

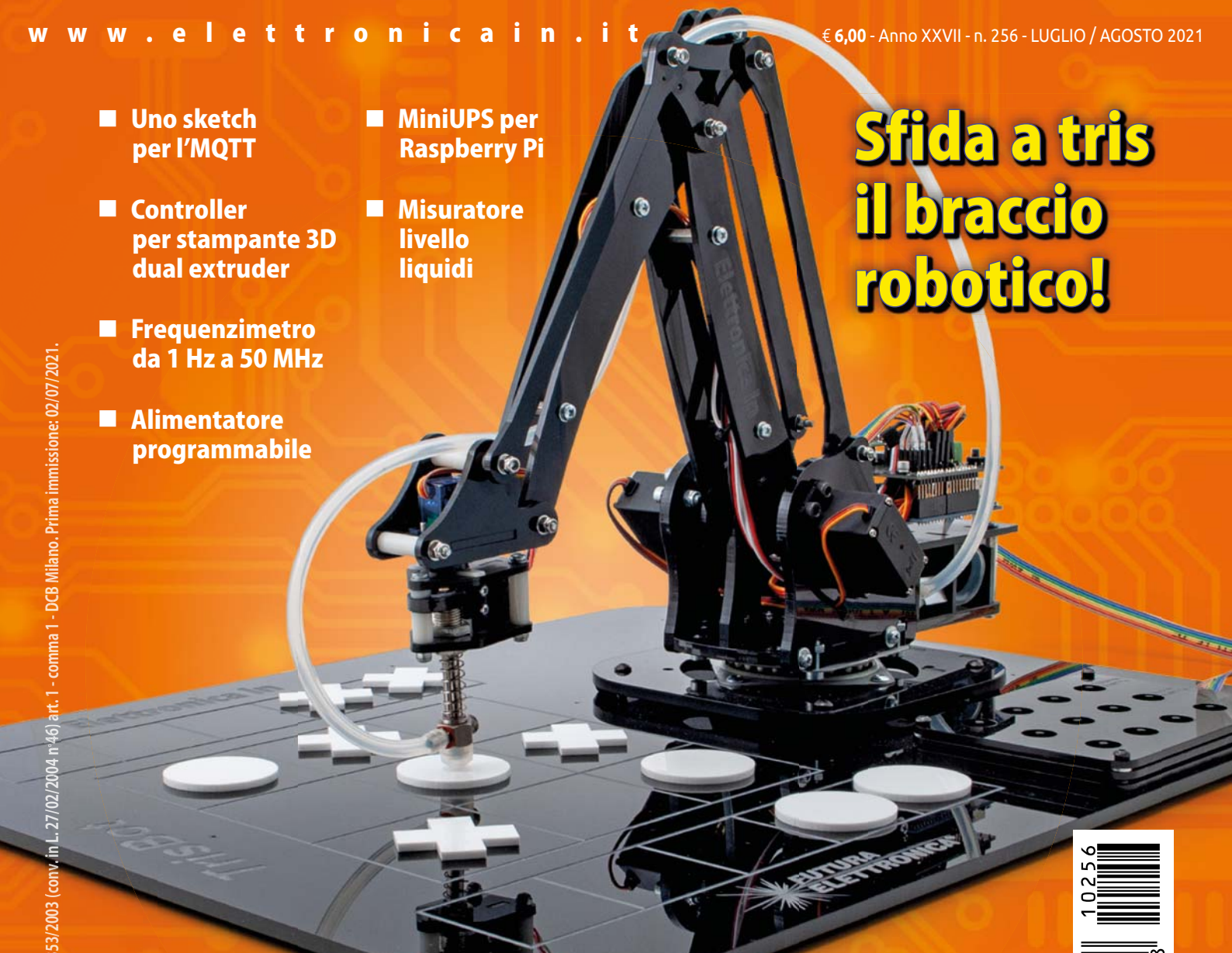
Elettronica In

www.elettronica.in.it

€ 6,00 - Anno XXVII - n. 256 - LUGLIO / AGOSTO 2021

- Uno sketch per l'MQTT
- MiniUPS per Raspberry Pi
- Controller per stampante 3D dual extruder
- Misuratore livello liquidi
- Frequenzimetro da 1 Hz a 50 MHz
- Alimentatore programmabile

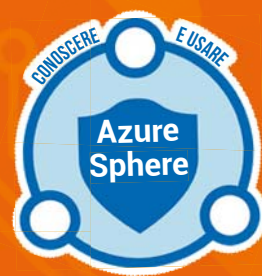
Sfida a tris il braccio robotico!



Controlla la cisterna con Telegram



Interruttore crepuscolare



Per lo smart working, per chattare con amici e parenti comodamente da casa!

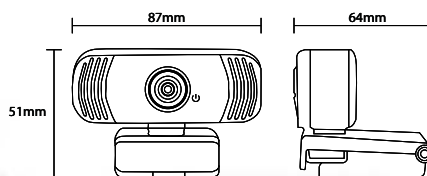
Prezzo IVA inclusa.



€24,00
Cod. FR795

WEBCAM FULL HD 1080P

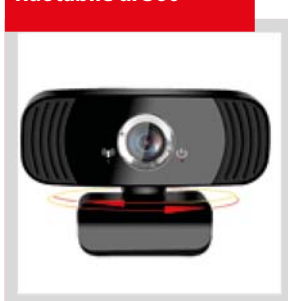
Webcam Full HD 1080P particolarmente indicata per seguire lezioni online, effettuare videoconferenze, videochat, videosorveglianza, registrazioni video, ecc. Dispone di lenti prive di distorsione, microfono omnidirezionale con riduzione del rumore, regolazione manuale della messa a fuoco, angolo di ripresa: 60° (verticale) – 120° (orizzontale), rotazione: 360°, LED di stato per alimentazione e attivazione webcam. Completamente Plug & Play (non richiede l'installazione di nessun driver), può funzionare con Windows 10/8/7/ Vista, Mac OS X, Linux, ChromeOS, Ubuntu e Android 4.0 o versioni successive. Basta collegarla alla porta USB del PC, attendere che il sistema operativo la rilevi e poi utilizzarla con Skype, Zoom, Facebook, ecc. Grazie alla sua base multifunzionale è possibile posizionarla e utilizzarla su PC fissi, PC portatili, smart TV, appoggiarla sulla scrivania oppure montarla su treppiede per macchine fotografiche.



- ✓ **Processore:** 3518EV300
- ✓ **Sensore immagine:** CMOS a colori
- ✓ **Risoluzione:** 1920 x 1080
- ✓ **Video stream:**
 - H.264: 1920 x 1080 / 1920 x 720 / 640 x 360
 - MJPEG: 1920 x 1080 / 1920 x 720 / 640 x 360
 - YUV: 640 x 360
- ✓ **Video frame rate (fps):** 30 fps (1280 x 720 / 1920 x 1080)
- ✓ **Uscita Video:** USB Video streaming
- ✓ **Controllo Immagine:**
 - messa a fuoco manuale
 - bilanciamento del bianco automatico
 - controllo automatico dell'esposizione (AEC, Automatic Exposure Control)
- ✓ **Otturatore elettronico:** automatico
- ✓ **Microfono:** omnidirezionale
- ✓ **Rapporto segnale/rumore:** > 50dB
- ✓ **Angolo di ripresa:** 60° (verticale) 120° (orizzontale)
- ✓ **Rotazione:** 360°
- ✓ **Sistemi operativi supportati:** Windows 10/8/7/Vista, Mac OS X, Linux, ChromeOS, Ubuntu e Android 4.0 o versioni successive
- ✓ **Alimentazione:** 5VDC / 160mA (tramite porta USB)
- ✓ **Temperatura di lavoro:** da -40°C a +70°C
- ✓ **Materiale:** ABS
- ✓ **Dimensioni (mm):**
 - Webcam: 87 x 64 x 51mm
 - lunghezza del cavo USB: 1,5m
- ✓ **Peso:** 180g



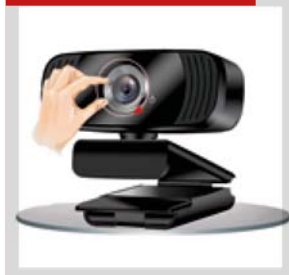
Ruotabile di 360°



Vari metodi di posizionamento



Supporta la messa a fuoco manuale





“Rifiutare” l'elettronica

Sono molte le giornate celebrative per sensibilizzare le persone a fare qualcosa per salvare il nostro pianeta. Vengono proposti molti (o forse è meglio dire i soliti) rimedi, che vanno dal riciclare e dare nuova vita ai rifiuti, al seguire determinati comportamenti per risparmiare risorse, energia ecc.

Tra gli scarti più difficili da smaltire c'è sicuramente l'e-waste, ovvero il rifiuto elettronico. I prodotti elettronici sono composti da centinaia di materiali diversi, e spesso troviamo anche cadmio,

mercurio, arsenico, antimonio e molti altri elementi dannosi per l'ambiente. Eppure anche per gli e-waste esistono delle soluzioni per evitare che queste sostanze vengano assorbite dalla nostra

Capire come funziona un circuito elettronico vi permetterà anche di ripararlo.

amata Terra. Tra la più semplice vi è quella di concedere una nuova vita allo “scarto”: ad esempio un telefono che consideriamo “vecchio” in realtà può essere utilizzato come riproduttore MP3, un PC può non essere più adatto per il nostro lavoro, ma può essere un'ottima console di gioco per i più piccoli. Quindi regalando o vendendo un dispositivo elettronico evitiamo di farlo finire nella spazzatura.

Ma cosa fare se il nostro dispositivo è guasto? La risposta più semplice è “ripariamolo”. Ed è qui che Elettronica In può essere d'aiuto. La nostra “mission” è spiegare l'elettronica e dare soluzioni innovative. Facciamo questo analizzando nel dettaglio i circuiti più semplici e dando nuove soluzioni utili per imparare l'elettronica. Capire come funziona un circuito elettronico vi permetterà anche di ripararlo. Non abbiamo l'ambizione di insegnarvi a riparare un microscopico circuito di una scheda madre, ma di darvi le nozioni per utilizzare gli strumenti giusti e magari riparare la radiosveglia che ha smesso di funzionare. Imparate e riparate, non esiste un pianeta B, almeno per il momento...



In copertina:
Un braccio robotico didattico in veste ludica. Impariamo la cinematica e la programmazione divertendoci.

Boris Landoni
✉ boris.landoni@elettronica.in

ON LINE

✉ elettronica.in

SI RINGRAZIA

Futura Elettronica
Mouser

Elettronica In

Rivista mensile, anno XXVII n. 256

LUGLIO / AGOSTO 2021

EDITORIALE

Direttore Responsabile

Boris Landoni

✉ boris.landoni@elettronica.in

Redazione

Stefano Garavaglia, Paolo Gaspari,
Davide Scullino, Gabriele Daghetta

✉ redazione@elettronica.in

Grafica

Alessia Sfulcini

✉ alessia.sfulcini@elettronica.in

Ufficio Pubblicità

Monica Premoli (0331-752668)

✉ monica.premoli@elettronica.in

Ufficio Abbonamenti

✉ Tel. 0331-752668

✉ abbonati@elettronica.in

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

✉ Tel. 0331-752668

Abbonamenti

Annuo 10 numeri Euro 45,00

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/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



Mensile
associato all'USPI,
Unione Stampa
Periodica Italiana

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

Prezzo di copertina Euro 6,00.

Gli arretrati nei formati cartaceo e digitale (pdf) sono acquistabili
sul sito della rivista al prezzo di Euro 6,00.

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/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.

© 2021 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

Sfida a tris il braccio robotico!

58



Didattica

Uno sketch per l'MQTT



27

Automazione

Interruttore crepuscolare

43



19

ARTICOLI

PRIMI PASSI

19 Misuratore livello liquidi

Visualizza su un display a LED il livello rilevato da un sensore di presenza di liquidi conduttivi, come l'acqua di rubinetto. Utile in pratica, è anche l'occasione per sperimentare con l'elettronica.

DIDATTICA

27 Uno sketch per l'MQTT

Inviando dati a ThingSpeak tramite rete mobile utilizzando il protocollo MQTT, la nostra libreria per Arduino e la shield GSM/GPRS.

AUTOMAZIONE

43 Interruttore crepuscolare

Permette di accendere e spegnere l'illuminazione elettrica, o altri utilizzatori, in base alle condizioni di luce ambientale.

Primi passi

Misuratore livello liquidi



19

Mondo 3D**Controller per stampante 3D dual extruder**

49

MONDO 3D**49 Controller per stampante 3D dual extruder**

Stampiamo a due colori in 3D fino a 40x40x40 cm: montiamo il secondo estrusore e configuriamo il client di stampa. Seconda ed ultima puntata.

ROBOTICA**58 Sfida a tris il braccio robotico!**

Un braccio robotico controllato da Arduino UNO e munito di sistema "Pick and Place" gioca a tris muovendo pedine fisiche su un campo da gioco e non perde mai.

STRUMENTAZIONE**67 Il mio frequenzimetro? In kit**

Montiamo e testiamo un compatto frequenzimetro utilizzabile sia nelle misure in campo che in laboratorio.

FISHINO**77 Controlla il livello con Telegram**

Fishino, un bot Telegram e un sensore a ultrasuoni permettono di monitorare il livello dell'acqua delle cisterne di accumulo dell'acqua piovana.

Strumentazione**Il mio frequenzimetro? in kit**

67

Didattica**Corso Azure Sphere**

115

Fishino**Controlla il livello con Telegram**

77

Raspberry Pi**MiniUPS: continuità per il lampone**

103

Strumentazione**Alimentatore da banco "retail"**

91

**STRUMENTAZIONE****91 Alimentatore da banco "retail"**

Testiamo un nuovo modulo alimentatore veramente versatile, in grado di erogare fino a 50V e 5A, dotato di protezioni e di ben 9 memorie per i settaggi più utilizzati.

RASPBERRY PI**103 MiniUPS: continuità per il lampone**

Soluzione per dotare il Raspberry Pi di funzionalità UPS con la possibilità di configurare il sistema in funzione della batteria LiPo utilizzata come riserva di backup. Prima puntata.

DIDATTICA**115 Conoscere e usare Azure Sphere**

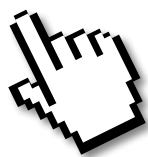
Utilizziamo il lettore RFID della nostra demoboard per creare un semplice database di tag per visualizzarne i dati in Azure IoT Central. Quinta puntata.

Un'offerta sempre più grande

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



mouser.it



MOUSER
ELECTRONICS



Da LED a fotodiiodo

Non tutti lo sanno, ma i LED e in generale i diodi a semiconduttore che hanno esposta (visibile dal contenitore) la giunzione, hanno un comportamento reversibile: prendiamo il caso dei comuni LED rossi, verdi e gialli con involucro in resina trasparente; ebbene, questi emettono luce se vengono polarizzati direttamente con una tensione che supera quella di soglia, però se vengono collegati in serie a un resistore di alto valore e polarizzati inversamente, colpendoli con la luce (magari quella di un laser a emissione rossa) manifestano ai loro capi una significativa differenza di potenziale. Analogamente, anche non in presenza di resistenza di carico, le giunzioni esposte alla luce producono una tensione e funzionano da generatori fotovoltaici, appunto, nel quadrante fotovoltaico, come fossero piccoli pannelli solari. Il grafico in questa pagina esplica il comportamento del diodo.

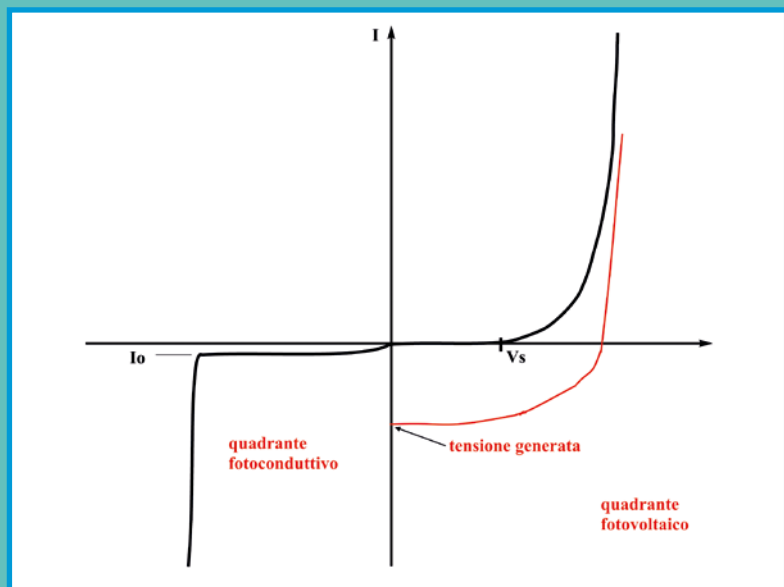
Tips & Tricks sono i nostri suggerimenti in pillole e i piccoli trucchi riguardanti sia l'hardware che il firmware, che sperimentatori, tecnici e progettisti elettronici vorrebbero conoscere per sviluppare meglio la propria passione: quella per l'Elettronica applicata. Ecco quelli che abbiamo preparato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

se le celle di Peltier quando sono percorse da corrente si scaldano da un lato e si raffreddano sulla faccia opposta, quando sottoposte a un salto termico producono tensione e perciò corrente elettrica. Sfruttare questa elettricità si può fare in vari modi: alimentando direttamente un utilizzatore, ma in

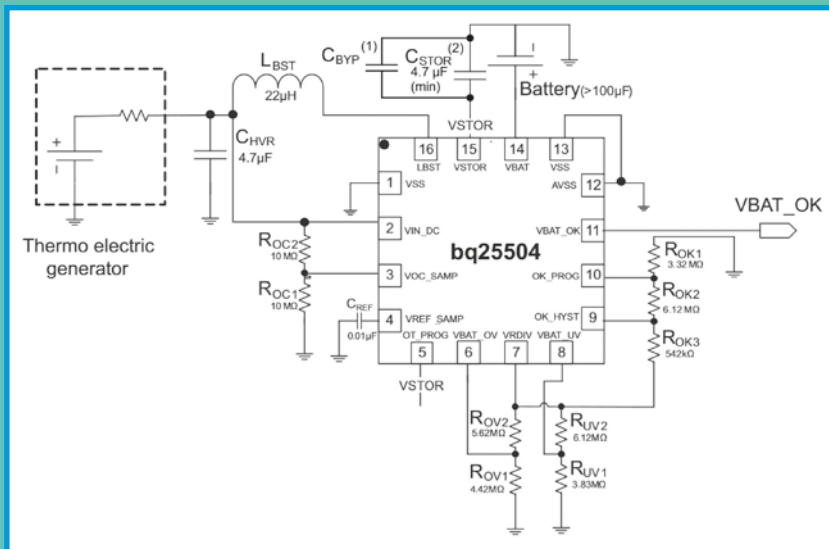
questo caso bisogna tenere conto del fatto che la massima energia trasferita si ha quando la resistenza elettrica del carico equivale quella interna del modulo termoelettrico, ovvero utilizzare un circuito elettronico in grado di presentare un carico quasi costante lato celle e caricare una batteria. Quest'ultima soluzione

Un po' di elettricità dal calore

Energy Harvesting è un termine che sta sempre più entrando nel nostro bagaglio tecnico e significa raccolta e recupero di energia altrimenti dispersa; nel dettaglio vuol dire andare a ricavare elettricità da fonti energetiche ambientali o di apparati e macchinari che naturalmente le disperderebbero: tipico è l'esempio del motore a benzina di un'automobile, che produce una gran quantità di calore da disperdere nell'ambiente tramite l'impianto di raffreddamento. Una delle forme di Energy Harvesting più quotate è quella che punta a produrre elettricità dal calore, sfruttando allo scopo i cosiddetti TEG (Thermo Electric Generator) o TEM (Thermo Electric Module) che sono praticamente celle di Peltier funzionanti alla rovescia, ovvero pile termoelettriche:



↑ Curva caratteristica del diodo: a sinistra il comportamento in polarizzazione inversa (fotodiiodo) e a destra quello da generatore fotovoltaico.



↑ Schema applicativo dell'integrato BQ25504 Texas Instruments.

permette anche di ovviare all'incostanza della fonte energetica termica, che può essere intermittente o variabile nel calore ceduto. Tra questi integrati, peraltro disponibili in commercio anche su breakout board, segnaliamo il BQ25504 della Texas Instruments, perché è un convertitore boost DC/DC ad alta efficienza, che integra uno stadio caricabatteria per singola cella agli ioni di litio, ma soprattutto perché grazie a una speciale circuitazione all'ingresso,

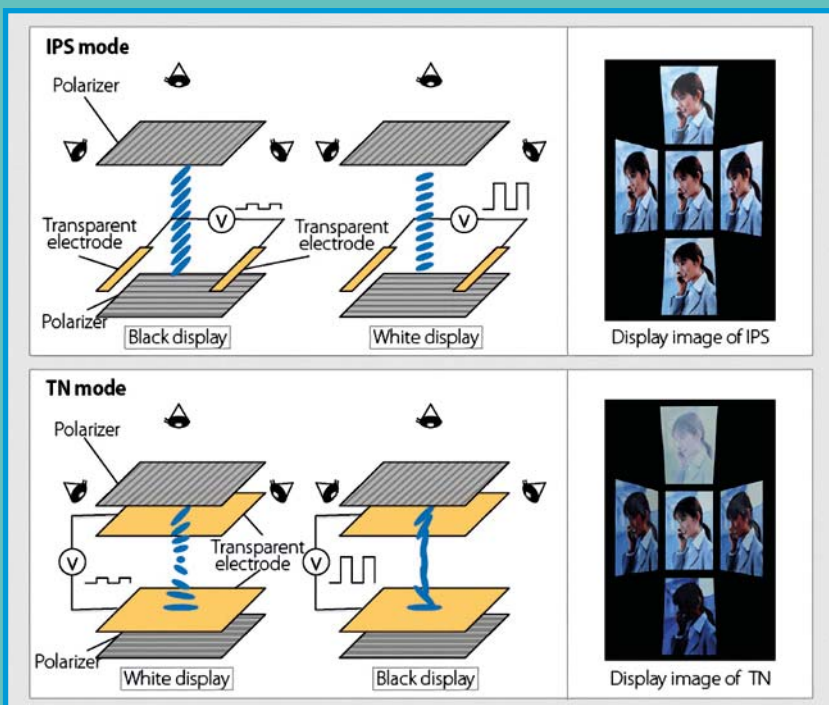
si accende con tensioni bassissime quali quelle fornite dai moduli termoelettrici e dalle celle di Peltier. L'integrato si avvia con una tensione minima di 600 mVcc e una volta che l'oscillatore del DC/DC converter è partito, rimane acceso con tensioni inferiori, fino a 30 mV (valore tipico); questo funzionamento si ottiene nella configurazione proposta dallo schema proposto in queste pagine, dove una rete LC retrocede parte della tensione generata dall'oscillatore verso

la cella di Peltier e quindi all'ingresso, così da creare una retroazione negativa che favorisce l'innesco dell'oscillatore stesso. Il chip implementa la regolazione MPPT (Maximum Power Point Tracking) dinamica, che permette di implementare il massimo trasferimento possibile di potenza tra la cella termoelettrica e il convertitore, variando, allo scopo, la resistenza di carico dinamica che esso costituisce; questo perché i moduli TEM, come anche i pannelli fotovoltaici e in generale i generatori elettrici, trasferiscono la massima potenza quando la loro resistenza interna eguaglia quella del carico.

Il BQ25504 è il primo di una nuova famiglia di soluzioni integrate intelligenti per applicazioni a bassissima potenza, come sensori e reti di sensori wireless (WSN) che hanno elevate esigenze di potenza e di funzionamento.

Display LCD IPS

L'ampliamento dell'angolo di visuale è stato uno dei principali obiettivi dei costruttori di pannelli LCD, che hanno sviluppato evoluzioni delle tecniche esistenti: il TN (Twisted Nematic) si è evoluto nel Super Twisted Nematic, con il quale comunque non si ottengono angoli di visione superiori a 150 gradi. Con il TFT si è arrivati a un angolo di visione di 160÷170° in orizzontale e circa 150° in verticale, ma ciò si paga con la relativa lentezza dell'aggiornamento dei pixel componenti l'immagine. Per coniugare i pregi del TFT e del TN è stato realizzato il display IPS (In Plane Switching): sviluppato dalla Hitachi nel 1996, inizialmente aveva tempi di risposta lunghi, che sono stati migliorati notevolmente nelle evoluzioni successive S-IPS (Super-IPS sviluppato dalla joint-venture LG- Philips) AS-IPS (Advanced Super IPS) e nell'ultima evoluzione chiamata H-IPS e utilizzata ad esempio da NEC, Mitsubishi ed Apple. La caratteristica principale dell'IPS è che consente di avere un angolo di visione praticamente uguale in orizzontale e in verticale, maggiore di 175°. Questa caratteristica viene ottenuta con una tecnologia costruttiva che permette al cristallo liquido di ruotare non sull'asse normale alla superficie dello schermo, ma sullo stesso piano di quest'ultimo (da qui il nome In Plane Switching). Tale caratteristica fa in modo che la luce, opportunamente polarizzata da pellicole applicate allo schermo, possa essere visibile bene anche da ampie angolazio-



↑ Funzionamento e visione del Twisted Nematic e dell'IPS a confronto.

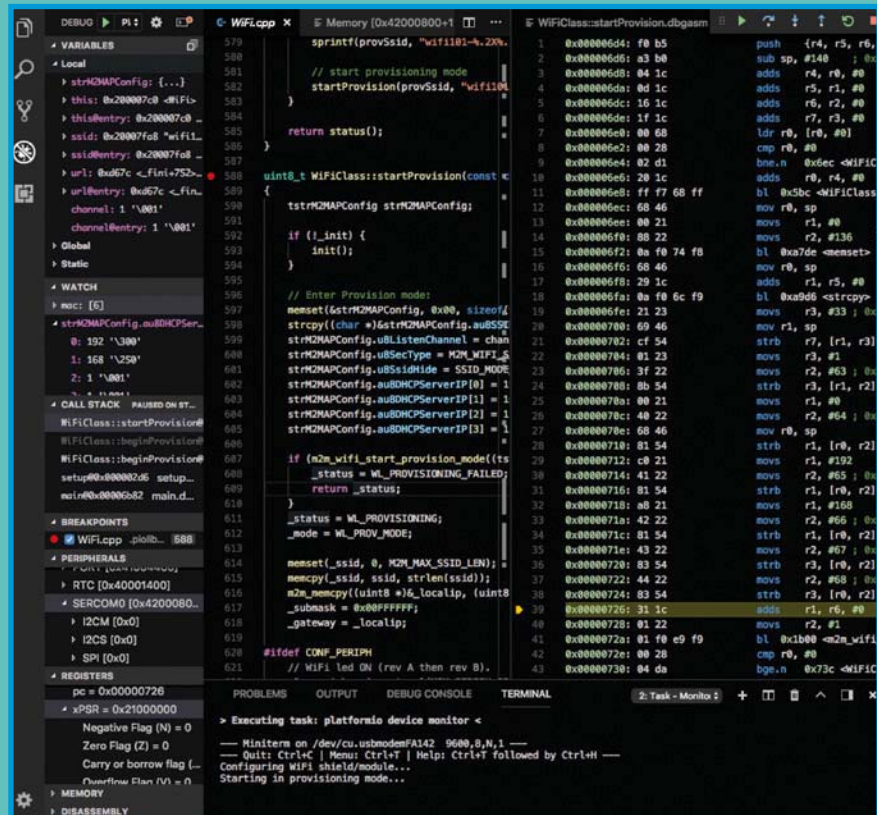
ni. Questa particolare struttura riduce la quantità di luce dispersa nella matrice, donando allo schermo IPS oltre agli ampi angoli di visione, una buona riproduzione dei colori. La tecnologia IPS è largamente impiegata nei pannelli dei display destinati alla grafica professionale. I pannelli IPS sono indicati per un utilizzo all'aperto grazie all'ottimo livello di retroilluminazione, con risultati di solito superiori anche all'OLED. Per quanto riguarda l'impiego nei tablet e negli smartphone, l'unica nota negativa è data dalla struttura del pannello e del touch screen, che in base alla tipologia possono soffrire del riflesso.

Una variante dell'IPS è la tecnologia Super PLS (Plane-to-Line Switching): nata in casa Samsung, migliora del 10% la brillantezza dell'immagine, costa meno da produrre e si applica anche ai display flessibili. I due tipi più comuni di LCD TFT sono gli schermi IPS e TN.

Lo schermo TN è noto per i suoi limiti nell'angolo di visione e per l'inversione dei colori ad angolazioni estreme.

Al contrario, i monitor IPS forniscono angoli di visuale più ampi e una riproduzione del colore di qualità nettamente migliore, rendendo i pixel paralleli, invece che perpendicolari. In un display IPS, i cristalli liquidi sono paralleli ai pannelli quando ricevono energia. In uno schermo TN, i cristalli si orientano in direzione perpendicolare rispetto alla parte superiore del pannello. I tablet e gli smartphone montano spesso schermi IPS perché sono sovente utilizzati per vedere filmati, fare chat video e archiviare foto. I professionisti della grafica traggono benefici dall'uso di uno schermo IPS perché un LCD IPS con una gamma di colori più vasta e angoli di visione più ampi consente di ottenere una precisione maggiore e di conseguenza una qualità eccellente. Le varianti della tecnologia IPS hanno le seguenti caratteristiche:

- Super TFT (IPS) che offre un angolo di visione più ampio;
- Super-IPS (S-IPS) che elimina l'effetto di cambiamento dei colori e aumenta la frequenza di refresh dei pixel e quindi la velocità di risposta;
- Advanced Super-IPS (AS-IPS) che offre una trasmittanza elevata della luce;
- Enhanced IPS (E-IPS) per avere un migliore angolo di visione diagonale e una riduzione del tempo di risposta;
- Horizontal IPS (H-IPS) che migliora il rapporto di contrasto attraverso una pellicola polarizzante Advanced True White che crea un colore bianco più naturale;



↑ L'interfaccia dell'IDE di PlatformIO eseguita in Microsoft Visual Studio Code.

- Professional IPS (P-IPS) che ha una notevole profondità di colore.

Programmare i micro con Visual Studio Code

Programmare i microcontrollori direttamente da PC utilizzando Microsoft Visual Studio Code: è questo che PlatformIO (www.platformio.com) consente di fare; si tratta di un'estensione della piattaforma che rende disponibile per lo sviluppo di codice uno specifico IDE integrato con Visual Studio Code. L'IDE è un ambiente di sviluppo integrato facile da usare ed estensibile, con una serie di strumenti di sviluppo professionali, che fornisce funzionalità per accelerare e semplificare la creazione e la distribuzione di prodotti integrati. Comprende un editor di codice sorgente multiplatforma, funzioni di completamento di codice intelligente basati su tipi di variabili, definizioni di funzioni e dipendenze di librerie, un flusso di lavoro multi-progetto con facile navigazione nella codebase del progetto, oltre a più riquadri e supporto

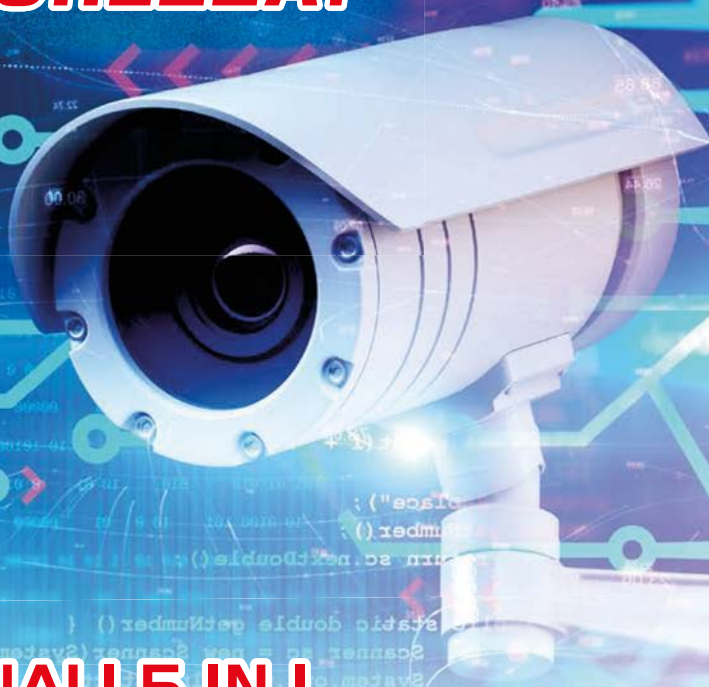
specifico. La piattaforma offre integrazione con l'interfaccia utente PlatformIO Home, con gestori di schede e librerie library, oltre a una procedura guidata di progetto intuitiva e un'ampia gamma di progetti di esempio. Offre inoltre un terminale integrato con PlatformIO Core (CLI) e un potente monitor seriale. Per utilizzarlo, se non si possiede nel computer Microsoft Visual Studio Code lo si scarica dal sito e una volta completata l'installazione, si apre l'Extension Manager di VSCode, si cerca dall'apposita casella PlatformIO IDE e si installa.

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

PROBLEMI DI SICUREZZA? A SORVEGLIARE CI PENSA LUI!



DVR A 4 CANALI 5 IN 1



€64,00

Cod. FR810-4



Videoregistratore a 4 canali 5 in 1 (AHD / CVI / TVI / IP / Analogico), dalle caratteristiche professionali. Dispone di 4 ingressi video (BNC) che possono essere utilizzati sia in modalità HD (AHD / CVI / TVI) che in modalità analogica. È dotato di interfaccia Ethernet (10-100 Base-T) che consente di gestire fino a 16 telecamere IP. Visione remota tramite PC connesso a Internet e Smartphone (iOS, Android) con possibilità di effettuare registrazioni e backup. Funzione OSD (On Screen Display) gestibile tramite mouse. La confezione comprende: DVR, mouse, alimentatore da rete (con uscita 12 VDC - 2A) e guida rapida di installazione in italiano. Hard Disk acquistabile separatamente.



VERSIONE A 8 CANALI

VERSIONE A 16 CANALI



DISPONIBILI ANCHE NELLE
VERSIONI A 8 CANALI

COD. FR810-8 € 82,00

E A 16 CANALI

COD. FR810-16 € 148,00

La fame nel mondo è un problema enfatizzato dallo spreco e dalla scarsa allocazione delle risorse alimentari. Tra le ragioni, la stabilità del raccolto (quanto tempo un raccolto può rimanere non raccolto prima di rovinarsi), il tempo, i parassiti, le malattie, la disponibilità di manodopera ecc. Un monitoraggio accurato di questi fattori può ridurre gli sprechi. Un ragionamento analogo si può applicare all'allevamento, perché il bestiame e il mangime necessari consumano circa un terzo della terra coltivabile. Raggiungere una maggiore sicurezza alimentare in modo sostenibile richiede una migliore supervisione del sistema alimentare. Il monitoraggio delle colture e del bestiame, reso possibile dall'IIoT, può migliorare l'efficienza del sistema

› IIoT per le risorse alimentari



alimentare. In quest'ottica si collocano i dispositivi LoRaWAN e le loro applicazioni: ad esempio, il servizio di agricoltura intelligente FarmLife di

itk monitora la salute del bestiame. Simile è la soluzione di monitoraggio IIoT proposta da lar.tech, basata con sensori etichettati sull'orecchio delle mucche, che trasmettono dati in modalità wireless in tempo reale tramite un gateway LoRaWAN. Un localizzatore GPS e un sensore biometrico monitorano i parametri vitali e la posizione del bestiame. Per quanto riguarda l'agricoltura, Oizom ha sfruttato l'infrastruttura di Tata Communications utilizzando la connettività di rete LoRaWAN per la sua soluzione di agricoltura intelligente Agribot, che consente di monitorare l'irrigazione. Anche Sensoterra ha creato una soluzione IIoT che può ridurre fino al 30% il consumo idrico per l'agricoltura. www.semtech.com

› Stellantis e Foxconn per l'auto connessa

Si chiama Mobile Drive ed è la joint-venture creata da Foxconn e Stellantis in collaborazione con FIH Mobile Ltd. per accelerare i tempi di introduzione nel mercato delle più avanzate tecnologie di in-car e connected-car del settore. Mobile Drive implementerà soluzioni tecnologiche in grado di incrementare la presenza di servizi, software ed elettronica di consumo avanzata, come l'interfaccia HMI e servizi di infotainment, ma anche di tecnologie per i veicoli del futuro, che saranno sempre più interconnessi (V2V, ossia comunicazione Veichle to Veichle) e connessi all'infrastruttura stradale (V2I, ossia Veichle to Infrastructure) nell'intento di diffondere a breve la guida autonoma.

La capacità di Foxconn di sviluppare a livello globale software e hardware per smartphone ed elettronica consumer, si unirà alle tecnologie automobilistiche di Stellantis, colosso dell'auto che da inizio anno unisce le competenze di



FCA e gruppo PSA. I veicoli del futuro saranno sempre più guidati e caratterizzati dal software e i produttori chiedono soluzioni sempre più creative e basate sul software, che permettano di connettere i conducenti e i passeggeri al veicolo, dentro e fuori. Mobile Drive verrà incontro a queste aspettative, mettendo le soluzioni software e i relativi hardware a disposizione non

solo di Stellantis, ma anche di altre case automobilistiche. Mobile Drive si focalizzerà su Infotainment, telematica e sviluppo di piattaforme Cloud Service attraverso innovazioni di software che dovrebbero includere applicazioni basate su Intelligenza Artificiale, 5G, servizi over-the-air avanzati e integrazioni smart cockpit.

www.stellantis.com/it

› Modelli di Machine Learning dalle query SQL

Con Amazon Redshift, il data warehouse su Cloud di Amazon AWS, è possibile usare dei database SQL per interrogare e combinare exabyte di dati strutturati e semi-strutturati nel proprio warehouse, nei database operativi e nel data lake. Ma se occorre elaborare questi dati per addestrare modelli di Machine Learning e utilizzare questi modelli per generare insight dai dati del warehouse bisogna esportare i dati di training da Amazon Redshift a un bucket Amazon Simple Storage Service (Amazon S3) e poi configurare e avviare un processo di Machine Learning, che però, richiede svariate competenze e più di persone per portarlo a termine. Oggi tutto questo viene semplificato perché uno specifico ser-

vizio Cloud, che è Amazon Redshift ML, aiuta a creare, addestrare e distribuire modelli di Machine Learning direttamente in cluster Amazon Redshift. Per creare un modello di Machine Learning si utilizza una semplice query SQL per specificare i dati da utilizzare per il training, e il valore di uscita che si desidera prevedere.

Dopo avere eseguito il comando SQL per creare il modello, Redshift ML esporta in modo sicuro i dati specificati da Amazon Redshift verso i bucket S3 e richiama Amazon SageMaker Autopilot (il servizio che permette di configurare e avviare processi di Machine Learning) per preparare i dati (pre-processing e feature engineering), selezionare l'algoritmo appropriato

› Da Nowi e Murata PMIC LoRa

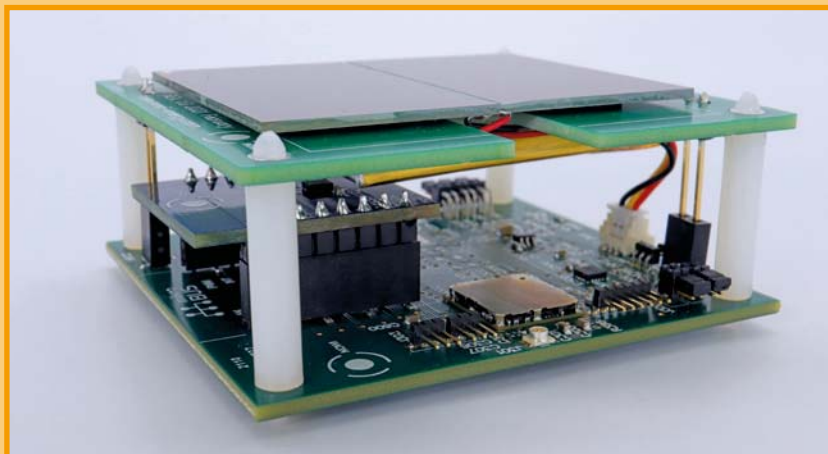
Murata e Nowi collaborano allo sviluppo di piattaforme IoT autonome per l'energia sostenibile, da utilizzare nelle soluzioni "Plug & Forget" basate sulla connettività LoRa. L'importanza dell'energy harvesting per le piattaforme IoT sta diventando sempre più evidente, poiché svolge un ruolo fondamentale nell'aumentare la durata, l'usabilità e la fattibilità dei dispositivi IoT. L'obiettivo della collaborazione è portare sul mercato piattaforme e progetti di riferimento che utilizzano i moduli LoRa miniatura a bassissimo consumo energetico Murata, alimentati dal PMIC (Power Management Integrated

Circuit) per energy harvesting estremamente compatto ed efficiente della Nowi. La piattaforma modulare compatta misura 60x65 mm e supporta la connettività LoRa implementata attraverso il modulo 1SJ o ABZ della Murata, inviando i dati dal sensore esterno collegato alla presa del sensore di bordo. In combinazione con il PMIC di raccolta dell'energia NH2 ultra-miniatura di Nowi, la piattaforma consente lo sviluppo semplice e veloce di prodotti connessi LoRa autonomi a basso costo. Il PMIC NH2 ricava elettricità dalle fonti di energia ambientale per caricare un'ampia varietà di elementi di accumulo di energia attraverso

la sua funzione MPPT efficiente e ottimizzata. Inoltre, l'NH2 offre una facilità d'uso senza precedenti, con una facile implementazione in fase di sviluppo. Il modulo 1SJ è una soluzione modem LoRaWAN stand-alone o opzionalmente controllata da comando AT che misura solo 10x8x1,6 mm, rendendolo il più piccolo disponibile oggi al mondo. Si basa sul transceiver a radiofrequenza Semtech SX1262 di seconda generazione (RFIC). Il dispositivo incorpora diverse modalità a bassa potenza che consentono all'RTC di funzionare assorbendo una corrente tipica di soli 1,3 µA. Inoltre, il package in resina offre robustezza fisica con un intervallo di temperatura da -40°C a +85°C. Il tipo ABZ è un altro modulo compatto autonomo che misura solo 12,5x11,6x1,76 mm, incapsulato in un contenitore schermato in metallo e comprendente un ricetrasmittitore wireless a spettro esteso a raggio ultra lungo Semtech SX1276.

Entrambi i modem LoRa sono integrati con un microcontrollore (MCU) ARM Cortex M0+ a 32 bit della serie STMicro STM32L0. Ogni MCU include 192 kB di memoria Flash e 20 kB di RAM e dispone di memoria sufficiente per incorporare le applicazioni degli utenti e di ospitare altri stack di modulazione.

www.nowi-energy.com/news/



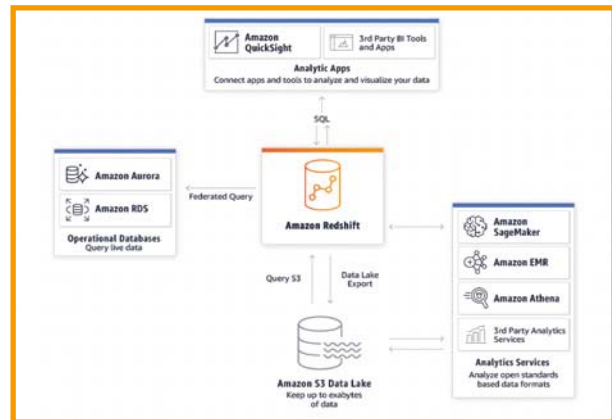
pre-costruito e applicare l'algoritmo per il training del modello. È possibile, opzionalmente, specificare l'algoritmo da utilizzare, che per esempio può essere XGBoost.

Redshift ML gestisce tutte le interazioni tra Amazon Redshift, S3 e SageMaker, compresi tutti i passaggi coinvolti nel training e nella compilazione.

Quando è stato addestrato, Redshift ML utilizza Amazon SageMaker Neo per ottimizzare il modello per il deployment e lo rende disponibile come funzione SQL. È possibile utilizzare la funzione SQL per applicare il modello di Machine Learning ai dati in query, report e dashboard.

Oltre alla disponibilità generale, Redshift ML ora include anche nuove funzionalità, tra le quali si possono elencare il supporto di Amazon Virtual Private Cloud (VPC).

www.aws.amazon.com/it/redshift/



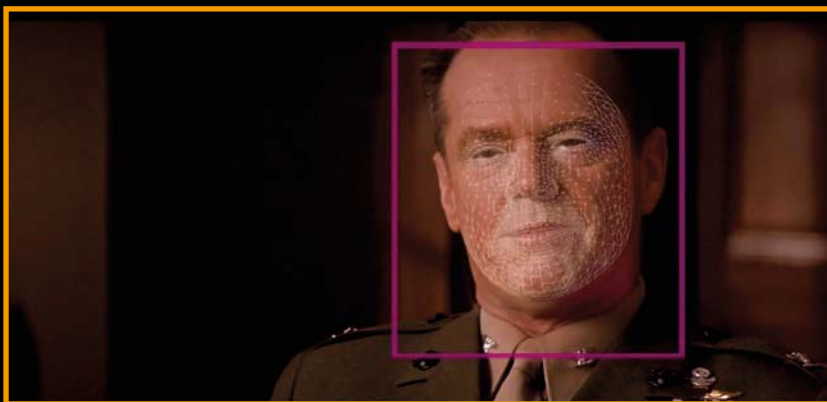
› Intelligenza Artificiale per un doppiaggio impeccabile

Far doppiare gli attori in varie lingue ad un computer, attraverso un sistema di riconoscimento facciale capace di leggere il labiale e le espressioni: questo è stato sperimentato in un'applicazione che vede ancora una volta il cinema avvalersi del computer. Dopo le impressionanti applicazioni di computer grafica in grado di inserire nei film improbabili e fantastiche scene e animazioni, trasformazioni ed effetti speciali, ci pensa l'Intelligenza Artificiale a strabiliare il mondo della "celluloide" con un algoritmo in grado di acquisire le espressioni facciali e il labiale degli

attori per riconoscere le parole nella lingua originale e fornire un impeccabile doppiaggio tradotto. Merito della collaborazione tra la startup londinese Flawless AI ed il Max Planck Institute for Informatics, da cui è nata una tecnologia in grado di catturare digitalmente la recitazione degli attori da fotogrammi di film o TV 2D e trasformarli in un modello 3D al computer. Dopo aver addestrato l'algoritmo di Intelligenza Artificiale ad apprendere la voce e la mimica facciale di specifici attori, è possibile generare versioni modificate che cambiano la voce e

l'espressione facciale per adattarsi a una lingua completamente diversa. Diverse aziende hanno utilizzato l'Intelligenza Artificiale per doppiare film, programmi TV, pubblicità e altri contenuti in varie lingue mantenendo le voci originali, ma l'approccio di Flawless AI va oltre il semplice rifacimento dell'audio, rimodellando i movimenti della bocca e le espressioni facciali dell'attore per adattarli al nuovo doppiaggio linguistico. Oltre ad assumere esperti supervisori degli effetti visivi di Hollywood, Flawless AI ha collaborato con i ricercatori del Max Planck Institute, che utilizza tecniche di Intelligenza Artificiale basate su reti neurali profonde per modificare digitalmente gli abiti che un attore indossa nella stessa scena, generare modelli 3D dei volti delle persone basati su immagini 2D e creare figure 3D animate al computer basate su singole immagini 2D di una persona. Qualcosa di più di quel che si vede nei video "deepfake" assistiti dall'AI. L'approccio Flawless AI potrebbe consentire il doppiaggio senza interruzioni dei film e dei programmi TV in più lingue, in tempi e con costi contenuti.

www.flawlessai.com/product



Decoder TV digitale terrestre DVB-T, DVB-T2, H.265/HEVC



DVB-T/T2
TERRESTRIAL

**H.265
HEVC**

**FULL HD
1080**

USB

HDMI

**mpeg
VIDEO**

**XVID
VIDEO**

cod. TP265

€ 29,00



Ricevitore d'antenna per ricevere il segnale TV digitale terrestre nel nuovo standard di trasmissione DVB-T, DVB-T2 H.265/HEVC. Oltre ai nuovi standard l'apparecchio può ricevere trasmissioni nei vecchi standard DVB-T H.264 e MPEG-4. Il dispositivo offre una perfetta qualità delle immagini Full HD 1080p. Offre la possibilità di collegare al connettore USB un disco rigido esterno e registrare programmi TV e radio. Può essere inoltre connesso a internet tramite il connettore RJ45 per utilizzare servizi come Weather, YouTube, IPTV, RSS, ecc. (I termini di utilizzo di questi servizi di terze parti sono soggetti alle aziende erogatrici). 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.



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

Tool per la sicurezza degli FPGA da Microchip

I sistemi mission-critical sono minacciati da cyber-criminali che tentano di estrarre informazioni critiche sui programmi (CPI) tramite gli FPGA che li governano; per mitigare tale rischio, Microchip ha creato lo strumento di sviluppo DesignShield per i suoi FPGA, che implementa contromisure per ridurre il rischio di clonazione, furto di proprietà intellettuale, reverse engineering o inserimento di Trojan. DesignShield è stato creato per gli sviluppatori di sistemi aerospaziali, di difesa e altri sistemi ad alta sicurezza, dai criminali informatici che cercano di acquisire un flusso di bit di un FPGA dal sistema sul campo. Scorgia il reverse engineering del flusso di bit, che spesso può includere CPI, oscurando il suo equivalente logico attraverso una combinazione di logica e tecniche di crittografia basate sul routing. DesignShield è disponibile su licenza come parte del Microchip's Early



Access Program, comprendente dispositivi e strumenti di progettazione FPGA. Basati sulla memoria Flash non volatile, gli FPGA Microchip offrono una sicurezza migliore rispetto alle alternative basate su SRAM che espongono dati bitstream sensibili a ogni ciclo di accen-

sione; includono anche funzionalità di sicurezza che impediscono l'overbuilding e la clonazione, proteggono l'IP di progettazione e forniscono una root of trust, comunicazioni sicure e capacità antimanomissione.

www.microchip.com

Gate-driver single-channel Infineon



Infineon propone la famiglia EiceDRIVER 1EDB di gate-driver single-channel con isolamento galvanico ingresso-uscita di 3 kV rms (UL 1577). La loro immunità ai transitori di modo comune (CMTI) supera i 300 V/ns. I gate-driver trovano applicazione in alimentatori switching, UPS, caricatori per veicoli elettrici. Sono dotati di sorgenti separate e a bassissima resistenza (0,95 Ω) e uscite capaci di commutare 5,4A di picco in modo source e 9,8 A di picco in modo sink;

tali caratteristiche consentono di ridurre le perdite di commutazione dei MOSFET di potenza. L'accuratezza del ritardo di propagazione è di ± 4 ns, contribuendo a ridurre le perdite di commutazione.

La tipica velocità di bloccaggio dello stadio di uscita è di soli 20 ns e supporta il funzionamento del sistema funzionalmente sicuro, soprattutto durante l'avvio.

La famiglia EiceDRIVER 1EDB è composta da quattro chip distinti per blocco in caso di sotto-tensione in uscita (UVLO):

- 1EDB7275F, da 0 V per MOSFET operanti a livelli logici;
- 1EDB8275F, da 0 V per MOSFET con livelli normali;
- 1EDB6275F, da 12V per MOSFET SiC 650V con pilotaggio a 15V);
- 1EDB9275F, da 14V per MOSFET SiC 650 V con pilotaggio a tensione ≥ 18 V.

www.infineon.com

Il modulo Bluetooth 5.1 supporta il mesh networking per applicazioni Massive IoT

U-blox ha utilizzato il SoC Bluetooth LE nRF52833 di Nordic Semiconductor per i moduli stand-alone Bluetooth 5.1 NINA-B4D. Dotati di CPU con architettura open, possono avere preinstallato Wirepas Massive, una soluzione di rete mesh decentralizzata su larga scala progettata per applicazioni Massive IoT. La serie NINA-B4 comprende il NINA-B406, dotato di antenna on board e il NINA-B400, con connettore U.FL per antenna esterna. La serie NINA-B4 viene fornita con il software u-connect di u-blox e può eseguire Wirepas Massive ed applicazioni di dispositivi di terze parti insieme al protocollo Wirepas. Wirepas Massive può essere distribuito in applicazioni di smart tracking, smart logistics, smart building, smart industry e smart metering. Il NINA-B4 ha un range di tem-

peratura operativa fino a 105 °C e permette il posizionamento indoor di precisione.

Il SoC nRF52833 combina un processore Arm Cortex M4 a 64 MHz e 32 bit con unità in virgola mobile (FPU), con un transceiver a 2,4 GHz che supporta anche il software Thread, Zigbee e IEEE 802.15.4, con potenza massima TX di +8 dBm e sensibilità RX di -95 dBm. Integra interfacce NFC-A, ADC, SPI a 32 MHz, UART/SPI/TWI, PWM, I²S e PDM ed è supportato dai SoftDevices S113, 122 o S140.

www.u-blox.com



Regolatore continuo Maxim per backup a supercondensatori



Per l'alimentazione di backup a supercondensatori, Maxim rende disponibile MAX38889, ultimo della famiglia di regolatori di backup Continua, capace di una regolazione del 2,5% e di un'efficienza di picco del 94%. In applicazioni come contatori di utenze intelligenti o telecamere per cruscotti automo-

bilistici, il regolatore Continua MAX38889 funziona in modalità buck per caricare una fonte di alimentazione di backup, come un supercondensatore. Quando manca l'alimentazione principale, il regolatore inverte la direzione, aumentando la tensione del supercondensatore per alimentare il sistema,

fino al ripristino dell'alimentazione principale. Il MAX38889 è adatto ad applicazioni portatili e non, come scanner di prezzi al dettaglio e telecamere di sorveglianza e altre utilizzate in casa, edilizia, automotive, automazione industriale e sanità IoT.

www.maximintegrated.com

Renesas lancia il modulo LTE CAT-M1 per il massive IoT

RYZ014A è il primo modulo IoT cellulare di Renesas che supporta la specifica LTE Cat-M1. Questa tecnologia consente ai dispositivi di connettersi direttamente a Internet senza bisogno di gateway, ma facendo affidamento sull'infrastruttura di comunicazione esistente di proprietà degli operatori di rete mobile. Ciò rende economico collegare ciascun dispositivo, il che è particolarmente vantaggioso quando si installano reti ampie che richiedono un basso costo per unità e che si adattano a tutto il mondo.

