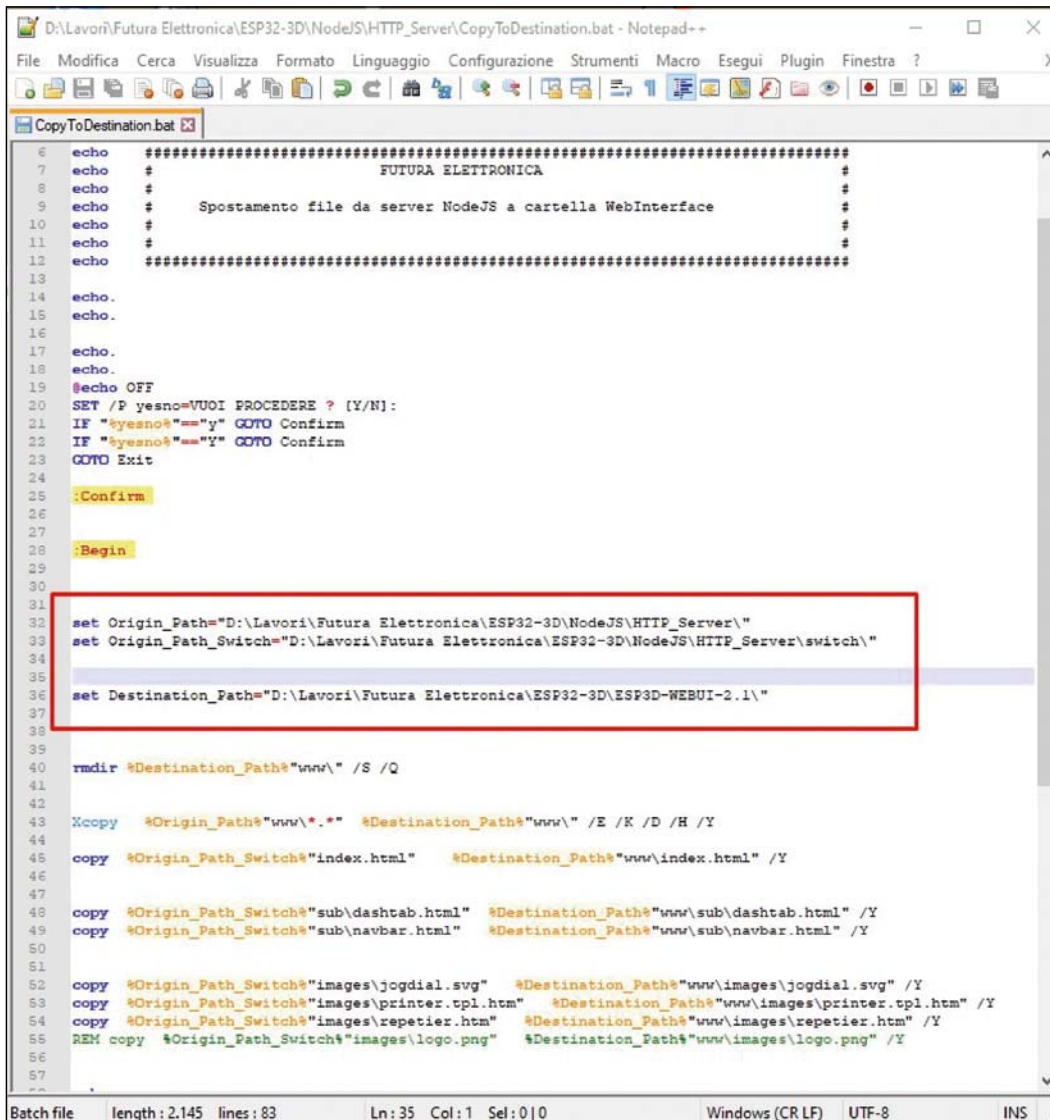
C++ all'interno del modulo ESP3D per decidere cosa fare.

Viene quindi inviato un comando tramite la porta USART alla Motherboard della stampante 3D (7); il processore ATmega2556 elabora la richiesta (8) e quindi invia i relativi comandi di movimento alla stampante 3D (9).

Al punto (10) la stampante 3D esegue i comandi di movimento.

Viene quindi ritornato un comando al processore ATmega2556 (11). Il processore ATmega2556 elabora la risposta (12).

Al punto (13) il comando viene trasferito tramite USART al modulo ESP3D. Il codice C++ del modulo ESP3D elabora quindi la risposta (14) ed invia la richiesta al Proxy Node.js che la elabora e la ridirige al



```
6 echo #####
7 echo #
8 echo #
9 echo #
10 echo #
11 echo #
12 echo #####
13
14 echo.
15 echo.
16
17 echo.
18 echo.
19 @echo OFF
20 SET /P yesno=VUOI PROCEDERE ? [Y/N]:
21 IF "%yesno%"=="y" GOTO Confirm
22 IF "%yesno%"=="Y" GOTO Confirm
23 GOTO Exit
24
25 :Confirm
26
27
28 :Begin
29
30
31
32 set Origin_Path="D:\Lavori\Futura Elettronica\ESP32-3D\NodeJS\HTTP_Server\"
33 set Origin_Path_Switch="D:\Lavori\Futura Elettronica\ESP32-3D\NodeJS\HTTP_Server\switch\"
34
35 set Destination_Path="D:\Lavori\Futura Elettronica\ESP32-3D\ESP3D-WEBUI-2.1\"
36
37
38
39
40 rmdir %Destination_Path%\www\ /S /Q
41
42
43 Xcopy %Origin_Path%\www\*.* %Destination_Path%\www\ /E /K /D /H /Y
44
45 copy %Origin_Path_Switch%\index.html %Destination_Path%\www\index.html /Y
46
47
48 copy %Origin_Path_Switch%\sub\dashtab.html %Destination_Path%\www\sub\dashtab.html /Y
49 copy %Origin_Path_Switch%\sub\navbar.html %Destination_Path%\www\sub\navbar.html /Y
50
51
52 copy %Origin_Path_Switch%\images\jogdial.svg %Destination_Path%\www\images\jogdial.svg /Y
53 copy %Origin_Path_Switch%\images\printer.tpl.htm %Destination_Path%\www\images\printer.tpl.htm /Y
54 copy %Origin_Path_Switch%\images\repetier.htm %Destination_Path%\www\images\repetier.htm /Y
55 REM copy %Origin_Path_Switch%\images\logo.png %Destination_Path%\www\images\logo.png /Y
56
57
58
```

Fig. 8
Le variabili da personalizzare.

Fig. 9
I file
componenti del
codice Arduino.

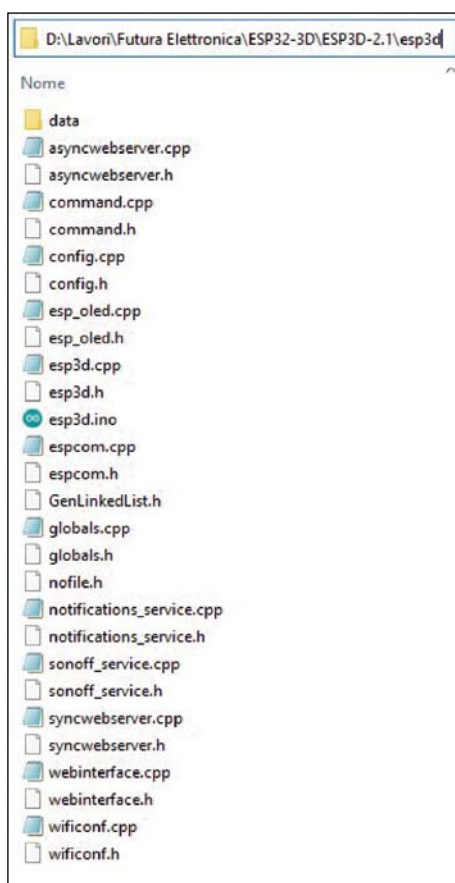


Fig. 10
Impostazioni
scheda nell'IDE.

Browser Web tramite il canale Websocket (14-15) con cui vengono inviati dei dati alla interfaccia WEB del Browser.

Il codice Javascript nel Browser elabora i dati ricevuti dal Websocket ed aggiorna le due text box che riportano la posizione X, Y, Z dell'Estrusore (17 e 18).

Infine l'utente apre L'IDE e modifica il codice HTML o Javascript o CSS per adattare il funzionamento dell'interfaccia alle proprie necessità o per correggere un eventuale errore.

Per testare le modifiche al codice HTML e Javascript o CSS, non è quindi necessario ricaricare nulla sul modulo ESP3D, ma semplicemente si lavora sui file non compressi presenti nella cartella www del server Web Nodejs.

Solo quando funzionerà tutto si passerà alla generazione del file compresso con Gulp.

RUOLO DEL PROXY

A qualcuno sarà venuto spontaneo domandarsi a cosa serve il Proxy Node.js. Ebbene, se aprite lo

script che lo definisce, vedrete che ci sono pochissime righe di codice e non compie alcuna elaborazione, si limita a fare da intermediario tra le richieste del Browser e le risposte dirette al Browser originate dal modulo ESP3D.

La risposta è da ricercarsi su una imposizione di sicurezza (CORS) imposta dai moderni browser, i quali impediscono che le richieste fatte dal browser siano dirette ad indirizzi IP diversi da quelli che hanno originato la richiesta. In parole povere, noi riceviamo le pagine WEB da un server con un certo IP (che si trova sul nostro PC) e senza l'intermediazione del proxy dovremmo ridirigere alcune richieste e ricevere delle risposte da un IP diverso, che nello specifico è quello del modulo ESP32-DEV. Questa cosa non è permessa dalle politiche restrittive CORS e quindi tutto il nostro sistema non potrebbe funzionare. Usando il proxy invece, sia le pagine che otteniamo dal Server Node.js che le risposte dal modulo ESP3D appaiono sempre come provenienti dallo stesso IP, perché il server Proxy gira sempre sul nostro PC ed avrà anch'esso lo stesso IP del server

web Node.js. In questo modo abbiamo bypassato la restrizione CORS e possiamo lavorare tranquilli e senza intoppi.

Al termine della fase di debugging, se tutto ha funzionato a dovere, quello che dovremo fare è spostare il contenuto della cartella *www* presente dentro il server Node.js nella cartella dove l'esecuzione dello script di GULP ci permetterà di avere il file compresso *index.html.gz*.

Per agevolarvi, abbiamo automatizzato anche questa procedura. È sufficiente cliccare sul file *CopyTo-Destination.bat* per ottenere il risultato desiderato. Nella **Fig. 8** è possibile vedere le tre variabili che devono essere personalizzate.

In queste tre variabili dovreste inserire i vostri percorsi delle cartelle.

Ora qui dobbiamo aprire una piccola parentesi. Per poter utilizzare il server Web di Node.js abbiamo dovuto apportare delle piccole modifiche ad alcuni file contenuti all'interno della cartella *www*, in primis al file *index.html*.

Queste modifiche sono state necessarie in virtù della diversa tecnologia usata per servire pagine web nel modulo ESP32-DEV e nel server web Node.js. Il server web del modulo ESP32-DEV usa una sintassi diversa per includere i file e quindi abbiamo dovuto riscrivere alcuni file per adattarli al funzionamento del server Web di Node.js. I file originali che ci serviranno per il processo di compressione mediante GULP sono stati salvati dentro una cartella di appoggio che si chiama *switch*.

L'esecuzione del file *CopyToDestination.bat* per prima cosa rimuove la cartella *www* presente nel percorso di destinazione, poi il contenuto della cartella *www* all'interno del server Node.js verrà copiato nella cartella di destinazione di GULP e verranno ripristinati i file originali presenti nella cartella *switch* nelle rispettive posizioni.

CODICE ARDUINO DEL MODULO ESP3D

Come potete vedere in **Fig. 9**, anche il codice C++ che gira sul modulo ESP3D è stato suddiviso in più parti, in modo da renderne più agevole la manutenzione. Il file principale è *esp3d.ini* e aprendolo con l'IDE di Arduino avremo una serie di Tab nella interfaccia, ognuno per ogni file incluso nel progetto. È importante che la Board Utilizzata sia "ESP32 Dev Module" e che le opzioni selezionate siano quelle elencate nella **Fig. 10**.

Nella cartella Library visibile in **Fig. 11**, abbiamo raccolto tutte le librerie necessarie per la compilazione del progetto ESP3d. In questo articolo non ci soffermeremo sulle istruzioni per importare le librerie

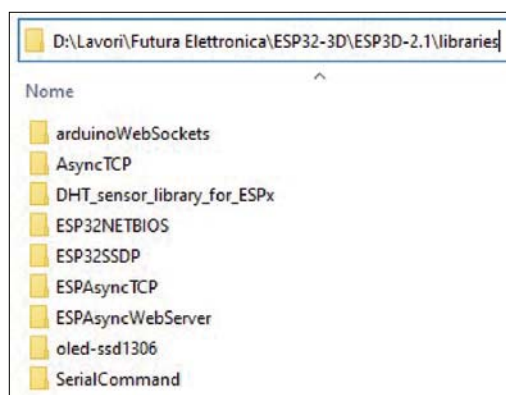


Fig. 11
Le librerie
dell'ESP32.

ed altri dettagli minori perché abbiamo già trattato queste informazioni in altri articoli riguardanti l'ESP32 e quindi vi rimandiamo alla consultazione di tali articoli.

CONCLUSIONI

Bene, in questa puntata vi abbiamo spiegato come preparare l'ambiente per poter apportare delle modifiche al codice del dispositivo ESP3D. Ci siamo dedicati particolarmente alla parte relativa all'interfaccia grafica implementata in HTML e poi abbiamo indicato anche il codice Arduino per il quale, a parte la complessità dello stesso, non ci sono accorgimenti specifici da prendere in considerazione. Come al solito, il codice e le librerie utilizzate nel progetto descritto in questo articolo sono disponibili al seguente url di Github: <https://github.com/open-electronics/ESP3D>; per ragioni di spazio, qui non ci è stato possibile riportare il listato per intero. Nella prossima puntata vedremo come configurare le Notifiche. Ciò detto, non ci resta che augurarvi buon lavoro. ■



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. 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

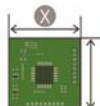
1 Layer

2 Layers

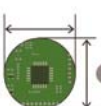
4 Layers

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

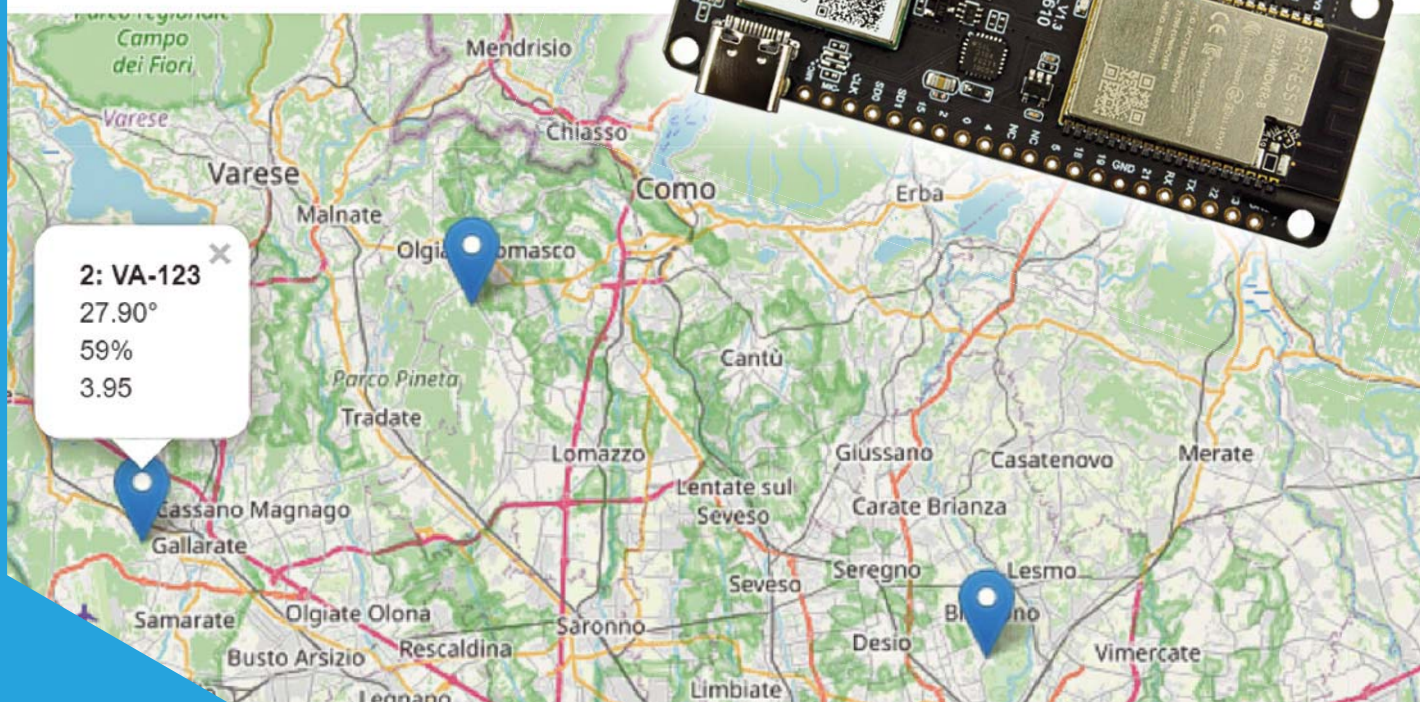
Futura Group srl
Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775

**FUTURA
ELETTRONICA®**

METEO Net Map

T-CALL

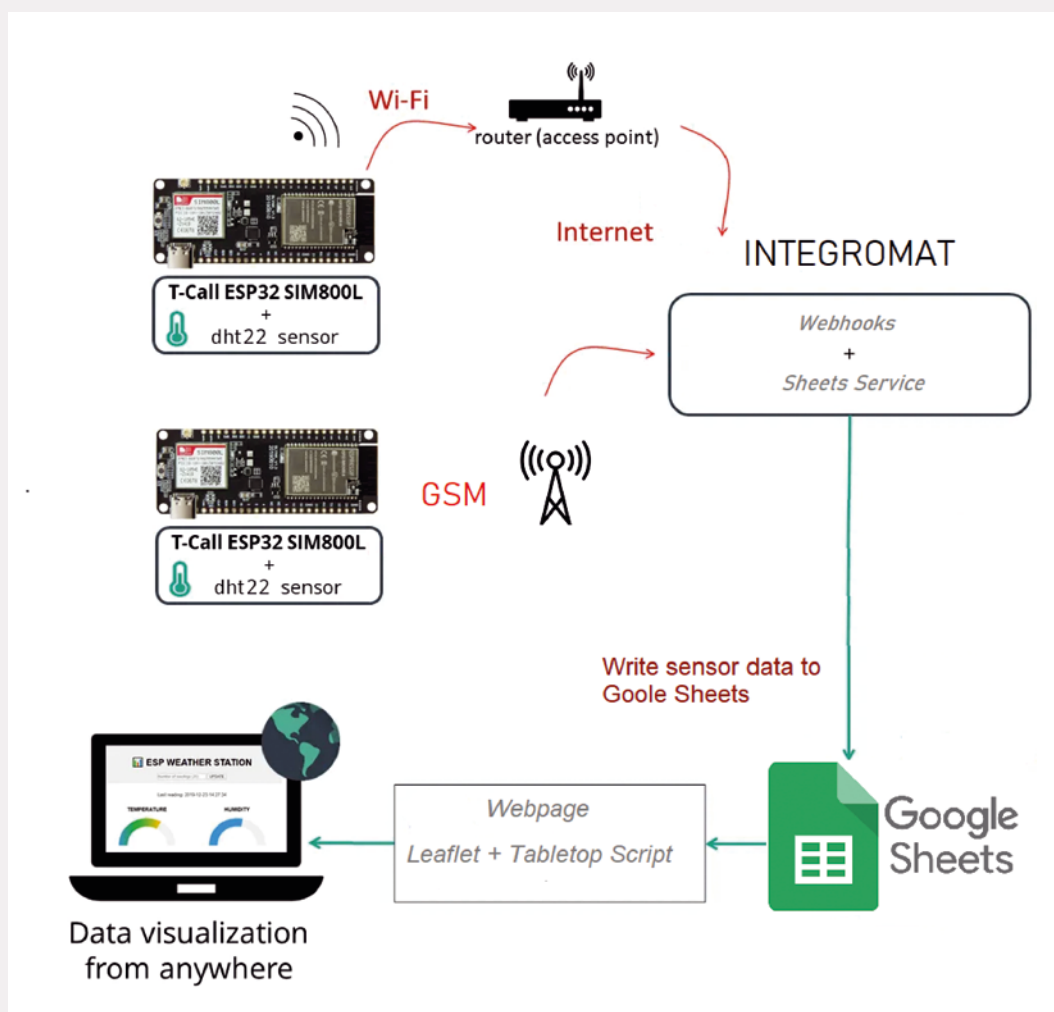
dell'ing. **MIRCO SEGATELLO**



Permette di geolocalizzare in un'unica mappa la posizione di più dispositivi IOT connessi in rete tramite WiFi o GSM con la possibilità di visualizzare i parametri ambientali rilevati da ciascuno di essi.

Eccoci nuovamente a parlare di dispositivi IoT e mappe: lo facciamo riprendendo l'articolo pubblicato nel numero 261 della nostra rivista dedicato a Meteo-NetMap. In quell'articolo avevamo visto come utilizzare un modulo NodeMCU (con a bordo un microcontrollore ESP8266) per inviare dati ambientali verso un foglio di calcolo online di Google. La cosa di particolare interesse di quel progetto era che più dispositivi potevano interagire con il foglio di calcolo, popolandolo ciascuno con i propri dati. Il foglio di calcolo era poi disponibile per ulteriori elaborazioni e comunque visibile da qualsiasi web browser. Ma non ci siamo limitati a questo, perché il passo successivo è stato mettere a disposizione questi dati in una mappa che potesse visualizzare la posizione dei vari dispositivi e, a richiesta, visualizzare gli ultimi dati acquisiti.

Fig. 1
Schema
rappresentativo del
sistema.



Il nome stesso del progetto -MeteoNetMap- indica che il principale utilizzo di questo sistema è la gestione di più stazioni meteo distribuite nel territorio. Sempre nell'articolo del MeteoNetMap avevamo messo in evidenza due criticità: la prima era riguardante l'accesso dei dispositivi alla rete, mentre la seconda era dovuta alla gestione dei livelli di sicurezza di Google: nel primo caso, siccome i microcontrollori ESP8266 dispongono solo di un modulo WiFi, il sistema proposto richiedeva che ciascun dispositivo potesse accedere ad una rete WiFi. La seconda criticità nasce dal fatto che affinché ciascun dispositivo potesse scrivere direttamente in un foglio di calcolo della Google, era necessario scrivere e rendere attivo un apposito script: un'operazione non molto agevole. In questo articolo vi spiegheremo come abbiamo risolto queste criticità, offrendo nuovi spunti allo sviluppo del progetto, la cui impostazione di sistema rimane sostanzialmente la stessa, ed è rappresentata in **Fig. 1**.

PREDISPOSIZIONE FOGLIO DI CALCOLO

Procediamo ora con la predisposizione del foglio di calcolo che riceverà i dati dai vari sensori. Accedete ai servizi della Google e create un nuovo foglio di calcolo (sheet) ed assegnategli un nome a vostro piacere, nel nostro caso MeteoNetMap. Usate la prima riga per identificare le varie colonne assegnando a ciascuna di esse un identificativo come visibile in **Fig. 10**; affinché sia compatibile con il codice d'esempio esposto in queste pagine, è importante rispettare i nomi con le dovute maiuscole e minuscole.

DISPOSITIVI IOT

Questa volta abbiamo optato per una compatta board basata sul modulo WiFi ESP32 e sul modulo GSM GPRS SIM800L chiamata TTGO T-Call (nello specifico la versione 1.4) ed è distribuita dalla Futura Elettronica (www.futurashop.it) con codice HR1761.

Dispone di memoria Flash da 4 MB, PSRAM da 8 MB, Wi-Fi, Bluetooth, 18 GPIO, chip di gestione dell'alimentazione (IP5306) con circuito di carica per batteria Li-ion/Li-Po e convertitore USB-seriale CP2104 per la programmazione, perfetta per la nostra applicazione. Il modulo GSP/GPRS SIM800L integrato era proprio quello che ci mancava per assicurare la connettività laddove non fosse disponibile una rete WiFi. Alla scheda colleghiamo il sensore ambientale DHT22 (codice prodotto DHT22); il positivo va connesso al pin 5Vout, il segnale al pin 18 e la massa a GND, come visibile in **Fig. 2**.

Se la lettura dei valori dovesse presentare dei problemi consigliamo di aggiungere una resistenza di pull-up del valore di 10K connessa tra il positivo ed il pin del segnale ed eventualmente un condensatore di bypass da 100 nF. Questa board viene fornita con un'antenna adesiva ma consigliamo sin da subito l'utilizzo dell'antenna a stilo più performante, anch'essa fornita nella confezione. La scheda può essere alimentata tramite la porta USB-C di cui è dotata ma offre altresì la possibilità di essere alimentata tramite una singola cella al litio, di cui è possibile leggere la tensione.

SETUP DEL PROGETTO

Prepariamo l'ambiente di lavoro Arduino per poter programmare schede basate su microcontrollori ESP32, pertanto accediamo alla sezione impostazioni dal menu File dell'IDE di Arduino e nella sezione "URL aggiuntive per il gestore schede"

inseriamo il link https://dl.espressif.com/dl/package_esp32_index.json, nel caso vi siano presenti altri link separateli con una virgola. Dal menù Strumenti cliccate su "Schede" e poi su "Gestione schede" e cercate quella di nome *Esp32 by Espressif Systems* ed installatela. Dal menu Strumenti cliccate su "Gestione librerie" ed installate la libreria *DHT sensor library by Adafruit* e la libreria *TinyGSM by Volodymyr Shymanskyi*. Per maggiori dettagli sull'uso del sensore ambientale DH22 potete leggere il tutorial al link <https://learn.adafruit.com/dht>. Per la programmazione della scheda in "Strumenti-Scheda" selezionate *ESP32 Wrover module*, lasciando tutte le altre impostazioni ai valori predefiniti. Abbiamo previsto due distinti codici: uno che utilizza il modulo WiFi per connettersi alla rete ed uno che usa il modulo GSM SIM800L. In sintesi, il nostro dispositivo dovrà leggere i dati dal sensore ambientale connettersi alla rete WiFi o GSM ed inviare i dati al foglio di calcolo tramite un opportuno servizio web, quindi predisporre in modalità a basso consumo sino al prossimo invio. Il listato che sfrutta la connessione WiFi si chiama *TCALL_IntegromatWiFi.ino* di cui qui di seguito riportiamo la parte necessaria alla definizione dei parametri di funzionamento:

```
const char *ssid = "SSID";
const char *password = "PASSWORD";
const char* host = "hook.integromat.com";
String integromatUrl = "integromatUrl";
```

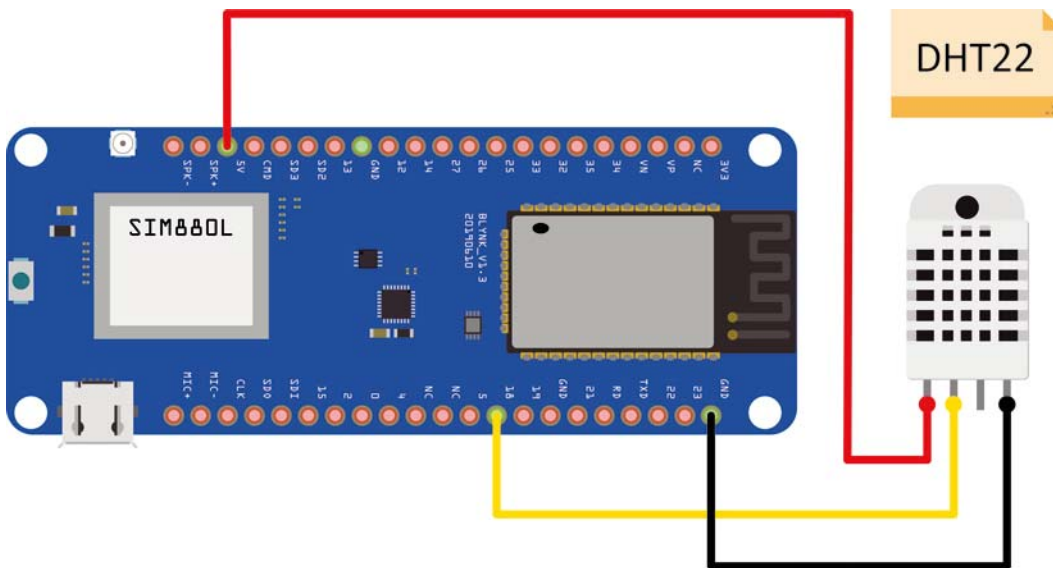


Fig. 2
Configurazione hardware.

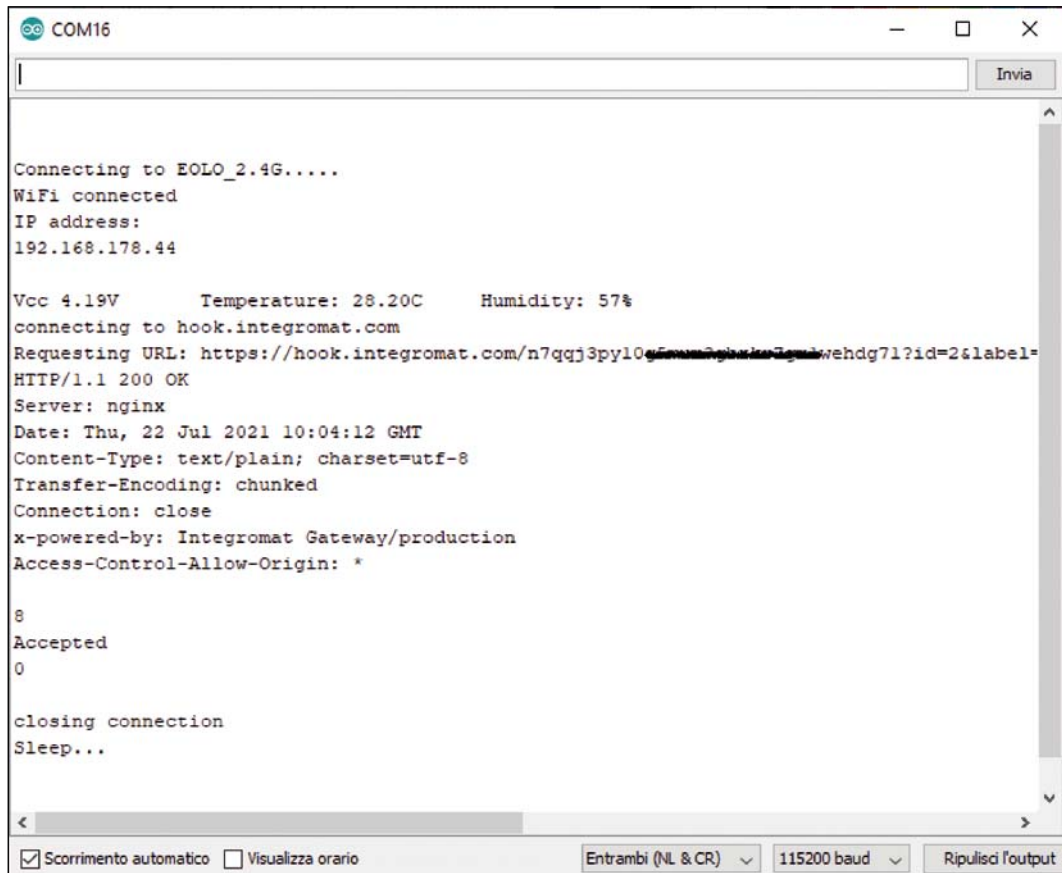
Listato 1

```
if (!client.connect(host, httpPort))
{
  Serial.println("connection failed");
  return;
}
url = integromatUrl + "?id=" + stationID + "&label=" + stationLabel + "&Latitude=" + latitude + "&Longitude=" + longitude + "&Power=" + batteryLevel + "&Temperature=" + temperature + "&Humidity=" + humidity;
client.print(String("GET ") + url + " HTTP/1.1\r\n" + "Host: " + host + "\r\n" + "Connection: close\r\n\r\n");
String section="header";
while(client.available())
{
  String line = client.readStringUntil('\r');
  Serial.print(line);
}
```

```
String stationID = "2";
String stationLabel = "VA-123";
String latitude = "45.6605746";
String longitude = "8.7578234";
#define TIME_TO_SLEEP 3600
```

I campi *ssid* e *password* fanno riferimento alle credenziali di accesso alla vostra rete WiFi, *inte-*

gromatUrl è l'indirizzo fornito da Integromat di cui parleremo più approfonditamente successivamente. Il parametro *stationID* specifica quale riga del foglio di calcolo è assegnato al dispositivo (deve essere univoco per ciascun dispositivo) mentre *stationLabel* è semplicemente il nome assegnato alla stazione. I parametri *latitude* e *longitude* specificano la posizione nella quale sarà installata la



```
COM16
Connecting to EOLO_2.4G.....
WiFi connected
IP address:
192.168.178.44

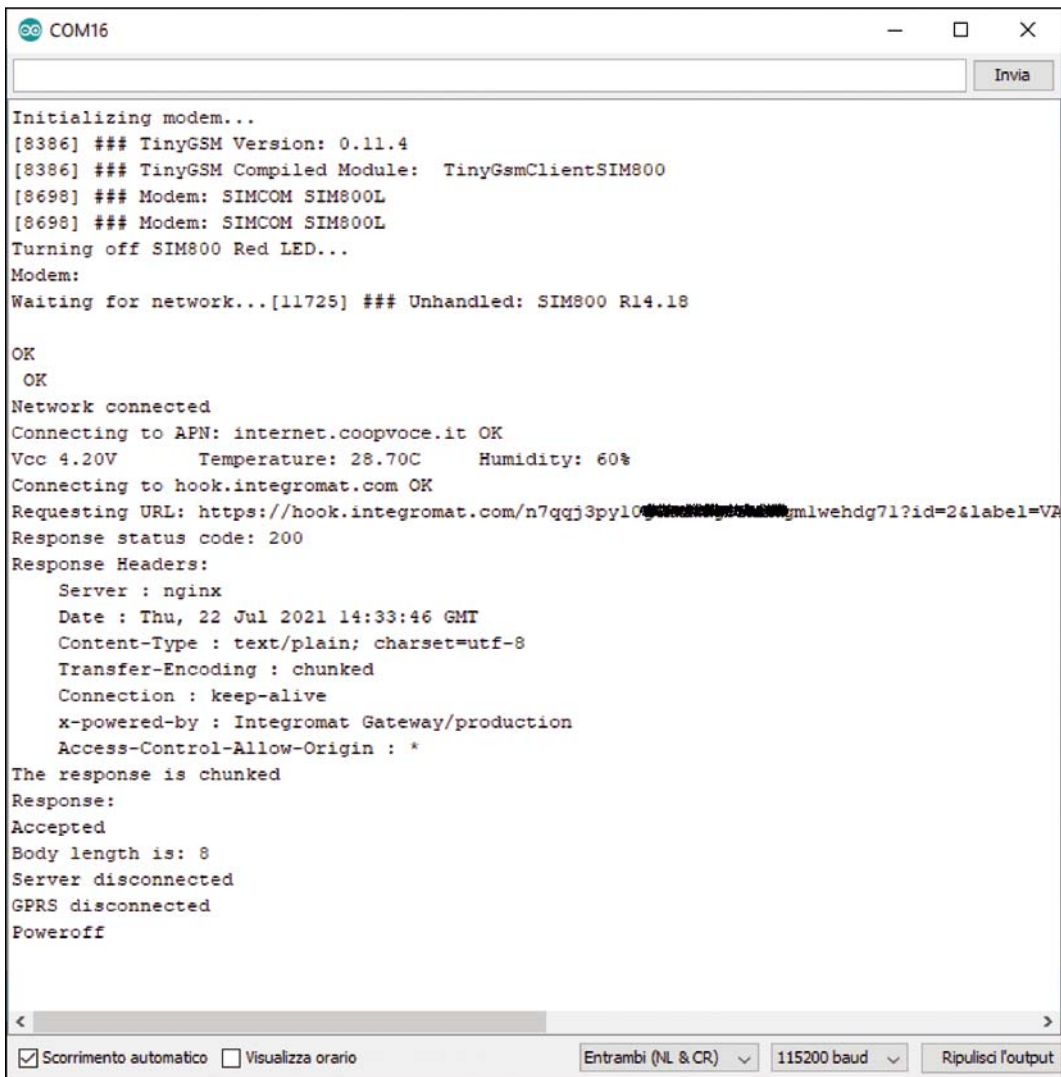
Vcc 4.19V      Temperature: 28.20C      Humidity: 57%
connecting to hook.integromat.com
Requesting URL: https://hook.integromat.com/n7qqj3pyl0g5mua2ghzku7gwhdg7l?id=2&label=
HTTP/1.1 200 OK
Server: nginx
Date: Thu, 22 Jul 2021 10:04:12 GMT
Content-Type: text/plain; charset=utf-8
Transfer-Encoding: chunked
Connection: close
x-powered-by: Integromat Gateway/production
Access-Control-Allow-Origin: *

8
Accepted
0

closing connection
Sleep...
```

Fig. 3
Schermata di
Serial monitor
per lo sketch con
connessione WiFi.





```
COM16
Invia

Initializing modem...
[8386] ### TinyGSM Version: 0.11.4
[8386] ### TinyGSM Compiled Module: TinyGsmClientSIM800
[8698] ### Modem: SIMCOM SIM800L
[8698] ### Modem: SIMCOM SIM800L
Turning off SIM800 Red LED...
Modem:
Waiting for network...[11725] ### Unhandled: SIM800 R14.18

OK
OK
Network connected
Connecting to APN: internet.coopvoce.it OK
Vcc 4.20V      Temperature: 28.70C      Humidity: 60%
Connecting to hook.integromat.com OK
Requesting URL: https://hook.integromat.com/n7qqj3pyl0g4wv2m1wehdg71?id=2&label=VA
Response status code: 200
Response Headers:
  Server : nginx
  Date : Thu, 22 Jul 2021 14:33:46 GMT
  Content-Type : text/plain; charset=utf-8
  Transfer-Encoding : chunked
  Connection : keep-alive
  x-powered-by : Integromat Gateway/production
  Access-Control-Allow-Origin : *
The response is chunked
Response:
Accepted
Body length is: 8
Server disconnected
GPRS disconnected
Poweroff

< [ ] >
☒ Scorrimto automatico ☐ Visualizza orario
Entrambi (NL & CR) 115200 baud Ripulisci l'output
```

Fig. 4
Schermata di Serial monitor per lo sketch con connessione GSM.

stazione, valori facilmente ricavabili tramite Google Maps, il parametro `TIME_TO_SLEEP` specifica ogni quanto tempo (in secondi) il dispositivo si attiverà per inviare i dati. Nel **Listato 1** riportiamo la parte di codice che si occupa di costruire la stringa ed inviarla al servizio web di Integromat, mentre in **Fig. 3** la schermata di Serial Monitor di Arduino quando lo sketch è in esecuzione. Un secondo sketch, di nome `TCALL_IntegromatG-`

`SM.ino`, prevede invece l'invio dei dati tramite rete GSM; riportiamo nel **Listato 2** la parte relativa alla definizione dei parametri per la connessione alla rete, gli altri parametri sono identici al caso precedente. Nel nostro caso la SIM utilizzata non aveva una password di accesso quindi l'unico campo compilato è il nome dell'apn (Access Point Name) fornito dall'operatore telefonico e che trovate nelle

↓ Listato 2

```
const char apn[] = "internet.coopvoce.it"; // Your APN
const char gprsUser[] = ""; // User
const char gprsPass[] = ""; // Password
const char simPIN[] = ""; // SIM card PIN code, if any
```

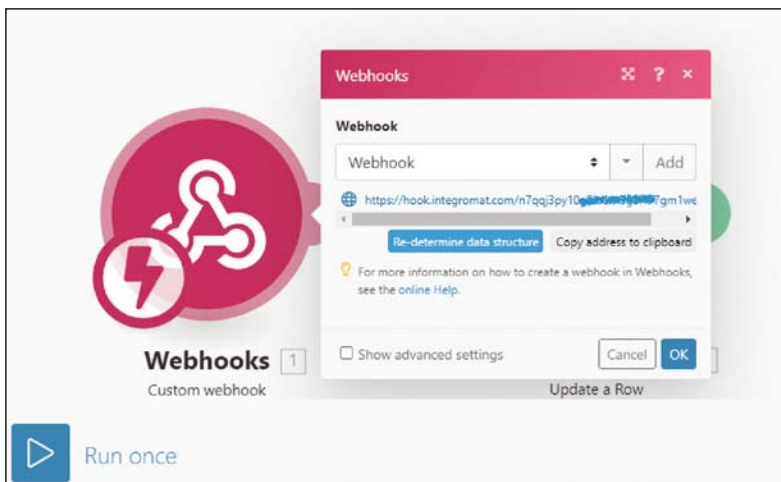


Fig. 5
Servizio
Webhook di
Integromat.

impostazioni dello smartphone. Il listato completo è stato perfezionato partendo dagli esempi forniti dal costruttore della scheda T-Call e dagli esempi forniti dalla libreria TinyGSM utilizzata per costruire la richiesta https da inviare a Integromat.

Il portale Integromat utilizza la tecnologia SSL (dall'inglese Secure Sockets Layer) che garantisce la sicurezza di una connessione a Internet e protegge i dati sensibili scambiati fra due sistemi, pertanto la richiesta deve giungere con estensione HTTPS (Hyper Text Transfer Protocol Secure); questa tecnologia viene abilitata nello sketch dalla riga `TinyGsmClientSecure client(modem)`, ovviamente il modulo GSM deve supportare questa tecnologia nativamente.

Rispetto a prima, il listato è più complesso dovendo gestire il modulo GSM che deve essere attivato e configurato, successivamente occorre gestire la connessione alla rete GSM e poi la connessione al server Integromat per poi inviare i dati misurati dal sensore, per chiarezza lo sketch è suddiviso in funzioni specifiche:

- `setupModem`, provvede all'attivazione del modulo SIM800L e alla sua configurazione di avvio;

- `connectToNet`, parte del programma adibita alla connessione alla rete GSM;
- `sendDataToServer`, questa provvede alla connessione al server di Integromat e all'invio dei dati del sensore;
- `modem.poweroff`, provvede a disattivare il modem.

Abbiamo previsto un buon debug, con una serie di informazioni sullo stato di avanzamento delle operazioni; in **Fig. 4** è proposta la schermata di Serial Monitor con l'esecuzione dello sketch.

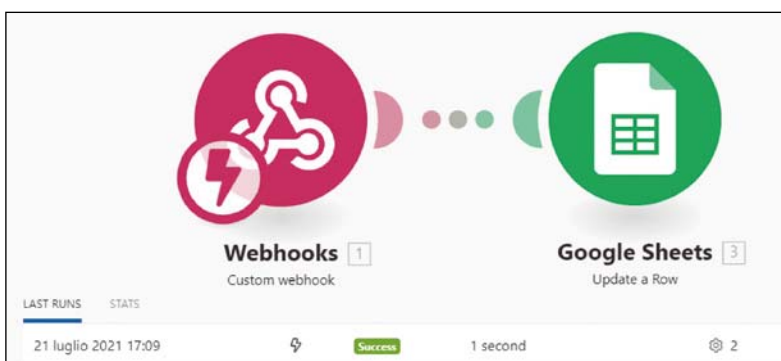
Il listato si conclude con le due righe necessarie a porre il microcontrollore ESP32 in modalità deep sleep per garantire il minor consumo possibile, rimane comunque attivo l'integrato IP5306 che nella scheda ha il compito di gestire l'alimentazione in totale, multimetro alla mano, abbiamo misurato un assorbimento di circa 0,5 mA. Da alcune prove eseguite abbiamo misurato un assorbimento medio di circa 200 mA durante le operazioni sopra elencate che durano circa 30 secondi per cui, ipotizzando un invio ogni ora una batteria da 1.000 mA/h garantirebbe qualche giorno di funzionamento. In entrambi gli sketch abbiamo previsto la lettura della tensione ai capi della batteria, resa disponibile tramite un partitore al pin 35 del microcontrollore; se quest'ultima non fosse presente si leggerebbe sempre una tensione di 4,2 volt fornita dal circuito di ricarica.

IL SERVIZIO INTEGROMAT

Per semplificare l'accesso al foglio di calcolo da parte di dispositivi esterni sfruttiamo una piattaforma su base cloud che permette di integrare ed automatizzare i principali strumenti web.

La prima che abbiamo testato è IFTTT, già usata in precedenti progetti che però è assai limitata nelle funzioni (almeno per la versione free), pertanto abbiamo preferito optare per Integromat, decisamente più professionale e ricca di funzioni utili al nostro progetto. L'indirizzo di riferimento è integromat.com e come al solito è necessario creare un account, vista la successiva necessità di integrare i servizi Google abbiamo optato per l'accesso con le credenziali Google. La versione free (quella usata per questo articolo) limita a 1.000 operazioni al mese, 100 MB di traffico e invio con intervallo minimo di 15 minuti, giusto per rendere operativo il sistema a livello hobbistico, ma esistono piani tariffari che permettono di estendere di molto il suo utilizzo. Appena entrati nel portale la prima cosa da fare è creare un nuovo scenario, partendo dalla Dashbo-

Fig. 6
Servizio
Webhook
collegato
al servizio
Google
Sheets.



ard principale clicchiamo su "Create a new scenario" ed inseriamo i due servizi a noi necessari: Webhooks e Google sheets (nella versione update row). Il primo servizio, denominato webhook (**Fig. 5**) provvederà a recepire la chiamata web da parte del nostro dispositivo IOT, estrapolarne i dati e passarli al successivo servizio (Google sheets) che a sua volta andrà a scriverli nel foglio di calcolo precedentemente preparato.

Cliccando sopra l'icona Webhook si aprirà un pop-up con in evidenza il link al servizio (**Fig. 5**), annotatelo (Copy address to clip board) perché andrà inserito nel listato per Arduino al posto di *IntegromatUrl*. Se cliccate sull'icona Run Once attiverete (per una sola volta) lo scenario e potrete verificarne il funzionamento inserendo il link in un web browser, nella barra di stato in basso avrete un report delle chiamate eseguite. Per finire connettete i due servizi presenti nello scenario come visibile in **Fig. 6**.

Integromat è fatto molto bene ed è in grado di rilevare in automatico la struttura dati inviata dal nostro dispositivo, pertanto clicchiamo su "Re-determine data structure" e predisponiamo la scheda IoT per l'invio di una serie di dati, è importante che nello sketch sia stato correttamente impostato il parametro *IntegromatUrl*. Attendete la scritta *successfully determined* ad indicare che la procedura è andata a buon fine. Occupiamoci ora del servizio che dovrà inviare i dati al foglio di calcolo, quindi clicchiamo sull'icona Google sheets, si aprirà un pop-up (**Fig. 7**) dal quale possiamo impostare l'autorizzazione all'accesso dei servizi della Google

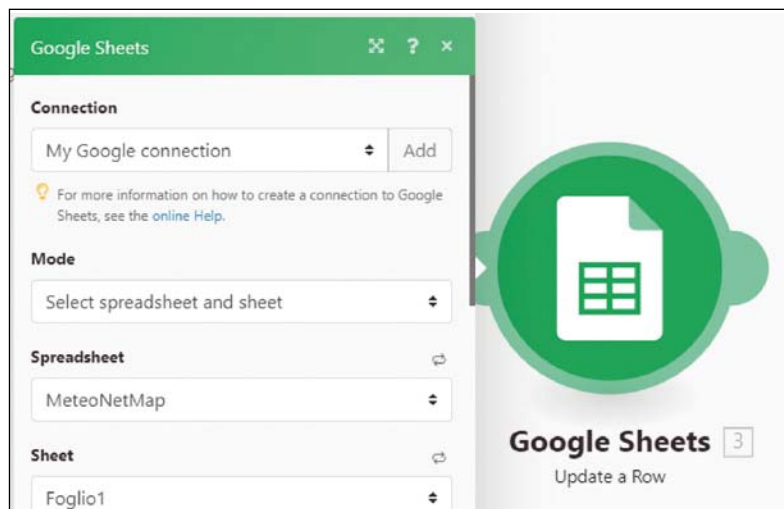


Fig. 7
Impostazione servizio Google Sheets.

(tab connection) ed il nome del foglio di calcolo (tab Spreadsheet). Quando fornirete ad Integromat l'accesso ai servizi della Google vi mostrerà un elenco dei fogli di calcolo attivi dal quale scegliere quello utile al progetto. Per specificare quale riga sarà aggiornata clicchiamo su "tab Row number" e sul pop-up che si aprirà selezioniamo la voce *1.id*; in questo modo quando un dispositivo invierà il proprio ID questo valore verrà usato per specificare su quale riga andranno scritti i dati ricevuti (**Fig. 8**). Continuiamo a completare tutti i campi necessari riferiti a ciascuna colonna del foglio di calcolo, come visibile in **Fig. 9**.

Notate che i campi Date e Time vengono costruiti tramite l'istruzione *formatDate* e la variabile *now* disponibili nel tab *Date and time* del pop-up.

Fig. 8
Impostazione campo Row number.

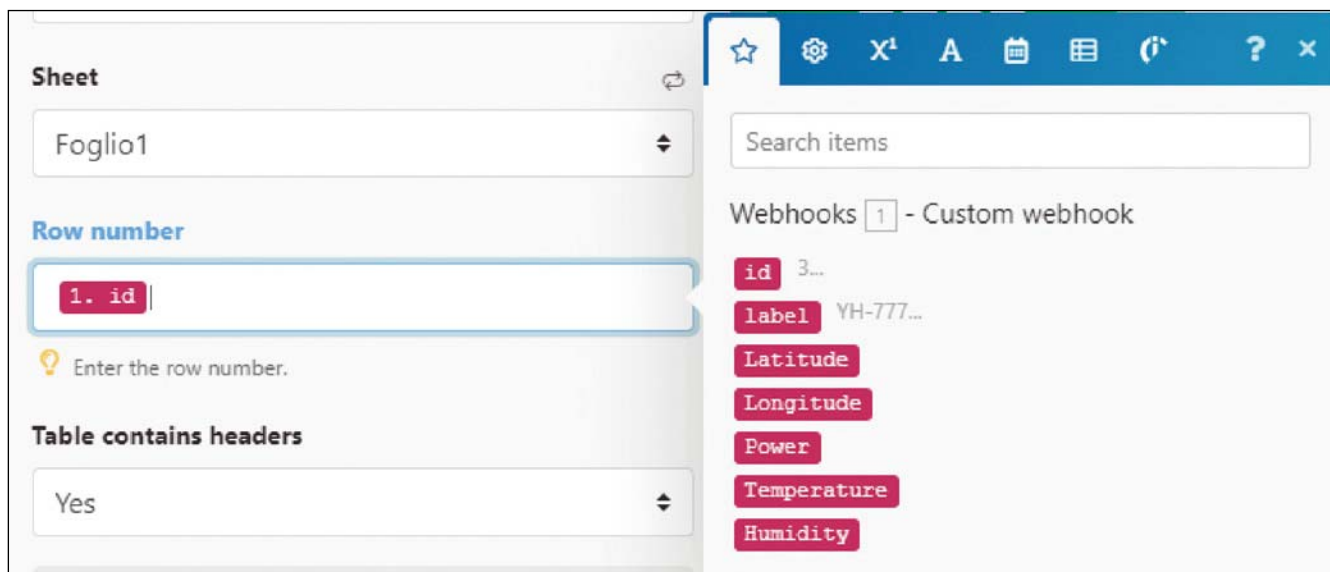


Fig. 9
Impostazione
campi del servizio
Google sheet.

Google Sheets

Values

ID (A)
1. id

Label (B)
1. label

Latitude (C)
1. Latitude

Longitude (D)
1. Longitude

Date (E)
formatDate(now ; DD.MM.YYYY)

Time (F)
formatDate(now ; HH:mm)

Power (G)
1. Power

Temperature (H)
1. Temperature

comando in basso a sinistra. È possibile verificarne il funzionamento inserendo da web browser il seguente link di esempio:

IntegromatUrl?id=2&label=VA-123&Latitude=45.6605746&Longitude=8.7578234&Power=4.19&Temperature=23.54&Humidity=56

Potete testare il servizio Integromat via web browser oppure interagire con esso tramite i dispositivi IOT opportunamente programmati, se tutto va a buon fine vedrete il foglio di calcolo riempirsi con i dati inviati, come visibile in **Fig. 10**. Dal foglio di calcolo accedendo al tab File e successivamente cliccando su "pubblica sul web" otterrete il link di condivisione facilmente integrabile nel vostro sito personale per avere i parametri ambientali sempre a portata di mano.

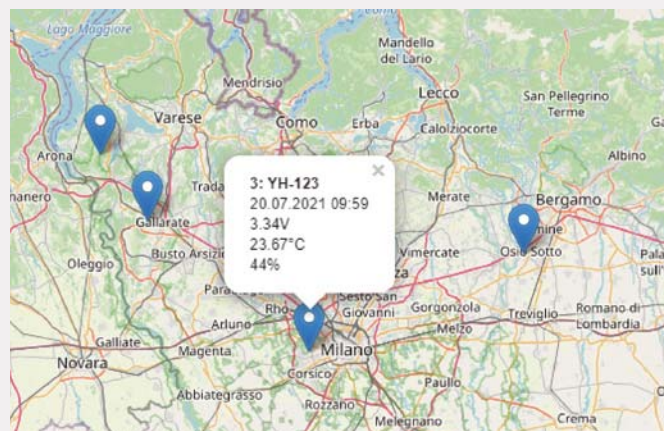
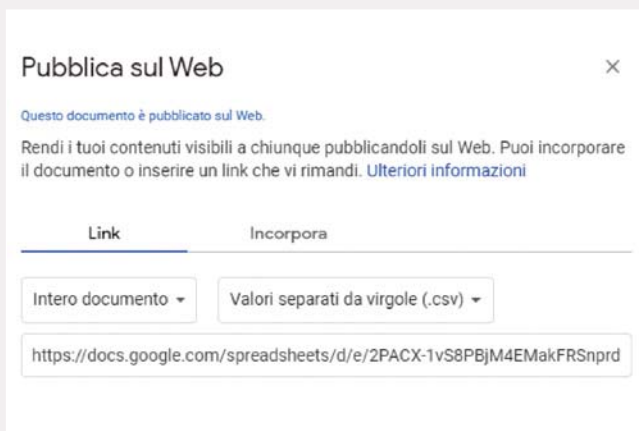
CREAZIONE DELLA MAPPA

Il passo successivo è creare la mappa e popolarla con tanti segnaposto tante sono le righe del foglio di calcolo. Per leggere i valori della tabella ci appoggiamo al JavaScript di nome Papa Parse (<https://www.papaparse.com/>) che permette in modo semplice di accedere ai vari campi di un foglio di calcolo della Google, conoscendo semplicemente l'indirizzo url del foglio. Per costruire la mappa ci appoggiamo alla libreria java-script open source della Leaflet (<https://leafletjs.com/>) utile per la realizzazione di mappe interattive a partire dalle mappe OpenStreetMap. Partendo dagli esempi disponibili sul sito della Leafletjs è possibile costruire una mappa personalizzata che posizioni un segnaposto nelle esatte coordinate fornite dal foglio di calcolo (colonne latitude e longitude), al click sopra ciascuno di essi abiliteremo un pop-up che visualizzi gli ultimi dati ambientali rilevati con la data e l'ora del rilevamento. Procediamo ora alla creazione della pagina web che dovrà visualizzare la mappa: per fare questo ci serve l'accesso al foglio di calcolo, pertanto in Google sheets clic-

Come potete intuire le possibilità di gestione sono veramente molto ampie e con un po' di pratica risultano anche molto semplici da implementare e soprattutto non è necessario scrivere neppure una riga di codice. Adesso il servizio Integromat è completato, non resta che abilitarlo con l'apposito

Fig. 10
Foglio di calcolo
popolato da 4
dispositivi.

	A	B	C	D	E	F	G	H	I
	ID	Label	Latitude	Longitude	Date	Time	Power	Temperature	Humidity
1	2	VA-123	45.6605746	8.7578234	23.07.2021	12:09	3.95	27.90	59
2	3	YH-123	45.4628328	9.1076926	20.07.2021	09:59	3.44	23.67	44
3	4	YH-666	45.6111051	9.2715243	20.07.2021	09:56	3.21	22.67	23
4	5	VA-MIO	45.7605746	8.9578234	21.07.2021	17:09	4.10	26.78	56



chiamo su "Pubblica sul web" e copiamo l'indirizzo internet fornito per la condivisione (**Fig. 11**). Tra i file di questo numero della rivista troverete il file di nome *MeteoNetMapSite.txt* apritelo e modificate la riga `var public_spreadsheet_url` inserendo l'indirizzo precedentemente ricavato. Accedete al servizio site della Google e cliccate su crea un nuovo sito, tralascieremo tutte le infinite possibilità di personalizzazione, noi ci concentriamo sull'integrazione della mappa. Dal menu a destra di nome "Inserisci" selezionate "Incorpora", si aprirà un editor nel quale copiare tutto il testo contenuto nel file *MeteoNetMapSite.txt*, verrà creato un riquadro all'interno della pagina che eseguirà questo listato, facendo visualizzare la mappa costruita con i dati presenti nel foglio di calcolo. Non vi resta che pubblicare la pagina e creare il link di condivisione utilizzabile da qualsiasi browser, come visualizzato in **Fig. 12**. Se volete la tabella sempre visibile nella pagina principale potete utilizzare il comando "Drive" dal tab "Inserisci" e selezionare il foglio di calcolo da incorporare nella pagina.

CONCLUSIONI

La scheda TTGO T-Call si è rivelata un buon hardware per applicazioni di networking come quella qui descritta; la possibilità di essere alimentata tramite una batteria per il funzionamento stand alone la rende ideale alla realizzazione di dispositivi portatili e quindi, anche di device per IoT. Se c'è un'osservazione da fare, un piccolo neo da evidenziare, è che inspiegabilmente dal costruttore non è stato previsto un sistema di ricarica alternativo alla porta USB, come ad esempio un connettore per un pannello solare, pertanto sarebbe necessario integrare la scheda con un Solar power manager esterno oppure utilizzare per l'alimentazione un power bank con pannello

solare, che fornirebbero una discreta soluzione di ricarica autonoma attraverso l'energia solare, in perfetto stile Energy Harvesting. Da questo punto di vista, una scheda Maduino SIM808 sarebbe stata migliore se non fosse che il modulo GSM non supporta la tecnologia SSL per l'invio sicuro dei dati che è richiesta dalla nostra applicazione e dai servizi di Google. Il progetto è stato pensato per dispositivi IoT posizionati in luoghi fissi di cui sono note le coordinate, ma connettendo un ricevitore GPS alla scheda TTGO T-Call potremmo aggiornare anche i campi relativi a longitudine e latitudine e la mappa fornirebbe le coordinate dei dispositivi, anche se questi sono in movimento; in questo caso si parlerebbe di ambiti applicativi come la gestione di flotte di veicoli. Il progetto che vi abbiamo descritto è di una certa complessità e purtroppo per motivi di spazio abbiamo dovuto tralasciare alcuni dettagli, dandoli per assodati; per questa ragione, a meno di non approfondire individualmente i singoli aspetti, per metterlo a punto è necessaria una certa abilità nel gestire reti di dati e servizi web. Ma siamo certi che ce la farete ugualmente a realizzarlo e utilizzarlo!

Fig. 11
Pubblicazione del foglio di calcolo.

Fig. 12
Pagina web con la mappa dei dispositivi IOT.



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. La scheda TTGO T-Call (cod. HR1761) è in vendita a Euro 42,00 mentre il sensore temperatura/umidità (cod. DHT22) è disponibile a 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>

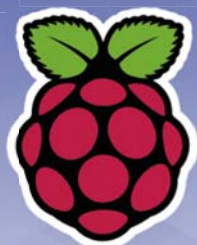
RASPBERRY PI 400!

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.

MOTION TIME LAPSE IN CLOUD

di ANDREA CHIAPPORI

Basato su ESP32-CAM, utilizza Google Apps Script per automatizzare la registrazione su Cloud senza richiedere supporti di memorizzazione. Prima puntata.



Il modulo ESP32-CAM abbina le funzionalità WiFi del microcontrollore ESP32 con una telecamera integrata e per le sue proprietà abbiamo scelto di integrarlo nel mondo Cloud, utilizzando un linguaggio di scripting in Cloud che Google ci mette a disposizione e con il quale potremo arricchire le nostre applicazioni in ambito domestico. L'applicazione che presenteremo ci permetterà di realizzare video dell'ambiente circostante *time-lapse*, in cui un periodo piuttosto lungo (qualche ora o anche giorni) verrà condensato in pochi minuti semplicemente usando un frame rate (che solitamente è di 30 frame al secondo) molto più basso del normale (ad esempio anche un frame all'ora). Fino a qui nulla di speciale, ma, mentre solitamente le singole immagini che andranno a formare il video finale vengono salvate nella memoria microSD del modulo ESP32-CAM (dotato di supporto per memory card), questa volta sfrutteremo

The image displays three different ESP32 development boards. On the left is a USB module with a silver USB-A connector. In the center is a standard module with a black PCB and white pin headers, accompanied by a color-coded pin list: IO pins 1-15 (green), power pins (red), and GPIO pins 16-21 (blue). On the right is a module with a black PCB and a camera module attached, with a color-coded pin list for power, I/O, and camera-specific pins.

Avremo già a disposizione uno sketch di esempio (*CameraWebServer*) che appunto trasforma il nostro

CONVERTITORE	ESP32-CAM
5V	5V
GND	GND
TX	RX(GPIO3)
RX	TX (GPIO1)

modulo in una Web camera con la quale saremo in grado, dopo aver inserito il nome e la password della nostra rete locale, di connetterci ad una pagina internet nella quale potremo vedere lo streaming della telecamera e cambiare una serie di impostazioni ed effetti tramite un pannello laterale di configurazione. Possiamo utilizzare inizialmente questo codice per verificare che il nostro modulo ESP32-CAM funzioni correttamente, ma per la nostra applicazione ne riscriveremo uno molto più semplice, che si limiterà a scattare singole foto.

GOOGLE APPS SCRIPT

Esistono già numerosi servizi che sfruttano il Cloud per permetterci di risparmiare spazio, condividere immagini, automatizzare alcuni nostri automatismi, gestire strumentazione domotica; tutti più o meno configurabili per rispondere alle nostre esigenze, ma a volte può capitare che manchino di qualcosa di nostro gradimento e non ci permettano di prendere veramente il controllo dell'intero flusso per attuare a pieno i nostri progetti. Google cerca di rispondere a questo tramite *Apps Script*, una piattaforma gratuita con un elevato grado di controllo da parte nostra, che ci permette di gestire le nostre chiamate REST API (come ad esempio POST e GET) verso il server in Cloud andando a scrivere il codice che reagirà di conseguenza in base alle nostre più svariate esigenze. Il linguaggio utilizzato è basato su *Javascript*

e fornisce anche librerie per interfacciarsi con altre applicazioni web di Google, come Drive, Calendar, Sheets, Docs, Gmail ed altre. Una volta gestito il collegamento (tramite REST API) tra il nostro dispositivo e *Apps Script*, sarà possibile creare e modificare un qualsiasi tipo di file sul nostro spazio in Drive, aggiungere eventi personalizzati a Calendar o leggere da Calendar e schedulare processi, inviare e leggere mail, salvare report su un foglio elettronico e visualizzarli sotto forma di grafico, memorizzare i nostri log su un file di testo, insomma tutto quello che può venire in mente nel mondo Cloud e tutto programmaticamente a partire dal codice che scriveremo. In **Fig. 2** è rappresentato lo spazio di utilizzo della nostra applicazione e come *Apps Script* interagisce con i vari moduli; il verso delle frecce indica la direzione del flusso usato nel nostro caso particolare; andiamo, però, con ordine e partiamo dall'inizio. Quello che serve è innanzitutto un proprio account Gmail grazie al quale possiamo accedere alla pagina di *Apps Script* dall'indirizzo <https://script.google.com/home>. Comparirà una schermata come quella in **Fig. 3**, in cui vedremo l'elenco dei nostri progetti, tutti salvati nei server di Google, a questo punto potremo aprirne uno o crearne uno nuovo tramite pulsante *New Project* per visualizzare i file contenuti all'interno ed editarli. Il file principale viene chiamato *Code.gs*, ma possiamo crearne altri, all'interno dello stesso progetto, a nostro piacimento.

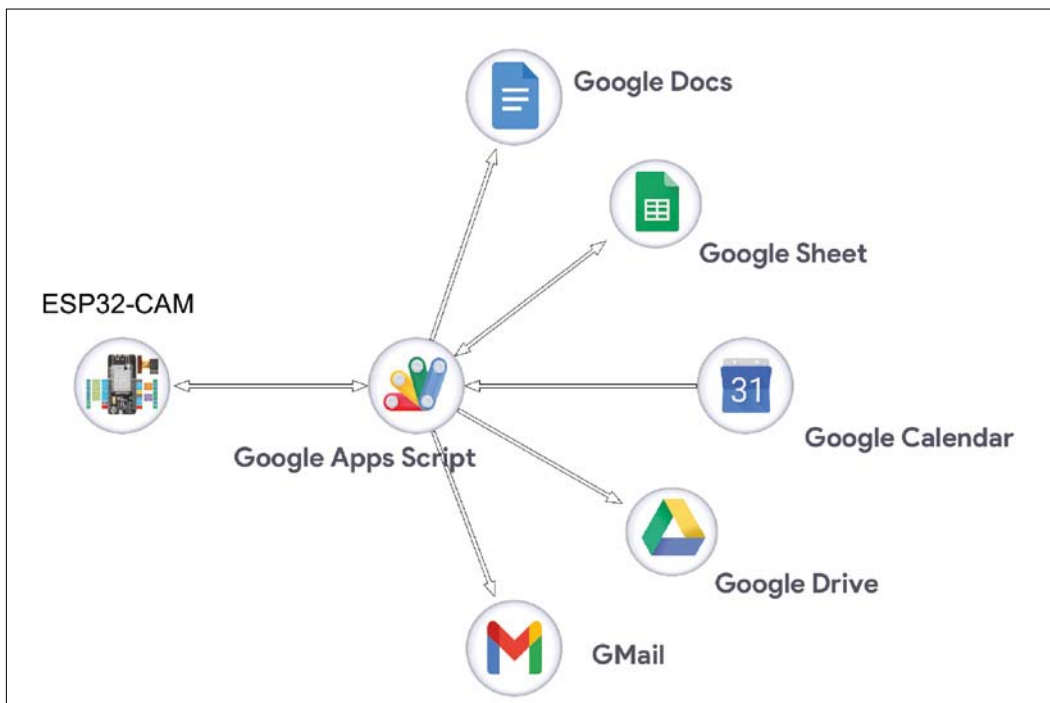
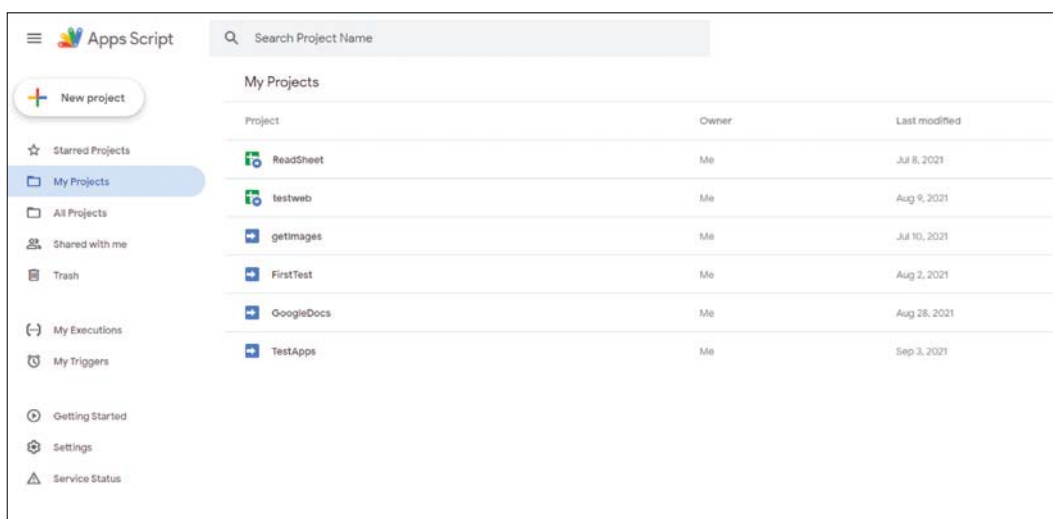


Fig. 2
Interazione di
Apps Script con
i moduli usati
nell'applicazione.

Fig. 3
Lista dei progetti
in Apps Script.



Non vi sarà un vero e proprio *main*, ma ogni file conterrà un insieme di funzioni alcune delle quali, però, dovranno chiamarsi necessariamente in modo particolare per poter rappresentare il punto di accesso da parte dei nostri dispositivi (ESP32, cellulare, o siti) verso lo script che andremo a scrivere. Queste sono la *doPost(e)* che viene chiamata ogni qualvolta verrà eseguita una richiesta POST all'indirizzo del nostro progetto (indirizzo che verrà comunicato in fase di pubblicazione del progetto, come vedremo) e la *doGet(e)* che viene chiamata a seguito di una richiesta GET. Il parametro *e*, in entrambi i casi, contiene l'oggetto della richiesta e all'interno delle funzioni potremo accedere ai singoli valori utilizzando alcuni oggetti che ci vengono in aiuto. Ad esempio, nel caso di una richiesta GET di questo tipo:

```
<indirizzo-server-google>?temperatura=20.1&umidita=40%
```

potremo accedere ai parametri passati usando *e.parameter.temperatura* che conterrà 20.1 e *e.parameter.umidita* che conterrà 40% (come in **Fig. 4**). Analogamente, una richiesta POST che contenga come parametro un oggetto JSON potrà essere

ricevuta e parserizzata da Google utilizzando la riga di codice:

```
var parsedData = JSON.parse(e.postData.contents);
```

ed accedendo ai singoli valori richiamandoli dall'oggetto *parsedData*, ad esempio *parsedData.temperatura*. Questo esempio è visibile in **Fig. 5**; entrambe le funzioni possono far parte, come in questo caso, dello stesso progetto (*TestApps*). Entrambe queste due funzioni dovranno poi terminare con la riga di codice

```
return ContentService.createTextOutput(result)
```

in modo da restituire una stringa di risposta che salveremo nella variabile stringa *result* e che potrebbe anche essere un oggetto JSON (utilizzando *JSON.stringify(jsonObj)*). Nei primi test, a scopo di debug, potremmo addirittura caricare sulla stringa *result* i parametri di input ricevuti per testare la corretta acquisizione. Nel corpo di queste funzioni (dopo il parse dei parametri di ingresso e il *return*) scriveremo il nostro codice utilizzando le librerie messe a disposizione da

Fig. 4
Esempio di funzione
doGet().





Fig. 5
Esempio
di funzione
doPost().

Google e che, come vedremo, sono molto comode. Dopo aver scritto le funzioni del nostro nuovo progetto, dovremo renderle disponibili premendo sul bottone *Deploy* in alto a destra (come appare in **Fig. 5**) e scegliendo l'opzione *New Deployment*. Si aprirà una finestra come in **Fig. 6** in cui potremo scegliere il tipo di applicazione che, per ora, sarà *Web App*, anche se in futuro sarà possibile aggiungere nuovi tipi. A questo punto si aprirà una finestra come quella in **Fig. 7** in cui potremo dare una descrizione al nostro progetto e scegliere a chi renderla pubblica, se a tutti, solo chi ha un account Google o solo a noi. Dopo aver premuto il bottone *Deploy*, vedremo una finestra come quella in **Fig. 8** in cui viene visualizzata la stringa dell'*Id* univoco per il nostro progetto e l'indirizzo completo della nostra applicazione web a cui faremo le richieste POST e GET. Ora il nostro nuovo progetto comparirà nell'elenco dei progetti già visti in **Fig. 3** e manterrà sempre lo stesso *Id*. Ogni volta che andremo modificare il progetto, dovremo rendere pubbliche le modifiche andando sempre a cliccare sul bottone *Deploy* in alto a destra, ma questa volta sceglieremo *Manage Deployments* e si aprirà una finestra come in **Fig. 9** in cui dovremo cliccare sul pulsante con la matita (in alto a destra) ed impostare "New Version" nel campo *Versione* per rendere attivabile il pulsante *Deploy* che ci permetterà l'aggiornamento; prima di ciò, però, possiamo ancora modificare la descrizione del progetto e modificare l'accessibilità dello stesso. Il campo *Version* mantiene anche tutte le versioni precedenti per cui potremo renderle attive selezionando quella di interesse ed effettuando il deploy; sulla parte sinistra della finestra viene infatti, mantenuta tutta la storia del progetto. Prima di addentrarci nel codice riguardante la nostra applicazione possiamo esaminare alcuni esempi, per comprendere quando sia potente questo strumento. Per farlo useremo un comodo tool dedicato per testare le REST API, chiamato *Postman* e scaricabile a questo indirizzo <https://www.postman.com/downloads>.

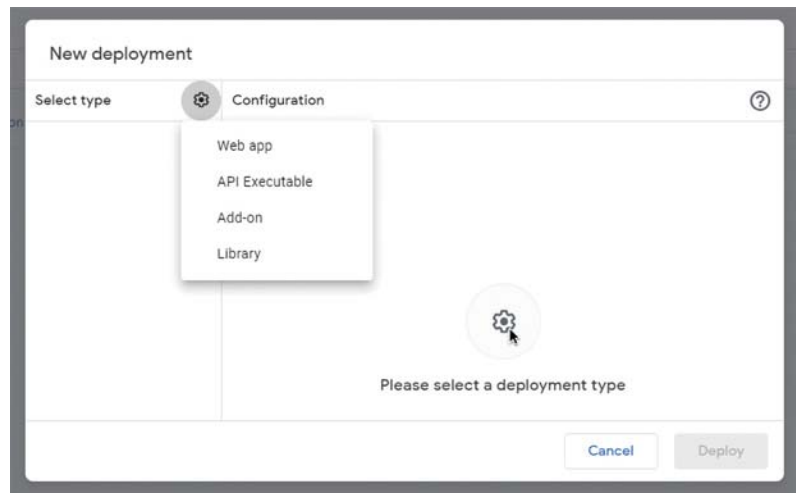


Fig. 6
Creazione di un
nuovo progetto.

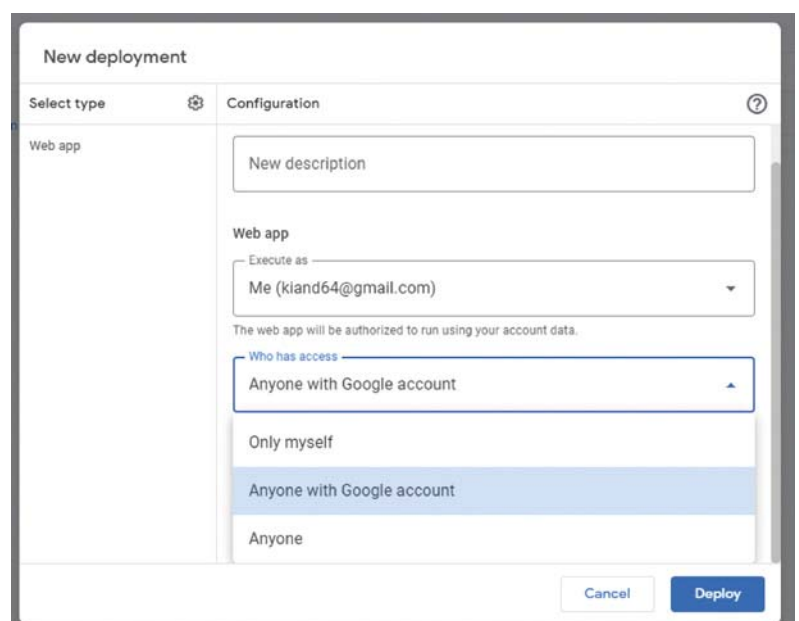


Fig. 7
Impostazioni del
nuovo progetto.

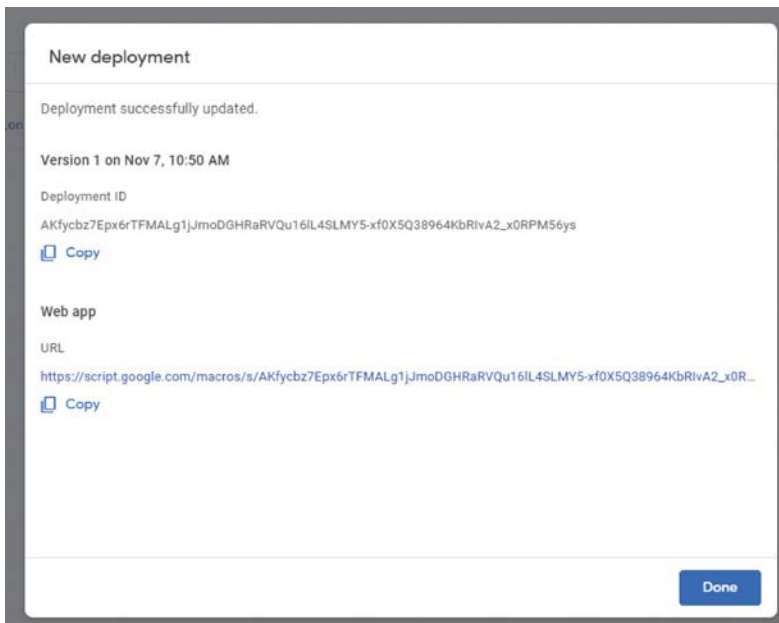


Fig. 8
Identificativo del nuovo progetto.

PRIMI TEST CON POSTMAN

Postman ci servirà per inoltrare le REST API (sia GET che POST) al Cloud Google in modo semplice prima di scrivere il firmware sul nostro dispositivo ESP32, permettendoci così di emularlo e testando le nostre funzioni di *Apps Script* riducendo i tempi morti di programmazione ed eliminando alcune variabili. Abbiamo già il primo scheletro di progetto in cui vengono gestite le POST e le GET quindi scriveremo entrambe le richieste con *Postman* e verificheremo che queste restituiscano le risposte desiderate, nel

Fig. 9
Aggiornamento del progetto.

nostro caso, per ora, sono le stesse che passeremo nella stringa (per il caso della GET) o nel corpo (per il caso del POST) della richiesta.

Per entrambe le richieste ci serve l'Url (comprensivo di *Id*) fornito dal progetto *Apps Script* in fase di deploy a cui aggiungeremo eventualmente alcuni parametri. Dal momento che abbiamo scelto temperatura e umidità come test, passeremo questi due campi con valori inventati.

Per quanto riguarda la GET avremmo quindi una stringa di questo tipo:

```
https://script.google.com/macros/s/
AKfycbwyLOINPv0ccbp1gv_mwsDqfP4kVy_T17fefbY5Kh1Hk3c
1y7KN21nBhTLswEuedckBng/exec?temperature=20,4&humidi
ty=31
```

mentre per quanto riguarda la POST la stringa sarà più semplicemente:

```
https://script.google.com/macros/s/
AKfycbwyLOINPv0ccbp1gv_mwsDqfP4kVy_T17fefbY5Kh1Hk3c1y
7KN21nBhTLswEuedckBng/exec
```

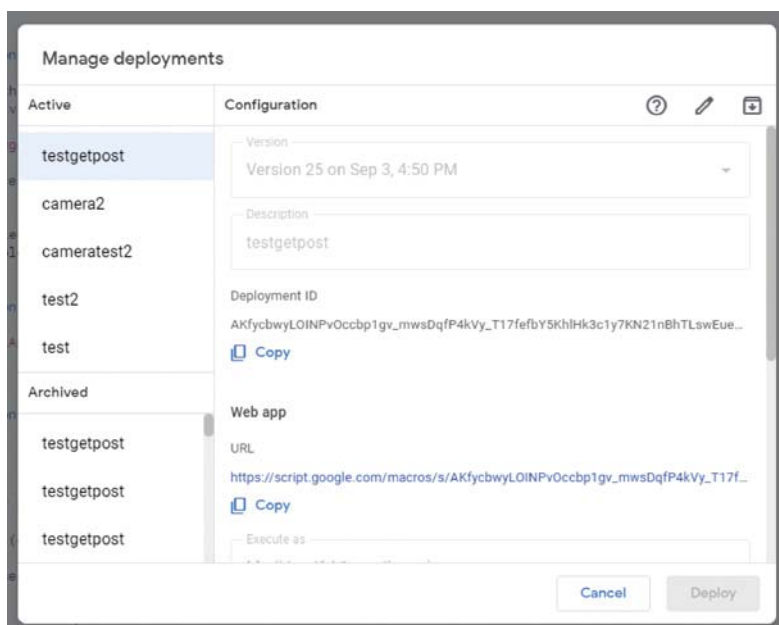
ma nel corpo della post inseriremo una stringa JSON per il passaggio dei parametri che sarà di questo tipo:

```
{
  "temperature" : "20.5",
  "humidity" : "50%"
}
```

Il tutto si traduce con *Postman* nelle **Fig. 10** e **Fig. 11** in particolare possiamo vedere come nel caso della GET la stringa di comando comprenda già i parametri nella *query string* (dopo il punto interrogativo), che si possono aggiungere o modificare anche in modo più 'visuale' nella tabella sottostante, mentre nel caso del POST la stringa di comando è semplicemente quella fornita dal progetto e pertanto viene aggiunta la stringa JSON nella finestra *Body*. In entrambe le figure, nella parte inferiore della schermata vedete le stringhe di risposta che restituiscono gli stessi valori, confermando il funzionamento delle nostre funzioni del progetto.

FUNZIONI APPS SCRIPT

A questo punto possiamo iniziare a far pratica con il gran numero di funzioni messe a disposizione da Google in *Apps Script* che ci permetteranno di interagire con i vari moduli di Google e ai quali potremmo già accedere per il fatto di possedere un account



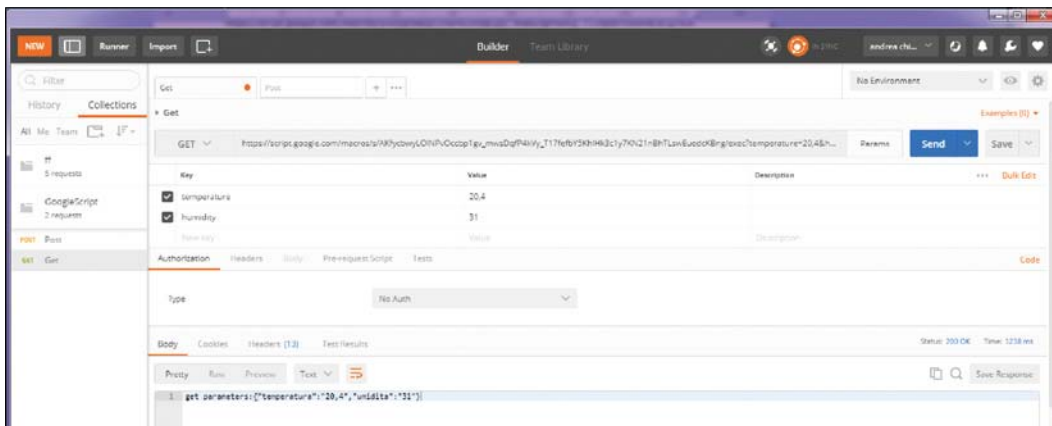


Fig. 10
Esempio di
richiesta GET
con Postman.

Google. In questa fase creeremo alcune nostre funzioni per la gestione delle mail e per l'accesso a Google Calendar a Sheets (il foglio elettronico di Google) a Google Drive e Google Docs. Tutte queste funzioni saranno poi utilizzate nella nostra applicazione finale per i vari scopi, ma ci potranno servire, in futuro anche per altri progetti; suddivideremo queste funzioni in tanti file (*Apps Script* assegnerà di default estensione .gs) separati in base all'area di interesse in modo da poterli riutilizzare anche in seguito con più facilità. Vedremo anche come poter eseguire il debug della singola funzione direttamente online, dall'Ide nel browser, molto importante e soprattutto comodo per poter testare separatamente le proprie funzioni in modo veloce perché non richiede il deploy dell'intero progetto.

Funzioni per la gestione della mail

Nel **Listato 1** sono rappresentate le funzioni create per gestire la mail, in particolare la *sendMail()* verrà usata per inviare una mail con l'informazione sulla chiusura del processo di acquisizione di tutte le immagini ed è praticamente un wrapper della fun-

zione di libreria di Google *sendEmail()* appartenente all'oggetto *GmailApp*, con la differenza che nell'oggetto vogliamo anteporre la stringa con il nome del nostro progetto; per farlo useremo l'oggetto *ScriptApp* per ottenere l'*Id* del progetto tramite funzione *getScriptId()* e l'oggetto *DriveApp* per ottenere il nome del progetto tramite le funzioni *getFileById()* e *getName()*.

Entrambi gli oggetti contengono numerose funzioni di libreria che possono essere esplorate tramite il comodo intellisense fornito internamente al browser e dotato anche di alcuni esempi con descrizione.

Da notare come per ottenere il nome del progetto si ricorra al metodo *DriveApp* che quindi richiama il modulo Drive di Google perché ogni progetto non è altro che un file salvato nel repository di Drive. Infine utilizzeremo la funzione *log* dell'oggetto *Logger* per visualizzare, in una finestra di log e solo in fase di debug, come vedremo, il nome del progetto. La funzione *getMail()*, sempre nello stesso listato, usa la funzione *getInboxThreads()* dell'oggetto *GmailApp*, per leggere l'oggetto delle mail del proprio

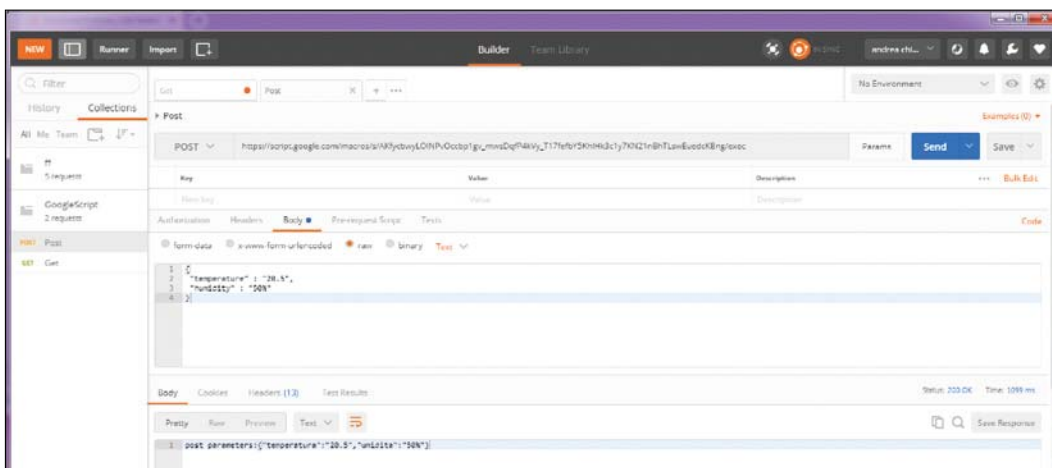


Fig. 11
Esempio di
richiesta POST
con Postman.

Listato 1 - Funzioni per la gestione dell'e-mail.

```
function sendMail(recipient, subject, body)
{
    var id = ScriptApp.getScriptId();
    var projectName = DriveApp.getFileById(id).getName();
    Logger.log(projectName);
    GmailApp.sendEmail(recipient, projectName + ":" + subject, body);
}

function getMail()
{
    var threads = GmailApp.getInboxThreads();
    for (var i = 0; i < threads.length; i++)
    {
        Logger.log(threads[i].getFirstMessageSubject());
    }
    var len = threads.length;
    Logger.log("len:" + len);
}
```

account Google (tramite funzione *getMessages()*) nella finestra di log.

Questa funzione non verrà usata ai fini del nostro progetto, ma solo a livello didattico per testare la lettura delle mail ed in ottica di un futuro utilizzo.

Funzioni per la gestione del calendario

Nel **Listato 2** sono visualizzate le funzioni per la

gestione di Google Calendar attraverso le quali possiamo impostare alcuni eventi e la loro durata nel proprio calendario o leggere quelli già impostati in precedenza tramite il nostro cellulare, da computer o da altri progetti *Apps script*.

Nel nostro progetto useremo la *checkEventInCalendar()* per verificare se sul nostro calendario è presente un evento chiamato *eventName* (ad esempio "motioncamera") nella stessa fascia oraria della richiesta. Per fare questo passeremo alla funzione *getEvents()* l'intervallo di tempo in cui cercare l'evento che nel nostro caso sarà lo stesso istante, in modo da interrompere l'acquisizione delle immagini nel caso ci trovassimo già fuori dall'intervallo.

Ovviamente sono disponibili altre funzioni come, ad esempio, la *getEventsForDay()* che controlla la presenza dell'evento in data odierna e che probabilmente non useremo, ma l'invito è, come al solito, quello di esplorare i vari metodi degli oggetti per trovare quelli che più potranno tornarci utili.

Infine la nostra funzione avrà *eventName* come parametro che verrà internamente passato al metodo *getEvents()* per verificare l'esistenza dell'evento richiesto alla tale data. In **Fig. 12** vedete un esempio di come impostare il periodo di esecuzione dell'applicazione operando solamente su Google Calendar. La funzione *setEventInCalendar()* aggiungerà, invece, un evento al nostro calendario che verrà identificato con un nome ed inizierà alla data specificata e durerà per un tempo impostabile nell'ordine dei minuti (parametro *duration_min*); non verrà utilizzata nel nostro progetto, ma potrebbe essere utile in altre applicazioni future.