Il modulo RYZ014A risolve i problemi di progettazione e certificazione RF, PTCRB/GCF e del vettore per l'operatore di rete scelto dal cliente, in Nord America, APAC ed Europa.

Cellular IoT è uno dei pilastri della strategia di connettività di Renesas che include tra gli altri anche Bluetooth LE, Ultra-wide band (UWB) e Wi-SUN.

Renesas sta collaborando con Sequans Communications, uno sviluppatore e fornitore leader di chip e moduli 5G e 4G per IoT, sui moduli RYZ014A. Sequans offre la tecnologia hardware e software



Monarch LTE CAT-M1 utilizzata nel modulo e ha una vasta esperienza nell'implementazione di IoT wireless in tutto il mondo.

La tecnologia Monarch di Sequans è un eccezionale complemento all'ineguagliabile esperienza MCU fornita da Renesas. Renesas offre inoltre un portale web-based, gestito da ONE Tech, che fornisce un facile accesso per leggere i valori dei sensori e gestire i piani delle carte SIM da remoto, riducendo i costi operativi. Renesas ha combinato i mo-

duli RYZ014A con numerosi componenti del suo ampio portafoglio, tra cui circuiti integrati di alimentazione, caricatori agli ioni di litio, sensore da luce a digitale e più MCU della famiglia RA per creare combinazioni certificate che consentono di implementare rapidamente il monitoraggio remoto dei processi industriali e sistemi di irrigazione automatica degli impianti con connessione LTE. Renesas offre oltre 200 combinazioni che includono dispositivi compatibili.

www.renesas.com

Il fotorelé Toshiba commuta fino a 1,2A

Toshiba Electronics introduce il nuovo fotorelé TLP4590A, in package DIP6 di tipo 1-Form-B (normalmente chiuso), che ha una corrente nominale per lo stato di on di 1,2A. Inoltre, il componente è caratterizzato da una tensione nominale del terminale di uscita nello stato di off di 60V. La sua sensibilità è migliorata grazie

alla bassa corrente di attivazione dei LED, pari a 2mA. Le principali applicazioni di questi fotorelé comprendono quelle relative all'automazione degli edifici, e includono i sistemi di riscaldamento, ventilazione e condizionamento dell'aria e i sistemi sicurezza, nonché gli allarmi antincendio. I dispositivi trovano anche impiego in

varie tipologie di apparecchiature industriali, come i controllori logici programmabili (PLC) le interfacce I/O e l'hardware di controllo dei sensori. La loro tensione di isolamento di 5.000 Vrms (valore minimo) ne consente l'utilizzo in situazioni in cui sono richiesti elevato isolamento.

www.toshiba.com



STM certifica il chipset G3-PLC



STM annuncia il chipset certificato per G3-PLC Hybrid Powerline e comunicazione wireless basato su SoC multiprotocollo programmabile ST8500 ed RTX RF S2-LP sub-GHz a bassissima potenza. La specifica G3-PLC Hybrid consente alle apparecchiature smart-grid, smart-city, industriali e IoT di selezionare il miglior canale wireless o powerline disponibile in qualsiasi momento, automaticamente in base alle condizioni della rete. Il chipset combina il SoC di comunicazione powerline multiprotocollo programmabile ST8500 e il line driver STLD1 con il ricetrasmittitore radio sub-GHz a bassissima potenza S2-LP dell'azienda.

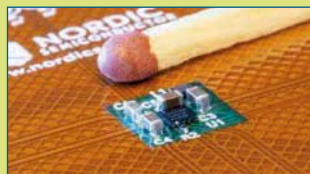
La programmabilità del SoC consente l'implementazione definita da software in grado di supportare un ampio portafoglio di stack di protocolli powerline, nelle bande di frequenza CENELEC e FCC. Il SoC ST8500 è incapsulato in un QFN56; STLD1 e S2-LP sono in package QFN24. Basato su 6LowPAN e IPv6 e combinando la connettività RF con lo stack di protocollo G3-PLC nativo, il SoC ST8500 consuma meno di 100 mW. Contiene un DSP e un core ARM Cortex-M4F per l'elaborazione e la gestione del sistema. Sia il DSP che il core ARM hanno il proprio codice e dati SRAM su chip, mentre le periferiche includono un motore crittografico AES a 128/256 bit sono integrate per soddisfare le esigenze delle applicazioni smart-energy. L'S2-LP è un RTX RF a bassissima potenza, destinato alla

www.st.com

Primo PMIC per Nordic

Nordic Semiconductor annuncia il suo primo circuito integrato di gestione dell'alimentazione (PMIC): siglato nPM1100, combina un regolatore di ingresso compatibile USB con protezione da sovratensione, caricabatteria da 400 mA e regolatore di tensione step-down da 150 mA. Il PMIC garantisce un'alimentazione affidabile e un funzionamento stabile per i SoC nRF52 e nRF53 di Nordic e massimizza la durata della batteria. Può essere utilizzato anche come PMIC generico per qualsiasi applicazione utilizzando batterie al litio. L'nPM1100 ha una corrente di riposo di 470 nA.

Il regolatore di ingresso richiede da 4,1 a 6,6 V e fornisce una tensione non regolata da 3 a



5,5V, con 500 mA di corrente. Il PMIC supporta il rilevamento USB Standard Downstream Port (SDP), Charging Downstream Port (CDP) e Dedicated Charger Port (DCP). A valle del regolatore di ingresso si trova un DC/DC buck con soft start, che fornisce fino a 150 mA di corrente con uscita regolata selezionabile fra 1,8 e 3,0 V. L'nPM1100 non richiede software di configurazione: tutte le impostazioni sono configurabili tramite pin. Il PMIC funziona tra -40 e 85 °C.

www.nordicsemi.com

Sensore di posizione magnetico di Allegro per l'ADAS

Allegro MicroSystems ha annunciato il sensore magnetico rotante lineare A31315, per applicazioni automobilistiche e industriali. Combina le comprovate tecnologie ad effetto Hall planare e verticale di Allegro per misurare i componenti del campo magnetico lungo tre assi (X, Y, Z), consentendo reali capacità di rilevamento 3D con un'ampia gamma dinamica magnetica senza saturazione. Il sensore A31315 fornisce la precisione di misura e le prestazioni necessarie per soddisfare i severi requisiti ADAS riguardanti i sistemi di sterzo, frenata, trasmissione e l'acceleratore. Il front-end flessibile 3D Hall e l'architettura di elaborazione del segnale configurabile, consentono rilevamenti di posizione lineare e di posizione rotativa estremamente accurati e assolute fino a 360°, facilitando le sfide di integrazione del sistema fornendo una maggiore libertà nel posizionamento del sensore. I dispositivi Allegro 3DMAG esistenti, i ALS31300 e ALS31313, supportano anche applicazioni magnetometriche 3D in cui sono necessari tutti e tre i componenti magnetici (BX, BY, BZ) per tracciare movimenti magnetici complessi.

La precisione superiore e la diagnostica su chip aiutano a migliorare la sicurezza di conducente, passeggero e pubblica per veicoli autonomi e con guida assistita.

Il basso consumo energetico e le robuste opzioni di gestione dell'alimentazione aiutano a prolungare e ottimizzare la durata della

batteria.

Rispondendo alle esigenze di sicurezza critiche dei sistemi di guida autonoma e delle applicazioni ADAS, il sensore di posizione A31315 ad alta precisione contiene funzionalità diagnostiche su chip avanzate per garantire un funzionamento affidabile e sicuro. Supporta il rilevamento della posizione sia rotativo che lineare con minimo errore angolare nativo sulla temperatura su qualsiasi piano (<1,2° sull'intervallo operativo completo del sensore -40°C ~ +150°C). Seguendo le linee guida sulla sicurezza funzionale dell'elemento di sicurezza fuori contesto (SEooC), il sensore A31315 supporta l'integrazione a livello di sistema ASIL-B (singolo die) e ASIL-D (doppio die) in conformità con ISO 26262 ed è qualificato per il settore automobilistico al grado AEC-Q100 0. Il sensore è disponibile in contenitore SOIC-8 e TSSOP-14.

A differenza delle tradizionali configurazioni a doppio die side-by-side, la costruzione di die impilate Allegro allinea gli elementi di rilevamento di entrambi i die, garantendo la misura di campi magnetici quasi identici. Questo consente prestazioni di corrispondenza dei canali superiori e soglie di confronto dei canali più strette.

www.allegromicro.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



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.

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.

CONVERTITORE DA HDMI AD AV PLUG AND PLAY

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

HDMI
↓
AV



€ 11,00

Cod. **BOXHDMITOAV**

CAVO SATA-USB 3.0 PER RASPBERRY PI E PC

Cavo adattatore Plug & Play da USB 3.0 (compatibile USB 1.0, 1.1 e 2.0) a SATA 22 pin. Permette di collegare alla porta USB di Raspberry Pi o di un PC, un hard disk SATA o SSD da 2,5". N.B. non adatto per hard disk SATA da 3,5". Non compatibile con Windows 10.

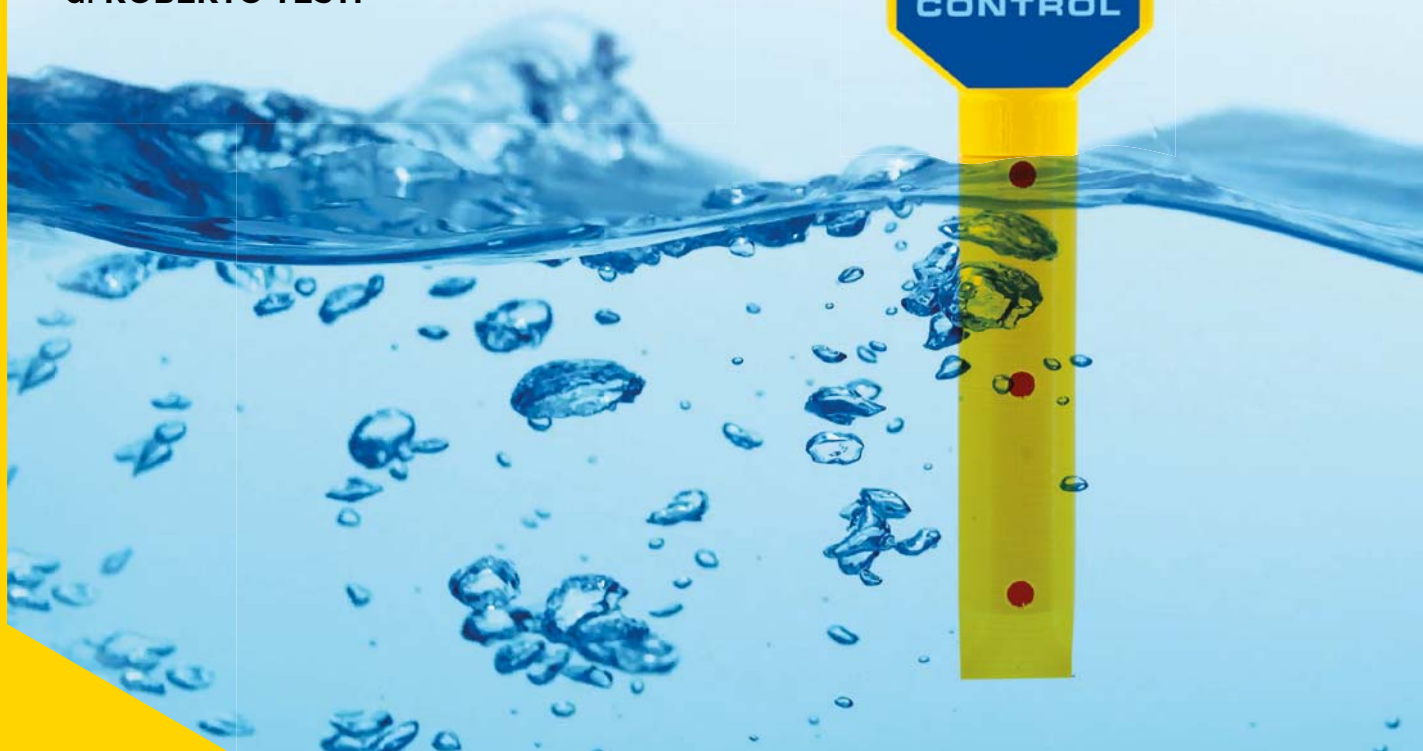


€ 7,90

Cod. **CAVOSATAUSB3**

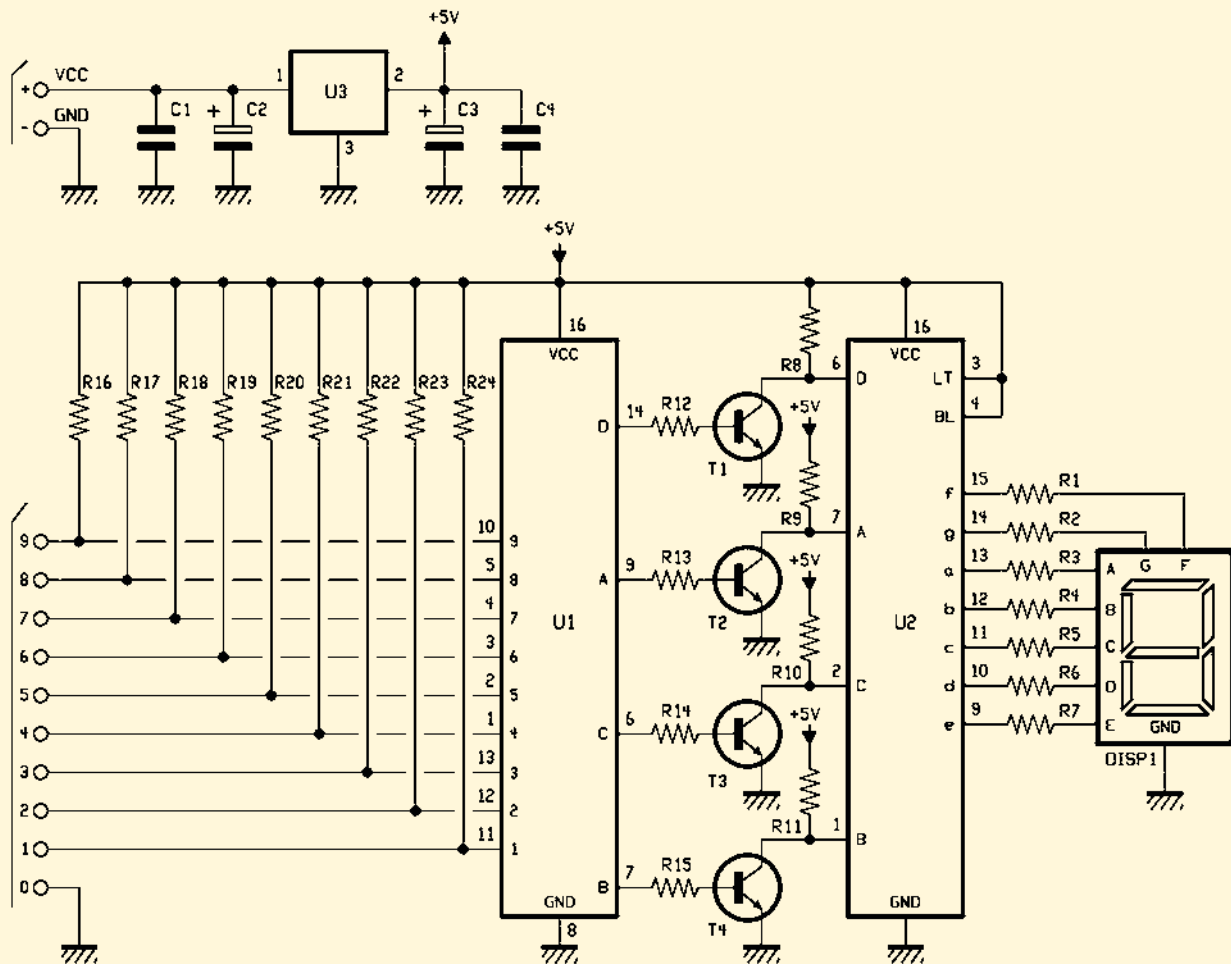
MISURATORE LIVELLO LIQUIDI

di ROBERTO TESTI



Visualizza su un display a LED il livello rilevato da un sensore di presenza di liquidi conduttivi, come l'acqua di rubinetto. Utile in pratica, è anche l'occasione per sperimentare con l'elettronica.

Esistono vari modi per rilevare il livello dei liquidi in serbatoi, cisterne, piscine e vasche (ad esempio per la raccolta di acqua piovana) ed anche per indicarlo, più o meno precisamente: si va dal semplice paio di contatti elettrici che, quando viene toccato dal liquido conduttivo, presenta una resistenza elettrica rilevabile da un circuito elettronico ai galleggianti collegati a potenziometri per ottenere una resistenza e quindi una tensione elettrica proporzionale al livello. In questo articolo vi proponiamo un'alternativa davvero originale che fa uso di logiche binarie e non di circuiti analogici come quelli appena elencati, per fornire un'indicazione discreta sulla presenza e sul livello del liquido contenuto in una vasca, cisterna ecc. Per l'esattezza il nostro progetto punta a fornire un'indicazione su 9 livelli, corrispondenti ciascuno a un contatto elettrico



collocato su un'estremità (sagomata a stecca) del circuito stampato (**Fig. 1**) che verrà bagnato e che quindi formerà una resistenza di basso valore insieme al contatto comune di massa (GND) che è il più profondo, vale a dire quello che starà più in basso nella stecca sensore.

SCHEMA ELETTRICO

Vediamo dunque di cosa si tratta andando ad analizzare lo schema elettrico, che vedete formato da un encoder decimale - BCD della famiglia 74HC e un BCD to 7-segment converter che trasforma il segnale numerico in formato BCD (4 bit) in un numero che verrà visualizzato da un comune e convenzionale display a LED a 7 segmenti più punto decimale. Ciascun input del decoder U1 (un 74HC147) è provvisto di resistore di pull-up e collegato a una piazzola della stecca rivelatrice di livello, in modo che il liquido, quando si trova nella parte corrispondente al

contatto più basso (livello minimo rilevabile) determini lo zero logico sul bit di peso minore (2 alla 0, quindi 1); la sequenza di posizionamento degli elettrodi è quindi tale che il liquido, salendo, determini l'incremento a passi di un'unità. In questo il compito è semplificato dal fatto che gli ingressi del decoder sono decimali e non binari, quindi andando dall'1 al 9 hanno peso 1, 2, 3, 4, 5 e così via. Quando tutti e 9 sono aperti e pertanto ad uno logico (gli input sono a logica negata...) l'uscita dell'encoder vale 0 (decimo stato).

Rimanendo sull'uscita, il formato in cui sono espressi (codificati) i valori d'ingresso è binario BCD, ossia su quattro bit con peso 8421: 1 corrisponde al bit A, 2 al B, 4 al C e 8 al D. Pertanto la conversione da decimale a BCD è la seguente:

1 = 0001

2 = 0010

3 = 0011

4 = 0100
 5 = 0101
 6 = 0110
 7 = 0111
 8 = 1000
 9 = 1001
 10 = 1010
 11 = 1011
 12 = 1100
 13 = 1101
 14 = 1110
 15 = 1111

Ciò detto, sappiamo che per come sono state disposte le piazzole corrispondenti agli input dell'U1, la sequenza man mano che il liquido sale di livello, è 1, 2, 3, 4, 5, 6, 7, 8, 9, partendo dallo 0. Questi sono i numeri che il display a LED ci proporrà grazie all'intermediazione del driver U2, di tipo BCD to 7-segment, che convertirà le combinazioni binarie di 4 bit in segnali elettrici sulle sette linee di cui dispone in uscita, per accendere i LED dei 7 segmenti (il punto decimale qui non ci serve...) del DISP1; quest'ultimo è del tipo a catodo comune, quindi ogni linea delle a, b, c, d, e, f, g (quelle dei rispettivi segmenti) corrisponde a un anodo e va pilotata con il livello logico alto.

Notate che tra le uscite dell'U1 e gli ingressi BCD dell'U2 abbiamo inserito quattro inverter logici basati su transistor NPN montati ad emettitore comune e impiegati per invertire le condizioni logiche fornite dal 74HC147, il quale ha le uscite a logica invertita, mentre gli input del 74HC4511 sono a logica diretta; infatti ogni uscita BCD dell'U1 pilota la base di un transi-

stor, il cui collettore presenta una condizione logica opposta alla corrispondente linea di input dell'U2, in quanto nella configurazione a emettitore comune, un livello alto sulla base corrisponde alla saturazione del transistor e quindi alla massima corrente di collettore, che determina la caduta dell'intera tensione di alimentazione sulla resistenza di carico (R8 per T1, R9 per T2, R10 per T3 ed R11 per T4).

Le linee di uscita del 74HC4511 pilotano ciascuna un segmento del display mediante un resistore di limitazione della corrente. Il 74HC4511 dispone di due linee di controllo siglate BL ed LT, che qui sono state disabilitate ponendole entrambe a livello logico alto (positivo di alimentazione); BL è un ingresso che permette di forzare lo spegnimento dell'intero display (ad esempio nella gestione di visualizzatori a più digit quando si opera in multiplex) a prescindere dallo stato delle linee e dalla cifra che il display dovrebbe visualizzare di conseguenza. LT, invece, corrisponde alla funzione del Lamp Test e se viene posto a zero logico forza l'accensione di tutti i LED, per verificare l'integrità dei singoli segmenti del visualizzatore.

Concludiamo l'analisi del circuito con lo stadio di alimentazione, che partendo dalla tensione continua d'ingresso, applicata ai contatti +Vcc e -/GND, permette di ricavare i 5Vcc stabilizzati occorrenti al funzionamento della logica e quindi degli integrati U1 e U2 e del display a LED; tale stadio è formato dal regolatore lineare a tre terminali U3 (è il popolare 78L05) in contenitore TO-92, che partendo dalla tensione applicata al suo terminale E rispetto a massa (M, che è l'elettrodo comune a ingresso ed uscita) eroga

CARATTERISTICHE TECNICHE

-  **Tensione di alimentazione:**
8+15 Vcc
-  **Corrente assorbita:**
50 mA
-  **Livelli rilevabili:**
9
-  **Visualizzazione numerica con display a LED**
-  **Indicazione livello sotto il minimo**

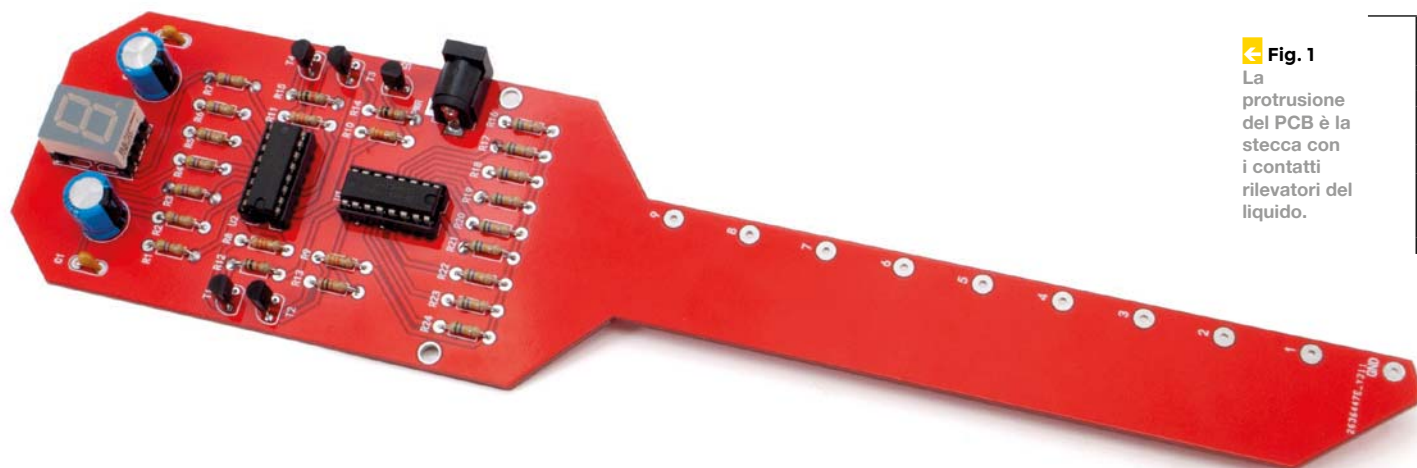
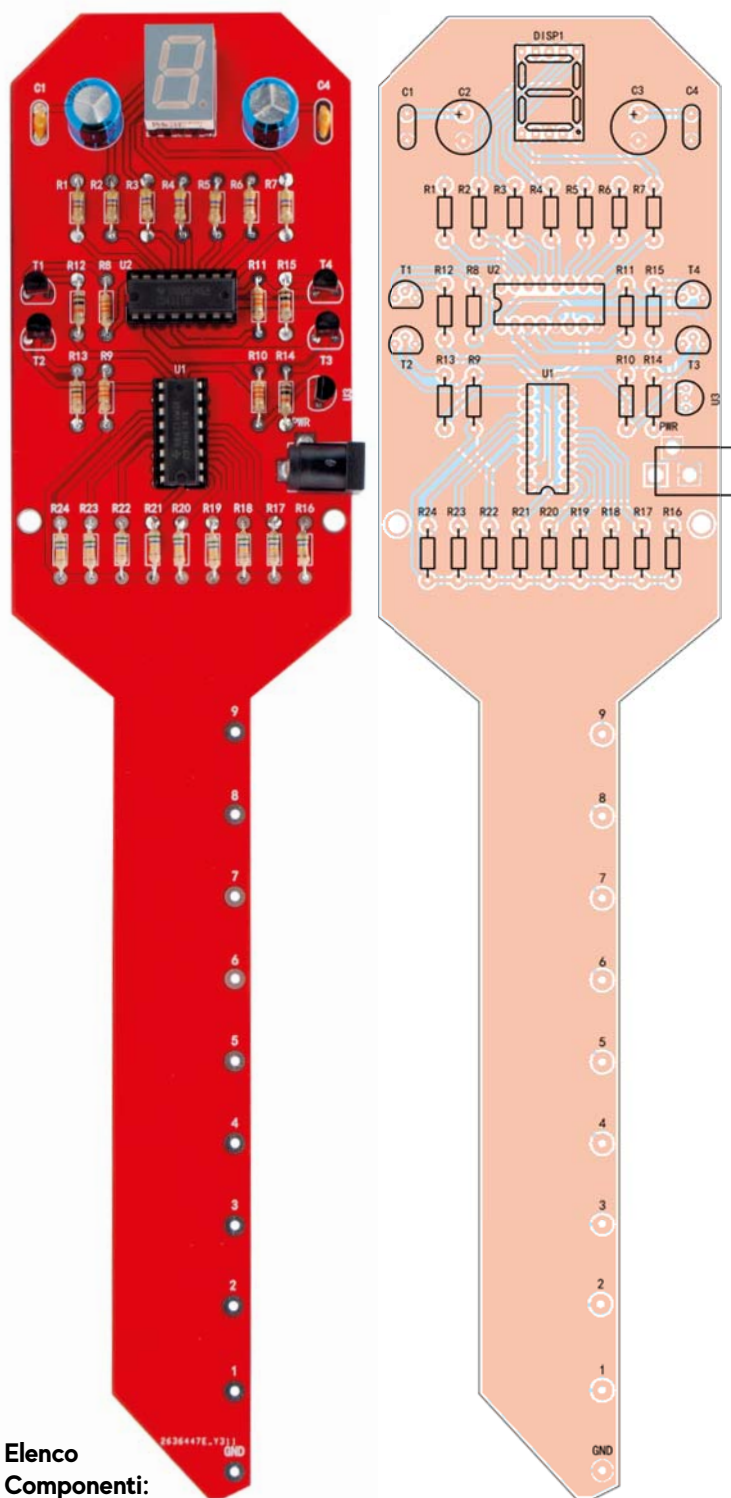


Fig. 1
 La
 protrusione
 del PCB è la
 stecca con
 i contatti
 rilevatori del
 liquido.



Elenco Componenti:

R1÷R7: 470 ohm
 R8÷R11: 33 kohm
 R12÷R15: 10 kohm
 R16÷R24: 560 kohm
 C1,C4: 100 nF multistrato
 C2,C3: 470uF elettrolitico
 T1-T4: BC547
 U1: 74HC147

U2: CD4511
 U3: 78L05
 DS1: display 7 segmenti CC

Varie:
 - zoccolo 8+8
 - strip femmina tornito 5PIN
 - stampato S1554 63x276 mm

5V ben regolati, filtrati dai condensatori C3 (filtra l'eventuale ripple residuo sull'alimentazione) e C4 (sopprime eventuali disturbi impulsivi) per ottenere una tensione pulita e idonea a garantire il funzionamento preciso del decoder decimale – BCD e del driver per display BCD to 7-segment.

L'alimentazione fornita al regolatore dal connettore di ingresso (Jack DC) viene filtrata da un'ulteriore coppia di condensatori: C1 che attenua eventuali interferenze impulsive captate dai fili di collegamento e C2 che livella la tensione laddove provenga da un alimentatore AC/DC. La tensione da applicare al circuito dev'essere di valore compreso fra 8 e 15 Vcc e può provenire da una pila, da una batteria o da un alimentatore AC/DC anche non stabilizzato.

REALIZZAZIONE PRATICA

Bene, riteniamo esaurita la spiegazione del funzionamento del circuito e passiamo alle note costruttive: l'intero rilevatore di livello liquidi prende posto su un circuito stampato che è stato progettato non solo per ospitare i componenti elettronici, ma anche, su un'apposita protrusione a stecca, gli elettrodi per rilevare i livelli e, all'estremità, il contatto comune (massa) a tutti e 9. Come al solito il circuito stampato si realizza per fotoincisione partendo dalle due tracce lato rame (il PCB è a doppia ramatura) e impressionando le facce di una basetta ramata presensibilizzata, che inizialmente lascerete rettangolare, quindi procedendo allo sviluppo e all'incisione; inciso il circuito si forano le piazzole e prima di procedere al montaggio dei componenti si va a sagomarlo come previsto. Allo scopo si ritaglia l'eccedenza con un seghetto o con una cesoia per metalli, poi si rifinisce il bordo con delle lime da ferro di grana grossa inizialmente (lima "bastarda" o per sgrossatura) e fine successivamente (lima media o fine) ovvero con carta vetrata.

Ottenuta la sagoma finale si passa alla saldatura dei componenti occorrenti, iniziando da resistenze e zoccoli per i due integrati logici e per il display (qui lo zoccolo si ottiene con due file di strip femmina da 5 poli ciascuna) e proseguendo con i transistor ed il regolatore 78L05, da orientare come mostrato nel piano di montaggio visibile in queste pagine. Si passa quindi ai condensatori non polarizzati, alla presa jack DC da circuito stampato per l'alimentazione e si procede con gli elettrolitici, ognuno da orien-

CONTROLLARE IL LIVELLO ...DELLA CO₂

Monitorare con un rilevatore elettronico la CO₂ nelle aule scolastiche per evitare la diffusione del COVID-19, allo scopo di garantire le lezioni in presenza: è quanto previsto dal progetto "Misuriamo la CO₂ a scuola" cui ha aderito il Comune di Molfetta, che ha acquistato e fatto installare 19 sensori in tutti gli istituti comunali per rilevare la qualità dell'aria. Tenere aperte le porte, aprire le finestre e far circolare l'aria all'interno dei luoghi chiusi sono buone pratiche per prevenire la diffusione del virus. Già, ma come si fa a capire quando cambiare l'aria? La soluzione proposta dal progetto è affidarsi a uno strumento in grado di desumere, dalla qualità dell'aria, quando bisogna aprire; e la qualità si può valutare in base alla concentrazione di anidride carbonica determinata da rilevatori collegati in WiFi per controllare in tempo reale dati e statistiche con una app, cambiano colore, come un semaforo, in base alla concentrazione di CO₂. Sotto la soglia delle 400 parti per milioni (ppm) di CO₂ si accende una luce verde, tra i 400 e i 700 ppm diventa



gialla, tra i 700 e i 1.000 ppm, e sopra i 1.000, quando la situazione diventa più critica, la luce è rossa. Una volta installato il rilevatore alle spalle della cattedra, saranno gli stessi ragazzi a notare il cambiamento cromatico e ad aprire porte e finestre in caso di colore arancione o rosso per far cambiare l'aria. Il dispositivo intelligente e le web app sono stati realizzati dall'azienda Befreest (befreest.com).

tare come indicato nel piano di montaggio; per completare il circuito si inseriscono nei rispettivi zoccoli il 74HC147 (U1) il 74HC4511 (U2) e il display a sette segmenti (DISP1) riferendosi, per il corretto orientamento, al piano di montaggio. Comunque, se avete fatto le cose bene, ossia se nell'inserire e saldare gli zoccoli avete verificato di disporre la tacca di riferimento come nel piano di montaggio, per l'inserimento di U1 e U2 non

dovete fare altro che allineare gli integrati con le tacche di riferimento dei rispettivi zoccoli. Abbiamo anche previsto un bel contenitore stampato in 3D (chi volesse realizzarselo troverà il relativo file .STL sul nostro sito www.elettronicaain.it insieme ai file del progetto) che racchiude l'intero circuito stampato, dotato di aperture nella parte alta per applicare l'alimentazione al jack DC (Fig. 2) e vedere il display, e nella parte

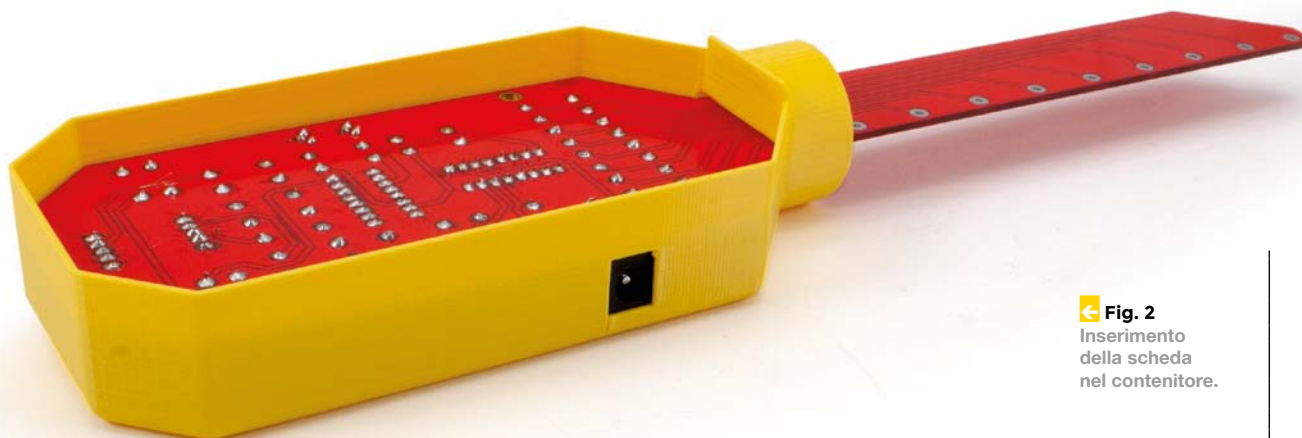


Fig. 2
Inserimento
della scheda
nel contenitore.

Fig. 3
Il rivelatore
inscatolato
e pronto per l'uso.



bassa
per far
entrare
il liquido nel quale
la parte sensore sarà
immersa per rilevare il
livello; quest'ultima porzione
riporta vari fori disposti equidi-
stanti (potete vederlo nella foto
in Fig. 3, che riporta il rivelatore
completo e racchiuso nel contenito-
re) che abbiamo previsto per evitare
la formazione di bolle d'aria (se non vi
fossero, non sarebbe certo che il liquido
penetrasse portandosi al livello del conteni-
tore da monitorare).

Per evitare che il liquido nel serbatoio o cisterna da monitorare, salendo troppo, quindi oltre il livello massimo previsto, penetri nella zona di circuito stampato che ospita i componenti elettronici, creando inaccettabili e deleteri cortocircuiti, è consigliabile sigillare con del silicone acetico trasparente o bianco la feritoia del contenitore da cui passa la parte a stecca del PCB, utilizzata per rilevare il livello del liquido; anche il coperchio, una volta chiuso il circuito nel contenitore, può essere sigillato con silicone, sempre allo scopo di fornire un buon grado di protezione dal bagnamento.

Per l'uso ricordate di installare il dispositivo in modo che sia ben stabile (onde evitare che scivoli o cada nel liquido di cui deve rilevare il livello) e protetto da eventuali schizzi di liquido; disponetelo con il lato del display rivolto all'esterno della vasca o serbatoio dove dovrà rilevare il livello, così da vedere facilmente il numero dall'esterno ed anche a una certa distanza. Per fissare il rivelatore di livello liquidi utilizzate delle fascette da elettricista o un collarino in plastica affittato alla parete o ad una staffa. Il cavetto dell'alimentazione dovrà essere ben fissato in modo che rimanga in alto e non si bagni; se volete, potete anche prevedere l'alimen-

tazione a pile o batterie da collocare all'interno della parte del contenitore che ospita l'elettronica, magari adattando quest'ultima; resta inteso che con le pile o batterie dovrete fare i conti con l'autonomia di funzionamento, che però potete aumentare inserendo un pulsante normalmente aperto in serie al filo positivo dell'alimentazione (oppure interrompendo la pista della linea positiva di alimentazione che proviene dal jack DC) così da attivare il rivelatore solamente quando volete conoscere il livello, lasciandolo spento in tutto il tempo rimanente.

CONCLUSIONI

Bene, con questo abbiamo terminato la descrizione di un progetto semplice e alla portata di tutti, che permette di ottenere un'indicazione visiva e immediata sul livello del liquido contenuto in un contenitore di vario genere. Ricordiamo che il nostro misuratore va "tarato" ossia posizionato in base all'escursione del livello del liquido da monitorare, in modo che la punta tocchi più o meno il minimo e che il massimo non oltrepassi la base della zona del contenitore dove si trova l'elettronica.

Ricordiamo, infine, che il misuratore va bene per rilevare e indicare il livello di qualsiasi liquido purché non sia infiammabile né corrosivo per i metalli. □



Cosa occorre?

I componenti utilizzati in questo progetto sono facilmente reperibili. Il PCB può essere realizzato tramite il servizio PCBSERVICE disponibile sul sito di Futura Elettronica all'indirizzo futurashop.it/produzionePCB

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

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.



LED COB

LIGHTS

**100 LUMEN
WATT**



Realizza lettere, numeri per pannelli pubblicitari, illumina al meglio i prodotti nella tua vetrina e tanto altro con le barrette a LED COB. Ricoperte da gomma siliconica sono realizzate su supporto in alluminio per una migliore dissipazione di calore. Utilissime per svariati utilizzi!



Cod. LEDCOB200B



Cod. LEDCOB200W



Cod. LEDCOB200V



Cod. LEDCOB200R

BARRETTE DI LED COB 12V-15W LUNGHEZZA 20 CM

Tipo di LED: COB chip
Angolo del fascio: 120°
Flusso luminoso: 100 lumen/watt
Vita media: 50.000 ore
Alimentazione: 12 VDC
Potenza: 15W
Consumo: 900 mA
Temperatura di funzionamento: da -20°C a +60°C
Dimensioni: 200x10x3 mm
Peso: 11 grammi

€5,90
A BARRETTA

**Collega
insieme più
barrette di
colori
differenti!**

Prezzi IVA inclusa.

Alimentale con:

MAX 1
barretta di
LED COB

ALIMENTATORE SWITCHING
12V - 15W - 1,25A

€5,00

Cod. POS15W12V



MAX 2
barrette di
LED COB

ALIMENTATORE
SWITCHING
12V - 36W - 3A

€8,50

Cod. ADL-40-12



MAX 6
barrette di
LED COB

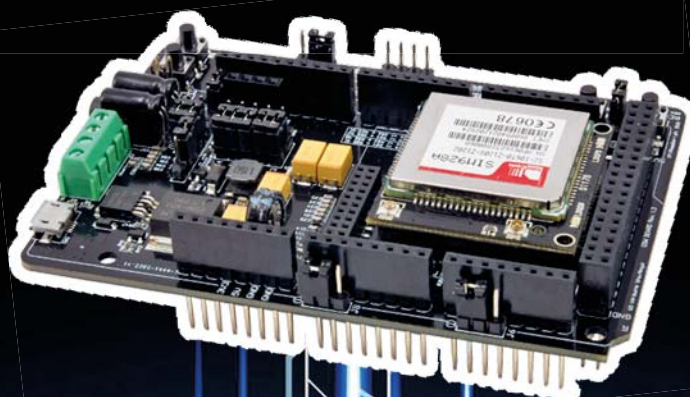
ALIMENTATORE
SWITCHING
12V - 100W -
8,5A

€20,00

Cod. MW10012



UNO SKETCH PER L' MQTT



dell'ing. **MATTEO DESTRO**



Inviando dati a ThingSpeak tramite rete mobile utilizzando il protocollo MQTT, la nostra libreria per Arduino e la shield GSM/GPRS.

N

el fascicolo n° 255 abbiamo introdotto la nostra libreria per la gestione del protocollo MQTT e anticipato la pubblicazione di uno sketch finalizzato allo studio del protocollo stesso e rivolto a tutti quelli, tra voi, interessati all'argomento e curiosi di sperimentare le combinazioni di codifica dei pacchetti

MQTT. Tuttavia riteniamo necessario affrontare l'argomento con un ulteriore sketch di approfondimento che mostrerà al lettore come interfacciarsi a un server MQTT per inviargli dei dati attraverso azioni di PUBLISH. Mostreremo anche come ricevere dei dati dal server attraverso l'azione di SUBSCRIBE. Il nuovo sketch che andiamo a descrivere sfrutta la piattaforma ThingSpeak di cui abbiamo già parlato in altre occasioni e ci permette di inviare i dati tramite protocollo MQTT e di visualizzarli sia graficamente che numericamente e se neces-

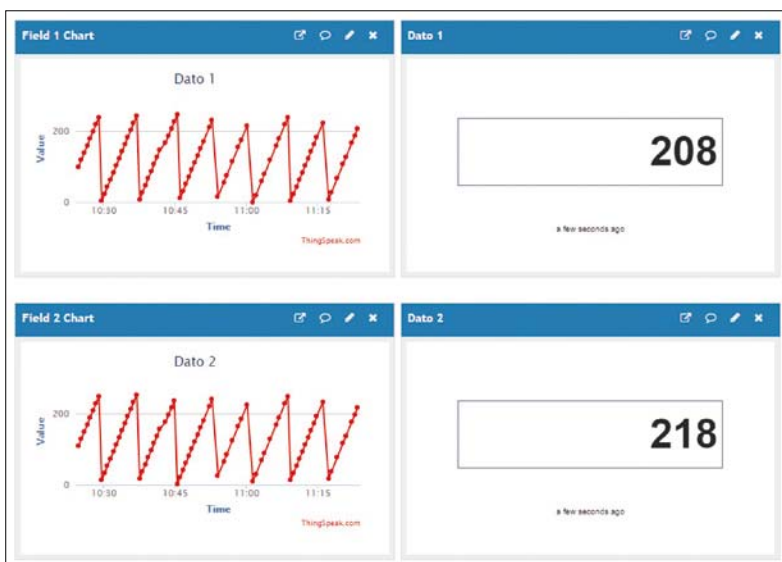


Fig. 1
Configurazione campi nuovo canale di test su ThingSpeak.

sario è possibile elaborare i dati inviati con MatLab. Il nostro sketch, semplicemente manda dei dati numerici progressivi a tempi regolari a due campi distinti appositamente creati su ThingSpeak i quali vengono poi visualizzati sia in modo grafico che numerico. Il valore numerico ovviamente rappresenta il valore istantaneo inviato mentre la visualizzazione grafica rappresenta l'andamento nel tempo dei valori. Lo sketch utilizzerà i pacchetti MQTT CONNECT e

PUBLISH rispettivamente per connettersi al server e per inviare i dati. Il pacchetto PUBLISH viene inviato con QoS = 0

INTRODUZIONE ALLO SKETCH

Lo sketch che presenteremo è composto da quattro file distinti e, come già spiegato in precedenti articoli, la suddivisione aiuta a comprendere e mantenere più facilmente il codice:

- **"GSM_MQTT_311_TcpIp.ino"** → contenente tutte le dichiarazioni di variabili e costanti nonché le funzioni **"void setup()"** e **"void loop()"**;
- **"DigitalOutput.ino"** → utilizzato per la configurazione degli I/O di uscita;
- **"TcpIp_Mqtt_SM.ino"** → contenente il codice di gestione della connessione TCP/IP al server ThingSpeak e relativa gestione dei pacchetti MQTT della nostra libreria; sono presenti anche le macchine a stati per la gestione dei flussi di connessione, test e invio dati;
- **"TimersInt.ino"** → contenente il codice di gestione della base dei tempi basata sul Timer 5 dell'Arduino Mega 2560; la base dei tempi è fissata a 2ms.

Il file **"GSM_MQTT_311_TcpIp.ino"** vede in cima una serie di costanti definite tramite la direttiva al compilatore **"#define"** tra cui abbiamo **"#define TEST_MQTT_PING_CMD"**, la quale abilita l'invio

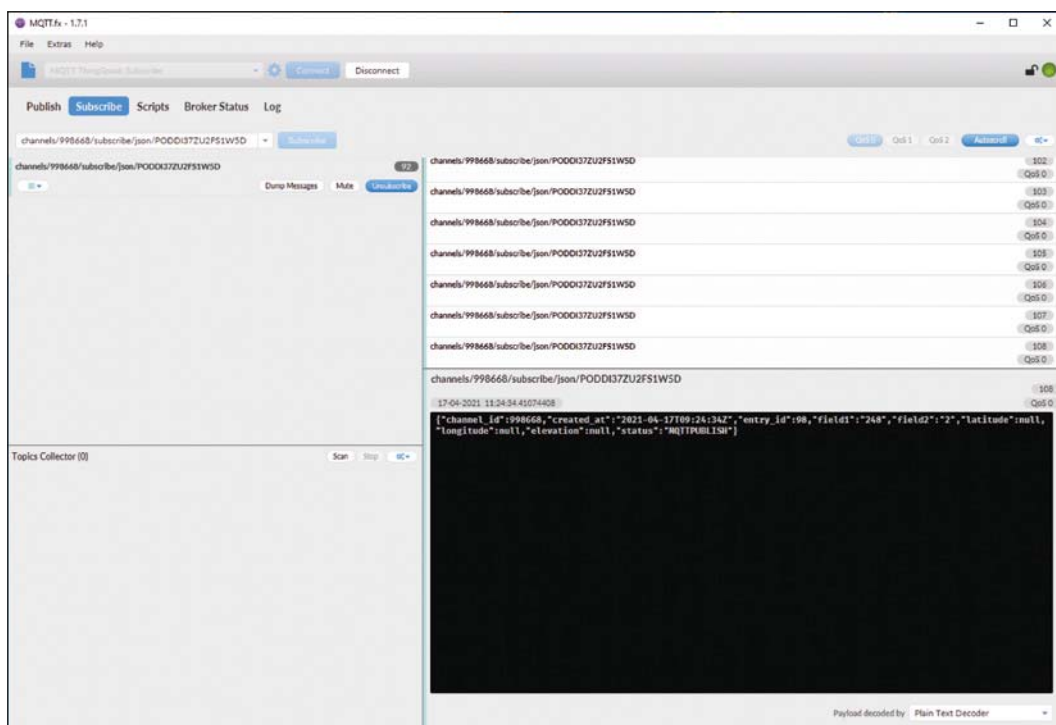


Fig. 2
Richiesta MQTT SUBSCRIBE da software "MQTT.fx".

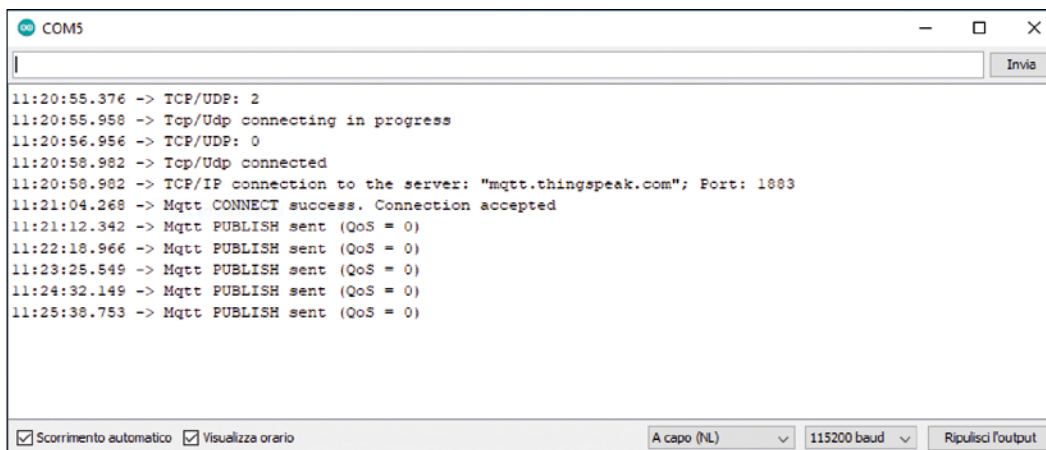


Fig. 3
Monitor
seriale
Arduino –
Solo MQTT
PUBLISH
abilitato.

del pacchetto MQTT PING al server, e **#define TEST_SUBSCRIBE_QOS_0** la quale abilita l'invio del pacchetto MQTT SUBSCRIBE; se non usate vanno commentate. Dopodiché c'è una serie di definizioni riguardanti gli stati della macchina a stati usata per la gestione dell'invio dei comandi AT nonché della gestione dei pacchetti MQTT inviati e ricevuti. Come al solito si usa la direttiva al compilatore **"PROGMEM"** per salvare in memoria Flash le stringhe usate durante lo sketch e stampate sul monitor seriale di Arduino. Infine seguono le dichiarazioni delle variabili usate tra cui vale la pena menzionare le variabili che identificano lo stato passato, presente e futuro della macchina a stati che gestisce sia l'invio dei comandi AT che dei pacchetti MQTT ovvero:

- **PastStateSendCmd**
- **PresentStateSendCmd**
- **FutureStateSendCmd**

Vedremo più avanti a cosa servono e come vengono gestite. Un'altra variabile utile è **"FieldValue"** la quale, come suggerisce il nome, serve per contenere il valore del campo da inviare al server ThingSpeak.

COSA FA LO SKETCH

Lo sketch è stato pensato con l'unico scopo di inviare dei dati a un nuovo "canale" (Channel) creato sulla piattaforma ThingSpeak; in particolare il canale mette a disposizione due soli campi detti **"Field 1"** e **"Field 2"**. Il nuovo canale è stato nominato **"MQTT_Test"**. La **Fig. 1** mostra come sono stati configurati i due campi; come si vede, è stata scelta sia una rappresentazione di tipo grafica che di tipo numerica come accennato precedentemente.

Ogni 60 secondi il codice invia i dati ai due campi usando il pacchetto MQTT PUBLISH, i valori dei due campi vengono incrementati di 20 unità dopo ogni invio, così da avere dei valori diversi di volta in volta.

La **Fig. 2** mostra i dati che sono stati inviati al server MQTT dal nostro sketch. Questo è possibile grazie al software **"MQTT.fx"** che può essere utilizzato sia per inviare dei pacchetti MQTT di PUBLISH che fare richiesta di dati tramite pacchetti MQTT SUBSCRIBE come fatto nel nostro caso. Quindi una volta inviata la richiesta al server quest'ultimo inizierà a mandare, tramite pacchetto MQTT PUBLISH, i dati ricevuti dal client (ovvero il nostro sketch) con gli ultimi valori pubblicati. Questo ci aiuta a capire se il nostro codice e relativa libreria stanno lavorando correttamente.

In questa fase lo sketch è abilitato a inviare solo i dati tramite pacchetto MQTT PUBLISH; i pacchetti MQTT SUBSCRIBE e PING sono disabilitati.

Il monitor seriale di Arduino, visibile in **Fig. 3** mostra l'invio del pacchetto MQTT PUBLISH con cadenza 60 secondi circa, come già spiegato, e, a conferma di ciò, abbiamo la **Fig. 4**, la quale mostra la spia seriale connessa tra il modulo GSM e la scheda Arduino. In fondo alla predetta figura si vedono i comandi AT usati per inviare il pacchetto MQTT PUBLISH.

Se ora attiviamo la possibilità di fare richiesta di ricevere i dati dal server tramite invio pacchetto MQTT SUBSCRIBE (rimozione del commento alla direttiva al compilatore **#define TEST_SUBSCRIBE_QOS_0**) otterremmo come risultato che:

1. lo sketch continuerà a inviare i dati al server con cadenza 60 secondi circa;
2. il server ad ogni variazione dei dati invierà a sua volta un pacchetto MQTT PUBLISH al client; questo avverrà sia quando siamo noi



Fig. 4
Spia seriale invio
comandi AT al
motore GSM.

```

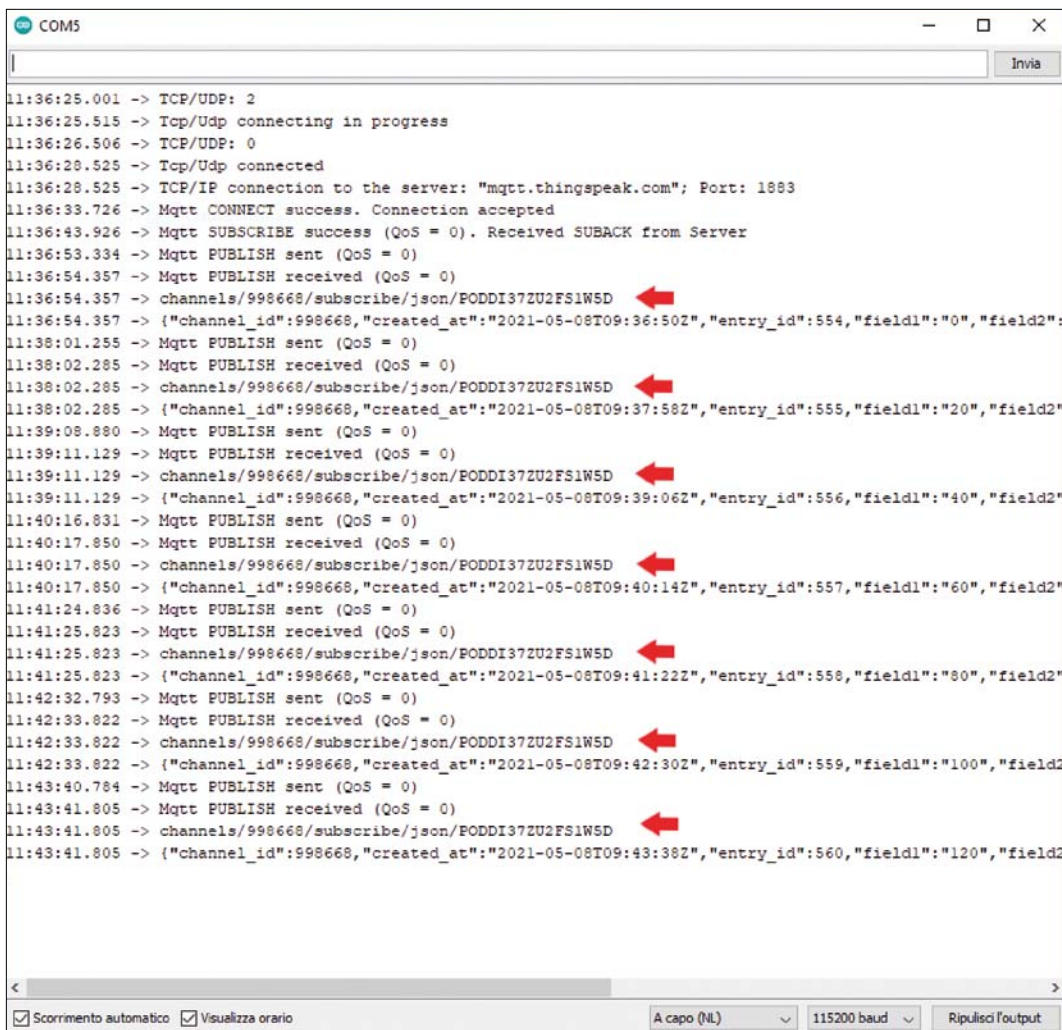
AT+CIPSTATUS<CR><LF>
<CR><LF>
OK<CR><LF>
<CR><LF>
STATE: IP STATUS<CR><LF>
AT+CIPSTART="TCP","mqtt.thingspeak.com",1883<CR><LF>
<CR><LF>
OK<CR><LF>
AT+CIPSTATUS<CR><LF>
<CR><LF>
OK<CR><LF>
<CR><LF>
STATE: TCP CONNECTING<CR><LF>
<CR><LF>
CONNECT OK<CR><LF>
AT+CIPSTATUS<CR><LF>
<CR><LF>
OK<CR><LF>
<CR><LF>
STATE: CONNECT OK<CR><LF>
AT+CIPSEND<CR><LF>
<CR><LF>
> <DLE>[E<NUL>]<EOT>[MQTT<EOT>]<A<ETX>]<è<NUL>]<DC3>Device Disconnected<NUL>]<DC2>mdestro@tiscali.it<NUL>]<DLE>
]H2TRGZ4LY140GYE<SUB>]<CR><LF>
SEND OK<CR><LF>
<CR><LF>
+CIPRXGET: 1,"54.157.36.210:1883"<CR><LF>
AT+CIPRXGET=2,255<CR><LF>
<CR><LF>
+CIPRXGET: 2,4,0,"54.157.36.210:1883"<CR><LF>
<STX> <NUL>]<NUL>]<CR><LF>
OK<CR><LF>
AT+CIPSTATUS<CR><LF>
<CR><LF>
OK<CR><LF>
<CR><LF>
STATE: CONNECT OK<CR><LF>
AT+CIPSEND<CR><LF>
<CR><LF>
> 00<NUL>] (channels/998668/publish/8KKSM2SMBCSBSMfffield1=0&field2=10&status=MQTTPUBLISH<SUB>]<CR><LF>
SEND OK<CR><LF>
AT+CIPSTATUS<CR><LF>
<CR><LF>
OK<CR><LF>
<CR><LF>
STATE: CONNECT OK<CR><LF>
AT+CIPSEND<CR><LF>
<CR><LF>
> 0P<NUL>] (channels/998668/publish/8KKSM2SMBCSBSMfffield1=20&field2=30&status=MQTTPUBLISH<SUB>]<CR><LF>
SEND OK<CR><LF>
AT+CIPSTATUS<CR><LF>
<CR><LF>
OK<CR><LF>
<CR><LF>
STATE: CONNECT OK<CR><LF>
AT+CIPSEND<CR><LF>
<CR><LF>
> 0P<NUL>] (channels/998668/publish/8KKSM2SMBCSBSMfffield1=40&field2=50&status=MQTTPUBLISH<SUB>]<CR><LF>
SEND OK<CR><LF>

```

ad aggiornare i dati, è come se avessimo un feedback di avvenuta ricezione da parte del server, sia quando attraverso il software "MQTT.fx" si eseguirà un'azione di PUBLISH sempre sul server MQTT.

Tutto ciò è visibile nella **Fig. 5**, dove si nota che ad ogni azione di PUBLISH da parte del client verso il server segue un'azione di ricezione dati tramite pacchetto MQTT PUBLISH dal server verso il client in formato JSON come richiesto (il pacchetto ricevuto è evidenziato da una freccia nella figura). Invece la **Fig. 6**, mette in evidenza la ricezione di un pacchetto MQTT PUBLISH inviato dal server al client a seguito della ricezione di un pacchetto dati MQTT dal software "MQTT.fx" facente le veci di un secondo client connesso al server. Cosa vuol dire quanto appena spiegato? Semplicemente che ci siamo messi nella situazione in cui ci

sono due client distinti, uno è un software che gira su PC mentre l'altro è uno sketch Arduino, entrambi connessi allo stesso server ovvero ThingSpeak. Quando anche il secondo client invia dei dati al server con il pacchetto MQTT PUBLISH, questi vengono rigirati al client realizzato con lo sketch Arduino. Le frecce presenti nella figura mostrano il pacchetto inviato dal server e i due campi "field1" e "field2" che contengono i valori 50 e 120. Ora se attiviamo anche la direttiva al compilatore "#define TEST_MQTT_PING_CMD" otterremo anche l'invio del pacchetto MQTT PING con cadenza determinata da un timer impostato a 120 secondi. Ogni volta che il timer scade il codice appena possibile invia il pacchetto MQTT PING. La **Fig. 7** evidenzia ciò; come si vede, il pacchetto di PING viene inviato con una cadenza di circa 3 minuti, come indicato dalle frecce inserite nella figura. Con questo abbiamo concluso la descrizione a



```
COM5
11:36:25.001 -> TCP/UDP: 2
11:36:25.515 -> Tcp/Udp connecting in progress
11:36:26.506 -> TCP/UDP: 0
11:36:28.525 -> Tcp/Udp connected
11:36:28.525 -> TCP/IP connection to the server: "mqtt.thingspeak.com"; Port: 1883
11:36:33.726 -> Mqtt CONNECT success. Connection accepted
11:36:43.926 -> Mqtt SUBSCRIBE success (QoS = 0). Received SUBACK from Server
11:36:53.334 -> Mqtt PUBLISH sent (QoS = 0)
11:36:54.357 -> Mqtt PUBLISH received (QoS = 0)
11:36:54.357 -> channels/998668/subscribe/json/FODDI37ZU2FS1W5D
11:36:54.357 -> {"channel_id":998668,"created_at":"2021-05-08T09:36:50Z","entry_id":554,"field1":"0","field2":
11:38:01.255 -> Mqtt PUBLISH sent (QoS = 0)
11:38:02.285 -> Mqtt PUBLISH received (QoS = 0)
11:38:02.285 -> channels/998668/subscribe/json/FODDI37ZU2FS1W5D
11:38:02.285 -> {"channel_id":998668,"created_at":"2021-05-08T09:37:58Z","entry_id":555,"field1":"20","field2":
11:39:08.880 -> Mqtt PUBLISH sent (QoS = 0)
11:39:11.129 -> Mqtt PUBLISH received (QoS = 0)
11:39:11.129 -> channels/998668/subscribe/json/FODDI37ZU2FS1W5D
11:39:11.129 -> {"channel_id":998668,"created_at":"2021-05-08T09:39:06Z","entry_id":556,"field1":"40","field2":
11:40:16.831 -> Mqtt PUBLISH sent (QoS = 0)
11:40:17.850 -> Mqtt PUBLISH received (QoS = 0)
11:40:17.850 -> channels/998668/subscribe/json/FODDI37ZU2FS1W5D
11:40:17.850 -> {"channel_id":998668,"created_at":"2021-05-08T09:40:14Z","entry_id":557,"field1":"60","field2":
11:41:24.836 -> Mqtt PUBLISH sent (QoS = 0)
11:41:25.823 -> Mqtt PUBLISH received (QoS = 0)
11:41:25.823 -> channels/998668/subscribe/json/FODDI37ZU2FS1W5D
11:41:25.823 -> {"channel_id":998668,"created_at":"2021-05-08T09:41:22Z","entry_id":558,"field1":"80","field2":
11:42:32.793 -> Mqtt PUBLISH sent (QoS = 0)
11:42:33.822 -> Mqtt PUBLISH received (QoS = 0)
11:42:33.822 -> channels/998668/subscribe/json/FODDI37ZU2FS1W5D
11:42:33.822 -> {"channel_id":998668,"created_at":"2021-05-08T09:42:30Z","entry_id":559,"field1":"100","field2":
11:43:40.784 -> Mqtt PUBLISH sent (QoS = 0)
11:43:41.805 -> Mqtt PUBLISH received (QoS = 0)
11:43:41.805 -> channels/998668/subscribe/json/FODDI37ZU2FS1W5D
11:43:41.805 -> {"channel_id":998668,"created_at":"2021-05-08T09:43:38Z","entry_id":560,"field1":"120","field2":

Scorrimto automatico Visualizza orario A capo (NL) 115200 baud Ripulisci l'output
```

Fig. 5
Monitor
seriale
Arduino con
attivata la
richiesta di
SUBSCRIBE.

grandi linee di cosa fa lo sketch realizzato, ora proviamo a dare qualche dettaglio in più su come usare la libreria per fare quanto appena descritto.

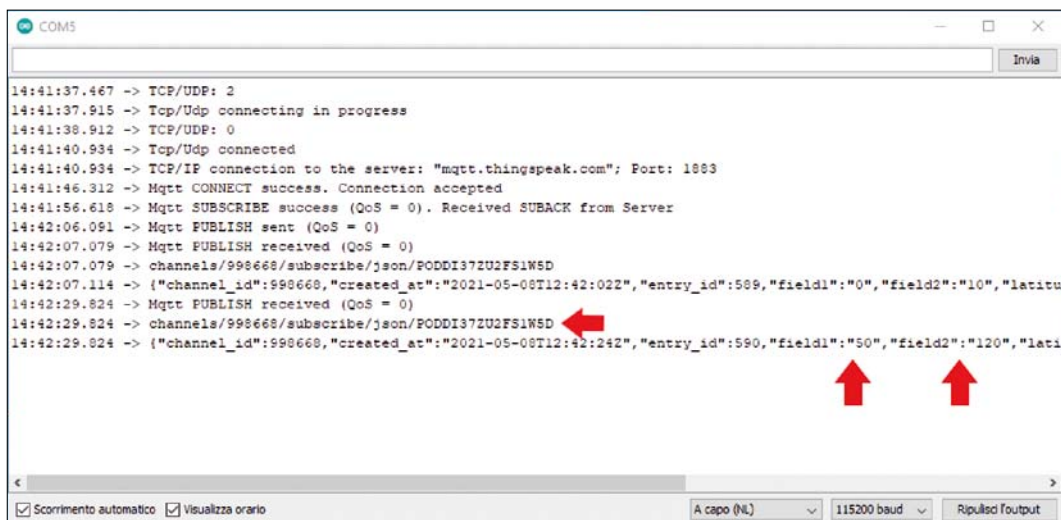
COME FUNZIONA LO SKETCH

La prima cosa da fare per utilizzare correttamente lo sketch è programmare la EEPROM con le stringhe necessarie alla corretta gestione della libreria MQTT. Per fare ciò si usa lo sketch **"GSM_WriteEeprom.ino"** fornito a corredo delle nostre librerie e già ampiamente usato. In testa allo sketch ci sono le solite direttive al compilatore per salvare in Flash le stringhe di testo che verranno poi usate per programmare la EEPROM. Lo spezzone di codice che ci riguarda è quello proposto nel **Listato 1**. Come si può vedere, ci sono tutti i campi usati dalla libreria MQTT ovvero:

- **"client Identifier"** o codice ID usato con il pacchetto MQTT CONNECT;
 - gli eventuali messaggi **"Will Topic"** e **"Will Message"** da usare sempre con il pacchetto MQTT CONNECT se attivati tramite apposito flag;
 - **"User Name"** e **"Password"** anch'essi da usare con il pacchetto MQTT CONNECT;
 - il **"Topic"** da usare con il pacchetto MQTT PUBLISH (questo dipende dall'applicazione e dal server cui ci si connette);
 - i quattro **"Topic"** da usare per le richieste di SUBSCRIBE.
- In questo caso abbiamo messo un **"Topic"** per richiedere solo il campo dati **"field1"**; un **"Topic"** per richiedere solo il campo dati **"field2"**; un **"Topic"** per richiederli entrambi e infine un **"Topic"** che richiede il campo dati in formato JSON e attualmente usato nello sketch.
- Ovviamente nello sketch ci sono delle direttive al compilatore per selezionare anche gli altri **"Topic"**.



Fig. 6
Situazione con due Client connessi al Server e richiesta SUBSCRIBE attiva.



```

14:41:37.467 -> TCP/UDP: 2
14:41:37.915 -> Tcp/Udp connecting in progress
14:41:38.912 -> TCP/UDP: 0
14:41:40.934 -> Tcp/Udp connected
14:41:40.934 -> TCP/IP connection to the server: "mqtt.thingspeak.com"; Port: 1883
14:41:46.312 -> Mqtt CONNECT success. Connection accepted
14:41:56.618 -> Mqtt SUBSCRIBE success (QoS = 0). Received SUBACK from Server
14:42:06.091 -> Mqtt PUBLISH sent (QoS = 0)
14:42:07.079 -> Mqtt PUBLISH received (QoS = 0)
14:42:07.079 -> channels/998668/subscribe/json/PODDI37ZU2FS1W5D
14:42:07.114 -> {"channel_id":998668,"created_at":"2021-05-08T12:42:02Z","entry_id":589,"field1":"0","field2":"10","latitu
14:42:29.824 -> Mqtt PUBLISH received (QoS = 0)
14:42:29.824 -> channels/998668/subscribe/json/PODDI37ZU2FS1W5D
14:42:29.824 -> {"channel_id":998668,"created_at":"2021-05-08T12:42:24Z","entry_id":590,"field1":"50","field2":"120","lati

```

Nello sketch di programmazione della EEPROM le direttive al compilatore **"PROGMEM"** sono divise in sezioni le quali possono essere abilitate o disabilitate tramite apposite direttive al compilatore **"#define"**. Ad esempio la sezione riguardante la gestione della libreria MQTT è gestita dalla direttiva al compilatore **"#define WRITE_MQTT_CODE"**, se questa è commentata si ha che le stringhe esposte sopra non verranno memorizzate

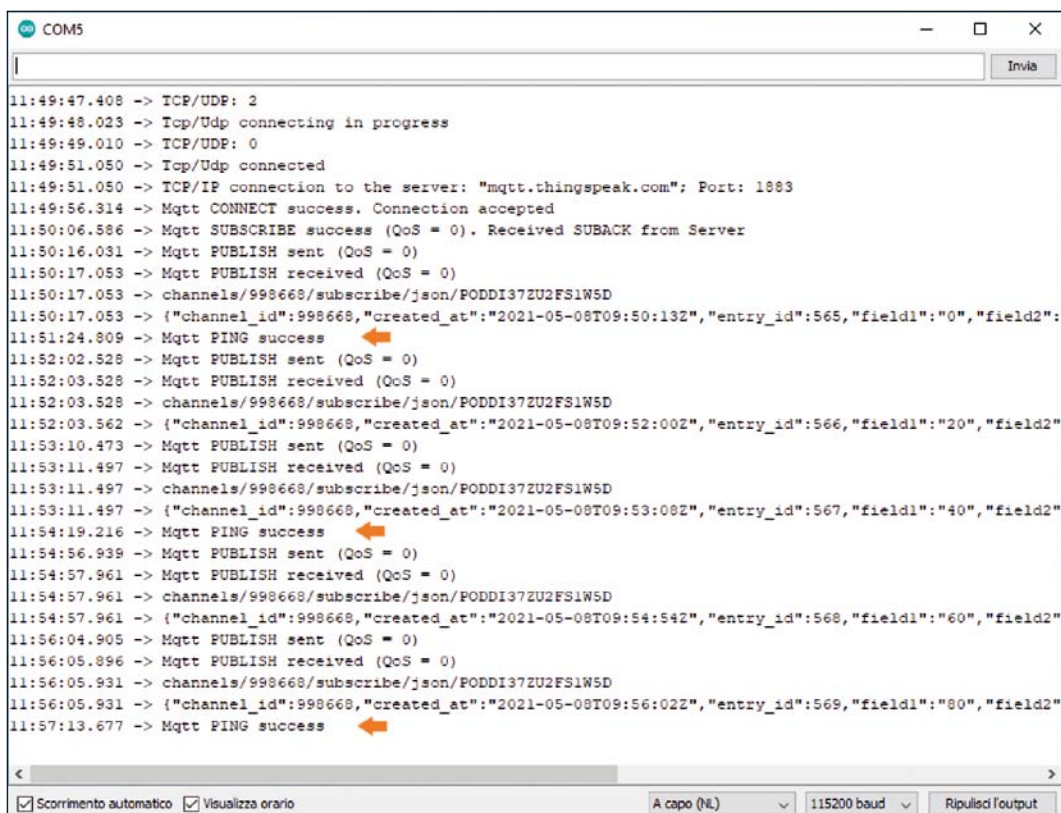
in EEPROM. Inoltre nella sezione inerente le connessioni TCP/IP ci sono le direttive al compilatore per la connessione al server ThingSpeak ovvero:

```

const char FLASH_REMOTE_DOMAIN_SERVER_0[] PROGMEM
= "\mqtt.thingspeak.com\";
const char FLASH_REMOTE_SERVER_PORT_0[] PROGMEM
= "\1883\";

```

Fig. 7
Monitor serial Arduino con attivata la richiesta di SUBSCRIBE e PING.



```

11:49:47.408 -> TCP/UDP: 2
11:49:48.023 -> Tcp/Udp connecting in progress
11:49:49.010 -> TCP/UDP: 0
11:49:51.050 -> Tcp/Udp connected
11:49:51.050 -> TCP/IP connection to the server: "mqtt.thingspeak.com"; Port: 1883
11:49:56.314 -> Mqtt CONNECT success. Connection accepted
11:50:06.586 -> Mqtt SUBSCRIBE success (QoS = 0). Received SUBACK from Server
11:50:16.031 -> Mqtt PUBLISH sent (QoS = 0)
11:50:17.053 -> Mqtt PUBLISH received (QoS = 0)
11:50:17.053 -> channels/998668/subscribe/json/PODDI37ZU2FS1W5D
11:50:17.053 -> {"channel_id":998668,"created_at":"2021-05-08T09:50:13Z","entry_id":565,"field1":"0","field2":
11:51:24.809 -> Mqtt PING success
11:52:02.528 -> Mqtt PUBLISH sent (QoS = 0)
11:52:03.528 -> Mqtt PUBLISH received (QoS = 0)
11:52:03.528 -> channels/998668/subscribe/json/PODDI37ZU2FS1W5D
11:52:03.562 -> {"channel_id":998668,"created_at":"2021-05-08T09:52:00Z","entry_id":566,"field1":"20","field2"
11:53:10.473 -> Mqtt PUBLISH sent (QoS = 0)
11:53:11.497 -> Mqtt PUBLISH received (QoS = 0)
11:53:11.497 -> channels/998668/subscribe/json/PODDI37ZU2FS1W5D
11:53:11.497 -> {"channel_id":998668,"created_at":"2021-05-08T09:53:08Z","entry_id":567,"field1":"40","field2"
11:54:19.216 -> Mqtt PING success
11:54:56.939 -> Mqtt PUBLISH sent (QoS = 0)
11:54:57.961 -> Mqtt PUBLISH received (QoS = 0)
11:54:57.961 -> channels/998668/subscribe/json/PODDI37ZU2FS1W5D
11:54:57.961 -> {"channel_id":998668,"created_at":"2021-05-08T09:54:54Z","entry_id":568,"field1":"60","field2"
11:56:04.905 -> Mqtt PUBLISH sent (QoS = 0)
11:56:05.896 -> Mqtt PUBLISH received (QoS = 0)
11:56:05.931 -> channels/998668/subscribe/json/PODDI37ZU2FS1W5D
11:56:05.931 -> {"channel_id":998668,"created_at":"2021-05-08T09:56:02Z","entry_id":569,"field1":"80","field2"
11:57:13.677 -> Mqtt PING success

```



⬇ Listato 1

```
#ifndef WRITE_MQTT_CODE
const char FLASH_MQTT_CLIENT_IDENTIFER[] PROGMEM = "4CEE7E16DB6144DCB97E6FFE94FE9363";
const char FLASH_MQTT_WILL_TOPIC[] PROGMEM = "Device Disconnected";
const char FLASH_MQTT_WILL_MESSAGE[] PROGMEM = "Device Disconnected";
const char FLASH_MQTT_USER_NAME[] PROGMEM = "mdestro@tiscali.it";
const char FLASH_MQTT_PASSWORD[] PROGMEM = "H2zzzZ4L3Y1zzzYE";
const char FLASH_MQTT_PUBLISH_TOPIC_NAME[] PROGMEM = "channels/998668/publish/8KKS25MBCSBSWWF";
const char FLASH_MQTT_SUBSCRIBE_TOPIC_FILTER_1[] PROGMEM = "channels/998668/subscribe/fields/field1/PODDI37ZU2FS1W5D";
const char FLASH_MQTT_SUBSCRIBE_TOPIC_FILTER_2[] PROGMEM = "channels/998668/subscribe/fields/field2/PODDI37ZU2FS1W5D";
const char FLASH_MQTT_SUBSCRIBE_TOPIC_FILTER_3[] PROGMEM = "channels/998668/subscribe/fields/+PODDI37ZU2FS1W5D";
const char FLASH_MQTT_SUBSCRIBE_TOPIC_FILTER_4[] PROGMEM = "channels/998668/subscribe/json/PODDI37ZU2FS1W5D";
#endif
```

Questi servono per connettersi in TCP/IP al server usando la porta 1883. Solo dopo avere instaurato una connessione stabile, si può inviare il pacchetto MQTT CONNECT usando le stringhe di connessione sopra esposte. Ora possiamo iniziare ad analizzare lo sketch: si inizia dalla funzione "**void setup()**" per poi giungere alla funzione "**void loop()**"; la prima serve per inizializzare l'hardware e le variabili dello sketch. In particolare si hanno:

- inizializzazione strutture dati contenente gli indirizzi in EEPROM in cui sono memorizzate le stringhe di testo usate dalla libreria GSM come codici PIN, PUK, Password ecc.;
- inizializzazione strutture dati libreria MQTT;
- lettura dati dalla EEPROM;

- abilitazione degli interrupt di libreria per la gestione dei moduli GSM, abilitazione della seriale usata per il debug e la stampa a video delle stringhe di testo usate dallo sketch, configurazione UART per gestione modulo GSM, attivazione modalità GPRS con singola connessione e accensione modulo GSM;
- inizializzazione macchina a stati gestione interfaccia UART modulo GSM;
- inizializzazione macchina a stati gestione funzioni sketch.

Conclusa l'inizializzazione il codice esegue la funzione "**void loop()**" nella quale trovano posto:

- richiamo funzione di gestione macchina a stati del modulo GSM;
- richiamo funzione di gestione macchina a stati dello sketch MQTT.

⬇ Listato 2

```
void ProcessSM_Gsm(void) {
  Gsm.ExecuteUartState();
  if (Gsm.GsmFlag.Bit.GsmInitInProgress == 1) {
    Gsm.InitGsmSendCmd();
    Gsm.InitGsmWaitAnswer();
  } else if (Gsm.GsmFlag.Bit.GprsInitInProgress == 1) {
    Gprs.InitGprsSendCmd();
    Gprs.InitGprsWaitAnswer();
  } else {
    Gsm.UartContinuouslyRead();
    Gsm.ProcessUnsolicitedCode();
    Gsm.GsmAnswerStateProcess();
  }

  #ifndef TEST_SUBSCRIBE_QOS_0
    CheckPublishDataFromServer();
  #endif
  Io.LedBlink(PIN_LED9, 25, T_1SEC);
}

PrintTcpUdpState();
}
```

La prima funzione elencata è visibile nel **Listato 2** e inizia con la gestione della seriale di comunicazione verso il modulo GSM (**Gsm.ExecuteUartState()**); segue del codice condizionale necessario all'inizializzazione del modulo GSM dove la prima sequenza di comandi AT inizializza le funzioni base mentre la seconda sequenza serve per configurare la modalità GPRS per la connessione TCP/IP. Esaurite le due sezioni di inizializzazione il codice richiamerà all'infinito le tre funzioni di libreria presenti nella sezione "else" del codice condizionale in esame, in quanto i due flag che indicavano quale operazione di inizializzazione eseguire sono ormai settati a "0". Durante l'inizializzazione si vedrà lampeggiare il LED rosso LD12 della nostra scheda GSM mentre a conclusione delle inizializzazioni inizierà a lampeggiare il LED verde LD9 della scheda demo.



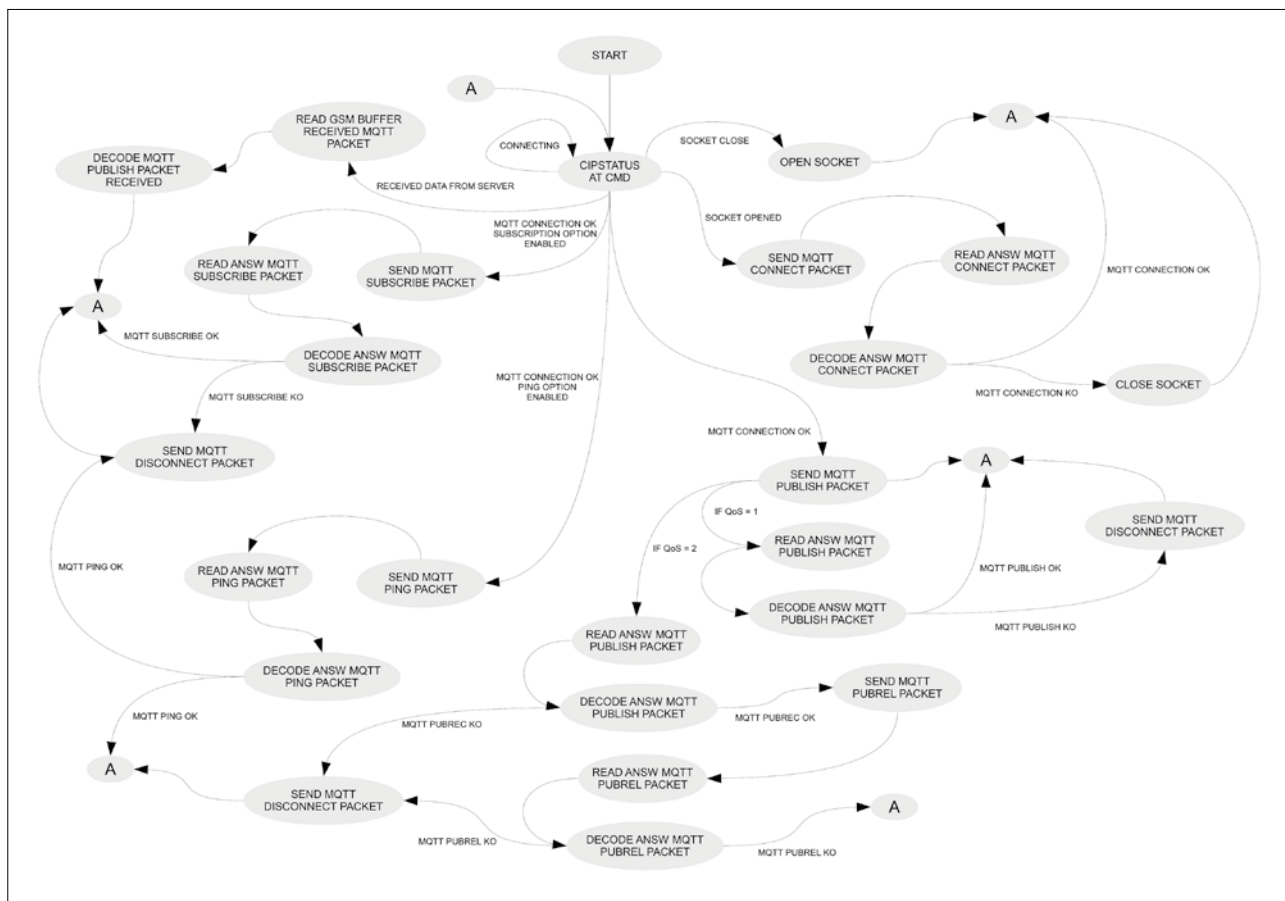


Fig. 8
Macchina a stati
di gestione dello
sketch.

La seconda funzione presente in "void loop()" richiama la macchina a stati di gestione dello sketch e quindi invio di comandi AT e gestione dei pacchetti MQTT di interesse. Questa sezione di codice può essere rappresentata con il diagramma di stato visibile nella **Fig. 8**. Come si può osservare, sono stati rappresentati i possibili stati in cui il codice può trovarsi a lavorare, alcuni dei quali sono attivi solo se abilitati come accennato all'inizio dell'articolo. Quindi il codice vede in testa l'invio del comando AT per verificare lo stato della connessione dati GPRS, se il Socket di connessione risulta chiuso il codice provvederà ad aprirlo connettendosi al server ThingSpeak tramite il comando AT+CIPSTART per SIMCOM o AT+QIDNSIP per QUECTEL passandogli gli opportuni parametri di connessione. Una volta aperto il Socket abbiamo che il sistema può passare all'invio del pacchetto MQTT CONNECT sempre passando per il test della connessione dati. Quindi il sistema invia il pacchetto dati al server ThingSpeak e, allo stato successivo, ne legge la risposta e relativa decodifica. Se il pacchetto MQTT ricevuto conferma l'avvenuta connessione il sistema può passare allo stato

successivo e cominciare a inviare i dati al server usando il pacchetto MQTT PUBLISH altrimenti si vede costretto a chiudere il Socket e ricominciare da capo. L'invio del pacchetto MQTT PUBLISH avviene ogni 60 secondi e viene fatto con "QoS" impostato a "0", di conseguenza nessuna risposta da parte del server come mostrato dal diagramma di stato. Il percorso lungo gli stati varia se si imposta "QoS" a "1"; in tal caso il sistema dopo l'invio del pacchetto MQTT PUBLISH si aspetta di ricevere dal SERVER il pacchetto di risposta MQTT PUBACK e quindi dovrà processare prima lo stato di lettura dei dati ricevuti e successivamente lo stato di decodifica degli stessi. Nel caso in cui la pubblicazione dei dati non vada a buon fine il sistema chiude la connessione tramite pacchetto MQTT DISCONNECT e ricomincia il ciclo da capo altrimenti prosegue al prossimo invio dati con cadenza 60 secondi. Se invece "QoS" è impostato a "2" si ha che l'invio dei dati tramite pacchetto MQTT PUBLISH richiede più passaggi come si vede dal diagramma. Infatti dopo l'invio del pacchetto MQTT il sistema deve leggere la risposta da parte del server e decodificarla, se OK procedere con l'invio del

CHE COS'È MQTT.FX

"MQTT.fx" è un software che si scarica dal sito ThingSpeak e serve principalmente per inviare i pacchetti MQTT PUBLISH o a ricevere i pacchetti stessi dopo un invio di richiesta di SUBSCRIBE al Server. Prima di inviare i pacchetti MQTT è necessario connettersi al relativo servizio MQTT.

La Fig. 9 mostra la schermata iniziale del software, dalla quale si vedono delle "cartelle o tabs" che identificano l'azione che si vuole fare, ad esempio "Publish" o "Subscribe". Mentre appena sotto il menù c'è una lista di possibili link di connessione e relativo pulsante per avviare la medesima nominato "Connect". Per creare una nuova connessione clicca- re sull'icona dell'ingranaggio, prima del pulsante "Connect", la quale richiamerà la finestra di configurazione come mostrato nella Fig. 10. Quindi si dovranno compilare i campi con gli opportuni parametri di connessione, come ad esempio il "Broker Address", "Broker Port" e via discorrendo. Nella sotto-cartella "User Credentials" si andranno a compilare i campi "User Name" e "Password" necessari per effettuare il collegamento al server. Tutti questi parametri sono dipendenti dal server a cui ci si conatterà e quindi variano di volta in volta.

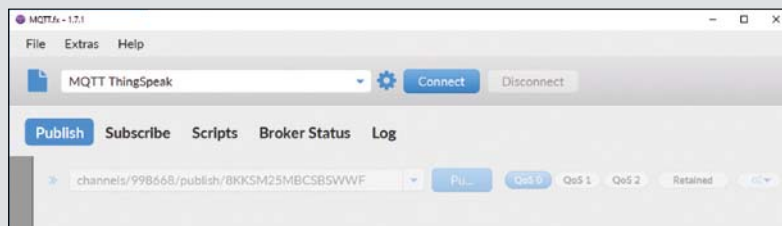


Fig. 9 - Schermata principale software "MQTT.fx".

Fig. 11 - Esempio di invio pacchetto MQTT PUBLISH.

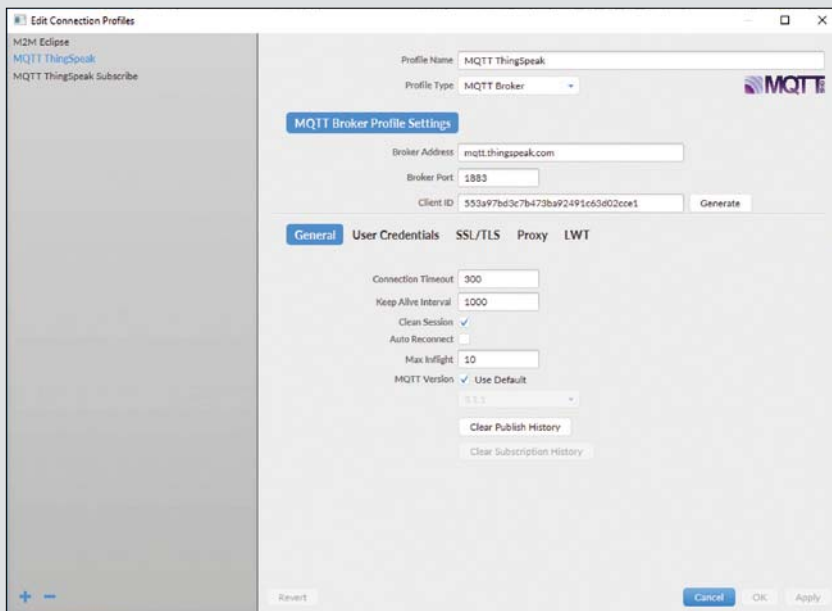
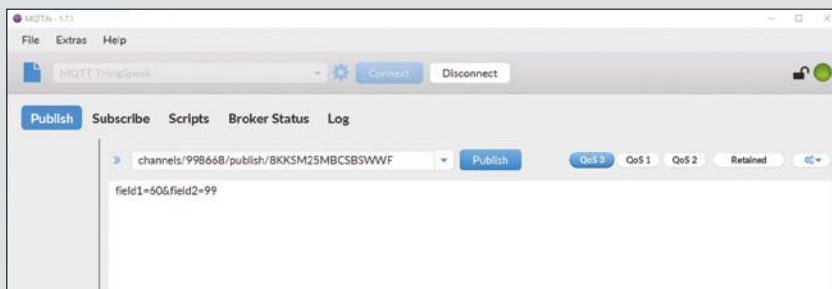


Fig. 10 - Schermata di configurazione connessione al Server.

Fatto ciò si può avviare la connessione cliccando sul pulsante "Connect". Se la connessione va a buon fine si attiveranno le cartelle "Publish", "Subscribe" ecc. A questo punto dovete inserire il "Topic" nella lista dei possibili "Topics" e compilare la "Text box" con le stringhe di testo contenenti i dati da inviare. Come esempio guardiamo la Fig. 11.

Il nostro "Topic" è "channels/998668/publish/8KKSM25MBCS5WVF" e dipende dall'applicazione e dal Server a cui ci si connette; mentre sotto nella "Text box" ci sono i dati da inviare in formato stringa ovvero "field1=60&field2=99". Questo identifica quali campi si vuole aggiornare e il valore da assegnare ad essi. Cliccare quindi sul pulsante "Publish", a fianco della lista dei "Topics", per inviare il pacchetto MQTT. Una volta inviato il pacchetto, se sullo Sketch c'è attiva la richiesta di SUBSCRIBE, si vedranno arrivare i dati appena inviati tramite un pacchetto MQTT PUBLISH inviato dal server stesso.



Listato 3

```
void CheckPublishDataFromServer(void) {
    if (TcpIp.TcpIpHandleReceivedDataFlag.Bit.DataReadyToRead == DATA_READY_TO_READ) {
        if (FutureStateSendCmd == CMD_ATQ_CIPSTATUS) {
            if (Isr.TimeOutWait > 0) {
                TcpIp.TcpIpHandleReceivedDataFlag.Bit.DataReadyToRead = 0;
                BckTimeOut = Isr.TimeOutWait;
                BckPastStateSendCmd = PastStateSendCmd;
                BckPresentStateSendCmd = PresentStateSendCmd;
                BckFutureStateSendCmd = FutureStateSendCmd;

                SetCmdToGetAnswer();
                FutureStateSendCmd = DECODE_MQTT_PUBLISH_RECEIVED;
                Isr.TimeOutWait = T_1SEC;
            }
        }
    }
}
```

Listato 4

```
void SetCmdToGetAnswer(void) {
    #if defined(SIMCOM_SIM800C) || defined(SIMCOM_SIM900) || defined(SIMCOM_SIM928A)
        TcpIp.SetCmd_AT_CIPRXGET(2, 0, 255);
    #endif
    #ifdef QUECTEL_M95
        TcpIp.SetCmd_AT_CIPRXGET(255);
    #endif
}
```

pacchetto MQTT PUBREL e attendere nuovamente risposta dal server e relativa decodifica della stessa. Se anche questo passaggio va a buon fine allora il sistema prosegue con l'invio del successivo pacchetto MQTT PUBLISH altrimenti chiude la connessione e ricomincia da capo il ciclo.

Ora vediamo cosa succede se è attivo il codice per l'invio del pacchetto MQTT PING (ricordarsi di togliere il commento alla riga di codice **"#define TEST_MQTT_PING_CMD"**). Il pacchetto MQTT PING viene inviato allo scadere del timer da 120 secondi a esso dedicato, ma prima deve terminare le azioni in corso. Quindi appena possibile il pacchetto MQTT PING verrà inviato restando poi in attesa di leggere e decodificare la risposta da parte del server. Se tutto è OK il sistema continua con le sue attività, altrimenti chiude la connessione e ricomincia da capo riconnettendosi e via di seguito. Infine, se anche la modalità SUBSCRIBE è attiva (de-commentare la riga di codice **"#define TEST_SUBSCRIBE_QOS_0"**), subito dopo la connessione al server MQTT avverrà l'invio della richiesta di SUBSCRIBE tramite apposito pacchetto MQTT come evidenziato dal diagramma di stato. Anche in questo caso il sistema attende la risposta dal

server e ne esegue la sua decodifica. Se è OK prosegue nelle sue attività, altrimenti ricomincia tutto da capo. L'invio del pacchetto MQTT SUBSCRIBE avviene una sola volta dopodiché il sistema attende che arrivino dei pacchetti MQTT PUBLISH da parte del server. L'intercettazione dei dati inviati dal server viene presa in carico dalla libreria GSM, la quale gestisce gli UNSOLICITED CODE. Ovviamente questo è possibile perché nel **Listato 2** vengono richiamate le due funzioni di libreria **"Gsm.UartContinuouslyRead()"** e **"Gsm.ProcessUnsolicitedCode()"**. La prima serve letteralmente per catturare le stringhe inviate in modo autonomo dal modulo GSM; questa funzionalità viene abilitata durante la configurazione dello stesso. La seconda, invece, decodifica i dati ricevuti dal modulo GSM; supporta una certa varietà di UNSOLICITED CODE ma non tutti. Di volta in volta che ne sorge la necessità, ne verranno aggiunti degli altri arricchendo le potenzialità della libreria.

I dati inviati dal Server, tramite pacchetto MQTT PUBLISH, vengono intercettati dalla libreria tramite la ricerca della stringa **"+CIPRXGET:"** per quanto riguarda SIMCOM e **"+QIRD:"** per quanto riguarda QUECTEL.



Intercettate tali stringhe e verificato i parametri ricevuti viene settato un flag che indica che ci sono dei dati nel buffer del modulo GSM che necessitano di essere letti. Il flag in questione è **"TcpIp.TcpIpHandleReceivedDataFlag.Bit.DataReadyToRead"** e viene gestito nel nostro sketch tramite il codice mostrato nel **Listato 3**. Come si può vedere, la funzione controlla che il flag menzionato prima sia settato a "1" e in caso positivo si assicura che la macchina a stati del nostro sketch, per lo stato futuro, sia impostata per l'invio del comando AT+CIPSTATUS. Se queste due condizioni sono soddisfatte, la funzione controlla che la variabile "TimeOutWait" sia maggiore di zero e, se anche questa condizione è soddisfatta, si procede con la lettura dei dati presenti nel buffer del modulo GSM; ma prima il sistema salva gli stati passato, presente e futuro della macchina a stati che saranno successivamente ripristinati dopo la lettura fuori sequenza dei dati presenti nel buffer del modulo GSM. La funzione che si occupa di iniziare la lettura dei dati si chiama **"void SetCmdToGetAnswer(void)"** ed è mostrata nel **Listato 4**. A seconda del modulo GSM usato verrà richiamata la giusta funzione per la lettura dei dati. In entrambi i casi si sta dicendo al modulo GSM che si vogliono leggere fino a un massimo di 255 byte. A seconda dei casi il valore può essere aumentato o diminuito. Ricordate che il valore da usare deve essere compatibile con il buffer usato nell'applicazione. In questo caso noi abbiamo maggiorato il buffer per la gestione della seriale fino a 512 byte. Quindi nelle due funzioni mostrate nel **Listato 4** si potrebbe anche dire di leggere fino a un massimo, ad esempio, di 300 byte o 400 byte a seconda di quanti dati ci si aspetta di ricevere dal server tramite pacchetto

MQTT PUBLISH. Un numero maggiore di byte potrebbe servire se invece di due campi da leggere ce ne fossero impostati ad esempio quattro e via scorrendo. In questi casi la stringa JSON ricevuta è più lunga dell'attuale e quindi non contenibile in soli 255 byte.

Eseguita questa funzione viene settato il nuovo stato futuro **"FutureStateSendCmd"** affinché possa puntare allo stato di decodifica dei dati ricevuti dal server tramite pacchetto MQTT PUBLISH (**"DECODE_MQTT_PUBLISH_RECEIVED"**).

Conclusa l'esecuzione della decodifica dei dati ricevuti il sistema ripristina le variabili dello stato passato, presente e futuro, riportando in linea il flusso del codice in esecuzione.

Ora analizziamo come sfruttare le funzioni messe a disposizione della libreria MQTT e usate in questo sketch. La prima funzione che analizziamo è quella che ci permette di instaurare una connessione al server MQTT ovvero il pacchetto MQTT CONNECT. La funzione che viene richiamata dallo sketch è **"uint8_t SetMqttConnectPacket(void)"** ed è mostrata nel **Listato 5**. Come si può osservare, in testa alla funzione vengono settati una serie di Flags tra cui quelli che indicano alla libreria che sono necessari **"UserName"** e **"Password"** per accedere al server. Tali stringhe sono già state lette dalla EEPROM e memorizzate in SRAM, nelle appropriate strutture dati, e pronte all'utilizzo da parte della libreria MQTT.

Quindi una volta ripulito il buffer seriale si procede al richiamo della funzione **"Mqtt.Packets_CONNECT"** alla quale si passa come unico parametro l'indirizzo di partenza dell'array usato.

Quanto appena fatto però non è ancora sufficiente perché abbiamo solo riempito un buffer con il pacchetto MQTT CONNECT, quello che manca

Listato 5

```
uint8_t SetMqttConnectPacket(void) {
    uint8_t Index;

    Mqtt.ConnectFlags.Flag.FlagByte      = 0x00;
    Mqtt.ConnectFlags.Flag.Bit.UserName   = 1;
    Mqtt.ConnectFlags.Flag.Bit.Password   = 1;
    Mqtt.ConnectFlags.Flag.Bit.CleanSession = 1;
    Mqtt.KeepAlive = 1000;

    Gsm.ClearBuffer(&Gsm.GSM_Data_Array[0], (sizeof(Gsm.GSM_Data_Array)/sizeof(Gsm.GSM_Data_Array[0])));
    Mqtt.Packets_CONNECT(&Gsm.GSM_Data_Array[0]);
    Index = Mqtt.MessageLen;
    Gsm.GSM_Data_Array[Index++] = ASCII_SUB;
    return(Index);
}
```



ancora è l'azione di invio del pacchetto tramite comando AT apposito. Quindi ci vengono in aiuto le due funzioni di libreria `"TcpIp.SetCmd_AT_CIPSEND(false, 0)"` e `"TcpIp.SetCmd_AT_CIPSEND_Step2(SetMqttConnectPacket())"`; la prima viene richiamata allo stato `"CMD_AT_CIPSEND_MQTT_CONN_STP_1"` mentre la seconda allo stato `"CMD_AT_CIPSEND_MQTT_CONN_STP_2"`. Spiando la comunicazione seriale vediamo quanto segue:

```
AT+QISEND<CR><LF>
<CR><LF>
> <DLE>E<NUL> <EOT>MQTT<EOT>A<ETX>è<NUL> <DC3>Device
Disconnected<NUL> <DC2>mdestro@tiscali.it<NUL> <DLE>H
2TRGZ4L3Y140GYE<SUB><CR><LF>
SEND OK<CR><LF>
```

Quindi come primo step abbiamo l'invio del comando AT `"AT+QISEND"` seguito, dall'invio dei dati che compongono il pacchetto MQTT CONNECT. Il carattere `">"` indica che il modulo GSM è pronto a ricevere i dati da inviare. Quindi si capisce che i pacchetti MQTT sono tutti inviati tramite il comando AT+QISEND (QUECTEL) o AT+CIPSEND (SIMCOM) ed entrambi richiedono due step per essere completati. Lo sketch oltre a gestire il pacchetto MQTT CONNECT gestisce, ovviamente, anche il pacchetto DISCONNECT il quale viene configurato dalla funzione mostrata nel **Listato 6**.

In questo caso non ci sono parametri da configurare, è sufficiente richiamare la funzione di libreria per il pacchetto MQTT DISCONNECT con indicato l'indirizzo in memoria da cui iniziare a salvare i dati. Stesso discorso di prima per l'invio dei dati tramite comando AT. Veniamo ora al pacchetto MQTT PUBLISH il quale viene gestito dalla funzione `"uint8_t SetMqttPublishPacket(void)"` la quale, oltre a richiamare la funzione di libreria MQTT appropriata, deve occuparsi di caricare i dati nell'array `"Mqtt.PublishPayload"` mentre l'array `"Mqtt.`

`PublishTopicName"` viene caricato all'inizio dello sketch grazie ai dati memorizzati in EEPROM come visto all'inizio del paragrafo. Il **Listato 7** mostra la funzione per la preparazione e generazione del pacchetto MQTT PUBLISH.

Quindi la parte iniziale del codice si occupa di gestire l'array per il payload mentre la seconda parte configura i flag di gestione e richiama la funzione di libreria per la creazione del pacchetto MQTT PUBLISH. Detto ciò possiamo ora alla funzione che ci permette di creare il pacchetto MQTT SUBSCRIBE da inviare al server per la richiesta dati. Questo viene fatto richiamando la funzione `"uint8_t SetMqttSubscribePacket(void)"` visibile nel **Listato 8**. La parte in cima al codice si occupa di settare dei flag necessari alla generazione del pacchetto; dato che ci sono fino a un massimo di quattro `"Topic Filter"` gestibili dalla libreria bisogna indicare alla struttura dati quale dei quattro si vuole usare settando a `"1"` il flag di abilitazione `"EnFilter"`. Per puntare alla struttura dati corretta ci pensa la costante `"SUBSCRIBE_TOPIC_FILTER_USED"` la quale viene definita e impostata all'inizio dello sketch tramite direttiva al compilatore `"#define"`. I valori ammessi sono 0, 1, 2 e 3. In questo esempio si è scelto il formato `"JSON"` e quindi il valore da assegnare alla costante è 3. Gli altri valori definiscono i seguenti `"Topic Filter"`:

- `"0"` - Subscribe `"Field 1"`
`"channels/998668/subscribe/fields/field1/PODDI37ZU2FS1W5D"`
- `"1"` - Subscribe `"Field 2"`
`"channels/998668/subscribe/fields/field2/PODDI37ZU2FS1W5D"`
- `"2"` - Subscribe `"All Fields"`
`"channels/998668/subscribe/fields/+PODDI37ZU2FS1W5D"`
- `"3"` - Subscribe `"Json Format"`
`"channels/998668/subscribe/json/PODDI37ZU2FS1W5D"`

Listato 6

```
uint8_t SetMqttDisconnectPacket(void) {
    uint8_t Index;

    Gsm.ClearBuffer(&Gsm.GSM_Data_Array[0], (sizeof(Gsm.GSM_Data_Array)/sizeof(Gsm.GSM_Data_Array[0])));
    Mqtt.Packets_DISCONNECT(&Gsm.GSM_Data_Array[0]);
    Index = Mqtt.MessageLen;
    Gsm.GSM_Data_Array[Index++] = ASCII_SUB;
    return(Index);
}
```

⬇ Listato 7

```
uint8_t SetMqttPublishPacket(void) {
    uint8_t Index = 0;

    memset(Mqtt.PublishPayload, 0, (sizeof(Mqtt.PublishPayload)/sizeof(Mqtt.PublishPayload[0])));
    Gsm.ReadStringFLASH((uint8_t *)STR_MQTT_FIELD_1, ((uint8_t *)Mqtt.PublishPayload), strlen(STR_MQTT_
FIELD_1));
    Index = strlen(Mqtt.PublishPayload);
    Index += Gsm.ConvertNumberToString(FieldValue, ((uint8_t *)Mqtt.PublishPayload) + Index), 0);
    Mqtt.PublishPayload[Index++] = ASCII_AMPERSAND;
    FieldValue += 10;
    Gsm.ReadStringFLASH((uint8_t *)STR_MQTT_FIELD_2, ((uint8_t *)Mqtt.PublishPayload + Index), strlen(STR_
MQTT_FIELD_2));
    Index = strlen(Mqtt.PublishPayload);
    Index += Gsm.ConvertNumberToString(FieldValue, ((uint8_t *)Mqtt.PublishPayload) + Index), 0);
    FieldValue += 10;
    Mqtt.PublishPayload[Index++] = ASCII_AMPERSAND;
    Gsm.ReadStringFLASH((uint8_t *)STR_STATUS_MQTT_PUB, ((uint8_t *)Mqtt.PublishPayload + Index),
strlen(STR_STATUS_MQTT_PUB));

    Mqtt.PublishFlags.Flag.FlagByte = 0x00;
    Mqtt.PublishFlags.Flag.Bit.Retain = 0;          // Set Retain bit to zero
    Mqtt.PublishFlags.Flag.Bit.QoS = 0;            // Set QoS bit to zero. ThingSpeak work only with QoS = 0
    Mqtt.PublishFlags.Flag.Bit.Dup = 0;           // Set Dup bit to zero
    if (Mqtt.PublishFlags.Flag.Bit.QoS > 0) {
        Mqtt.PacketIdentifier = PACKET_IDENTIFIER; // Set packet identifier
    }
    Gsm.ClearBuffer(&Gsm.GSM_Data_Array[0], (sizeof(Gsm.GSM_Data_Array)/sizeof(Gsm.GSM_Data_Array[0])));
    Mqtt.Packets_PUBLISH(&Gsm.GSM_Data_Array[0]);
    Index = Mqtt.MessageLen;

    Gsm.GSM_Data_Array[Index++] = ASCII_SUB;
    return(Index);
}
```

Tutti i **“Topic Filter”** sono memorizzati in EEPROM e pre-caricati all’inizio dello sketch nelle apposite strutture dati. L’ultimo parametro da configurare è **“QoS”** che in questo caso va settato obbligatoriamente a **“0”** in quanto ThinkSpeak non supporta le altre modalità. In fondo al codice abbiamo quindi

la chiamata alla funzione che si occupa di generare il pacchetto MQTT e che necessita come al solito del solo parametro che indica da quale indirizzo di memoria iniziare a salvare i dati del pacchetto. Infine la funzione per la gestione del pacchetto MQTT PING, ovvero **“uint8_t**

⬇ Listato 8

```
uint8_t SetMqttSubscribePacket(void) {
    uint8_t Index = 0;

    // Enable Topic filter 1; array index = 0 (Max number of “Topic Filter” supported by library is 4.
    Array index from 0 to 3)
    Mqtt.Subscribe[SUBSCRIBE_TOPIC_FILTER_USED].Flags.FlagByte = 0x00;
    Mqtt.Subscribe[SUBSCRIBE_TOPIC_FILTER_USED].Flags.Bit.EnFilter = 1;
    Mqtt.Subscribe[SUBSCRIBE_TOPIC_FILTER_USED].QoS = MQTT_SUB_QOS_0; // Set QoS = 0
    for Subscribe command
        Mqtt.PacketIdentifier = PACKET_IDENTIFIER; // / /
    Set packet identifier

    Gsm.ClearBuffer(&Gsm.GSM_Data_Array[0], (sizeof(Gsm.GSM_Data_Array)/sizeof(Gsm.GSM_Data_Array[0])));
    Mqtt.Packets_SUBSCRIBE(&Gsm.GSM_Data_Array[0]);
    Index = Mqtt.MessageLen;

    Gsm.GSM_Data_Array[Index++] = ASCII_SUB;
    return(Index);
}
```


Listato 9

```
uint8_t SetMqttPingPacket(void) {
    uint8_t Index = 0;

    Gsm.ClearBuffer(&Gsm.GSM_Data_Array[0], (sizeof(Gsm.GSM_Data_Array)/sizeof(Gsm.GSM_Data_Array[0])));
    Mqtt.Packets_PINGREQ(&Gsm.GSM_Data_Array[0]);
    Index = Mqtt.MessageLen;

    Gsm.GSM_Data_Array[Index++] = ASCII_SUB;
    return(Index);
}
```

SetMqttPingPacket(void) la quale è mostrata nel **Listato 9**. Come si vede dal codice, non ci sono parametri da impostare è sufficiente richiamare la funzione di libreria **uint8_t SetMqttPingPacket(void)** passandogli come parametro l'indirizzo in memoria da cui iniziare a scrivere.