Listato 2 - Funzioni per la gestione del calendario.

```
function checkEventInCalendar(eventName)
{
    var t1 = new Date();
    var t2 = new Date();
    t2.setSeconds(t1.getSeconds() + 1);
    Logger.log(t1);
    Logger.log(t2);

    var calendar;
    calendar = CalendarApp.getDefaultCalendar();
    var options = new Object();
    options.search = eventName;
    var events=calendar.getEvents(t1, t2, options);
    var len = events.length;
    Logger.log("len:" + len);
    return (len > 0);
}

function setEventInCalendar(eventName, eventDate,
duration_min)
{
    var t1 = eventDate;
    var t2 = new Date();

    t2.setSeconds(t1.getSeconds()+60*duration_min);
    var calendar=CalendarApp.getDefaultCalendar();
    var event=calendar.createEvent("test", t1, t2);
    Logger.log("Event ID: " + event.getId());
    return event;
}
```

Funzioni per la gestione del foglio elettronico

Nel **Listato 3** sono visualizzate alcune funzioni con

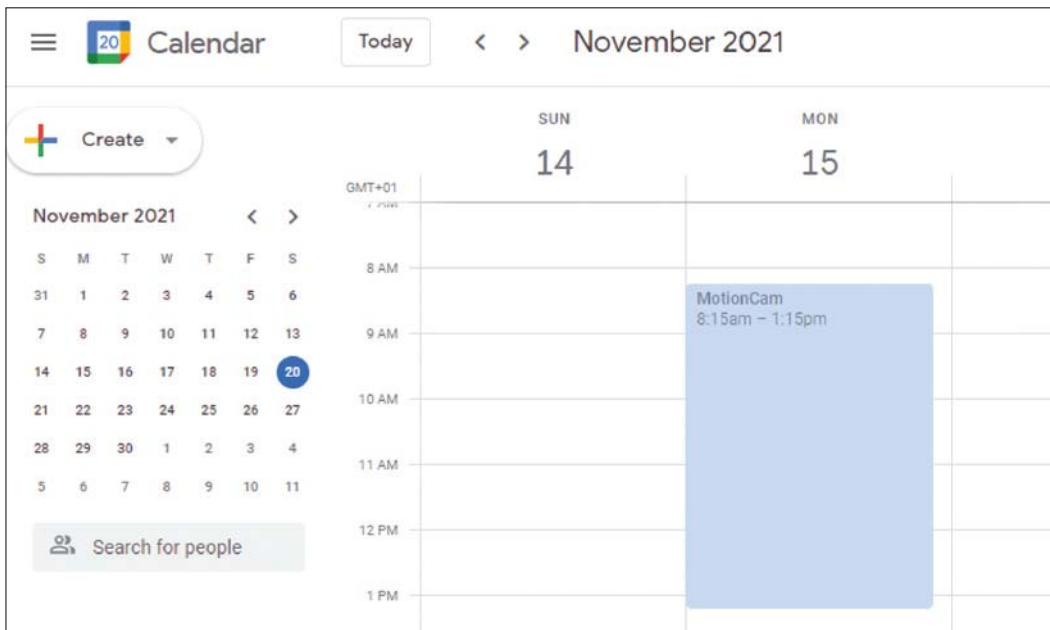


Fig. 12
Controllo
dell'applicazione
tramite evento
in Calendar.

Listato 3 - Funzioni per testare foglio elettronico.

```
var sheet;
var FileId = "15rFfBmJfSUeoJ27oB62NUhYiGy1-hYhaPBphr9nqDqQ";
var FileSheetName = "Sheet1";

function openSheet(stringId, stringPage)
{
    var sheetFile=SpreadsheetApp.openById(stringId);
    var sheet=sheetFile.getSheetByName(stringPage);
    return sheet;
}

function testWrite(range, value)
{
    range.setValue(value);
    range.setFontColor("red");
}

function testRead(range)
{
    var values = range.getValues();
    return values;
}

function onOpen()
{
    sheet = openSheet(FileId, FileSheetName);
}

function sandbox()
{
    sheet = openSheet(FileId, FileSheetName);
    // test read
    var rangeVal = sheet.getRange(1,1);
    var val = testRead(rangeVal);
    Logger.log(val[0][0]);
    // testwrite
    var date = Utilities.formatDate
    (new Date(), "GMT+1", "dd/MM/yyyy HH:mm");
    rangeVal = sheet.getRange('A9');
    testWrite(rangeVal, date);
}
```

le quali saremo in grado di leggere e scrivere determinate celle del foglio elettronico di Google (Sheets). Possiamo creare un nuovo file Sheets direttamente da codice (verrà salvato automaticamente su Drive), ma anche aprire un file Sheets già esistente, tramite l'utilizzo dell'*Id* del file che compare nella barra dell'indirizzo internet una volta aperto il file interessato; ad esempio nel caso di un indirizzo di questo tipo

<https://docs.google.com/spreadsheets/d/15rFfBmJfSUeoJ27oB62NUhYiGy1-hYhaPBphr9nqDqQ/edit#gid=0>

L'*Id* del nostro file sarà *15rFfBmJfSUeoJ27oB62NUhYiGy1-hYhaPBphr9nqDqQ*

Dal momento che *Apps Script* fornisce una funzione di callback (*onOpen()*) che viene chiamata all'avvio dell'esecuzione dello script, la utilizzeremo per aprire il file con la nostra funzione *openSheet()* che internamente utilizza il metodo *openById()* dell'oggetto *SpreadsheetApp*. In questo modo caricheremo tutto il file sulla variabile *sheet* e non dovremo aprire ogni volta il file. Occorre specificare anche quale pagina del foglio elettronico utilizzeremo tramite il metodo *getSheetByName()* al quale verrà passato il nome del foglio, nel nostro caso "Sheet1".

La funzione *testWrite()* è semplicemente un test per scrivere un valore *value* (in colore rosso) su una cella identificata dal parametro *range* che deve essere passato alla funzione utilizzando il metodo



Fig. 13
Schermata
del foglio
elettronico per la
configurazione
e l'esecuzione
dell'applicazione.

	A	B	C	D	E	F	G	H
1	Setup				Links			
2	IntervalTime-sec	MaxFrames	ProjectName		Link-Doc	Link-Calendar	Link-Drive	
3	60	1000	MotionCam18		MotionCam18.log	Calendar	MotionCam18	
4								
5	Daily Range							
6	StartAt	EndAt	Enable Range					
7	10:31	15:00	☐					
8								
9								
10	Control Panel							
11	Command	Flash	ClearLog					
12	-	☐	☐					
13								
14								
15	Last Get	Status	Mode		Image			
16	2021-11-20 18:04:46	Active	Sheet					
17								
18	Statistic							
19	NumberOffFrames	StartTime	ElapsedTime(min)		Decoding-status	Processing file	DecodeAll	
20	6	2021-11-20 18:04	8		OFF	-	☐	
21								
22	Logs							
23	20/11/2021 18:04:55 Starting							
24	20/11/2021 18:04:58 Started by command							
25	20/11/2021 18:05:18	Received_file_1	data:image/jpeg:bas	☒				
26	20/11/2021 18:06:39	Received_file_2	data:image/jpeg:bas	☐				
27	20/11/2021 18:08:03	Received_file_3	data:image/jpeg:bas	☐				
28	20/11/2021 18:09:25	Received_file_4	data:image/jpeg:bas	☐				
29	20/11/2021 18:10:49	Received_file_5	data:image/jpeg:bas	☐				

getRange() (sempre applicato alla variabile *sheet* caricata all'avvio) per identificare le celle desiderate. In particolare *getRange()* può ricevere come parametri sia una stringa (ad esempio *sheet.getRange('A1')*) sia due interi (ad esempio *sheet.getRange(1,1)*). Allo stesso modo la funzione *testRead()* restituisce i valori compresi nel parametro passato *range*; da notare come il range non debba essere necessariamente una cella soltanto, ma potrebbe essere una riga, una colonna o una sotto-matrice, il valore restituito dal metodo *getValues()* sarà infatti un array bidimensionale al quale poi potremo accedere con *getValues()[0][0]* nel caso si tratti di una cella singola. Altre funzioni più specifiche (non presenti nel lista-to, ma scaricabili dal sito della rivista) serviranno per gestire la lettura e scrittura di determinate celle in modo da utilizzare il foglio elettronico online sia come configuratore della nostra applicazione, sia come monitor per verificare il corretto funzionamento e sia come generatore di comandi. Facendo infatti riferimento alla **Fig. 13**, il foglio utilizzato (*Sheet1*) è organizzato in modo che nella sezione di *Setup* sia possibile impostare l'intervallo di tempo tra una acquisizione e l'altra, l'ora di

inizio e fine dell'acquisizione o il massimo numero di frame possibili oltre il quale forzare la chiusura dell'applicazione. Tutti questi parametri, insieme al valore del campo *Command* che potrà contenere le stringhe *Start*, *Stop*, *Pause*, *Resume* verranno letti ciclicamente dal dispositivo ESP32 (tramite le REST API viste prima) che a sua volta invierà alcune informazioni sul proprio stato andando a popolare (tramite le funzioni del nostro script) le celle di *Status*, *Logs* e *Statistics*, nonché ovviamente, l'immagine vera e propria. Per semplificare l'iterazione con l'utente il campo *Command* è vincolato ad assumere solo le stringhe di comando possibili tramite un menu a tendina e una volta impartito il comando la stessa cella verrà settata con un valore neutro (stringa "-") in modo da evitare ripetizioni ulteriori. Questo foglio elettronico rappresenterà quindi una sorta di telecomando per il nostro dispositivo ed ipotizzando che sarà soggetto a cambiamenti anche per quanto riguarda il layout, si può scegliere di posizionare manualmente le singole celle usando un testo per il nome del campo, mentre per leggere e scrivere il valore corrispondente verrà cercato in automatico, tramite script, la cella sottostante, in modo da poterle

spostare anche in seguito senza cambiare righe di codice. Infine per avere tutto sotto controllo alcune celle sono state dotate, sempre tramite script, di formule per contenere link alla cartella Drive contenente le immagini e al file di log. Ovviamente questo è solamente uno dei possibili approcci per interagire remotamente con il nostro dispositivo esterno e potrà essere modificato ed ampliato a piacere.

Funzioni per la gestione di file e cartelle

Nel **Listato 4** sono visualizzate le funzioni per la gestione dei file su Drive; tramite alcuni metodi dell'oggetto *DriveApp* (*getFoldersByName()*, *createFolder()*, *getFiles()*) saremo facilmente in grado di creare la funzione *deleteFolderDrive()* per cancellare una cartella su Drive (in fase di inizializzazione, per assicurarci di partire da una situazione pulita) e la funzione *createFolderDrive()* per crearne una nuova dove, tramite le

Listato 4 - Funzioni per la gestione del trasferimento delle immagini.

```
var WorkingFolder = "MotionCam";
function createFolderDrive(folderName)
{
  var folders = DriveApp.getFoldersByName(folderName);
  if (!folders.hasNext()) {
    return folder = DriveApp.createFolder(folderName);
  }
  else{
    return folders.next();
  }
}
function deleteFolderDrive(folderName)
{
  var folders = DriveApp.getFoldersByName(folderName);
  if (folders.hasNext()) {
    var files = folder.getFiles();
    while (files.hasNext()){
      var file = files.next();
      file.setTrashed(true);
    }
    folder.setTrashed(true);
  }
}
function saveTextFileDrive(text, folderName, fileName)
{
  var folders = DriveApp.getFoldersByName(folderName);

  if (!folders.hasNext()) {
    return;
  }
  var folder = folders.next();
  var file = folder.createFile(fileName, text);
  file.setDescription("Uploaded by ESP32-CAM");
  return file.getId();
}
function saveDataFileDrive(blob, folderName, fileName)
{
  var folders = DriveApp.getFoldersByName(folderName);
  if (!folders.hasNext()){
    return null;
  }
  var folder = folders.next();
  var file = folder.createFile(blob);
  file.setName(fileName);
  file.setDescription("Uploaded by ESP32-CAM");
  return file.getId();
}
function sandBox()
{
  var folder = createFolderDrive(WorkingFolder);
  var blob = Maps.newStaticMap().setCenter('via Roma, Genova IT').getBlob();
  saveDataFileDrive(blob, WorkingFolder, "file.bmp");
  saveTextFileDrive("text in file", WorkingFolder, "file.txt");
}
```

successive funzioni, verranno copiati i fotogrammi acquisiti dal modulo ESP32.

Il metodo *next()*, applicabile sia su file o cartella per accedere sia su oggetto file che cartella per accedere ad eventuale altro file o cartella sullo stesso livello, è usato anche per cancellare tutti i file interni ad una stessa cartella. Il metodo *hasNext()* è invece usato per verificare che esista un file o una cartella. Per la creazione dei file (una volta identificata la cartella di destinazione) viene usato il metodo *createFile()*, sempre dell'oggetto *DriveApp*, al quale possiamo passare il nome del file ed il testo interno per ritrovare questo nuovo file sul nostro repository Drive. Questo è quello che viene usato nella funzione *saveTextFileDrive()* e che può tornare utile per scrivere un file di log dell'applicazione, ma a noi interessa salvare un'immagine e per questo scopo possiamo usare sempre lo stesso metodo *createFile()*, ma al quale questa volta passeremo un solo parametro di tipo *blob*, come è stato fatto nella funzione *saveDataFileDrive()*. Il *blob* è un contenitore di dati di vario tipo ed è anche usato per immagini come serve a noi; a titolo di esempio, si può notare nella funzione di test *sandBox()*, come usare il metodo *getBlob()* dell'oggetto *Maps* per ottenere, con appena due righe di codice, un file immagine con la mappa desiderata; il nome del file viene poi determinato tramite il metodo *setName()*. Nel progetto, le immagini acquisite dalla webcam vengono inviate tramite REST API e codificate Base64, con conversione da binario a

stringa di testo ASCII, quindi una volta ricevute dallo script dovremo prima decodificarle Base64 e creare un blob da passare alla funzione appena creata, che ritroveremo nel repository come file *jpg*. Siccome in Drive è possibile aggiungere cartelle con lo stesso nome, potremmo dover aggiungere al codice ulteriori controlli.

Funzioni per la gestione della decodifica delle immagini

Proprio per la decodifica Base64 ricorriamo ad un altro oggetto fornito da *Apps Script*, ovvero *Utilities*. In realtà questo oggetto ha al suo interno interessanti funzioni che, oltre a vari algoritmi di codifica e decodifica, offrono la possibilità di impostare un ritardo temporale (tramite metodo *sleep()*), parse- rizzare file csv in un array (*parseCSV()*), serializzare e deserializzare oggetti JSON (*jsonStringify()*, *jsonParse()*) comprimere e decomprimere file (*zip()*, *unzip()*), gestire la formattazione delle stringhe e delle date (*formatString()*, *formatDate()*) e, non ultimo, creare *blob* di dati (*newBlob()*) come servirà al nostro caso. Nel **Listato 5** è visualizzata la funzione *decodeImage()* che accetta come parametro una stringa Base64, in formato URI, contenente, quindi, anche l'informazione sulla codifica esattamente come la riceveremo dal modulo ESP32 ed usando i metodi *base64Decode()* e *newBlob()* otterremo il *blob* pronto per essere caricato nella cartella di destinazione, insieme agli altri file. In questo caso di

↓ Listato 5 - Decodifica dell'immagine in un blob.

```
var testImageBase64 = "data:image/jpeg;base64,%2F9j%2F4AAQSkZJR...gABAQEAAAAAAD%";
function decodeImage(encodedImage)
{
    var contentType =
    encodedImage.substring(encodedImage.indexOf(":")+1, encodedImage.indexOf(";"));
    var data = encodedImage.substring(encodedImage.indexOf(",")+1);
    data = decodeURIComponent(data);
    var decodedData = Utilities.base64Decode(data);
    var blob = Utilities.newBlob(decodedData, contentType, "esp32-cam.jpg");
    return blob;
}
function sandBox()
{
    var blob = decodeImage(testImageBase64);
    // send me the blob
    var recipient = Session.getActiveUser().getEmail();
    var response = GmailApp.sendEmail(recipient, "test", "hi, test",
    {
        attachments: blob,
        name: 'Automatic Emler Script'
    });
    Logger.log("response: " + response);
}
```



Listato 6 - Funzioni per la gestione di un documento di testo.

```
var doc;
var FileDocId = "1d-TU_sWMsYsz6sFZJh1WIFK0xRlr7LotCJ4N69u95VQ";
function openDoc(stringId)
{
    var doc = DocumentApp.openById(stringId);
    return doc;
}
function createDoc(name)
{
    var doc = DocumentApp.create(name);
    return doc.getId();
}
function clearDoc()
{
    var body = doc.getBody();
    body.clear();
}
function testWrite(value)
{
    var body = doc.getBody();
    var date = Utilities.formatDate(new Date(), "GMT+1", "dd/MM/yyyy HH:mm");
    body.appendParagraph(date + ":" + value);
}
function testRead()
{
    var body = doc.getBody();
    var text = body.getText();
    Logger.log(text);
}
function sandbox()
{
    doc = openDoc(FileDocId);
    clearDoc();
    testWrite("test");
    testRead();
    createDoc("Log2");
}
```

test, nella *sandbox()* invieremo il *blob* alla nostra mail Google come allegato.

Funzioni per la gestione dei documenti

Volendo utilizzare un file di testo specifico per creare un file di log dell'applicazione, potremmo creare un documento di Google da aggiungere al nostro archivio Drive e che ci servirà anche nelle prime fasi di debug, usandolo nelle GET e nelle POST per stampare i parametri ricevuti dal modulo ESP32 o per scrivere l'immagine codificata e fare le opportune verifiche. L'oggetto da utilizzare sarà *DocumentApp* e nel **Listato 6** vedete le funzioni di test per aprire un file di testo esistente (*openDoc()*) tramite l'Id, scrivere e leggere informazioni di testo (*testWrite()*, *testRead()*) e anche creare un documento nuovo tramite il metodo *create()*. Una volta ottenuto l'oggetto Doc potremo accedere al contenuto del testo tramite il metodo *getBody()* e, nel caso della scrittura, aggiungere una riga in fondo tramite il metodo *appendParagraph()* che ci permetterà di ag-

giungere una riga di log. Qui useremo semplici linee di testo, anche se *DocumentApp* ci consentirebbe di creare testo formattato, aggiungere link, margini e stili particolari.

CONCLUSIONI

Si conclude qui questa prima puntata. Nella prossima vedremo come grazie a Google potremo gestire la nostra ESP32-CAM e i fotogrammi acquisiti. 



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.

STAZIONE DI MONITORAGGIO QUALITÀ DELL'ARIA

L'importanza della qualità dell'aria ci ha spinto a progettare una valida centralina di monitoraggio, della quale nella puntata precedente abbiamo descritto la filosofia progettuale e l'hardware, spiegandovi poi come realizzarla.

In questa seconda puntata affronteremo gli aspetti software descrivendo sia il firmware di sistema, sia il software di interfacciamento da Personal

Computer (o Raspberry Pi) che ci permetterà di consultare ed elaborare a distanza i dati trasmessi dall'unità su rete wireless LoRa.

FIRMWARE

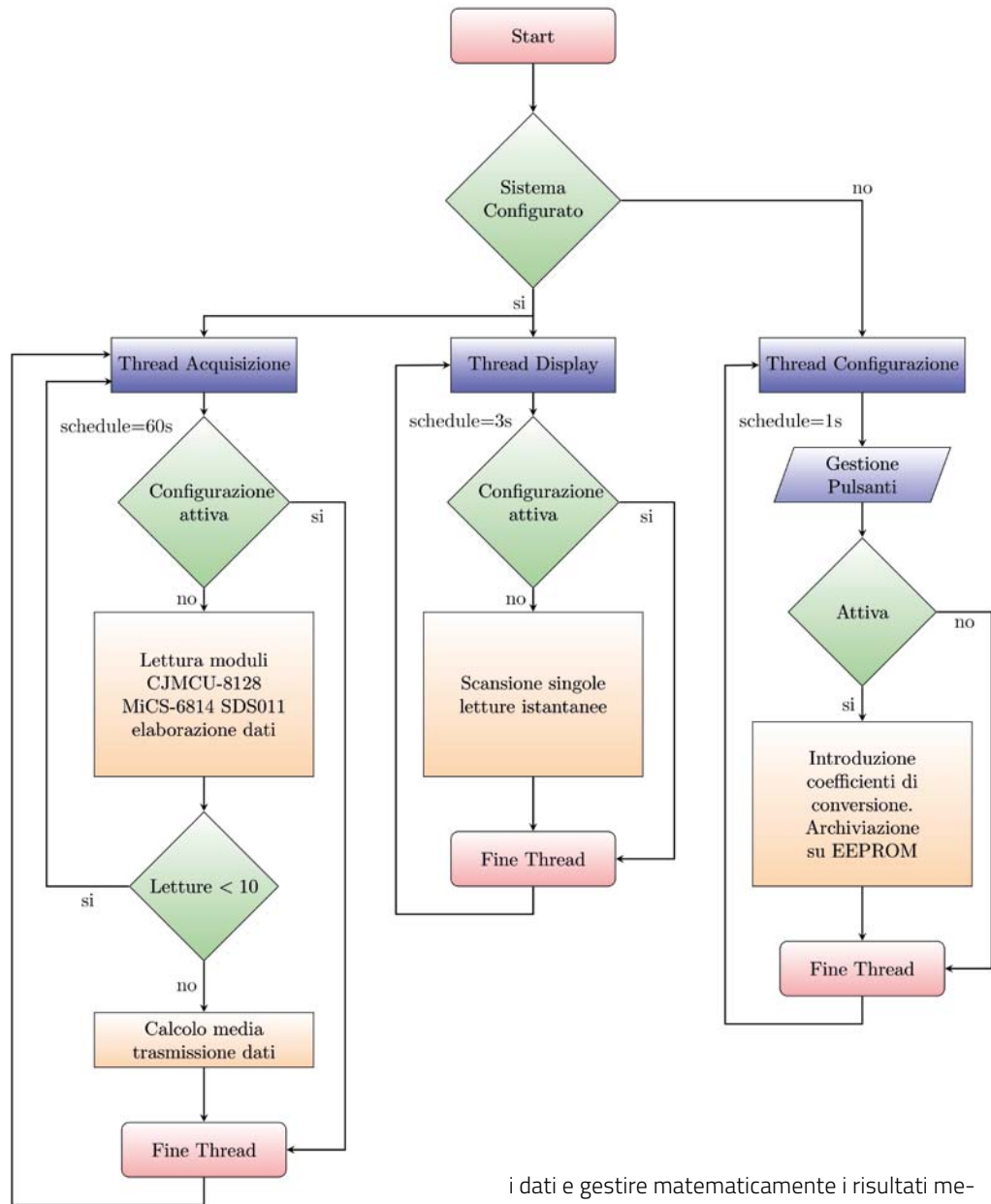
Il firmware del nostro progetto è meglio descritto dal flow-chart proposto nella **Fig. 1**, che evidenzia le routine base componente il codice scritto per il sistema di monitoraggio ambientale. L'ossatura del programma è costituita da tre

Dotata di sensori specifici per gas e polveri, rileva anche grandezze meteorologiche e permette un'analisi approfondita dell'inquinamento atmosferico, trasmettendo su rete LoRa i valori. Seconda e ultima puntata.

di **VALTER SARTORI**



→ **Fig. 1**
Diagramma
a blocchi del
firmware.



routine eseguite mediante l'utilizzo di Thread, algoritmo software che permette l'esecuzione di ogni singola routine schedulando opportunamente i singoli intervalli.

Nella **Fig. 2** sono evidenziate le istruzioni che configurano i singoli Thread.

Come si può vedere il programma è stato creato utilizzando l'IDE di Visual Studio Code con l'estensione PlatformIO.

Osservando il flow-chart possiamo notare che le tre routine hanno lo scopo di, la prima, acquisire

i dati e gestire matematicamente i risultati mediandoli su un intervallo di dieci minuti, la seconda, visualizzare i dati elementari a display ed eventuali allarmi, la terza è dedicata alla configurazione e memorizzazione su eeprom dei coefficienti di calcolo.

Il programma è costituito da sette classi di oggetti oltre al file main.cpp e al suo header Auxiliary.h; la **Fig. 3** mostra l'elenco delle classi utilizzate. Ogni classe è stata realizzata per gestire il proprio modulo di acquisizione. Ad esempio la classe Ads1115.cpp ha permesso la completa gestione del convertitore A/D ADS1115 e la conversione in concentrazioni volumetriche delle letture voltmetriche derivate dai tre circuiti a corrente costante.

```

33 // Creazione oggetti
34 ThreadController ctr = ThreadController(); // inizializzazione controllo Thread
35 Thread acquisizione = Thread(); // inizializzazione Thread di acquisizione
36 Thread display = Thread(); // inizializzazione Thread di visualizzazione misure a display
37 Thread configurazione = Thread(); // inizializzazione Thread di configurazione con controllo pulsanti
38 ThreadController groupOfThreads = ThreadController(); // inizializzazione oggetto di controllo threads multipli

```

```

display.onRun(lcd_display); // inizializzazione routine gestione display
display.setInterval(5000); // eseguita ogni 5 s
acquisizione.onRun(analisi); // inizializzazione routine acquisizione dati
acquisizione.setInterval(60000); // eseguita ogni 60 s
configurazione.onRun(setMode); // inizializzazione routine di configurazione parametri di calcolo
configurazione.setInterval(1000); // gestione pulsanti per setup
ctr.add(&configurazione);
groupOfThreads.add(&display);
groupOfThreads.add(&acquisizione);
ctr.add(&groupOfThreads); // configurazione Threads

```

Fig. 2
Configurazione
tramite Thread.

Sono state realizzate le classi per la gestione del modulo CJMCU 8128, per il display, per il trasmettitore LoRa, per il polverimetro SDS011, per la gestione della configurazione e in fine per la gestione della porta seriale.

La **Fig. 4** e la **Fig. 5** illustrano i metodi che vengono richiamati dagli oggetti delle classi Sds011.cpp e Ads1115.cpp.

Particolare attenzione può essere posta osservando i metodi readPm10() e readPm25() della classe Sds011, usata per il nostro polverimetro, dove vengono elaborati i dati provenienti dalla connessione seriale e ricostruiti per fornire un risultato leggibile. La creazione delle classi sopra citate ha permesso una stesura del codice assai ordinata e facilmente modificabile in base alle esigenze che si presentavano di volta in volta e si presenteranno in futuro. L'utilizzo di oggetti ha permesso anche un notevole risparmio di memoria su Arduino Uno, con un'occupazione della RAM di circa l'89%.

La classe LoRaBoard.cpp permette l'utilizzo delle routine di trasmissione dati, richiamate circa ogni dieci minuti per la trasmissione dei dati mediati. Risulta interessante osservare il metodo loraSend() dove viene composto il pacchetto dati con il codice attribuito all'elemento e il valore misurato. Questi due campi sono suddivisi dal carattere 'I' che permette ad un programma di ricezione dati la possibilità di "splittare" le informazioni. Il metodo può essere comunque modificato e personalizzato per trasferire con qualsiasi formato i dati raccolti (**Fig. 6**). Durante le varie fasi di misura, il firmware effettua un controllo sulla funzionalità delle reti bus di collegamento, presentando nell'ultima riga del display una stringa di allarme.

Gli allarmi presi in considerazione riguardano la mancata inizializzazione del modulo LoRa (a display: LoRa fault), l'anomalia della trasmissione seriale dell' SDS011, con la verifica dell'header del pacchetto dati (a display: TX PM fault) e la mancanza dati in uscita dall' SDS011 (a display: Dati PM fault).

Se non si presenta nessun allarme, viene rappresentata a display la scritta 'OK', è possibile che si presenti momentaneamente un allarme, ma se si risolve, si ripresenta la scritta 'OK'.

La risoluzione di eventuali allarmi costanti presenti a display, potrebbe richiedere in primis un riavvio del sistema e, se non sufficiente, un controllo sui cavi di connessione e sui connettori di supporto dei moduli.

Per ottenere risultati attendibili occorre effet-

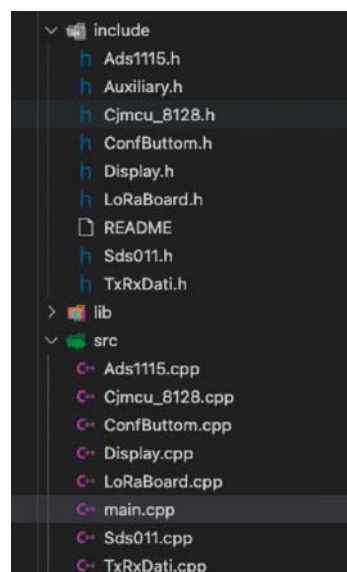


Fig. 3
Classi e Header
utilizzati.

```

#include <Arduino.h>
#include <Sds011.h>
#include <stdio.h>

SoftwareSerial sds(RX, TX);

Sds011::Sds011()
{
}

void Sds011::sdsInit() // Inizializza la comunicazione seriale
{
    sds.begin(9600);
}

void Sds011::svuotaBuffer() // Azzerare il buffer di comunicazione
{
    while(sds.available())
    {
        _tmp = sds.read();
    }
}

void Sds011::cancellaFrame() // Azzerare la matrice char frame per poter accogliere i nuovi dati
{
    memset(frame, 0, N_CH);
}

void Sds011::memoFrame() // Lettura porta seriale
{
    for(int i=0; i<N_CH; i++)
    {
        frame[i] = sds.read();
        delay(10);
    }
}

bool Sds011::startFrame() // verifica la correttezza della trasmissione controllando il carattere di header
{
    startDelimiter = frame[0];
    if(startDelimiter == 0xAA)
    {
        _sd = true;
    }
    else
    {
        _sd = false;
    }
    return _sd;
}

float Sds011::readPm25() // conversione e calcolo concentrazione PM 2.5
{
    _byte_1 = frame[2];
    _byte_2 = frame[3];
    _pm2_5 = _byte_1;
    _pm2_5 += (_byte_2 << 8);
    _polveri2_5 = ((float)_pm2_5)/10;
    return _polveri2_5;
}

```

 **Fig. 4**
 La routine per
 la gestione
 dell'SDS011 da
 firmware.

```

float Sds011::readPm10() // conversione e calcolo concentrazione PM 10
{
    _byte_1 = frame[4];
    _byte_2 = frame[5];
    _pm10 = _byte_1;
    _pm10 += (_byte_2 << 8);
    _polveri10 = ((float)_pm10)/10;
    return _polveri10;
}

bool Sds011::dataReady()
{
    if(sds.available() > 9) // Verifica la disponibilità dei dati
    {
        _sd = true; // dati pronti
    }
    else
    {
        _sd = false; // dati non disponibili
    }
    return _sd;
}

```

```

#include "Arduino.h"
#include "Ads1115.h"
#include <math.h>

Adafruit_ADS1115 ads;

void Ads1115::adsInit() // configurazione ADS1115
{
    ads.setGain(GAIN_ONE);
    ads.begin();
}

float Ads1115::adsGetValue(int channel) // lettura canali analogici e conversione in mV
{
    _channel = channel;
    _adc = ads.readADC_SingleEnded(_channel);
    adc_v = _adc * 0.125; // in mV
    return adc_v;
}

float Ads1115::resistanceValue() // calcolo della resistenza e assegnazione
coefficienti all'oggetto
{
    switch(_channel)
    {
        case 0:
            adc_r = (adc_v/1000)/_I_CO;
            cf_1 = _CO_1;
            cf_2 = _CO_2;
            p_mol = _MOL_CO;
            break;

        case 1:
            adc_r = (adc_v/1000)/_I_NH3;
            cf_1 = _NH3_1;
            cf_2 = _NH3_2;
            p_mol = _MOL_NH3;
            break;

        case 2:
            adc_r = (adc_v/1000)/_I_NO2;
            cf_1 = _NO2_1;
            cf_2 = _NO2_2;
            p_mol = _MOL_NO2;
            break;
    }
    return adc_r;
}

float Ads1115::ppmConvert(float R0) // calcolo concentrazione in ppm
{
    _R0 = R0;
    ratio = adc_r/_R0;
    ppm = cf_1 * pow(ratio, cf_2);
    return ppm;
}

float Ads1115::molConvert(float tAmb) // calcolo concentrazione volumetrica
{
    _tAmb = tAmb;
    concentration = (ppm * 12.187 * p_mol)/(273.15 + _tAmb);
    return concentration; // in mg/mc
}

```

Fig. 5
La routine
firmware per
la gestione
dell'Ads1115.

tuare una serie di tarature con l'ausilio di misure campione rilevate da centraline ambientali di zona.
Non avendo a disposizione bombole con miscele

di gas certificate con cui potremmo inondare i sensori ed essendo questa procedura assai costosa, possiamo fare riferimento ai valori misurati dalle centraline gestite dagli enti preposti

➔ **Fig. 6**
La classe_
LoRaBoard_cpp.

```
#include <Arduino.h>
#include <LoRaBoard.h>

bool LoRaBoard::loraInit()
{
    LoRa.setPins(csPin, resetPin, irqPin);
    if(!LoRa.begin(frequency))
    {
        return true;
    }
    else
    {
        return false;
    }
}

void LoRaBoard::loraRxMode()
{
    LoRa.enableInvertIQ();
    LoRa.receive();
}

void LoRaBoard::loraTxMode()
{
    LoRa.idle();
    LoRa.disableInvertIQ();
}

void LoRaBoard::loraSend()
{
    LoRa.beginPacket();
    LoRa.print(" ");
    LoRa.print(_codElement);
    LoRa.print("|");
    LoRa.print(_value);
    LoRa.print(";");
    LoRa.endPacket(true);
}

void LoRaBoard::loraSetValue(float value)
{
    memset(_value, 0, DIMFRAME);
    dtostrf(value, 0, 1, _value);
}

void LoRaBoard::loraCodElemento(int codice)
{
    memset(_codElement, 0, DIMID);
    itoa(codice, _codElement, 10);
}
```

➔ **Fig. 7**
Schermata del
software di con-
figurazione della
centralina.

al monitoraggio dell'aria e, a ritroso, pervenire al calcolo della RO per ognuno degli elementi quali CO, NH₃, NO₂.

I sensori utilizzati in questo progetto hanno una buona classe di precisione ma, prendendo ad esempio il barometro, il valore assoluto della misura potrebbe avere uno scarto rispetto al valore letto da un sistema di monitoraggio ambientale certificato. Il motivo è dato dal fatto che la lettura fornita dal nostro sensore possiede un offset che può essere corretto via software.

Per attuare queste correzioni abbiamo utilizzato la EEPROM di Arduino Uno per memorizzare permanentemente tutti i coefficienti usati per il calcolo sia delle concentrazioni, sia dei parametri ambientali.

La terza routine del programma, il thread di configurazione, è dedicata proprio a questa funzione e utilizza un programma esterno realizzato in Python che, sfruttando le librerie PyQt5 e la trasmissione via seriale, visualizza i dati presenti sulla scheda e, mediante particolari algoritmi di calcolo, ne permette la correzione.

Per quanto riguarda i parametri atmosferici (pressione, temperatura, umidità) vengono ricalcolati i coefficienti di offset, mentre per gli elementi CO, NH₃ e NO₂ il programma permette il ricalcolo della RO sempre avvalendosi di particolari algoritmi. La selezione dell'elemento da tarare, la predisposizione software per la memorizzazione dei parametri da memorizzare su EEPROM e la gestione di questa routine del programma è permessa dai tre pulsanti presenti sulla scheda display.

Il pulsante P1 attiva la routine e in un secondo momento permette la memorizzazione su EEPROM, P2 elenca ogni singola grandezza disponibile alla taratura e P3 è dedicato all'uscita dalla routine.

Durante queste operazioni, il controllo dei pulsanti è gestito sia a livello locale (di pannello display), sia a livello di software in esecuzione sul Personal Computer. Per ottenere i coefficienti più corretti possibile occorrerà affidarsi a dati provenienti da centraline gestite dagli enti locali o a misure ambientali fornite da stazioni meteorologiche con un buon grado di precisione.

Venendo a conoscenza dei valori degli elementi da noi misurati attraverso dati reali certificati, abbiamo a disposizione i valori corretti che possiamo immettere nel software di configurazione Air-QualitySerial.py.

La **Fig. 7** evidenzia il form con cui si presenta il programma appena avviato. Per poter utilizzare il

software sopra citato, occorre configurare Python con le librerie PyQt5, disponibili gratuitamente in Rete. Facendo riferimento a questa immagine, possiamo vedere in alto a sinistra il widget tipo Combo-Box dove sono elencate le porte seriali attive collegate al pc. Prima di avviare questo software è fondamentale collegare mediante cavo il Personal Computer alla scheda Arduino Uno.

Aperto il combobox, occorre selezionare la porta seriale alla quale è collegata la scheda Arduino e premere il pulsante Connessione.

Se l'handshake è andato a buon fine verrà presentato il nome della porta sulla label centrale e Arduino Uno sarà interessata da un segnale di reset.

Dopo aver atteso qualche minuto dal reset, occorre premere il pulsante P1 fino a quando comparirà a display un nuovo pannello grafico con menù, come dimostrato in **Fig. 8**.

Premendo il pulsante P2 appariranno sia a display, sia a pc i dati relativi ad ogni elemento di misura. Ad ogni pressione del pulsante P2 il firmware della scheda scansiona le grandezze soggette a taratura e presenta i dati sia a display, sia a video sul Personal Computer (**Fig. 9**).

Se notiamo uno scostamento importante tra il valore letto dai nostri sensori e il valore corretto certificato, possiamo inserirlo, facendo attenzione all'unità di misura (mg/m^3 piuttosto che $\mu\text{g}/\text{m}^3$ oppure hPa corrispondenti al mbar piuttosto che mmHg).

A video sono ben evidenziate le unità di misura con cui la scheda lavora, per cui all'occorrenza occorrerà convertire i dati.

Dopo aver introdotto il valore corretto nell'edit di testo contrassegnato, possiamo avviare



Fig. 8
Menù a display.

il ricalcolo con il pulsante omonimo. Nel caso della pressione atmosferica, della temperatura e dell'umidità verrà ricalcolato l'offset.

Nel caso degli elementi CO, NH₃, NO₂ verrà ricalcolata la R0 mediante un algoritmo funzione inversa dell'equazione legata alle curve proposte nelle **Fig. 13**, **Fig. 14** e **Fig. 15** della prima puntata di questo articolo, cui vi rimandiamo.

Il nuovo coefficiente correttivo apparirà nella casella testuale in fondo alla finestra, pronto ad essere trasmesso (**Fig. 10**).

Prima di premere il pulsante Trasmette, occorre abilitare la seriale della scheda Arduino Uno a ricevere i dati, per cui occorre premere nuovamente il pulsante P1. Sul display apparirà una casella lampeggiante che indica che il sistema è pronto a ricevere i dati (**Fig. 11**).

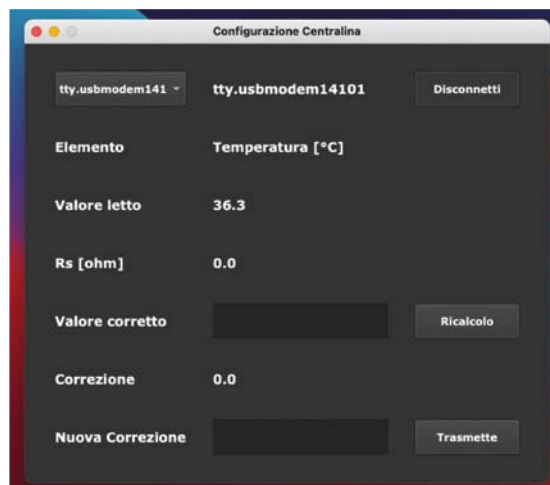
A questo punto possiamo premere il pulsante Trasmette e a display comparirà il nuovo coefficiente, memorizzato anche su EEPROM (**Fig. 12**). Questa procedura deve essere ripetuta per ogni elemento che differisce in modo sostanziale dal valore certificato reale.

I valori trasmessi attraverso il modulo wireless LoRa potranno essere ricevuti da un sistema di elaborazione compatibile.

Possiamo dare una dimostrazione di cosa è pos-

Fig. 9

Visualizzazione a display e su PC dei dati e dei coefficienti di calcolo.



→ **Fig. 10**
Schermata
del PC per la
correzione dei
valori misurati.



sibile ottenere dal nostro sistema di monitoraggio ambientale offrendo alcuni screenshot tratti da un programma, sempre scritto in Python, che, gestendo un database, permette una elaborazione grafica di ogni singolo elemento misurato (Fig. 13).

ATTIVAZIONE DEL SISTEMA DI MONITORAGGIO

Al termine della realizzazione pratica delle schede, occorre procedere per gradi verificando la funzionalità di ognuna delle schede che compongono il progetto.

Iniziando dalla scheda alimentatore, applicare una tensione in ingresso (Vin) sul connettore CN1 preferibilmente del valore di 12 Vcc, senza problemi di polarità in quanto i diodi D2, D3, D4 e D5 proteggono la scheda da questo problema.

I LED siglati LD1, LD2 e LD3 si illumineranno indicando la presenza di ogni singola tensione di alimentazione.

Utilizzando i Test-point presenti sulla scheda, verifichiamo il valore delle tensioni presenti sulle varie uscite. Su Vin si ripresenterà la tensione applicata in ingresso, polarizzata correttamente, che sarà utilizzata per alimentare Arduino Uno. Su TP2 e TP3 verificheremo la correttezza dei valori di 5 V e 3,3 volt.



→ **Fig. 11**
Display in modalità
attesa dati
aggiornati.

Su TP4 verificheremo che la tensione possa variare tra 1,5V e 4V ruotando il trimmer RV1.

Al termine di questi test sarà possibile collegare a CN3 il flat-cable per alimentare la scheda AnalyzeShield.

La scheda dovrà essere ancora tenuta separata da Arduino Uno e non dovrà essere corredata di alcun modulo o altri collegamenti.

Inizialmente si dovrà controllare la presenza delle tensioni sugli header montati e poi potremo procedere con la verifica dei circuiti a corrente impressa. La procedura per la configurazione dei tre circuiti prevede, a scheda non alimentata, l'inserzione del modulo di calibrazione (vedere pag 68 riv. 261 paragrafo: *Schede di calibrazione*) con le resistenze più basse. Collegando un multimetro tra il TP di massa e i singoli TP che simulano le misure di NO₂, NH₃, CO, dovrà essere riportata la tensione letta ai valori indicati nella **Tabella 1** nella prima puntata di questo articolo, mediante l'aggiustamento dei trimmer RV1 per NH₃, RV2 per CO e RV3 per NO₂.

Occorre mantenere il modulo di calibrazione per un tempo tale da raggiungere la regimazione termica dei componenti e controllare se i valori di tensione restano inalterati.

Terminata la prima calibrazione di corrente, sostituiamo la scheda di calibrazione a bassa resistenza con quella ad alta resistenza. Occorrerà procedere ad alcune correzioni dei trimmer. In questo caso è importante raggiungere i valori di tensione che più si avvicinano alle correnti teoriche e verificare le tensioni sostituendo più volte le schede di taratura.

Ultimata la configurazione dei circuiti a corrente impressa, possiamo inserire i moduli di misura e controllo dei segnali, nonché il modulo LoRa, facendo molta attenzione ad inserirli nel verso corretto. Tramite l'utilizzo di connettori flat dobbiamo collegare alla scheda AnalyzeShield la scheda Display e la scheda SDS011.

La scheda Display dovrà alimentare e gestire il display 2004 sempre attraverso un apposito flat-cable. Dalla scheda Alimentatore predisporremo i flat cable necessari all'alimentazione della scheda Display (CN2), della scheda Arduino Uno (CN5) e della ventola utilizzata per il trasporto del campione (CN4).

Quando il sistema viene avviato occorre tenere conto che l'acquisizione dei dati avviene ogni 60s, per cui inizialmente a display verranno visualizzati dati non corretti o semplicemente nulli.

Nel file main.cpp alla riga 830 (funzione setup())

Fig. 2) l'intervallo applicato al thread acquisizione (acquisizione.setInterval(60000)) può essere settato a 10 s per eventuali test e controlli, ma in via definitiva deve essere impostato a 60 s. La riga di codice è comunque commentata e vengono identificati i valori da impostare nei due casi. Verificato l'interfacciamento e la funzionalità delle schede, si può procedere con la configurazione di ogni singolo elemento, come precedentemente spiegato nel paragrafo dedicato alla descrizione del firmware.

CONCLUSIONI

Per dare vita al nostro progetto occorre alloggiare i componenti del sistema di analisi in modo che il flusso campione non venga alterato prima dell'analisi. Utilizzando un armadio chiuso possiamo alloggiare i componenti come si vede nelle foto in queste pagine, dove la scheda di analisi è localizzata verso il basso vicino alla ventola, ma non eccessivamente, in quanto occorre lasciare lo spazio per l'accesso al connettore seriale di Arduino Uno, che viene utilizzato durante la calibrazione. Questa è la soluzione adottata nel nostro prototipo, come vedete nelle foto in queste pagine. All'interno dell'armadio sono state predisposte alcune canaline percorse dai cavi di alimentazione, antenna e i flat di alimentazione delle schede. La scheda di alimentazione è stata alloggiata in maniera tale da avvicinare il più possibile i connettori in uscita alla canalina. La scheda del polverimetro è stata collegata alla griglia di prelievo mediante un tubo in silicone e la ventola è stata installata in modo da consentire di indirizzare il flusso d'aria direttamente sulla zona della AnalyzeShield dove sono alloggiati i moduli CJMCU-8128 e del MICS-6814; questo ottimizzerà il rilevamento. La scheda Display e il relativo LCD2004 sono stati montati centralmente per permettere un agevole accesso ai pulsanti.