Lo sketch utilizza i LED presenti sulla scheda GSM demo per evidenziare l'invio dei pacchetti MQTT e in particolare abbiamo queste associazioni:

- MQTT CONNECT → Led rosso LD4
- MQTT PUBLISH → Led rosso LD5
- MQTT SUBSCRIBE → Led giallo LD6
- MQTT PING → Led giallo LD7

Così diventa evidente a colpo d'occhio quando il codice sta iniziando l'invio di un pacchetto MQTT verso il server ThingSpeak.

Facciamo un'ultima considerazione sulle tre variabili per la gestione degli stati nominate **"PastStateSendCmd"**, **"PresentStateSendCmd"** e **"FutureStateSendCmd"**. La variabile **"PresentStateSendCmd"** serve per indicare alla macchina a stati lo stato che si vuole eseguire nel momento in cui si richiama la funzione **"void ProcessSM_MqttPacket_AT_Cmd(void)"** mentre la variabile **"FutureStateSendCmd"** indica lo stato futuro che la macchina a stati dovrà eseguire al passo successivo. Tale variabile viene settata ogni volta che si conclude l'elaborazione di uno stato. Alla fine di alcuni stati è anche necessario impostare lo stato passato **"PastStateSendCmd"** necessario al codice per prendere alcune decisioni quando la variabile **"PresentStateSendCmd"** è settata sullo stato **"CMD_ATQ_CIPSTATUS_ANSW"**. Infatti tutte le volte che si testa la connessione di rete TCP/IP al fine di essere sicuri di essere ancora connessi al server il sistema, solo in questo frangente, deve prendere una decisione su quale stato futuro deve andare e per fare ciò ha bisogno di sapere da

quale stato passato è arrivato grazie alla variabile **"PastStateSendCmd"**. In questo modo si riescono a gestire tutte le situazioni compresi gli errori dovuti a pacchetti MQTT non andati a buon fine. Con questo riteniamo conclusa la trattazione dello sketch di interazione con il server ThingSpeak tramite pacchetti MQTT.

CONCLUSIONI

Con questa abbiamo concluso con l'implementazione del protocollo nella nostra libreria GSM. Data l'importanza del protocollo MQTT, oggi sono disponibili moduli GSM con integrate le relative funzionalità, tuttavia la nostra implementazione rimane comunque valida e utile alla realizzazione di applicazioni di ogni genere.

Detto ciò non escludiamo che in futuro, integrando nuovi moduli GSM nella nostra libreria, sia possibile utilizzare anche i comandi AT per gestione pacchetti MQTT integrati nei moduli GSM di nuova generazione. □



Cosa occorre?

Lo shield universale GSM (cod. WWGSMESHIELD) è in vendita presso Futura Elettronica a Euro 54,90. Viene fornito montato ad esclusione degli strip line (compresi) che vanno saldati manualmente. Lo shield si può interfacciare con uno dei seguenti moduli GSM: interfaccia con M95 (cod. FT1128M) disponibile a Euro 29,00, modulo cellulare miniaturizzato con SIM800 (cod. FT1308M) in vendita a Euro 29,00.

Il materiale va richiesto a:

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





IMPARA L'ELETTRONICA FACILMENTE CON TOTEM MINI LAB



€ 112,00
cod. TOTEMMINILAB

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

COMPLETALO CON LE SCHEDE DI ESPANSIONE!



ESPANSIONE AUDIO

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

€ 39,50
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.

€ 39,50
Cod. TOTEMSENSORSIDE

OSCILLOSCOPIO 2 CANALI 70 MHZ

- ✓ Frequenza di campionamento fino a 1 GS/s
- ✓ Funzione FFT integrata
- ✓ Software per analisi in tempo reale con il PC
- ✓ Porta USB Host

cod. DS072

€ 319,00



Disponibile anche
nella versione a 200MHz

cod. DS0202

€ 459,00

OSCILLOSCOPIO 2 CANALI 100 MHZ CON GENERATORE DI FORME D'ONDA



cod. DS0F6102

€ 429,00

- ✓ Frequenza di campionamento fino a 1 GS/s
- ✓ Funzione FFT integrata
- ✓ Software per analisi e controllo in tempo reale con il PC
- ✓ Porta USB Host / USB Device

Disponibile anche
nella versione a 200MHz

cod. DS0F6202

€ 549,00

GENERATORE DI FUNZIONI E FORME D'ONDA ARBITRARIE 2 CANALI 30 MHZ

- ✓ Frequenza onda DC 30 MHz
- ✓ 2 canali
- ✓ Frequenza di campionamento 250 MSa/s
- ✓ Risoluzione verticale 16 Bit

cod. HDG32B

€ 439,00



Disponibile anche
nella versione a 60MHz

cod. HDG62B

€ 510,00

OSCILLOSCOPIO PORTATILE 2 CANALI - 40 MHZ CON MULTIMETRO

Dispone di ampio display LCD TFT a colori da 2,8. La larghezza di banda è di 40 MHz, 250 MSa/s di velocità di campionamento in tempo reale a canale singolo. Il multimetro fornisce misurazioni di tensione continua e alternata, corrente continua e alternata, resistenze, capacità, test diodi e continuità. È interfacciabile a PC tramite software scaricabile gratuitamente. Supporta Windows 7 / 8 / 10 a 32 e 64 bit.



cod. MH02C42

€ 178,00

Disponibile anche nella versione
a 2 canali 70 MHz con generatore
di forme d'onda e multimetro

cod. MH02D72GEN

€ 268,00

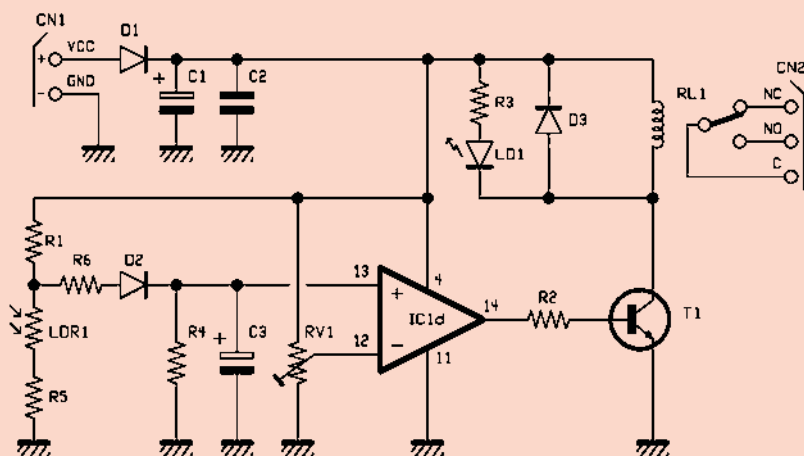
INTERRUTTORE CREPUSCOLARE

di DAVIDE
SCULLINO



Permette di accendere e spegnere l'illuminazione elettrica, o altri utilizzatori, in base alle condizioni di luce ambientale.

Nella pubblica illuminazione, nei piazzali, nelle aree aperte illuminate, sappiamo quanto sia utile che le lampade vengano gestite da un automatismo che le accende all'ora giusta e le spenga a tempo, ma anche e soprattutto un sistema in grado di accendere l'illuminazione quando le condizioni di luce ambientale lo richiedono, spegnendole al mattino quando il sole illumina nuovamente. Questo qualcosa è l'interruttore crepuscolare, cosiddetto perché si attiva al crepuscolo accendendo le lampade quando la luce del sole o comunque del giorno, non basta più. L'interruttore crepuscolare è un circuito classico dell'elettronica e non "tramonterà" mai, almeno fin quando al giorno seguirà la notte. Di crepuscolari ne abbiamo già proposti in passato, ma trattandosi di qualcosa che serve sempre ritor-



niamo sull'argomento proponendone uno nuovo, in queste pagine, la cui realizzazione sarà insieme un'occasione per affidare all'automazione qualcosa che altrimenti si farebbe manualmente e un momento di ripasso dell'elettronica tradizionale: quella, per intendersi, di prima che le logiche programmabili avessero il sopravvento. Il nostro circuito è sostanzialmente un interruttore sensibile alla luce e può essere utilizzato non solo per controllare lampade ma anche per fornire un segnale elettrico a un sistema domotico o per aprire o chiudere tapparelle e coperture motorizzate, che ad esempio vanno aperte in presenza della luce e chiuse al buio o viceversa. Dunque, lasciamo i preliminari e andiamo subito a vedere come è fatto il nostro circuito, dando

Il crepuscolare inserito nel contenitore.



per prima cosa uno sguardo allo schema elettrico corrispondente.

SCHEMA ELETTRICO

Si tratta di qualcosa di davvero semplice: un amplificatore operazionale montato come comparatore ed una fotoresistenza che ci serve a rilevare il livello di illuminazione nell'ambiente; completa il circuito l'attuatore, che nel nostro caso è un piccolo relé. Per rilevare l'illuminazione ci serviamo del fotoresistore siglato LDR1, il quale presenta massima resistenza al buio (circa 1 Mohm) e la minima (alcune centinaia di ohm) in corrispondenza dell'esposizione ad una forte intensità luminosa; questa prerogativa ci permette di rilevare il livello d'illuminazione dell'ambiente sulla base del valore resistivo assunto dal componente, ovvero, inserendo il fotoresistore in un partitore di tensione, di farlo riferendoci alla tensione ottenuta in uscita da quest'ultimo. L'uso di un partitore ci consente di usare un comparatore con il quale definiamo la soglia di tensione, corrispondente ad un certo valore di luminosità, in corrispondenza del quale il relé deve essere eccitato. L'inserimento di un trimmer RV1 nella rete di applicazione della tensione di riferimento del comparatore ci lascia liberi di definire il livello di luminosità al quale il crepuscolare deve entrare in funzione. Ma vediamo il funzionamento nel dettaglio, ipotizzando di partire dalla condizione di totale oscurità; in questo caso la resistenza assunta da LDR1 è molto più elevata di quella di R1 ed R5, quindi la tensione presente tra il nodo formato da essa con R1 ed R6 è circa uguale a quella che si rileva a valle del diodo D2 e perciò la stessa che si presenta all'ingresso non-invertente dell'operazionale IC1d contenuto nell'LM324 siglato IC1. Se il cursore del trimmer RV1 è lontano dalla linea positiva (ossia dal catodo del D2) la tensione presente sull'ingresso invertente dell'operazionale è inferiore a quella localizzata sul non-invertente, quindi l'uscita dell'IC1d si porta a livello logico alto e polarizza in base T1, il cui collettore conduce corrente ed alimenta simultaneamente la bobina del relé ed il bipolo R3/LD1, facendo illuminare il LED (segnalando così l'attivazione dell'uscita del crepuscolare) ed eccitare RL1. Lo scambio di quest'ultimo si chiude tra i contatti C ed NO, chiudendo il circuito dell'utilizzatore ad essi collegato. Quando la luce nell'ambiente aumenta, la tensione portata da R6 e D3 al piedino 5 dell'IC1d comincia a scendere, a causa del fatto che la resistenza del fotoresistore prende a calare

progressivamente, in relazione con l'intensità della luce che ne colpisce la superficie sensibile; ad un certo punto l'ingresso non-invertente si trova ad un potenziale inferiore a quello portato sull'invertente dal cursore del trimmer RV1 ed il comparatore inverte lo stato della propria uscita, la quale passa a livello basso e lascia interdire il transistor T1. Ora il LED si spegne e l'equipaggio mobile del relé ricade. Se la luminosità nell'ambiente scende di nuovo, il piedino 14 dell'U1 torna a livello alto e RL1 si eccita nuovamente (inoltre il LED si riaccende).

Il punto in cui il relé viene eccitato ed il LED acceso si decide mediante il trimmer RV1: portando il cursore di questo componente verso massa si riduce la tensione in corrispondenza della quale il comparatore torna a riposo e quindi è richiesto che ci sia molta luce nell'ambiente per disattivare il relé, mentre, al contrario, andando verso il catodo del diodo D1 la tensione che il piedino 5 deve raggiungere cresce e per innescare il relé il fotoreistore deve presentare valori di resistenza più elevati e quindi occorre che faccia più buio. Del circuito del comparatore notate che il diodo D2 (serve a portare il potenziale dalla R6 all'operazione, evitando che C3 si scarichi attraverso essa) è stato inserito per realizzare, insieme al condensatore C3, una sorta di rete anti-pendolarismo indispensabile ad evitare sia che il comparatore commuti al verificarsi di una brevissima variazione di luce (dovuta ad esempio al sorvolo

di un uccello o al passaggio di una persona o di un'automobile) sia che nel passaggio dal buio alla luce e viceversa il relé cominci a trillare perché il comparatore commuta ripetutamente in quanto il valore resistivo assunto dal fotoreistore oscilla nell'intorno di quello che determina la commutazione. Quest'ultima situazione si potrebbe anche evitare retroazionando IC1d in positivo, realizzando così un circuito ad isteresi (ossia con due diverse soglie di commutazione) ma in questo caso abbiamo optato per il comparatore normale, filtrando la tensione fornita dal partitore comprendente il fotoreistore mediante una rete RC. Detto ciò non ci resta che vedere il circuito di alimentazione, composto dal diodo D1 (che protegge dall'inversione di polarità ai morsetti di ingresso) posto a valle della morsettiera di alimentazione e dai condensatori C1 e C2, scopo dei quali è filtrare l'alimentazione, in special modo se viene prelevata da un alimentatore da rete. Il circuito richiede, per funzionare, una tensione continua, meglio se stabilizzata (altrimenti il comparatore può oscillare in prossimità della tensione di soglia, malgrado la rete RC di filtro) di valore compreso tra 9 e 12 volt; la corrente richiesta è dell'ordine dei 30 milliampere, grazie anche all'adozione per RL1 di un relé sub-miniatura la cui bobina assorbe circa 15 mA.

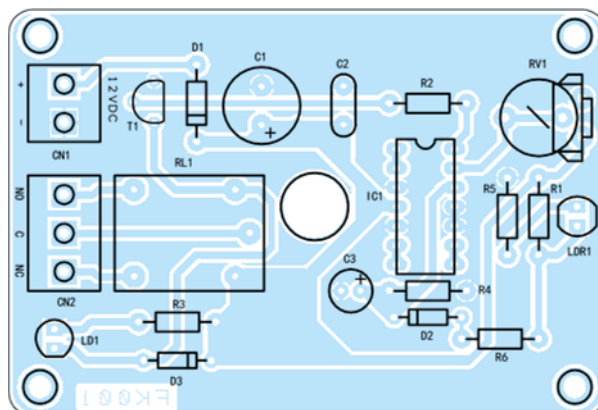
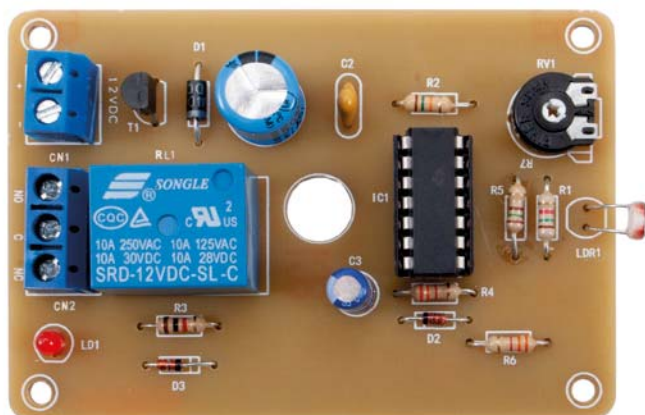
Concludiamo lo studio della schema elettrico con il diodo D2, che si trova in antiparallelo alla bobina del RL1 e che quindi, in condizioni normali

CARATTERISTICHE TECNICHE

-  **Tensione di alimentazione:**
12÷15 Vcc
-  **Corrente assorbita:**
30 mA
-  **Tensione commutabile in continua:**
30 Vdc
-  **Tensione commutabile in alternata:**
250 Vac
-  **Corrente commutabile:**
10 A
-  **Soglia di luce regolabile**
-  **Sensore a fotoresistenza**



Il circuito
completato.



Elenco Componenti:

R1: 1K5 Ohm
R2: 150 KOhm
R3: 1 KOhm
R4,R6: 3K3 Ohm
R5: 150 ohm
RV1: 1 KOhm
C1: 470 μ F 25VL elettrolitico
C2: 100 nF multistrato

C3: 100 μ F 25VL
elettrolitico
D1: 1N4007
D2, D3: 1N4148
LDR1: Fotoresistenza
LD1: led rosso 3 mm
T1: BC547
U1: LM324

RL1: relé 12 VL
1 scambio

Varie:
- zoccolo 7 + 7
- morsetto 3 poli
- morsetto 2 poli
- stampato 78,5x54 mm

è interdetto; questo componente serve quando il transistor, interdicendosi, interrompe la corrente nella bobina mobile del relé, allorché a causa della natura induttiva delle induttanze, la stessa reagisce generando un'extratensione inversa che, se non venisse soppressa proprio dal fatto che il diodo, polarizzato all'inverso, la cortocircuita, andrebbe a danneggiare la giunzione di collettore del T1.

REALIZZAZIONE PRATICA

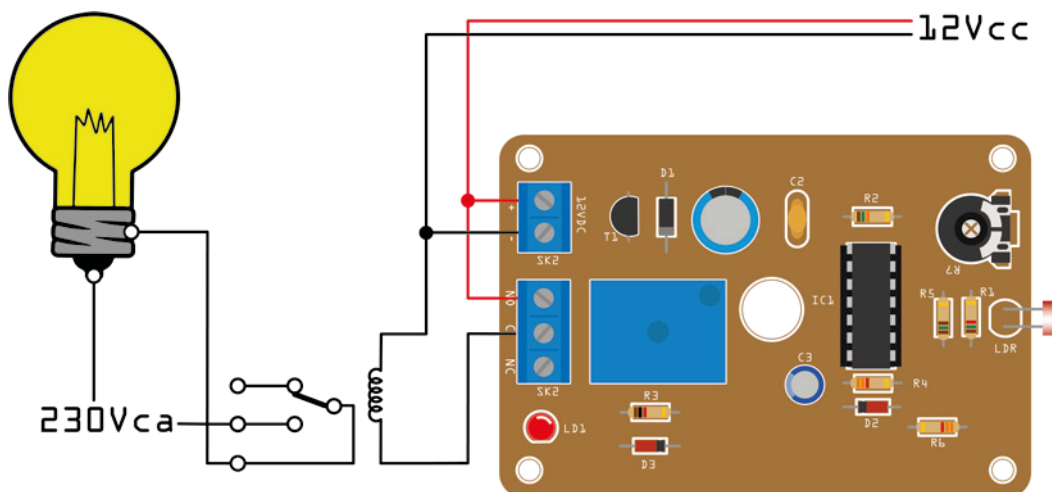
Passiamo adesso alla costruzione del crepuscolare, per il quale abbiamo previsto un circuito stampato monofaccia che ospiterà tutti i compo-

nenti; la basetta va realizzata preferibilmente per fotoincisione dopo averne scaricato dal nostro sito Internet www.elettronica.it la traccia lato rame (la trovate in formato file immagine insieme agli altri file di questo numero della rivista). Incisa e forata, sarà pronta per accogliere i vari elementi, che consigliamo di montare in ordine di altezza rammentando che resistenze e diodi devono essere posizionati in verticale, secondo le indicazioni del piano di montaggio illustrato in queste pagine. Sistemati resistenze, condensatori non polarizzati e diodi (LED compreso, il cui lato smussato deve essere rivolto al lato vicino dello stampato) collocate lo zoccolo per l'LM324, il pic-

Fig. 1
Il contenitore forato per rendere accessibili fotoresistenza e morsettiere.



schema di CABLAGGIO



colo relé ed il transistor BC547, che va orientato con il lato piatto rivolto proprio al RL1; passate quindi agli elettrolitici ed al trimmer (che va montato in verticale) e concludete con le morsettiere a passo 2,54 mm per l'alimentazione del circuito ed il collegamento dell'utilizzatore allo scambio del relé. Infine, potete inserire l'LM324 nel proprio zoccolo, badando che la sua tacca di riferimento sia rivolta al lato vicino del circuito stampato. La fotoresistenza può essere sistemata sul circuito stampato (non ha polarità, quindi potete collegarla come preferite) da una qualsiasi delle due facce o montata all'esterno collegandola con due spezzoni di filo in guaina, purché non più lunghi di due o tre metri. Il circuito stampato, una volta completato, può essere inserito in un contenitore in plastica, rendendo accessibile attraverso una finestrella il fotoresistore e ricavando un'apertura per il passaggio dei fili elettrici di collegamento del carico e dell'alimentazione (**Fig. 1**). Nel nostro prototipo abbiamo utilizzato un contenitore Kradex KZ23, reperibile da Futura Elettronica (anche on-line su www.futurashop.it) con il codice prodotto 5180-KZ23.

Per l'installazione ricordate che la fotoresistenza non deve mai essere rivolta verso i corpi illuminanti (le lampade) ma piuttosto verso il basso, perché altrimenti all'accendersi delle luci verrà inevitabilmente illuminata e il crepuscolare sarà disturbato al punto da far ricadere il relé, il qual poi tornerà attivato dal buio e si innescherà un pendolarismo, ossia una pulsazione ritmica del relé con la conseguente pulsazione delle lampade.

Per l'utilizzo pratico ricordate che il relé da noi usato permette di commutare correnti fino a 500 mA in circuiti funzionanti a non più di 120 Vcc o 100 Vca; per comandare una lampadina o un carico elettrico funzionante con la tensione di rete A 230 Vca dovete servirvi di un servo relé da 12V far passare dallo scambio (contatti C ed NO) la tensione di alimentazione del circuito (prelevata da Vcc) in modo da alimentare la bobina del servo relé quando RL1 si chiude. Lo scambio del servo relé permetterà quindi di alimentare l'utilizzatore. Se con il crepuscolare intendete comandare una tapparella motorizzata che deve chiudersi automaticamente alla sera e riaprirsi al ritorno del sole, usate i contatti C ed NO come interruttore; la tapparella dovrà prevedere un microswitch o un sensore di presenza per evitare di schiacciare chi si trova eventualmente sul passaggio. ■



Cosa occorre?

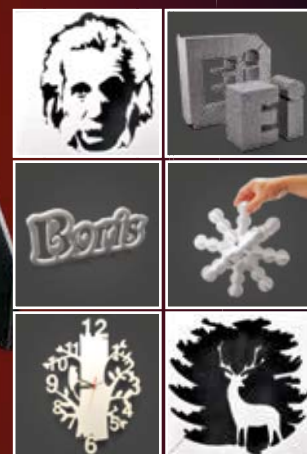
I componenti utilizzati in questo progetto sono facilmente reperibili. Il circuito crepuscolare (cod. FK001) è disponibile in scatola di montaggio a Euro 8,00, il contenitore previsto (cod. KZ23) è in vendita a Euro 2,10 e deve essere opportunamente fresato per ospitare il circuito crepuscolare.

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>**

DAI FORMA ALLE TUE IDEE!



cod. FOAMCUTTER

€ 320,00

PLOTTER TAGLIA POSTIROLO - IN KIT

Realizza, intagliandole a caldo, sagome di ogni genere partendo da lastre di polistirolo, polistirene ed altri materiali espansi a bassa temperatura di fusione. Collegalo subito al PC e utilizzalo per realizzare tutto quello che vuoi: modelli di vario genere, scritte, scatole, sagome per decorazioni, lettere scatolate, loghi, simboli, puzzle 3D e molto altro!

CARATTERISTICHE TECNICHE:

- **Tecnica di taglio:** a filo caldo
- **Materiali lavorabili:** espansi a bassa temperatura di fusione
- **Dimensioni:**
 - **Plotter:** 775x690x220 mm
 - **Area lavorabile:** 500x480 mm
 - **Spessore lavorabile:** 120 mm
- **Risoluzione:** 0,5 mm
- **Tolleranza X Y:** 0,012 mm
- **Velocità di taglio:** 300 mm/minuto (varia in funzione della tipologia di materiale lavorato)
- **Alimentazione:** 12 VDC / 3 A
- **Temperatura filo:** regolabile da software
- **Gestione:** da PC tramite interfaccia USB

Prezzo IVA inclusa.

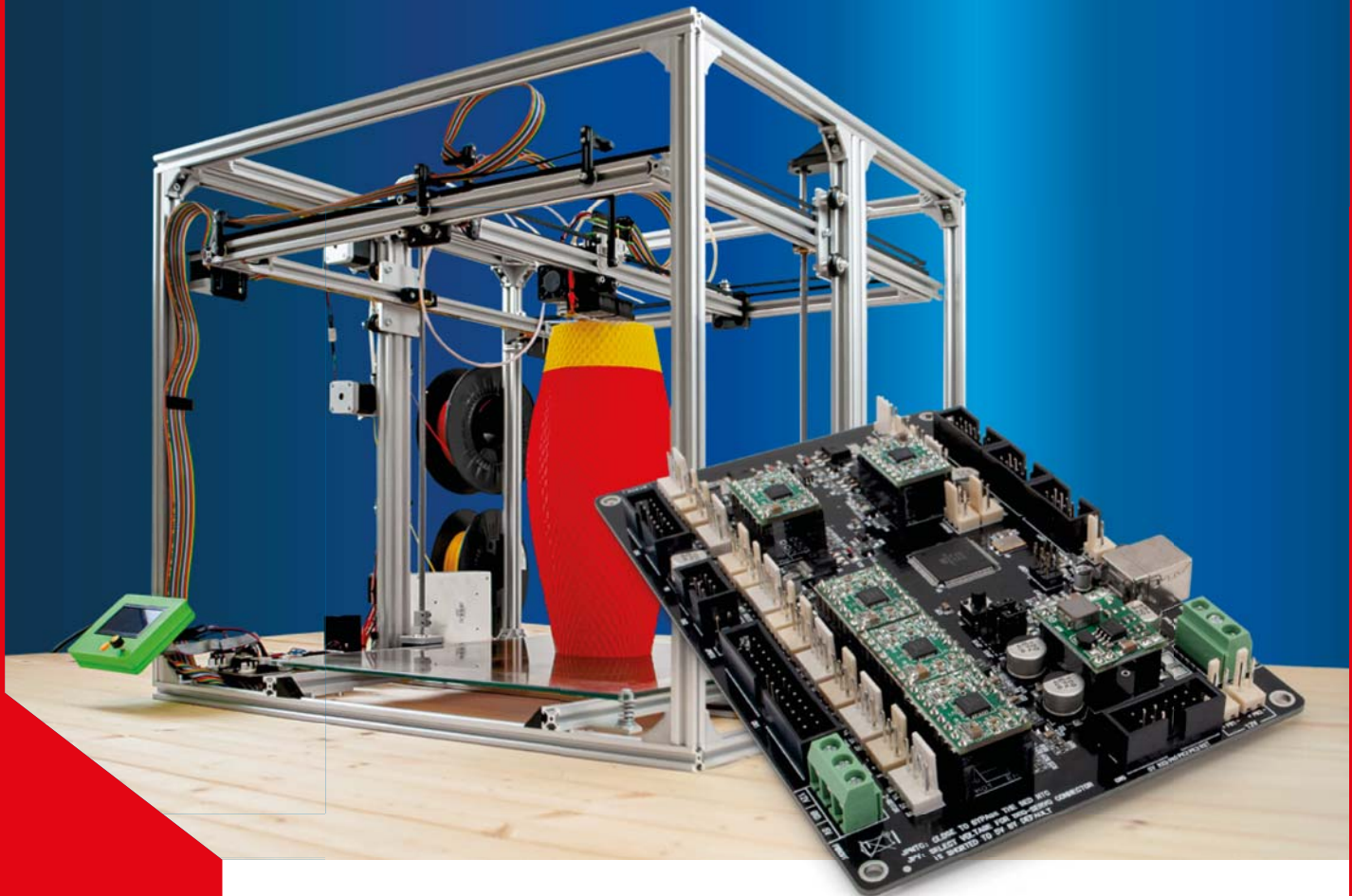
**FUTURA
ELETTRONICA®**

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

www.futurashop.it

CONTROLLER 3D4040

DUAL EXTRUDER

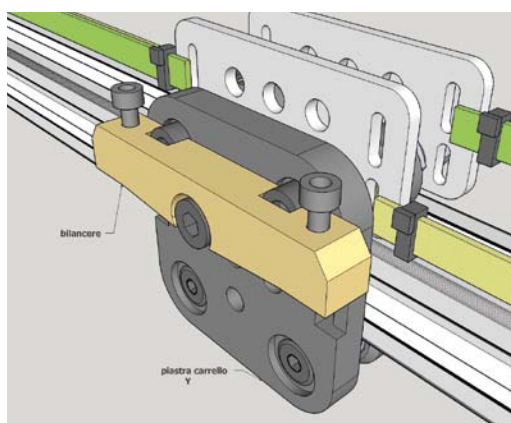


di **DAVIDE SCULLINO**

Stampiamo a due colori in 3D fino a 40x40x40 cm: montiamo il secondo estrusore e configuriamo il client di stampa. Seconda ed ultima puntata.

La nostra stampante 3D in tecnologia FDM 3D4040 sviluppata in collaborazione con la Futura Elettronica ha accresciuto la sua nutrita serie di caratteristiche con una novità: la possibilità di stampare a due filamenti mediante una semplice modifica della testa di stampa con una che aggiunge un secondo estrusore. Tale upgrade ha richiesto la riprogettazione della scheda di controllo, che in vero aveva bisogno di una rivisitazione, essendo nata per la "sorella minore" 3Drag e successivamente adattata alla gestione della 3D4040. Ci siamo lasciati, nel fascicolo scorso, dopo avervi appunto descritto questa nuova scheda controller, sia in teoria che nella pratica (descrizione e programmazione) ed aver introdotto il progetto della stampa a doppio estrusore nel suo insieme. Ricordiamo che per poter essere utilizzata, la scheda control-

Fig. 1
Piastra carrello
della testa di
stampa con
bilancere.



ler richiede che vi sia caricato lo sketch derivato dal firmware Marlin, scaricabile dal nostro sito www.elettronica.in insieme agli altri file del progetto; il caricamento si esegue da Personal Computer nel quale siano installati l'IDE Arduino (almeno in versione 1.6x) e i driver FTDI della scheda, che sono poi quelli del chip U3 (FT232RL) della scheda controller. Se non appare la porta perché non è stato trovato il driver, che andrà quindi installato scaricando il relativo pacchetto dalla pagina web <http://www.ftdichip.com/Drivers/VCP.htm> del computer, quindi avviando l'installazione e seguendo le istruzioni a video (durante l'installazione il cavo USB diretto alla scheda dev'essere disconnesso e andrà collegato quando indicato).

Collegate quindi la scheda al computer mediante un cavo USB A/B e avviate l'IDE Arduino per caricare il firmware che avete scelto; selezionate prima la scheda attraverso il menu Strumenti, scegliendo Arduino Mega 2560 ADK. La programmazione va effettuata senza alimentare la scheda: provvederà il computer attraverso la presa USB, almeno per

quel che riguarda la parte che interessa ai fini del caricamento dello sketch (microcontrollore e componenti essenziali).

Selezionata la scheda cliccate sul menu File, quindi impartite il comando Apri e dalla finestra di dialogo che si propone selezionate il file contenente lo sketch; fatto ciò, provvedete alla verifica e, infine, al caricamento mediante l'apposito pulsante. Avete così archiviato la questione "controller" e potete staccare dal PC e mettere da parte la scheda in attesa di effettuare i cablaggi.

MODIFICHE ALLA STAMPANTE PER APPLICARE IL DOPPIO ESTRUSORE

Passiamo adesso alla realizzazione della stampante sotto l'aspetto della parte meccanica e dei relativi elementi (riscaldatori e termistori) che vanno applicati alla testa di stampa, nonché del motore per il secondo filamento.

Iniziamo a realizzare la nostra modifica partendo dai cablaggi della testa di stampa, ricordando che nella 3D4040 abbiamo previsto, per rendere il cablaggio più ordinato, di collegare i cavi delle teste e quelli delle ventole a una basetta, sulla quale sono presenti delle morsettiere previste allo scopo e sul lato corto della quale abbiamo un connettore flat. La basetta realizza le connessioni tra il connettore e le morsettiere.

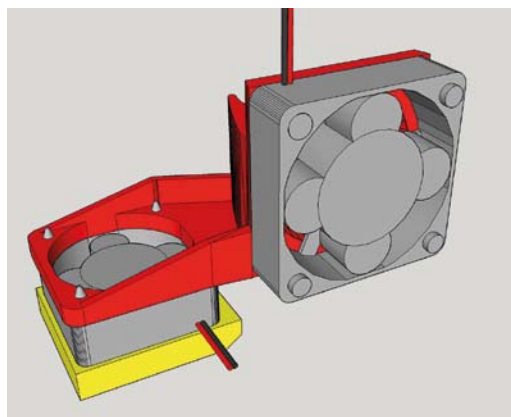
Bisogna quindi fissare tale basetta, tramite l'apposito supporto, al profilo centrale del telaio porta-testa, il quale riporta il cablaggio della testa di stampa, che qui completeremo con le connessioni del secondo estrusore.

Come prima operazione occorre procurarsi il "kit di modifica doppio estrusore" e assemblare l'insieme: prendere quindi il "bilancere" e inserirvi nelle asole rettangolari 2 dadi M3 e di seguito inserire ed avvitare altrettante viti M3x16 TCEI fino a far sporgere dalla parte inferiore del particolare circa 2-3mm di filetto. Fissare quindi il bilancere alla piastra carrello asse Y mediante una vite M5x10 TCEI ottenendo l'insieme mostrato nella **Fig. 1**.

Poi con le viti autofilettanti 2,5x14 TB a croce dovete fissare una ventola 40x40 alla clip ventola DX, in modo che il flusso d'aria sia diretto dalla ventola verso la clip ventola DX. Successivamente andate a fissare alla clip ventola DX anche una ventola 30x30 col relativo beccuccio in plastica tramite 4 viti autofilettanti 2,9x22 TS a croce; stavolta il flusso d'aria deve uscire nella direzione dei fori del beccuccio.

Procedete con il montaggio degli stessi elementi anche sulla clip ventola SX seguendo la procedura

Fig. 2
Assemblaggio
delle ventole.



appena descritta ma orientando i cavetti della ventola 30x30 come visibile nella **Fig. 2**. Assemblare quindi le testine alloggiandole nelle relative sedi del supporto testine dopo averle orientate esattamente come in **Fig. 3**. Applicare poi dalla parte opposta il collare testine ed inserire, nei due fori esterni, 2 viti M5x30 TCEI. Dopo aver controllato attentamente l'orientamento delle testine, serrare poco per volta le due viti quanto basta per bloccare in posizione le testine come si vede in **Fig. 3**.

Applicare la clip ventola DX e quella SX sul corpo alettato delle relative testine allineando il foro di ciascuna ventola con quello presente sul supporto testine. Inserire in ciascun foro una vite autofilettante 2,9x19 TS a croce per fissare le due clip al supporto testine ottenendo l'insieme visibile nella **Fig. 4**.

Fatto questo si va a fissare l'insieme così costituito, ossia il modulo testine, con le apposite viti alla piastra del carrello Y, come proposto nella **Fig. 5**. I quattro cavetti delle testine e quelli delle ventole dovranno essere raggruppati ciascuno sul proprio lato ed uniti mediante fascette in modo che possano raggiungere i morsetti che verranno successivamente fissati alle piastre delle cinghie. Bisogna quindi dotare i due estrusori delle guide per il filamento ed allo scopo occorre inserire nel raccordo della testina di stampa 1 il tubetto in PTFE lungo 1170mm e in quello della testina 2 il tubetto lungo 1 metro. I tubetti devono essere

spinti fino a farli andare in battuta (entreranno nel raccordo per circa 25 mm); l'insieme risulterà come in **Fig. 6**.

I tubetti (bowden) guida-filo andranno poi portati ai relativi stepper-motor di alimentazione, che prenderanno posto ciascuno, alla propria altezza, sul montante del lato su cui è fissato anche uno dei motori di elevazione del carrello porta-testine (**Fig. 7**).

Per vincolare i tubetti di alimentazione occorre far passare il tubetto in PTFE da 1m che fuoriesce dalla testina 2 nell'occhiello superiore di una molla quindi fissare quest'ultima al profilo superiore del telaio allineandola con la cava destra del montante; poi far passare il tubetto in PTFE da 1170mm che fuoriesce dalla testina 1 nell'occhiello superiore della seconda molla che dovrà essere fissata allo stesso profilo del telaio ma, questa volta, allineata con la cava sinistra del montante.

Per completare il gruppo di alimentazione della testa occorre montare i bracci supporto rocchetto ognuno da inserire nell'apposita fessura della relativa staffa, quindi facendola scorrere verso il basso fino a mandarla in battuta. Inserire quindi le viti M4x10 TCEI nei relativi fori avvitando per qualche giro altrettanti dadi a "T" M4 e fissare i supporti rocchetto al telaio nel punto visibile nella **Fig. 8**, ossia il primo ad un'altezza di 110 mm ed il secondo ad un'altezza di 330 mm.

A questo punto si procede con il cablaggio della testina 2: per prima cosa bisogna fissare al carrello

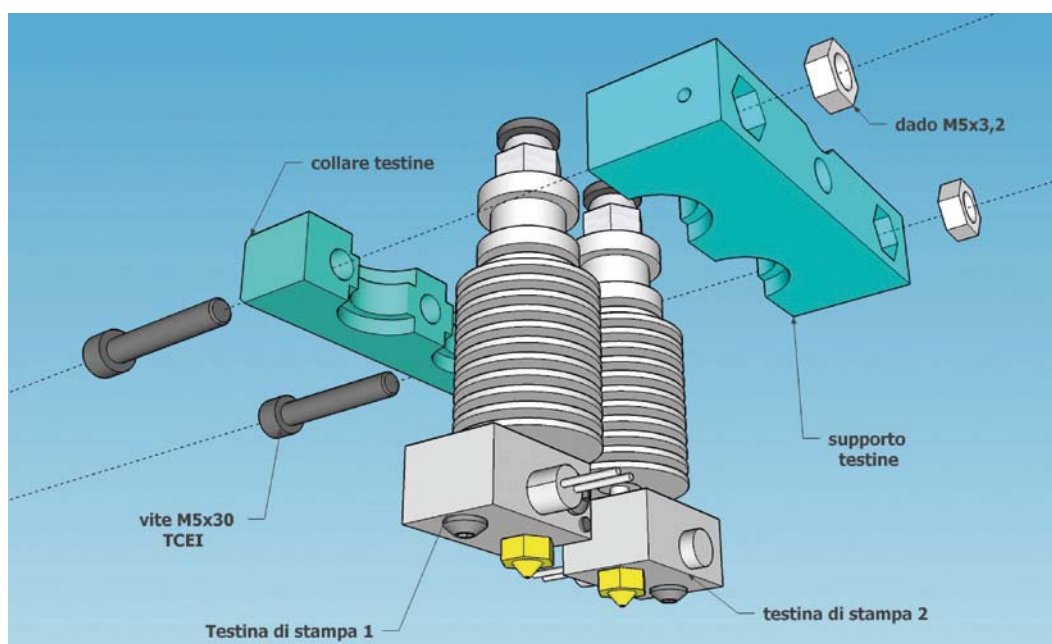
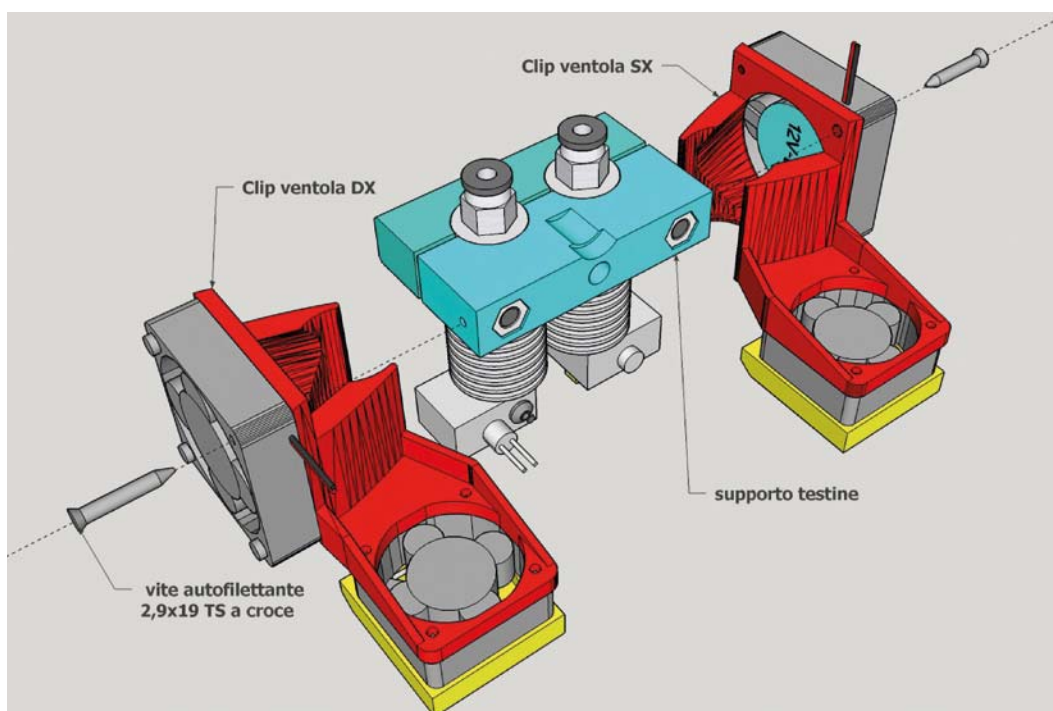


Fig. 3
Esploso
dell'assemblaggio
degli estrusori.

Fig. 4
Assemblaggio
di estrusori
e ventole.



della testina di stampa la scheda morsettiera della testina 2 utilizzando due viti autofilettanti 2,9x13 TS a croce. Facendo riferimento alla **Fig. 9**, collegare alla morsettiera i cavetti provenienti dalla testina 2 (quella di sinistra) rispettando quanto indicato dalla serigrafia riportata sul PCB (guardando l'immagine da sinistra verso destra: terminali + e – ventola raffreddamento oggetto stampato, terminali riscaldatore, terminali NTC e per ultimo terminali + e – ventola raffreddamento testina di stampa).

Con questo è stata cablata la morsettiera; ora bisogna portare il cablaggio alla scheda controller ed allo scopo si deve prendere il flat cable a 14 vie intestato su entrambi i lati con connettore femmina FC14, inserirne nel relativo connettore maschio della scheda della morsettiera un'estremità, quindi cablare l'altra estremità nei relativi connettori maschi sulla scheda controller. Fatto ciò bisogna prendere la seconda scheda morsettiera e posizionarla sopra a quella appena cablata, fissandola al distanziale esagonale mediante una vite M3x8 TCEI completa di rondella dentata M3; si esegue quindi il cablaggio dell'altra testina, ossia la 1, analogamente a quanto fatto già per la 2. Si collega quindi il flat-cable alla scheda morsettiera corrispondente e alla scheda controller. L'insieme del cablaggio della testa di stampa risulterà come in **Fig. 10**.

Il resto del cablaggio della macchina si svolge come già descritto nell'articolo dedicato alla 3D4040 e nelle istruzioni allegate al kit di montaggio.

MESSA A PUNTO DELLA STAMPANTE

Una volta assemblata e cablata la stampante occorre eseguire alcune operazioni preliminari; per prima cosa, se non è stato installato il piatto riscaldato bisogna chiudere il ponticello JPNTC. Poi occorre tarare i driver, operazione da svolgere prima di poter muovere gli assi con il software di gestione; tale operazione va sempre effettuata quando si cambia scheda controller e quindi anche in questo caso, dove è richiesto di sostituire la scheda originale con quella per il doppio estrusore. Dunque, dopo aver alimentato la stampante prendete un multimetro ed impostatelo sulla portata 2V DC fondo scala, posizionate sul morsetto negativo della scheda controller il puntale Nero e quello Rosso sulla piazzola di test del driver del Motore asse Z. Con il cacciavite dovete regolare molto delicatamente il trimmer adiacente alla piazzola di test fino ad ottenere un valore di tensione pari a 0,660 V. Assicuratevi che il valore impostato rimanga stabile anche dopo aver rimosso la punta del cacciavite; diversamente agite nuovamente sul trimmer.

Andate quindi sul driver relativo al motore asse X e impostate col relativo trimmer un valore di tensio-

ne compreso tra 0,415 e 0,420 V Max. Assicuratevi che il valore impostato rimanga stabile anche dopo aver rimosso la punta del cacciavite; diversamente, agite nuovamente sul trimmer per correggerlo.

Ora dovete ripetere la stessa operazione (ed impostare lo stesso valore) per i driver dei motori per l'Asse Y, Estrusore 1 ed Estrusore 2.

CALIBRAZIONE DELLA STAMPANTE

Andiamo adesso a vedere come calibrare la parte elettromeccanica della 3D4040 e in che modo configurare il client di stampa per farlo; nella spiegazione ci riferiremo al client Repetier-Host. Dunque, per prima cosa dovete collegare il cavo USB alla stampante, l'alimentatore alla rete, quindi collegare il cavo USB al PC e controllare che sia stato già installato il driver per il chip FTDI, creando una nuova porta seriale. Se avete già caricato il firmware nella scheda, di sicuro il driver è già presente e funzionante.

Fatto ciò dovete avviare Repetier-Host e cliccare sull'icona "Impostazioni stampante" (in alto a destra) per creare il profilo della 3D4040/DUE. Nel campo Stampante scrivere il nome 3D4040/DUE mentre nella cartella Connessione impostare la porta COM associata alla stampante e un Baud Rate di 250.000. Lasciate invariati gli altri parametri quindi cliccate Applica per salvare il nuovo profilo.

Nella cartella Estrusore (**Fig.11**) andate a impostare 2 estrusori e 2 ventole, 250 per Max. Temperatura Estrusore e 76 per Max. Temperatura piano. Impostare 0.4 per Diametro ugello Estrusore 1 e 2 (la scelta del colore è a propria discrezione) e per ultimo inserire i valori di scostamento X e Y per il secondo estrusore (i valori visibili nell'immagine sono predefiniti e potrebbe essere necessario modificarli dopo aver eseguito il test di allineamento testine). Cliccare su Applica.

Selezionare la cartella Opzioni piano di stampa ed inserire 389.8 come dimensioni X Max e Larghezza area di stampa, 425 per Y Max, 24.14 per Piano da davanti, 400.01 per Profondità area di stampa e 400 per Altezza area di stampa. Premere Applica. Il passo successivo dopo questa procedura è verificare il collegamento fra software e stampante: con stampante accesa e collegata al PC premere il pulsante "Connetti" in alto a sinistra: nella parte bassa dell'interfaccia dovrebbe apparire il nome della stampante connessa e i valori di temperatura

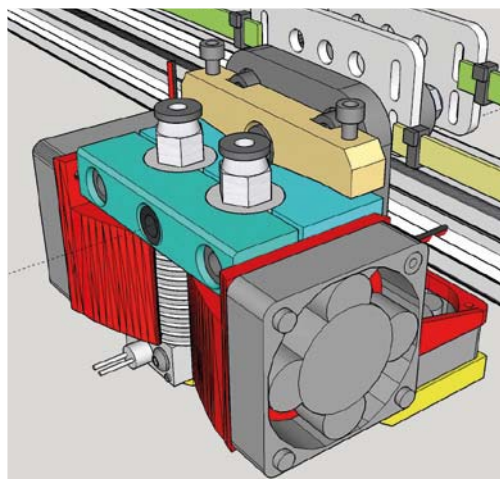


Fig. 5
La doppia
testa di
stampa
assemblata.

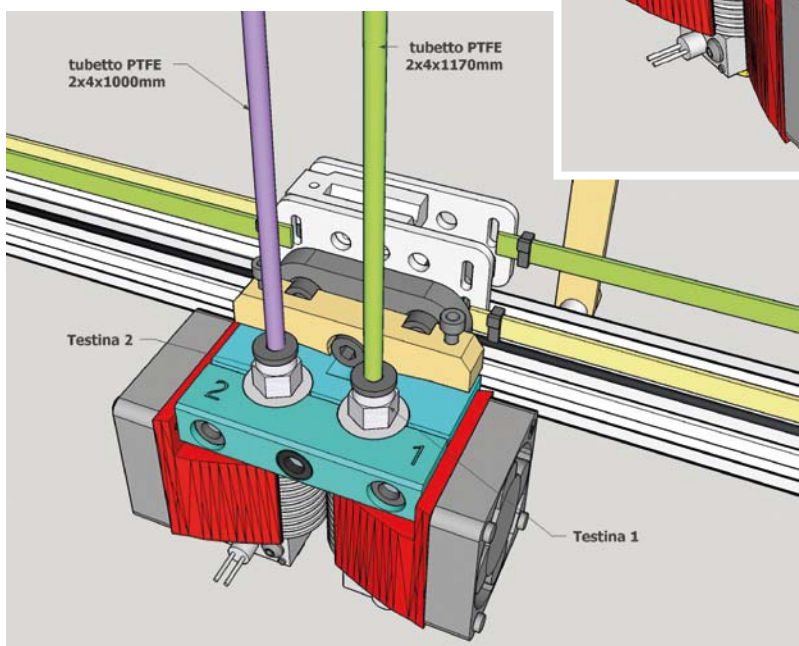


Fig. 6
Testine di
stampa
complete di
guida-filo.

Fig. 7
Complesso di alimentazione
delle due testine di stampa.

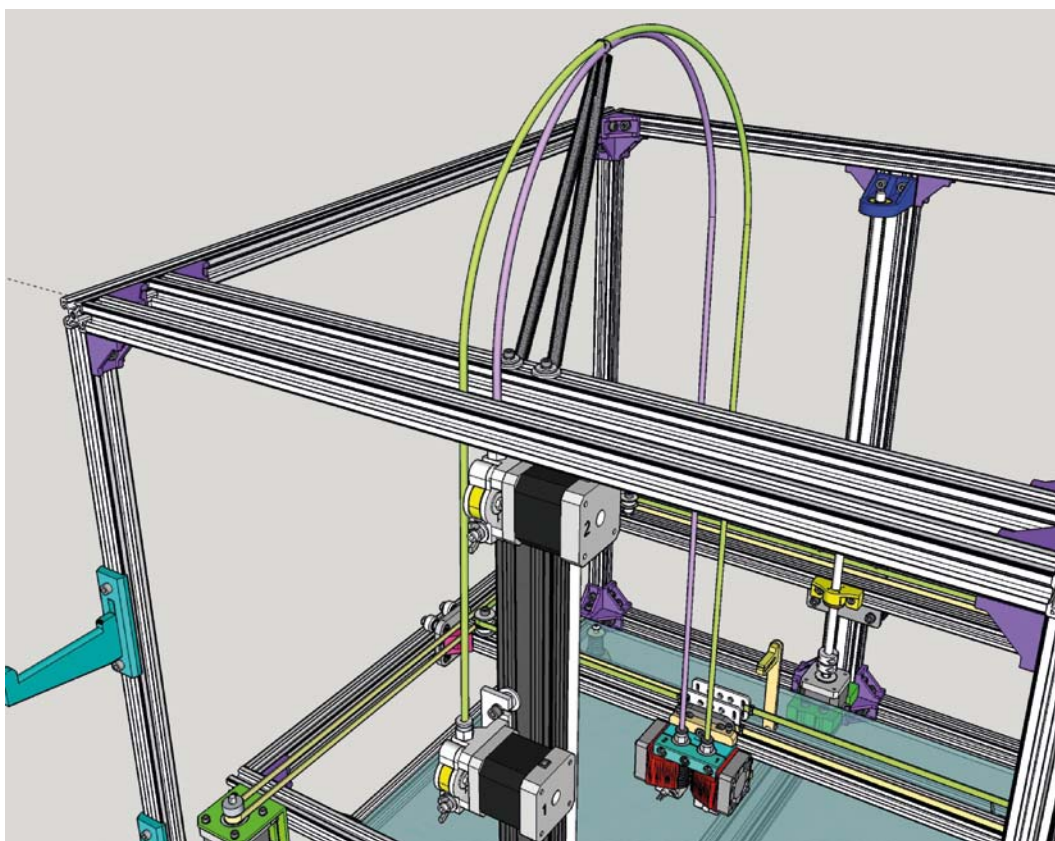
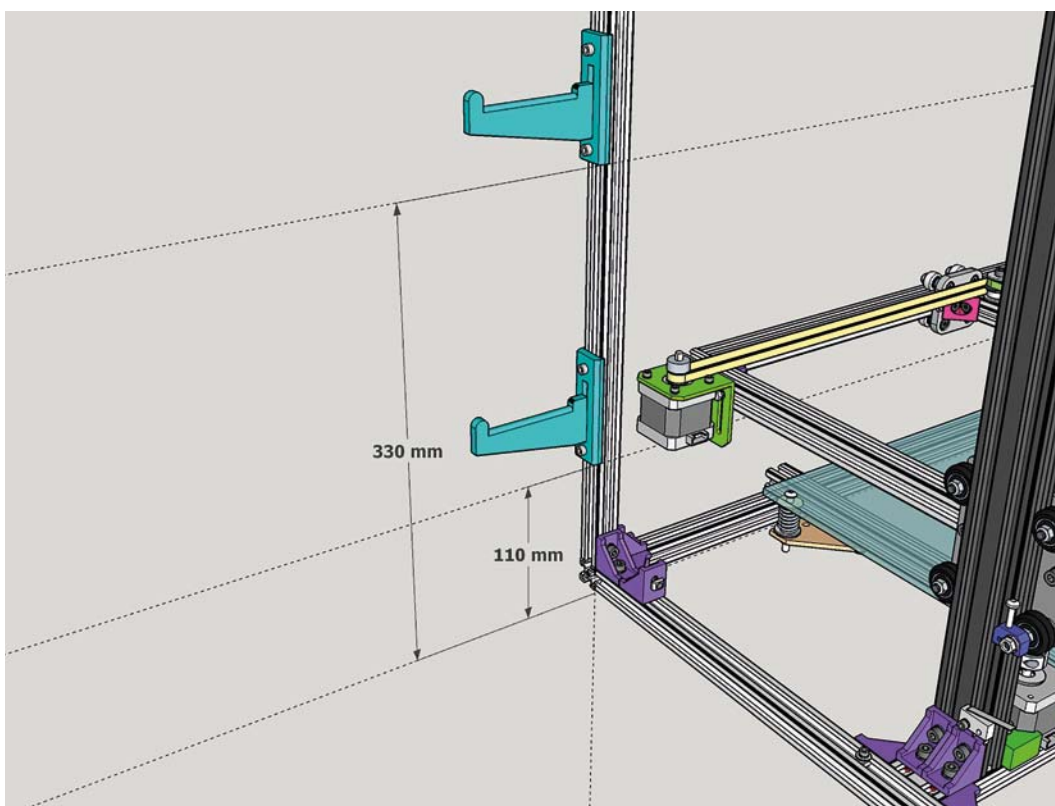


Fig. 8
Posizionamento
dei bracci porta
rocchetto sul telaio.



degli estrusori e del piano di stampa.
Ora si passa all'allineamento delle testine e dei finecorsa: come prima operazione occorre avvitare per una decina di giri la vite del finecorsa asse Z per assicurarsi che il carrello Z non vada a sbattere sul piatto in vetro.

Con una chiave a brugola regolare le 4 viti di fissaggio del piatto in vetro in modo che tutti gli angoli distino dal profilo in alluminio sottostante della stessa misura (circa 12 - 13 mm). Con questa distanza il piatto dovrebbe essere sufficientemente fermo. Allentare leggermente la vite centrale M5x40 TCEI che fissa il modulo testine al carrello

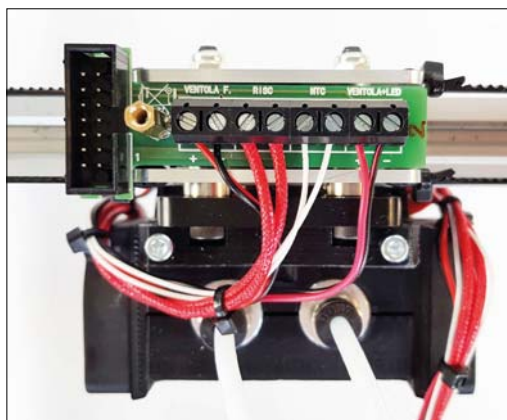


Fig. 9
Cablaggio della singola testa di stampa.

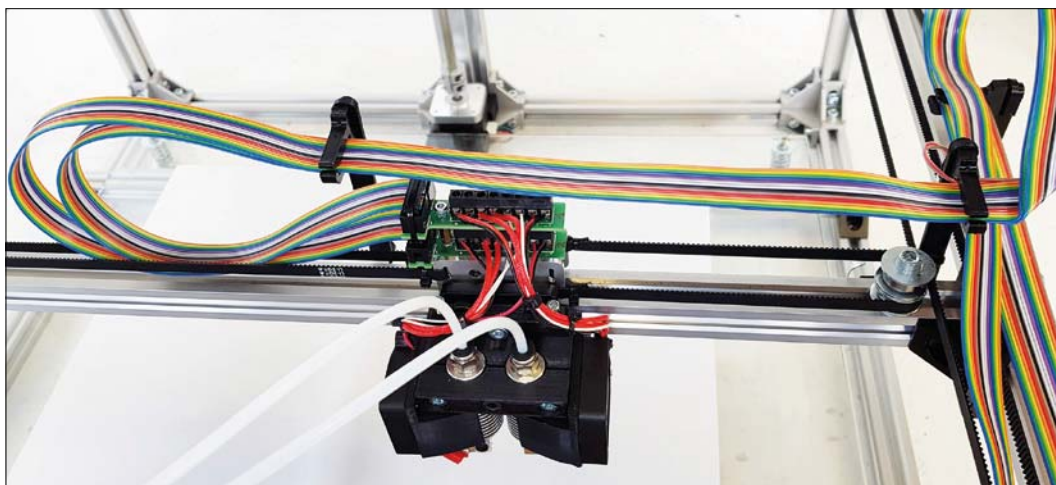


Fig. 10
La testa di stampa cablata.

Y mediante il bilanciere, per lasciarlo inclinare di qualche grado in basso a destra come mostrato nella **Fig. 12**, quindi serrare la vite quanto basta per mantenere le testine bloccate in posizione durante la regolazione del piatto.
Cliccare su Connetti di Repetier-Host e quando il collegamento è stato stabilito selezionare la cartella **Controllo manuale** sul lato destro dell'interfaccia utente.

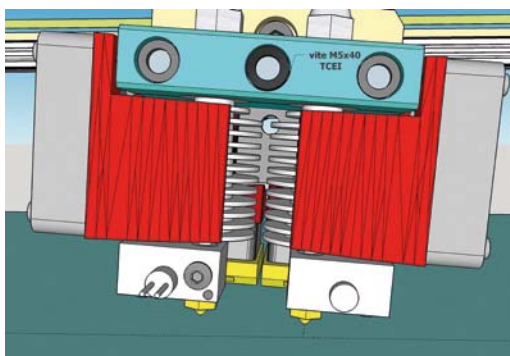
Con i comandi presenti in questa sezione è possibile gestire i movimenti degli assi, l'estrusore, la velocità, la portata nonché il riscaldamento del piatto e della testina di stampa. Premere il pulsante X HOME (il tasto con il simbolo Casetta e lettera X): il carrello X si dovrebbe spostare nella posizione definita HOME ossia verso il profilo su cui sono fissati i motori X e Y (se ciò non avviene verificare che la stampante sia alimentata ed eventualmente i collegamenti interessati). Ora premere il pulsante Y HOME: la testina si dovrebbe spostare verso il motore X. Ora premere i tasti a freccia X/Y per muovere gli assi X e Y al limite della loro corsa per

Impostazioni stampante			
Stampante: 3D4040 DUE			
Connesione Stampante Estrusore Opzioni piano di stampa Scripts Avanzate			
Numero di estrusori:	2		
Numero ventole:	2		
Max. Temperatura Estrusore:	250		
Max. Temperatura piano:	76		
Volume Max per secondo:	12	[mm³/s]	
☐ La stampante ha un estrusore misto (un ugello per tutti i colori)			
Estrusore 1			
Nome:			
Diametro:	0.4	[mm]	Scostamento Temperatura: 0 [°C]
Color:			
Scostamento X:	0		Scostamento Y: 0 [mm]
Estrusore 2			
Nome:			
Diametro:	0.4	[mm]	Scostamento Temperatura: 0 [°C]
Color:			
Scostamento X:	0		Scostamento Y: 24.14 [mm]
OK Applica Annulla			

Fig. 11
Impostazioni estrusore in Repetier-Host.

verificare la copertura della massima area di stampa e che tutto possa muoversi liberamente senza incontrare ostacoli.

Fig. 12
Allentamento della vite del bilancere per consentire alla testa di ondeggiare durante l'allineamento al piatto.



Sempre con i tasti a freccia posizionare le testine di stampa al centro del piatto (circa +200 X, + 200Y). Partendo dal presupposto che il frame del carrello Z sia perfettamente parallelo con la base del telaio, premere il tasto Z Home per far abbassare il carrello Z fino al suo limite inferiore. Se l'ugello più basso non arriva a sfiorare il piatto in vetro, svitare di qualche giro la vite del finecorsa Z quindi cliccare nuovamente Z HOME; se l'ugello arriva a sfiorare il piatto allora significa che la regolazione provvisoria dello zero Z è corretta altrimenti bisogna agire nuovamente sulla vite fino a quando non si raggiunge detta condizione (per comodità è possibile usare un foglio di carta per controllare che l'ugello non sia troppo distante o troppo premuto sul piatto in posizione di HOME: in pratica si deve poter far scorrere un sottile foglio di carta fra piatto ed estrusore senza che resti dello spazio). A questo punto alzare di qualche millimetro le testine quindi posizionarle sopra ad un angolo qualsiasi del piatto. Effettuare l'HOME Z e verificare con il foglio di carta la distanza dell'ugello più basso dal piatto in vetro. Se necessario regolare quanto basta la relativa vite del piatto per ottenere la condizione ottimale. Ripetere la stessa operazione per gli altri angoli. Al termine riposizionare ancora una volta l'ugello su ciascun angolo per assicurarsi che non vi siano state variazioni. Il piatto di stampa risulta così complanare con il modulo testine ma ora occorre regolare l'inclinazione di quest'ultimo per far sì che entrambi gli ugelli distino in egual modo dal piatto di stampa. Per effettuare la regolazione occorre riposizionare il carrello Y al centro del piatto, allentare leggermente la vite centrale che fissa il modulo testine (in modo che quest'ultimo possa oscillare all'occorrenza) quindi cliccare nuovamente Z HOME; sicuramente dopo la discesa del carrello Z i due ugelli non arriveranno a sfiorare contemporaneamente il piatto. Posizionare tra gli ugelli e il piatto di stampa un foglio di carta. Svitare poco per volta la vite del finecorsa Z ed effettuare

nuovamente Z HOME fino a quando entrambi gli ugelli non arrivano a sfiorare contemporaneamente il foglio di carta esercitando la stessa pressione. Raggiunta la condizione, con un cacciavite esagonale dovete regolare poco per volta le due viti M3 poste sul bilancere fino a quando non arrivano in battuta sul supporto delle testine impedendogli di oscillare.

Ora serrate a fondo la vite centrale per bloccare definitivamente il blocco testine.

Controllate che il foglio di carta possa scorrere ancora sotto entrambi gli ugelli allo stesso modo; se necessario ritoccare le due viti M3 per correggere eventuali differenze. Al termine effettuare nuovamente Z HOME e all'occorrenza regolare la vite del finecorsa Z per ottenere la distanza ottimale degli ugelli dal piatto di stampa. Se il piatto dispone del riscaldatore la regolazione definitiva della vite del finecorsa Z deve essere effettuata solamente dopo che il piatto avrà raggiunto la temperatura predefinita, per tenere conto della dilatazione termica. Va inoltre tenuto presente che se sul piatto in vetro viene applicato un nastro per migliorare l'aderenza dell'oggetto stampato è necessario agire sulla vite del finecorsa Z (avvitandola) per compensare lo spessore dell'elemento aggiunto. In posizione di HOME l'ugello deve trovarsi in prossimità della vite di fissaggio del piatto (angolo sinistro frontale) e non sopra di essa; se necessario rivedere il posizionamento dei finecorsa e/o del piatto di stampa.

Bene, fatto tutto questo la vostra 3D4040 è pronta per stampare a due colori: prendete il vostro file .STL, preparatelo, caricatelo nell'engine di stampa e trasformatelo in ... solida realtà. □



Cosa occorre?

Il materiale utilizzato in questo articolo è disponibile presso Futura Elettronica. Il kit di espansione (cod. DUAL3DKIT) per aggiungere un estrusore alla stampante 3D 3D4040 è in vendita al prezzo di Euro 140,00. Il kit comprende un sensore finefilo (cod. YB785175MM, disponibile anche separatamente a Euro 9,00), il motore stepper con spingifilo, la testina riscaldante con sensore di temperatura e tutte le minuterie e accessori per trasformare la 3D4040 (disponibile in kit a Euro 999,00) in una stampante 3D con doppio estrusore. La board di controllo, completa di driver, è disponibile separatamente con il cod. FT1370M a Euro 119,00. 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>**

Taglia e incidi con **Cricut MAKER**



Prezzo IVA inclusa

Macchina da taglio intelligente



€ 439,00

cod. CRICUTMAKER

Dai vita alla fantasia e alla creatività con Cricut Maker l'innovativa macchina da taglio. Grazie a Design Space, il software di progettazione gratuito, potrai realizzare praticamente qualsiasi progetto fai-da-te, dall'arte 3D all'arredamento per la casa, gioielli, progetti di carta e molto altro ancora e inviarti direttamente a Cricut Maker dal tuo tablet, dal PC o dallo smartphone. Cricut Maker taglia in modo rapido e accurato oltre 300 materiali: tessuti, pelle, carta, cartamodelli, cuoio, fogli di balsa e tanto altro. Espandi le potenzialità di Cricut Maker utilizzando utensili dedicati, alcuni inclusi nella confezione, altri acquistabili separatamente.



SFIDA A TRIS IL BRACCIO ROBOTICO!

Un braccio robotico controllato da Arduino UNO e munito di sistema "Pick and Place" gioca a tris muovendo pedine fisiche su un campo da gioco e non perde mai.

di **GABRIELE BIANCHI**

Era il 10 febbraio 1996 quando in un'anima saletta di Philadelphia il russo Garry Kasparov, campione del mondo di scacchi, stava sfidando un avversario molto particolare: Deep Blue. Non stiamo parlando di un essere umano, bensì di un calcolatore realizzato da IBM dall'incredibile potenza computazionale. Alla sua 37ª mossa Kasparov rinunciò e concesse la vittoria al computer che, per la prima volta nella storia, aveva battuto il campione del mondo. In quell'occasione la macchina aveva bisogno di un mediatore, un uomo che doveva fornire all'algoritmo le mosse eseguite da Kasparov, leggere i risultati del calcolo e infine muovere fisicamente il pezzo indicato sulla scacchiera; inoltre, nonostante la prima sconfitta, il russo vinse tre incontri successivi e ne pareggiò due; fu comunque un'impresa straordinaria in un momento storico in cui Google non esisteva e il Wi-Fi non era ancora nato. L'idea così affascinante di quella che il New York Times dell'epoca definì come «un'inattesa vittoria della macchina sull'uomo» è stata lo stimolo che ci ha spinti a creare questo progetto: un braccio robotico RoboArm che sfida l'uomo a Tris. Rispetto all'impresa di Deep Blue il coefficiente di difficoltà è, in questo caso, notevolmente inferiore: il tris, a differenza degli scacchi, ha un'infinità di combinazioni in meno. Abbiamo quindi approfittato di questo vantaggio rendendo il braccio robotico completamente invincibile, il peggior risultato che potrà ottenere è un pareggio e, al contempo, cercherà di vincere in tutti i casi in cui è possibile scegliendo la mossa migliore ad ogni azione. Inoltre, a differenza di Deep Blue, l'effetto "uomo contro macchina" è qui accentuato, infatti è proprio il robot a muoversi autonomamente per spostare le pedine e, al termine dello scontro, se richiesto, rimetterle a posto senza bisogno di mediatori.

COME FUNZIONA

Il campo da gioco, realizzato in plexiglass, è composto dal tipico schema del tris costituito da nove caselle disposte in un quadrato 3x3; ai lati troviamo le pedine (anch'esse in plexiglass) che riproducono "X" e "O" attribuite rispettivamente



Fig. 1
Tastierino
utilizzato dal
giocatore con
lo schema
del tris in
miniatura e i
tre pulsanti di
servizio.

al robot e allo sfidante.

L'input da parte del giocatore avviene tramite un tastierino a matrice 4x3 in cui nove tasti riproducono il campo da gioco del tris in miniatura, a ciascun pulsante corrisponde una casella. Troviamo il pulsante "termina partita", che, se premuto durante il proprio turno, comunica al robot che non si vuole più continuare; "nuova partita", invece, può essere premuto al termine della sfida per iniziare una nuova e infine con "riordina pedine" il robot si muoverà autonomamente recuperando dal campo le pedine utilizzate per rimetterle negli appositi spazi laterali. In **Fig. 1** possiamo vedere un'immagine del tastierino con la copertura in plexiglass incisa a laser. Analizziamo ora una partita tipo.

Il robot sceglie casualmente chi dovrà effettuare la prima mossa; nel caso in cui sia lui il fortunato estratto, si muove autonomamente, prende la prima pedina a croce e la posiziona sul campo in base alla decisione dell'algoritmo. Attende quindi che il giocatore prema un pulsante sul tastierino e, quando questo avviene, recupera la pedina avversaria e la posiziona dove indicato. Nel caso in cui venga estratto il giocatore come primo, il robot attende semplicemente l'input. La partita continua fino alla vittoria del robot o a un pareggio; a quel punto attenderà che lo sfidante prema eventualmente il tasto "riordina pedine" e "nuova partita".

L'HARDWARE

La descrizione completa del braccio robotico e della sua elettronica sono state presentate sui numeri 227 e 228 della rivista: "Roboarm" è realizzato in plexiglass (le parti sono tagliate a laser) e ha

quattro gradi di libertà con la possibilità di montare una pinza (sia in verticale che in orizzontale) per afferrare oggetti. La mobilità è affidata a tre servo RC analogici con ingranaggi in metallo da 13kg*cm e due mini servo da 1,2 kg*cm; i primi permettono il movimento della base, della spalla e del gomito, mentre gli altri controllano la pinza (rotazione e apertura/chiusura).

Al posto della pinza è possibile installare un sistema "Pick and Place", appositamente studiato per "Roboarm", che consente di sollevare degli oggetti grazie alla sua ventosa e alla pompa a vuoto di cui è dotato.

Più nel dettaglio un'elettrovalvola a tre vie permette la commutazione tra pompa e sfiato in modo da poter controllare l'aspirazione e il rilascio tramite Arduino. Sostituendo la pinza con il sistema di aspirazione non utilizzeremo un servomotore poiché non è in questo caso necessario ruotare gli oggetti presi, possiamo evitare di controllare anche il servomotore del polso (che sarà però fisicamente presente in quanto parte integrante del polso stesso). Controlleremo quindi solo i tre servomotori da 13kg*cm.

Abbiamo utilizzato l'apposito shield (presentato nel numero 228) compatibile con Arduino/Fishino UNO che permette di gestire i servo e il sistema di Pick and place fornendo la corrente necessaria. Poiché, come affermato in precedenza, utilizzeremo solo alcuni dei servomotori disponibili, nella **Tabella 1** vediamo l'assegnazione dei componenti e degli output con i relativi pin Arduino (solo quelli usati in questo progetto).

Il dispositivo di input è una tastiera composta da 12 pulsanti collegati a matrice (4 righe x 3 colonne)

Tabella 1

COLLEGAMENTO SULLO SHIELD	PIN ARDUINO	FUNZIONE
S1	D11	Servo rotazione base
S2	D10	Servo flessione spalla
S3	D9	Servo flessione gomito
OUT1	D2	Pompa 5V
OUT2	D4	Valvola a 3 vie

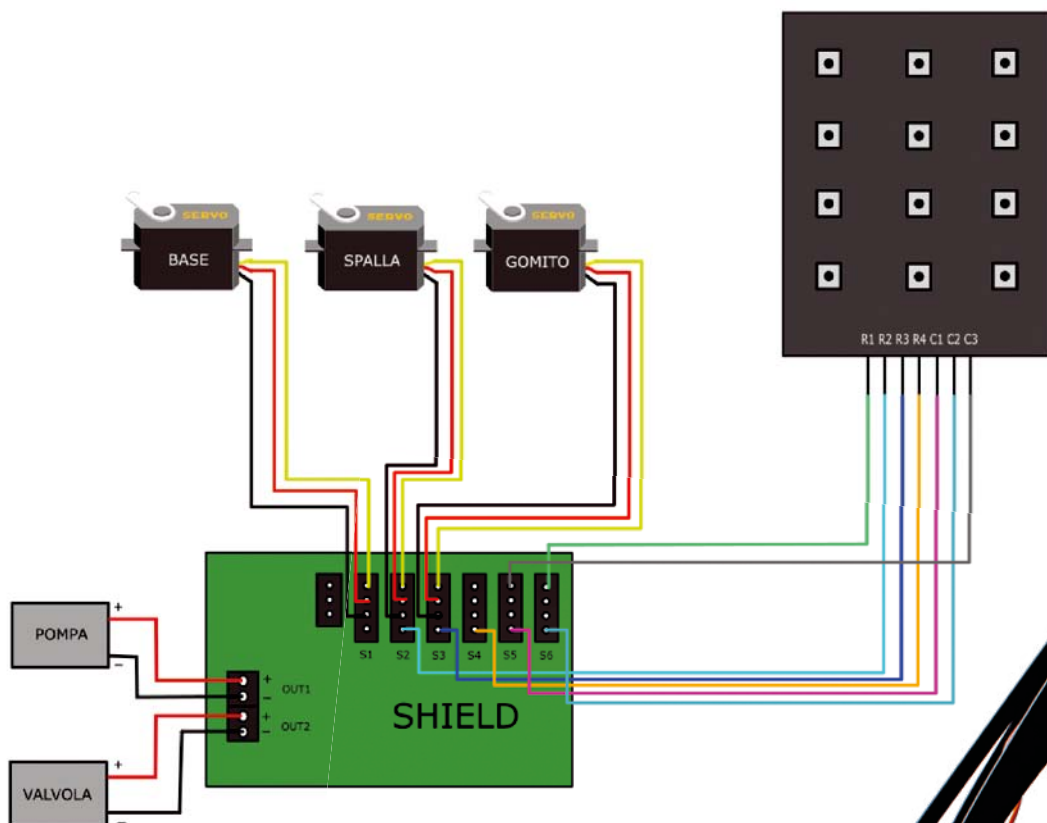


Fig. 2
Schema
generale con
collegamenti
allo shield di
servomotori,
pompa,
valvola a
tre vie e
tastierino.

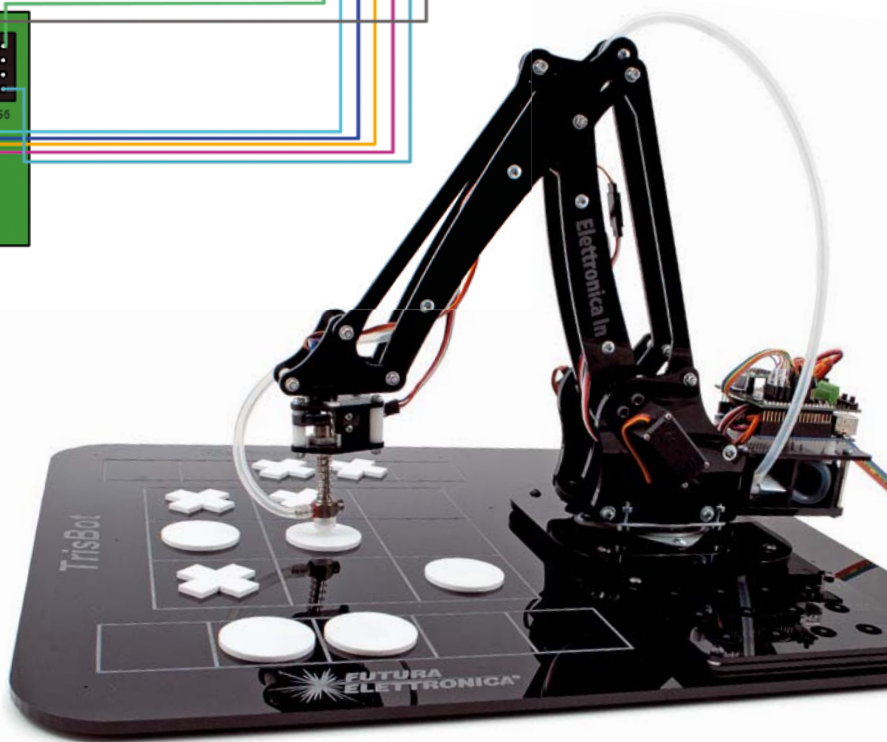
con 7 pin. Nella **Fig. 2** possiamo vedere lo schema generale del progetto.

Per connettere la tastiera ad Arduino tramite lo shield abbiamo sfruttato i pin dei servomotori non utilizzati e i pin di feedback facendo riferimento allo schema elettrico dello shield. Alcuni di questi sono analogici (utilizziamo da A1 ad A5), nel codice li abbiamo però dichiarati come digitali denominandoli rispettivamente da 15 a 19.

PROGRAMMAZIONE

Prima di parlare dell'algoritmo vincente vero e proprio abbiamo la necessità di registrare le posizioni del robot nel campo da gioco in funzione degli angoli dei tre servomotori.

Nel complesso le posizioni saranno: le nove del tris, cinque laterali per le pedine "X" e cinque laterali per le pedine "O". Ci sono vari modi per ottenere i tre angoli (le posizioni dei tre motori), in questo caso abbiamo pensato di sfruttare il tastierino già a disposizione e scrivere un codice che permette di controllare il robot tramite i pulsanti. In particolare le prime tre righe della tastiera (vista in verticale con il connettore verso il basso) controllano i tre servo; per ciascuna riga il primo pulsante aumenta



l'angolo e l'ultimo lo diminuisce. Quando premiamo un pulsante il servo corrispondente si muove, tenendolo premuto continuerà il movimento. Una volta portato il braccio nella posizione desiderata utilizziamo la quarta riga come debug: premendo ciascuno dei tre pulsanti mostreremo a monitor seriale l'ultima posizione data in input ai rispettivi tre servomotori. Utilizziamo questo procedimento per trovare tutte le diciannove posizioni che ci serviranno nel codice del tris vero e proprio. Nel **Listato 1** troviamo la funzione "tastierino" che, in base al pulsante premuto, stampa i valori degli

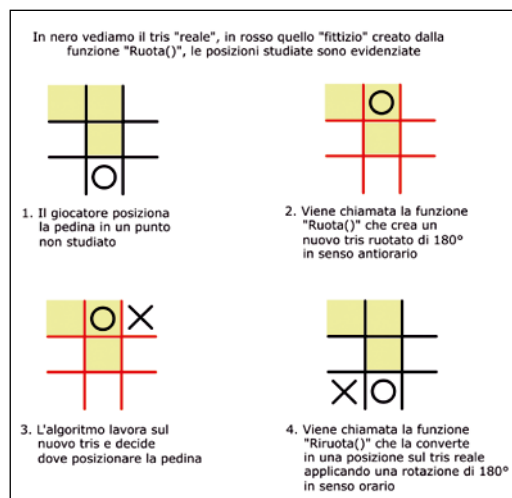
↓ Listato 1

```
void tastierino()
{
    int premuto = 0;
    for(int column = 0; column < numCols && premuto == 0; column++){
        digitalWrite(colPins[column],LOW);          // Attiva la colonna corrente per la lettura.
        for(int row = 0; row < numRows && premuto == 0; row++){      // Scorro tutte le righe cercando un tasto premuto
            if(digitalRead(rowPins[row]) == LOW){      // E' premuto il tasto
                premuto = 1;
                delay(debounceTime);                  // attendo che il segnale si stabilizzi
                if(row == 3){ //Devo stampare i valori
                    while(digitalRead(rowPins[row]) == LOW); //Aspetto che il tasto sia rilasciato
                    if(column == 0)
                        Serial.print("Base: ");
                    else if(column == 1)
                        Serial.print("Spalla: ");
                    else if(column == 2)
                        Serial.print("Gomito: ");
                    Serial.println(String(angolo[column])); //Stampo l'angolo relativo al servomotore che ho scelto in base alla colonna
                }else
                    movimento(angolo, row, column);
            }
        }
        digitalWrite(colPins[column],HIGH);          // Rendo inattiva la colonna.
    }
}
```

angoli oppure chiama la funzione "movimento". I valori trovati saranno organizzati nel nuovo codice in array bidimensionali in cui le righe rappresentano le diverse posizioni e le colonne i valori degli angoli dei tre servomotori. Ora dobbiamo scrivere il codice vero e proprio. Per fare in modo che il robot vinca sempre abbiamo due possibilità: o utilizzare l'intelligenza artificiale e quindi "insegnargli" a giocare a tris, oppure scrivere tutte le possibili combinazioni di mosse manualmente. Noi abbiamo optato per la seconda.

Ad una prima occhiata sembra una pazzia, in effetti facendo qualche calcolo otteniamo che le combinazioni possibili in una partita sono ben 255.168! E quindi come abbiamo fatto a scriverle tutte? In realtà quel numero enorme non tiene conto delle simmetrie nello schema del tris, inoltre, ci sono delle situazioni in cui la vittoria non è possibile per nessuna delle due parti e non è quindi necessario considerare l'evoluzione della partita. Con queste ipotesi le combinazioni si riducono notevolmente e possono essere scritte in un codice, ma come approfittiamo delle simmetrie? Tutte le mosse saranno divise in due macro gruppi: quando il robot parte per primo (e quindi "attacca") e quando invece "difende", ovvero è il giocatore a partire. Il campo da gioco del tris è rappresentato nel codice da un array bidimensionale 3x3, in cui quando non ci sono pedine ciascuna casella vale 1, quando viene posizionata una pedina "X" la casella vale 3 e per la pedina "O" del giocatore il valore è 5. In questo modo possiamo tenere sotto controllo tutta la partita ed effettuare rapidi calcoli numerici per controllare, per esempio, se in una riga, colonna o diagonale qualcuno ha fatto tris. In aggiunta all'array tabella della partita "reale" creiamo un altro array bidimensionale 3x3, ma che rappresenta un tris "fittizio" in cui tutte le situazioni sono riportate a uno dei casi base che abbiamo studiato. La prima mossa è decisiva (sia nel caso

➡ **Fig. 3**
Procedimento
algoritmico
di rotazione
per gestire le
simmetrie.



▼ Listato 2

```
void ruota90(int pos[][COLS], int pos1[][COLS])
{
    for(int i=0; i < 3; i++)
        for(int j = 0; j < 3; j++)
            pos1[i][j] = pos[j][abs(i-2)];
}

void ruota180(int pos[][COLS], int pos1[][COLS])
{
    for(int i = 0; i < 3; i++)
        for(int j = 0; j < 3; j++)
            pos1[i][j] = pos[abs(i-2)][abs(j-2)];
}

void ruota270(int pos[][COLS], int pos1[][COLS])
{
    for(int i = 0; i < 3; i++)
        for(j = 0; j < 3; j++)
            pos1[i][j] = pos[abs(j-2)][i];
}
```

in cui inizi il robot, sia viceversa), è la prima pedina infatti a decretare il tipo di simmetria.

Facendo riferimento alla **Fig. 3** (in cui possiamo vedere il procedimento algoritmico), abbiamo scritto i casi possibili solo quando la prima pedina viene messa in una delle tre posizioni evidenziate.

Quando questo non accade viene chiamata la funzione "Ruota()" il cui scopo è prendere il tris "reale" e ruotarlo di 0°, 90°, 180° o 270° in modo da portare la pedina in una delle tre posizioni base; nel **Listato 2** vediamo come esempio le funzioni "Ruota90()", "Ruota180()" e "Ruota270()" che vengono chiamate da "Ruota()" in base alla situazione e che creano effettivamente il tris fittizio (dichiarato come pos1[3][3]).

Su questo nuovo array bidimensionale lavorerà l'algoritmo che, dopo aver capito la mossa migliore da fare, tramite la funzione "Riruota()", convertirà

▼ Listato 3

```
void chiudiMioTris(int pos1[][COLS], int * n, bool * verificaTris, bool * bloccaterzamossa, bool * bloccaMossaRandom, bool * vittoria)
{
    int i, j, stop;
    stop = 0;
    for(i = 0; i < 3 && stop == 0; i++){
        /*Ho una riga da chiudere*/
        if(pos1[i][0] * pos1[i][1] * pos1[i][2] == 9){
            for(j = 0; j < 3 && stop == 0; j++)
                if(pos1[i][j] == 1){
                    * n = 3 * i + j;
                    pos1[i][j] = 3;
                    stop = 1;
                }
        }
        /*Ho una colonna da chiudere*/
        else if (pos1[0][i] * pos1[1][i] * pos1[2][i] == 9){
            for(j = 0; j < 3 && stop == 0; j++)
                if(pos1[j][i] == 1){
                    * n = 3 * j + i;
                    pos1[j][i] = 3;
                    stop = 1;
                }
        }
        /*Ho una diagonale da chiudere*/
        else if (pos1[0][0] * pos1[1][1] * pos1[2][2] == 9 && pos1[i][i] == 1){
            * n = 3 * i + i;
            pos1[i][i] = 3;
            stop = 1;
        }
        else if (pos1[0][2] * pos1[1][1] * pos1[2][0] == 9 && pos1[i][2-i] == 1){
            * n = 2 * i + 2;
            pos1[i][2-i] = 3;
            stop = 1;
        }
    }
    if(stop == 1){
        * verificaTris = true;
        * bloccaterzamossa = true;
        * bloccaMossaRandom = true;
        * vittoria = true;
    }
}
```


↓ Listato 4

```
void sistemaTutto()
{
    int i, j;
    numXprese = 0;
    numOprese = 0;
    for(i = 0; i < 3; i++){
        for(j = 0; j < 3; j++){
            if(pos[i][j] == 3){ //ho trovato X
                n = (3 * i) + j;
                //prendo la pedina
                servoMovementup(20, PosTris[n][0], PosTris[n][1], PosTris[n][2], 100, 90, 90);
                delay(1000);
                digitalWrite(valve, HIGH);
                digitalWrite(pump, LOW);
                delay(1000);
                servoMovementdown(20, PosBase[0], PosBase[1], PosBase[2], 100, 90, 90);

                //metto la pedina a lato del campo
                servoMovementup(20, RecuperoX[numXprese][0], RecuperoX[numXprese][1], RecuperoX[numXprese][2], 100, 90, 90);
                delay(1000);
                digitalWrite(valve, LOW);
                digitalWrite(pump, HIGH);
                delay(500);
                servoMovementdown(20, PosBase[0], PosBase[1], PosBase[2], 100, 90, 90);

                numXprese++;
            }else if(pos[i][j] == 5){ //ho trovato O
                n = (3 * i) + j;
                //prendo la pedina
                servoMovementup(20, PosTris[n][0], PosTris[n][1], PosTris[n][2], 100, 90, 90);
                delay(1000);
                digitalWrite(valve, HIGH);
                digitalWrite(pump, LOW);
                delay(1000);
                servoMovementdown(20, PosBase[0], PosBase[1], PosBase[2], 100, 90, 90);

                //metto la pedina a lato del campo
                servoMovementup(20, RecuperoO[numOprese][0], RecuperoO[numOprese][1], RecuperoO[numOprese][2], 100, 90, 90);
                delay(1000);
                digitalWrite(valve, LOW);
                digitalWrite(pump, HIGH);
                delay(500);
                servoMovementdown(20, PosBase[0], PosBase[1], PosBase[2], 100, 90, 90);

                numOprese++;
            }
        }
    }
}
```

la posizione fittizia in una posizione sul tris reale in base alla rotazione di partenza.

Ogni volta che verrà posizionata una nuova pedina, sarà applicata la stessa rotazione, verrà quindi calcolato l'output e infine riconvertito in una posizione reale. Oltre alle mosse valutate, come abbiamo già detto, ci sono dei casi in cui la partita non può essere vinta da nessuno dei due contendenti, in questa situazione la funzione "Random()" semplicemente posiziona la pedina in una casella libera. Abbiamo notato che basta scrivere le possibilità per le prime tre mosse sia in attacco che in difesa,

oltre la terza o il gioco termina perché qualcuno può chiudere un tris, oppure finisce in pareggio. Ad ogni ciclo vengono chiamate le funzioni "chiudiMioTris()", "chiudiSuoTris()", "controllavittoria()" e "controllapareggio()".

La funzione "chiudiMioTris()" è visibile nel **Listato 3**, in particolare possiamo vedere il controllo di righe, colonne e diagonali tramite cicli for in cui la condizione è numerica (poiché le pedine vengono considerate come numeri).

Potrebbe essere interessante spiegare la funzione "primaMossa()" che viene chiamata quando è il

robot a dover scegliere dove posizionare la prima pedina. Abbiamo analizzato i vari casi e abbiamo concluso che la probabilità di vittoria è maggiore con la prima mossa al centro; mettendo invece la pedina negli angoli è leggermente inferiore; le rimanenti posizioni hanno una probabilità molto bassa di vittoria.

Abbiamo quindi strutturato la funzione in modo da estrarre dove posizionare la pedina tenendo conto delle varie probabilità sopra citate.

Per quanto riguarda la parte di codice che controlla il movimento del robot, abbiamo inizialmente utilizzato la libreria "Braccio.h" a disposizione, tuttavia questa creava dei problemi nel prendere e rilasciare la pedina. Andando ad analizzare il funzionamento della libreria troviamo un ciclo while in cui, ad ogni iterazione, per ciascun servomotore, viene incrementato/diminuito l'angolo, il ciclo termina quando tutti i servomotori sono arrivati in posizione.

Poiché nel nostro caso la ventosa deve arrivare a toccare il terreno, se usiamo questo codice c'è la possibilità che quando la punta sta toccando terra, la base deve terminare il movimento e quindi trascina per il campo la ventosa.

Per ovviare a questo problema abbiamo scritto una funzione interamente dedicata prendendo spunto dalla libreria stessa, ma che esegue i tre movimenti separatamente; nel caso in cui il robot stia prendendo la pedina, l'ultimo movimento che farà sarà la discesa della punta (quindi il controllo del servo gomito) mentre in caso di rilascio della pedina il primo movimento sarà alzare la punta. La funzione "sistemaTutto()" permette di rimettere tutto a posto a lato del campo, nel **Listato 4** possiamo vederne il contenuto.

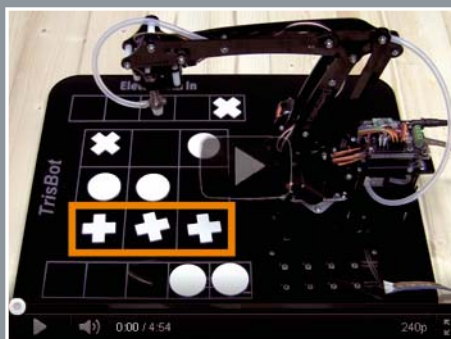
Viene letto l'array bidimensionale che corrisponde al tris "reale" alla ricerca di pedine, quando ne viene trovata una inizia la sequenza di movimento, le variabili "numXprese" e "numOprese" tengono il conto di quante pedine ho già posizionato in modo da metterle in fila ai lati del campo.

CONCLUSIONI

Con questo progetto abbiamo quindi convertito una sfida di programmazione (rendere un macchinina invincibile a tris) in un qualcosa di pratico, grazie al braccio robotico e al sistema di pick and place. Possiamo effettivamente vedere delle azioni meccaniche e non solamente la restituzione di un risultato a schermo.

Un possibile miglioramento futuro potrebbe essere l'eliminazione del tastierino e l'implementazione

Guarda TrisBot in azione su YouTube!



Vuoi vedere in azione il nostro braccio robotico che gioca a tris? Nulla di più semplice: collegati alla pagina <https://youtu.be/P43znyVhyHM>. Potrai anche vedere i video di moltissimi altri nostri progetti proposti in passato.

di un sistema di rilevazione dell'input ancora più automatico magari con una telecamera; oppure semplicemente cambiare gioco e chissà se un giorno anche il nostro robot potrà competere con un campione del mondo di scacchi! □



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. Il braccio robotico (cod. ROBOARM) è in vendita a Euro 89,00, il sistema Pick and Place (cod. PEPARM) è disponibile a Euro 45,00, la board di controllo Arduino UNO (cod. ARDUINOUNOREV3) costa Euro 24,50, la shield per il controllo del braccio robotico (cod. FT1397M) è in vendita a Euro 24,50, la tastiera con 12 pulsanti (cod. FT1362M) è disponibile a Euro 6,00. La base di gioco è stata realizzata tramite il servizio di taglio laser, disponibile all'indirizzo bit.ly/LaserServiceFE. 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

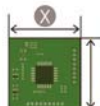
1 Layer

2 Layers

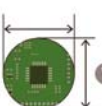
4 Layers

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®**

IL MIO FREQUENZIMETRO? IN KIT

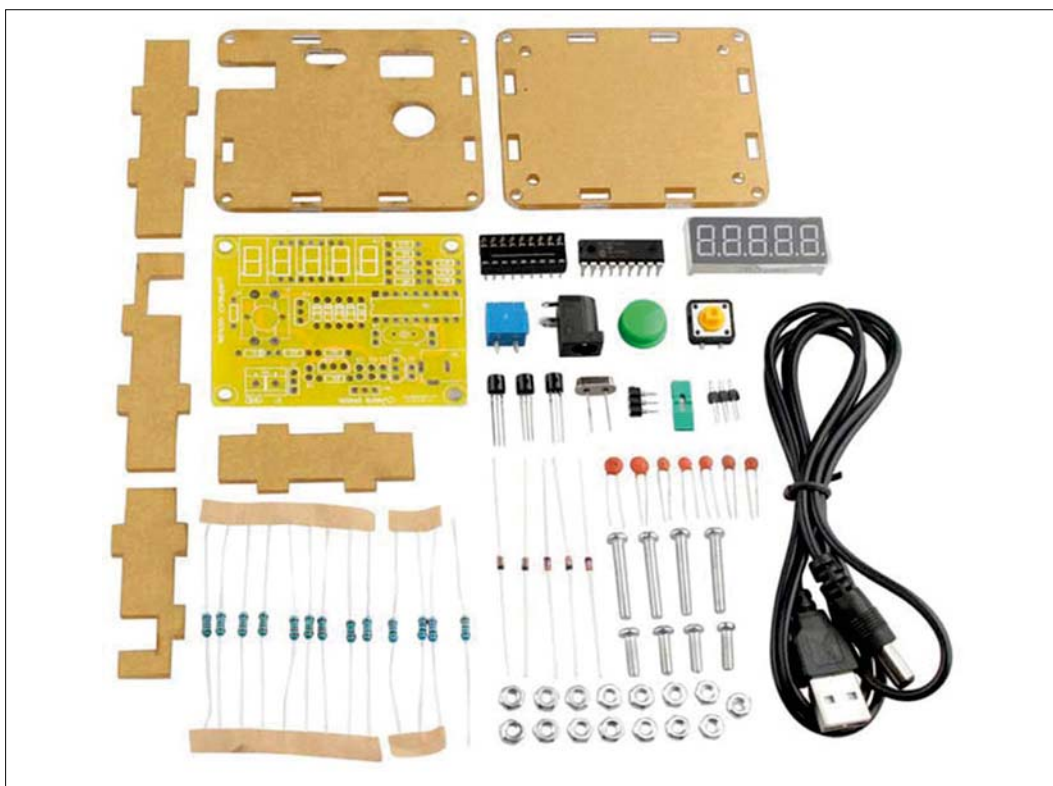
del Prof. MICHELE MENNITI



Montiamo e testiamo un compatto frequenzimetro utilizzabile sia nelle misure in campo che in laboratorio.

Il frequenzimetro è uno strumento che non può mancare in un laboratorio degno di tale nome. Molti pensano che disporre di un oscilloscopio digitale significhi poter fare a meno di questo strumento, visto che nelle varie misure è prevista anche la lettura della frequenza dei segnali ma, a meno che non si parli di un DSO professionale del costo di svariate migliaia di euro, non riusciremo mai ad avere la stessa precisione e risoluzione di un frequenzimetro. Lo strumento che vi presentiamo in questo articolo viene venduto in kit dalla Futura Elettronica (con il codice prodotto 1606-FRE50MHZ) comprendente il circuito stampato, il contenitore e tutta la necessaria componentistica (**Fig. 1**); il montaggio è molto semplice, alla portata di qualunque hobbista, ed il costo è veramente irrisorio, in quanto i due componenti fondamentali sono un PIC16F628A, opportunamente programmato, ed un display 5 cifre a sette segmenti, con punto decimale su ogni singola cifra. Vi diciamo subito che questo kit si adatta alla maggior parte degli utilizzi ed è molto utile quando volete misurare frequenze di cui non conoscete l'entità oppure per testare dei

Fig. 1
Il contenuto della
confezione del kit
1606-FRE50MHZ.



quarzi. In **Fig. 2** vediamo il kit montato e pronto all'utilizzo. Nella confezione è disponibile anche un pratico contenitore realizzato in Plexyglass che lo protegge da eventuali urti (**Fig. 3**). Ma iniziamo subito analizzando le caratteristiche di questo strumento.

CARATTERISTICHE DELLO STRUMENTO

Il nostro Frequenzimetro è in grado di leggere segnali digitali da 1 Hz a 50 MHz ed il valore della frequenza fondamentale dei quarzi da 4 MHz a 40 MHz; ciò significa che se misuriamo un quarzo con frequenza di oscillazione data da una seconda



Fig. 2
Configurazione minimal del Frequenzimetro.



Fig. 3
Il Frequenzimetro completo di contenitore.

o terza armonica, cioè una frequenza multipla di quella fondamentale, il circuito mostrerà sul display quest'ultima e non il valore nominale stampato sul corpo del quarzo.

Le caratteristiche tecniche complete del frequenzimetro sono le seguenti:

- microcontrollore PIC 16F628A con quarzo da 20 MHz;
- display 5 cifre a sette segmenti, con punto decimale su ogni singola cifra;
- risoluzione a cinque cifre con punto decimale mobile;
- misura di quarzi e segnali logici standard;
- circuito opzionale per segnali sinusoidali a bassa ampiezza;
- gamma di frequenza da 1 Hz a 50 MHz per segnali logici, da 4 MHz a 40 MHz per la misura della frequenza armonica fondamentale dei quarzi;
- commutazione automatica della gamma di lettura con diversi tempi di gate;
- aggiunta o sottrazione opzionale di un offset di frequenza (programmabile);
- modalità di risparmio energetico configurabile, con spegnimento automatico del display in assenza di variazioni di lettura oltre 15 secondi;
- alimentazione 5÷2V, assorbimento 25 mA max.

SCHEMA ELETTRICO

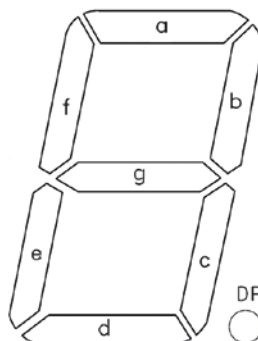
Il cuore del circuito è l'eccellente microcontrollore Microchip PIC16F628A, che lavora a 20 MHz, in virtù di un quarzo applicato ai pin dell'oscillatore interno; il micro, oltre a leggere il segnale, si occupa di pilotare i 5 digit del display a LED a catodo comune, con la tecnica del multiplexing. Poiché il numero di pin disponibili non permette di pilotare le cinque cifre, il progettista ha fatto ricorso ad un abile stratagemma tecnico. In pratica, mentre i primi 4 digit sono pilotati rispettivamente dai pin 2, 1, 17 e 18, l'accensione del quinto digit (l'ultimo a destra, cifra meno significativa) dipende, solo per frequenze uguali o maggiori di 10 kHz (condizione dettata dal firmware), dalla combinazione degli altri quattro. I diodi D2, D3 e D4, ed il transistor NPN U3 costituiscono di fatto una porta NAND a 4 ingressi, per cui se anche uno solo dei 4 pin indicati in precedenza si trova a livello LOW (digit acceso), tale livello si troverà sulla base del transistor, mantenendolo in stato di interdizione. Solo nel momento in cui tutti i digit saranno spenti (pin a livello HIGH), il resistore R5, collegato al pin 2, piloterà la base di U3 mandandolo in saturazione e collegando quindi il pin 5 del display a massa, pro-



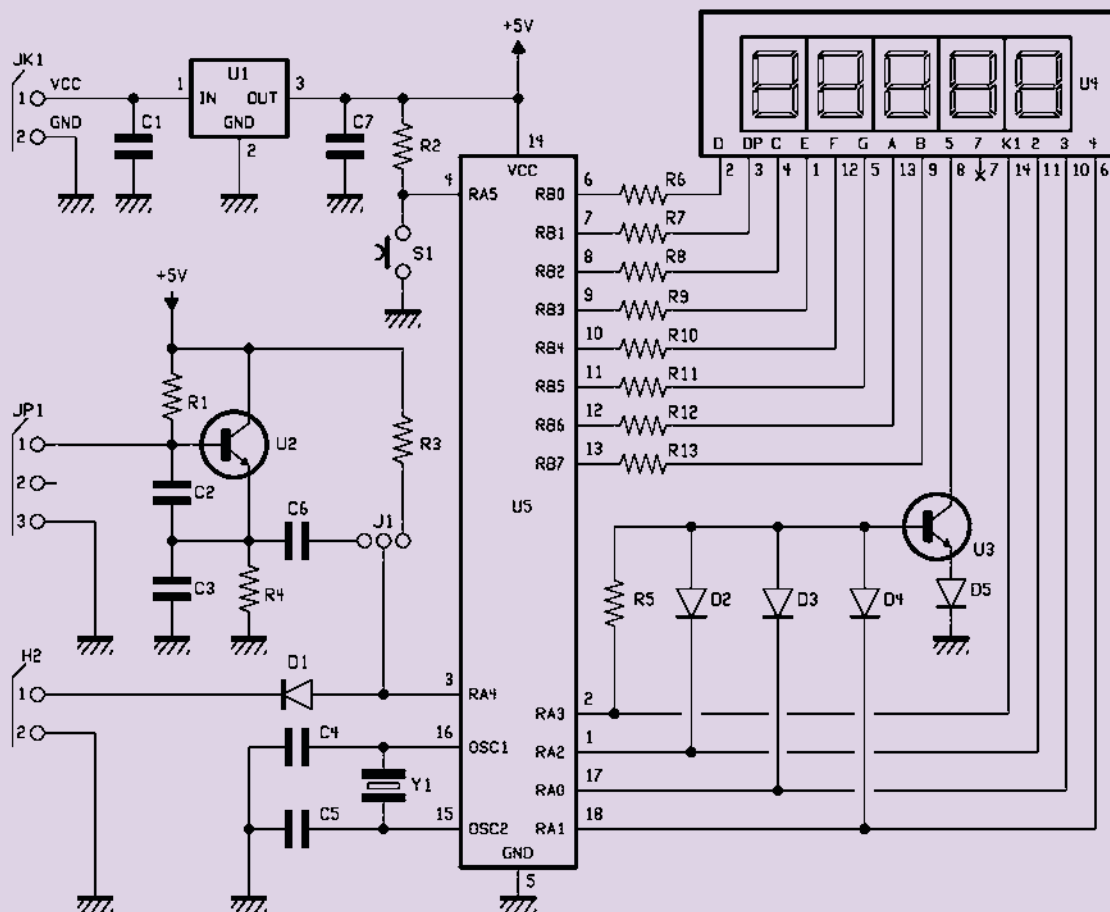
◀ **Fig. 4**
Visualizzazione di 5 digit in multiplexing.

vocandone l'accensione. Per tale ragione all'accensione del frequenzimetro vedrete che lo zero, o la cifra meno significativa, per frequenze fino a 9.999 Hz, saranno visualizzati sul quarto digit. Chi non ha chiaro il funzionamento di un sistema multiplexer potrebbe avere difficoltà a comprendere questo meccanismo, quindi facciamo un piccolo approfondimento. La tecnica del multiplexing, nel caso dei display, consiste nel pilotaggio in contemporanea di tutti i 7 segmenti dei vari digit, e nel pilotaggio sequenziale dei catodi (o anodi, secondo i casi) degli stessi digit. Vediamo un esempio: immaginiamo di dover visualizzare la lettura "67.592" (**Fig. 4**). Vengono eseguite 5 operazioni in sequenza (fare riferimento alla **Fig. 5**):

1. il microcontrollore mette in stato HIGH (livello logico alto) i pin necessari per formare il numero 6 (tutti ad eccezione di "b" e DP), poi mette in LOW il solo pin 2 (gli altri tutti in HIGH), in tal modo il numero 6 viene mostrato solo sul primo digit (partendo da sinistra, cifra più significativa);
2. il microcontrollore pone a livello alto i pin necessari per formare il numero 7 con il punto decimale (segmenti "a", "b", "c" e DP), poi mette in LOW il solo pin 1, in tal modo il numero 7 viene mostrato solo sul secondo digit;
3. il microcontrollore mette in HIGH i pin necessari per formare il numero 5 ("a", "f", "g", "c", "d"), poi mette in LOW il solo pin 17, in tal modo il numero 5 viene mostrato solo sul terzo digit;
4. il microcontrollore mette in HIGH i pin necessari per formare il numero 9 (tutti ad eccezione di "e" e DP), poi mette in LOW il solo pin 18, in



◀ **Fig. 5**
Display a 7 segmenti con punto decimale.