In alto, montati su barra DIN troviamo i fusibili in ingresso e un interruttore di accensione del sistema. Alcuni morsetti facilitano le connessioni tra i componenti. Il cavo RG58 collegato al BNC della scheda AnalyzeShield viene portato all'esterno per essere collegato ad un'antenna a stilo 868 MHz (codice 8110-AS868). L'alimentazione in ingresso viene prelevata da una batteria esterna a 12 Vcc, onde evitare black-out. La batteria potrebbe essere connessa ad un sistema di ricarica a pannelli solari o ad un sistema convenzionale di mantenimento di carica collegato alla rete elettrica di distribuzione a 230 Vca. □



Fig. 12
Display con gli aggiornamenti memorizzati.



Fig. 13
Software per l'elaborazione dei dati. Attraverso un programma in Python, sfruttando le librerie pyqt5 è possibile l'elaborazione dei dati.



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. Il modulo qualità dell'aria con MICS-6814 (cod. MODMICS6814) è in vendita a Euro 45,00, il sensore con CCS811 (cod. MODCJMCU8128) è disponibile a Euro 48,50, il sensore in grado di misurare il particolato (cod. PMSSENSOR) costa Euro 56,00, il traslatore di livelli (cod. MODPCA9306) è disponibile a Euro 5,80, il DAC con ADS1115 (cod. MODADS1115) è in vendita a Euro 11,00, l'RTX LoRa (cod. RFM95-868) costa Euro 13,50. I prezzi si intendono IVA compresa. I circuiti stampati possono essere realizzati con il servizio di stampa PCB dal sito www.futurashop.it/ produzionePCB

Il materiale va richiesto a:

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

Nuovo allarme evoluto WiFi-GSM/GPRS



€119,00

Cod. FR783



Controllabile con:



Smart Life



Tuya Smart



Amazon Alexa



Google Home

Centralina antifurto wireless programmabile, controllabile da remoto tramite Smartphone sfruttando la rete GSM/GPRS o Wi-Fi e direttamente con Amazon Alexa o Google Home. Le App "Smart Life" o "Tuya Smart" permettono di configurare e gestire il funzionamento dell'allarme in ogni suo dettaglio. È possibile aggiungere sensori ed accessori opzionali per coprire fino a 99 zone. La confezione comprende: la centralina, due radiocomandi, un sensore wireless per porte/finestre, un sensore PIR wireless, due chiavi TAG RFID, la batteria di backup e l'alimentatore di rete. Manuale di utilizzo in italiano.

Prezzi IVA inclusa



Accessori
opzionali:

Sensore PIR
wireless



€14,00

Cod. FR783PIR

Sensore
porte/finestre
wireless



€11,00

Cod. FR783MAGN

Radiocomando
supplementare



€9,00

Cod. FR783RAD

Transponder
a portachiavi



€5,00

Cod. FR783TAG

Sirena wireless
da interno



€29,00

Cod. FR783SIR

Sirena wireless
da esterno con
pannello solare



€65,00

Cod. FR783SIREXT

Temperatura sempre sotto controllo!

Termometri a infrarossi senza contatto



€ 38,00
Cod. TP213N

Termometro IR con puntatore LASER (da -20°C a +400°C)

Termometro a infrarossi UNI-T con puntatore LASER per la misura a distanza della temperatura (da -20°C a +400°C). Dispone di display LCD con retroilluminazione, possibilità di visualizzazione in gradi centigradi o in gradi Fahrenheit, spegnimento automatico. La lettura della temperatura è istantanea. La confezione comprende: il termometro e due batterie da 1,5 V tipo AAA.

Termometro IR con puntatore LASER (da -32°C a +420°C)

Termometro a infrarossi UNI-T con puntatore LASER per la misura a distanza della temperatura (da -32°C a +420°C). Dispone di display LCD con retroilluminazione, possibilità di visualizzazione in gradi centigradi o in gradi Fahrenheit, spegnimento automatico. La lettura della temperatura è istantanea (250 ms) con la memorizzazione della soglia minima e massima in caso di scansione continua e prolungata. Memorizzazione fino a 5 set di valori. La confezione comprende: il termometro, la pila da 9 V e la borsa da cintura per il trasporto.

€ 78,00
Cod. TP224



Prezzi IVA inclusa.

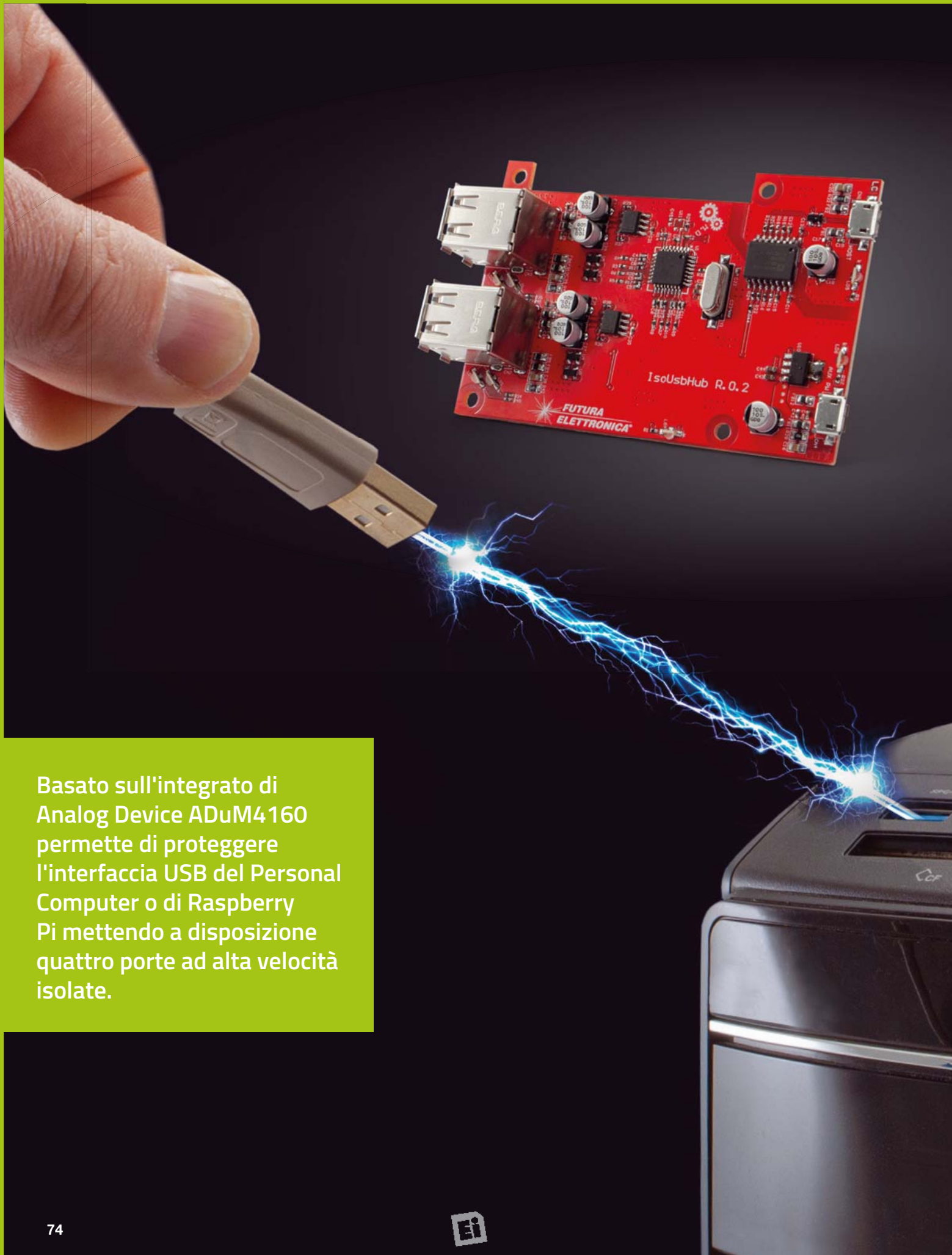


€ 19,00
Cod. FR811

Termometro IR per la lettura della temperatura corporea o di superfici (da +34°C a +42,9°C)

Dispone di ampio display a LED retroilluminato che mostra la lettura in gradi centigradi o Fahrenheit (selezionabile). Gli infrarossi consentono al termometro di leggere la temperatura corporea a una distanza di 5-10 cm in meno di un secondo. L'apparecchio si spegne dopo 8 secondi di standby. Il termometro ha una precisione del display di 0,1°C con un errore massimo di $\pm 0,2^\circ\text{C}$ e un range $34,0^\circ\text{C} \sim 42,9^\circ\text{C}$. Dispone inoltre di segnalazione di allarme con soglia impostabile, cambio di colore del display per indicare le soglie di temperatura misurate. Offre la possibilità di memorizzare fino a 32 misurazioni.





Basato sull'integrato di Analog Device ADuM4160 permette di proteggere l'interfaccia USB del Personal Computer o di Raspberry Pi mettendo a disposizione quattro porte ad alta velocità isolate.

HUB USB 2.0 FULL SPEED ISOLATO

di **MATTEO DESTRO**

A

l giorno d'oggi l'interfaccia USB è largamente utilizzata in moltissime applicazioni per collegare al computer (oppure anche al Raspberry Pi o altri device) periferiche di ogni genere che possono andare dalla semplici stampanti multifunzione, mouse, schede Arduino, ecc. agli apparecchi industriali o biomedicali.

In contesti di applicazione al di fuori dell'ambito domestico, e quindi in condizioni ambientali particolarmente ostili, potrebbe essere necessario ricorrere ad opportune tecniche di isolamento e protezione dell'interfaccia USB.

In questo articolo vediamo come realizzare un HUB USB con interfaccia verso il computer isolata; l'integrato che si occuperà di assolvere a questo compito è il transceiver ADuM4160 della Analog Device.

Quando si collega un dispositivo industriale o elettromedicale al computer tramite il cavo USB potrebbe succedere che la periferica collegata abbia un differente potenziale di terra rispetto alla presa a cui è collegato il computer. In questo caso potrebbe crearsi un percorso verso terra attraverso il cavo USB il quale potrebbe provocare un danneggiamento dei componenti incontrati lungo il percorso di alimentazione, ad esempio sulla scheda madre.

Questa è una condizione potenzialmente distruttiva e che bisognerebbe evitare.

Oltre a questo caso, l'introduzione di un dispositivo con porta USB isolata porterebbe alla soluzione dei problemi legati ad una alimentazione non sufficiente fornita dalla porta stessa. Infatti ricordiamo che le porte USB standard forniscono al massimo 500mA di corrente la quale per alcuni dispositivi potrebbe non essere sufficiente.

Viene quindi comodo l'utilizzo di un transceiver USB in quanto l'alimentazione necessaria dal lato della periferica verrà fornita da appositi alimentatori i quali sono totalmente svincolati dalla sezione di alimentazione presente sulla scheda madre del computer a cui ci si collega.

Quindi possiamo riassumere che l'isolamento galvanico viene utile in applicazioni dove:

- Si rende necessario fornire una sicurezza elettrica nei confronti di tensioni potenzialmente letali
- Si rende necessario fornire una sicurezza elettrica in caso di tensioni potenzialmente distruttive per l'host che fornisce la connessione USB come ad esempio un computer, Raspberry Pi ecc



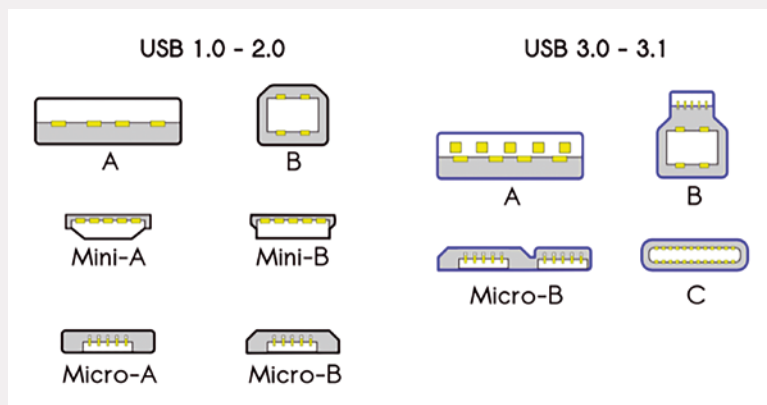


Fig. 1
Tipologia
connettori
USB.

- Può essere usato per eliminare gli effetti del rumore elettrico e delle differenze di tensione di modo comune create dal loop di terra
- Per fornire alimentazioni con correnti superiori a 500mA
- Può essere usato come traslatore di livello. In moltissime applicazioni industriali le elettroniche a microprocessore sono alimentate a +3V3 e quindi anche l'interfaccia USB potrebbe avere bisogno di lavorare al livello di tensione di +3V3

Per risolvere i problemi sopra esposti ci viene in aiuto, come già accennato, il transceiver ADuM4160 il quale garantisce un isolamento fino a 5000V per un massimo di 60 secondi, realizzato con la tecnologia proprietaria iCoupler ed elettroniche CMOS ad alta velocità per la gestione delle comunicazioni USB da ambo i lati.

LA STORIA DELLO STANDARD USB E LA SUA EVOLUZIONE

Per chi non lo sapesse l'acronimo USB sta per Universal Serial Bus ed è uno standard di comunicazione seriale il quale attraverso un cavo porta alla periferica sia la comunicazione, sotto forma di segnale differenziale, sia l'alimentazione (5V, 500mA).

Lo sviluppo dello standard iniziò nel 1994 e al progetto hanno partecipato le società Compaq, Hewlett-Packard, IBM, Microsoft e Nortel. L'obiettivo che si intendeva raggiungere era la semplificazione della connessione di dispositivi esterni al PC sostituendo la moltitudine di connettori presenti sul retro dei computer dell'epoca (RS232, LPT, SCSI ecc) e allo stesso tempo aumentare la velocità di comunicazione e trasferimento dati nonché rendere più semplice la configurazione software dei dispositivi. Quindi le società sopra indicate hanno lavorato presso la Intel per dare vita ai primi

integrati che supportavano il nuovo standard USB, era il lontano 1995.

La specifica USB 1.0 è stata introdotta nel Gennaio del 1996 e il primo sistema operativo che ne diede supporto fu Windows 95. Nel 1998 anche Apple, con iMAC G3, diede supporto al nuovo standard rimuovendo tutte le interfacce RS232, la porta ADB, SCSI ecc. Questa mossa portò altri produttori a passare al nuovo standard.

Nel 2000 fu rilasciata la specifica USB 2.0 mentre nel Novembre del 2008 uscì la specifica USB 3.0 a cui seguirono le specifiche USB 3.1 nel Luglio del 2013 e USB 3.2 nel Settembre 2017. Quest'ultima esiste in quattro differenti varianti. I due standard USB 3.0 e 3.1 oggi corrispondono agli standard USB 3.1 Gen 1 e USB 3.1 Gen 2 giusto per confondere un pochino le idee. Nell'Agosto del 2019 è stata rilasciata la specifica USB 4.0.

Ad ogni specifica rilasciata si ha avuto un incremento di prestazioni per quanto riguarda la velocità nel trasferimento dati, infatti:

- USB 1.0 – Low Speed 1.5 Mbit/s
- USB 1.1 – Full Speed 12 Mbit/s
- USB 2.0 – High Speed 480 Mbit/s
- USB 3.1 Gen 1 – Super Speed 4.8 Gbit/s
- USB 3.1 Gen 2 – Super Speed+ 10 Gbit/s
- USB 3.2 Gen 1x1 – 5 Gbit/s (praticamente è l'equivalente della USB 3.1 Gen 1 o USB 3.0)
- USB 3.2 Gen 1x2 – 10 Gbit/s
- USB 3.2 Gen 2x1 – 10 Gbit/s (praticamente è l'equivalente della USB 3.1 Gen 2)
- USB 3.2 Gen 2x2 – 20 Gbit/s
- USB 4.0 – 40 Gbit/s

Con l'evoluzione delle specifiche oltre alla velocità di trasferimento dati sono cambiati anche i connettori USB utilizzati per i collegamenti tra il computer e le varie periferiche fino ad arrivare al più recente e maturo connettore Type-C.

La **Fig. 1** mostra la tipologia, ad oggi, dei connettori USB disponibili sul mercato.

Il sistema USB è asimmetrico questo vuol dire che c'è un singolo gestore (HOST) e molte periferiche collegate ad albero (CLIENT) attraverso dei dispositivi chiamati HUB o concentratori. Sono supportate un massimo di 127 periferiche per gestore ma nel conteggio devono essere calcolati anche il gestore e gli HUB. Detto ciò il numero di periferiche collegabile è leggermente inferiore. La lunghezza massima del cavo non deve superare i 5 metri oltre questo limite sono necessari degli HUB che ne amplifichino il segnale. Il numero massimo di HUB che si possono collegare in cascata è pari a cinque.

- Perfil 1: 5V@2A (10W)
- Perfil 2: 5V@2A (10W) o 12V@1,5A (18W)
- Perfil 3: 5V@2A (10W) o 12V@3A (36W)
- Perfil 4: 5V@2A (10W) o 12V/20V@3A (36W/60W)
- Perfil 5: 5V@2A (10W) o 12V/20V@5A (60W/100W)

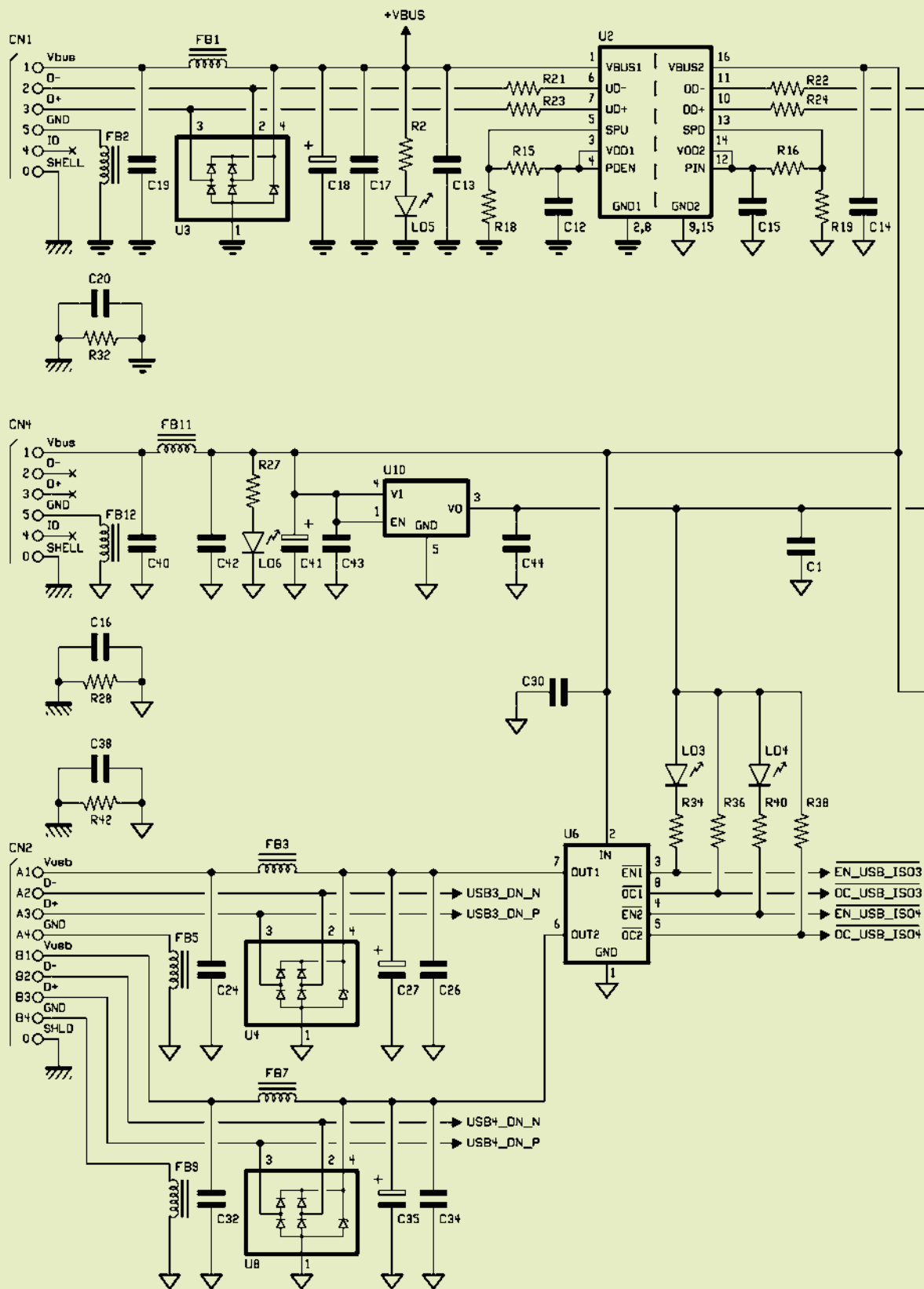
DESCRIZIONE SCHEMA ELETTRICO

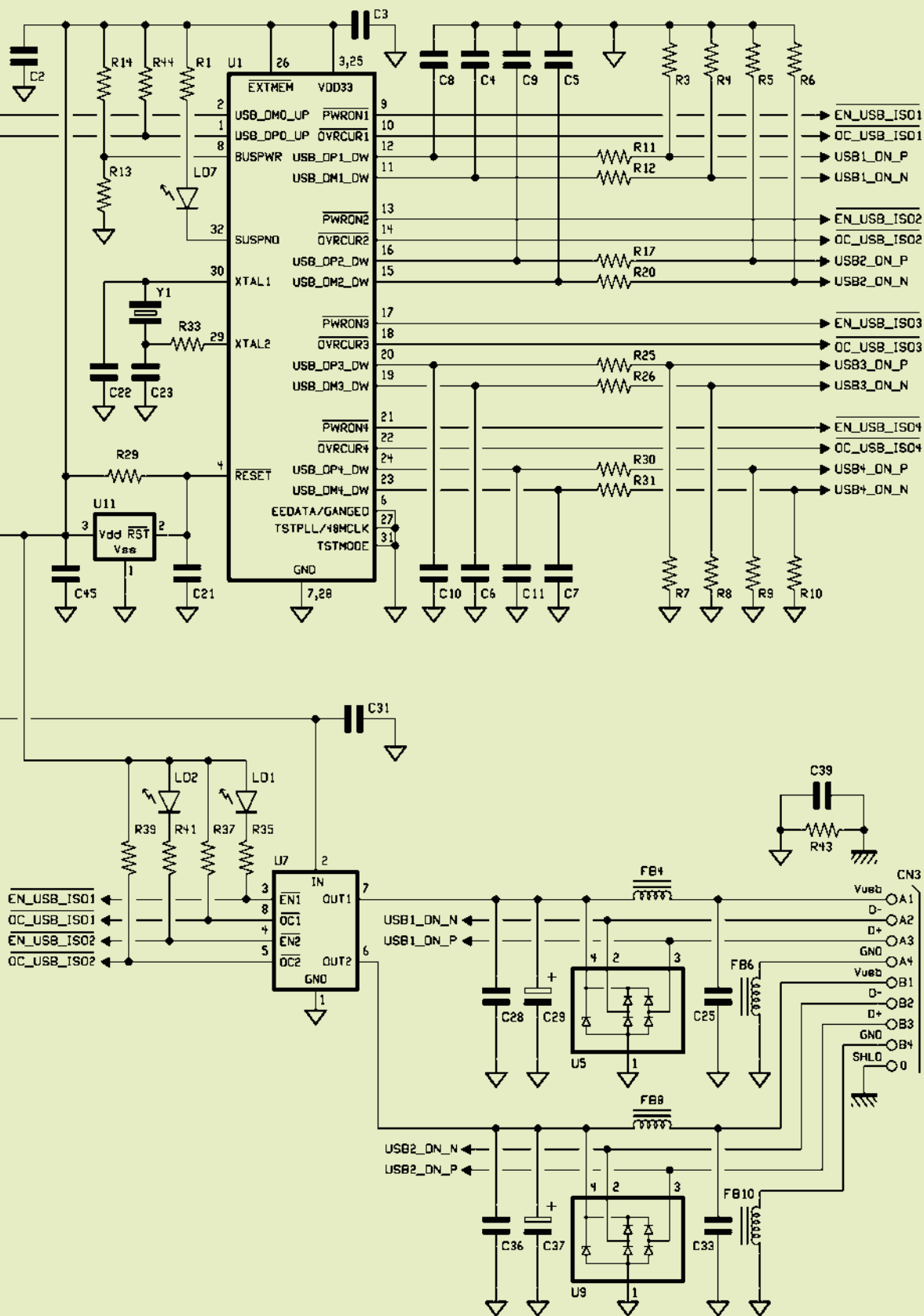
blocchi visibile in **Fig. 2**. Come si può osservare ci sono due connettori μ USB, il primo serve per collegare il computer alla scheda e quindi porta sia le linee dati che l'alimentazione necessaria alla sezione isolata mentre la seconda μ USB serve come alimentazione ausiliaria e quindi fornisce le alimentazioni alla sezione di elettronica adibita al collegamento delle periferiche che si vuole isolare dal computer. In questo caso le linee dati del secondo connettore non servono.

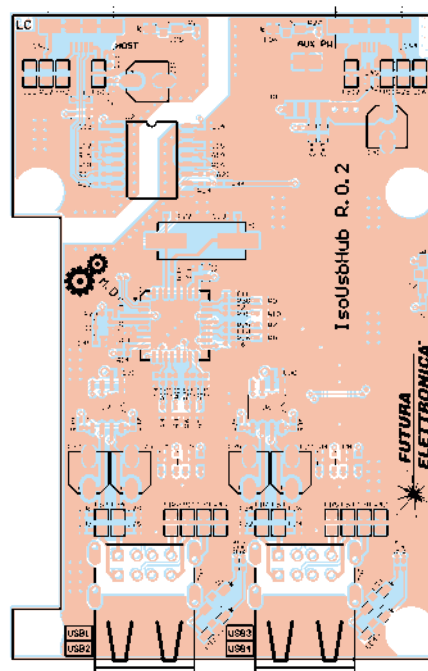
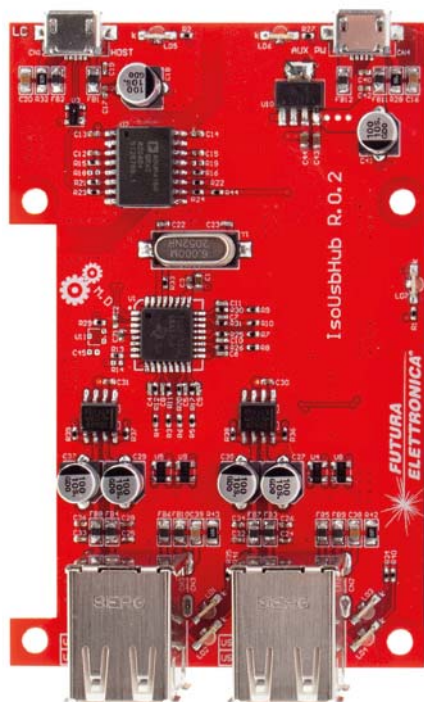
Passiamo allo schema elettrico iniziando la trattazione dal connettore μ USB CN1 il quale, come detto prima, porta sia l'alimentazione che i dati. In pratica è il collegamento all'Host device il quale potrebbe essere un computer oppure una

The diagram illustrates a USB hub circuit with the following components and connections:

- USB HOST (UPSTREAM):** A USB connector connected to the ADuM4160.
- Analog Device ADuM4160:** A digital isolator that provides isolation between the USB Host and the TUSB2046B. It has pins for +5VDS, D0+, D0-, D1+, D1-, and +5V_ISO.
- Auxiliary Power Supply:** A power source providing +5V and +5V_ISO to the ADuM4160 and the TUSB2046B.
- Texas Instruments TUSB2046B:** A USB hub controller that manages the downstream ports. It has pins for EN 1-4, OC 1-4, +5VDS, +5VDS_ISO, +5VDS_ISO2, +5VDS_ISO3, +5VDS_ISO4, USB_DP0_UP, USB_DM0_UP, +5V, and four pairs of USB_DP1_DW and USB_DM1_DW.
- Texas Instruments Power Distribution Switch:** A switch that provides +5V_ISO and EN/OC signals to the TUSB2046B and the downstream ports.
- USB DOWNSTREAM 1, 2, 3, 4:** Four USB connectors connected to the downstream ports of the TUSB2046B.







Elenco Componenti:

R1, R2, R27, R34, R35, R40, R41: 4,7 kohm 1%
 R3, R4, R5, R6, R7, R8, R9, R10, R29: 15 kohm 1%
 R11, R12, R17, R20, R21, R22, R23, R24, R25, R26, R30, R31
 22 ohm 1%
 R13, R14, R36, R37, R38, R39 10 kohm 1%
 R15, R16, R18, R19 0 ohm 1%
 R28, R32, R42, R43 1 Mohm 1%
 R33, R44 1,5 kohm 1%
 C1, C43, C44: 10 μ F 25 V ceramico
 C2, C3, C12, C13, C14, C15, C17, C26, C28, C30, C31,
 C34, C36, C42, C45: 100 nF 50 V ceramico (X7R / X5R)
 C4, C5, C6, C7, C8, C9, C10, C11: 22 pF 16 V ceramico
 (X7R / X5R)
 C16, C20, C38, C39: 1 nF 1kV ceramico (X7R)
 C18, C27, C29, C35, C37, C41: 100 μ 10 V elettrolitico
 C19, C24, C25, C32, C33, C40: 10 nF 50 V ceramico (X7R
 / X5R)
 C21: 1 μ F 16 V ceramico (X7R / X5R)
 C22, C23 27 pF 16 V ceramico (X7R / X5R)

LD1, LD2, LD3, LD: LED verde
 LD5 LED blu
 LD6: LED rosso
 LD7: LED giallo
 Y1: quarzo 6 MHz
 U1: TUSB2046BVF4RG4
 U2: ADuM4160BRWZ
 U3, U4, U5, U8, U9: PRTR5V0U2X
 U6, U7: TPS2042BQDRQ1
 U10: LP38693MP-3.3/NOPB
 U11: MIC803-30D2VC3
 FB1, FB2, FB3, FB4, FB5, FB6, FB7, FB8, FB9, FB10:
 BLM21PG121SN1D
 FB11, FB12: BLM21SP111BH1D
 CN1, CN4: Micro USB A/B SMT
 CN2, CN3: doppia presa USB A

Varie:

- Circuito stampato S1590 (85x56 mm)

Raspberry Pi o altro. Ad indicare l'inserimento del cavo μ USB nel connettore CN1, e quindi presenza di alimentazione dal lato computer, ci pensa il led blu LD5. Le due linee di alimentazione (+V_{BUS} e GND) sono filtrate tramite due ferriti per poi incontrare un TVS (Transient Voltage Suppressor) posto sia sulle linee dati che sul V_{BUS}, questo garantisce

un minimo di protezione in caso di scariche elettrostatiche.

Le linee dati raggiungono infine il transceiver ADuM4160 attraverso due resistenze da 22R e si collegano alla porta UpStream dello stesso. L'alimentazione V_{BUS} viene quindi collegata al pin 1 dell'integrato ADuM4160 il quale grazie al suo LDO

HUB USB 4 PORTE

La funzione di HUB USB viene demandata all'integrato Texas Instruments TUSB2046B il quale presenta le seguenti caratteristiche:

- Completamente compatibile con le specifiche USB Full-speed
- Una porta UpStream e quattro porte DownStream
- Tutte e quattro le porte DownStream supportano sia la velocità Full-speed (12Mbit/Sec) che Low-speed (1.5Mbit/Sec). Non è supportata la velocità High-Speed (480Mbit/Sec)
- Possibilità di monitorare le alimentazioni sulle quattro porte DownStream al fine di gestire eventuali assorbimenti anomali di corrente. Può lavorare sia su singola porta che su tutte le porte contemporaneamente; quest'ultima modalità è detta "GANGED mode"
- Possibilità di collegare una EEPROM esterna per la programmazione del "Vendor ID" e del "Product ID". Il bus di comunicazione verso la memoria EEPROM viene attivato portando basso il pin EXTMEM
- Non sono richiesti driver specifici per l'utilizzo con i più svariati sistemi operativi.

La Fig. A mostra uno dei possibili modi di utilizzo dell'integrato TUSB2046B, in particolare si vede una connessione a cascata di più HUB

USB al fine di ottenere fino a 10 porte per la connessione dei più disparati dispositivi. Si noti come due porte DownStream del primo TUSB2046B vengano usate per connettere le porte UpStream di altri due TUSB2046B.

La Fig. B mostra lo schema a blocchi interno dell'integrato. In alto si vede la porta UpStream da collegare alla porta USB del computer mentre in basso si vedono le quattro porte DownStream a cui collegare le periferiche. In basso a destra trova posto la sezione di gestione e controllo delle condizioni di anomalia sulle alimentazioni denominata "Hub Power Logic". Come detto prima l'integrato può lavorare in modo indipendente su ogni singola porta o su tutte assieme in modalità "GANGED mode". La modalità di funzionamento viene decisa tramite il pin 6 nominato EE-DATA/GANGED. Il pin 8, detto BUSPWR, serve per discriminare se le porte DownStream debbano essere alimentate direttamente dal cavo USB connesso al PC/RaspberryPi oppure se debbano essere alimentate da una sorgente di alimentazione indipendente. Ad esempio una alimentazione tramite connettore μ USB con i consueti caricabatterie degli smartphone/tablet o altro.

Completa l'elettronica la sezione dedicata al collegamento di una EEPROM

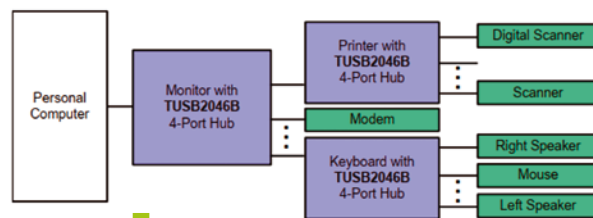
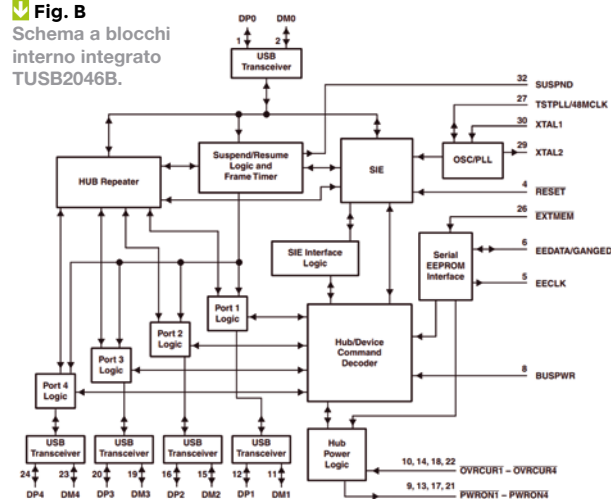


Fig. A
Esempio di utilizzo dell'Hub USB.

esterna facoltativa. Affinché l'HUB USB possa monitorare le alimentazioni delle porte DownStream ad esso connesse si devono introdurre degli appositi integrati per la distribuzione delle alimentazioni. Questi non sono altro che degli switch analogici con implementata l'elettronica per la limitazione della corrente in uscita. Se la corrente supera una soglia fissata dal costruttore si ha che l'interruttore elettronico viene aperto togliendo l'alimentazione alla porta DownStream interes-

sata dall'anomalia. Quando si verifica questo evento viene mostrato un messaggio di errore nel sistema operativo in uso a cui è connesso l'HUB USB. Nel nostro progetto abbiamo usato gli integrati TPS2042BQDRQ1 che gestiscono due alimentazioni in modo indipendente. Questi sono connessi all'integrato TUSB2046B tramite due linee elettriche; una per abilitare/disabilitare l'alimentazione (PWRON) e l'altra per segnalare l'anomalia all'HUB USB (OVRCLR).

Fig. B
Schema a blocchi interno integrato TUSB2046B.



interno può generare la tensione di uscita VDD1 usata per configurare la velocità Full Speed sulla linea dati lato UpStream (pin SPU) e abilitare i pull-down sulle linee DownStream (pin PDEN). Sul lato opposto del transceiver ritroviamo l'alimentazione V_{BUS_ISO} collegata al pin 16 dello stesso la quale viene fornita attraverso il connet-

tore μ USB CN4 che, come visibile nello schema a blocchi, serve solo per portare l'alimentazione ausiliaria alla scheda. Anche in questo caso ci sono due ferriti di filtro ma nessun TVS per le eventuali scariche elettrostatiche. Quando viene fornita alimentazione tramite il connettore CN4 il led rosso LD6 si accende.

Per isolare una linea USB differenziale si può prendere in considerazione l'integrato della Analog Device ADuM4160 il quale assicura la piena compatibilità alla USB 2.0 sia Low Speed che Full Speed.

Il componente ha le seguenti

caratteristiche:

- Protezione al cortocircuito sulla porta UpStream
- Immunità ai transienti di modo comune superiori a 25kV/us
- Soddisfa la normativa UL1577 la quale richiede che il dispositivo debba essere in grado di garantire una tensione di isolamento di 5000V per un minuto. Inoltre garantisce una tensione di isolamento pari al 120% del valore nominale ovvero 6000V applicati per un secondo

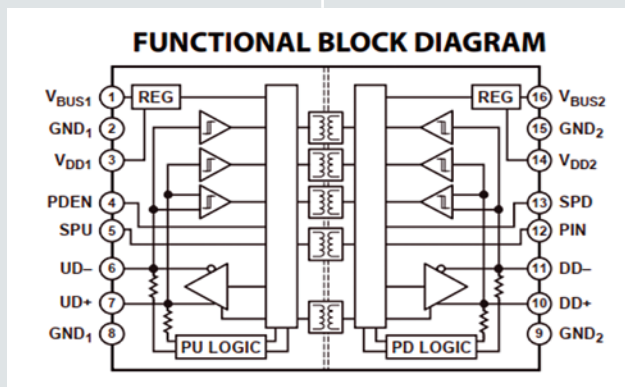
- Soddisfa la normativa IEC 60601-1. 125 V_{RMS} (176 V_p)
- Soddisfa la normativa IEC 60950-1. 380 V_{RMS} (537 V_p)
- Soddisfa la normativa VDE V 0884-10: 846V_p.

La Fig. C mostra lo schema a blocchi interno dell'integrato ADuM4160.

L'integrato sfrutta la tecnologia iCoupler che abbinata alla logica interna a CMOS ad alta velocità e a una costruzione monolitica dei trasformatori di isolamento interni permette di realizzare dispositivi di isolamento delle porte USB altamente efficienti e compatibili con gli standard Full Speed e Low Speed.

Inoltre la tecnologia iCoupled in-

Fig. C
Schema a blocchi interno integrato ADuM4160.



L'alimentazione V_{BUS_ISO} permette al transceiver di generare, attraverso il suo secondo LDO interno, la tensione di uscita VDD2 usata per configurare la velocità Full Speed sulla linea dati lato DownStream (pin SPD) e abilitare i pull-up sulle linee UpStream (pin PIN).

Le due linee dati lato DownStream vengono quindi portate all'HUB USB attraverso due resistenze da 22R; in più la linea dati D+ deve avere un pull-up da 1K5 (R44) collegato alla tensione di alimentazione isolata +3V3_ISO la quale viene generata dal regolatore di tensione U10.

L'alimentazione +3V3_ISO serve per alimentare l'HUB USB TUSB2046B della Texas Instruments in particolare i pin 3 e 25.

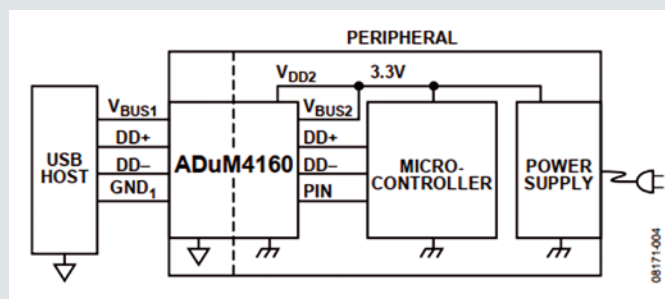


Fig. D
Tipica applicazione di isolamento porta USB UpStream.

troduce un ritardo di propagazione del segnale paragonabile agli hub USB standard o ai cavi stessi.

La Fig. D mostra la tipica applicazione di utilizzo dell'integrato ADuM4160 dove quest'ultimo viene integrato in una periferica la quale ha la necessità di interfacciarsi a un PC o altro device tramite porta USB.

I punti fondamentali sono:

- L'USB Host provvede a fornire l'alimentazione alla sezione UpStream dell'integrato ADuM4160 attraverso un cavo USB standard
- La sezione DownStream dell'integrato viene alimentata dalla periferica stessa la quale ha una sua alimentazione indipendente
- Le due linee differenziali D+ e D- del microcontrollore presente sulla periferica e connesse alla porta

DownStream dell'integrato vengono riportate sulla parte UpStream e connesse alle linee D+ e D- del cavo USB collegato al PC o altro device

- La velocità di funzionamento dell'interfaccia USB viene decisa in fase di progettazione tramite la configurazione di appositi pin dell'integrato e non può essere cambiata durante il funzionamento configurando dei registri o altro. Quindi si deve decidere se lavorare in modalità FullSpeed o LowSpeed e configurare entrambi i lati UpStream e DownStream a lavorare alla stessa velocità
- Le resistenze di pull-up e pull-down sono integrate nel silicio del dispositivo.

Di particolare interesse è il pin 8 (BUSPWR) che serve per indicare quale è la sorgente di alimentazione per le porte USB DownStream.

Se il pin è a livello logico alto indica all'integrato che l'alimentazione arriva dal cavo USB connesso alla porta UpStream, ma non è il nostro caso, viceversa indica all'integrato che si ha a disposizione una sorgente di alimentazione indipendente.

Il segnale su questo pin non può essere cambiato dinamicamente durante il funzionamento ma solo configurato al Power-Up. Noi abbiamo scelto di collegarlo a massa tramite una resistenza di pull-down da 10K (R13).

Il pin 26 (EXTMEM) serve per indicare se c'è collegata una memoria EEPROM all'interfaccia di

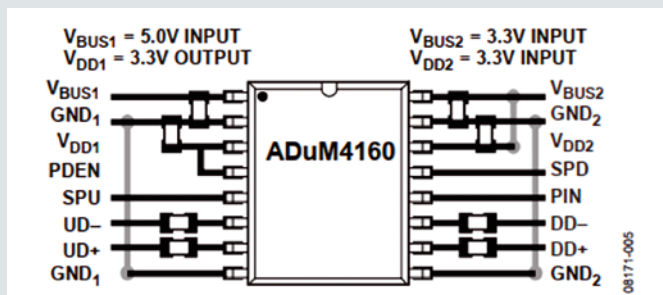


Fig. E
Layout PCB
raccomandato.

La Fig. E mostra il layout raccomandato per il PCB sul quale possiamo fare alcune osservazioni:

- In molte applicazioni USB la tensione di alimentazione +3V3 viene ricavata dalla tensione V_{BUS} a +5V tramite un regolatore di tensione LDO
- L'integrato ADuM4160 integra al suo interno un regolatore di tensione LDO sia lato UpStream che DownStream. Questo viene comodo per alimentare direttamente i driver delle linee D+ e D-. Tali alimentazioni sono rese disponibili ai pin V_{DD1} e V_{DD2} dell'integrato. Le alimentazioni a +5V devono essere portate ai pin V_{BUS1} e V_{BUS2} .

- Se lato DownStream è disponibile solo l'alimentazione a +3V3 questa può essere portata direttamente ai pin V_{BUS2} e V_{DD2} . Questo disabiliterà il regolatore LDO interno come visibile nella possibile configurazione mostrata in Fig. E. Nella nostra applicazione entrambi i pin V_{BUS1} e V_{BUS2} ricevono una alimentazione +5V. Il primo la ottiene dalla linea + V_{BUS} mentre il secondo dalla linea + V_{BUS_ISO} . Un regolatore di tensione LDO dedicato genererà i +3V3 necessari ad alimentare l'elettronica della scheda nella parte isolata (DownStream).

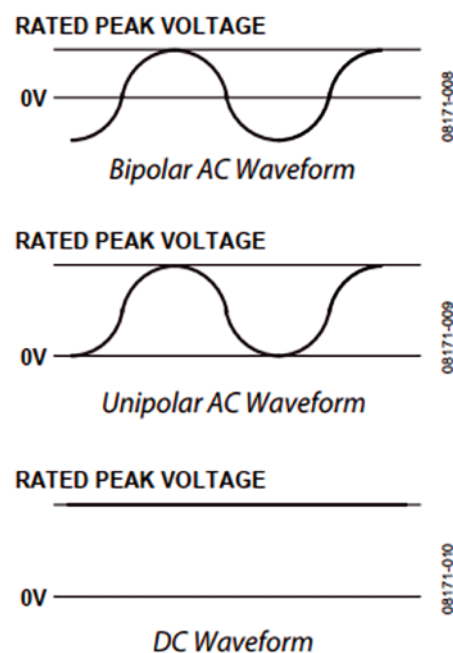
L'integrato, tra le sue caratteristiche, indica il tempo di vita della sezione di isolamento. Infatti tutte le strutture isolanti prima o poi tendono a deterio-

rarsi per via delle sollecitazioni ricevute nel tempo. La velocità di degradazione della sezione isolante dipende dalle caratteristiche della forma d'onda applicata ad essa.

La Fig. F mostra le tre tipologie di tensioni che si possono vedere applicate alla parte isolante dell'integrato. L'isolamento fatto con la tecnologia iCoupler degrada a velocità diverse a seconda della tipologia di forma d'onda

applicata. La forma d'onda più deleteria e degradante è quella bipolare mentre quelle unipolari o in corrente DC sono le meno impattanti. L'integrato ha una vita stimata media di 50 anni rispettando ovviamente le massime tensioni ammesse discusse prima. Se l'integrato è soggetto a ambienti di lavoro più gravosi di quelli massimi consentiti vedrà una riduzione della vita media della parte isolante.

Fig. F
Tipologia di
tensioni che
portano al
breakdown
della parte
isolante.



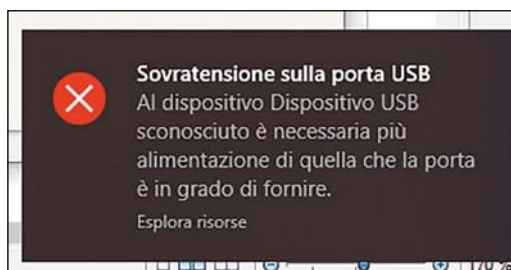
comunicazione. Se il pin è a livello logico alto, come nel nostro caso, significa che non c'è nessuna memoria collegata e quindi il pin 5 (EECLK) lo si può lasciare non connesso. In caso contrario bisognerà collegare una memoria EEPROM alle linee di comunicazione EECLK e EEDATA.

Se non c'è nessuna memoria EEPROM collegata allora il pin 6 (EEDATA/GANGED) serve per configurare la modalità di gestione delle sovracorrenti sulle porte DownStream. Quindi se connesso a massa il monitoraggio delle sovracorrenti avviene su singola porta in modo indipendente e questo significa che se c'è una sovracorrente sulla porta DownStream 1 solo quella porta sarà oggetto di intervento da parte dell'integrato. Più avanti fa-

remo un esempio esaustivo. Se invece il pin viene posto a livello logico alto si attiverà la modalità GANGED e quindi una sovracorrente su una singola porta porterà alla disabilitazione di tutte e quattro le porte DownStream. Noi abbiamo preferito la modalità a singola porta.

La gestione delle sovracorrenti richiede degli appositi integrati, nel nostro caso U6 e U7, i quali sono dei semplici "Current-Limited Power-Distribution Switches". Ognuno di essi gestisce due porte DownStream: porte 1 e 2 assegnate a U7 e porte 3 e 4 assegnate a U6. Questi integrati dialogano con l'HUB USB tramite le linee EN_USB_ISOx e OC_USB_ISOx dove la x indica il numero di porta da 1 a 4. Quindi l'HUB USB dopo essere stato alimentato

Fig. 3
Errore su porta USB.



e enumerato dal computer a cui è stato connesso porterà a livello logico basso le linee di abilitazione $\overline{EN_USB_ISOx}$ attivando così le porte DownStream e di conseguenza le loro alimentazioni. Infatti questi integrati sono degli switch analogici di alimentazione e di conseguenza la tensione $+V_{BUS_ISO}$ raggiungerà le porte USB DownStream solo quando le linee di abilitazione andranno a livello logico basso. L'attivazione delle porte DownStream viene evidenziata dai led di colore verde LD1, LD2, LD3 e LD4; se accesi vuol dire che sulle porte USB è presente l'alimentazione e quindi pronte all'uso. Per ogni porta DownStream è stato previsto l'uso di ferriti di filtro sulle alimentazioni e relativi TVS sulle linee dati.

Ora cosa succede se su una porta si ha una sovracorrente? Semplicemente l'integrato si accorge che c'è un assorbimento anomalo sopra soglia e quindi porta bassa la linea $\overline{OC_USB_ISOx}$ dove "OC" sta per "Over Current". L'HUB USB di conseguenza porterà la corrispondente linea $\overline{EN_USB_ISOx}$ a livello logico alto spegnendo l'alimentazione sulla porta interessata. Sul sistema operativo in uso comparirà un messaggio a indicare che c'è stato un problema sulla periferica collegata (Fig. 3).

La Fig. 4 mostra cosa succede a livello elettrico; il segnale in verde è la corrente assorbita dalla porta mentre il segnale in giallo è la tensione di alimentazione presente sulla porta in esame. Come si può osservare la corrente assorbita dalla porta passa da circa 200mA a un picco di oltre 3A, questo fa intervenire immediatamente la protezione togliendo alimentazione alla porta incriminata, infatti il segnale giallo crolla a 0V. Il led corrispondente si spegne e sul sistema operativo compare il messaggio descritto prima.

Gli integrati U6 e U7 garantiscono una corrente di 500mA come da specifica USB, tuttavia la soglia di intervento impostata di fabbrica ha un valore tipico di 1A e un valore massimo di 1,25A come indicato nel datasheet.

La Fig. 5 mostra una situazione diversa rispetto alla precedente ovvero un assorbimento anomalo di corrente ma non continuativo (picco a quasi 2A). In questo caso il sistema, come per il precedente caso, apre il circuito togliendo alimentazione. Tuttavia con cadenza di circa 500mSec tenta il riarmo della porta. In questo esempio la situazione viene recuperata al secondo tentativo e l'alimentazione sulla porta USB DownStream ritorna stabile con assorbimento 200mA. In questa situazione non viene visualizzato nessun messaggio da parte del sistema operativo.

Infine la Fig. 6 mostra una situazione di assorbimento di corrente progressivo da parte della porta DownStream. In questo caso si può vedere che la protezione interviene a circa 1A di corrente. Si notano anche i ritentativi di riarmare la porta senza successo. Il sistema ha tentato fino a un massimo di sei volte, dopodiché è stata segnalata una anomalia.

Fig. 4
Cortocircuito permanente sulla porta USB.

Fig. 5
Cortocircuito temporaneo sulla porta USB.





lia sulla porta da parte del sistema operativo. Quando il sistema operativo mostra il messaggio di errore l'unico modo per ripristinare la situazione di lavoro normale è staccare la periferica dalla porta DownStream incriminata e infine ri-enumerare l'HUB USB staccando e riattaccando il cavo μ USB connesso al computer.

Dopo questa breve digressione riprendiamo la trattazione dei pin dell'HUB USB e in particolare il pin 4 (RESET). Questo pin è attivo basso e serve per mantenere in condizione di reset l'integrato. Al power up questo pin deve rimanere basso per almeno 1mSec per dare il tempo alla alimentazione di stabilizzarsi e alle logiche interne di inicializzarsi. Per ottenere questo risultato si può realizzare un semplice circuito RC, nel nostro esempio abbiamo una resistenza da 15K e un condensatore da 1μ F, oppure usare un apposito integrato detto "Microprocessor Supervisor" il quale mantiene in reset l'integrato per un tempo prefissato al PowerUp. Quello che abbiamo predisposto sul circuito mantiene in reset l'integrato per un periodo di 20mSec. Integrato U11 predisposto ma non montato (MIC803-30D2VC3).

La **Fig. 7** mostra come agisce il circuito RC sul pin di RESET. In particolare abbiamo:

- Il segnale in giallo è l'alimentazione $+V_{BUS_ISO}$
- Il segnale in viola è il segnale presente sul pin di RESET
- Il segnale in azzurro è il clock

Si vede chiaramente che viene data alimentazione al sistema e il segnale sul pin di RESET sale lentamente, questo permette sia alla alimentazione di stabilizzarsi che all'HUB USB di inicializzare le sue logiche interne. Il clock diventa stabile dopo un

periodo superiore ai 100mSec.

Giusto per completezza si veda la **Fig. 8** la quale mostra l'andamento del segnale sul pin di RESET nel caso fosse montato il "Microprocessor Supervisor" al posto del circuito RC. Il concetto è lo stesso ma in questo caso si vede chiaramente che l'HUB viene mantenuto in reset per 20mSec al Power Up per poi essere rilasciato. Dopo circa 120mSec il clock diventa stabile e l'integrato pronto per lavorare.

Un quarzo da 6MHz deve essere collegato ai pin 30 (XTAL1) e 29 (XTAL2) affinché il PLL interno possa generare i 48MHz necessari alla periferica. Completano la configurazione del clock i pin 27 (TSTPLL/48MCLK) e 31 (TSTMODE) i quali devono essere connessi a massa.

Completiamo la descrizione con le quattro porte DownStream e relative linee di controllo degli

Fig. 6
Assorbimento crescente sulla porta USB fino a intervento protezione.

Fig. 7
Reset con circuito RC.

Fig. 8
Reset con integrato MIC803-30D2VC3.





Fig. 9
Assieme
scheda
RaspberryPi
3B+ e scheda
IsoHubUsb.

switch di alimentazione sopra descritti. Ogni coppia D+ e D- delle porte DownStream prevede due resistenze serie da 22R e due condensatori da 22pF da posizionare il più vicino possibile ai pin dell'integrato. Infine lato connettore USB Type-A si devono montare delle resistenze di pull-down. Anche queste devono essere posizionate il più vicino possibile ai pin dell'integrato. Con questo possiamo ritenere conclusa la trattazione dello schema elettrico.

IL LAYOUT DEL PCB

Alcune considerazioni sul layout della scheda; le protezioni ESD e EMI implementate devono essere posizionate il più vicino possibile ai connettori USB. Il quarzo da 6MHz deve essere posizionato il più vicino possibile all'integrato TUSB2046B mentre eventuali integrati di alimentazione devono essere posti il più lontano possibile dall'integrato TUSB2046B e dalle linee differenziali D+ e D-. Le linee differenziali devono essere disegnate in modo che la loro impedenza differenziale sia di $90R \pm 10\%$ e, se possibile, sul Top Layer con un piano di massa continuativo che corre sotto di esse; si consiglia quindi di usare PCB da 4 Layer o più. Ovviamente questo dipende dalla vostra applicazione. I piani di massa sotto le linee differenziali non devono mai essere spezzati e le linee differenziali non devono mai avere angoli a 90 gradi nei cambi direzione. L'angolo da utilizzare per un cambio di direzione deve essere maggiore o uguale a 135 gradi.

La lunghezza delle piste non deve superare i 4000mils tipici fino a un massimo di 12000mils e le due linee differenziali devono essere di pari lunghezza con una tolleranza massima di 50mils. Si devono evitare gli stubs e il numero massimo di vias consentiti è pari a 4. Ovviamente se si cambia layer, ad esempio si passa dal Top Layer al Bottom Layer ci si deve assicurare che il layer interno più prossimo abbia un piano di massa continuo che corra sotto le piste differenziali. La scheda è stata disegnata per essere sia col-

legata a una porta USB del computer (Desktop o Notebook) oppure a una porta USB di una qualsiasi RaspberryPi. Se si utilizza una RaspberryPi come Host nessuno dei segnali presenti sul connettore da 40 pin è stato utilizzato. La scheda è stata sagomata affinché il connettore da 40 poli sia libero da qualsiasi impedimento. La **Fig. 9** evidenzia quanto appena detto e mostra l'assieme della scheda IsoHubUsb Rev.0.2 e una scheda RaspberryPi 3B+.

Il piano di montaggio mostra il layout del PCB, mantenendo le 4 porte USB DownStream Type-A in basso e quindi il connettore 40 poli RaspberryPi a sinistra.

In alto a sinistra trovate il connettore μ USB (CN1) per la connessione alla USB del computer e relati-

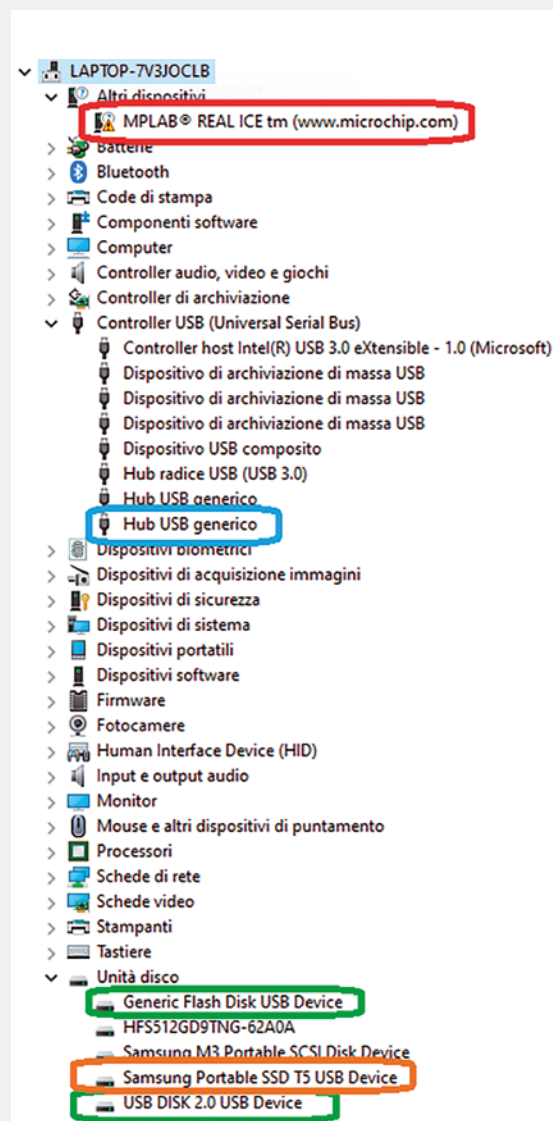


Fig. 10
Periferiche
collegate al nostro
HUB USB isolato.

vo led blue LD5 (sul PCB è presente in serigrafia la scritta HOST). Invece sulla destra c'è il connettore µUSB per l'alimentazione ausiliaria lato periferiche (CN4) e il relativo led rosso LD6 (sul PCB è presente in serigrafia la scritta AUX PW).

Per quanto riguarda le porte DownStream USB Type-A queste sono raggruppate a due a due; in particolare il connettore CN3 identifica le porte USB1 e USB2 mentre il connettore CN2 le porte USB3 e USB4. Per ogni porta c'è il relativo led verde (LD1 abbinato a USB1, LD2 abbinato a USB2 ecc).

Inoltre lato periferiche è ben evidente l'HUB USB Texas Instruments (U1) e i due integrati per la gestione delle alimentazioni delle porte DownStream U6 e U7.

COLLEGAMENTO A PORTA USB NOTEBOOK

Abbiamo collegato la porta µUSB CN1 alla porta USB 3.0 Type-A di un notebook e sfruttando un alimentatore esterno abbiamo alimentato la scheda lato periferiche collegandoci al connettore µUSB CN4. Il nostro dispositivo viene riconosciuto da Windows 10 senza problemi e identificato come "Hub USB Generico".

La **Fig. 10** mostra le periferiche che abbiamo connesso al nostro HUB USB isolato.

Di seguito l'elenco:

1. Abbiamo collegato in prima battuta un Real ICE della Microchip per lo sviluppo firmware. Questo compare nell'elenco visibile in figura. Cerchiato in rosso
2. Abbiamo collegato un altro Hub USB di terze parti. Perfettamente riconosciuto dal sistema operativo. In figura è cerchiato in azzurro
3. All'Hub di terze parti abbiamo collegato due chiavette USB. Anch'esse riconosciute dal sistema operativo. Nella figura sono cerchiati in verde
4. Infine abbiamo collegato un SSD con interfaccia USB R.3.1 a un'altra porta libera del nostro HUB USB e anche questa periferica è visibile dal sistema operativo. In figura è cerchiata in arancione. Su questo disco è presente una partizione cifrata ed è risultata perfettamente funzionante. Ovviamente si sentono i limiti di una connessione USB 2.0 Full Speed

CONCLUSIONI

A questo punto possiamo ritenere conclusa la trattazione del nostro HUB USB isolato.

Le possibili applicazioni per questo tipo di architettura sono molteplici, tra cui il collegamento

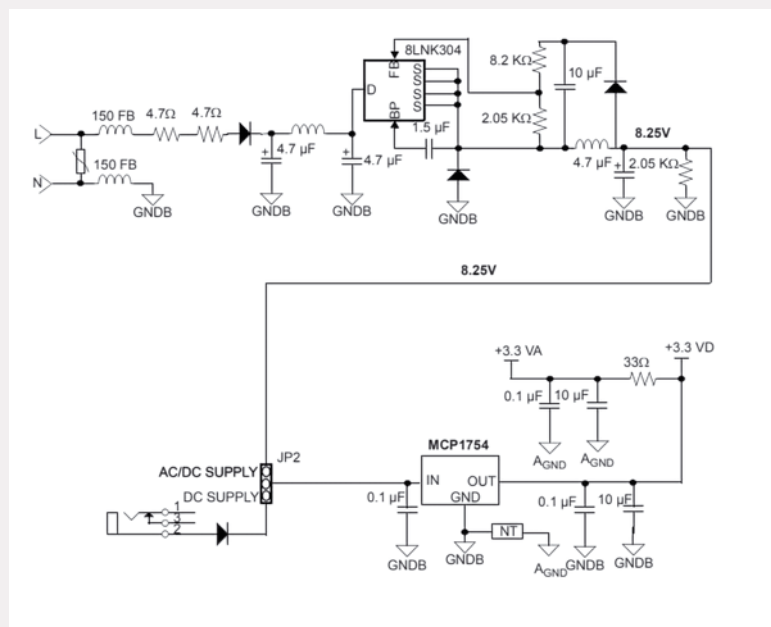


Fig. 11
Alimentazione da rete 230VAC

di periferiche di sviluppo come Real-Ice, PicKit; schede per movimentazione relay di potenza; schede per la lettura di ingressi digitali o analogici ecc. Nel caso in cui si dovessero avere applicazioni con schede elettroniche alimentate dalla rete elettrica 230V senza trasformatore di isolamento, come visibile in **Fig. 11**, si consiglia di realizzare un hardware con solo il transceiver ADuM4160 da collegare alle linee differenziali D+ e D- di un micro-controllore così da avere un collegamento 1:1 tra il computer e la periferica senza un HUB interposto. In questo modo si evita di danneggiare eventuali periferiche collegate alle altre porte DownStream Type-A connesse all'HUB USB. □



Cosa occorre?

I componenti utilizzati in questo progetto sono di facile reperibilità. Le board Raspberry Pi sono disponibili in diverse versioni all'indirizzo bit.ly/rpiboards. Il PCB può essere realizzato con il servizio PCBPRODUCTION all'indirizzo www.futurashop.it/produzionePCB. I prezzi si intendono IVA compresa.

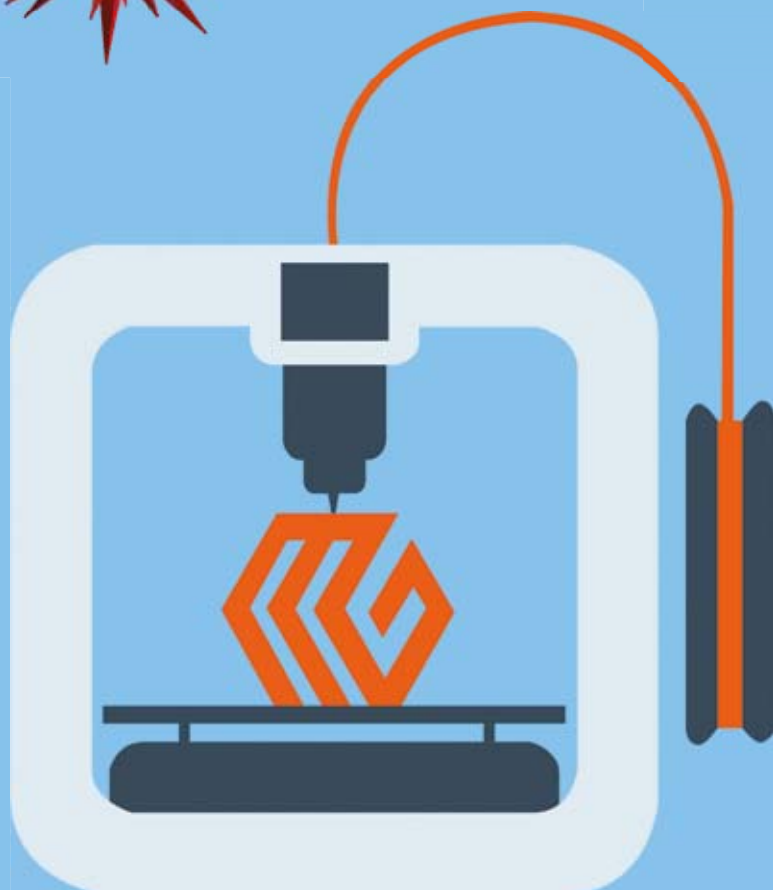
Il materiale va richiesto a:

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

Trasformiamo le tue idee in realtà!

Servizio taglio laser e incisione

Futura Elettronica ti offre il servizio di taglio laser ed incisione per realizzare pannelli, parti meccaniche, mascherine, scritte, decori ed altro ancora in plexiglass (o metacrilato).



Servizio stampa 3D

Futura Elettronica mette inoltre a disposizione le proprie stampanti 3D FDM, per chi vuole realizzare un oggetto tridimensionale con dimensioni fino a 40x40x40 cm.

Informazioni dettagliate su
www.futurashop.it

HI-TECH

CONTROLLER UNIVERSALE PER CNC

del Prof. **MICHELE MENNITI**



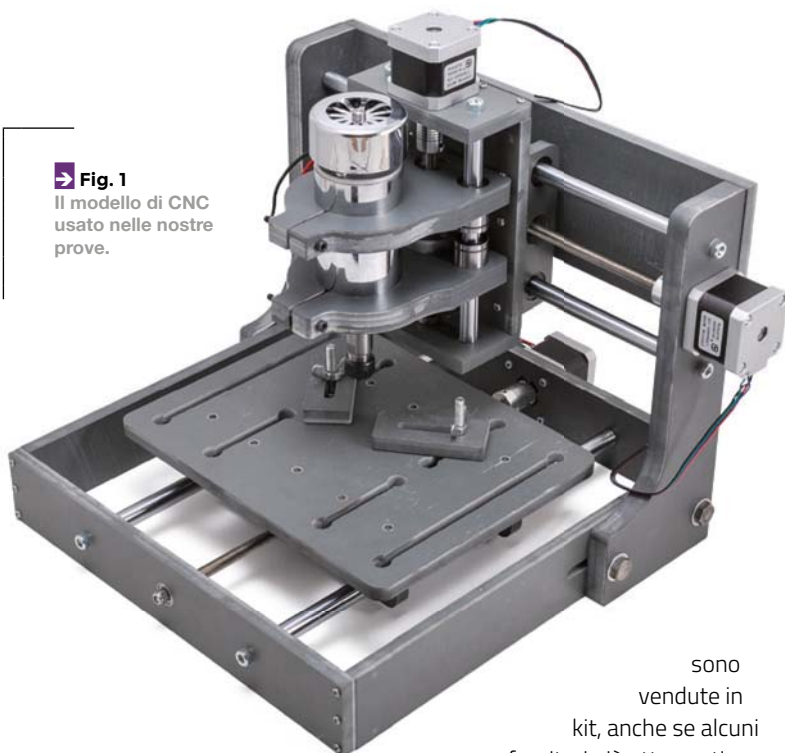
Presentiamo una scheda, basata su un Arduino Nano, completa di tutte le funzionalità hardware e software per pilotare macchine CNC a tre assi.

N

egli ultimi anni la riduzione dei costi delle parti meccaniche e la progettazione di piccole macchine CNC (Computer Numerical Control) dedicate espressamente alla realizzazione artigianale di PCB, hanno dato il via ad una proliferazione di progetti riguardanti la meccanica da un lato e l'elettronica di controllo dall'altro.

Fondamentalmente, una CNC atta a produrre circuiti stampati, del tipo a tre assi, è costituita da: struttura di supporto, realizzata in genere in alluminio o in PVC, un piatto per il fissaggio del PCB, una serie di barre cilindriche in acciaio cromato, tre motori passo-passo, un elettromandrino (spindle), una scheda elettronica per il controllo dei tre motori, un alimentatore idoneo e tutti i necessari accessori per l'assemblaggio della macchina. Quasi tutte le CNC progettate per uso hobbistico

Fig. 1
Il modello di CNC
usato nelle nostre
prove.



sono
vendute in
kit, anche se alcuni
fornitori più attrezzati pro-
pongono in alternativa la versione
già montata.

Come sempre, in questi casi, si può trovare materiale di eccellente qualità, con prestazioni considerevoli, ma anche prodotti che spesso non valgono nemmeno il poco denaro richiesto per il loro acquisto. Ma se la componente meccanica è facilmente valutabile anche solo guardando le immagini del prodotto e leggendo le specifiche tecniche dei motori e, soprattutto, dello spindle, quando si tratta della scheda elettronica di controllo, le cose cambiano molto, in quanto bisogna conoscere il tipo e versione di firmware adottato e, di conseguenza, tutte le opzioni disponibili, in aggiunta al solo pilotaggio dei tre motori passo-passo e dello spindle.

Vi anticipiamo subito che, nel nostro caso, abbiamo usato una meccanica CNC, in grado di fare lavorazioni da 200x180x60 mm, reperibile sul sito www.futurashop.it con il codice CNC2018 (**Fig. 1**), mentre per la scheda elettronica che vi presentiamo in questo articolo, abbiamo adottato il firmware open source GRBL v1.1.h, ultima versione disponibile nel momento in cui scriviamo, scaricabile dalla pagina <https://github.com/gnea/grbl/releases>. In queste pagine realizzeremo il controller per CNC e vi insegneremo a programmarlo in modo ottimale per la meccanica che sceglierete di adottare, usando come esempio la CNC2018 che rimane la scelta ottimale per realizzare PCB.

DESCRIZIONE GENERALE

Il progetto della nostra scheda è basato proprio sulla versione GRBL 1.1, della quale abbiamo sfruttato ogni caratteristica, portando all'esterno tutti i vari ingressi ed uscite del microcontrollore, che andremo poi ad illustrare. Abbiamo inoltre previsto la possibilità di utilizzare un accessorio che si reperisce facilmente in commercio, un controller offline, dotato di un proprio microcontrollore, un piccolo display LCD ed alcuni pulsanti, che permette di caricare il file da fresare in una micro SD card e di gestire le principali fasi dell'incisione mantenendo il PC scollegato, una cosa molto utile in alcune situazioni.

Infine, per permettere una agevole riparazione, in caso di guasti, abbiamo fatto ricorso a due moduli esterni, per il microcontrollore e per l'alimentazione generale, invece di prevedere tutta la componentistica sul PCB principale.

MODULI INTEGRATI

Il cuore del nostro controller è costituito da una board Arduino NANO v.3, che va innestata su una coppia di HEADER presenti sul PCB principale. Arduino si occupa di gestire, tramite il firmware al quale abbiamo accennato, tutti i segnali in entrata ed in uscita. I pin I/O della board sono interfacciati con i tre driver per i motori passo-passo, il connettore per il controller offline, il connettore per i controlli esterni ed il circuito di comando dello spindle.

Vediamo una panoramica dei collegamenti dei pin:

- D2+D7: gestiscono i comandi STEP e DIR dei driver dei tre motori degli assi X, Y e Z;
- TX1 e RX0: sono i pin utilizzati per il collegamento seriale con il controller offline;
- RESET: è il reset del micro, controllabile sia dal controller offline che mediante il connettore esterno;
- D8: gestisce l'ENABLE dei driver dei tre motori degli assi X, Y e Z;
- D9, D10 e D12: vanno al connettore esterno e servono per controllare il comportamento degli switch di finecorsa opzionali dei tre assi X, Y e Z;
- D11: gestisce il segnale PWM per il controllo della velocità dello spindle;
- D13: va al connettore esterno e serve per cambiare la direzione di rotazione dello spindle;
- A0: va al connettore esterno e serve per dare il comando di ABORT (stop) alla macchina;
- A1: va al connettore esterno e serve per dare il comando di FEED-HOLD (pausa lavoro) alla macchina;

- A2: va al connettore esterno e serve per dare il comando di CYCLE-START-RESUME (ripresa lavoro) alla macchina;
- A3: va al connettore esterno e serve per pilotare una ventola esterna quando viene fornito il comando di COOLER ENABLE (abilitazione raffreddamento);
- A5: va al connettore esterno e serve per ricevere il segnale della sonda (PROBE) di controllo della posizione dello spindle;
- A4, A6 e A7: non sono utilizzati, ma sono portati sul connettore esterno, come predisposizione per usi futuri.

L'altro modulo che abbiamo utilizzato è quello di alimentazione, uno switching di tipo step-down da 5 A, il cui costo è decisamente inferiore a quello dei singoli componenti, se li avessimo previsti direttamente sul PCB.

Iniziamo la descrizione dello schema elettrico proprio da quest'ultimo. Intanto diciamo subito che il modulo sarà necessario solo se lo spindle della vostra CNC dovrà lavorare a tensioni superiori a 12 Vdc, in caso contrario non servirà e si potrà alimentare il controller direttamente con un alimentatore esterno da 12 V, in grado di erogare una corrente sufficiente per lo spindle e per i tre motori. In questo caso occorrerà collegare tra loro, mediante un ponte, i fori E1 ed E4 presenti sul PCB, usando uno spezzone di filo di diametro adeguato per lo scorrimento di una corrente di circa 5 A. Nel nostro caso lo spindle (cod. ELEMANDER11, reperibile sul sito www.futurashop.it) richiede una tensione di 36 V, che otterremo da un alimentatore esterno, come il modello MW35036, che ha 350 W di potenza. Questa tensione, applicata all'ingresso POWER del controller, provoca l'accensione del LED rosso LD1, ed è applicata al circuito di alimentazione dello spindle, costituito dal mosfet Q1, con i suoi resistori di polarizzazione, alcuni condensatori, due diodi schottky e due induttori di potenza; il mosfet, un IRLZ44NSPBF in formato D2PACK, viene pilotato dal segnale SPINDLE PWM (pin D1 di Arduino NANO), in modo da poter regolare agevolmente la velocità di rotazione dello spindle, senza perdita di potenza. Quando il mosfet va in conduzione si accende il LED blu LD4.

Il modulo step-down, prima di essere saldato sul PCB, dovrà essere alimentato alla tensione richiesta dal vostro spindle e tarato, tramite il trimmer di precisione, per avere sull'uscita una tensione di 12 V. Tale tensione fa accendere il LED arancione LD2 sul PCB principale, e serve per alimentare i tre mo-

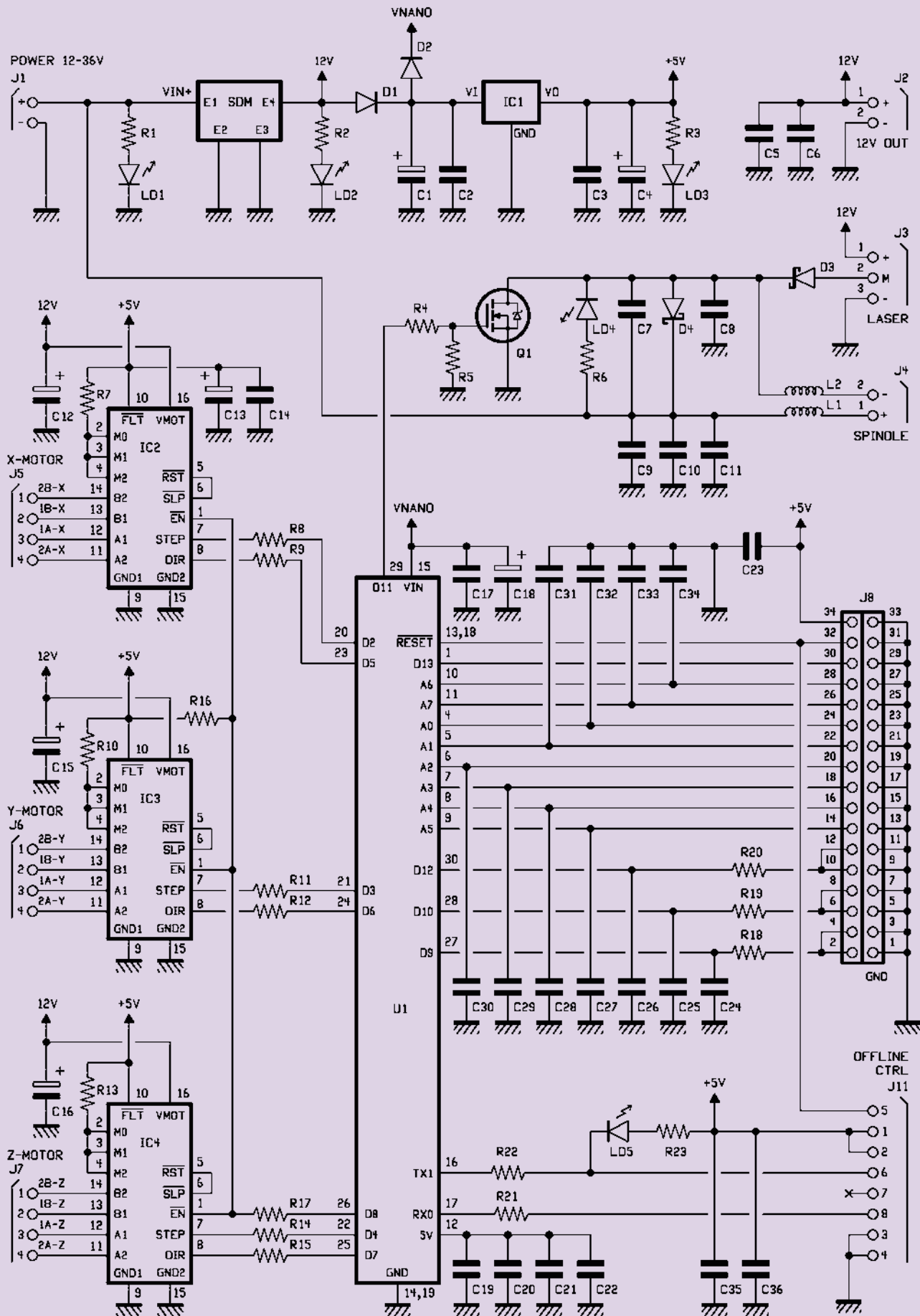
tori passo-passo degli assi X, Y e Z, mediante i driver DRV8825; essa è inoltre presente sul connettore 12V_OUT come tensione di servizio che, nel nostro caso, sarà usata per alimentare una ventola di raffreddamento per i tre driver e, se necessario, per lo stesso modulo step-down. Questa tensione viene poi applicata al circuito regolatore lineare costituito da IC1, un 7805 in formato TO252, e da alcuni condensatori con funzione di filtro (C1 e C4) e anti-autooscillante (C2 e C3) che ci permette di avere una tensione stabile e ben filtrata di 5 V, la cui presenza è indicata dall'accensione del LED verde LD3. Tale tensione serve per alimentare i tre driver DRV8825 ed è anche presente sul connettore esterno e su quello del controller offline, per alimentarne il circuito elettronico. L'uscita del modulo switching non viene applicata direttamente al 7805, in serie abbiamo previsto un diodo schottky 1N5819 (D1) al fine di evitare ritorni di tensione dal condensatore C1, che potrebbero danneggiare il modulo switching; inoltre la caduta di tensione di circa 0,4 V diminuisce leggermente la dissipazione di potenza del 7805, dovuta al drop-out (differenza di tensione tra ingresso ed uscita del regolatore) di 7 V, ridotto in tal modo a 6,6 V. Una funzione simile è svolta dal diodo D2, che porta l'alimentazione al pin VIN di Arduino NANO, alimentando la board; i condensatori C17 e C18 svolgono le stesse funzioni di quelli usati a monte del 7805. La tensione di 12 V applicata all'Arduino NANO, viene ridotta al suo interno a 5 V, necessari per alimentare il microcontrollore ATmega328P-AU, e viene portata sul pin 12 (da noi denominato 5V_2) della stessa board. Per evitare che eventuali disturbi, provocati dal modulo switching, dai motori passo-passo o dallo spindle, possano provocare malfunzionamenti all'Arduino NANO, a questo pin abbiamo collegato un filtro antidisturbo, costituito dai quattro condensatori C19+C22.

Al pin 6 del connettore J11, destinato al controller offline, abbiamo collegato il LED giallo LD5, che si accenderà ad ogni comando impartito tramite i pulsanti del controller oppure tramite il programma di gestione della CNC (Candle), in pratica è un visualizzatore dell'attività della comunicazione seriale con Arduino NANO.

Approfondiremo l'uso di questo accessorio opzionale alla fine di questo articolo.

La ricca dotazione di LED, ben cinque, è fondamentale per tenere costantemente sotto controllo tutte le tensioni coinvolte nel funzionamento del nostro controller e delle periferiche esterne ad esso collegate; per comodità riportiamo l'elenco





completo nella **Tabella 1**.

Per il nostro controller abbiamo utilizzato tre driver di tipo DRV8825 prodotti dalla POLOLU (**Fig. 2**). Per pilotarli correttamente servono sette pin del micro da collegare ai segnali ENABLE (D8), STEP (D2+D4) e DIR (D5+D7); i primi (pin 1) sono tutti collegati al pin D8 di Arduino NANO, tramite il resistore R17 in serie, e sono mantenuti a livello HIGH da R16, un resistore con funzione di pull-up (collegato all'alimentazione del driver); gli STEP (pin 7) sono collegati ai pin D2, D3 e D4, mentre i DIR (pin 8) sono collegati ai pin D5, D6 e D7. I pin RESET (5) e SLEEP (6) di ogni driver sono cortocircuitati tra loro. I pin M0, M1 e M2 (2,3,4) servono per settare la risoluzione degli step, in base alla tabella della **Tabella 2**.

In pratica la condizione di M0-M1-M2 in LOW rappresenta la minima risoluzione, in quanto ad ogni impulso corrisponde uno step completo del motore; nel nostro caso, poiché serve la massima precisione possibile, abbiamo settato M0-M1-M2 su HIGH, mediante i resistori di pull-up R7, R10 e R13; in tal modo ad ogni impulso corrisponderà una rotazione del motore di 1/32mo di passo. I pin VMOT (12) vanno collegati ai 12 V, per l'alimentazione del motore; i condensatori C12, C15 e C16 da 100 µF, sono fortemente consigliati dal datasheet del driver, per la protezione contro eventuali spike dovuti al motore. I pin FLOAT (10) vanno collegati ai 5 V, con la coppia di condensatori di filtro C13 e C14 verso massa; i pin GND1 e GND2 vanno collegati alla massa generale del circuito. I pin 11 (2A), 12 (1A), 13 (1B) e 14 (2B) di ogni driver, collegati ai connettori J7, J8 e J9, sono destinati agli avvolgimenti del motore, vedremo poi come realizzare i tre cavetti. Il piccolo trimmer SMD presente sulla scheda del driver serve per la taratura in corrente, vedremo questa cosa nella sezione della realizzazione pratica.

Il connettore esterno è costituito da una barra strip a doppia fila a 90°, con 17+17 contatti; una delle due file ha tutti i pin collegati a massa, in quanto i corrispondenti pin dell'altra fila, per svolgere la loro funzione, vanno appunto collegati a massa, mediante opportuno contatto. Per esempio, il pin PROBE, presente sul pin 14, viene riconosciuto dall'Arduino come segnale, nel momento in cui viene cortocircuitato a massa, mentre il segnale COOLER ENABLE, che è un segnale in uscita, quando viene attivato dall'Arduino passa da LOW ad HIGH, sempre rispetto a massa. I pin del connettore esterno collegati ai pin analogici

ELENCO LED		
Sigla	Colore	Funzione
LD1	ROSSO	ALIM. ESTERNO
LD2	ARANCIONE	12 V
LD3	VERDE	5 V
LD4	BLU	SPINDLE
LD5	GIALLO	Controller OFF-LINE

Tabella 1
Tabella dei LED.

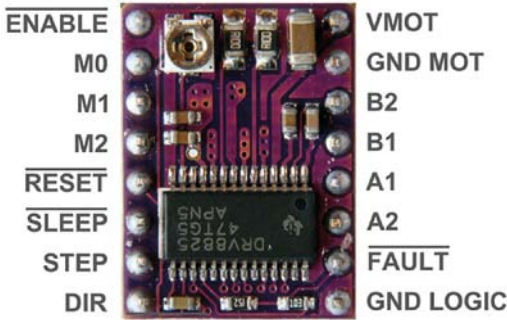


Fig. 2
Il driver POLOLU DRV8825.

ci A0+A7, hanno un condensatore da 100 nF verso massa, con funzione di filtro antidisturbo; i tre pin destinati agli switch di finecorsa sono dotati dei gruppi RC R18-C24, R19-C25 e R20-C26, che svolgono la funzione anti-debounce, per evitare false segnalazioni di finecorsa.

IL MONTAGGIO DEI COMPONENTI

Nello specifico box è visibile l'elenco dei componenti necessari per realizzare il nostro circuito. Come si vede, i componenti elettronici sono tutti SMD, quindi la realizzazione del controller richiede conoscenza di questi componenti e idonea manualità nelle saldature. Per il montaggio dei componenti del nostro controller abbiamo predisposto un circuito stampato a doppia faccia, per la cui

Tabella 2
Tabella dei settaggi del DRV8825.

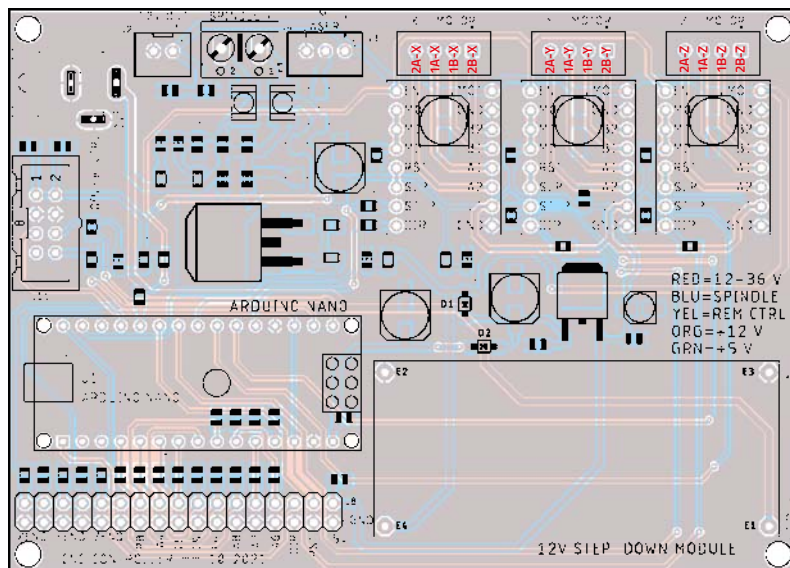
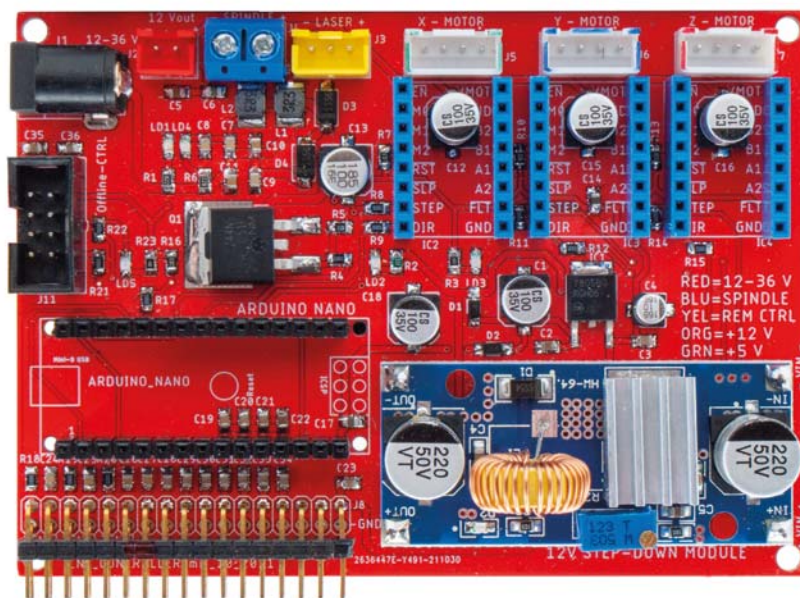
MODE0	MODE1	MODE2	MICROSTEP RESOLUTION
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	1/4 step
High	High	Low	1/8 step
Low	Low	High	1/16 step
High	Low	High	1/32 step
Low	High	High	1/32 step
High	High	High	1/32 step

Elenco Componenti:

R1, R6: 8,2 kohm (0805)
 R2: 2,2 kohm (0805)
 R3, R23: 1 kohm (0805)
 R4: 10 ohm (0805)
 R5, R7, R10, R13, R16: 10 kohm (0805):
 R8, R9, R11, R12, R14, R15, R17, R18, R19, R20,
 R21, R22: 100 ohm (0805):
 C1, C12, C15, C16, C18: 100 μ F 25 V
 elettrolitico ($\varnothing$ 6,3 mm)
 C13: 100 μ F 16 V elettrolitico ($\varnothing$ 6,3 mm)
 C4: 10 μ F 16 V elettrolitico ($\varnothing$ 4 mm)
 C6, C14, C17: 10 μ F ceramico (0805)
 C5, C10, C11, C22: 1 μ F ceramico (0805)
 C2, C3, C7, C8, C9, C19, C20, C21, C23, C24,
 C25, C26, C27, C28, C29, C30, C31, C32, C33,
 C34, C35, C36 : 100 nF ceramico (0805)
 D1, D2: BAT42W-7-F
 D3, D4: SS56B
 LD1: LED rosso (0805)
 LD2: LED arancione (0805)
 LD3: LED verde (0805)
 LD4: LED blu (0805)
 LD5: LED giallo (0805)
 Q1: IRLZ44NSTRPBF
 L1, L2: 3,3 μ H / 4,4 A
 IC1: LM78M05CDT/NOPB
 IC2, IC3, IC4: DRV8825 Pololu
 U1: Arduino Nano V30
 SDM: modulo switching step-down 5 A
 Vin 8÷36V Vout 1,25÷32V
 J1: plug DC 2,1 mm
 J2: JST 2 poli p. 2,54 mm
 J3: JST 3 poli p. 2,54 mm
 J4: morsetto 2 poli p. 5 mm
 J5, J6, J7: JST 4 poli p. 2,54 mm
 J8: strip maschio 2x17 - 90° p. 2,54 mm
 J9: connettore IDC maschio 2x4

Varie:

- N. 6 pin strip femmina 8 vie p. 2,54 h. 8 mm
- N. 2 pin strip femmina 15 vie p. 2,54 h. 8 mm
- circuito stampato S1593 (104,5 x 75 mm)



realizzazione troverete i file gerber, nella sezione download del sito www.elettronica.in. Seguendo il piano di montaggio, come sempre conviene iniziare dai componenti più piccoli e più bassi: resistori e condensatori non polarizzati, diodi e LED, induttori, regolatore 7805 e mosfet; quindi si passa ai condensatori elettrolitici, ai connettori ed agli header, facendo attenzione che siano posizionati in modo perfettamente verticale rispetto al PCB, altrimenti poi ci saranno difficoltà di inserimento dei tre driver e di Arduino NANO. Un buon sistema per montare

correttamente gli header consiste nell'innestarli nei pin del modulo che poi andranno ad ospitare, e poi inserirli nel PCB e saldarli, in tal modo si avrà la certezza della perfetta corrispondenza con il modulo. In ultimo occorre fissare il modulo switching step-down, posizionandolo a contatto del PCB e facendo combaciare i quattro fori, nei quali si dovranno inserire quattro spezzoni di filo rigido, saldandoli sia sulla faccia superiore del modulo che su quella inferiore del PCB; per fare una cosa più elegante si possono saldare nei 4 fori del PCB dei

pin strip singoli, e poi inserire il modulo su di essi e saldarli sulla faccia superiore dello stesso. In **Fig. 3** potete vedere il nostro prototipo inserito in un contenitore stampato in 3D.