Elenco Componenti:

R1: 100 kohm 1/4w
 R2, R3, R5: 10 kohm 1/4w
 R4, R6-R13: 1 kohm 1/4w
 Y1: Quarzo 20 MHz
 S1: Pulsante da c.s.
 C1, C7: Condensatore Ceramico 100 nF (104)
 C6: Condensatore Ceramico 1 nF (102)
 C2, C3, C4, C5: Condensatore Ceramico 22 pF

D1-D5: Diodo 1N4148
 U1: Regolatore di tensione 7550-1
 U2, U3: Transistor S8550
 U4: Display 4 digit
 U5: Microcontrollore programmato PIC16F628
 JK1: Plug di alimentazione DC
 JP1: Connettore da c.s. 3 pin
 H2: Morsetto 2 poli
 J1: Connettore da c.s. 3 pin

- tal modo il numero 9 viene mostrato solo sul quarto digit;
- il microcontrollore pone in HIGH i pin necessari per formare il numero 2 ("a", "b", "g", "e", "d"), ed anche i quattro pin dei catodi (2, 1, 17, 18); questa condizione, come abbiamo visto, manda in conduzione U3 e il numero 2 viene mostrato sul quinto digit.
 - il microcontrollore ricomincia la sequenza dal

punto 1, e così finché non cambierà il valore misurato.

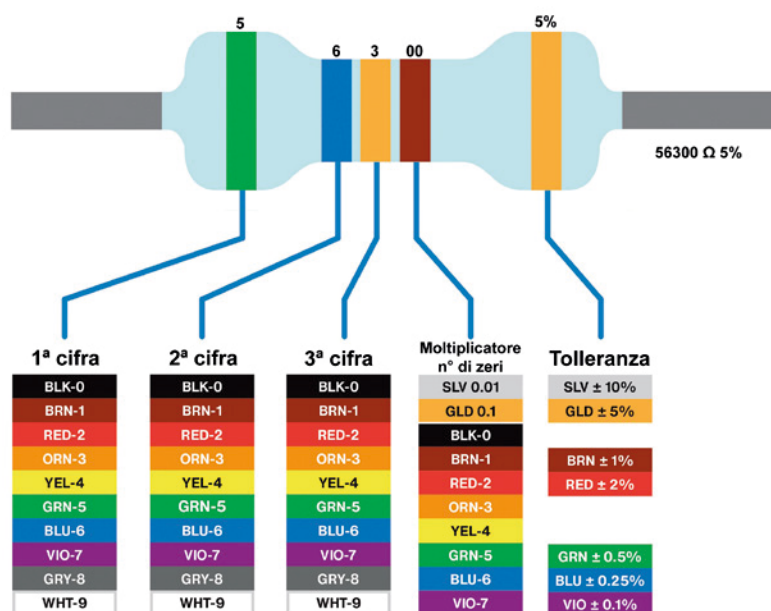
Spiegando il funzionamento del multiplex abbiamo utilizzato il termine sequenza, perché le cinque operazioni suddette avvengono separatamente, e per ogni digit che si accende, gli altri quattro restano spenti; però noi li vedremo accesi contemporaneamente e ciò è dovuto alla frequenza elevata con la quale il microcontrollore ripete la sequenza

ed al fenomeno della persistenza visiva dell'occhio umano, che non consente di percepire cambiamenti molto rapidi, ovvero che ci fa vedere insieme immagini composte molto rapidamente.

Seguendo questo ragionamento si potrebbe obiettare che il quinto digit allora dovrebbe accendersi sempre, quando gli altri sono spenti, durante il multiplexing; in effetti sarebbe così ma, come abbiamo detto, ci pensa il firmware a tenere spento il digit quando si misurano frequenze inferiori a 10.000 Hz, semplicemente tenendo il LOW tutti i pin che pilotano i 7 segmenti ed il DP, nel momento in cui il suo catodo sarebbe attivo. Osservando lo schema elettrico possiamo notare che il segnale da misurare, applicato al connettore a vite H2, giunge al pin 3 (RA4) del micro tramite il diodo D1; la sua funzione è quella di tagliare l'eventuale componente negativa del segnale, riducendola a 0,6÷0,9 V, valore tollerato dall'ingresso del microcontrollore; lo scopo è quello di far leggere al frequenzimetro anche segnali non logici, infatti vengono misurati anche segnali sinusoidali di questo tipo, che altrimenti non potrebbero essere applicati, perché potrebbero danneggiare il microcontrollore. Purtroppo questo espediente ha un costo notevole, in termini di prestazioni, infatti durante i nostri test siamo riusciti a misurare una frequenza massima di 12-13 MHz (sia onda quadra che sinusoidale). Se si ha necessità di andare oltre tale frequenza, occorre sostituire il diodo D1 con un ponticello collegando quindi il pin 3 direttamente al connettore d'ingresso.

Chiaramente in questo caso si possono applicare all'ingresso solo segnali digitali con livello LOW corrispondente a 0 V; però così si va anche oltre i 50 MHz garantiti, infatti abbiamo rilevato senza problemi segnali fino a 54 MHz.

Un'importante funzionalità del nostro Frequenzimetro è la possibilità di misurare la frequenza fondamentale dei quarzi. È previsto un ingresso specifico, costituito dall'header JP1, sul quale bisognerà inserire (nei due pin esterni) il quarzo da misurare. Il circuito necessario per far oscillare il quarzo è un oscillatore Colpitts standard. Il transistor U2 fornisce il guadagno, mentre la retroazione dell'oscillatore avviene tramite C2 e C3. L'uscita dell'oscillatore viene presa dall'emettitore di U2 tramite C6, che taglia la componente continua e porta il segnale al pin 3 del microcontrollore; l'ingresso diretto e quello per i quarzi non sono utilizzabili indifferentemente, per tale ragione, il circuito prevede i tre pin di J1 sui quali andrà inserito un cappuccio jumper per collegare l'uno o



l'altro ingresso al pin 3 del micro.

Il pulsante S1, assieme al resistore di pull-up R2, serve per entrare nella modalità di programmazione del microcontrollore, ne approfondiremo l'uso nella parte relativa al funzionamento.

Chiudiamo la descrizione dello schema elettrico con la sezione di alimentazione. Il circuito richiede una tensione stabile e filtrata di 5V (ma può funzionare anche a 3,3 V, sebbene perdendo in prestazioni), allo scopo è stato dotato di un regolatore lineare LDO (Low Dropout) che permette di applicare al jack JK1 una tensione (va benissimo un qualsiasi alimentatore switching collegabile alla rete elettrica) tra 5 e 12 Vcc. Il circuito ha un consumo ridottissimo, che non supera i 25 mA.

MONTAGGIO

Come detto si tratta di un kit da montare, quindi valgono le regole consuete, si inizia dai componenti più piccoli, nell'ordine: diodi, resistori, quarzo, condensatori, zoccolo, transistor, pulsante, connettori vari ed infine il display; quest'ultimo ha i pin molto lunghi, se si usa il contenitore in dotazione essi vanno inseriti completamente nel PCB, in caso di un eventuale montaggio sul pannello di un



Fig. 6
Codifica dei resistori a 5 bande.

Fig. 7
Settaggio del jumper J1.

→ **Fig. 8**
Tabella delle
gamme di
frequenza.

Portata	Risoluzione	Visualizzazione sul display	Punto decimale
9.999 kHz	1 Hz	X.XXX	Lampeggio 1 sec.
99.999 kHz	1 Hz	XX.XXX	Lampeggio 1/2 sec.
999.99 kHz	10 Hz	XXX.XX	Lampeggio 1/4 sec.
9.9999 MHz	100 Hz	X.XXXX	fisso
50.000 MHz	1 kHz	XX.XXX	fisso

contenitore diverso, è possibile stabilire a quale distanza deve stare dal PCB. L'identificazione dei componenti è piuttosto semplice; i diodi sono tutti uguali, bisogna solo fare attenzione alla loro polarità, facendo combaciare la fascetta che hanno su uno dei due lati con quella serigrafata sul PCB. I condensatori sono identificabili con le sigle stampigliate sul loro corpo, le abbiamo riportate tra parentesi nell'elenco componenti. I resistori sono di precisione all'1% di tolleranza ed hanno cinque bande; per la loro identificazione si può far riferimento al codice dei colori, con relativo esempio, riportato in **Fig. 6**; in ogni caso il loro valore si misura molto facilmente con un qualsiasi multimetro digitale o anche un tester analogico. L'integrato 7555 ed i due transistor S8050 hanno la stessa forma, ma sulla loro faccia piatta è stampata la loro sigla, quindi basta leggere. Infine il display deve essere montato seguendo l'orientamento della serigrafia sul PCB e sul piano di montaggio, facendo attenzione alla posizione dei puntini decimali posizionati in basso a destra in ognuno dei 5 digit.

USO DELLO STRUMENTO

Se il montaggio è stato effettuato correttamente sarà sufficiente collegare un qualsiasi alimentatore di rete da 5÷12 V 0,5 A (va benissimo il modello 8240-TIP009 reperibile su futurashop.it) per vedere tutti i digit accendersi per un istante e poi spegnersi, lasciando acceso solo il quarto che

normalmente mostrerà uno zero.

Se invece il display dovesse mostrare un qualsiasi valore maggiore di zero occorrerà fare un'operazione di azzeramento via menu, vedremo più avanti come.

L'uso dello strumento è molto semplice, essendo facilitati dal fatto che il software seleziona il gate automaticamente, in base al valore misurato, per ottenere la massima risoluzione possibile. La prima operazione da fare è posizionare correttamente il jumper su J1; per la lettura di segnali applicati al connettore H2, il jumper va posizionato come nel particolare della **Fig. 7** a sinistra (verde); se invece bisogna controllare la frequenza di un quarzo allora il jumper va posizionato come nel particolare a destra (rosso).

A questo punto basta applicare un segnale di ampiezza sufficiente all'ingresso H2 o, nel secondo caso, inserire un quarzo nell'header JP1, per leggere istantaneamente il valore della frequenza. A proposito della lettura automatica, nella **Fig. 8** riportiamo la Tabella delle gamme di frequenza, con relativa visualizzazione sui 4 o 5 digit ed il comportamento del punto decimale; come si può notare la prima portata sfrutta solo 4 digit, dalla seconda in poi invece viene utilizzato anche il quinto.

Abbiamo testato una serie di quarzi standard per verificare il corretto funzionamento dell'oscillatore e del Frequenzimetro stesso.

Nella **Fig. 9** abbiamo riportato la tabella dei risultati di questi test.

Il quarzo da 2 MHz non viene letto, come ci si aspettava, infatti nelle caratteristiche tecniche abbiamo specificato che la frequenza minima è 4 MHz. I quarzi da 4, 10 e 20 MHz vengono letti regolarmente con una precisione di ± 1 digit, mentre quello da 40 MHz mostra un valore di 13,334 MHz; anche in questo caso si rientra in quanto riportato nelle specifiche, cioè che l'oscillatore lavora sull'armonica fondamentale, quindi è chiaro che il quarzo provato da noi lavora in terza armonica ($13,333 \times 3 = 39,999$).

→ **Fig. 9**
Risultati dei test di
lettura dei quarzi.

Test misurazione quarzi	
Valore nominale	Valore misurato
2 MHz	0
4 MHz	4,0001 MHz
10 MHz	9,9999 MHz
20 MHz	19,999 MHz
40 MHz	13,334 MHz

S1 breve	S1 lungo	Display
	Prog - apertura del menu	
Quit	uscita dal menu	
PSAVE	NoPSAVE (o viceversa) e Quit	
Add	Aggiunta del valore visualizzato e Quit	
Sub	Sottrazione del valore visualizzato e Quit	
Zero	Azzerà l'offset aggiunto/sottratto con Add/Sub e Quit	
Table	455.00 - 3.9990 - 4.1943 - 4.4336 - 10.700 e Add/Sub	

Fig. 10
Menu di
programmazione.

IL MENU DI PROGRAMMAZIONE

Diamo ora uno sguardo al menu di programmazione; in premessa bisogna ricordare che una pressione breve del pulsante S1 serve per passare alla voce di menu successiva o per navigare all'interno della voce di menu (nel caso di "Table"), mentre per attivare la voce di menu selezionata occorre una pressione lunga (circa 2 secondi). Per entrare nel menu occorre una pressione lunga del pulsante S1, appare subito la voce "**Prog**", rilasciando il pulsante appare la voce "**Quit**", per uscire senza modificare nulla basta una pressione lunga in questa fase. Nella **Fig. 10** abbiamo inserito le voci di menu in sequenza e la funzione svolta con la pressione lunga di S1; inoltre nell'ultima colonna abbiamo riportato le foto del display in quanto, essendo a 7 segmenti e non alfanumerico, non sempre le parole sono di immediata lettura.

Una prima pressione breve di S1 passa alla voce "**PSave**" o "**NoPSave**", nel primo caso è attivo il risparmio energetico, per cui dopo 15 secondi dall'ultima lettura, in mancanza di variazioni, il display si spegne, nel secondo caso invece il display resterà sempre acceso, anche in assenza di segnale in ingresso. Una pressione lunga di S1 provocherà il passaggio da una opzione all'altra e poi l'uscita dal menu.

Una seconda pressione breve di S1 farà scorrere il menu sulla voce successiva "**Add**"; questa opzione serve per aggiungere alla lettura reale un valore prefissato (offset), rappresentato dal valore visualizzato sul display nel momento in cui si entra nel

menu oppure da uno dei valori predefiniti presenti in "**Table**". È sufficiente una pressione lunga di S1 per memorizzare tale valore e trovarselo stabilmente aggiunto ad ogni lettura effettuata. Un esempio chiarirà meglio questa opzione: poniamo di aver necessità di aggiungere 1.000 Hz ad ogni lettura del Frequenzimetro; la prima operazione da effettuare è quella di applicare un segnale da 1000 Hz al Frequenzimetro, quindi si entra nel menu, si seleziona l'opzione "**Add**" e poi una pressione lunga di S1, da questo momento qualunque sia la frequenza letta mostrerà 1000 Hz in più, quindi 10 kHz diventeranno 11.000 kHz, 1 MHz diventerà 1.0010 MHz, ecc. Se invece la frequenza da aggiungere rientra in quelle memorizzate nel microcontrollore, e disponibili nel sottomenu "**Table**", allora non serve alcun segnale in ingresso, basta selezionare il valore in "**Table**" e poi una pressione lunga di S1, il menu mostra la voce "**Add**", una nuova pressione lunga e la programmazione sarà fatta. Da "**Add**" una pressione breve di S1 porta alla voce "**Sub**", per la quale vale tutto quanto spiegato per "**Add**" con la sostanziale differenza che le frequenze saranno sottratte alla lettura reale.

Da "**Sub**" una pressione breve di S1 porta alla voce "**Zero**", il cui scopo è quello di azzerare l'eventuale offset aggiunto o sottratto in precedenza. Nota bene: potrebbe capitare che il Frequenzimetro mostri un valore errato rispetto a quello atteso (ovviamente bisognerà essere sicuri della precisione della fonte di generazione del segnale misurato) anche in assenza di segnale, probabilmente riporta

Fig. 11
Tabella dei valori
Add-Sub in
EEPROM

Frequenza	HEX Add	HEX Sub	Note Sub
455000 Hz	00 06 F1 58	FF F9 0E A8	4294967296-455000
3999000 Hz	00 3D 05 18	FF C2 FA E8	4294967296-3999000
4194304 Hz	00 40 00 00	FF C0 00 00	4294967296-4194304
4433619 Hz	00 43 A6 D3	FF BC 59 2D	4294967296-4433619
10700000 Hz	00 A3 44 E0	FF 5C BB 20	4294967296-10700000

un offset inserito in fase di test del microcontrollore, da parte del produttore del kit. Per eliminarlo sarà sufficiente usare la funzione **"Zero"**.

Da **"Zero"** una pressione breve di S1 porta alla voce **"Table"** che, come detto, contiene una serie di valori prememorizzati in EEPROM, raggiungibili con una pressione lunga di S1 e precisamente: 455,0 (kHz) 3,9990 (MHz) 4,1943 (MHz) 4,4336 (MHz) e 10,700 (MHz); una pressione breve di S1 permette di navigare tra i 5 valori, una pressione lunga di S1 seleziona il valore visualizzato.

A questo punto il menu si porta automaticamente sulla voce **"Add"**, se si vuole aggiungere la frequenza selezionata alla lettura basta una pressione lunga di S1, se invece la si vuole sottrarre serve prima una pressione breve per passare a **"Sub"** e poi una pressione lunga per attivare il valore; infine, se non si vuole applicare il valore selezionato, basteranno una serie di pressioni brevi su S1 fino a visualizzare **"Quit"** e poi una pressione lunga per uscire dal menu.

Apriamo ora una parentesi prettamente tecnica: i valori di offset di frequenza vengono salvati come numeri interi a 32 bit nella EEPROM del PIC (nelle prime quattro posizioni di memoria della EEPROM, primo byte alto, ultimo byte basso). Se non si dispone di un generatore di segnale per produrre la frequenza di offset per la programmazione, oppure non si può usare la frequenza del BFO dell'apparato radio, è possibile inserire il valore di offset con un programmatore PIC adatto. Occorre convertire la frequenza in Hertz (direttamente per **"Add"**, sottraendola da 4.294.967.296 per **"Sub"**) in un numero esadecimale e immettere questo valore nel campo EEPROM DATA del programmatore PIC. Alcuni tipi di programmatore richiedono l'abilitazione dell'editor HEX prima di digitare i valori nella finestra della memoria. Nella **Fig. 11** troviamo l'elenco delle frequenze prememorizzate nella EEPROM del PIC con le relative corrispondenze esadecimali per la funzioni **"Add"** e **"Sub"**.

Per la conversione si può ricorrere ad un convertitore on line come per esempio: <https://it.calcuworld.com/convertitori/convertitore-esadecimale/>.

CONCLUSIONI

Abbiamo letteralmente sviscerato un semplice ed economico strumento di misura disponibile in scatola di montaggio e destinato quindi all'autocostruzione; come abbiamo visto, si è ben comportato con ogni tipo di test e quindi possiamo tranquillamente consigliarlo a chi abbia necessità di misurare delle frequenze senza dover per forza fare un investimento nell'acquisto di uno strumento professionale.

La versatilità di questo piccolo strumento è eccellente, tanto che, per esempio, montato nel contenitore in dotazione è un ottimo tester per quarzi; nel nostro caso, una configurazione minimale permette di integrarlo nello stesso box di uno strumento che genera segnali digitali, così da ottenere la visualizzazione dei segnali e implementare un apparato completo.

Dotandolo invece di un semplice stadio di preamplificazione (sulla Rete si trovano moltissimi circuiti di questo tipo) può essere utilizzato per la lettura di radiofrequenze; infine il basso consumo consente di realizzarlo in versione portatile a batteria. Insomma, i limiti sono solo quelli imposti dalla nostra fantasia! □



Cosa occorre?

Il frequenzimetro è disponibile in kit (cod. FRE50MHZ) a Euro 17,90.
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>

Strumenti di misura per veri professionisti!



Rilevatore da parete 3 in 1 (legno, metalli, cavi elettrici)

Rileva la presenza di cavi elettrici sotto tensione (con passaggio di corrente), strutture in legno (telai, travetti, barre, ecc), oggetti metallici (travi, telai, tubi, tondini, perni, barre in metallo). Dispone di ampio display retroilluminato, buzzer di segnalazione, pulsante di calibrazione automatica. Si spegne automaticamente dopo 3 minuti di inutilizzo. Alimentazione con batteria da 9 V (acquistabile separatamente), dimensioni 156x70x26 mm.

€29,50

Cod. FR760

Rilevatore da parete professionale (legno, metalli, cavi elettrici)

Rileva la presenza di cavi elettrici sotto tensione (con passaggio di corrente), strutture in legno (telai, travetti, barre, ecc), oggetti metallici (travi, telai, tubi, tondini, perni, barre in metallo). Dispone di ampio display LCD retroilluminato, buzzer di segnalazione, spegnimento automatico dopo 5 minuti di inutilizzo. Alimentazione tramite batteria al litio (integrata) da 3,7 volt 1500 mAh. Dimensioni 195x90x35 mm.



- ✓ Professionale
- ✓ Batteria al litio ricaricabile integrata da 3,7 V - 1500 mAh con autonomia fino a 10 ore
- ✓ Case robusto
- ✓ Prestazioni migliorate

€78,00

Cod. FR801

Misuratore di distanza fino a 80 m con LASER verde e messaggi vocali.

Metro LASER multifunzione compatto con display LCD retroilluminato, messaggi vocali, menu a icone, livella digitale, memoria interna (in grado di memorizzare fino a 100 misurazioni) e batteria ricaricabile agli ioni di litio. È in grado di effettuare le seguenti misurazioni: distanza (da 20 cm a 80 metri), area, volume, misurazione pitagorica, misurazione verticale e molte altre. Visualizza lo stato di carica delle batterie; si spegne automaticamente se inattivo (laser dopo 30 secondi, strumento dopo 2 minuti). Grado di protezione IP54. Dimensioni 120x50x26 mm.



€79,00

Cod. FR798

Misuratore di distanza LASER da 20 cm a 30 m

Compatto metro LASER con display LCD retroilluminato. Permette di misurare in modo estremamente facile e veloce la distanza (da 20 cm a 30 metri), l'area e il volume. Dispone di spegnimento automatico del LASER e del dispositivo; visualizza anche lo stato di carica delle batterie. Alimentazione con 2 batterie mini stilo da 1,5 volt tipo AAA (incluse). Dimensioni 105x34x20 mm.



€27,00

Cod. FR797

Prezzi IVA inclusa



Fishino

Design With Simplicity

I TUOI PROGETTI DIVENTANO WIRELESS

Fishino UNO, Fishino MEGA, Fishino32, Fishino GUPPY e Fishino SHARK sono schede di sviluppo Arduino-compatibile dotate di WiFi e lettore di schede microSD. Grazie ad un completo pacchetto di librerie (scaricabile dal sito www.fishino.it), si ha la possibilità di gestire tutti i propri progetti dal web in maniera semplice e veloce!

Fishino UNO



cod. FISHINOUNO
€ 36,00

- Controller: ATmega328
- Alimentazione:
 - da 7 a 12 Vdc (tramite plug)
 - 5 Vdc (tramite porta USB)
- Compatibile al 100% con Arduino UNO
- Modulo WiFi
- Interfaccia per scheda MicroSD
- Modulo RTC con batteria di mantenimento
- Sezione di alimentazione a 3,3V potenziata
- Compatibile con shield e schede millefori
- Peso 25 g
- Dimensioni 76,5 x 53,5 x 14 mm

- Controller: ATmega2560
- Alimentazione:
 - 3,6 Vdc (tramite batteria esterna al litio)
 - da 3,5 a 20 Vdc (tramite plug)
 - 5 Vdc (tramite porta USB)
- Circuito di ricarica per batteria al litio
- Compatibile al 100% con Arduino MEGA
- Modulo WiFi
- Interfaccia per scheda MicroSD
- Modulo RTC con batteria di mantenimento
- Stadio di alimentazione switching (5Vdc e 3,3Vdc)
- Compatibile con shield e schede millefori
- Peso 37 g
- Dimensioni 101,5 x 53,5 x 15 mm



Fishino MEGA



cod. FISHINOMEGA
€ 55,00

Fishino32



cod. FISHINO32
€ 65,00

- Controller: PIC Microchip a 32 bit
- Alimentazione:
 - 3,6 Vdc (tramite batteria esterna al litio)
 - da 3,5 a 20 Vdc (tramite plug o sull'ingresso Vin)
 - 5 Vdc (tramite porta USB)
- Circuito di ricarica per batteria al litio
- Compatibile con Arduino UNO
- Modulo WiFi
- Modulo RTC con batteria di mantenimento
- Clock 120 MHz, riducibile via software
- Interfaccia per scheda microSD
- Codec audio stereo
- Peso 29 g
- Dimensioni 76,5 x 53,5 x 14 mm

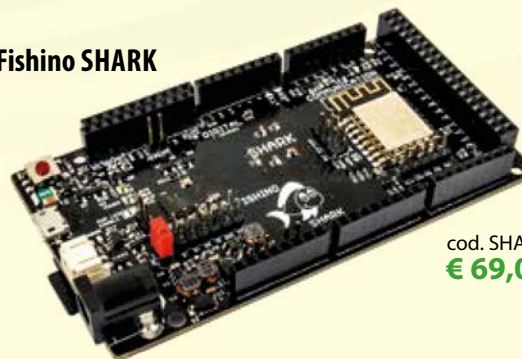
Fishino GUPPY



cod. GUPPY
€ 39,00

- Controller: ATmega328
- Alimentazione:
 - 3,6 Vdc (tramite batteria esterna al litio)
 - da 6,5 a 20 Vdc (tramite PIN VIN)
 - 5 Vdc (tramite porta USB)
- Circuito di ricarica per batteria al litio
- Compatibile al 100% con Arduino NANO
- Modulo WiFi
- Interfaccia per scheda MicroSD
- Peso 10 g
- Dimensioni 75 x 20 x 18,5 mm

Fishino SHARK



cod. SHARK
€ 69,00

- Formato standard Arduino MEGA
- Processore PIC32MX470F512
- 512 kB di ROM
- 128 kB di RAM
- Frequenza di clock di 105 MHz
- Interfaccia USB nativa, sia device che host
- RTC incorporato
- Lettore per schede microSD
- Modulo WiFi integrato
- Codec audio

- Tensione di alimentazione: 3-20Vdc
- Alimentazione a batteria, plug e/o via connettore USB
- Funzionamento interno a 3,3V
- Pin DIGITALI 5V-tolerant
- Ricarica automatica di una batteria LiPo
- Spegnimento da software con processore in standby
- Wake-up tramite pin esterno o in tempi prefissati

Fishino PIRANHA



cod. PIRANHA
€ 49,00



- Alimentazione:
 - 3,7 Vdc (tramite batteria esterna al litio)
 - da 3,5 a 20 Vdc (tramite plug)
 - da 3,5 a 20 Volt sull'ingresso Vin
 - 5 Vdc (tramite porta USB)
- Circuito di ricarica per batteria al litio
- Compatibile al 100% con Arduino MKR1000
- Modulo WiFi
- Interfaccia per scheda MicroSD
- Peso 10 g
- Dimensioni 62 x 25 x 16 mm

Prezzi IVA inclusa.

FISHINO

CONTROLLA IL LIVELLO CON **TELEGRAM**

di EMANUELE SIGNORETTA



Fishino, un bot Telegram e un sensore a ultrasuoni permettono di monitorare il livello dell'acqua delle cisterne di accumulo dell'acqua piovana.

Con il passare degli anni alcune criticità stagionali si acuiscono sempre più e, nel caso della stagione calda, diventa rilevante il problema della siccità. Indistintamente dalla zona, il caldo torrido dei mesi estivi lascia ogni anno i rubinetti di molte famiglie e aziende a secco.

Le cause sono numerosissime: agricoltura intensiva, allacci abusivi, prosciugamento delle fiumare, afflusso di turisti, ma anche l'enorme dispersione negli acquedotti. Tutto ciò ci impone di centellinare i consumi idrici, ma spesso è insufficiente, soprattutto in alcuni ambiti. Ecco che divengono utili le cisterne di accumulo, come quelle di acqua piovana, ma in questo caso, trattandosi spesso di serbatoi interrati o poco visibili, si pone il problema di sapere quando iniziare a risparmiare.

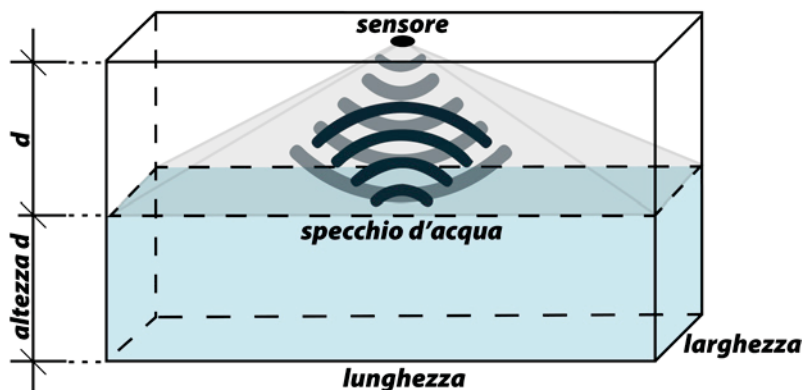
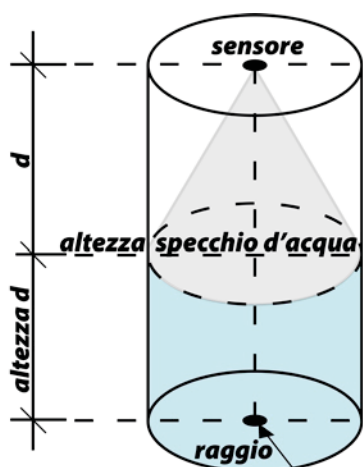


Fig. 1
Misurazione
della quantità di
acqua presente
nel serbatoio a
base circolare o
rettangolare.

In questo articolo proponiamo un progetto che permette di essere costantemente aggiornati sul livello d'acqua presente nel serbatoio, ricevendo delle notifiche su Telegram ogni qualvolta scenda sotto una soglia da noi impostata.

PRINCIPIO DI FUNZIONAMENTO

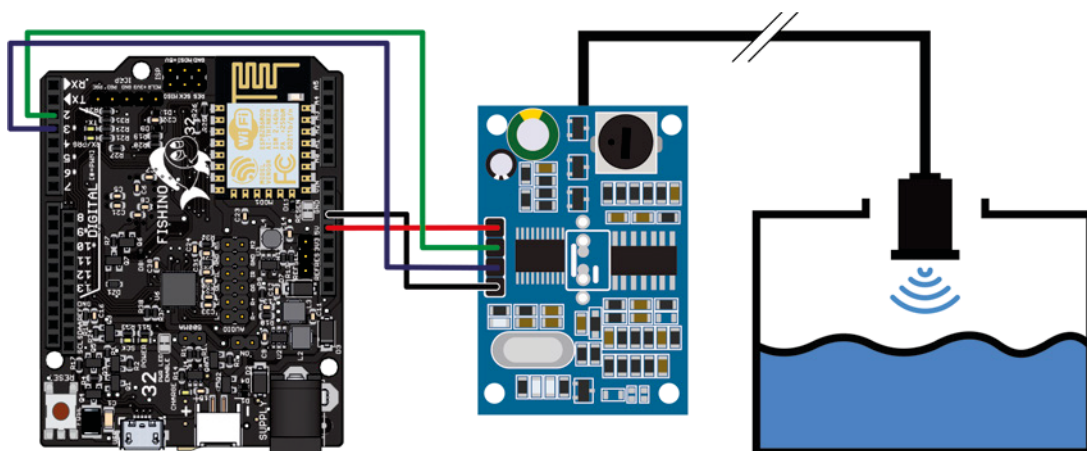
Supponiamo di voler misurare il livello dell'acqua di un comune serbatoio. La maggior parte delle cisterne sono dei parallelepipedi (hanno base rettangolare) o dei cilindri (a base circolare); il volume si determina nel primo caso come il prodotto dei tre lati (base x altezza x profondità) e nel secondo caso come il prodotto dell'area di base per l'altezza ($\pi \times \text{raggio}^2 \times \text{Altezza}$). Per misurare il volume istantaneo della cisterna (**Fig. 1**) è quindi sufficiente calcolare la differenza tra l'altezza massima dell'acqua e la profondità; se ciò viene fatto con un dispositivo

elettronico in grado di compiere la misura della distanza da solidi e liquidi, si posiziona il sensore a una certa altezza, si determina la distanza dal fondo e a quel punto quello è il valore minimo corrispondente a cisterna vuota. Si calcola poi il valore a cisterna piena e quando la distanza dalla cima allo specchio d'acqua supera un certo valore si può generare un allarme, che nel nostro caso viene mandato a tutti gli utenti presenti in una lista configurabile tramite messaggi. Dunque, il sistema che vi proponiamo permette, una volta stabilito qual è il livello al quale è posizionato il sensore a ultrasuoni, di determinare il calo del liquido contenuto.

L'HARDWARE

In questo progetto sono stati usati un sensore ad ultrasuoni impermeabile (1606-JSN-SR04T) come

schema di CABLAGGIO



PIN FISHINO	PIN SENSORE
5V	VCC
GND	GND
2	TRIG
3	ECHO

Tabella 1
Interconnessione
tra sensore
JSN-SR04T e
Fishino.



Fig. 2
Il sensore JSN-SR04T.

quello visibile nella **Fig. 2** e una scheda Fishino a 32 bit (la Fishino32 o la Piranha sono più che sufficienti). Il collegamento è intuitivo ed è realizzabile utilizzando 4 jumper maschio-femmina e seguire lo schema di connessione riportato nella **Tabella 1**. Poiché vengono utilizzati solamente 2 pin di I/O della scheda è possibile apportare delle modifiche al progetto, come collegare un relé per gestire un segnalatore audiovisivo. Il tutto è stato fissato su dei supporti appositamente disegnati e stampati in 3D che può essere sigillato.

Abbiamo utilizzato un alimentatore switching da 12V 1,5A per alimentare le schede (8822-AL-12V1A2).

CREAZIONE DEL BOT TELEGRAM

Per comunicare con la scheda Fishino è necessario creare un bot Telegram con relativa chiave API. Abbiamo già trattato ampiamente l'argomento nei numeri precedenti di questa rivista, perciò ci soffermeremo solamente sulla creazione del bot. Reclamoci dunque sull'app Telegram, cerchiamo e avviamo il **BotFather** (**Fig. 3**) e lanciamo il comando `/newbot`. Riceveremo un messaggio (in inglese) in cui ci verrà chiesto di inserire il nome del bot: scriviamone uno e inviamo il messaggio. Riceveremo un secondo messaggio in cui ci verrà chiesto

di inserire uno username nel formato `"xxxxxxbot"`. Scegliamone uno adeguato e confermiamo, allorché riceveremo un messaggio di avvenuta creazione del bot: nello stesso troveremo un token per l'accesso all'API HTTP: copiamolo e mettiamolo da parte, dato che ci servirà successivamente (**Fig. 3**).

LO SKETCH

Sebbene la configurazione hardware sia molto semplice, la configurazione software è relativamente complessa a causa di una serie di funzioni che servono a manipolare i dati degli utenti: legge-

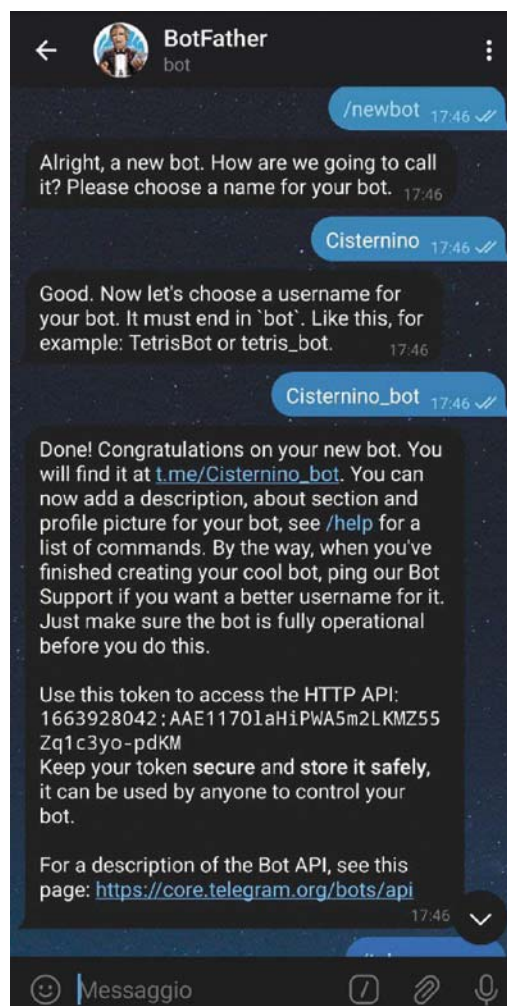


Fig. 3
Creazione del
bot Telegram.

re e scrivere i Telegram Id, i nomi degli utenti, la soglia di allerta e gli intervalli di notifica sull'EEPROM. Abbiamo dovuto inoltre implementare una logica a menù, per permettere all'utente di configurare con maggiore semplicità la scheda e poter modificare alcuni parametri senza caricare nuovamente lo sketch.

Lo sketch è disponibile nella sezione download del sito *elettronica.in.it*, ma per gli eventuali aggiornamenti potete consultare il repository GitHub github.com/signorettae/Cisternino.

Come potete notare nello sketch sono presenti alcuni "TODO", questo poiché durante la fase "embrionale" del codice sono nate altre idee che, anche se non sono state ancora sviluppate, potrebbero esserlo in futuro.

La struttura del codice è la medesima dello sketch *FishGram_SendReceive* della libreria *FishGram.h*.

Analizziamo adesso i vari listati che compongono lo sketch, a partire dal **Listato 1**.

Il **Listato 1** contiene la parte di codice adibita all'inclusione delle librerie e alla definizione del primo set di variabili. Tra le librerie "inusuali" notiamo in particolare la **EVERY.h**, che si occupa di richiamare

un set di istruzioni ad un determinato intervallo di tempo, la **ctype.h** per la gestione dei caratteri delle stringhe (verifiche e modifiche sui formati).

Successivamente vengono dichiarate le variabili, *trigPin* su cui viene generato il segnale degli ultrasuoni ed *echoPin* per la lettura del segnale di ritorno. Segue "a ruota" una serie di *#define* per gli slot EEPROM da utilizzare per la gestione dei dati e degli utenti. A seguire troviamo le definizioni delle variabili. Per chi non lo sapesse, una struttura aggregata può essere volgarmente assunta come una matrice di più variabili. Questa struttura definisce un set di variabili da associare a ciascun utente: troviamo un *uint32_t* per il Telegram ID, due *int*, una per impostare lo stato dello slot ad utilizzato e una per la lunghezza della stringa contenente il nome. La stringa contenente il nome ha una dimensione massima di 10 caratteri più 1 per il terminatore. Partendo dalla struttura definita precedentemente viene creato un numero di utenti pari al valore della variabile *num_utenti*.

Nel **Listato 2** è contenuto il codice che calcola il volume della cisterna. Con il comando *#define* RECT andiamo a definire la sagoma della cisterna,

↓ Listato 1

```
//Aggiunta librerie

#include<Every.h>
#include<FishinoEEPROM.h>
#include <SPI.h>
#include <Fishino.h>
#include <JSONStreamingParser.h>
#include <FishGram.h>
#include <EVERY.h>
#include <ctype.h>
#include <string.h>

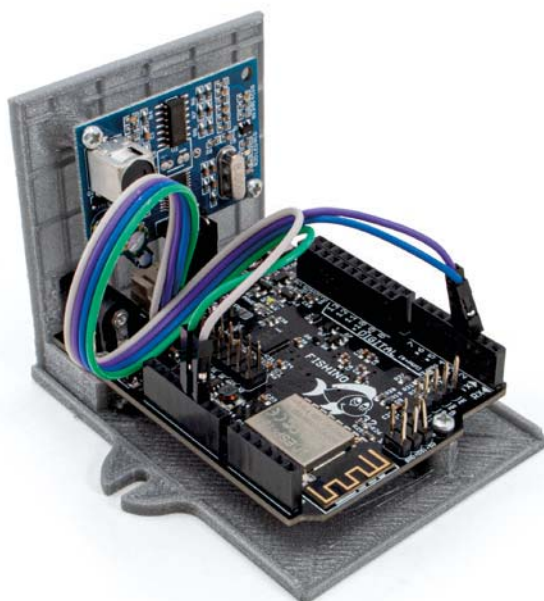
#define trigPin 2 //Pin che genera l'impulso
#define echoPin 3 //Pin per la lettura del ritorno
#define slot_soglia 2 //slot EEPROM di partenza per il salvataggio della soglia di allerta
#define slot_intervallo 9 // slot EEPROM di partenza per il salvataggio dell'intervallo di lettura
#define EEPROM_Data_Start 20 //slot EEPROM di partenza per il salvataggio dei dati degli utenti
#define num_utenti 3 //numero utenti presenti nella configurazione. MODIFICARE IN BASE ALLE NECESSITA'
uint32_t intervallo; //intervallo di lettura
uint32_t soglia; // soglia di allerta
int misura;
int scelta_utente = 0;
int new_tid = 0;
int new_len = 0;
int new_used = 0;
char new_name [11];
long duration;
int menu = 0; //variabile per la gestione della logica a menù
typedef struct { //crea un tipo aggregato
    uint32_t telegram_id = 00000000;
    int used = 0;
    int len = 0;
    char name[11] ;
} User;
User utenti [num_utenti];
```

Listato 2

```
define RECT // OR CIRC
#if defined(RECT)
float larghezza = 1;
float lunghezza = 2;
float altezza = 4;
float superficie = larghezza * lunghezza; //calcola la superficie dello specchio d'acqua
float volume = superficie * altezza * 1000; //calcola il volume della cisterna (in metri cubi)
float valore_massimo_misurazione;
#elif defined(CIRC)
float pi = 3.1415;
float raggio = 0;
float altezza = 2.5;
float superficie = ((raggio) ^ 2 * pi); //calcola la superficie dello specchio d'acqua
float volume = superficie * altezza * 1000; //calcola il volume della cisterna (in metri cubi)
float valore_massimo_misurazione = 2.5;
#else
#error "Formato non supportato"
#endif
```

in questo caso rettangolare. Se la cisterna dovesse avere base circolare basterebbe scrivere `#define CIRC`. In caso di base rettangolare andiamo a definire le tre dimensioni di larghezza, lunghezza e altezza. Sebbene i valori inseriti siano interi, i tipi sono float. Passiamo dunque a calcolare la superficie, moltiplicando la base per l'altezza. Per ottenere il volume moltiplichiamo la superficie per l'altezza e per la costante 1000 (ricordiamo che l'unità di misura è il m³ e che 1m³ = 1.000 litri di H₂O). Definiamo `valore_massimo_misurazione` come l'altezza a cui andremo a fissare il sensore. Il codice è molto simile nel caso in cui la base sia circolare, con le uniche differenze che viene calcolata utilizzando la corrispondente formula per il calcolo del volume.

Il **Listato 3** contiene il codice utilizzato per misurare il livello dell'acqua. Viene creata la funzione *getSensorDistance()*. Questa restituisce un numero intero, ottenuto eseguendo una serie di calcoli. Per prima cosa viene creata una variabile, *distance*, successivamente il pin dell'innesco viene portato a livello logico basso per 5 microsecondi. A seguire viene riportato a livello logico alto per 10 microsecondi, in modo tale che il sensore possa generare l'impulso ad ultrasuoni. Infine viene letto il segnale dal pin di ritorno e si calcola il tempo impiegato per percorrere tutto il percorso. Questo valore, moltiplicato per 0,034 (derivato dalla velocità del suono in km/h) e diviso per 2 (a noi interessa conoscere la metà del tragitto totale percorso) ci restituisce la distanza tra il sensore e lo specchio d'acqua.



 La centralina elettronica cui collegare il cavetto del sensore a ultrasuoni.

Listato 3

```
int getSensorDistance(void) {
    int distance;
    digitalWrite(trigPin, LOW);
    delayMicroseconds(5);
    // Fa generare l'impulso al sensore settando al livello logico alto il trigPin per 10 microsecondi
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    // Legge l'echoPin. pulseIn() restituisce la durata dell'impulso (lunghezza dell'impulso) in microsecondi:
    duration = pulseIn(echoPin, HIGH);
    // calcola la distanza
    distance = duration * 0.034 / 2;
    return distance;
}
```

Il **Listato 4** contiene la parte di codice inerente all'invio dei messaggi agli utenti e dei primi parametri di configurazione.

Per non essere troppo dispersivi abbiamo omissso

alcune funzioni presenti all'interno del codice; maggiori informazioni le potete comunque trovare nello sketch completo scaricabile dal nostro sito insieme ai file del progetto.

Listato 4

```
// fishgram event handler -- just display message on serial port
bool FishGramHandler(uint32_t id, const char *firstName, const char *lastName, const char *message) {
    misura = getSensorDistance(); //legge i dati dal sensore richiamando la funzione creata precedentemente
    TODO Sistemare questo comando
    if (firstName)
        Serial << firstName << " ";
    else
        Serial << F("Qualcuno ");
    Serial << F("ha scritto ") << message << "\n";
    int len = 0;
    while (message[len] != '\0')
        len++;
    char msg_conv[len + 1];
    strcpy(msg_conv, message);
    int i = 0;
    while (msg_conv[i]) {
        msg_conv[i] = tolower(msg_conv[i]);
        i++;
    }
    //String msg_conv = String(message);
    //msg_conv.toLowerCase();
    Serial.print("Il messaggio convertito ai caratteri lowercase è: ");
    Serial.println(msg_conv); //stampa la nuova stringa convertita
    if (menu == 0) { //se la variabile menu è pari a zero, è possibile richiamare questi comandi:
        EVERY((intervallo * 60000)) { //moltiplica la variabile intervallo per 60000. questo perchè l'intervallo
            indica i minuti, e in ciascun minuto sono presenti 60k ms
            int check = 3; //per evitare che il sistema vada in loop sulla lettura dei dati abbiamo impostato un
            massimo di 3 tentativi di lettura. Se tutti e tre dovessero andare
                // a vuoto il sistema ci informerà dell'errore di lettura.
            if (check != 0 && check != -1) {
                int mis = getSensorDistance();
                if (mis == 0 || mis > valore_massimo_misurazione) {
                    check--;
                    delay(1000); //attende 1s per evitare che gli ultrasuoni vadano a creare interferenze}
                }
                int vol_misurazione = (mis / 100) * superficie; //la misurazione è in cm. Dividiamo per 100 per
                ottenere la misurazione in metri
                if (intervallo != 0 && vol_misurazione <= soglia) { //controlla che le notifiche non siano disa-
                    bilitate e che il valore letto sia inferiore alla soglia di allerta
                    for (int i = 0; i < num_utenti; i++) {
                        if (utenti[i].used == 1) {
                            String messaggio_alert = "Ciao ";

```

Listato 4 (continua)

```
        messaggio_alert += String (utenti[i].name);
        messaggio_alert += " , il livello dell'acqua è inferiore rispetto alla soglia impostata.\n";
        messaggio_alert += "Il livello è ";
        messaggio_alert += String (mis);
        messaggio_alert += " litri.";
        FishGram.sendMessage(utenti[i].telegram_id, messaggio_alert.c_str());
    }
}
}
check = -1;
}
if (check == 0){ //Se si ottengono 3 misurazioni consecutive errate, l'utente viene informato
dell'errore
    for (int i = 0; i < num_utenti; i++) {
        if (utenti[i].used == 1) {
            String messaggio_alert = "Ciao ";
            messaggio_alert += String (utenti[i].name);
            messaggio_alert += " , ho provato a effettuare le misurazioni non risultavano corrette. \n";
            messaggio_alert += "Prova a controllare che il sensore sia funzionante e posizionato cor-
rettamente";
            FishGram.sendMessage(utenti[i].telegram_id, messaggio_alert.c_str());
        }
    }
}
}
[...]
```

```
    else if (strcmp(msg_conv, "imposta soglia") == 0) { // "imposta soglia" per impostare la soglia di
allerta
        String risposta_menu_soglia = "Sei entrato nel menu' per la configurazione della soglia di allerta.\n";
        risposta_menu_soglia += "L'attuale valore è: ";
        risposta_menu_soglia += String(soglia);
        risposta_menu_soglia += ".\nInserisci il valore dell'acqua al di sotto del quale riceverai le noti-
fiche.";
        FishGram.sendMessage(id, risposta_menu_soglia.c_str());
        menu = 1; //entra nel menù 1
    }
    else if (strcmp(msg_conv, "imposta intervallo") == 0) { //imposta l'intervallo di verifica del livello
dell'acqua
        String risposta_menu_intervallo = "Sei entrato nel menu' per la configurazione dell'intervallo di
tempo di verifica.\n";
        risposta_menu_intervallo += "Il valore attuale è: ";
        risposta_menu_intervallo += String(intervallo);
        risposta_menu_intervallo += ".\nOgni quanti minuti vuoi ricevere le eventuali notifiche?\nN.B.: Im-
postando l'intervallo a 0, verranno disabilitate le notifiche.";
        FishGram.sendMessage(id, risposta_menu_intervallo.c_str());
        menu = 2; //entra nel menù 2
    }
    else if (strcmp(msg_conv, "utenti") == 0) { //mostra la configurazione attuale degli utenti e chiede
se vogliamo apportare delle modifiche
        String risposta_user1 = "Sei entrato nel menu' di configurazione dei destinatari delle notifiche. \n";
        risposta_user1 += "Il numero massimo di utenti e' ";
        risposta_user1 += String(num_utenti);
        risposta_user1 += ".\nPer variarlo è necessario modificare lo sketch.\n ";
        risposta_user1 += "La configurazione attuale e': \n";
        for (int i = 0; i < num_utenti; i++) { //esegue un ciclo for per aggiungere alla stringa i dati degli
utenti
            risposta_user1 += "Utente num. ";
            risposta_user1 += String(i + 1); //aggiunge 1 per numerare a partire da 1 anziché dell'indice i (=0)
            risposta_user1 += ". Nome: ";
            // for (int n = 0; n < utenti[i].len && utenti[i].name[n] != '\0'; i++) {
            risposta_user1 += utenti[i].name; // aggiunge il nome dell'utente
            //}
            risposta_user1 += ". Id: ";
            risposta_user1 += String(utenti[i].telegram_id); //aggiunge il corrispettivo Telegram ID
            risposta_user1 += " \n\n";
        }
        risposta_user1 += "\n\n\n";
        risposta_user1 += "Vuoi apportare delle modifiche? (Y/n)";
        FishGram.sendMessage(id, risposta_user1.c_str());
        menu = 3; //entra nel menù 3
    }
}
```

Seguendo l'ordine dei comandi, notiamo che inizialmente il Fishino elabora il messaggio in arrivo, determinandone prima la lunghezza e successivamente creando una stringa di dimensioni incrementate di 1 (per il terminatore).

Successivamente nella nuova stringa (*msg_conv*) verranno copiati tutti i caratteri del messaggio, ma convertiti ai caratteri lowercase.