Due note importantissime: il modulo va alimentato e tarato prima di inserirlo sul PCB, ruotando il perno del trimmer di precisione, per ottenere sui pin OUT+ e OUT- una tensione di 12 V, altrimenti si rischia di danneggiare qualche componente del circuito; inoltre occorre fare attenzione al verso del modulo, prima di saldarlo, bisogna far combaciare i suoi ingressi IN+ e IN- rispettivamente con i due fori E1 ed E3, serigrafati VIN+ e VIN-.

I COLLEGAMENTI

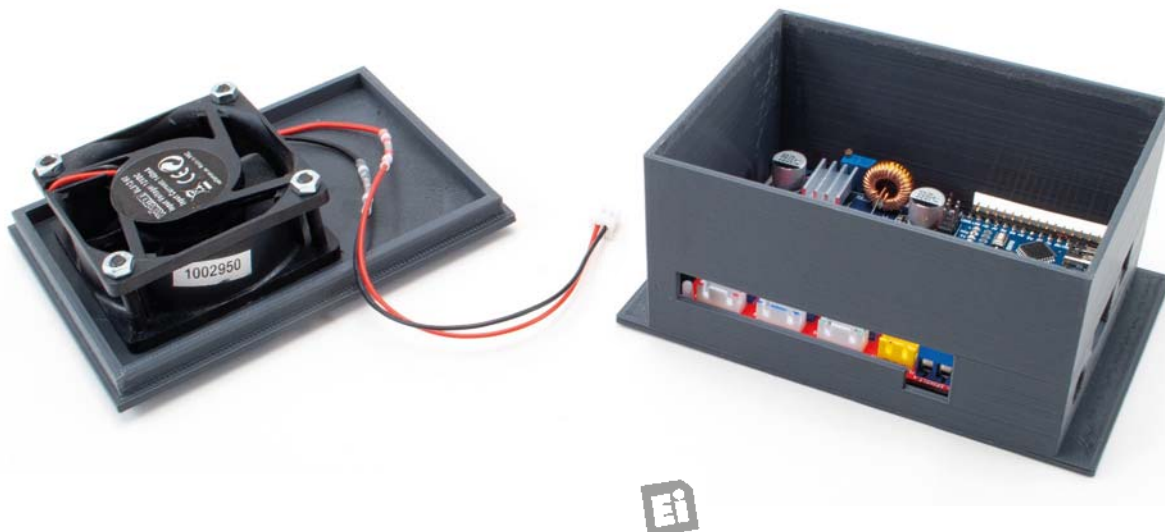
Per collegare il nostro controller alle varie periferiche esterne potete fare riferimento allo schema di cablaggio nella pagina successiva.

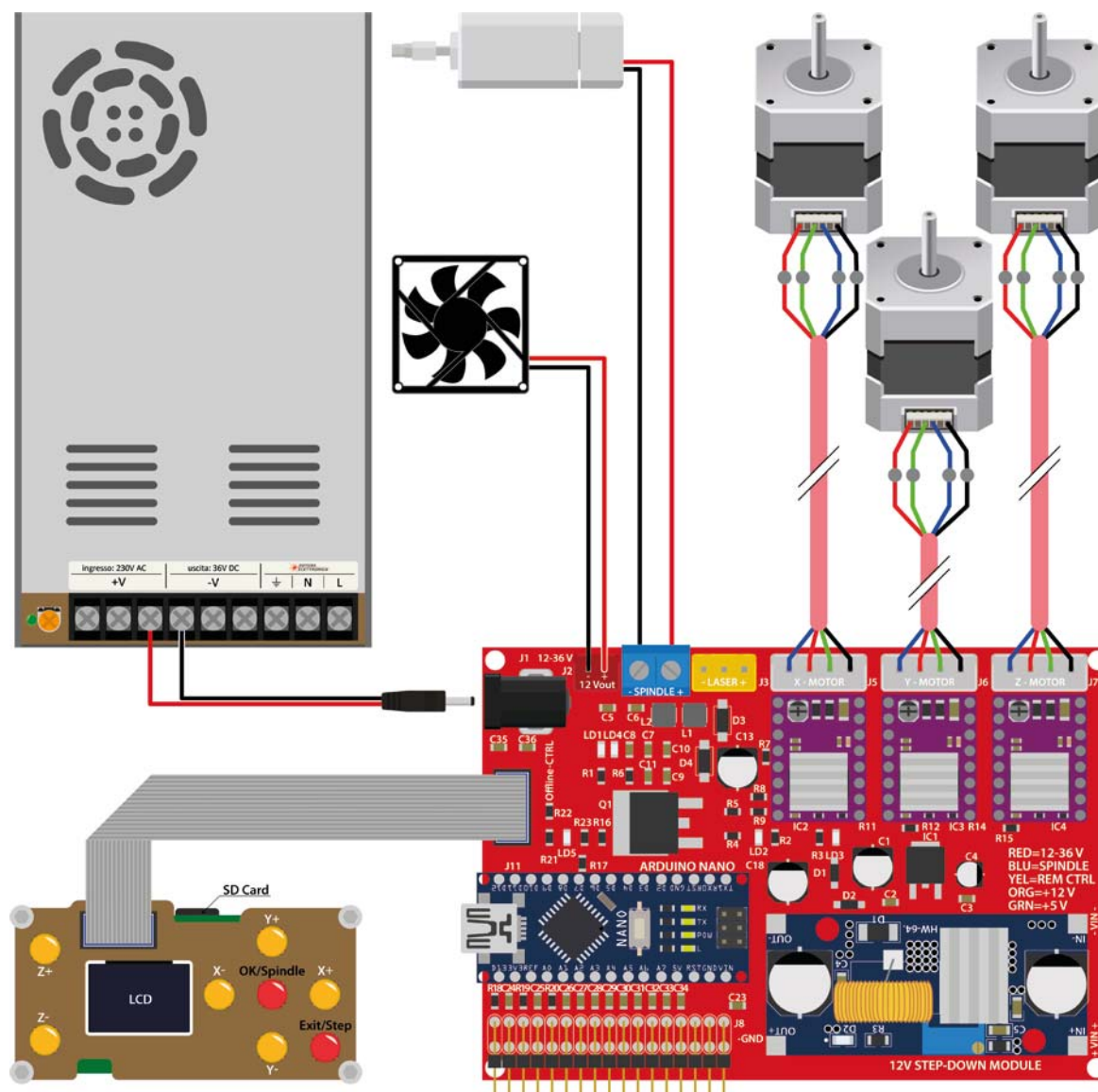
Affinché i motori passo-passo funzionino correttamente, devono essere collegati ai rispettivi connettori mediante dei cavetti che, nel nostro caso, hanno uno spinotto JST a 4 pin passo 2,54 mm (lato controller) ed uno spinotto JST a 6 pin passo 2 mm (lato motore).

Le connessioni vanno realizzate in base alla configurazione degli avvolgimenti, secondo lo schema riportato in **Fig. 4**. Nella parte superiore dell'immagine sono riportate l'immagine del driver e la rappresentazione degli avvolgimenti del motore passo-passo. Le immagini ed i colori vi permetteranno di realizzare facilmente i tre cavetti o anche modificare quelli in dotazione, se necessario. Bisogna fare molta attenzione ad osservare l'orientamento dei due spinotti, seguendo le linguette per il loro innesto nelle prese del PCB e dei motori, comunque la foto nella sezione inferiore dell'immagine sarà sicuramente di aiuto. Lo schema di collegamento è standard e vale per qualsiasi tipo di motore, ma la posizione delle connessioni po-

trebbe risultare differente, se userete un modello diverso da quello in dotazione alla CNC usata da noi. Allo spinotto PLUG DC va applicata la tensione di lavoro dello spindle, nel nostro caso 36 V, utilizzando un alimentatore esterno in grado di erogare la necessaria corrente, che servirà sia per lo spindle che per i tre motori passo-passo; questi ultimi saranno alimentati dal modulo step-down, ma anche la loro corrente sarà erogata all'origine dall'alimentatore esterno. Dovrete realizzare un cavetto a due fili, con sezione idonea alle correnti in gioco, e poi saldare il V+ al pin centrale dello spinotto PLUG, ed il V- a quello esterno. L'unico connettore a vite presente serve per l'alimentazione dello spindle, anche in questo caso servono due fili di sezione idonea e bisogna rispettare la polarità ma, in caso di errore, non succederà niente di particolare, solo che l'asse dello spindle girerà in senso antiorario, a quel punto basterà invertire i fili tra loro. Il connettore dei 12 V andrà utilizzato per collegare una ventola di dimensioni adeguate (per esempio il modello BLS12/60, reperibile sul sito www.futurashop.it) da posizionare sopra la scheda, in modo da raffreddare costantemente sia il modulo step-down che i tre driver DRV8825, sui quali andranno applicati i piccoli dissipatori adesivi di alluminio, che in genere sono in dotazione. Anche se le specifiche POLOLU, nel caso dei motori usati da noi (1,5 A) non richiederebbero nemmeno i dissipatori, consigliamo di adottarli comunque e di aggiungere la ventola. Il contenitore, da noi disegnato espressamente per questo controller, è predisposto per il montaggio di questo accessorio; nella sezione download troverete il file "sti" per la stampa 3D del contenitore. Se decidete di adottare il controller offline, dovreste semplicemente inserire lo spinotto IDC nella relativa presa sul PCB, il verso è obbligato e le connessioni sono standard. Vedremo più avanti come si utilizza.

Fig. 3
Il prototipo inserito all'interno del contenitore stampato in 3D.





TARATURA DEI DRIVER

A questo punto bisogna effettuare la taratura dei tre driver DRV8825 per il corretto funzionamento dei motori. Questa operazione si può eseguire anche senza la presenza di Arduino NANO; il circuito deve essere alimentato, come specificato prima, ed i motori devono essere collegati ad esso tramite i rispettivi cavetti. Bisogna settare la VREF dei driver in base alla corrente assorbita dal motore usato; nel nostro caso, il modello con codice STEPMOT03, reperibile su www.futurashop.it, è di 1,5 A; la tensione di riferimento si calcola mediante la semplice formula: $VREF = Corrente/2$, quindi $1,5/2 = 0,75$ V. Questa tensione si ottiene ruotando il trimmer SMD presente sul driver con un piccolo giravite, con la punta in ceramica o in plastica

(importante per evitare cortocircuiti che distruggerebbero il driver!). Serve mano leggera ed un po' di pazienza, la tensione va misurata con un multimetro digitale, posizionando il puntale negativo a massa e quello positivo direttamente sul trimmer metallico. La VREF può avere una tolleranza di ± 10 mV, quindi, nel nostro caso, tra 0,74 e 0,76 V. Nella **Fig. 5** i tre trimmer sono circondati da un riquadro giallo, sono molto piccoli, quindi servirà una punta sottilissima per la loro regolazione.

PROGRAMMAZIONE DI ARDUINO NANO

La board Arduino NANO, prima dell'innesto sul PCB, va programmata con l'ultima versione del firmware, scaricabile dal link: <https://github.com/gnea/grbl/releases/download/v1.1h.20190825/>

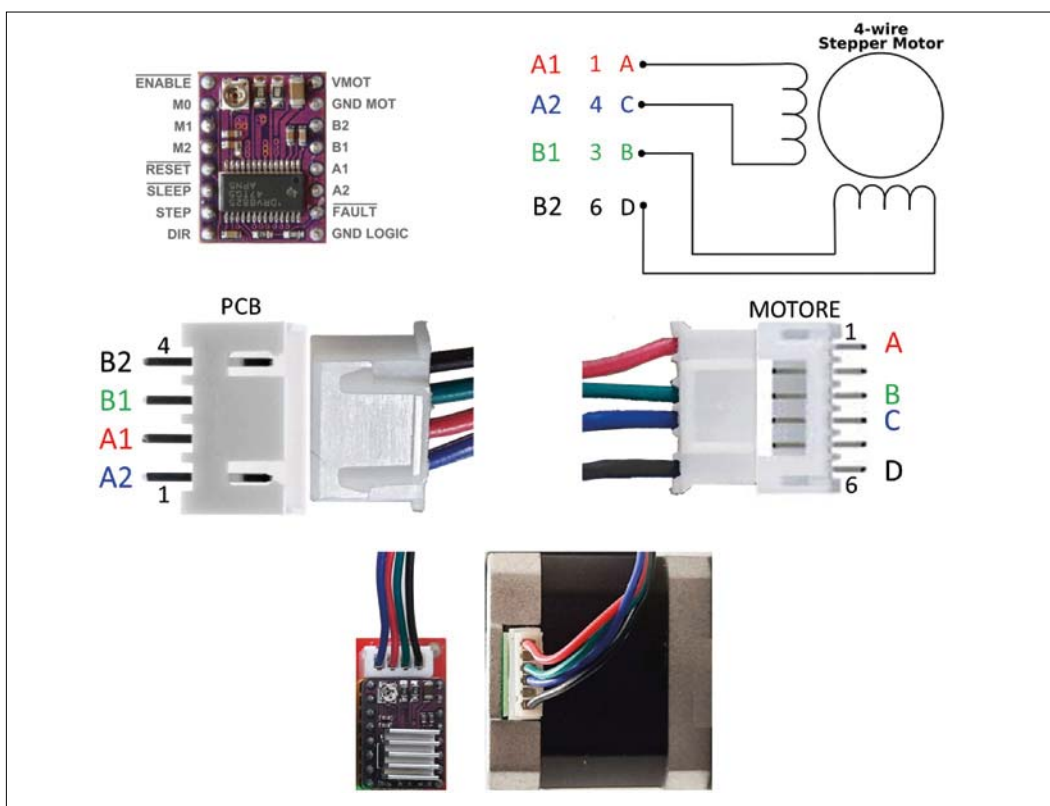


Fig. 4
Cavetti di collegamento per i motori passo-passo.

grbl_v1.1h.20190825.hex. L'operazione è molto semplice, occorre scaricare l'ultima versione del programma **xLoader.exe** dal link <https://github.com/xinabox/xLoader/releases> (nel momento in cui scriviamo è la 1.339) ed eseguirla; nella piccola maschera che si apre (**Fig. 6**), occorre cliccare sui "...", e fornire il percorso del file *grbl_v1.1h.20190825.hex*, selezionare il dispositivo COM3 e la porta COM alla quale è collegato l'Arduino NANO, quindi cliccare su UPLOAD; se è stato fatto tutto correttamente, dopo alcuni secondi, alla base della maschera, apparirà il messaggio "uploaded". Specifichiamo che i nostri test sono stati effettuati su una board originale, quindi non possiamo garantire il buon esito del caricamento se userete un modello compatibile. Terminata questa operazione è possibile inserire la board nei due connettori header, naturalmente con l'alimentazione spenta e facendo attenzione al verso di inserimento, la porta USB mini deve essere rivolta verso il bordo del PCB. L'ultima operazione da fare è quella di caricare nella EEPROM di Arduino NANO i parametri della CNC, senza di essi il comportamento del movimento degli assi sarebbe completamente errato! Bisogna scaricare il programma Candle, versione 1.1.7, dal seguente link: https://github.com/Denvi/Candle/releases/download/v1.1/Candle_1.1.7.zip. Una volta decompres-

sa la cartella, occorre collegare il PC all'Arduino NANO tramite un cavetto USB mini, accendere l'alimentatore esterno ed eseguire il file *Candle.exe*.

Nota bene: le funzionalità del programma Candle vengono disattivate automaticamente se al PCB è collegato il controller offline, quindi, durante questa procedura, esso dovrà essere scollegato. Al primo avvio probabilmente tutta la sezione dei pulsanti apparirà disattivata, bisogna andare nel menu Service-Settings e selezionare la porta COM alla quale è collegato Arduino NANO, in genere è sufficiente premere sul pulsantino situato accanto al campo "Port" per la selezione automatica, in caso contrario bisognerà andare per tentativi, selezionando tra le voci in elenco (**Fig. 7**).

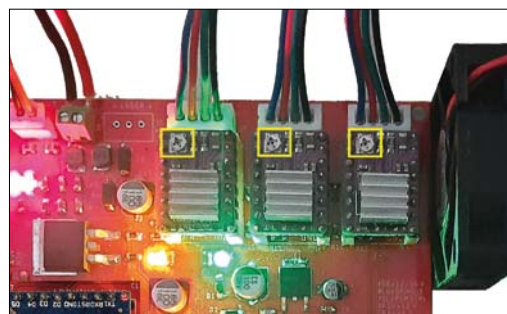


Fig. 5
Taratura dei driver DRV8825.

Appena si clicca su OK le icone del programma risulteranno attivate, in questa fase è bene evitare di utilizzarle, perché si potrebbero provocare comportamenti errati dei motori. L'unica sezione che ci interessa è il riquadro posizionato in basso a destra, chiamato "**Console**" (**Fig. 8**) che è il monitor del linguaggio grbl, infatti qui si vedranno tutti i comandi eseguiti manualmente dall'utente, o in automatico durante l'incisione. Noi lo useremo per inviare i parametri della nostra CNC alla EEPROM di Arduino NANO; questi parametri devono essere forniti dal produttore della CNC, se non si riesce ad averli, si potrà comunque partire da quelli usati da noi e successivamente procedere per tentativi, come vedremo.

Nella **Tabella 3** abbiamo riportato l'elenco dei comandi con i relativi valori per la CNC2018; la maschera necessaria per la scelta del valore di alcuni di essi è visibile nella **Tabella 4**.

Abbiamo aggiunto delle note esplicative, alcuni di questi parametri sono fortemente consigliati nella wiki, altri riguardano funzioni opzionali, per ora vedremo solo quelli che potrebbe essere necessario personalizzare, disponendo di un modello di CNC e/o di motori diversi dai nostri, dando però per scontato che siano tutti e tre uguali tra loro.

Dopo queste informazioni, vedremo come passare i parametri alla macchina.

Intanto consigliamo di creare una tabella con tutti i 34 parametri, e riportarvi i valori che vi abbiamo indicato, modificandoli opportunamente, in base alle vostre esigenze, così poi sarà più comodo inserirli in sequenza nella Console.

Vediamo ora alcuni parametri che dovrete personalizzare in base ai componenti utilizzati:

- **\$30**: questo valore serve per utilizzare correttamente la regolazione di velocità dello spindle, quindi bisogna conoscerne le caratteristiche. Nel nostro caso lo spindle lavora a 9.000 rpm se alimentato a 36 V. Chiariamo che, se disponete di un motore che lavora a 1.000 rpm, fornendo un valore maggiore non si ottiene certo un aumento della velocità, quindi è bene mettere sempre il valore reale;
- **\$100+102**: numero di step che il motore deve compiere per generare un movimento di un millimetro. Questo valore va calcolato con la formula: $step_giro * microstep_drv / mm_giro$, in cui: *step_giro* va ricavato dal datasheet del motore, ma nella stragrande maggioranza dei casi è 200; *microstep_drv* è l'impostazione del driver che alimenta il motore, nel nostro

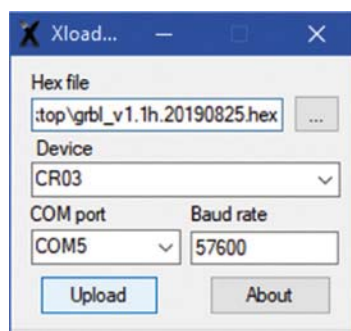


Fig. 6
Caricamento del firmware su
Arduino NANO.

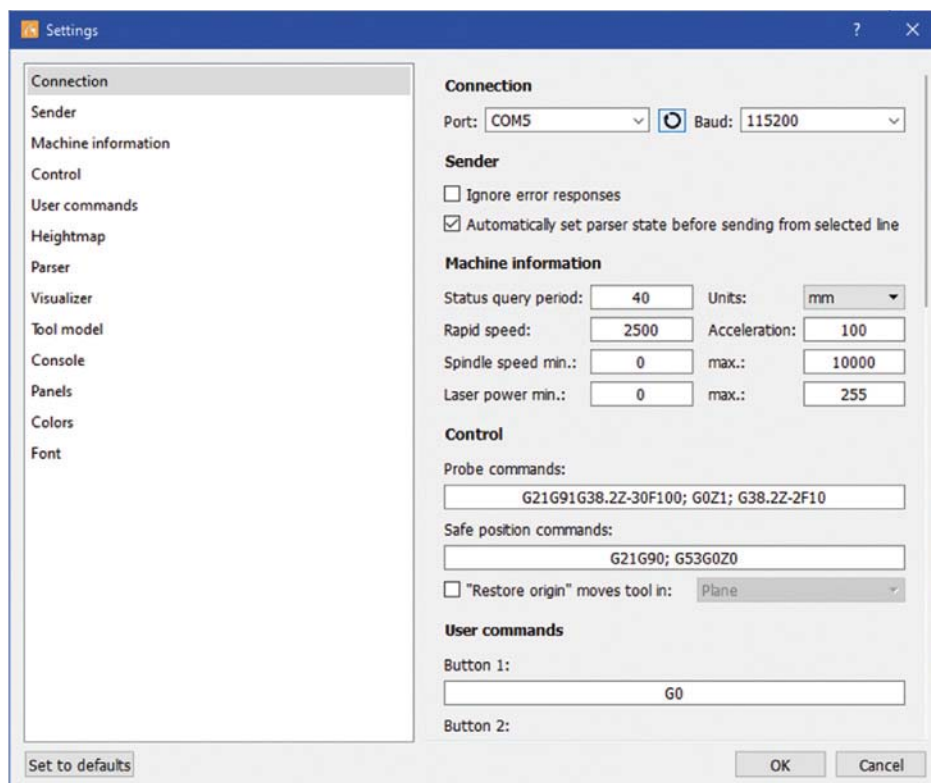


Fig. 7
Il programma Candle.

COMANDO	DESCRIZIONE	VALORI	NOTE
\$0=10	durata impulso di step, μ s	v. datasheet motore	10 è il valore standard consigliato
\$1=255	ritardo disattivazione motori dopo il movimento, ms	1÷254, 0=disabilitato, 255=motori sempre attivi	255 valore consigliato
\$2=0	maschera inversione impulso di step	standard = basso -> alto (durata \$0) -> basso	maschera uguale a quella del \$3
\$3=2	maschera inversione direzione assi	0÷7, vedi Tabella 4	trovare valore per la corretta direzione dei 3 motori
\$4=0	stato pin abilitazione stepper, boolean	0=alto disabilita, basso abilita, 1=basso disabilita, alto abilita	
\$5=0	stato pin limiti assi, boolean	0=alto disabilita, basso abilita, 1=basso disabilita, alto abilita	
\$6=0	stato pin probe, boolean	0=alto disabilita, basso abilita, 1=basso disabilita, alto abilita	
\$10=1	maschera del report di stato	0=PM no, PL sì; 1= PM sì, PL no; 2=PL sì + Buffer	PM=posizione macchina, PL=posizione lavoro
\$11=0.010	deviazione giunzione, mm	valore basso=lento, più sicuro; alto=veloce, meno sicuro	rallentamento macchina per attraversamento angoli
\$12=0.002	tolleranza dell'arco, mm	valore basso=lento, più preciso; alto=veloce, meno preciso	precisione tracciato cerchi e archi
\$13=0	report posizionamento in mm/pollici, boolean	0=mm, 1=pollici	
\$20=0	limiti software, boolean	0=disabilitato, 1=abilitato	fa riferimento ai valori \$130÷132 - va abilitato \$22
\$21=0	limiti hardware, boolean	0=disabilitato, 1=abilitato	servono gli switch di finecorsa
\$22=0	ciclo di homing, boolean	0=disabilitato, 1=abilitato - ciclo iniziale di posizionamento	servono gli switch di finecorsa
\$23=0	maschera inversione direzione homing	da usare in base al posizionamento degli switch finecorsa	maschera uguale a quella del comando \$3
\$24=25	velocità di avanzamento verso l'home, mm/min		
\$25=500	velocità di ricerca degli switch finecorsa, mm/min		
\$26=25	antirimbalo per gli switch finecorsa, ms	5÷25 ms, ma anche maggiore se necessario	
\$27=1	scostamento dagli switch finecorsa dopo l'homing, mm		serve per evitare attivazioni indesiderate degli switch
\$30=9000	velocità massima del mandrino, rpm	spindle ELEMANDER11: 12V-3.000 / 24V-6.000 / 36V-9.000 / 48V-12.000	valore massimo correlato al PWM=255
\$31=0	velocità minima del mandrino, rpm		valore minimo correlato al PWM=0
\$32=0	modalità laser, boolean	0=disabilitato, 1=abilitato	
\$100=1600	step/mm asse X	steps_mm=step_giro*microstep_drv/mm_giro	step_giro = 200 microstep_drv = 32 mm_giro = 4
\$101=1600	step/mm asse Y	steps_mm=step_giro*microstep_drv/mm_giro	step_giro = 200 microstep_drv = 32 mm_giro = 4
\$102=1600	step/mm asse Z	steps_mm=step_giro*microstep_drv/mm_giro	step_giro = 200 microstep_drv = 32 mm_giro = 4
\$110=500	velocità massima asse X, mm/min		
\$111=500	velocità massima asse Y, mm/min		
\$112=500	velocità massima asse Z, mm/min		
\$120=500	accelerazione asse X, mm/sec ²		
\$121=500	accelerazione asse Y, mm/sec ²		
\$122=500	accelerazione asse Z, mm/sec ²		
\$130=200	Corsa massima asse X, mm	dimensione larghezza piatto	serve solo se si attiva \$20 ed eventualmente \$22
\$131=180	Corsa massima asse Y, mm	dimensione profondità piatto	serve solo se si attiva \$20 ed eventualmente \$22
\$132=55	Corsa massima asse Z, mm	escursione spindle	serve solo se si attiva \$20 ed eventualmente \$22

Tabella 3
I parametri di configurazione grbl per la CNC.

→ **Tabella 4**

Valore	Maschera	Inverti X	Inverti Y	Inverti Z
0	00000000	no	no	no
1	00000001	sì	no	no
2	00000010	no	sì	no
3	00000011	sì	sì	no
4	00000100	no	no	sì
5	00000101	sì	no	sì
6	00000110	no	sì	sì
7	00000111	sì	sì	sì

caso è 32 (come abbiamo visto a proposito dei settaggi M0, M1 e M2 dei DRV8825); *mm_giro* è lo spostamento dell'asse in mm quando si fa compiere alla barra filettata un giro completo; nel nostro caso, ruotando manualmente la barra filettata dell'asse X, il blocco del mandrino si è spostato di 4 mm, quindi $200 \times 32 / 4 = 1600$. Poiché i motori e le barre filettate della nostra CNC sono identiche tra loro, abbiamo usato lo stesso valore per i tre comandi; in caso contrario il calcolo va effettuato separatamente per ogni asse;

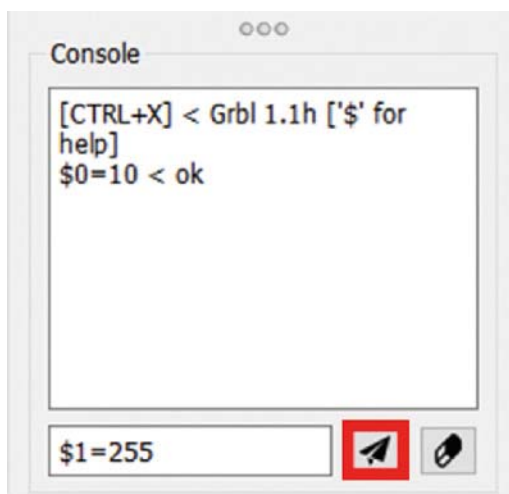
- **\$110+112:** velocità massima dell'asse in mm/min; il valore 500 in genere va bene per tutte le macchine, garantendo una buona velocità senza perdita di passi, ma vedremo più avanti come trovare la velocità ottimale per ogni motore;
- **\$120+122:** accelerazione dell'asse in mm/sec²; anche in questo caso noi abbiamo adottato il valore 500 e vedremo come trovare quello idoneo per altri modelli di motori;
- **\$130+132:** corsa massima dell'asse in mm; precisiamo che questi valori serviranno solo se si intende attivare l'opzione dei limiti software o

di quelli hardware, ma è bene inserirli ugualmente per usi futuri. Servono i valori dell'escursione del piatto nel suo verso di scorrimento (Y) e l'escursione del blocco del mandrino in orizzontale (X) ed in verticale (Z); per quest'ultimo valore conviene considerare anche la lunghezza del bulino; in pratica bisogna stringere il bulino nella morsa del mandrino e poi partire dall'alto e scendere fino a sfiorare il piatto con la punta.

Siamo pronti per il passaggio finale, che consiste nell'inviare questi parametri all'EEPROM di Arduino NANO. Durante questa fase non è necessario collegare l'alimentatore esterno e conviene lasciare staccati i fili dello spindle e gli spinotti dei tre motori. L'operazione è molto semplice, una volta avviato il Candle, bisogna controllare che nella Console appaia il messaggio "[CTRL+X] < Grbl 1.1h ['\$' for help]", se invece si leggerà "Serial port error 11: Impossibile trovare il percorso specificato." significa che non è stata settata correttamente la porta COM, e la cosa sarà confermata dal fatto che i vari pulsanti presenti nella maschera risulteranno disattivati. Una volta risolto questo problema, nel campo alla base della Console bisogna scrivere i codici dei parametri con il rispettivo valore, uno per volta, e quindi cliccare sul simbolo presente subito a destra (evidenziato con un riquadro rosso nella **Fig. 8**); se l'istruzione sarà andata a buon fine, la Console mostrerà sia il comando che la conferma di ricezione "< ok"; se invece esce un messaggio di errore è inutile andare avanti se prima non si risolve il problema.

I parametri vanno inseriti tutti, non solo quelli personalizzati, seguendo l'ordine della tabella; in questa fase consigliamo di non modificare i valori proposti da noi, eccetto quelli che abbiamo spiegato prima, se necessario. Dopo aver inserito tutti i 34 parametri dell'elenco occorre scollegare il cavo USB, inserire i fili dello spindle nel connettore a vite, sincerandosi che ci sia un contatto perfetto, inserire gli spinotti dei tre motori e della ventola di raffreddamento nei relativi connettori JST, collegare l'alimentatore esterno e accenderlo, quindi ricollegare il cavo USB. In questa fase non bisogna montare il bulino nel morsetto dello spindle perché, durante le prove, si potrebbe rompere, a causa di una manovra sbagliata.

La prima operazione da fare è il controllo dei tre motori, per verificare che i relativi pulsanti corrispondano ai rispettivi assi e le direzioni dei movimenti siano corrette. A tale scopo possiamo ricorrere al riquadro "Jog", visibile nella **Fig. 9**. Ricordiamo che l'eventuale controller offline



→ **Fig. 8**
Inserimento dei
parametri nella
EEPROM.

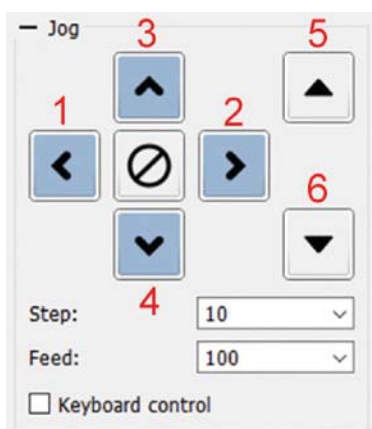


Fig. 9
Pulsanti di comando dei tre motori.

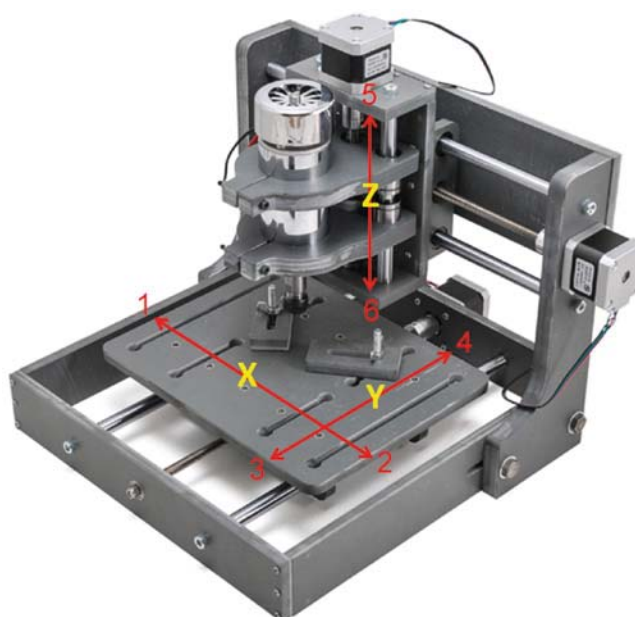


Fig. 10
Orientamento dei tre assi.

dovrà restare scollegato. Prima di tutto vanno settati i due campi **"Step=10"** e **"Feed=100"**, che rappresentano rispettivamente la lunghezza del movimento in mm e la velocità in mm/min. Questi valori sono ideali per le verifiche che dobbiamo fare; in fase di incisione il valore di Step sarà irrilevante, mentre Feed potrà essere portato alla velocità massima prevista per i motori.

Le quattro frecce 1-2-3-4 servono per gli assi X e Y, il pulsante centrale serve per bloccare i motori in caso di emergenza, le due frecce 5-6 servono per il movimento dello spindle. Per meglio comprendere l'orientamento dei tre assi si può fare riferimento alla **Fig. 10**, nella quale abbiamo riportato i riferimenti numerici dei 6 pulsanti.

Vediamo ora come si dovrebbero comportare i motori usando questi pulsanti; il condizionale è d'obbligo, in quanto potrebbero essere stati invertiti gli spinotti o potrebbero risultare invertite le direzioni dei movimenti. Prima di procedere bisogna far ruotare manualmente le barre filettate al fine di portare il blocco dello spindle più o meno al centro del piatto ed al centro della sua stessa corsa; questa precauzione ci permetterà di evitare scarrellamenti dei motori in caso di manovre o comportamenti errati. A questo punto possiamo iniziare il nostro test, la **Fig. 11** mostra i risultati attesi con un singolo clic su ogni pulsante.

Ad ogni clic su un pulsante deve corrispondere un movimento preciso di 10 mm lungo l'asse prescelto, nella direzione prevista. I pulsanti vanno testati tutti, iniziando da quelli direzionali. È necessario un chiarimento: mentre i pulsanti degli assi X e Z

provocano il movimento dello spindle nella stessa direzione delle frecce, quelli dell'asse Y provocano invece il movimento del piatto in direzione opposta alle frecce; in pratica premendo la freccia Y 3 il piatto si deve spostare in avanti, mentre con la freccia Y 4 il piatto si sposta verso dietro. La ragione di questo apparente controsenso sta nel fatto che tutti i pulsanti di comando sono riferiti al movimento dello spindle, ma mentre sugli assi X e Z esso si muove effettivamente, sull'asse Y è fermo, ed è il piatto che deve spostarsi in direzione opposta. Quindi, quando si decide di spostare lo spindle in avanti è il piatto che va dietro, e viceversa.

Vediamo le possibili problematiche e come risolverle. Se la pressione di un pulsante provoca il movimento di un asse diverso da quello previsto, significa che sono stati invertiti gli spinotti sul controller, è sufficiente rimetterli a posto.

Pulsante	Movimento
←	Lo spindle si muove verso sinistra
→	Lo spindle si muove verso destra
↑	Il piatto si muove verso avanti
↓	Il piatto si muove verso dietro
▲	Lo spindle si muove verso l'alto
▼	Lo spindle si muove verso il basso
⊗	Qualsiasi movimento in corso si blocca

Fig. 11
Movimenti dei motori.



Fig. 12
Controller offline

Se la pressione di un pulsante provoca il movimento dell'asse previsto ma in direzione opposta a quella attesa, significa che il motore è alimentato con polarità opposta a quella richiesta. Questo problema si risolve via software, tramite un comando da inviare con la Console, ma prima di eseguirlo occorre testare tutti i pulsanti e vedere se il problema riguarda solo un motore o due, o addirittura tutti e tre. Facciamo un esempio: se i movimenti degli assi X e Y risultano invertiti, mentre quelli dell'asse Z corrispondono, dobbiamo localizzare sulla maschera riportata nella **Tabella 4** la riga che inverte i soli assi X e Y, quindi la n.3; il comando da dare sulla Console è **"\$3=3"**, subito dopo bisogna spegnere l'alimentatore esterno, attendere una decina di secondi, riaccenderlo e quindi ripetere i test di tutti i pulsanti; è importante ricontrollare tutto, perché potrebbe succedere che si sistemi un movimento ma se ne inverta un altro. In genere il comando corretto è immediatamente risolutivo, in caso contrario si può andare tranquillamente per tentativi, fino a trovare il valore di \$3 (tra 0 e 7) che permette il corretto movimento di tutti e tre gli assi. Se i motori non si muovono correttamente, facendo molto rumore oppure bloccandosi, significa che i valori di velocità (**\$110+112**) e/o di accelerazione (**\$120+122**) sono eccessivi, quindi bisogna diminuirli fino ad ottenere un comportamento normale; conviene operare su un solo motore, inviando il comando tramite Console, testando prima la velocità e poi, eventualmente l'accelerazione.

Una volta trovata la combinazione con i massimi valori accettati, conviene ridurli di un 30% e poi assegnarli inviandoli ai tre motori (sempre a patto che siano identici tra loro).

Una volta che è tutto a posto, si può aumentare il

valore di Step a 50, per selezionare un movimento di 5 cm, quindi far partire il motore dell'asse X (per comodità) e premere subito dopo il pulsante di blocco, per verificare che effettivamente il movimento si blocchi istantaneamente.

L'ultima verifica da fare riguarda la precisione dello spostamento, serve un righello di precisione, meglio ancora se si dispone di un calibro digitale; bisogna impostare Step=20 e prendere il punto di riferimento della posizione attuale del blocco dello spindle, quindi dare un comando di movimento, prima sull'asse X e poi su quello Z, e verificare se lo spostamento corrisponde esattamente a 20 mm, quindi effettuare la verifica dell'asse Y con lo spostamento del piatto; la verifica va fatta su entrambe le direzioni di ogni asse. Se lo spostamento dovesse risultare errato significa che non è stato calcolato correttamente il valore dei parametri **\$100+102**; posto che **step_giro=200** è un valore standard quasi universale e **micro-step_drv=32** è obbligato dall'impostazione data da noi ai DRV8825, l'unico valore da verificare è **mm_giro**, operazione semplice, come abbiamo spiegato in precedenza. Per concludere vediamo come usare il controller offline, per chi deciderà di dotarsi di questo accessorio opzionale. In **Fig. 12** possiamo osservare il modello utilizzato per le nostre prove, sul quale abbiamo riportato le funzioni dei pulsanti.

La sua caratteristica peculiare è quella di permettere di lavorare con la CNC senza dover necessariamente disporre di un PC, essendo necessario solo il file generato dal software CAM; esso viene caricato su una micro SD card, la quale può essere anche di bassissima capacità, data l'esigua dimensione dei file. Il controller offline, una volta collegato allo specifico connettore, parte automaticamente ed esclude il Candle, rendendo di fatto inutile il collegamento al PC. All'accensione, la mancata presenza di una micro SD card nello slot viene segnalata istantaneamente sul piccolo display LCD; appena si inserisce la scheda il display mostra il menu visibile in **Fig. 13**.

I due pulsanti X-/X+ permettono di spostarsi tra le tre voci del menu, mentre il pulsante OK attiva la voce selezionata. Il menu **"Ctrl"** è quello che permette di accedere alle funzioni di controllo della macchina (**Fig. 14**).

Il movimento dei tre assi avviene premendo i rispettivi pulsanti, l'attivazione o disattivazione dello spindle si ottiene premendo il pulsante OK, mentre il pulsante Step, ad ogni pressione, cambia il relativo valore. Per tornare al menu principale

Fig. 13
Menu di controllo
della CNC



occorre tenere premuto per un paio di secondi il pulsante Step che assume, in questo caso, la funzione Exit. Il menu **"File"** mostra l'elenco dei file presenti nella micro SD card, i pulsanti Y-/Y+ permettono di selezionare quello da utilizzare, il



Fig. 14
Menu di controllo della CNC.

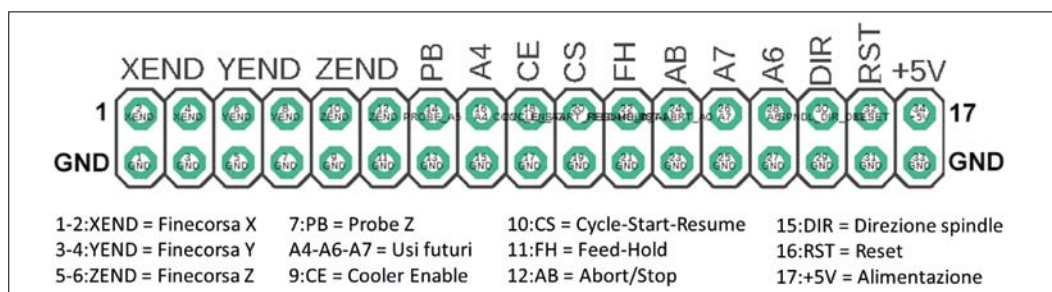


Fig. 15
I contatti del connettore esterno.

pulsante Exit riporta al menu principale senza selezionare alcun file, il pulsante OK seleziona quello attivo e mostra il messaggio **"Ready to print"**; premendo nuovamente il pulsante OK si avvia immediatamente il processo di incisione della CNC, ed il display indicherà costantemente la percentuale di esecuzione, mentre una pressione di circa due secondi sul pulsante Exit annulla la selezione, o interrompe il processo di incisione, e ritorna al menu File. Il menu **"Sset"** apre una maschera divisa in due colonne, la prima (Move) permette di selezionare il valore di Step (0.1, 1, 5, 10 mm), la seconda (Bdrte) serve per settare la velocità della comunicazione seriale con Arduino NANO, tra 9600 e 115200 bps, quest'ultimo è il valore di default. Gli spostamenti in questo menu avvengono mediante i pulsanti X-/X+/Y-/Y+ mentre il pulsante Exit conferma le selezioni e torna al menu principale. Come potete vedere l'uso di questo controller offline è semplice ed intuitivo, chiaramente non può garantire tutte le funzioni del programma Candle però, quando non si hanno particolari esigenze, si può rivelare molto utile. Una volta completate tutte le operazioni ed i necessari test, si potrà montare il controller nello specifico contenitore, per il quale abbiamo fornito il file .stl. Nel caso del nostro contenitore, il PCB andrà distanziato dalla base di circa 5 mm, ricorrendo a 4 distanziali in plastica (adesivi o a vite), in modo da poter sfruttare agevolmente le fessure predisposte per i collegamenti esterni. Per poter utilizzare correttamente il connettore esterno, occorre avere ben presente l'elenco delle connessioni ai pin strip, serigrafate sul PCB; consigliamo di riprodurle esternamente, magari con una etichetta adesiva. In **Fig. 15** abbiamo riportato il connettore esterno, visto dal lato dei

contatti, con le stesse sigle presenti sul PCB ed il loro significato, mentre la funzione l'abbiamo già descritta parlando dei pin di Arduino Nano. Da notare che la fila inferiore di pin è tutta collegata a massa, mentre i pin per i finecorsa X-Y-Z sono doppi in quanto abbiamo previsto la possibilità di montare due switch per ogni asse, in modo da controllare entrambe le estremità della corsa.

CONCLUSIONI

Termina qui la presentazione di questa eccellente scheda, che potrà sostituire quella in vostro possesso oppure rappresentare la dotazione di una meccanica acquistata senza controller. La riprenderemo con ulteriori articoli nei quali vi spiegheremo come usare alcuni programmi free per realizzare facilmente i vostri PCB in casa. ■



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>



La Presa su misura per te!

PRESA MULTIPLA SCHUKO 6 POSIZIONI CON PROTEZIONE DA SOVRATENSIONE PROTECTION

€ 21,⁵⁰

Cod. **WTPRESAM6P**



Presa 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.

€ 19,⁹⁰
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**

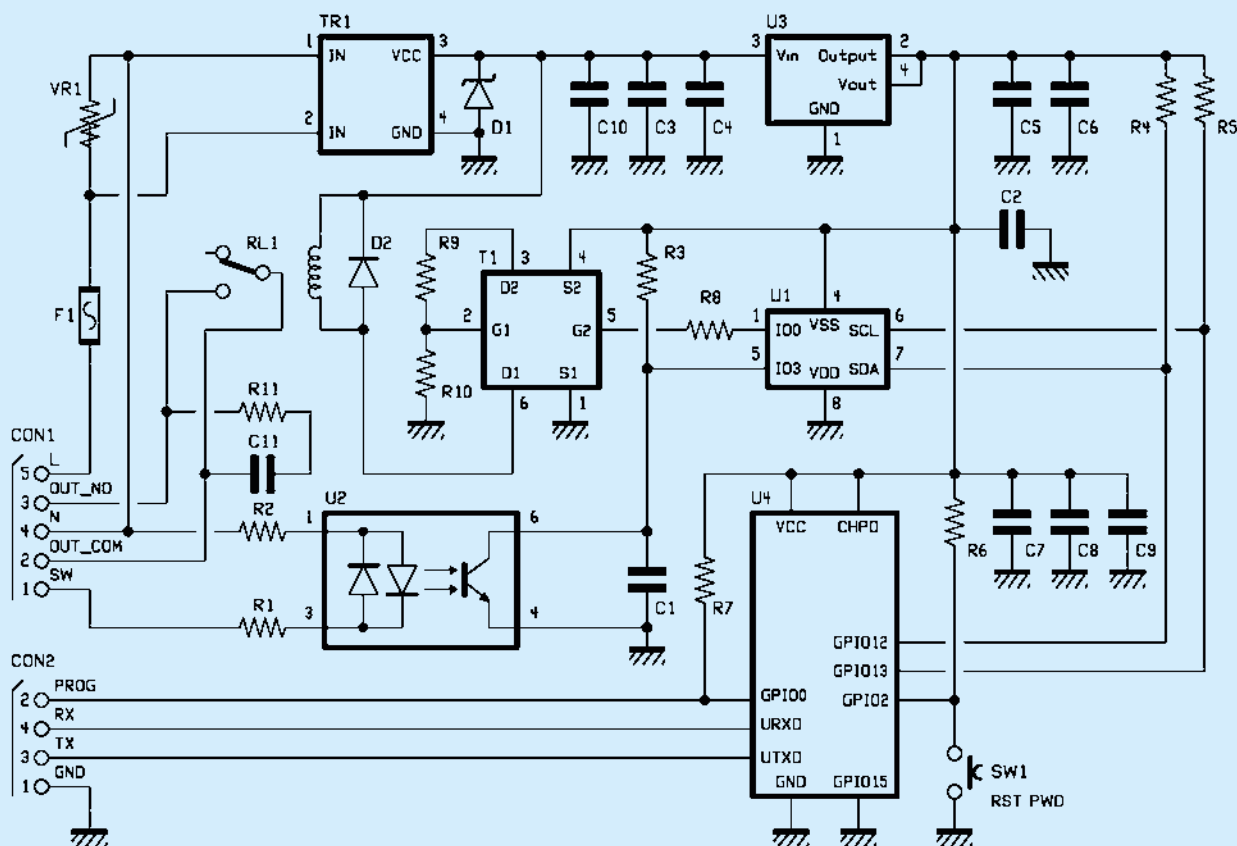
Presse in grado di proteggere i dispositivi elettrici dalle sovratensioni come quelle causate dai fulmini. Dispone inoltre di LED di stato che si accende appena viene alimentata.

SMART GATE LOCK

di MAURIZIO ŠKERLIČ

Serratura IoT controllata da smartphone o da assistente vocale Amazon Alexa e Google Assistant. Basato sul modulo WiFi ESP03 e sul server Sinric Pro, si configura tramite interfaccia web.

In questo mondo sempre più connesso, riceviamo numerose richieste di progetti inerenti al mondo dell'IoT e per andare incontro ai vostri desideri cerchiamo di proporre idee sempre più al passo con i trend. Lo "Smart gate lock" è un progetto che va in tale direzione, perché si tratta di una serratura elettronica per comandare l'apertura del cancello o la basculante del garage, da abbinare ai dispositivi dimmer IoT (Smart dimmer) e switch IoT (Smart switch singolo e doppio) pubblicati nel recente passato di Elettronica In, per allestire la propria casa domotica. Il progetto, qui descritto, è sostanzialmente un relé WiFi controllato mediante App da smartphone, mediante interfaccia web oppure attraverso l'assistente vocale Amazon Alexa o Google assistant. È un dispositivo compatto con interfaccia WiFi e dotato di un relé elettromeccanico con contatti puliti normalmente aperti per controllare serrature elettroniche oppure cancelli elettrici.



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 alimentato 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 ad alto livello 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 alte rispettivamente le linee SDA e SCL del segnale I2C. Il pulsante SW1 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, nello schema elettrico è siglato con T1. Il diodo D2 serve a sopprimere le extratensioni di segno opposto generate dalla bobina quando il relé viene diseccitato e serve a non far bruciare T1.

Il carico viene collegato tra i piedini 2 e 3 e la tensione di rete neutro e fase viene collegata rispettivamente ai piedini 4 e 5 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 dall'eventualità di una tensione eccessiva in uscita dall'alimentatore, in caso quest'ultimo si guastasse.

In parallelo ai contatti è stata inserita una rete snubber RC, formata dalla resistenza R11 e dal condensatore C11, che ha una duplice funzione: sopprime l'arco elettrico sui contatti (che si verifica con carichi induttivi o capacitivi) proteggendoli dalla precoce usura e riduce i disturbi elettromagnetici durante la commutazione. Nel caso vi servano dei contatti puliti, per esempio per comandare l'attivazione del motoriduttore del cancello, questi componenti potete eliminarli.

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.it 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 mantenere le dimensioni più piccole possibili, i PCB sono stati realizzati con vetronite da 0,8 mm.

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), della pasta flussante a media densità, di 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 le piazzole che di loro dovrebbero rimanere separate. Ciò detto, iniziate spalmando con un pennellino 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 alimentatore switching AC/DC e 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'e-

CARATTERISTICHE TECNICHE

- Carico massimo 350W
- 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 e BTicino
- Timer spegnimento automatico
- Controllo locale da pulsante
- Controllo remoto mediante App
- Sincronizzazione stato di apertura/chiusura
- Controllo vocale mediante Amazon Alexa e Google Assistant
- Collegamento in rete da router WiFi
- Dimensioni 36 x 43 x 21 mm

piano di MONTAGGIO

Elenco Componenti:

R1, R2: 150 kohm 300 V (0805)	T1: BSD235CH
R3: 470 kohm (0402)	VR1: V430CH8
R4, R5: 2,2 kohm (0603)	TR1: RAC05-05
R6, R7: 10 kohm (0603)	F1: F 2,5A
R8: 120 ohm (0402)	U1: PCA9536TK
R9: 10 ohm (0402)	U2: TLP184
R10: 10 kohm (0402)	U3: NCP1117ST33T3G
R11: 100 ohm 300 V (2010)	U4: ESP-03
C1: 1 µF 6,3 V (0603)	RL1: PCJ-112D
C2, C3, C6, C7: 100 nF (0402)	CON1: Morsettiera 5 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)	SW1: pulsante SMD 4.6 X 2.8 mm
C10: 220 µF 6,3 V (1206)	
C11: 100 nF 500 V (1812)	
D1: SMF5.0A	Varie:
D2: LL4148 (SOD123FL)	- circuito stampato S1594 (41x34 mm)

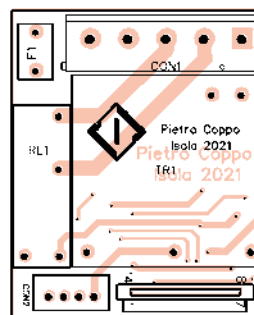
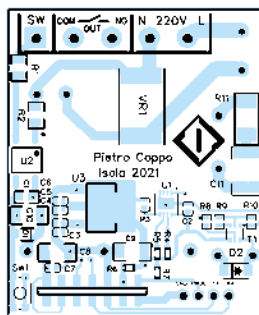


Fig. 1
Montaggio del
modulo ESP03 in
posizione verticale
sul PCB.



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 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

sterno 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 fluxante.

Fig. 2
Impostazioni di
programmazione
del modulo ESP03
nell'ambiente di
sviluppo
Arduino IDE.

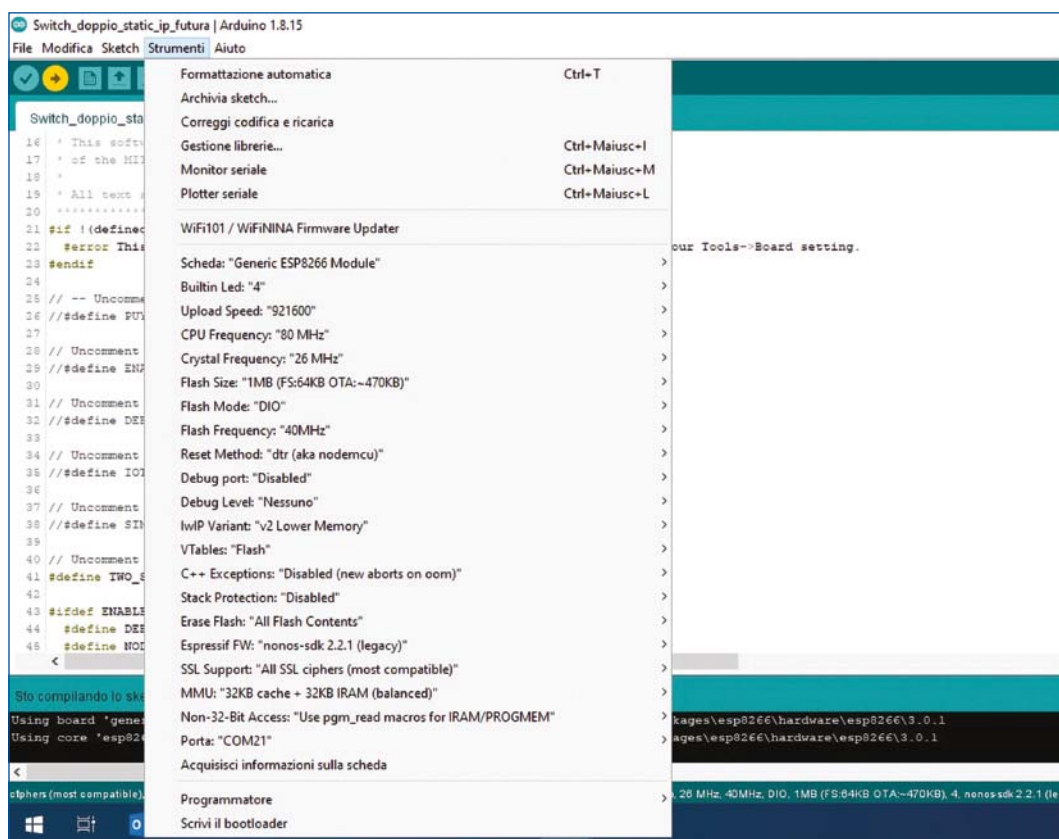


Fig. 3
Collegamento del
programmatore
seriale allo "Smart
gate lock".

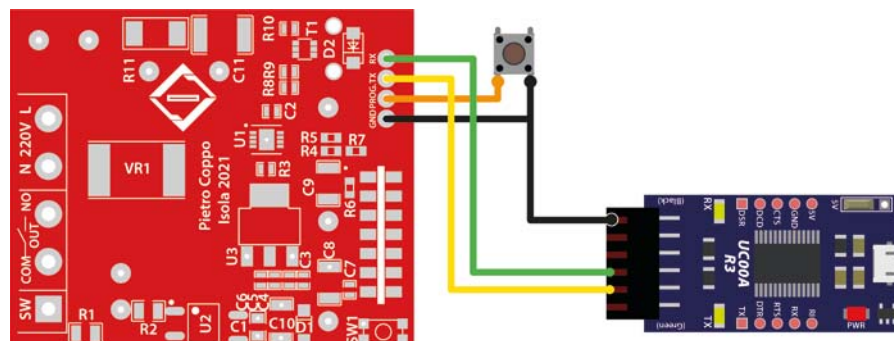




Fig. 4
Home page dello
"Smart gate lock".

Fig. 5
Schermata della
pagina web di
configurazione.

ESP8266, oppure con un qualsiasi convertitore USB TTL, purché compatibile con logica a 3,3 V, usato per programmare i moduli Arduino mini. Per il collegamento fate riferimento alla **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 lo "Smart gate lock" in modalità di programmazione premete il pulsante *Prog*, collegato al pin 2 del connettore CON2, prima di allacciarlo alla corrente elettrica, rilasciandolo dopo un secondo. Per il corretto collegamento del pulsante sul connettore CON2 fate riferimento allo schema di cablaggio proposto nella **Fig. 3**. Lo sketch è stato compilato utilizzando il package core ESP8266 versione 2.7.4 e le librerie *lotweb-conf* 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 almeno 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

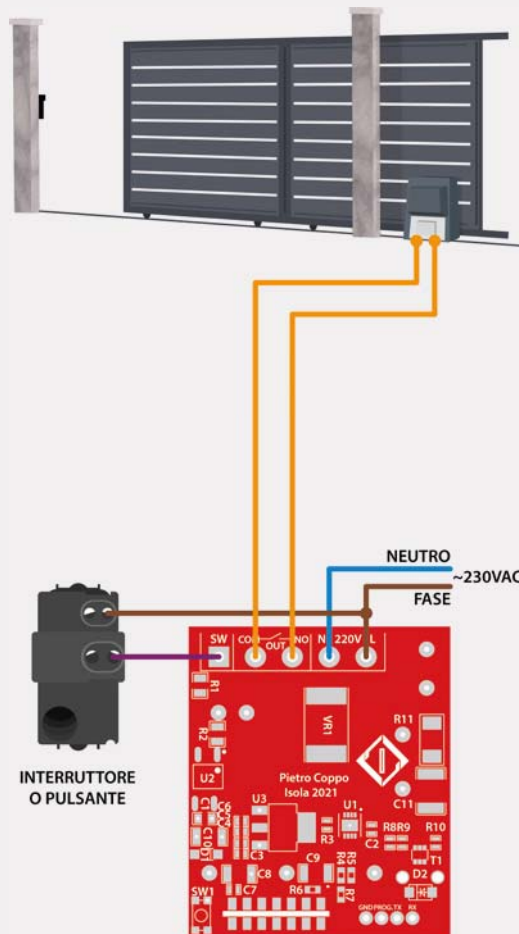


Fig. 6
Schema di
cablaggio dello
"Smart gate lock" al
cancello o alla porta
del garage.

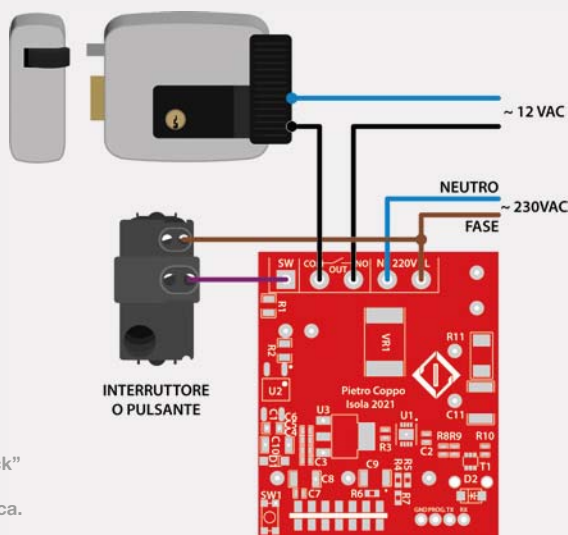


Fig. 7
Schema di cablaggio dello "Smart gate lock" alla serratura elettromeccanica.

Fig. 8
Impostazioni del Nuovo dispositivo in Sinric pro. Nella casella "Tipo di dispositivo" scegliete "SmartLock".

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 sketch togliete il commento davanti a `#define DEBUG_SKETCH`. La libreria `lotwebconf` di default fa visualizzare sulla seriale le chiavi del vostro account

`lot` in chiaro. Per motivi di sicurezza vi consigliamo di disabilitare tale opzione aggiungendo all'inizio del file `lotWebConfSettings.h` che si trova nella cartella SRC, la seguente riga di testo `#define IOTWEBCONF_DEBUG_DISABLED`.

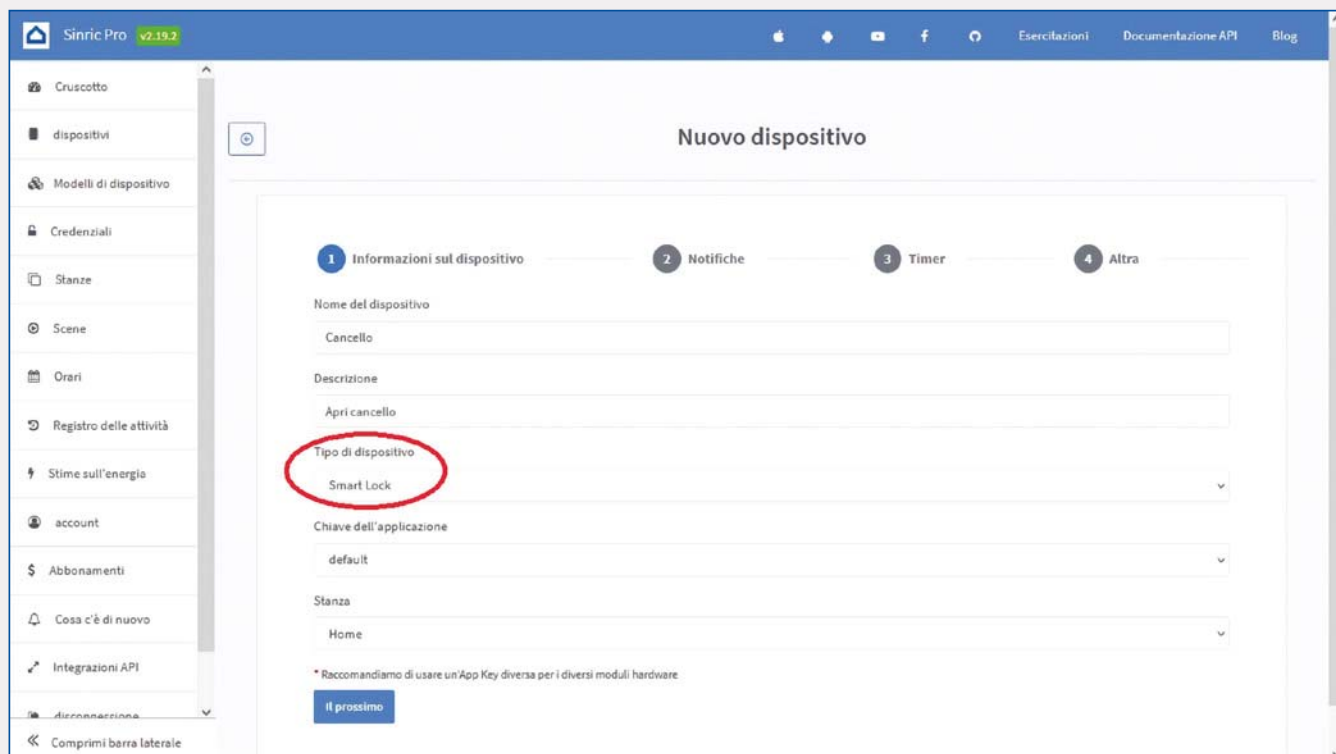
UTILIZZO DEL DISPOSITIVO

Bene, visto anche questo andiamo a spiegare le impostazioni dello Smart gate lock, 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 Control) dello Smart gate lock. L'indirizzo MAC è un numero a 48 bit composto da 6 numeri esadecimali da 8 bit separati dai due punti.



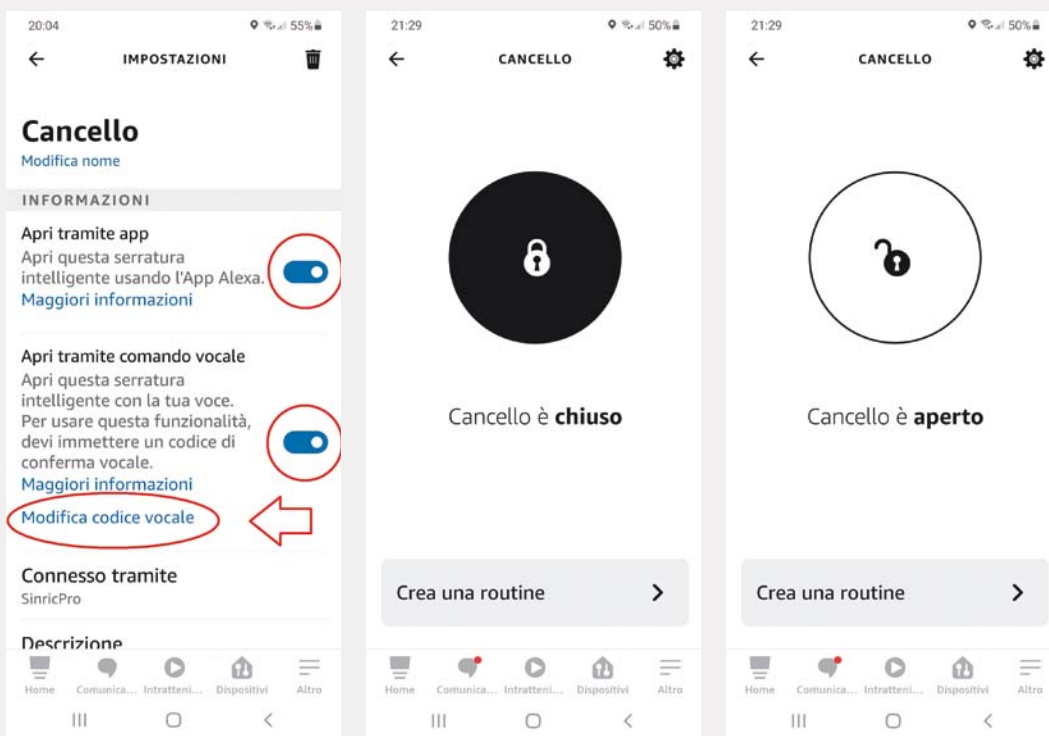


Fig. 9
Impostazioni
dell'App
Amazon
Alexa.

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

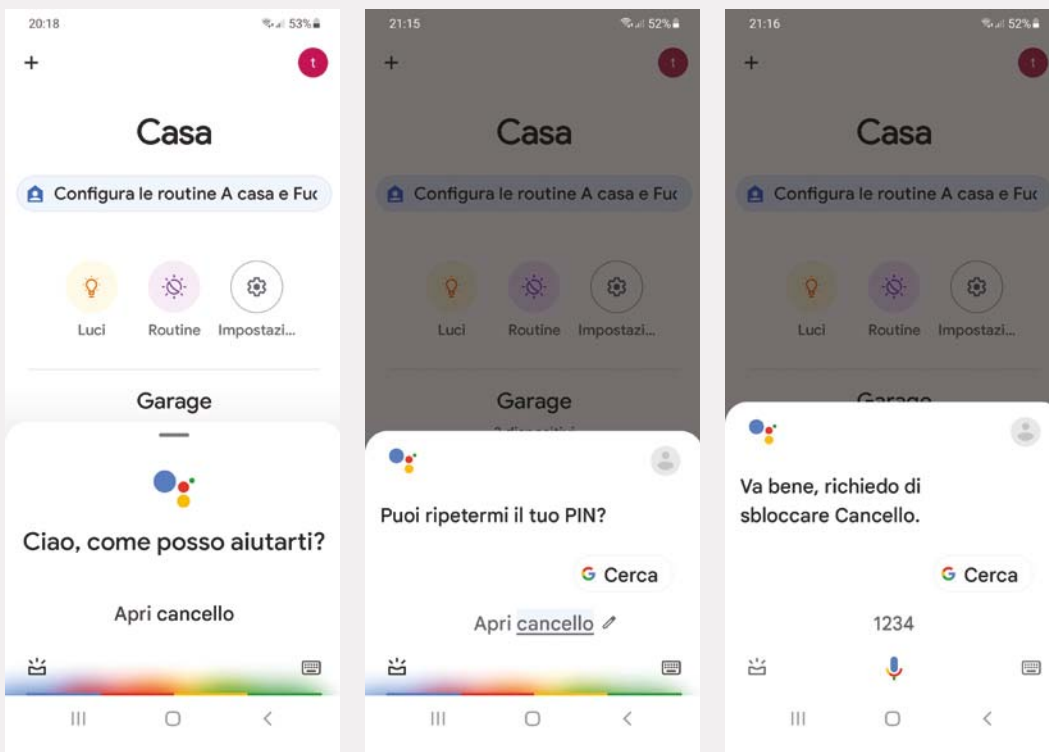
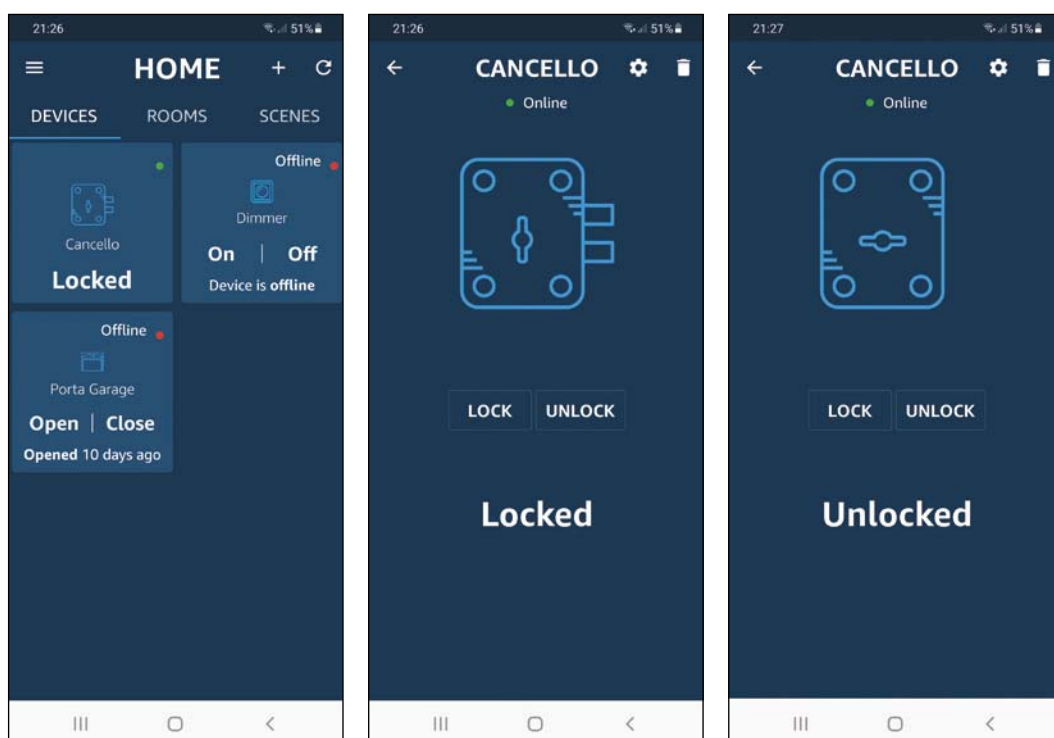


Fig. 10
Interfaccia
dell'App
Google home.

Fig. 11
Interfaccia dell'App
Sinric pro.



nome del dispositivo, modificare la password di sistema, inserire il nome e la password della rete SSID della rete WiFi di casa vostra e il "Startup delay" (le relative schermate sono proposte nella **Fig. 5**); quest'ultimo è il tempo in secondi che, ad ogni accensione o riavvio, lo Smart gate lock diventa un punto di accesso per consentirci di raggiungere la home page. Nella sezione "Device IP" selezionando la casella di spunta "Use 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 allo Smart gate lock, 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 primario 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. Terminata la configurazione di sistema salvatela cliccando sul pulsante Apply e verificate che il dispositivo si connetta correttamente alla rete wireless. Se avete scelto di ottenere il numero

IP in modo dinamico, per conoscere quello che vi ha assegnato il server DHCP della vostra rete, avete bisogno del numero MAC che avete precedentemente annotato. Aprite il terminale "cmd" di Windows con i permessi di amministratore e digitate il comando "arp -d *" per cancellare tutti gli host memorizzati, poi fate un ping di broadcasting della vostra rete ad esempio con il comando "ping 192.168.1.255" e infine lanciate il comando "arp -a". A terminale comparirà l'elenco dei numeri MAC di tutti i dispositivi presenti nella vostra rete, cercate quello corrispondente allo Smart gate lock, e annotate l'indirizzo IP che vi servirà per accedere alla pagina web del dispositivo.

Siccome le impostazioni sono uguali a quelle dei moduli IoT pubblicati in passato, andremo solo a elencarle. 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 pushbutton" 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".

In **Fig. 6** e **Fig. 7** trovate alcuni schemi esemplificativi di collegamento dello "Smart gate lock" che si applicano alla descrizione dell'utilizzo che stiamo facendo.

IMPOSTAZIONI DI COMANDO

Per funzionare con l'assistente vocale, lo Smart gate lock 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 "Smart lock". Ricordiamo 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. 8**).

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 dello Smart gate lock alla sezione "IoT credential", rispettivamente nei campi "Api Key", "Api Secrets" e "Device Id" (**Fig. 5** a destra). 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 aggiungete un nuovo, nella schermata successiva scegliete come dispositivo "altro" e successivamente cliccate sul tasto "trova dispositivi", quindi attendete la rilevazione dello Smart gate lock. Completate la configurazione del vostro sistema seguendo le istruzioni a schermo.

CONTROLLO VOCALE

Se sul vostro smartphone installate l'App Google Home, lo "Smart gate lock" 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 "Cancello".

Per attivare il relé pronunciate:

"Alexa, apri il cancello" oppure "Hey Google, apri il cancello"

Per disattivare il relé pronunciate:

"Alexa, chiudi il cancello" oppure "Hey Google, chiudi il cancello"

Alla richiesta di apertura del cancello, l'assistente vocale vi chiederà il PIN che avete memorizzato nelle impostazioni del dispositivo e se corretto, in chiusura il PIN non sarà più richiesto. La richiesta del codice PIN per le serrature elettroniche IoT è implementata di default in tutti gli assistenti vocali come sicurezza nel caso smarriste lo smartphone. Nell'App Amazon Alexa la funzione di sblocco delle serrature è disattivata per impostazione predefinita,

Fig. 12
Contenitori
e frutti
stampati
in 3D per
placche
Vimar, Gewiss
e BTicino.



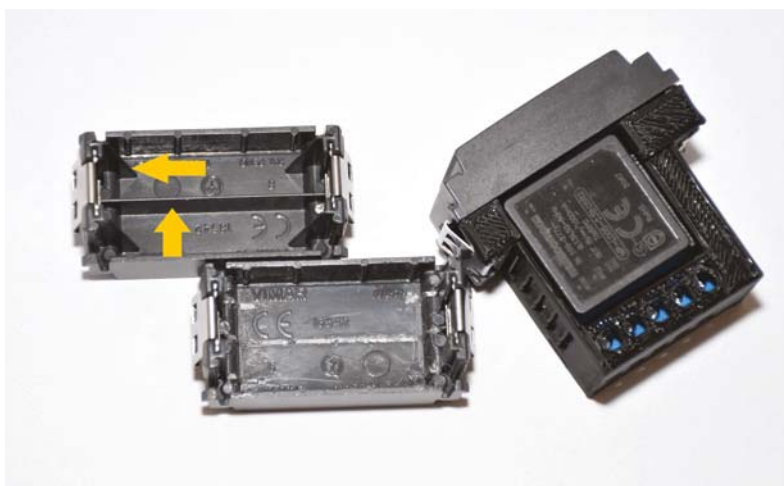


Fig. 13
Le frecce indicano i segmenti di plastica da eliminare dal copriforo Vimar Idea.

quindi bisogna attivarla manualmente e impostare il codice PIN cliccando su “Modifica codice vocale” come visibile in **Fig. 8**.
Come alternativa al controllo di Amazon e Google, potete impiegare l'app Sinric Pro, della quale la **Fig. 11** propone l'interfaccia di controllo.
Poiché l'apertura e la chiusura del cancello scorrevole o della sbarra condominiale del parcheggio avviene mediante un breve comando da un pulsante o dal radiocomando, nelle impostazioni del dispositivo impostate il Timer a un secondo.
Così una volta attivato il relé, con un comando vocale, questo dopo un secondo si disecciterà, simulando l'impulso di apertura. Lo stesso vale se il dispositivo dovesse aprire il cancelletto o la porta d'ingresso del condominio. La serratura elettromeccanica viene sbloccata da una bobina e se il relé rimanesse eccitato troppo a lungo rischierebbe di surriscaldarla e danneggiarla. In caso di commutazione di carichi induttivi o capacitivi ricordatevi di montare lo snubber RC formato dal condensatore C11 e dalla resistenza R11.

INTEGRAZIONE NELL'IMPIANTO ELETTRICO

Tra i file scaricabili dal sito web www.elettronica.in.it, nella cartella STL, trovate i file per la stampa 3D dei contenitori e dei frutti per i supporti Vimar Idea, Vimar Plana, Gewiss System, Gewiss Playbus e BTicino living, visibile in **Fig. 12**, che faciliteranno l'installazione del “Smart gate lock” nel vostro impianto elettrico domestico. Facciamo presente che l'adattatore Vimar Idea viene montato su un copri foro originale al quale devono essere asportati precedentemente i segmenti di rinforzo come visibile in **Fig. 13**.

Aiutatevi con le tronchesine per tagliare i reofori dei componenti oppure un paio di pinze. Fate attenzione al corretto orientamento dei coperchi dei contenitori, in modo che il forellino sia posizionato sopra al pulsante SW1, per poterlo schiacciare.
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 www.futurashop.it/index.php?route=stampa3d/stampa3d e chiedere un preventivo.

CONCLUSIONI

Con questo siamo arrivati al termine, sperando di aver soddisfatto i desideri di chi, tra voi, necessita di un dispositivo apricancello 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 “Smart gate lock” 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>

Costruisci e programma un ROBOT!



cod. KSR21
€ 42,00

Robot RICCIO - in kit

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



cod. KSR17
€ 29,90

Robot solare e idraulico 12 in 1 - in kit

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

Robot TOBBIE II con MICRO:BIT in kit

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



cod. KSR20
€ 89,00



Robot TOBBIE II in kit

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

cod. KSR18
€ 37,00



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



CONOSCERE E USARE

di FRANCESCO FICILI

Node-RED

5

Iniziamo a scoprire 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 ci occupiamo dell'implementazione delle funzionalità di connettività usando Node-RED, spaziando dal TCP all'UDP, fino ad arrivare all'HTTP. Quinta Puntata.

Come abbiamo visto sin dall'inizio del corso, Node-RED ha una spiccata propensione per le applicazioni in cui sia richiesto di implementare connettività di vario tipo. Essendo nato come un tool per lo sviluppo di applicazioni IoT, integra di default una serie di features che permettono di implementare in maniera molto semplice vari tipi di connessioni, usando diversi protocolli e standard di comunicazione. Questo aspetto è molto importante per le applicazioni IoT e, più in generale per le applicazioni connesse, in quanto è necessario che i nostri dispositivi interagiscano con l'hardware (ad esempio ricevano un dato da un sensore o comandino uno specifico attuatore), ma è altrettanto importante che tali dispositivi dialoghino tra di loro, con un server esterno, o con un dispositivo portatile come uno smartphone o un tablet.

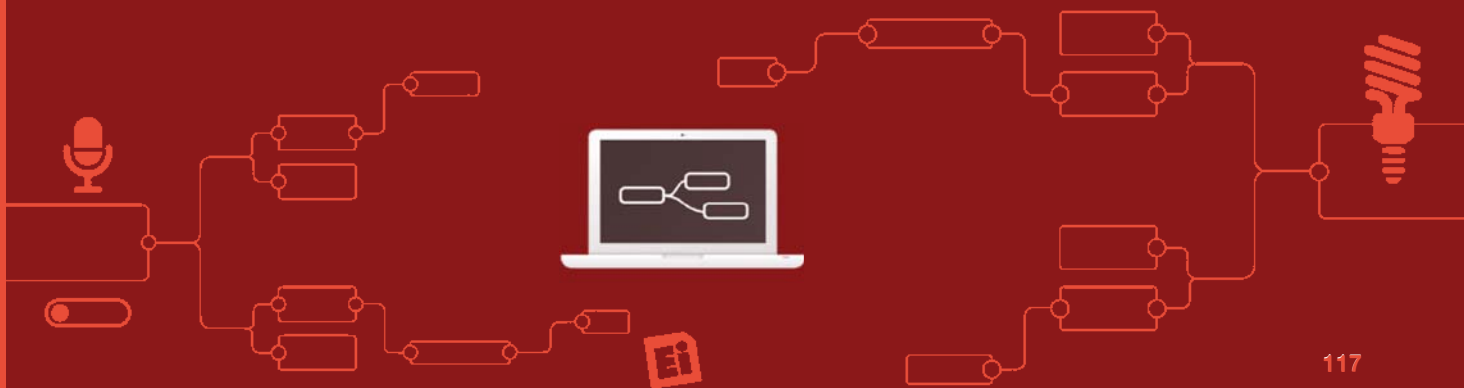
CONNECTIVITY IN NODE-RED

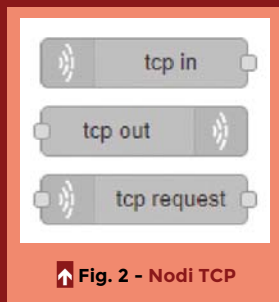
Come anticipato in precedenza, Node-RED dispone di una quantità di nodi dedicati alla connettività. Molti di questi nodi sono pre-installati e raggruppati nella sezione "Network". Tra questi troviamo:

- UDP
- TCP
- HTTP
- Websocket
- MQTT

Inoltre, sempre nella sezione Network, troviamo i nodi Serial, che abbiamo già analizzato nella terza puntata del corso.

La Fig. 1 riporta i nodi della sezione Network. In questa puntata ci concentreremo maggiormente su TCP, UDP e HTTP, lasciando la trattazione dell'MQTT e dei Websocket alla puntata successiva.





TCP

Il TCP, acronimo di Transmission Control Protocol, è un protocollo di trasmissione dati per reti informatiche ideato inizialmente nel 1973 da **Robert E. Kahn** e **Vinton G. Cerf**, durante un lavoro di ricerca. Questa idea iniziale di protocollo di trasmissione dati è stata migliorata ed espansa nel corso degli anni successivi, fino ad arrivare alla versione odierna che costituisce una delle colonne portanti della cosiddetta "famiglia di protocolli internet".

Il TCP è un protocollo di livello trasporto che permette a due nodi terminali di una rete di trasmissione dati di realizzare una connessione attraverso la quale può avvenire uno

scambio bidirezionale di informazione. Il TCP è in grado di riconoscere, nell'ambito di questo interscambio di informazione, perdite di dati, ed è pertanto definito un protocollo affidabile. Essendo posizionato al livello 4 della pila OSI (trasporto) il TCP deve necessariamente appoggiarsi su un protocollo di rete, e nella stragrande maggioranza dei casi questo protocollo è il protocollo IP, tanto che spesso si parla della combinazione dei due come di standard TCP/IP. A sua volta il protocollo TCP è spesso utilizzato dagli strati superiori della pila OSI come protocollo di trasporto per la realizzazione di protocolli di livello applicazione (ad esempio come nel caso dell'HTTP). Sebbene quest'ultimo sia il caso d'uso tipico, il protocollo TCP può essere anche utilizzato direttamente per realizzare trasmissione di dati affidabile tra due dispositivi, sebbene la grande proliferazione di protocolli di livello applicativo, estremamente standardizzati, specializzati e di semplice utilizzo e implementazione, renda questo caso sempre più raro.

Node-RED mette a disposizione una serie di nodi per la gestione del protocollo TCP, rappresentati in Fig. 2.

Sono presenti i seguenti nodi:

tcp in: questo nodo permette di ricevere dati tramite collegamento TCP. Il nodo consente sia di connettersi su una porta di un host remoto oppure di accettare connessioni in entrata su una specifica porta. I dati ricevuti sono inoltrati in output sul *msg.payload*.

tcp out: questo nodo permette di inviare dati tramite collegamento TCP. Il nodo consente sia di connettersi su una porta di un host remoto, di accettare connessioni in entrata su una specifica porta o di rispondere ad una richiesta TCP su una connessione esistente. Solo i dati presenti su *msg.payload* sono inviati in uscita.

tcp request: questo nodo implementa una semplice richiesta TCP, quindi esegue in sequenza le seguenti operazioni:

- Effettua la connessione
- Invia la richiesta
- Legge la risposta

Risulta molto utile per implementare sessioni di comunicazione, riducendo il numero di nodi da utilizzare. L'output può essere restituito in vari modi: aspettando la ricezione di un carattere speciale, dopo un timeout predefinito, dopo la ricezione di un determinato numero di caratteri, immediatamente, oppure si può scegliere di mantenere la connessione aperta non ritornando mai nulla in uscita.

Si noti che, di default, il nodo restituisce un array sul *msg.payload*, quindi per visualizzare l'output su una stringa è necessario una ulteriore elaborazione (ad esempio utilizzare il metodo *toString()* in un function node).

Vediamo un semplice esempio di utilizzo dei nodi TCP per la comunicazione tra dispositivi. Quello che ci proponiamo di realizzare è una semplice comunicazione bidirezionale su rete locale (ma sarebbe possibile eseguirla anche su rete internet) tra due nodi.

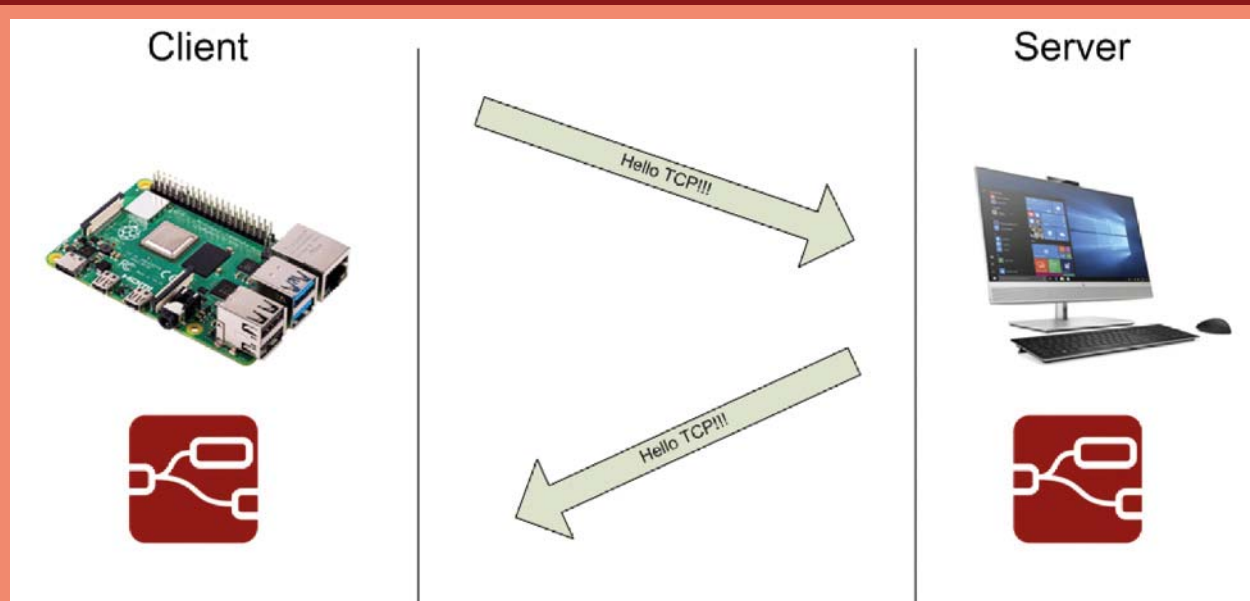


Fig. 3 - Setup utilizzato per l'esempio TCP.