Abbiamo fatto questa scelta in quanto in base al dispositivo che utilizziamo quando inviamo un

↓ Listato 5

```
else if (menu != 0) {
    if (strcmp(msg_conv, "annulla") == 0) { // se siamo entrati in un menù con valore >= 1, possiamo
        tornare alla homepage digitando il comando "annulla"
        FishGram.sendMessage(id, "Fatto, ho annullato la tua richiesta!");
        //reimposta il valore delle variabili temporanee
        new_tid = 0;
        new_name[0] = '\0';
        new_len = 0;
        menu = 0; //ritorna alla homepage
    }
    if (menu == 1) { // menù impostazione soglia
        uint32_t soglia_new = (uint32_t)(atoi(msg_conv)); //converte il messaggio ricevuto in un intero e
        successivamente fa il cast a uint32_t
        String risposta_imp_soglia = "Hai inserito il valore: ";
        risposta_imp_soglia += String(soglia_new);
        if (soglia_new >= volume) //se il valore inserito è maggiore del volume della cisterna, il sistema
        ci avvisa
            risposta_imp_soglia += ".\nIl valore di soglia non può essere maggiore del volume del serbatoio!";
        else { // in caso contrario salva i dati nell'EEPROM, aggiorna il valore della variabile "soglia"
        e ci avvisa
            FishinoEEPROM.write32u(slot_soglia, soglia_new);
            soglia = soglia_new;
            risposta_imp_soglia += ".\nFatto! Ho salvato il valore nell'EEPROM.";
        }
        FishGram.sendMessage(id, risposta_imp_soglia.c_str());
        menu = 0; //ritorna alla "homepage"
    }
    else if (menu == 2) { // menù impostazione intervallo di notifica
        uint32_t intervallo_new = (uint32_t)(atoi(msg_conv)); //converte il messaggio ricevuto in un intero
        e successivamente fa il cast a uint32_t
        String risposta_imp_intervallo = "Hai inserito il valore: ";
        risposta_imp_intervallo += String(intervallo_new);
        FishGram.sendMessage(id, risposta_imp_intervallo.c_str());
        FishinoEEPROM.write32u(slot_intervallo, intervallo_new); //salva il valore nell'EEPROM
        intervallo = intervallo_new; //aggiorna la variabile "intervallo"
        String risposta_imp_intervallo_1 = "\nFatto! Ho impostato scritto correttamente il valore nell'EE-
        PROM.";
        if (intervallo_new == 0) {
            Serial.println("Intervallo impostato a 0. Notifiche disabilitate");
            risposta_imp_intervallo_1 += "\nIntervallo impostato a 0. Notifiche disabilitate"; // Se il valore
            è pari a zero, ci avvisa che le notifiche verranno disabilitate
        }
        FishGram.sendMessage(id, risposta_imp_intervallo_1.c_str());
        menu = 0; //torna alla homepage
    }
    else if (menu == 3) { // menù editing utenti, precedentemente richiamato dal comando "utenti". Ci è
        stato chiesto se vogliamo modificare la configurazione attuale:
        if (strcmp(msg_conv, "n") == 0) {
            FishGram.sendMessage(id, "Fatto, ho annullato la tua richiesta!"); //se vogliamo mantenerla inva-
            riata basta digitare "n" per tornare alla homepage
            menu = 0;
        }
        else if ((strcmp(msg_conv, "y") == 0) || (strcmp(msg_conv, "s") == 0)) { // se vogliamo modificarla
        basta rispondere "y" oppure "s" e ci verrà chiesto quale utente vogliamo modificare
            FishGram.sendMessage(id, "Scegli quale utente aggiungere o aggiornare");
            menu = 4; //entra nel menù 4
        }
    }
}
```


messaggio il primo carattere può essere maiuscolo o minuscolo e soprattutto, per ovviare agli eventuali errori di battitura.

Possiamo riassumere parte delle funzioni dicendo che inviando il messaggio "*distanza*" verrà effet-

tuata una lettura istantanea dei dati e ci verrà recapitata come risposta la distanza che intercorre tra il sensore e lo specchio d'acqua.

Analogamente, come risposta al messaggio "*volume*" riceveremo il volume istantaneo della cisterna,

Listato 6

```
else if (menu == 4) { // se nel menù precedente abbiamo accettato di modificare la configurazione, adesso
dobbiamo scegliere quale utente editare
    scelta_utente = atoi (msg_conv); //converte il messaggio ricevuto in intero
    if (scelta_utente <= 0 || scelta_utente > num_utenti) { //se il numero inserito è <= 0 oppure maggiore
del numero degli utenti il sistema ci avvisa dell'errore e torna nell'homepage
        FishGram.sendMessage(id, "Scelta non valida, riprova!");
        menu = 0; //torna all'home page
    }
    else {
        String risp_scelta_utente = "Hai scelto l'utente #"; //se la scelta è corretta il sistema ci riepiloga
i dati dell'utente scelto: nome e Telegram ID e ci chiede se modificarlo o eliminarlo
        risp_scelta_utente += String (scelta_utente);
        risp_scelta_utente += " ";
        risp_scelta_utente += String (utenti[scelta_utente - 1].name);
        risp_scelta_utente += " ";
        risp_scelta_utente += String (utenti[scelta_utente - 1].telegram_id);
        risp_scelta_utente += "\nCosa vuoi fare? Scrivi D per eliminare, A per aggiungere o per aggiornare.";
        FishGram.sendMessage(id, risp_scelta_utente.c_str());
        menu = 5; // entra nel menù 5
    }
}
else if (menu == 5) { //chiede di inserire il Telegram ID ed entra nel menu 6
    if (strcmp(msg_conv, "d") == 0) { //se alla proposta di eliminare o aggiornare i dati di un utente
premiato rispondiamo "d", questi vengono eliminati
        utenti[scelta_utente - 1].telegram_id = 0; //azzerà il Telegram ID
        utenti[scelta_utente - 1].used = 0; //imposta il valore dell'utente a 0, cioè non utilizzato
        utenti[scelta_utente - 1].len = 0; //azzerà la lunghezza della stringa
        utenti[scelta_utente - 1].name[0] = '\0'; //azzerà la stringa contenente il nome
        FishinoEEPROM.write8u(EEPROM_Data_Start + 17 * (scelta_utente - 1), utenti[scelta_utente - 1].te-
legram_id); //azzerà il Telegram ID sull'EEPROM
        FishinoEEPROM.write8u(EEPROM_Data_Start + 17 * (scelta_utente - 1) + 1, utenti[scelta_utente -
1].used); //imposta il valore dell'utente a 0, cioè non utilizzato sull'EEPROM
        FishinoEEPROM.write8u(EEPROM_Data_Start + 17 * (scelta_utente - 1) + 2, utenti[scelta_utente -
1].len); //azzerà la lunghezza della stringa sull'EEPROM
        FishinoEEPROM.write8u(EEPROM_Data_Start + 17 * (scelta_utente - 1) + 3, utenti[scelta_utente -
1].name[0]); //azzerà la stringa contenente il nome sull'EEPROM
        FishGram.sendMessage(id, "Utente eliminato!"); //ci viene notificata l'avvenuta eliminazione
dell'utente
        scelta_utente = 0; //reimposta il valore della scelta utente a 0
        menu = 0; //torna alla homepage
    }
    else if (strcmp(msg_conv, "a") == 0) {
        FishGram.sendMessage(id, "Inserisci il Telegram ID: ");
        menu = 6; //entra nel menù 6
    }
}
else if (menu == 6) { //inserimento Telegram ID
    int n = 0;
    //TODO implementare controllo caratteri usando msg_conv[n].isAlpha. Così facendo si possono inserire
solo numeri
    new_tid = atoi (msg_conv); //converte il messaggio convertito in intero e salva il valore nella
variabile temporanea "new_tid"
    String risposta_menu6 = "Hai inserito il valore ";
    risposta_menu6 += String (new_tid);
    risposta_menu6 += ".\nSe hai sbagliato il valore scrivi 'annulla', altrimenti inserisci il nome
dell'utente (max 10 caratteri)."; //ci chiede di inserire il nome dell'utente
    FishGram.sendMessage(id, risposta_menu6.c_str());
    menu = 7; //entra nel menù 7
}
else if (menu == 7) { //inserimento nome
```

Listato 6 (continua)

```
int i = 0; // crea una variabile temporanea e la imposta a zero
while (i < 10 && msg_conv[i] != '\0') { // Per questioni di spazio sull'EEPROM il sistema accetta massimo 10 caratteri più il terminatore '\0'. Esegue un ciclo while fino a quando il valore di i è minore di 10 o non viene rilevato il terminatore
    // utenti[scelta_utente - 1].name[i] = msg_conv[i];
    new_name[i] = msg_conv[i]; // copia il carattere di indice i dal messaggio in ingresso al corrispettivo della variabile temporanea "new_name"
    new_len++; // incrementa il valore della lunghezza della stringa new_len
    i++;
}
new_name[new_len + 1] = '\0'; // aggiunge il terminatore
new_len++;
String risposta_menu7 = "Hai inserito il nome ";
risposta_menu7 += String(new_name);
risposta_menu7 += ".\nSe hai sbagliato a inserire il nome scrivi 'annulla'.";
risposta_menu7 += "\nConfermi le scelte inserite? (Y/n)"; // Il sistema ci chiede se vogliamo convalidare i dati inseriti finora
FishGram.sendMessage(id, risposta_menu7.c_str());
menu = 8; // entra nel menù 8
}
else if (menu == 8) { // per salvare i dati inseriti
    if ((strcmp(msg_conv, "y") == 0) || (strcmp(msg_conv, "s") == 0)) { // se si risponde "y" oppure "s" salva i dati sull'EEPROM
        utenti[scelta_utente - 1].telegram_id = new_tid; // passa il valore del Telegram ID dalla variabile temporanea alla variabile definitiva
        FishinoEEPROM.write32u(EEPROM_Data_Start + 17 * i, utenti[scelta_utente - 1].telegram_id); // salva il valore definitivo del Telegram ID sull'EEPROM
        utenti[scelta_utente - 1].used = 1; // imposta lo stato dell'utente a 1 (usato)
        FishinoEEPROM.write8u(EEPROM_Data_Start + 4 + 17 * i, utenti[scelta_utente - 1].used); // salva i dati sullo stato sull'EEPROM
        utenti[scelta_utente - 1].len = new_len; // passa il valore della lunghezza della stringa dalla variabile temporanea alla variabile definitiva
        FishinoEEPROM.write8u(EEPROM_Data_Start + 5 + 17 * i, utenti[scelta_utente - 1].len); // salva il valore definitivo della lunghezza della stringa sull'EEPROM
        for (int n = 0; n < (utenti[scelta_utente - 1].len); n++) // avvia un ciclo for per ciascun carattere della stringa: parte dall'indice i=0 fino alla lunghezza della stringa
        {
            utenti[scelta_utente - 1].name[n] = new_name[n]; // passa il carattere di indice i dalla stringa temporanea alla stringa definitiva
            FishinoEEPROM.write8u(EEPROM_Data_Start + 6 + n + 17 * i, utenti[scelta_utente - 1].name[n]); // salva il carattere di indice i dalla stringa definitiva all'opportuno slot EEPROM
            // Serial.print(utenti[i].name[n]);
        }
        utenti[scelta_utente - 1].name[utenti[scelta_utente - 1].len] = '\0'; // aggiunge il terminatore alla stringa definitiva
        // FishinoEEPROM.write8u(EEPROM_Data_Start + 6 + n + 17 * i, utenti[scelta_utente - 1].name[utenti[scelta_utente - 1].len]); // salva il terminatore sull'opportuno slot dell'EEPROM
        // TODO sistemare salvataggio carattere terminatore
        // for (int i = 0; i <
        FishGram.sendMessage(id, "Fatto, i dati sono stati salvati!"); // comunica l'avvenuto salvataggio
    }
    // strcpy(temp, input); // flash: 2366, SRAM: 254, works
    else if ((strcmp(msg_conv, "n") == 0)) {
        FishGram.sendMessage(id, "Fatto, i dati non verranno salvati!");
    }
    // resetta le variabili temporanee
    new_tid = 0;
    new_len = 0; // TODO probabilmente va settata a 1
    new_name[0] = '\0';
    scelta_utente = 0; // resetta la variabile temporanea della scelta utente
    menu = 0; // torna alla homepage
}
}
return true;
}
```

per i messaggi "*id*" e "*capacità*" riceveremo rispettivamente il nostro Telegram ID e il volume massimo della cisterna.

Dopo aver stampato i risultati della stringa convertita, troviamo uno statement di tipo *if...else*: se la variabile (*menu*) è pari a 0, come da default





Il sistema completo.

vengono richiamate la libreria *EVERY* e un'altra serie di statements *if...else*. In particolare, la funzione *EVERY* ha come argomento un valore, il tempo (in ms): ogni volta che sarà trascorso il tempo verranno eseguite le funzioni all'interno delle parentesi graffe. Verranno infatti eseguite le misurazioni e quest'ultime verranno ripetute per un massimo di tre tentativi intervallati di 1 secondo. Abbiamo fatto questa scelta per non fare andare in stallo il microcontrollore sulla lettura dei dati.

Se ad ogni misurazione il risultato ottenuto non sarà diverso da 0 o superiore al valore massimo della misurazione (per scongiurare eventuali errori), la variabile temporanea verrà decrementata fino a raggiungere il valore 0.

Se raggiungerà il valore 0 tutti gli utenti verranno informati dell'errore di lettura dei dati. Invece, se la misurazione risulterà valida, il sistema controllerà innanzitutto che le notifiche non siano disabilitate e che il volume corrispondente sia inferiore alla soglia impostata.

Quando entrambe le condizioni saranno soddisfatte, verrà eseguito un ciclo *for*, all'interno del quale verranno confezionati i messaggi che verranno inviati agli utenti.

Dopo aver inviato i messaggi, verrà assegnato alla variabile *check* il valore -1, per uscire dal loop.

Se invece il valore dovesse diventare 0, ciascun utente verrà informato dell'errore durante la lettura dei dati e verrà invitato a controllare i collegamenti e la posizione del sensore.

Quando invece inviamo il messaggio "*utenti*" il sistema ci invierà come risposta i dati di tutti gli utenti registrati. In particolare verranno riportati nel corpo della risposta il numero massimo degli utenti configurabili e successivamente l'elenco progressivo degli utenti con il corrispettivo nome e Telegram ID. Ci verrà chiesto se vogliamo modificare la configurazione e il sistema entrerà nell'opzione 3 del menù (che tratteremo a breve).

Il **Listato 5** comprende il codice che gestisce alcune delle opzioni del menù. Il caso particolare è se la variabile *menu* è diversa da 0. Con il messaggio "*annulla*", verremo automaticamente nella "homepage" e tutte le variabili temporanee verranno azzerate. Se la variabile *menu* corrisponde ad 1, logicamente saremo nella modalità per l'impostazione della soglia d'allerta. Se durante questa opzione invieremo un messaggio, quest'ultimo verrà convertito in una variabile intera mediante la funzione *atoi*. Successivamente il sistema controllerà che il valore di quest'ultima non sia superiore alla capacità massima del serbatoio. Se questa lo sarà, riceveremo un messaggio di avviso e verremo riportati nella homepage, altrimenti il valore verrà salvato dapprima sulla variabile *soglia* e successivamente sull'apposito slot dell'EEPROM. Il codice è pressoché uguale nel caso in cui la variabile *menu* sia 2, adibita cioè all'impostazione dell'intervallo di misurazione.

Quando la variabile *menu* è pari a 3 il sistema sarà in attesa di una nostra risposta per quanto riguarda la modifica della configurazione. Nello specifico, se la nostra risposta sarà "*n*" la nostra richiesta verrà annullata e verremo riportati nella homepage. In caso contrario, cioè in cui la nostra risposta è "*y*" oppure "*s*", ci verrà chiesto quale utente modificare, per poi entrare nel menù 4.

Il **Listato 6** è il più corposo del nostro sketch: si occupa dell'intera gestione degli utenti. Partendo dalla logica del menù 4, il sistema è in attesa di ricevere in input un numero, quello dell'utente da modificare. Utilizzando la funzione *atoi*, la stringa in ingresso verrà convertita in intero. Se questo numero risulterà minore o uguale di zero oppure maggiore del numero massimo degli utenti, ci verrà notificata un'inesattezza dei dati inseriti e si tornerà alla homepage. Altrimenti ci verrà comunicato di aver scelto l'utente identificato da quel numero, con il corrispettivo nome e Telegram ID.

Ci verrà dunque chiesto se vogliamo eliminare o modificare l'utente scelto e si entrerà nel menù 5. Notiamo che il comando `utenti[scelta_utente - 1].name` contiene il -1 poiché abbiamo preferito numerare gli utenti partendo da 1, mentre in memoria la numerazione parte da 0. Quando la variabile `menu` è 5 il sistema, già a conoscenza dell'utente oggetto dei nostri interessi si aspetta di sapere se vogliamo eliminarlo o aggiornarlo. Infatti, se invieremo il messaggio "d" verranno resettate tutte le variabili del corrispettivo utente e successivamente i cambiamenti verranno salvati anche sull'EEPROM. Ci verrà comunicata l'effettiva eliminazione dell'utente, verrà resettata la variabile "scelta utente" e si tornerà alla homepage. Se invece invieremo il messaggio "a" ci verrà chiesto di inserire il Telegram ID e si entrerà nel menù 6. Il menù 6 si occupa di ottenere il Telegram ID. In particolare, viene creata la variabile temporanea

"new_tid" che conterrà il valore ottenuto dopo aver passato nella funzione `atoi` il messaggio convertito. Ci verrà comunicato il Telegram ID inserito e ci verrà chiesto contestualmente se vogliamo annullare l'operazione in corso oppure di inserire il nome dell'utente. Si entrerà dunque nel menù 7. Il menù 7 si occupa della gestione del nome utente: viene creata la variabile temporanea "n" che ci serve per copiare all'interno di un'altra variabile temporanea "new_name" al più i primi 10 caratteri del nome. Si è resa opportuna questa scelta in quanto gli slot di memoria nell'EEPROM sono calcolati per contenere al più 10 caratteri (terminatore escluso) e non possiamo invadere i campi riservati ad altri utenti. Ci verrà dunque riportata la stringa (eventualmente troncata) e ci verrà chiesto se vogliamo confermare i dati inseriti, entrando nell'ultima opzione del menù, la 8. Il menù 8 è adibito al salvataggio dei dati inseriti: se inviamo il messaggio "y" oppure "s" il sistema

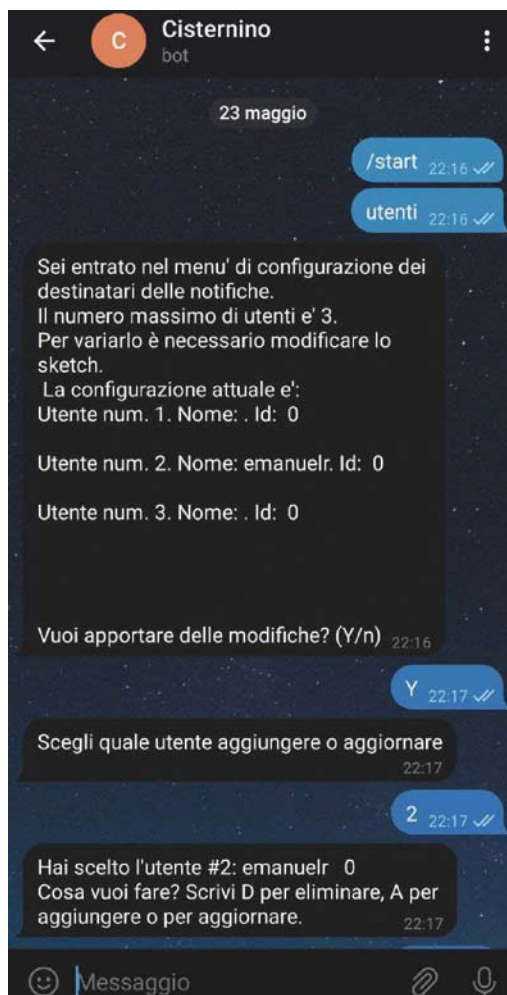
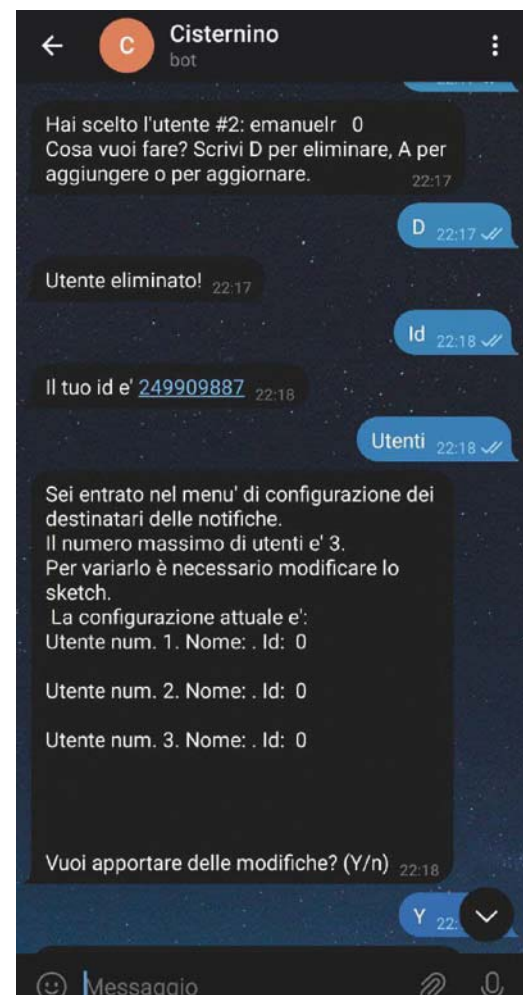


Fig. 4
Screenshot della
configurazione
del bot.



Listato 7

```
void setup(void) {
  [...]
  if (char(FishinoEEPROM.read8u(0)) != '#') { //necessario per pulire l'EEPROM al primo utilizzo. controlla
    che nello slot non sia presente il carattere '#'
    FishinoEEPROM.eraseAll(); //se presente pulisce l'intera EEPROM
    Serial.println("EEPROM resettata"); //ci comunica di sul monitor seriale dell'avvenuto reset dell'EEPROM
    FishinoEEPROM.write8u(0, '#'); //inserisce il carattere '#' nell'EEPROM
  }
  else {
    Serial.println("EEPROM già resettata"); //in caso contrario non fa nulla, si limita ad avvisarci sul
    monitor seriale
    intervallo = FishinoEEPROM.read32u(slot_intervallo);
    soglia = FishinoEEPROM.read32u(slot_soglia);
    for (int i = 0; i < num_utenti; i++) {
      utenti[i].len = FishinoEEPROM.read8u(EEPROM_Data_Start + 5 + 17 * i);
      //se l'EEPROM è stata resettata il valore risulterà 255, mandando in crash lo sketch. Evitiamo tutto
      ciò eseguendo un check sull'apposito slot EEPROM
      if (utenti[i].len != 255) { // se il valore è diverso da 255 legge tutti i dati dall'EEPROM, in par-
        ticolare
        utenti[i].telegram_id = FishinoEEPROM.read32u(EEPROM_Data_Start + 17 * i); //importa il Telegram ID
        utenti[i].used = FishinoEEPROM.read8u(EEPROM_Data_Start + 4 + 17 * i); //imposta lo stato dell'u-
        tente a 1, cioè utilizzato
        for (int n = 0; n < (utenti[i].len); n++) { //esegue un ciclo for per importare i caratteri della
          stringa del nome utente, va dall'indice i=0 alla lunghezza della stringa
          utenti[i].name[n] = char(FishinoEEPROM.read8u(EEPROM_Data_Start + 6 + n + 17 * i)); //importa
          il carattere di indice i dall'EEPROM
        }
        utenti[i].name[utenti[i].len] = '\0'; //aggiunge il terminatore
      }
      else { //in caso contrario
        utenti[i].telegram_id = 0; //imposta il Telegram ID dell'utente a 0
        utenti[i].len = 0; //imposta la lunghezza della stringa a 0.
        utenti[i].name[0] = '\0'; //la stringa contenente il nome dell'utente viene settata come vuota
        utenti[i].used = 0; //lo stato dell'utente viene impostato a 0, cioè non utilizzato
      }
    }
  }
}
```

salverà i dati delle variabili temporanee in quelle "definitive" e successivamente trasferirà i dati nei corrispondenti slot dell'EEPROM. D'altra parte, in caso dovessimo rispondere "n" il sistema ci comunicherà di aver annullato la nostra richiesta. In ogni caso le variabili temporanee verranno resettate e si tornerà alla homepage. La **Fig. 4** propone due schermate di configurazione del bot.

Il **Listato 7** contiene parte del codice contenuto nel *void setup()*. In ordine, viene effettuato un check sull'EEPROM. Al primo utilizzo può capitare che i valori al suo interno siano "sporchi". Per evitare che vengano caricati dati completamente sbagliati, potenzialmente in grado di mandare in crash il firmware abbiamo fatto sì che venisse controllato lo slot 0 dell'EEPROM. Se il carattere al suo interno dovesse essere un cancelletto (#), vorrà dire che l'EEPROM è stata già resettata in precedenza, in caso contrario verrà eseguito un reset della stessa. Successivamente vengono importati dagli appositi slot dell'EEPROM i valori dell'intervallo di misura-

zione e della soglia d'allerta. Infine, mediante un ciclo *for* vengono importati i valori della lunghezza delle stringhe contenenti i nomi. Come potete notare è presente un ciclo condizionale *if*; in particolare se questo valore risultasse pari a 255 (questo valore lo incontriamo se l'EEPROM è stata appena resettata) lo sketch andrebbe in crash, verrebbe comunque portato a 0 ma i dati dell'utente non verrebbero letti, ma lasciati vacanti di default. In caso contrario verranno caricati nell'ordine: il Telegram ID, lo stato dell'utente e uno ad uno (mediante un ciclo *for*) i caratteri del nome dell'utente. Nella **Tabella 2** vengono riassunte le varie opzioni per la variabile menu.

INSTALLAZIONE E FISSAGGIO DEL SENSORE

Una volta caricato lo sketch sul Fishino 32, possiamo eseguire un breve test al banco e verificare che tutto funzioni correttamente: colleghiamo con quattro jumper maschio-femmina ed effettuiamo alcune misurazioni. In assenza di problemi possia-



Tabella 2
Opzioni di menu.

VALORE DELLA VARIABILE MENU	LOGICA CORRISPONDENTE E POSSIBILI RISPOSTE
0	È possibile entrare in tutti i sottomenù
1	Imposta soglia di allerta (valore)
2	Imposta intervallo di verifica (valore)
3	Modifica utenti (si/no)
4	Scelta utente da modificare (valore)
5	Modificare o eliminare utente (a/d)
6	Inserimento Telegram ID (valore)
7	Conferma Telegram ID o inserimento nome (caratteri)
8	Conferma dati (si/no)

mo fissare la Fishino e la scheda del sensore agli appositi supporti stampanti in 3D. Questi supporti andranno fissati all'interno del contenitore in plastica; pratichiamo un piccolo foro su uno dei lati del

box in modo tale di far passare i cavi d'alimentazione e del sensore. Fissiamo il sensore sull'estremità superiore della cisterna utilizzando il supporto stampato in 3D e possiamo dare l'alimentazione alle schede. Creato e configurato il bot, dopo aver correttamente installato il sensore a ultrasuoni, sarà possibile verificare il funzionamento e ricevere i messaggi di stato o di allerta da parte del bot stesso (**Fig. 5**).

CONCLUSIONI

Bene, si conclude qui la presentazione del nostro progetto riguardante il monitoraggio del livello del liquido in una cisterna; l'idea di base può essere adattata a varie situazioni pratiche e a serbatoi e cisterne di tutte le dimensioni, ma anche al monitoraggio del livello di un bacino che sia naturale o artificiale. □

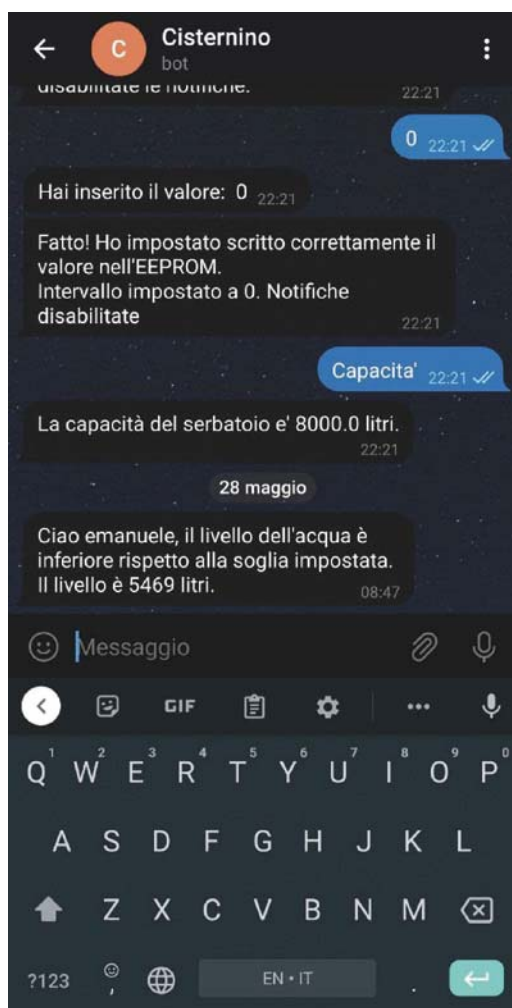


Fig. 5
Messaggio inviato dal bot quando il livello dell'acqua è inferiore alla soglia d'allerta.

Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. La board Fishino (cod. FISHINO32) è in vendita a Euro 65,00, il sensore a ultrasuoni a tenuta stagna (cod. JSN-SR04T) è disponibile a Euro 17,50. Per la stampa delle parti plastiche è disponibile il servizio di stampa 3D direttamente dal sito bit.ly/3DSericeFE. 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>

ALIMENTATORE DA BANCO "RETAIL"



di GABRIELE PERETTO

Testiamo un nuovo modulo alimentatore veramente versatile, in grado di erogare fino a 50V e 5A, dotato di protezioni e di ben 9 memorie per i settaggi più utilizzati.

In ogni laboratorio di elettronica non può mancare almeno un buon alimentatore, regolabile sia in tensione che in corrente. In questo articolo, vi presentiamo un nuovo modulo interamente programmabile. Questo alimentatore, denominato JT-DPS5005 (**Fig. 1**), è innanzitutto un modulo programmabile sia in tensione che in corrente. Possono essere impostati vari livelli di protezione e i dati preimpostati possono essere memorizzati su 9 posizioni (M1-M9) e richiamati a necessità. La posizione M0 è impostata di default all'accensione e tiene in memoria gli ultimi dati utilizzati. In **Tabella 1**, sono riportate le caratteristiche dichiarate dal fornitore. Per le nostre prove, abbiamo utilizzato un accurato alimentatore da laboratorio e in **Tabella 2** sono riportati i dati rilevati.

PANNELLO DI CONTROLLO

La **Fig. 1** illustra i 6 controlli del pannello, che sono 5 pulsanti + una manopola rotativa ad encoder. N.B. il 5° pulsante si attiva

Tabella 1
Dati fornitore modulo DPS 5005.

Tensione di ingresso	6-55 VDC
Tensione di uscita	0-50 VDC
Corrente di uscita	0-5 A
Risoluzione tensione di uscita	0.01 V
Risoluzione corrente di uscita	0.01 A
Ripple a massimo carico (5 A)	100 mVpp
Dimensioni display	1.44"
Dimensioni sede fissaggio pannello	39 x 71 (mm)
Dimensioni modulo	42 x 45 x 80 (mm)
Peso	80 gr.

Tabella 2
Dati controllati in laboratorio DPS 5005

Tensione di ingresso di prova	50 VDC
Tensione di uscita di prova	5-12-24-40 VDC
Corrente di uscita (Iout)	0-5 A
Ripple a 1 A	29 mVpp
Ripple a 5 A	89 mVpp
Efficienza a 5 V – da 1 A a 4 A	da 75 % a 82 %
Efficienza a 12 V – da 1 A a 4 A	da 85 % a 91 %
Efficienza a 24 V – da 1 A a 4 A	da 91.5 % a 94.5 %
Efficienza a 40 V – da 1 A a 4 A	da 94.7 % a 96.7 %
Errore di lettura strumento	± 10 mV - ± 10 mA

premendo la manopola stessa e serve per navigare all'interno del menu SET, sia da pannello principale che da MENU impostazione parametri.

La **Fig. 2**, illustra l'interfaccia del pannello principale, mentre in **Fig. 3** troviamo la pagina relativa ai settaggi. La **Tabella 3** illustra in dettaglio le varie funzioni disponibili.

OPERATIVITA'

All'atto dell'accensione, comparirà una schermata come da **Fig. 2**, dove sono visibili e disponibili tutti i parametri di uscita precedentemente utilizzati (MO), mentre in basso è visualizzata la tensione istantanea di alimentazione del modulo: normalmente sarà tra 40 e 50 Volt a seconda dell'alimentatore di base e non dovrà comunque superare i 55Vcc.

Menu principale

- Per variare la tensione di uscita, premiamo prima il pulsante V/↑, poi premendo ripetutamente il pulsante della manopola, ci si sposta all'interno dei campi di regolazione della tensione (**rif-1 Fig. 2**), e mediante la rotazione della manopola stessa si varia il valore di tensione da un minimo di 0 fino al massimo prelevabile. Questa regolazione può essere effettuata anche con uscita attivata e sotto carico (ON).
- Per variare la corrente di uscita, si preme prima il pulsante A/↓, poi si procede come sopra ma inerentemente al parametro della corrente (**rif-6 Fig. 2**). Anche in questo caso, la regolazione, può essere effettuata con uscita sotto carico (ON). Per entrambe le regolazioni, dopo una decina di secondi di inattività il menù esce dallo stato di modifica e torna in stato di normale funzionamento.
- Pulsante ON/OFF. Premendolo si attiverà l'uscita e la corrente sarà disponibile sui morsetti di prelievo mentre si accenderà di colore verde l'icona (**rif-11 Fig. 2**). Premendolo nuovamente si disattiverà l'uscita e l'icona diventerà di colore rosso.

Fig. 1
Modulo DPS5005.



Tabella 3
Descrizione menu principale e menu impostazioni del modulo DPS 5005

RIF	DESCRIZIONE MENU PRINCIPALE
1	Indica il valore di tensione impostata, in Volt
2	Visualizza il valore della tensione erogato in uscita, in Volt
3	Indica il valore di corrente erogato (carico), in Ampère
4	Indica la potenza erogata in Watt
5	Indica la tensione di alimentazione (IN) del modulo, in Volt
6	Indica il valore di corrente massimo impostato in Ampère
7	Blocco / sblocco dei parametri
8	Indicatore di funzionamento: Normale o Difettoso
9	Modalità di funzionamento (CV) Tensione costante (CC) Corrente costante
10	Numero di memoria richiamato (M1, M2...) campo vuoto = M0 default
11	Stato di uscita (verde: attivo, rosso: spento) attivabile da pulsante ON/OFF
RIF	DESCRIZIONE MENU IMPOSTAZIONI
12	Preset tensione di uscita in Volt
13	Preset corrente di uscita in Ampère
14	Preset Tensione massima di uscita (<i>Over Voltage Protection</i>) in Volt
15	Preset corrente massima di uscita (<i>Over Current Protection</i>) in Ampère
16	Preset della Potenza massima di uscita (<i>Over Power Protection</i>) in Watt
17	Imposta la luminosità del pannello LCD
18	Seleziona la memoria dei vari gruppi di dati da M0 a M9
19	S-INI: OFF / ON => Seleziona lo stato iniziale (<i>ad accensione</i>) di uscita
20	Visualizza i dati di uscita in corso (Volt / Ampère)

- Pulsante SET. Premendolo si entra e/o si esce dal menu: IMPOSTAZIONI (**Fig. 3**).

Menu impostazioni

Premendo il pulsante SET, si entrerà nella pagina delle impostazioni e il cursore si posizionerà in alto a sinistra sulla prima riga (**rif-12 Fig. 3**). Utilizzando i pulsanti A/↓, e V/↑, ci si potrà spostare tra le varie righe e impostare i relativi parametri. Per impostare i parametri bisogna scorrere all'interno di ogni singola riga e per farlo, bisogna premere ripetutamente il pulsante della manopola, per posizionarsi sul valore voluto. Poi si ruota la manopola per incrementare o decrementare il valore fino al raggiungimento del valore desiderato. Alla fine premendo nuovamente il pulsante SET, il cursore si riporterà ad inizio linea, dove si potrà ripetere il procedimento per ogni singola voce da: (**rif-12...rif-19**). L'ultima linea visibile (**rif-18**), della pagina corrente, serve per richiamare e memorizzare i dati impostati,



Fig. 2
Display menu principale.



Fig. 3
Display menu impostazioni.

**Tabella 4**

Caratteristiche dichiarate dal fornitore relative al modulo SW PMT-48V350.

Tensione di uscita	Regolabile da 40 Vdc a 53 Vdc
Assorbimento a vuoto	150 mA
Assorbimento a 5 A / 235 Vac	2.06 A
Rendimento medio a 5 A	94 %
Ripple a vuoto (0 A out)	258 mVpp – 20 mVrms
Ripple a 5 A	274 mVpp – 23 mVrms

da M1 a M9. Scorrendo ancora verso il basso con il tasto A/↓, si accede a una seconda pagina dove vi è ancora una riga (**rif-19 Fig. 3**), dove compare la dicitura S-INI (stato iniziale). Di default questo parametro è impostato OFF e noi consigliamo di lasciarlo così, ma può essere impostato anche su ON. Questo parametro imposta la funzione di uscita all'accensione in modo che eroghi immediatamente la tensione impostata e memorizzata in M0. Diversamente se è impostato su OFF, una volta acceso l'alimentatore, esso non erogherà nessuna tensione/corrente, fintanto che non si premerà il pulsante ON/OFF. Questa funzione risulta molto utile, perché se si utilizza come alimentatore da banco è sicuramente meglio utilizzare l'attivazione dopo aver fatto la verifica dei parametri desiderati e quindi impostare lo stato iniziale su OFF, ma se si utilizzasse il modulo per alimentare un'apparecchiatura dedicata, in questo caso sarebbe sicuramente auspicabile avere la tensione immediatamente disponibile senza dover ogni volta intervenire manualmente e quindi in questo caso va impostato su ON. Al termine della programmazione, premendo nuovamente il pulsante SET, si ritornerà alla pagina principale (**Fig. 2**).

Considerazioni

Come abbiamo visto, questo modulo presenta un numero elevato di pregi, quali: compattezza,

semplicità di utilizzazione, possibilità di programmazione di molteplici parametri, protezione da corto circuiti e impostazione del limite di corrente, funzionamento in CV (tensione costante) o CC (corrente costante) indicatore della Tensione, Corrente e potenza in Watt in uscita. Possono essere programmati e richiamati a necessità, vari parametri precedentemente memorizzati ed infine cosa non da poco, presenta un accettabile valore di "ripple" inferiore a 100 mV pp. Vediamo un esempio pratico applicativo. Disponendo ad esempio di un led che può erogare una potenza di 12 Watt con una tensione di 12V circa e con una CC (corrente costante) max. di 1 Ampère, tipo CREE XM-L2 o altro; per controllarne le caratteristiche, possiamo procedere nel modo seguente.

Impostiamo da pannello principale la tensione di uscita su una tensione superiore, per es. 15V e la corrente a 0.7A (700 mA) per sicurezza. Colleghiamo il led (precedentemente fissato su un dissipatore) rispettando la corretta polarità e quindi premiamo il pulsante ON/OFF. Otterremo valori simili a questi:

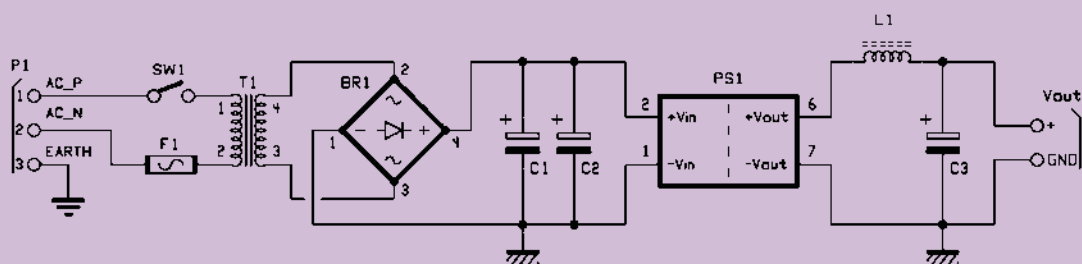
- Tensione di uscita: (auto impostata) 11.82 V
- Corrente di uscita: 0.694 A (auto impostata con livello di sicurezza)
- Potenza assorbita: 8.2 Watt.
- Icona CC (corrente costante) attivata (**rif.9 Fig. 2**)

Se aumenteremo progressivamente la corrente di uscita, noteremo che salirà di poco la tensione e quando arriveremo ad 1 Ampere otterremo circa 12.1 V per 12.1 W. Fatte le dovute riflessioni e disponendo di tutti i dati necessari, abbiamo deciso di utilizzare questo modulo per realizzare un economico ma valido alimentatore da banco. Innanzitutto dovremo scegliere una fonte di alimentazione primaria, di circa 50 Vcc.

ALIMENTAZIONE PRIMARIA

Il sistema base di alimentazione può essere un

Fig. 4
Schema
alimentazione
primaria con
trasformatore
e ponte diodi.



SCHEMA A

semplice trasformatore, seguito da un ponte raddrizzatore e da una serie di condensatori filtro.

In **Fig. 4** è rappresentato uno schema tipo.

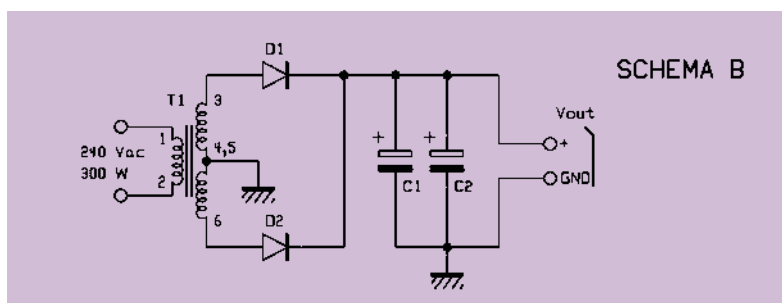
Se disponiamo di un trasformatore con presa centrale potremmo invece adottare lo schema di **Fig. 5**.

Il trasformatore dovrà risultare con secondario da 36V oppure 18+18, il ponte diodi da 80/100V-10A e i condensatori elettrolitici dovranno essere da 10.000 μ F-63V. Questi risultano fondamentali soprattutto in funzione della corrente di uscita.

Con un raddrizzatore a onda intera, otteniamo una tensione pulsante a 100 Hz come da **Fig. 6** e con due condensatori in parallelo da 10.000 μ F otterremo un ripple di circa 1Vpp a 100 Hz come da **Fig. 7**. Teniamo in considerazione che la tensione di uscita aumenterà di 1,41 Volte, per cui per ottenere 50Vcc dovremo scegliere un trasformatore con secondario da: $50/1.41=35.46$ V, quindi: 36Vca.

Tutto questo però comporta: l'utilizzo di un adeguato contenitore metallico, montaggi meccanici, cablaggi elettrici, dissipatore termico e come risultato finale anche un discreto ingombro e non trascurabile peso di tutto l'apparato.

Considerando i vari aspetti, abbiamo deciso di approssciare un sistema che prevedesse l'utilizzo di un efficiente alimentatore switching industriale il cui costo non supera quello di un buon trasformatore.



SCHEMA B

Fig. 5
Schema alimentazione primaria con trasformatore a presa centrale e due diodi.

In **Tabella 4** riportiamo i dati dell'alimentatore scelto, distribuito da Futura Elettronica (cod. DELTA35048).

CONSIDERAZIONI SUI REGOLATORI SW BUCK CONVERTER

Risulta fondamentale considerare che gli alimentatori switching, producono diverse tipologie di ripple, o più comunemente in Italiano "ondulazione residua". La **Fig. 8** illustra idealmente lo schema di uno SW tipo Buck con le sorgenti di "rumore" e che fondamentalmente risultano essere di tre tipi:

▪ Ripple da commutazione

In base alla tipologia del regolatore a commutazione e al suo principio di funzionamento, l'ondulazione che ne deriva è quasi sempre il

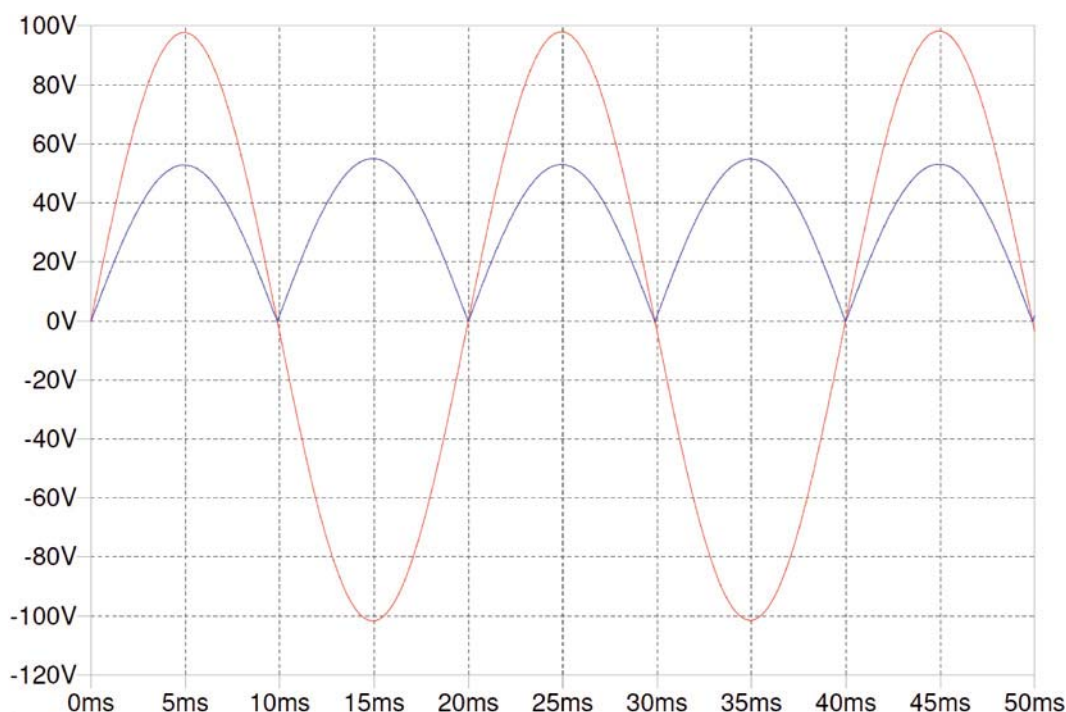
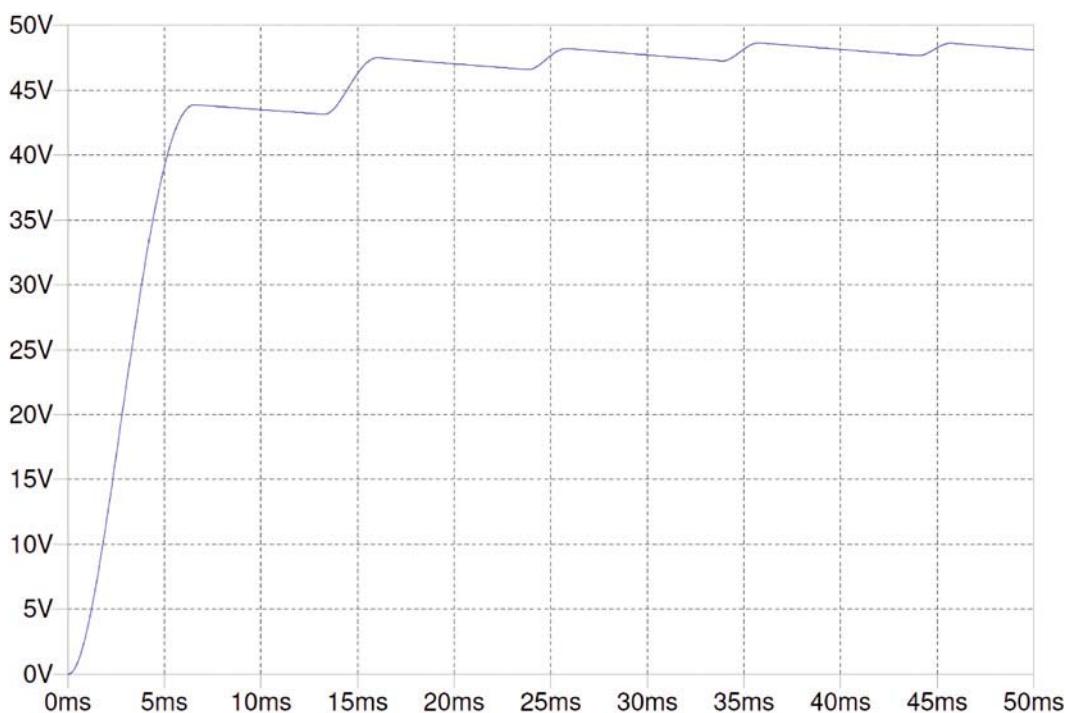


Fig. 6
Forma d'onda sinusoidale in ingresso e raddrizzata in uscita a doppia semi onda.

Fig. 7
Tensione di uscita
filtrata dai condensatori.



rumore principale in questo tipo di regolatore, dove l'ampiezza di tensione picco/picco, è generalmente compresa tra alcuni mV fino e qualche decina di mV e dovrebbe essere considerato un segnale periodico e prevedibile.

Può essere riconosciuto e misurato facilmente con un oscilloscopio, nel dominio del tempo o mediante la trasformata di Fourier nel dominio della frequenza; a patto che operi su una frequenza di commutazione fissa.

La **Fig. 8**, illustra schematicamente un tipico regolatore Buck. Poiché i due interruttori si accendono e si spengono alternativamente, la tensione del nodo SW, V_{SW} , è un'onda quadra ideale dipendente dal ciclo di lavoro, (duty-cycle) e dalla tensione di ingresso e dove V_{SW} può essere espressa dalle seguenti

equazioni:

$$V_{SW}(j\omega) = V_{IN} \sum_{k=-\infty}^{\infty} \frac{2\sin(k \times \pi \times D)}{k} \times \delta(\omega - k \times \omega_0)$$

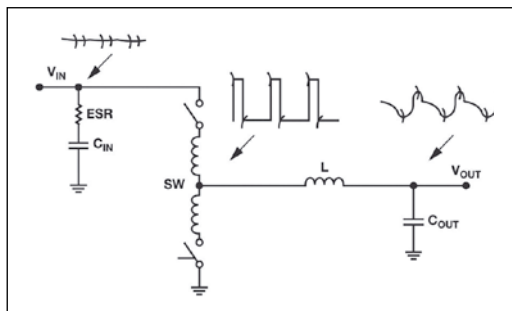
$$V_{SW}(t) = V_{IN} \sum_{k=-\infty}^{\infty} \frac{\sin(k \times \pi \times D)}{k \times \pi} \times e^{j\omega_0 \times k \times t} \quad k=0, \pm 1, \pm 2, \dots$$

dove: V_{IN} è la tensione di ingresso. D è il duty-cycle e che per i regolatori Buck, è uguale a V_{OUT}/V_{IN} . La componente fondamentale V_{SW} e le armoniche, quando viene fissato V_{IN} , dipendono unicamente dal ciclo di lavoro (duty-cycle). La **Fig. 9**, illustra l'ampiezza della componente fondamentale e relative armoniche V_{SW} in relazione al duty-cycle. La fondamentale domina l'ampiezza di ripple, quando il duty-cycle è circa a metà. A titolo conoscitivo La funzione di trasferimento dello stadio LC di uscita di un regolatore Buck è la seguente:

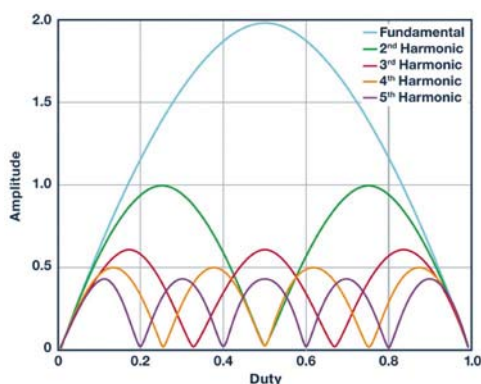
$$T_{LC}(j\omega) = \frac{ESR + j\omega \times ESL + \frac{1}{j\omega \times C_{OUT}}}{ESR + j\omega \times ESL + \frac{1}{j\omega \times C_{OUT}} + \frac{1}{j\omega \times C_L} (DCR + j\omega \times L)}$$

dove L è il valore dell'induttanza di uscita, DCR è il valore della resistenza dell'induttore e C_L è il

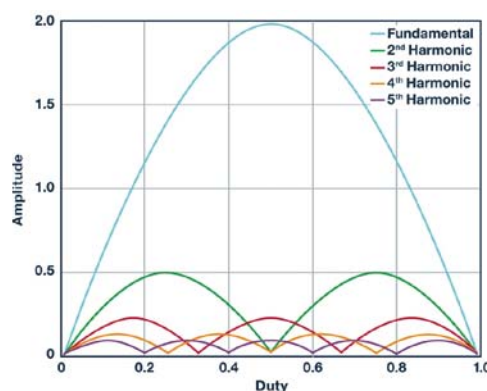
Fig. 8
Tipico schema di
regolatore buck
con picchi di alta
frequenza e rumore
d'uscita.



→ **Fig. 9**
Regolatore Buck -
Ampiezza della
VSW rispetto al
duty-cycle.



← **Fig. 10**
Regolatore
Buck –
Ampiezza
della VOUT
(dopo filtro
LC) rispetto
al duty-cycle.



valore di capacità parallela dell'induttore. C_{OUT} è il valore della capacità di uscita. ESL è l'induttanza serie del condensatore. ESR è il valore della resistenza serie del condensatore, j è l'unità immaginaria e ω è la pulsazione. Senza addentrarci nei dettagli delle leggi costitutive dei fasori e delle funzioni sinusoidali, ricordiamo soltanto che seguendo la legge di Ohm: $V=I \cdot R$; secondo le regole di derivazione abbiamo: $I=j\omega CV$ per il condensatore e $V=j\omega LI$ per l'induttore. Fatta questa debita parentesi cognitiva, tornando al nostro calcolo, otteniamo quindi che V_{OUT} può essere espresso come di seguito:

$$V_{OUT}(j\omega) = V_{SW}(j\omega) \times T_{LC}(j\omega)$$

Per semplificare il calcolo, assumiamo 20 dB per decade per lo stadio LC di uscita. il ripple V_{OUT} della fondamentale e relative armoniche in relazione al duty - cycle, sono rappresentate in **Fig. 10**. La terza armonica, o dispari, sarà maggiore di un'armonica pari quando il duty - cycle è vicino alla metà. Le armoniche più alte avranno un'ampiezza inferiore a causa della soppressione di LC e proporzioni piuttosto limitate rispetto all'ampiezza totale del ripple. Anche in questo caso, l'ampiezza fondamentale è comunque la componente principale nell'ondulazione di uscita del regolatore a commutazione.