Il nodo A effettuerà la connessione al nodo B (che funge quindi da server) su una specifica porta, ed invierà una stringa. Il nodo B, in ascolto sulla porta concordata, riceverà la stringa e risponderà con un eco (ossia ritrasmettendo la stessa stringa al nodo che ha effettuato la 'richiesta'). Si capisce subito che per implementare questo semplice esempio ci occorrono due dispositivi. Il primo dispositivo è la Raspberry Pi che abbiamo già utilizzato durante il resto del corso, mentre il secondo può essere un'altra Raspberry Pi (o dispositivo similare, in grado di far girare una istanza di Node-RED), oppure un PC. Come sappiamo infatti Node-RED può girare su vari tipi di macchine, inclusi PC windows. Questo è esattamente il setup che abbiamo utilizzato noi, usando una Raspberry Pi ed un PC windows, setup che risulta molto utile per effettuare test in cui è richiesta la comunicazione tra dispositivi, e che è anche rappresentato in **Fig. 3**. Per i dettagli su come installare e far girare Node-RED su windows potete fare riferimento alla guida ufficiale presente sul sito di Node-RED: <https://nodered.org/docs/getting-started/windows>.

Partiamo dall'implementazione del nodo client, che posizioniamo sulla Raspberry Pi e che ha il compito di inviare la richiesta (inviare la stringa "Hello TCP!!!") e restare in attesa della risposta (eco da parte del server).

Per effettuare la richiesta ci serviremo del nodo tcp request visto in precedenza.

Sia lato client che lato server useremo un carattere di terminazione (`\n`, newline) per indicare che la trasmissione dati è completa, quindi, dato che il nodo inject non permette nativamente di inserire caratteri speciali sulle stringhe, useremo un nodo function e il metodo `concat()` per aggiungere una nuova linea alla stringa.

Inoltre useremo anche un nodo function con il metodo `toString()` per convertire l'array ritornato in output dal nodo tcp request.

Per implementare questo flow ci occorrono quindi:

- Un nodo Inject
- Due nodi Function
- Un nodo Tcp Request
- Un nodo debug



Fig. 4 - Struttura del flow lato client.

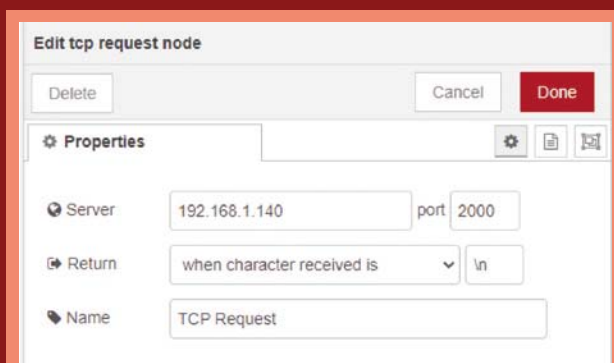


Fig. 5 - Configurazione del nodo TCP request.

La struttura del flow è rappresentata in **Fig. 4**. Vediamo in dettaglio la configurazione dei nodi. Il nodo Inject inietta nel flow semplicemente la stringa "Hello TCP!!!", mentre il nodo debug non ha configurazione. I nodi concat e toString implementano sostanzialmente dei metodi javascript che operano su `msg.payload`, per eseguire delle manipolazioni sulla stringa contenuta nell'oggetto JS. In particolare:

Nodo Function Concat: serve a concatenare il carattere speciale Newline (`\n`), tramite il seguente snippet di codice javascript:

```
msg.payload = msg.payload.concat('\n');
```

Nodo Function toString: serve a convertire l'output del nodo TCP request, che di default è un array, in una stringa, tramite il seguente snippet di codice javascript:

```
msg.payload = msg.payload.toString();
```

Infine vediamo come è configurato il nodo TCP request, avvalendoci della **Fig. 5**.

Il nodo TCP request richiede all'utente di fornire l'indirizzo dell'host (in questo caso l'indirizzo di rete locale 192.168.1.140, che rappresenta l'indirizzo IP del PC windows sul quale gira il nostro "echo server") e la porta (2000). Si noti che è stata usata la porta 2000 in quanto, su molti sistemi, l'accesso alle porte da 0 a 1024 è limitato. Infine il nodo richiede di configurare come debba essere ritornata la risposta: come indicato in precedenza, scegliamo di aspettare il carattere speciale newline. Passiamo ora al flow del server, implementato sul PC windows e rappresentato in **Fig. 6**.

Come si può vedere si tratta di un flow molto semplice, che fa uso di:

- Un nodo TCP in
- Un nodo TCP out
- Un nodo Function

Come è facile evincere, i nodi TCP in e TCP out servono, rispettivamente, per ricevere la richiesta ed inviare la risposta, mentre il nodo Function Concat ha la stessa funzione vista in precedenza, ossia concatenare il carattere newline. Vediamo in dettaglio la configurazione dei noti TCP (in quanto il nodo function è identico all'implementazione vista in precedenza), partendo dal nodo TCP in, la cui finestra delle proprietà è rappresentata in **Fig. 7**.

Il nodo TCP sta sostanzialmente in ascolto sulla porta 2000 (listen on port 2000) per connessioni in ingresso e fornisce in output uno stream in formato stringa (Output: stream of String), delimitato dal carattere speciale `\n` (delimited by `\n`). Ancora più semplice la configurazione del nodo TCP output, che deve sostanzialmente rispondere su una connessione TCP aperta, e quindi l'unica configurazione da fare è settare la proprietà Type come "Reply to TCP", come illustrato in **Fig. 8**.

I nostri flow lato client e lato server sono ora pronti e possiamo semplicemente testare questa implementazione iniettando alcune stringhe nel flow ed esaminando ciò che viene ricevuto sulla finestra di debug (**Fig. 9**), ossia l'eco della stringa inviata.

UDP

Dopo aver esaminato in dettaglio come implementare connessioni TCP in Node-RED possiamo ad esaminare l'UDP.



Fig. 6 - Flow del TCP server.

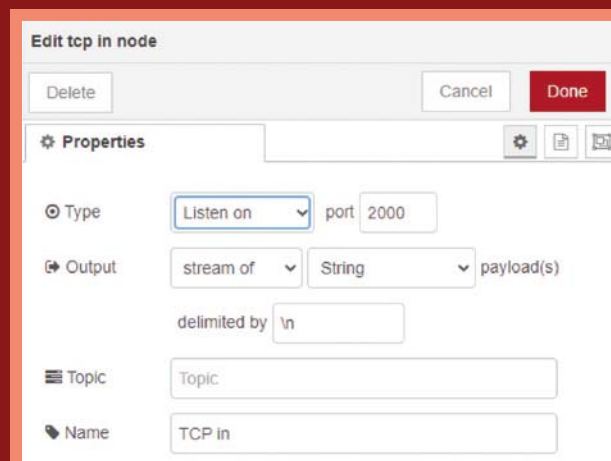


Fig. 7 - Configurazione del nodo TCP in.

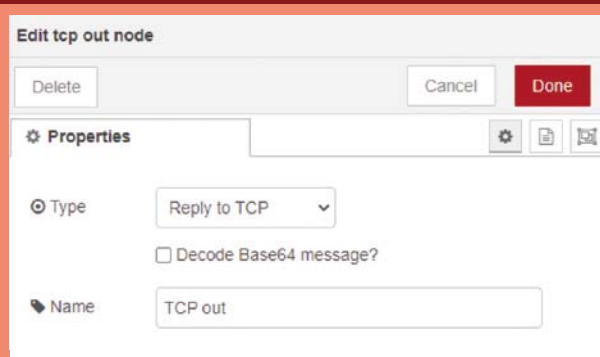


Fig. 8 - Configurazione del nodo TCP out.

L'UDP (User Datagram Protocol), è un altro protocollo di livello trasporto, a pacchetto, tipicamente usato in combinazione con un livello di rete IP. La caratteristica base dell'UDP, che lo differenzia dal suo parente prossimo, ossia il TCP, è quella di essere connectionless. In sostanza, mentre una trasmissione TCP ha luogo solo dopo una fase obbligatoria di autenticazione e la creazione di una connessione tra i due nodi che intendono scambiarsi dati, l'UDP rinuncia a questa caratteristica in funzione di una minore latenza nello scambio dati e quindi una maggiore velocità di trasmissione. Il fatto di essere connectionless semplifica notevolmente il

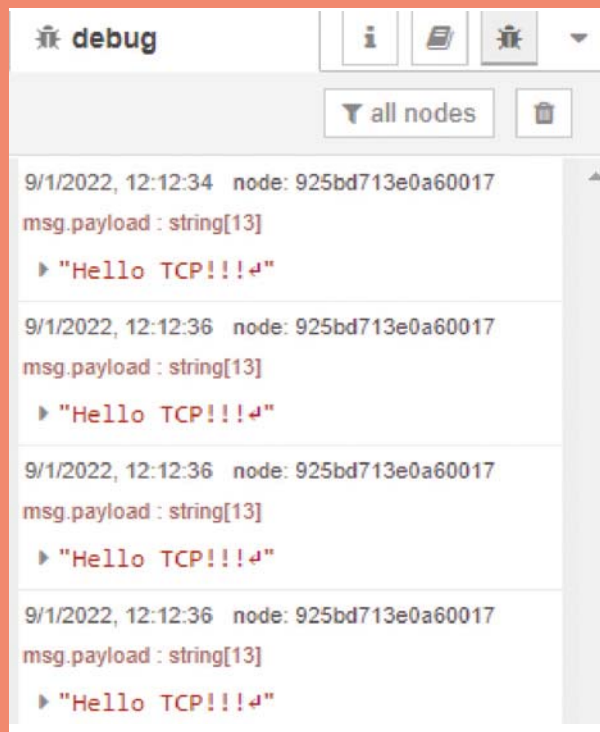


Fig. 9 - Finestra di debug (lato client) che mostra l'eco ricevuto dal server.

protocollo, che di fatto risulta essere una alternativa snella e con latenze ridotte al minimo del TCP. Il risvolto della medaglia è che l'UDP è estremamente meno affidabile del TCP, infatti non c'è garanzia di consegna dei pacchetti, non viene gestita automaticamente la ritrasmissione dei pacchetti persi e non è garantita la sequenzializzazione. Tutte caratteristiche che invece sono presenti nel TCP, che è sicuramente più lento, ma anche estremamente più affidabile. Di conseguenza applicazioni tipiche in cui si fa largo uso del protocollo UDP sono quelle in cui la perdita di dati è tollerabile in luogo di un maggior rate di trasmissione. Esempi tipici sono lo streaming audio/video, le richieste DNS e le connessioni VPN.

La gestione dell'UDP in Node-RED è realizzata con due nodi, riportati in **Fig. 10**.

udp in: questo nodo permette di ricevere dati tramite collegamento UDP. Il nodo invia in output su *msg.payload* il buffer di trasmissione ricevuto sulla porta specificata. E' supportato sia IPv4 che IPv6, come anche le trasmissioni multicast. L'output può essere formattato sia come buffer di dati che come stringa.

udp out: questo nodo permette di inviare dati tramite collegamento UDP. Il nodo permette di indirizzare uno specifico host su una data porta. Anche in questo caso è supportato sia IPv4 che IPv6.

Anche per l'UDP possiamo realizzare un semplicissimo esempio, con un setup identico a quello utilizzato per il TCP. In questo caso è leggermente improprio parlare di client e server, dal momento che non viene instaurata una connessione, ma abbiamo, di fatto, due nodi della rete (diciamo A e B) che comunicano bidirezionalmente.

Diciamo anche che il nodo A è il nodo che inizia la comunicazione, mentre il nodo B è quello che implementa la funzionalità di eco dei dati.

L'implementazione in questo caso è molto semplificata. Partiamo dal nodo A, che nel nostro specifico caso è sulla macchina Windows (ma le parti potrebbero essere tranquillamente invertite). Per implementare la comunicazione UDP secondo le nostre specifiche iniziali, dobbiamo realizzare un flow che includa:

- Un nodo inject
- Un nodo debug
- Un nodo UDP out
- Un nodo UDP in

Il flow che implementa il nodo A è riportato in **Fig. 11**.



Fig. 10 - Nodi UDP di Node-RED.

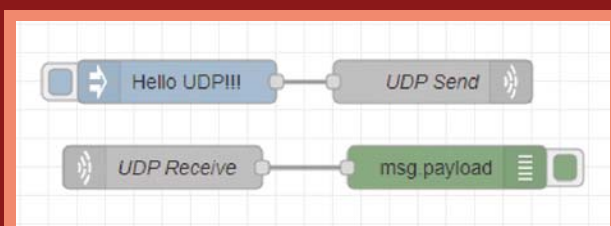


Fig. 11 - Flow che implementa il nodo A.

Vediamo il dettaglio della configurazione dei due nodi UDP, partendo dal nodo UDP send, la cui finestra delle proprietà è riportata in Fig. 12.

Come si può facilmente notare inseriamo la porta sulla quale vogliamo inviare i dati (3000 in questo caso) e l'indirizzo IP del nodo destinazione (ricordatevi sempre di adattarlo a quello del dispositivo destinazione nella vostra rete locale). Il resto delle configurazioni possono essere mantenute ai valori di default.

Passiamo adesso alla configurazione del nodo UDP Receive, illustrata in Fig. 13.

In questo caso ci basta specificare la porta sulla quale ricevere i dati e il tipo di output, che vogliamo sia una stringa. A questo punto ci possiamo spostare sul nodo B, il cui flow, semplicissimo, è rappresentato in Fig. 14.

Le configurazioni dei due nodi UDP Receive e UDP Send, sono sostanzialmente identiche a quelle viste per il nodo A. L'unica differenza è il campo Address del nodo Send, che, ovviamente, questa volta deve contenere l'indirizzo IP del nodo A.

L'implementazione dei nostri flow è ora completa e possiamo passare al test. Se tutto è stato fatto correttamente, ogni volta che inviamo una stringa dal nodo A ne riceveremo l'eco da parte del nodo B, su collegamento UDP.

La Fig. 15 mostra l'output ottenuto nella finestra di debug di Node-RED.

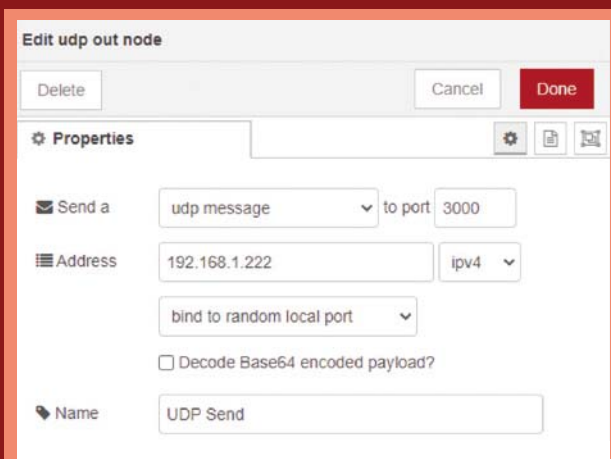


Fig. 12 - Finestra delle proprietà del nodo UDP send.

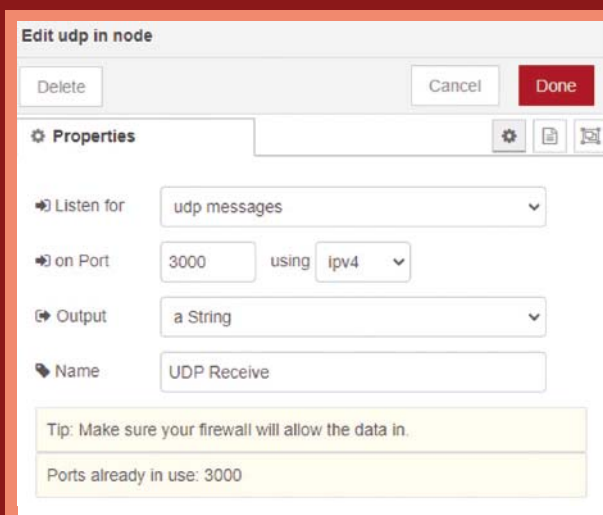


Fig. 13 - Finestra delle proprietà del nodo UDP receive.

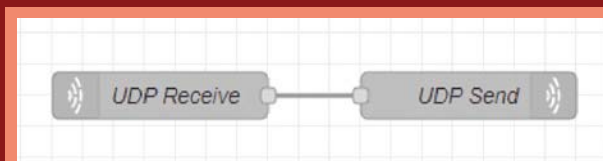


Fig. 14 - Flow del nodo B.

HTTP

L'acronimo HTTP sta per "HyperText Transfer Protocol", e sta ad indicare un protocollo sviluppato negli anni '80 da Tim Berners-Lee come una delle tre componenti costituenti il nucleo base del World Wide Web, che rappresenta uno dei principali servizi offerti da internet, e di cui tutti noi facciamo ampio uso ormai da più di due decenni.

Gli altri due componenti del nucleo base sono il linguaggio HTML e gli URL. Gli URL sono le stringhe che rappresentano i domini dei siti web, e l'HTML è il linguaggio che viene utilizzato per creare i contenuti delle pagine dei siti.

L'HTTP invece, è il protocollo che governa il trasferimento dei dati tra un client (ossia un web browser) e il server che ospita il sito che vogliamo visualizzare. Il protocollo HTTP è quindi un protocollo client-server e si colloca al livello applicazione della pila OSI.

L'HTTP è quindi generalmente utilizzato per permettere la comunicazione client-server nel World Wide Web, ossia per far comunicare i browser con i server.

Il suo funzionamento è molto semplice:

- il client esegue una richiesta di una certa risorsa (ad esempio una pagina di un sito web).
- Una volta ricevuta la richiesta il server cerca il contenuto desiderato (ad esempio la homepage del sito, index.html) ed invia prima un header con un codice di stato che serve ad informare il client dell'esito della richiesta.

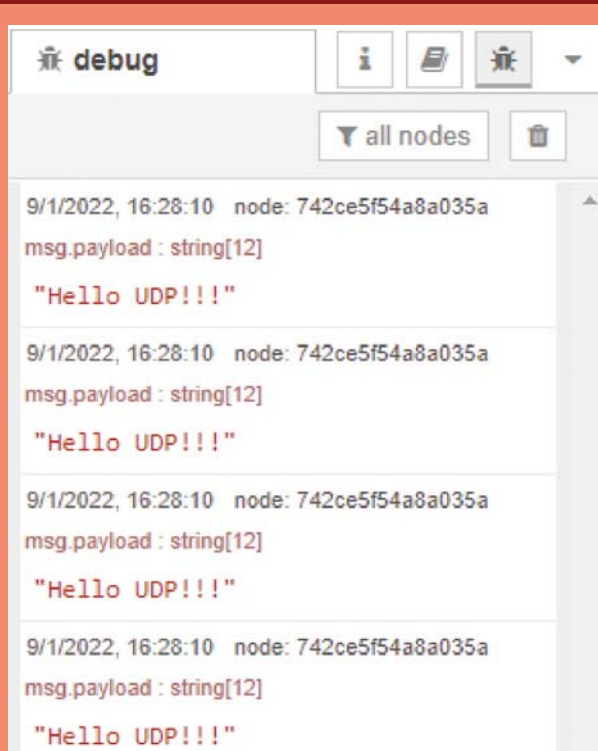


Fig. 15 - Output della finestra di debug del nodo A.

- Se il contenuto viene trovato, il server lo invia al client nel body del messaggio
- Il client riceve il dato e, se si tratta di un browser, lo visualizza come pagina web

C'è da notare che non sempre l'HTTP è utilizzato per la comunicazione tra un browser e un server che ospita un sito web; infatti si può utilizzare l'HTTP per realizzare comunicazione machine-to-machine, o per l'implementazione di servizi web come ad esempio API REST, o anche per accedere ad un database web. Node-RED fornisce 3 nodi per la gestione del protocollo http (Fig. 16):

http in: permette di creare un http endpoint per la gestione

delle richieste standard previste dal protocollo http (GET, POST, PUT, DELETE, PATCH), quindi in sostanza permette di implementare un webservice. Il nodo richiede anche il path locale della richiesta (campo URL).

http response: permette di inviare risposte alle richieste ricevute da un nodo http in.

http request: questo nodo implementa il lato client del protocollo http, quindi permette di inviare richieste ad una specifica URL e ritorna indietro le risposte ricevute dal server.

ESEMPIO PRATICO: CONTROLLO DI UN RELÈ TRAMITE HTTP

Concludiamo questa puntata dedicata alla connettività con il classico esempio pratico. Ciò che ci proponiamo stavolta è di implementare un semplice webserver che ci permetta di azionare un relè da remoto, usando una richiesta standard http. Per implementare questo webserver ci occorrono i seguenti nodi:

- 2 Nodi http in
- 1 Nodo http response
- 2 Nodi change
- 1 Nodo rpi-gpio out

Ciò che vogliamo fare è intercettare due differenti richieste http GET, riferite ai path interni fasulli /relay_on e /relay_off del nostro server.

A queste due richieste forniremo comunque risposta positiva e sfrutteremo i due flussi per triggerare 2 change node che andranno a comandare il nodo rpi-gpio out con i valori corretti per attivare e disattivare il relè. Il flow risultante è visibile in Fig. 17:

Analizziamo la configurazione dei Nodi http, aiutandoci con la finestra delle proprietà del nodo Relay ON (Fig. 18).

Come si può facilmente notare, abbiamo utilizzato un metodo http GET e la risorsa che vogliamo indirizzare è il path interno /relay_on (l'URL del server sarà l'indirizzo IP della raspberry).

Questo nodo verrà triggerato quando verrà effettuata una richiesta http su quel path specifico. Il nodo Relay OFF è del tutto simile, cambierà solamente il path, che sarà /relay_off.

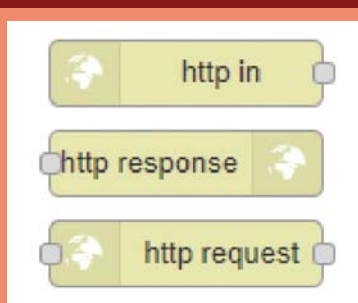


Fig. 16 - Nodi Node-RED per la gestione del protocollo HTTP.

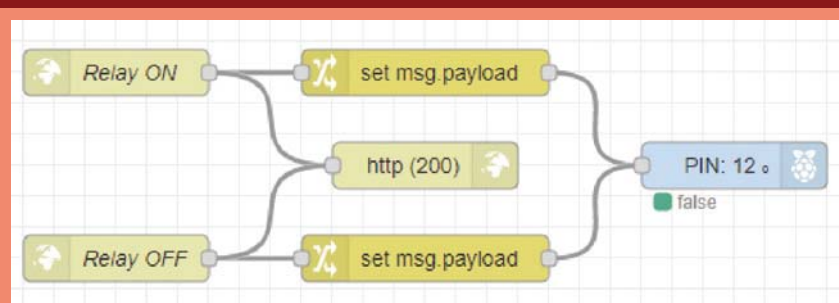


Fig. 17 - Flow che implementa il webserver per il controllo di un relè tramite http.

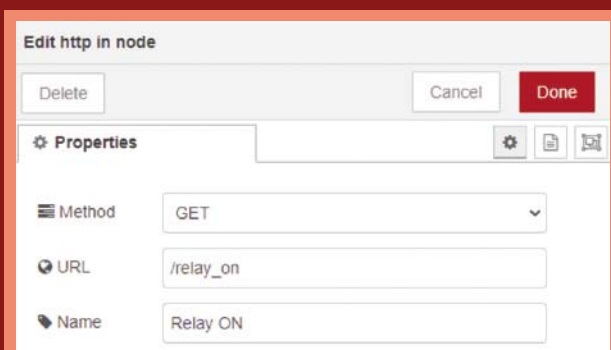


Fig. 18 - Proprietà del nodo Relay ON (http in).



Fig. 20 - HAT RPIRELEBOARD.

Alla richiesta è opportuno fornire una risposta, che nel nostro caso sarà sempre positiva, in quanto non abbiamo la reale possibilità che si verifichi un errore, dovendo controllare semplicemente un pin (controlli più sofisticati, in applicazioni reali sono comunque possibili).

Usiamo quindi un nodo http response in cui configuriamo lo status code come 200 (risposta positiva del server standardizzata in http), come si può vedere dalla Fig. 19, che rappresenta la finestra delle proprietà del nodo.

L'output dei due nodi http in, che viene triggerato ogni volta che viene fatta una richiesta GET dei path /relay_on e /relay_off, non è adatto a pilotare direttamente il nodo rpi-gpio out, quindi utilizziamo due nodi change per modificare opportunamente il contenuto del msg.payload.

Il nodo change che manipola il flusso del ramo relay_on setta a true il valore di msg.payload, mentre l'altro lo setta a false. Entrambi gli output dei due nodi vanno in input al nodo rpi-gpio out, che comanda il canale corrispondente al relè che vogliamo attivare. Per testare il nostro flow possiamo usare l'HAT RPIRELEBOARD, un HAT che monta 3 relè a 250 VAC/5 A e 30 VDC / 5 A, commercializzato da Futura Elettronica, una cui immagine è visibile in Fig. 20.

Per effettuare le richieste possiamo usare un qualsiasi browser, collegandoci ai seguenti indirizzi:

<IP Raspberry>:1880\relay_on → per attivare il relè

<IP Raspberry>:1880\relay_off → per disattivare il relè

In alternativa è possibile utilizzare delle app android che permettono di effettuare richieste http tramite shortcuts su una interfaccia grafica.

Una app di questo tipo molto funzionale e completamente gratuita è http shortcuts, che permette di creare shortcuts con nome, descrizione ed icone associate.

La creazione delle shortcuts è molto intuitiva ed un esempio di pannello che si riesce a realizzare è riportato in Fig. 21.

CONCLUSIONI

Siamo arrivati alla conclusione di questa puntata in cui abbiamo visto come gestire la connettività in Node-RED usando i protocolli TCP, UDP e HTTP. Queste caratteristiche ci consentono diverse possibilità implementative, garantendoci di poter gestire la comunicazione tra i dispositivi con diverse modalità. Nella prossima puntata introdurremo altri due standard molto utilizzati in ambito IoT: l'MQTT e i WebSocket.

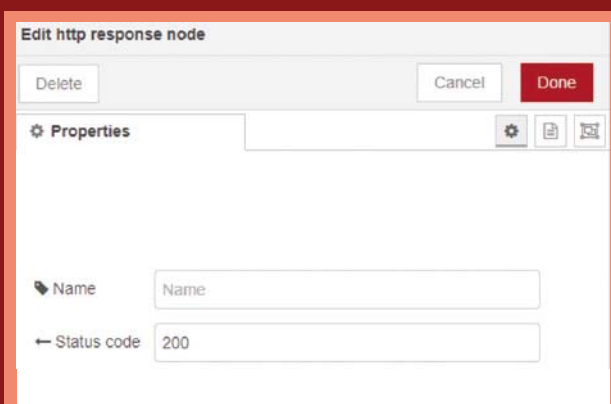


Fig. 19 - Finestra delle proprietà del nodo http response.

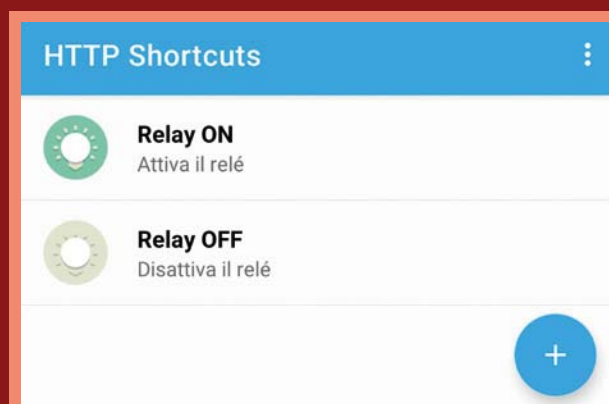


Fig. 21 - Pannello per la gestione del relè realizzato tramite HTTP shortcuts.

UN TEAM DI PROFESSIONISTI PER IL TUO PROGETTO



oltre **1500** progetti sviluppati
26 anni di attività
più di **15** progettisti



L'area R&D di Futura Group studia, progetta e realizza prototipi per differenti settori utilizzando tecnologie innovative. Nel team di Futura Group fa parte il personale interno e un numero elevato di collaboratori esterni, ciascuno specializzato nel proprio settore (Disegno CAD – Raspberry Pi – Arduino – Audio – Sviluppo firmware – Linux – ecc). La disponibilità di oltre 5000 prodotti elettronici e componenti consente di ridurre i tempi di test e sviluppo e di realizzare facilmente il primo prototipo. La decennale esperienza nel settore e la necessità di ottimizzare i tempi e i costi, ha portato Futura Group a selezionare una serie di terzisti per la produzione di piccola o grande serie.

I SERVIZI CHE TI OFFRIAMO



Progetto Completo

Partendo dalla tua idea progettiamo per te la soluzione.



Sviluppo PCB

Ottimizziamo il tuo prototipo disegnando un PCB compatto secondo le tue specifiche.



Scrittura Codice

Scriviamo il codice del core per dare vita alla tua applicazione.



Produzione Progetto

Portiamo il prototipo alla fase di produzione seguendo per te tutte le fasi.

Il robot magazziniere sposta 800 scatole all'ora

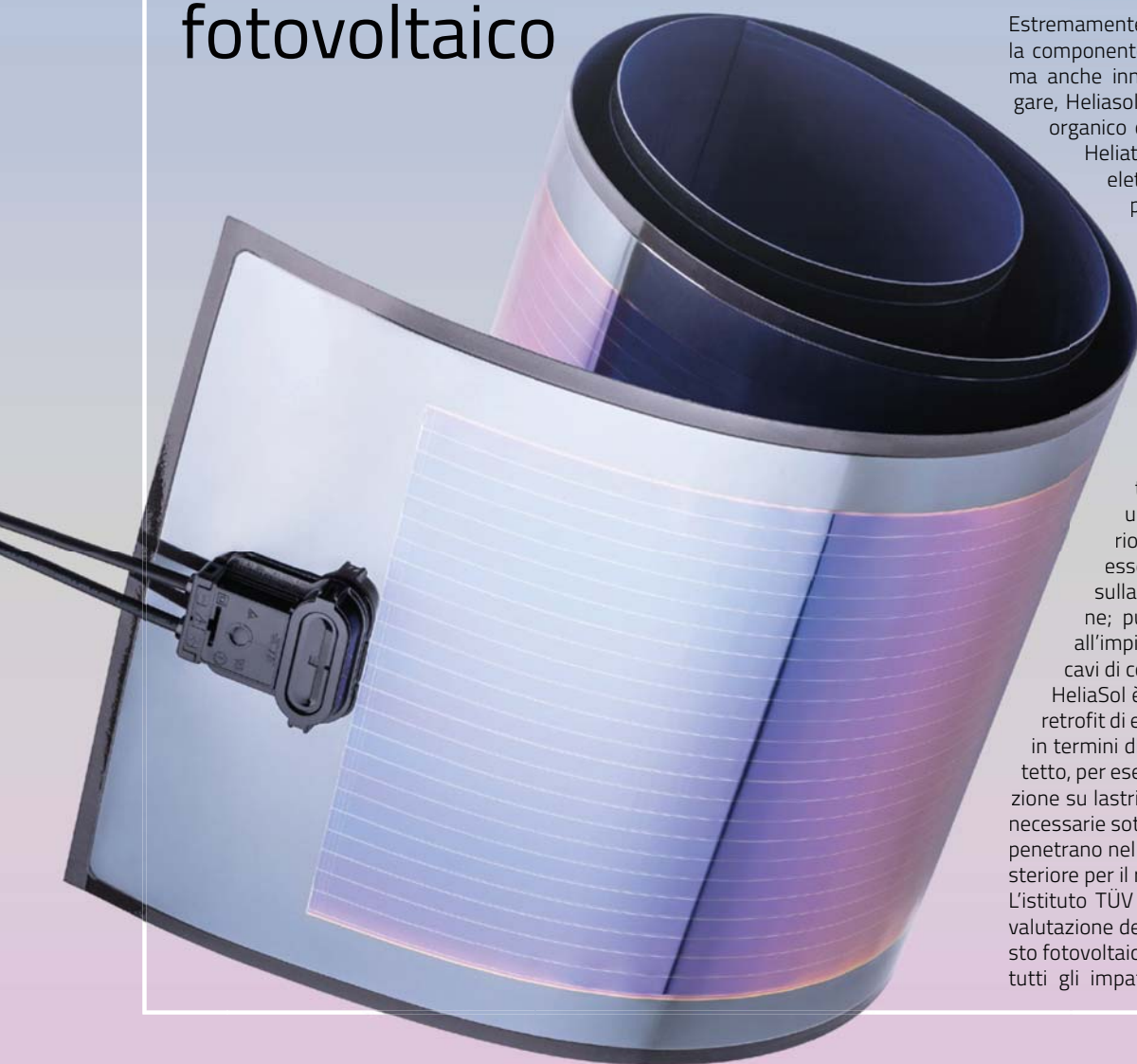
Siccome il COVID-19 mette a dura prova le catene di approvvigionamento globali, la logistica sta spingendo sull'automazione per mettere in sicurezza i lavoratori e aumentarne l'efficienza; ma molte operazioni di magazzino non si prestano all'automazione tradizionale, perché gli input e gli output di un processo non sono sempre ben definiti e non possono essere completamente controllati. Ecco perché si sta testando una nuova generazione di robot con l'intelligenza e la flessibilità per svolgere vari lavori tipici

degli umani: per esempio Stretch, della Boston Dynamics, capace di spostare scatole pesanti alla stessa velocità di un magazziniere esperto, afferrandole mediante ventose. Diversamente dai robot umanoidi e quadrupedi che hanno distinto Boston Dynamics, qui abbiamo un braccio massiccio, una pinza piena di sensori e una serie di ventose, e una base mobile omnidirezionale, che può trasferire scatole che pesano fino a 23 chili dal vano di carico di un camion a un nastro trasportatore, al ritmo di 800



l'ora. Un lavoratore umano esperto può spostare le scatole a una velocità simile, ma non tutto il giorno, mentre Stretch

La pellicola organica trasforma tutto in fotovoltaico



Estremamente sottile (meno di 2mm per la componente utile) flessibile e leggera, ma anche innovativa e facile da impiegare, Heliasol è un pannello fotovoltaico organico creato dall'azienda tedesca Heliatek, nato per creare energia elettrica dal sole dove i normali pannelli non sono applicabili, quindi per rivestire edifici, piloni, infrastrutture di vario genere, pensiline, tanto da poter essere applicata, ad esempio, anche al pilone di sostegno di un generatore eolico, realizzando così un generatore elettrico combinato eolico-fotovoltaico. La pellicola solare ha un adesivo nella parte inferiore, il che significa che può essere facilmente incollata sulla superficie di destinazione; può essere subito collegata all'impianto grazie ai connettori e cavi di collegamento integrati.

Heliasol è la soluzione perfetta per il retrofit di edifici in cui giocano un ruolo in termini di peso, il carico statico di un tetto, per esempio; proprio per l'installazione su lastrico solare o tetto, non sono necessarie sottostrutture di appoggio che penetrano nel tetto né la ventilazione posteriore per il raffreddamento.

L'istituto TÜV Rheinland ha certificato la valutazione del ciclo di vita (LCA) per questo fotovoltaico a film sottile, analizzando tutti gli impatti ambientali, dall'approv-

può lavorare 16 ore continuative. Lo scarico dei camion è uno dei lavori più difficili in un magazzino, perché servono due persone, mentre con Stretch è possibile avere una sola persona che scarichi quattro o cinque camion contemporaneamente. Tutto ciò che serve a Stretch è addestrarlo a identificare il vano di carico di un rimorchio pieno di scatole; l'uomo servirà solo per assicurarsi che tutto vada liscio e potrà intervenire se Stretch si imbatte in qualcosa che non può gestire.

Stretch segue gli standard per i robot industriali mobili stabiliti dall'American National Standards Institute e dalla Robotics Industry Association.

www.bostondynamics.com

vigionamento delle materie prime fino allo smaltimento. Heliasol è un pannello simile a quelli a film sottile per flessibilità ed economia, ma non richiede l'impiego di metalli pesanti come il cadmio o il piombo, il che facilita la gestione dei rifiuti a fine vita. Il risultato è un'impronta di carbonio estremamente bassa: 14,5 g di CO₂ per mq, grazie anche al fatto che per produrre un metro quadrato di pellicola fotovoltaica è sufficiente appena 1 g di materiale organico. Heliasol può essere incollato su vari materiali di substrato, come metallo, cemento, membrane, vetro, bitume e materie plastiche (anche plexiglass). L'estrema leggerezza (meno di 2 kg per metro quadrato) facilitano l'applicazione a qualsiasi superficie o copertura. Testato con successo in oltre 30 progetti pilota in tutto il mondo, con circa 2.000 metri quadrati di superficie installata, Heliasol inizialmente verrà offerto in pannelli delle dimensioni di 436x2.000 mm, ma l'azienda conta, in futuro, di rendere disponibili altri tagli. Heliatek non si fermerà qui, perché entro il 2023 metterà sul mercato Heliaplan, una pellicola fotovoltaica adesiva ancora più flessibile e ideale per rivestire vetrate e facciate di edifici.

www.heliatek.com

Heliatek®
Say hello to solar. Wherever you are

Batteria litio-zolfo: 5 volte più autonomia per l'auto elettrica

Sebbene sia già avviato, il mercato dei veicoli elettrici aspetta una spinta che deve arrivare dall'incremento dell'autonomia di marcia; e potrebbe arrivare dalla batteria litio-zolfo, secondo una ricerca condotta all'Università del Michigan (USA) mirata a risolvere uno dei problemi chiave dell'accoppiata litio-zolfo.

Questa tecnologia offre una capacità di accumulo cinque volte superiore a quelle delle tradizionali ricaricabili a ioni di litio; la maggiore densità energetica e i costi ridotti dovuti all'impiego dello zolfo al posto del litio, ne farebbero la soluzione ideale, se non fosse per il loro difetto che le rende inapplicabili: il bassissimo numero di cicli di carica/scarica consentito, causato dalla progressiva perdita di materiale attivo dall'elettrodo (il cosiddetto effetto shuffle del polisolfuro di litio) che ne determina una breve vita utile. Negli anni sono state tentate varie strade per aumentare la durata, ma ciascuna a scapito di capacità, velocità di carica, resilienza e sicurezza.

La soluzione adottata dagli scienziati della Michigan University consiste in una nuova membrana ispirata alla struttura fibrosa della cartilagine e composta da una rete di nanofibre aramidiche, riciclate

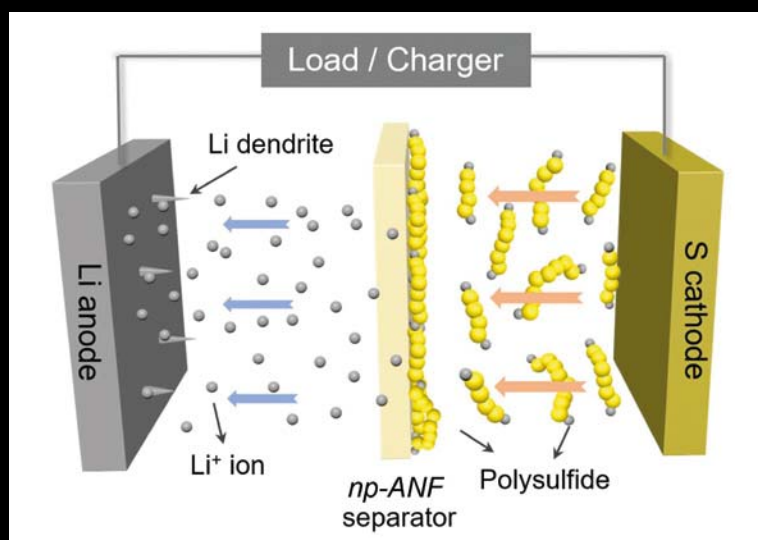
dal Kevlar. La membrana consente il passaggio degli ioni tra gli elettrodi ma blocca le particelle di polisolfuro di litio responsabili dell'accorciamento della vita utile.

Ispirandosi ai canali ionici biologici, i ricercatori hanno progettato una sorta di autostrade per gli ioni di litio in cui i polisolfuri di litio non possono superare i margini. Per le batterie al litio-zolfo ad alta energia, si preferisce un elettrodo denso con bassa porosità: le prestazioni ottimali si verificano con il 50 per cento di porosità.

Secondo i ricercatori si tratta di un design quasi perfetto e la batteria bio-ispirata ha mostrato una densità di capacità vicina al massimo teorico di 1.268 mAh/g e una vita utile di oltre 3.500 cicli di carica/scarica. Il dispositivo riesce anche ad operare alle temperature estreme tipiche dell'ambiente di utilizzo automobilistico, dal calore della ricarica in pieno sole al freddo dell'inverno.

Tuttavia, la vita nel mondo reale potrebbe essere più breve con la ricarica rapida, avvicinandosi ai 1.000 cicli, che comunque rappresentano una durata pratica di dieci anni.

researchgroups.msu.edu

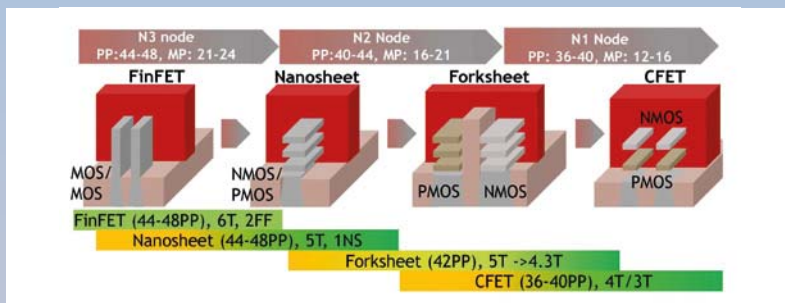


Forksheet: la chiave per il sub-2 nm

Intel ha depositato un processo produttivo chiamato "transistor Forksheet impilati" per costruire transistor sub-2nm, che dovrebbe riguardare un'architettura CMOS 3D impilata verticalmente in grado di consentire un numero maggiore di elementi rispetto alla tecnologia triple-gate-transistor più avanzata. Intel descrive l'impiego di transistor Nanoribbon e film sottili di germanio che agirebbero come separatori dielettrici tra gli strati di transistor impilati. Forksheet è un'evoluzione dei processi Nanosheet che fanno un significativo passo in avanti rispetto ai transistor FinFET, attualmente utilizzati

da TSMC per le sue produzioni di punta, a 4-5 nm mentre, sia Intel che Samsung, stanno utilizzando rispettivamente i processi RibbonFET e GAA (Gate-All-Around), due varianti della tecnologia Nanosheet. Le principali differenze tra Forksheet e Nanosheet sono la formazione della parete dielettrica e il distanziatore interno modificato, l'epitassia source-drain. L'evoluzione dovrebbe portare un aumento del 10% della velocità, del 24% dell'efficienza energetica a velocità operativa costante, con una riduzione del 20% in superficie.

www.intel.com



PoliMOVE vince l'Autonomous Challenge

Neanche ci siamo abituati alla Formula E, la competizione ai massimi livelli che di fatto è la Formula 1 delle auto elettriche, ed ecco apparire la Formula ...autonoma! Una competizione tra auto da corsa, ma prive di conducente. In questo contesto è un piacere vedere che il primo classificato di questa competizione, chiamata Autonomous Challenge e disputata nell'ambito del CES Show di Las Vegas, è stato il team italiano PoliMOVE, del Politecnico di Milano, aggiudicandosi oltre al primo posto, il succulento premio di 150.000 dollari. Alla competizione hanno partecipato vetture Dallara AV-21, le più avanzate auto da corsa a guida autonoma esistenti, guidate

dai software di intelligenza artificiale sviluppati da 9 team di 19 università di tutto il mondo. Le auto si sono sfidate in una serie di round successivi al termine dei quali la vettura di PoliMOVE ha fatto segnare anche il record di velocità sfrecciando a oltre 278 km/h: la massima velocità mai raggiunta da un veicolo a guida autonoma su un circuito ovale.

La competizione, organizzata dall'Energy Systems Network che ha lanciato anche la Indy Autonomous Challenge, è nata per incentivare l'evoluzione tecnologico nei veicoli a guida autonoma.

www.polimi.it



Un robot, 4 zampe e un braccio!

Unitree Robotics, startup cinese che si è fatta conoscere per l'offerta di robot su zampe a prezzi molto accessibili, per il mercato consumer (Go1 e Ben Ben, per esempio) ma anche industriale (B1 e Aliengo) ha annunciato un braccio robotico a quattro segmenti con 6 gradi di libertà destinato ad essere montato sul retro del suo più grande robot quadrupede. Per propagandarne le qualità, ha diffuso un video su YouTube dove il robot brandisce, con il braccio, una Light Saber, la fantascientifica spada laser di Guerre Stellari: un modo per evidenziare la scioltezza e rapidità di movimento del prodotto. Il braccio è disponibile nelle versioni Z1 Air e Z1 Pro, costruite con motoriduttori armonici per un gioco ridotto e il controllo della coppia; sono simili, ma Pro pesa 2 etti di più ed ha un carico utile di 3 ÷ 5 kg anziché 2 kg. La portata massima è di 0,7 metri, con una ripetibilità dei movimenti di 0,1 millimetri. Il braccio ha una velocità massima di 180 gradi/s. e utilizza il sistema operativo Ubuntu. La potenza elettrica è circa 500 W e sono supportati l'autoapprendimento e il rilevamento delle collisioni.

www.unitree.com



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

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