Per il regolatore Buck, l'ampiezza della fondamentale, sarà quindi correlata alla tensione di ingresso, duty-cycle, frequenza di commutazione e stadio LC e tutti questi parametri avranno un impatto sui requisiti dell'applicazione, come l'efficienza e la dimensione della scheda del circuito. Per ridurre ulteriormente il ripple, è sempre consigliabile integrare un post-filtro aggiuntivo.

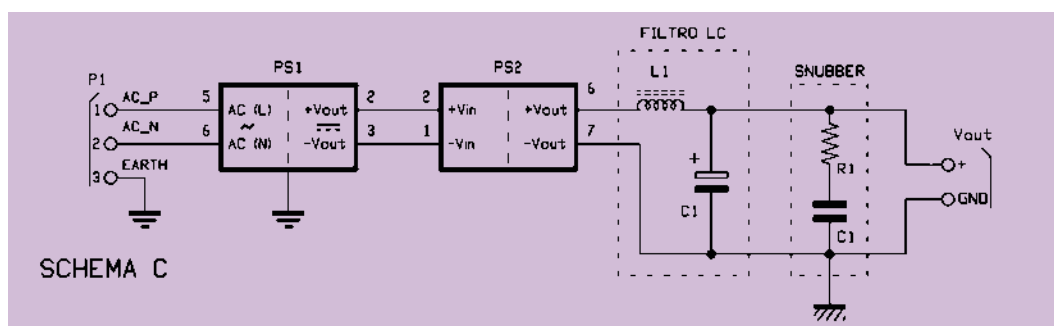
▪ Rumore a banda larga

Il rumore a banda larga in un regolatore a commutazione è un rumore di ampiezza casuale sulla tensione di uscita. Può essere rappresentato dalla densità del rumore in sovrافrequenza $V/\sqrt{Hz}$, o V_{rms} , che è parte integrante della densità all'interno dell'intervallo di frequenza. A causa delle caratteristiche dei semiconduttori e delle limitazioni di progettazione del filtro di riferimento, il rumore a banda larga, nei regolatori a commutazione, si trova generalmente nella gamma di frequenza da 10 Hz a 1 MHz e può risultare abbastanza difficile da ridurre tramite filtri aggiuntivi nelle gamme di bassa frequenza. La tipica tensione di ampiezza picco-picco del rumore a banda larga del regolatore buck è compresa tra circa 100 μV e 1mV, che è comunque molto inferiore al rumore di ripple di commutazione. Se si utilizza un efficiente filtro aggiuntivo per ridurre il rumore di ripple da commutazione, il rumore a banda larga potrebbe diventare il rumore principale nella tensione di uscita del regolatore stesso.

▪ Rumore di picco ad alta frequenza (High Frequency Spike and ringing noise)

Questo tipo di rumore è dovuto al picco ad alta frequenza e al cosiddetto "ringing noise", poiché la tensione di uscita viene generata dal transitorio di accensione o spegnimento dello switch. Considerando le induttanze e le capacità parassite nei circuiti / semiconduttori e nelle piste del circuito stampato (PCB); I veloci transitori di corrente causano picchi di tensione ad altissima frequenza e "ringing" al nodo SW. Il picco e il ringing aumenteranno proporzionalmente a carichi di corrente più elevati. In **Fig. 8** è illustrata teoricamente la forma che prende il picco per i regolatori Buck. A seconda della velocità di

Fig. 11
Schema
dell'alimentatore
da progetto.



accensione/spegnimento dello switch, il picco più alto risulta nell'intervallo tra 20 MHz a 300 MHz, quindi il filtro LC in uscita potrebbe non essere molto efficace alla soppressione a causa delle sue induttanze e capacità parassite. Rispetto ai vari tipi di ondulazione, il peggiore in assoluto è senza dubbio, questo tipo di rumore da radiazione che si sviluppa tra i nodi SW e VIN che avrà un impatto sulla tensione di uscita e su altri circuiti analogici a causa della sua frequenza molto elevata. Un metodo semplice e abbastanza efficace per ridurre questo tipo di disturbo, è l'utilizzo di uno "snubber" ovvero di un resistore con in serie un condensatore, posto tra positivo e negativo di uscita subito prima del filtro LC. Le formule di calcolo sono all'incirca queste: $Z_p = 2 \cdot \pi \cdot f_{Ring} \cdot L_p$; $R_{Snubber} \geq Z_p$; $C_{Snubber} = 1 / (2 \cdot \pi \cdot f_{Ring} \cdot C_{Snubber})$; dove Z_p è l'impedenza, f_{Ring} è la frequenza dei picchi e L_p sono le induttanze parassite. La difficoltà di calcolo sta proprio nel riuscire ad individuare le induttanze parassite. Di norma

questo accorgimento, dovrebbe comunque ridurre significativamente il rumore di picco sull'uscita. Inoltre, bisogna schermare o tenere lontane le sorgenti di rumore dal SW e dai nodi di ingresso dal lato di uscita e dal circuito analogico sensibile e schermare l'induttore di uscita. Un'attenta progettazione e posizionamento dei componenti sul layout è fondamentale. In terzo luogo, è necessario ottimizzare la velocità di accensione/spegnimento del regolatore di commutazione e ridurre al minimo l'induttanza parassita e la resistenza di commutazione al fine di ridurre efficacemente il rumore del nodo SW. Nel nostro alimentatore il valore più elevato è

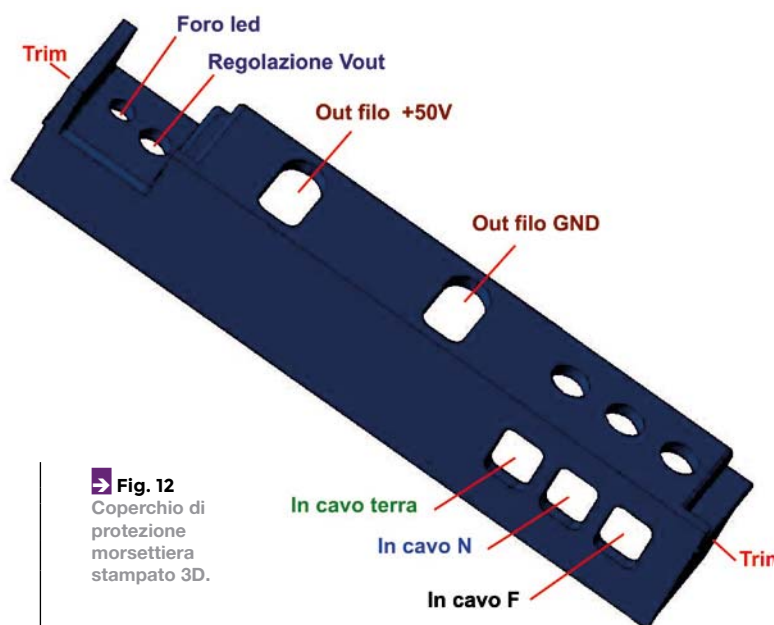


Fig. 12
Coperchio di
protezione
morsetteria
stampato 3D.

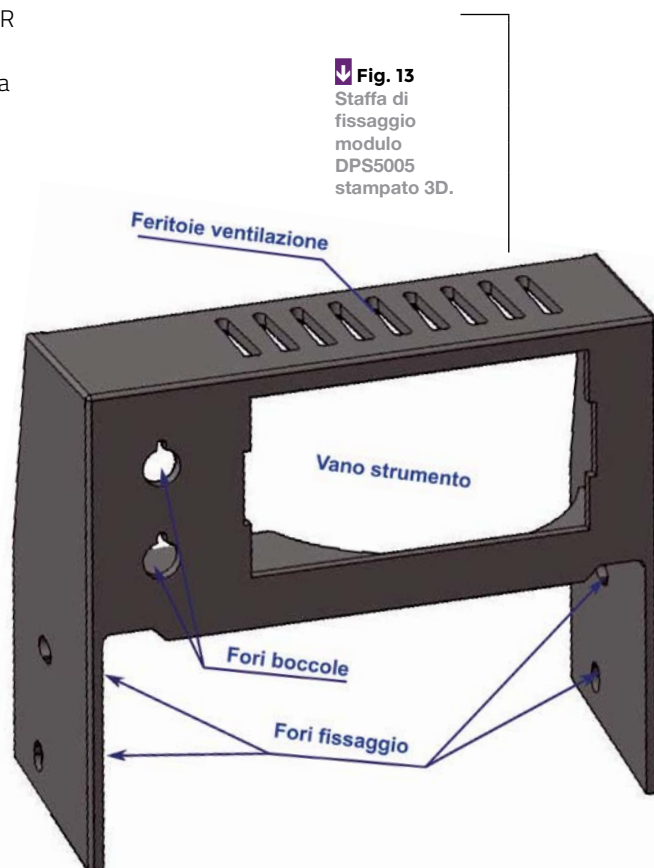
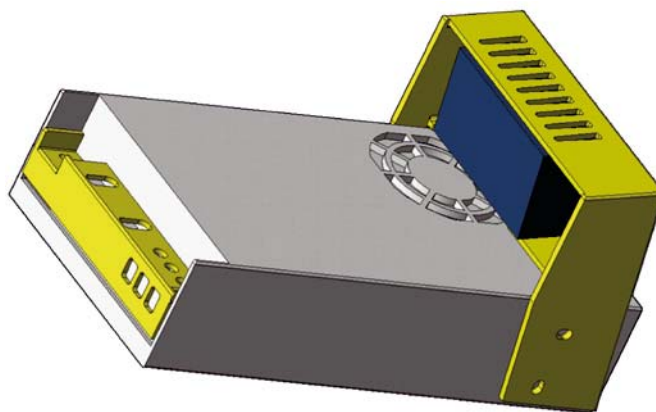
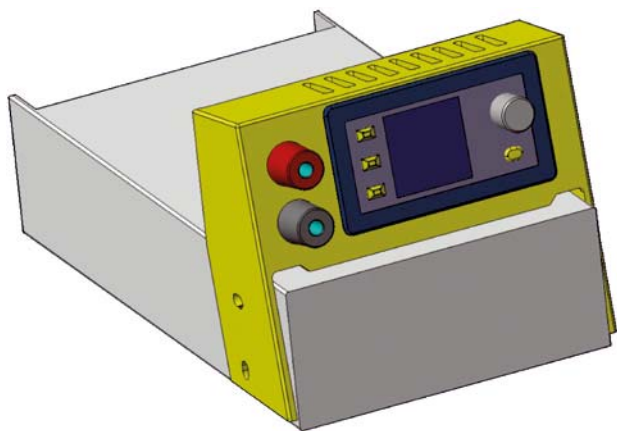


Fig. 13
Staffa di
fissaggio
modulo
DPS5005
stampato 3D.



proprio dovuto ai picchi di alta frequenza e che purtroppo si portano avanti fino all'uscita del modulo DPS5055. Dopo un po' di lavoro e di calcoli, siamo riusciti a ridurre il ripple totale di circa 10 volte portandolo a circa 25-30 mVpp – 5-6 mVrms, grazie soprattutto al filtro snubber che ha ridotto il rumore dovuto ai picchi HF, intorno a 36-37 MHz, di oltre 28 dB (27 volte in tensione) portandoli a valori di soli 10 mVpp; posizionato a valle del DPS5005 e che sarebbe stato doveroso sistemare all'interno del modulo subito all'uscita del connettore, ma poiché questo risulta alquanto difficile, lo abbiamo realizzato subito prima dell'uscita sulle bocche di alimentazione, su apposita basetta, non dopo aver affinato sperimentalmente i valori di R/C, in modo da compensare un po' induttanze e capacità parassite. Considerando costi / qualità e i risultati ottenuti abbiamo quindi elaborato un semplice progetto (**Fig. 11**), utilizzabile per alimentare la maggior parte dei progetti hobbistici o uso di laboratorio. Questo si compone appunto, di un modulo: DELTA35048, di un modulo: DPS 5005, di un filtro di uscita LC + rete snubber RC.

IL PROGETTO

Analizzando i componenti possiamo vedere che entrambi i moduli, soprattutto l'alimentatore primario risulta in versione "retail" ovvero (nudo) privo di protezioni e adatto quindi in ambiente industriale all'interno di un armadio.

A tal proposito abbiamo innanzitutto pensato alla sicurezza ed abbiamo realizzato un coperchio di protezione in materiale plastico stampato in 3D (**Fig. 12**). In secondo luogo abbiamo immaginato qualcosa di minimo indispensabile per il fissaggio del modulo DPS 5005 e per le bocche di uscita + e -. Abbiamo quindi progettato una apposita staffa (**Fig. 13**), da fissare nei fori laterali M4 disponibili

sull'alimentatore switching. Questa staffa è stata volutamente posizionata dal lato dove è presente la ventola di raffreddamento del modulo base, in modo tale che l'aria possa raggiungere e migliorare il raffreddamento del dissipatore del modulo DPS 5005. Il risultato finale del progetto è illustrato nella **Fig. 14**.

Filtro di uscita

Il filtro LC di uscita, è realizzato avvolgendo 3 spire di filo da 1 / 1.5 mm² con guaina isolante dentro ad un nucleo toroidale in ferrite per alta frequenza di Ø esterno 16 mm circa e Ø interno 6 mm circa con spessore 5.5 / 6 mm. In questo modo si otterrà una impedenza di circa 20 / 40µH. A questo punto dobbiamo realizzare una semplice basetta come da **Fig. 15**, dove montare la resistenza R1 e i condensatori C1 e C2, da assemblare successivamente sulle bocche di uscita. Realizzeremo la stessa partendo da un pezzo di vetronite con il rame su una sola faccia e taglieremo un rettangolo di circa 20 x 42 mm. A distanza di 20 mm (10 per parte dal centro) praticheremo due fori di diametro 6.5 mm. Poi con l'aiuto di un cutter e di un righello, realizzeremo due incisioni ad una larghezza di circa 5

Fig. 14
Staffa fissata
all'alimentatore
PMT-48V350W.

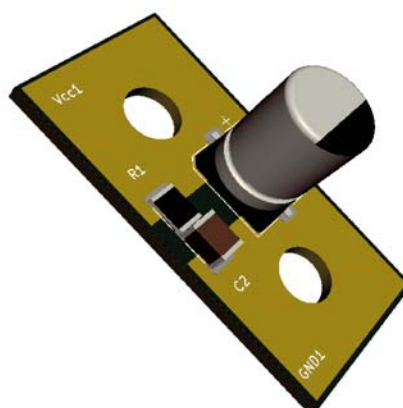
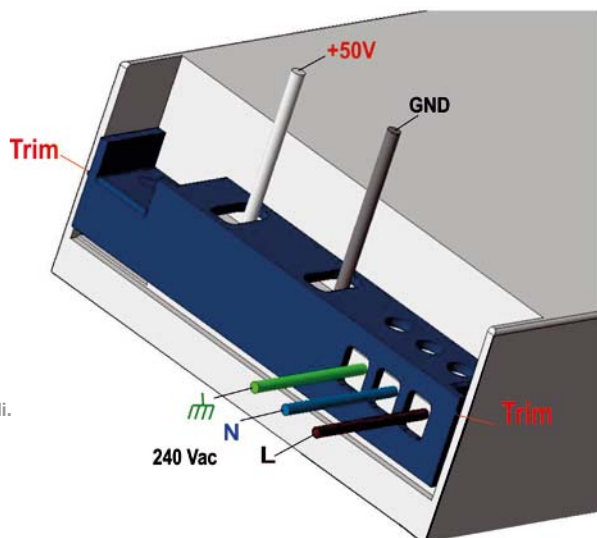


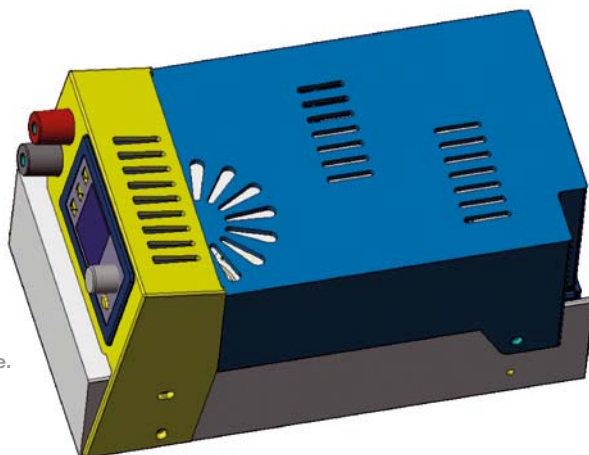
Fig. 15
Basetta filtro
LC + snubber.

Fig. 16
Assemblaggio
coperchio
di protezione
e passaggio fili.



mm centrato alla basetta stessa. Andremo quindi sempre aiutandoci con un cutter, ad asportare la strisciolina centrale creando così due piazzole una per il polo positivo e l'altra per il polo negativo. Per il condensatore C1, potremo utilizzare un comune condensatore verticale ripiegando i terminali a 90° sotto la base ed accorciando gli stessi in modo che fuoriescano dal corpo per 4-5 mm massimo da ogni lato. Quindi lo posizioniamo come da **Fig. 15** rispettando la polarità e lo saldiamo da entrambi i lati. Andiamo quindi a saldare la resistenza R1 che dovrà essere SMD tipo 4.5x3 oppure 3.2x1.6, avendo cura che una estremità sia sul lato positivo della basetta e l'altra sia esattamente a metà della parte senza rame. Ovviamente salderemo il solo lato rame. Ugualmente per il condensatore C2 che avrà le stesse dimensioni e che andrà saldato dal solo lato massa, avendo cura che l'altro lato sia in contatto con la resistenza. Questo condensatore dovrà essere ceramico e cosa importante di classe X7R. Alla fine salderemo tra loro i due componenti con una goccia di stagno.

Fig. 17
Alimentatore
montaggio finale.



Assemblaggio

Inizieremo con il montaggio del modulo DPS 5005. Per fare ciò dovremo adattare la sede della staffa realizzata mediante stampa 3D eliminando le eventuali imperfezioni e adattando bene la sede alle dimensioni del modulo mediante l'utilizzo di una lima piatta di circa 2x9 mm. Lo stesso faremo per le boccole di uscita rossa (+) e nera (-) a seconda del tipo utilizzato, in questo caso tramite l'ausilio di un trapano e di una punta di Ø appropriato.

Al termine, prima del montaggio finale, se lo desideriamo possiamo carteggiare bene tutte le superfici esterne della staffa, poi spruzzare una mano di primer per plastica ed infine verniciare il tutto con il colore desiderato. Successivamente montiamo le due boccole e poi saldiamo i due fili di uscita (nero per la massa e rosso (quello avvolto nel toroide) ai terminali - e + della basetta.

Posizioniamo quindi la stessa all'interno della staffa nei "tubetti filettati" delle boccole e tramite due rondelle elastiche dentate tipo "grower" e dadi dedicati la fissiamo. Al termine, montiamo la staffa sul modulo primario tramite 4 viti da M4x8. Prestare attenzione a non montare viti più lunghe che potrebbero creare CC all'interno. Successivamente attorcigliamo i fili di uscita (rosso e nero) in modo da ridurre la possibilità di captare disturbi di alternata e dopo averli accorciati al minimo indispensabile li fissiamo al connettore di uscita, badando bene a rispettarne la polarità.

COPERCHIO CONNETTORE

A questo punto possiamo inserire il modulo nella propria sede in modo definitivo e passare alla fase successiva. La morsettiera dell'alimentatore è completamente scoperta e quindi pericolosa e pertanto andremo ad adattare e montare il coperchio appositamente realizzato. A tal proposito, riferendoci alla **Fig. 16**, andremo ad adattare il coperchio in modo che si incastri saldamente tra le due pareti laterali del case in alluminio.

Come potrete notare sui lati del coperchio sono presenti due nervature (TRIM) che possono essere facilmente limate fino ad adattare perfettamente il coperchio. Successivamente inseriremo un filo da 1 - 1.5 mm² di sezione nel II° polo (V+) da sinistra ed un secondo filo nel V° polo sempre da sinistra (V-). Dopo averli fissati bene, li tireremo verso l'alto e dopo averli inseriti nelle apposite feritoie del coperchio, monteremo lo stesso in posizione ferma e stabile. Al termine inseriremo il cordone di alimentazione rispettando: terra /neutro e fase e chiuderemo bene le viti. Questo sistema assicura

la corretta protezione della morsettiera perché il coperchio non potrà essere tolto se non prima di scollegare il cordone di alimentazione stesso. Volendo aumentare la tenuta si potranno applicare due punti di colla a caldo sulle due estremità interne adiacenti alle pareti in alluminio. Non rimane che fissare alla morsettiera del modulo i due fili di uscita avendo cura di non tenerli eccessivamente lunghi e attorcigliandoli prima tra di loro e rispettando la corretta polarità.

TARATURA

L'operazione di taratura è estremamente semplice. Controllato bene che tutto sia a posto e non vi siano errori di cablaggio, possiamo accendere il nostro alimentatore. Dopo un brevissimo messaggio durante il quale il modulo esegue il set-up, il display si accende presentando una schermata come da Fig. 2. Ora prendiamo un cacciavite, possibilmente in plastica ed agendo sul trimmer posizionato sul retro dell'alimentatore, in corrispondenza del foro del coperchio, lo ruotiamo in modo da ottenere una tensione di circa 50 Vcc che potremo leggere nell'ultima riga del display (rif.5 Fig. 2). Con questa tensione, potremo prelevare in uscita dal modulo DPS 5005 fino a 47 V 5 A. Per salire fino a 48V si dovrà aumentare la tensione del modulo primario fino a 51 – 52 V max. Se pensiamo di non dover utilizzare mai tensioni così elevate, potremmo tarare l'uscita su 40 Vcc e prelevare fino a 36 Vcc max 5 A, senza gravare troppo sul modulo di uscita.

CONCLUSIONI

Al termine se lo desideriamo, per rendere la realizzazione più professionale, possiamo realizzare un coperchio plastico sempre con stampa 3D ed assemblarlo. Per prima cosa allentiamo le 4 viti di fissaggio della staffa e ruotiamo la stessa tutta verso l'esterno sfruttando i giochi dei fori di fissaggio appositamente realizzati per tale scopo. Facciamo poi scorrere il coperchio dall'esterno verso l'interno, in modo che la parte frontale si inserisca sotto alla staffa dello strumento per circa 5 mm, verificando che i fili di collegamento passino all'interno della apposita cavità e che i fori di fissaggio siano centrati alle filettature posteriori del case di alluminio, fissandolo successivamente con altre due viti da M4x8. Quindi ruotiamo la staffa verso l'interno in modo che vada in appoggio sul coperchio stesso, chiudendo infine le 4 viti di fissaggio della stessa.

A questo punto il tutto risulterà saldamente bloccato e stabile. Il risultato è visibile in Fig. 17. □

Diciamo subito che le misurazioni effettuate sugli alimentatori switching non sono facili da eseguire e possono essere oggetto di possibili errori. Se qualcuno si volesse cimentare, dovrà tenere in debita considerazione quanto segue:

- I cavi di uscita dall'alimentatore devono essere di sezione adeguata e di lunghezza limitata al necessario. La sezione e la lunghezza possono introdurre cadute di tensione anche di qualche centinaio di millivolt. Inoltre è sempre bene attorcigliarli tra di loro affinché captino meno rumore possibile. Ricordiamoci sempre che viviamo immersi nei campi elettromagnetici e che ogni filo, ogni pista del CS è una potenziale antenna.
- Stesso discorso vale per le connessioni, perché se allentate, saldate male, ossidate ecc., creano rumore e caduta di tensione.
- Le misurazioni del ripple, vanno fatte con un oscilloscopio a larga banda, dove in modo normale si visualizza la forma d'onda principale e poi "zoom-mado" sulla posizione centrale del picco, avremo la possibilità di analizzare gli spikes e la frequenza di "ringing"
- Utilizzare una sonda per oscilloscopio di qualità e posizionarla sempre su ingresso 1X. La sonda a 10X introduce notevole rumore e in alcuni casi può addirittura amplificare i picchi

HF. Pertanto è bene evitare. Ovviamente anche lo strumento dovrà essere calibrato in ingresso a 1X. Togliere il filo di massa con la clip e il cappuccio con la sonda a gancio. Intorno al cilindretto metallico di massa del puntale, avvolgere alcune spire di filo di rame stagnato di sezione 0.5 - 0.7 mm (Fig. A) e se necessario, lo possiamo fissare con un giro di nastro isolante. Controllare la calibrazione della sonda e relativo rumore mediante l'uscita di calibrazione onda quadra a 5V.

Se necessario ritoccare la calibrazione della sonda fino ad ottenere una perfetta onda quadra priva di deformazioni o increspature. A questo punto si può procedere con le misurazioni, avendo cura di effettuare le stesse il più possibile a ridosso dell'uscita del filtro, poiché più lontano si va e più rumore si introduce. Solo in questo modo misureremo effettivamente il rumore in uscita dall'alimentatore e non quello dei cavi e altre possibili fonti di disturbo.



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. Il modulo alimentatore step-down da pannello (cod. JT-DPS5005) è in vendita a Euro 54,00, l'alimentatore 48V (cod. DELTA35048) è disponibile a Euro 58,00. 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>

Where are you?

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



LOCALIZZATORE GPS/GSM/GPRS CON ASCOLTO AMBIENTALE

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



cod. FR763

€ 74,00



MICRO LOCALIZZATORE SATELLITARE GPRS/GSM

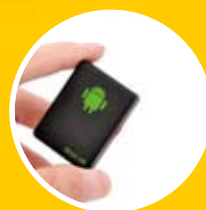
Tutto sotto controllo con il nuovo micro localizzatore GPRS/GSM. Per attivarlo basta inserire una SIM di qualsiasi operatore; inviando un SMS potrai attivare l'ascolto ambientale oppure ricevere un link con le coordinate che indicano la posizione in tempo reale del localizzatore. Dispone di batteria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

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



cod. A8TRACK

€ 14,00



LOCALIZZATORE TASCABILE GPS/GSM/GPRS

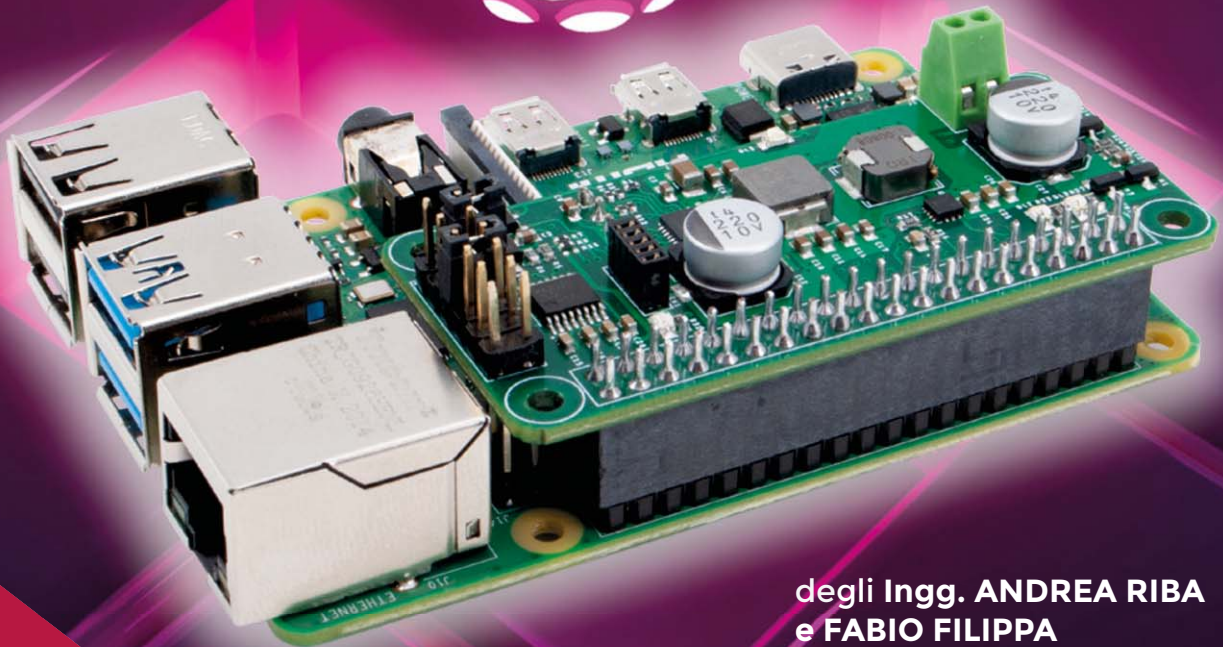
Ideale per la localizzazione personale e veicolare. Può inviare le proprie coordinate (latitudine e longitudine) via SMS verso telefoni cellulari, oppure, tramite tecnologia GPRS, ad un computer opportunamente configurato. Dispone di microfono integrato e batteria ricaricabile.



cod. TK102

€ 54,00

MINIUPS: CONTINUITÀ PER IL LAMPONE



degli Ingg. **ANDREA RIBA**
e **FABIO FILIPPA**

Soluzione per dotare il Raspberry Pi di funzionalità UPS con la possibilità di configurare il sistema in funzione della batteria LiPo utilizzata come riserva di backup. Prima puntata.

Sappiamo quanto importante sia assicurare il funzionamento continuativo di schede elettroniche Single Board Computer come Raspberry Pi, soprattutto per evitare perdite di dati o danni alle unità di storage; per questo esistono dispositivi chiamati UPS, realizzati anche in versione "hat" per le nostre

Raspberry Pi. Un esempio lo trovate in queste pagine, dove proponiamo una scheda che funge da gruppo di continuità, garantendo l'alimentazione grazie a una batteria di backup LiPo che mantiene in carica, nonché a chiudere i processi in esecuzione per spegnere la Raspberry Pi laddove l'autonomia fornita dalla batteria sia in esaurimento.

Il nostro MiniUPS si presta alla realizzazione di sistemi di alimentazione per sistemi elettronici che richiedano un backup energetico per prevenire la perdita di dati come, la Raspberry Pi (in tutti i suoi allestimenti e versioni) oppure Arduino o più

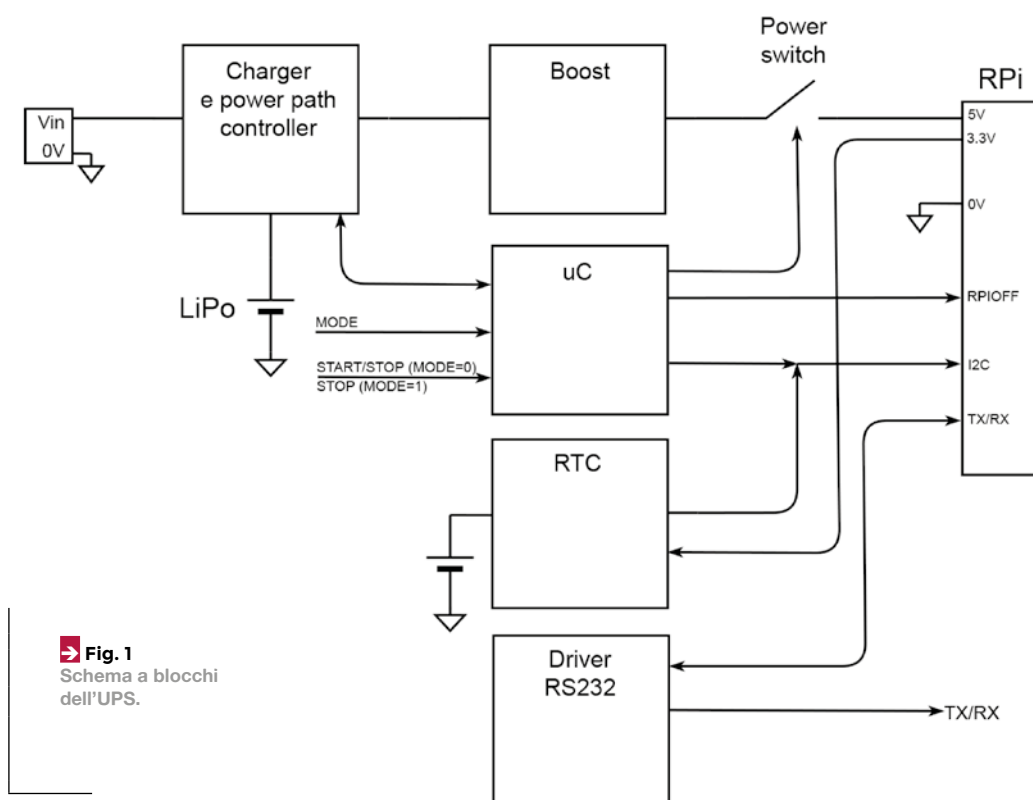


Fig. 1
Schema a blocchi
dell'UPS.

in generale per qualsiasi sistema elettronico per il quale sia richiesto un funzionamento affidabile h24. La tensione di alimentazione principale è compresa tra 3,9 e 14 Vcc; come tensione nominale sono consigliati $5V \pm 5\%$ mentre la corrente che l'alimentatore deve fornire dev'essere almeno 3A, fornibile su un apposito morsetto a vite presente sulla scheda. Nel caso in cui l'alimentatore non fosse in grado di erogare tutta la corrente necessaria, la scheda, entro certi limiti, è comunque in grado di compensare la riduzione di tensione diminuendo l'assorbimento di corrente in ingresso.

Il processo di carica della batteria può portare a notevole generazione di calore, il sistema si autoregola gestendo la corrente di carica in modo da mantenere la temperatura all'interno del campo operativo dello stadio di carica. La tensione di batteria è poi innalzata al livello della tensione di alimentazione del Raspberry Pi mediante uno stadio switching DC-DC in topologia boost che garantisce in uscita una tensione stabilizzata nominale di 5V circa. Il MiniUps è quindi una scheda di ricarica rapida intelligente per batterie LiPo a singola cella (1s) da 3,7V con corrente di ricarica massima da 3 A. Realizzata con un circuito stampato multistrato (4 strati) è dotata di un ingresso di accensione/spegnimento attivo basso ed è in grado di notificare

al mondo esterno l'imminente spegnimento del carico. In aggiunta la scheda può essere interrogata attraverso bus I²C per conoscere i parametri di funzionamento del sistema (presenza o assenza della tensione principale, tensione della batteria, tensione di ingresso, allarmi, eccetera), inoltre permette anche di salvare dati utente su una memoria EEPROM integrata nella scheda stessa ed attraverso alcuni diodi led fornisce una rappresentazione visuale dello stato di funzionamento del sistema. Oltre alle dotazioni legate alle funzionalità come UPS la scheda integra un modulo RTC (DS3231S) e relativa batteria (CR1025) connesso al bus I²C della Raspberry Pi ed un driver RS232 (MAX3232) connesso direttamente alle linee RX-TX della Raspberry Pi. Lo schema a blocchi di **Fig. 1** dettaglia la configurazione hardware della scheda UPS.

Nelle Raspberry Pi, a livello di risorse GPIO la scheda occupa una sola linea dedicata (pin 11, relativo al GPIO17) identificata con RPIOFF, in serie alla quale è stato posto un resistore di limitazione della corrente da 1k Ω , utilizzata per avvisare la Raspberry Pi o un generico carico che il sistema a breve verrà spento (entro 60s). Tale linea consente di salvare i dati delle applicazioni e permette uno spegnimento pulito del sistema, senza perdita di dati, eliminando il rischio di corrompere il contenuto della SD-Card

conseguente ad un'interruzione improvvisa della tensione di alimentazione. Per intercettare lo stato di questa linea è possibile installare un servizio che avverta il sistema di questo evento oppure controllarla in polling direttamente all'interno dell'applicazione utente, a corredo, per questo scopo, è fornito uno script Python pronto per essere utilizzato. Oltre a questo si hanno numerosi esempi di utilizzo del bus I²C e della porta seriale attraverso altrettanti script sempre in Python direttamente scaricabili dal repository GitHub accessibile da <https://github.com/Italsensor/RPI-MINI-UPS-RO>. Tutti gli ingressi e le uscite della scheda possono essere utilizzati saldando direttamente sulle piazzole oppure montando dei connettori passo 2,54 mm.

Nella **Tabella 1** sono riportate le caratteristiche della scheda e nella **Tabella 2** quelle della batteria LiPo in dotazione (quella di serie).

A titolo puramente indicativo, in riferimento a delle prove effettuate con una Raspberry Pi3B+ Rev 1.3, sistema operativo Raspbian 10 (buster), desktop grafico, browser aperto su due pagine di YouTube con filmati in esecuzione, LibreOffice con applicazione Writer aperta, gioco precaricato Soccer in esecuzione, mouse e tastiera USB connesse, monitor HDMI connesso e partendo da una condizione di batteria LiPo carica, l'autonomia complessiva risulta mediamente di 10 minuti (la tensione della batteria è pari a circa 3,6 V).

Con la modalità UPS, al ripristino della tensione di alimentazione, il sistema riparte già dopo circa

Tensione di uscita (alimentazione carico)	5,1V
Tensione di ingresso	3,9V ÷ 14V
Massima corrente assorbita	3A
Massima corrente in uscita	3A
Massima corrente di carica batteria	3A
Efficienza circuito di ricarica	>90%
Temperatura operativa	0° ÷ +70°C
Protezione sovratemperatura batteria	-
Corrente di carica impostata per la batteria predefinita	380mA
Modulo RTC, indirizzo I ² C	0x68
Batteria modulo RTC	CR1025 Li/MnO2
Indirizzo I ² C scheda	0x08

Tabella 1
Caratteristiche
tecniche.

Capacità nominale	420mAh
Tensione di batteria nominale	3.7V
Tensione batteria di fine carica	4.2V
Corrente di carica massima	420mA (1C)
Corrente di scarica massima	6300 mA (15C)
Autonomia alla corrente di scarica massima	≥3.8' @ 15C
Sensore integrato di temperatura (NTC)	presente
Protezione tensione massima di carica (overcharge)	4.28 ± 0.025V
Protezione sovracorrente di carica / scarica	450mA / 6300mA
Protezione corto circuito	presente
Protezione contro scarica profonda	3.0V±0.1V
Connessione	Connettore JST (S3B-XH-A)

Tabella 2
Caratteristiche
della batteria
LiPo di serie.



Fig. 2
Connettore per la
batteria.

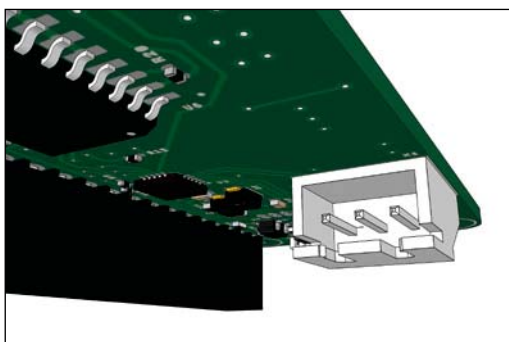


Fig. 3
La batteria LiPo.



1 minuto e 15"; la batteria, però, non è completamente carica, ma la tensione ai suoi capi, in presenza della tensione di alimentazione principale, è tale da permettere il superamento della soglia di accensione pari a circa 3,9 V.

Il connettore X1 della batteria (JST modello S3B-XH-A) si trova sul lato saldature della scheda, come proposto nella **Fig. 2**.

La relativa piedinatura è la seguente:

- 1 = +VBAT (positivo batteria);
- 2 = NTC (termistore NTC);
- 3 = 0V (negativo batteria).

La batteria proposta adatta al progetto è quella visibile nella **Fig. 3**: come vedete, dispone di un cavetto terminante con un connettore adatto a quello JST della scheda.

La piedinatura del connettore è corrispondente a quella del JST della scheda e i colori del cavetto sono: rosso per il positivo, nero per il negativo e bianco per il termistore che consente di monitorare la temperatura della cella LiPo.

Per quanto riguarda l'ingresso di alimentazione, il relativo connettore è una morsettiera a passo 5 mm (**Fig. 4**) della quale il contatto 1 è la massa (negativo) e il 2 il polo positivo.

La **Fig. 5** propone una panoramica delle linee di I/O e delle segnalazioni della nostra scheda UPS.

MODALITÀ DI FUNZIONAMENTO

Il sistema può operare con due modalità distinte:

- automatica (UPS mode);
- manuale (desktop mode); quest'ultima è l'impostazione predefinita.

La scelta relativa alla modalità di funzionamento, da effettuarsi sempre a carico spento o non collegato, è definita dall'impostazione del livello di tensione presente sul pin 6 (MODE) del connettore utente J2; se lasciato aperto il sistema utilizza la modalità automatica, invece cortocircuitando il pin 6 con il pin 5 (0V) il sistema utilizza la modalità manuale. J2 è un header maschio da due file con 6 vie ciascuna passo 2,54 mm, la cui piedinatura è proposta nella **Tabella 3**; è quindi possibile utilizzare un jumper passo 2,54 mm per cortocircuitare tra loro i due pin di configurazione come in **Fig. 6**.

La modifica della modalità di funzionamento deve essere effettuata sempre a carico non alimentato (led DL3 spento), si procede quindi con l'impostazione della modalità desiderata e si effettua il reset della scheda rimuovendo il ponticello tra i pin 1 e 2 e reinserendolo dopo alcuni secondi.

In alternativa la modalità si può modificare "on the fly" anche con sistema attivo, ma solo durante la fase di shutdown (durata 60 s), al termine dello shutdown il carico viene spento ed il sistema entra automaticamente nella modalità scelta.

La modalità UPS (accensione e spegnimento automatico in base alla tensione di alimentazione) si

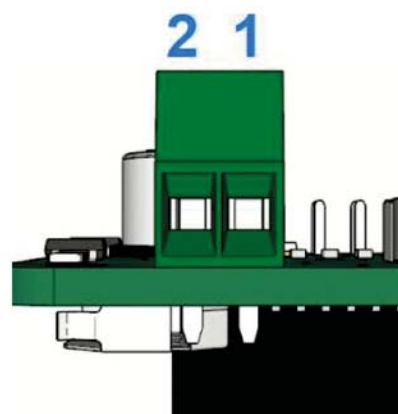


Fig. 4
Morsettiera di ingresso alimentazione della
scheda UPS.

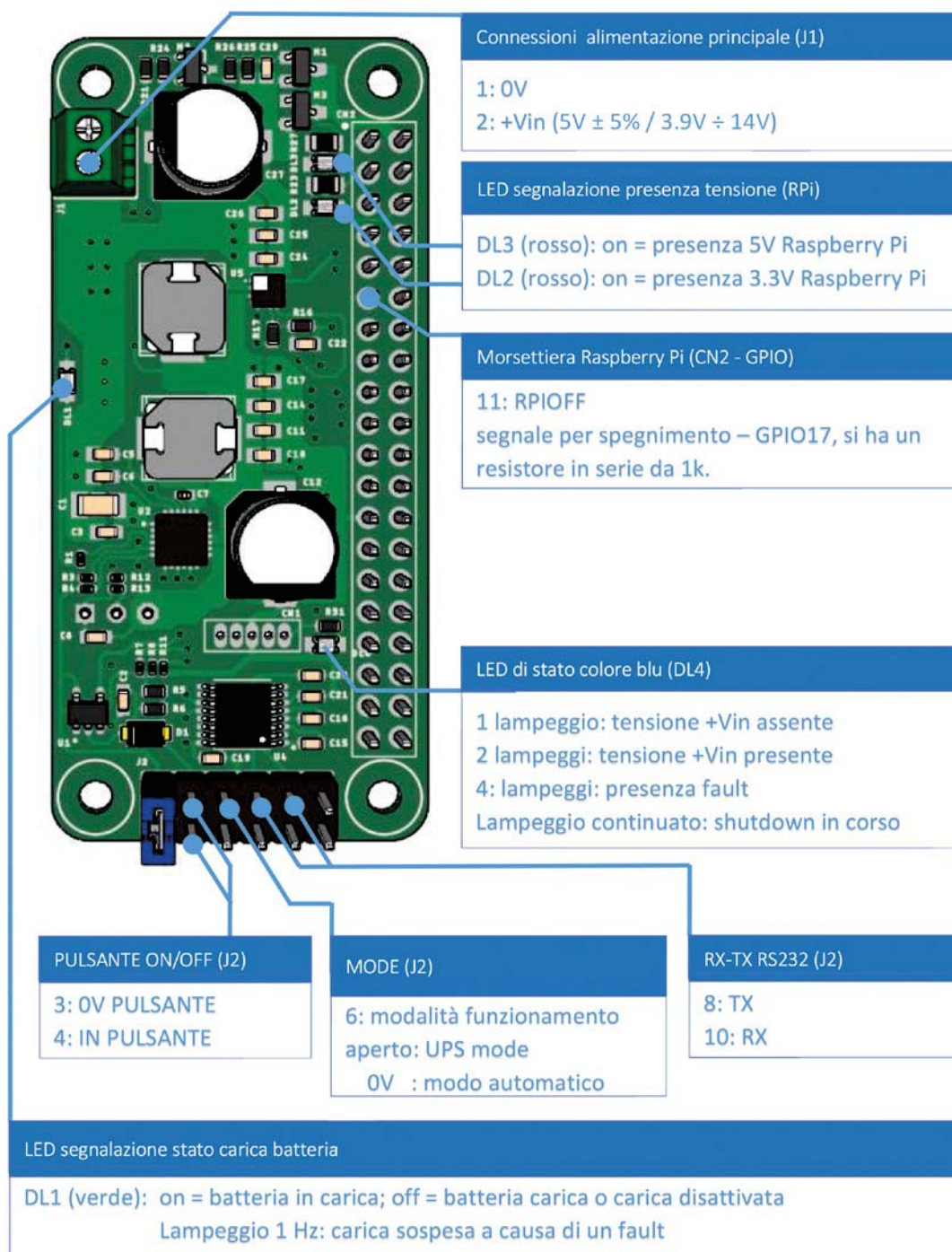


Fig. 5
Ingressi, uscite
e interfacce
della Mini UPS.

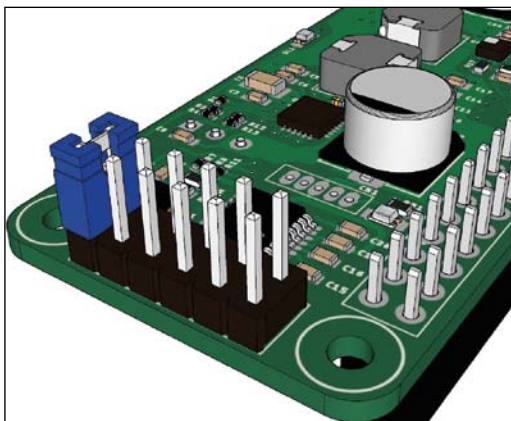
imposta lasciando aperto il pin 6 del connettore J2. Permette di gestire in modo automatico la linea di alimentazione del carico proprio come accade con un UPS tradizionale: in presenza della tensione di alimentazione principale, la scheda attiva automaticamente il processo di ricarica della batteria e quando la tensione di quest'ultima raggiunge i 3,9V

provvede ad alimentare il carico. Qualora la tensione principale venga a mancare il sistema entra nella modalità di backup, se la tensione di batteria diventa minore o uguale a 3,6V si ha l'attivazione della linea di shutdown RPIOFF (attiva alta), trascorsi ulteriori 60 s il sistema procede con la disconnessione del carico. Il carico verrà nuovamente alimen-

Tabella 3
Connettore utente
J2.

PIN	SEGNALE	FUNZIONE
1	VDD	Tensione 5V generale
2	VDDUP	Tensione sezione uP (5V)
3	0V	Connesso allo 0V
4	INBTN	Ingresso pulsante NO
5	0V	Connesso allo 0V
6	MODE	Scelta modo funzionamento
7	0V	Connesso allo 0V
8	TX	TX RPi
9	0V	Connesso allo 0V
10	RX	RX RPi
11	0V	Connesso allo 0V
12	-	Non collegato

Fig. 6
Connettore
J2 - Jumper
JP1 per reset
microcontrollore.



tato solo al ritorno dell'alimentazione principale e con una tensione della batteria di backup maggiore o uguale a 3,9V. È possibile interrompere questa modalità di funzionamento cortocircuitando il pin 4 con il pin 3 (0V) del jumper J2; in questo modo si avvia la sequenza di spegnimento ed è possibile -nel relativo intervallo di tempo- mantenere aperto il pin 6 del connettore J2 (si ritorna automaticamente alla modalità automatica) oppure cortocircuitarlo con il pin 7 (0V) per attivare, trascorsi i 60 s dello shutdown, la modalità di funzionamento manuale.

MODALITÀ MANUALE

Accensione e spegnimento attraverso pulsante esterno si impostano inserendo un jumper passo 2,54 mm nella terza posizione del connettore J2 (il pad 5 ed il pad 6 sono cortocircuitati) secondo quanto mostrato nella **Fig. 7**.

Questa modalità permette di accendere e spegnere la Raspberry Pi, oppure una generica scheda embedded o più in generale un carico generico, in modo controllato dall'utente, attraverso un pulsante esterno normalmente aperto oppure un contatto pulito tra il pin 4 ed il pin 3 (0V) del connettore J2. Il microcontrollore presente sulla scheda sovraintende le operazioni di accensione e spegnimento e controlla costantemente il valore della tensione di batteria. Il diodo LED DL3 di colore rosso si accende segnalando la presenza della tensione di alimentazione sui pin 2 e pin 4 del connettore CN2. Nel caso in cui il carico sia costituito da una Raspberry Pi dovrà accendersi anche il diodo LED DL2 indicando la presenza della tensione 3,3V generata dalla Raspberry Pi stessa.

L'accensione del carico si effettua connettendo tra loro il pin 4 ed il pin 3 del connettore J2 per un intervallo di tempo di almeno 3 s (la tensione di batteria deve comunque essere maggiore o uguale a 3,9V).

Lo spegnimento si effettua in modo analogo connettendo tra loro il pin 4 ed il pin 3 del connettore J2 per un intervallo di tempo di almeno 5 s. Non appena la scheda entra nella fase di spegnimento viene attivata la linea RPIOFF che può essere utilizzata, ad esempio nel caso in cui il sistema alimenti una scheda Raspberry Pi, per avviare il processo di shutdown evitando così una possibile perdita di dati legata ad una interruzione improvvisa dell'alimentazione, trascorsi 60s la scheda interrompe l'alimentazione al carico rimanendo in stand-by. Se la tensione di batteria dovesse diventare inferiore a 3,6V, indipendentemente dallo stato del pulsante di accensione/spegnimento, la scheda avvia automaticamente la procedura di shutdown attivando la linea RPIOFF; in seguito attende 60s ed al termine provvede a sezionare l'alimentazione del carico, infine il segnale RPIOFF è riportato al livello basso ed il sistema ritorna nello stato di attesa fino a quando la tensione di batteria non risulta maggiore di 3,9V, da questo momento in poi è possibile riaccendere il carico utilizzando il pulsante. In entrambe le modalità di funzionamento la batteria LiPo è sempre al sicuro, il sistema disattiva automaticamente il carico quando la tensione risulta inferiore a 3,6 V avviando la procedura di spegnimento proprio come se questa fosse stata richiesta manualmente dalla linea di ingresso di accensione/spegnimento, in questo modo il carico, se necessario, può effettuare uno shutdown in sicurezza, senza perdita di dati.

Il modulo è dotato di RTC integrato e relativa bat-

teria tampone (CR1025), la batteria si installa sul lato saldature del circuito stampato nell'apposito alloggiamento, come visibile in **Fig. 8**.

È importante inserire la batteria in modo tale che il polo negativo sia a contatto con il circuito stampato ed il polo positivo a contatto con l'alloggiamento metallico, inserire la batteria a rovescio può danneggiare irrimediabilmente il modulo RTC.

PREPARAZIONE E CONFIGURAZIONE DELLA SCHEDA

Il passo successivo prevede la configurazione di alcune caratteristiche della Raspberry Pi e l'installazione di tool specifici che potranno essere utilizzati per la lettura del modulo RTC; ecco la procedura.

- 1) A Raspberry Pi spenta, se non si dispone di una connessione di rete, collegare un monitor HDMI ed una tastiera su una porta USB libera. Se sulla Raspberry Pi è attivo il servizio SSH ed è disponibile una connessione di rete, ci si potrà connettere ad essa da un qualsiasi PC dotato di terminale PuTTY, mediante protocollo SSH; in questo caso non è quindi necessario collegare direttamente alla Raspberry Pi monitor e tastiera. Si inserisce quindi, nell'apposita sede, una microSDHC con il sistema operativo desiderato precaricato.
- 2) Verificare sulla scheda UPS che la modalità di funzionamento sia quella manuale (configurazione predefinita) in seguito inserire la scheda UPS nel connettore GPIO della Raspberry Pi.
- 3) Collegare la batteria LiPo alla scheda attraverso l'apposito connettore X1 presente sul lato saldature della scheda stessa.
- 4) Inserire la scheda nel connettore GPIO della Raspberry Pi.
- 5) Inserire il jumper di reset del microcontrollore nel connettore J2 come riportato nella **Fig. 5**: il diodo led DL4 potrebbe iniziare a lampeggiare.
- 6) Si applica al connettore J1 la tensione di alimentazione principale, si attende che la batteria abbia completato il ciclo di carica (il LED DL1 da verde fisso si spegne) e attraverso un pulsante normalmente aperto connesso su J2 tra il pin 4 ed il pin 3 (0V) da tenere premuto per almeno 3 secondi, si attiva la linea di accensione: la Raspberry Pi deve accendersi. I LED di stato DL2 e DL3 (entrambi di colore rosso) devono essere accesi e fissi, indicando la presenza della tensione a 5V e 3.3V sul connettore GPIO della Raspberry Pi.
- 7) Completato il *boot*, si effettua il login con credenziali root oppure come utente norma-

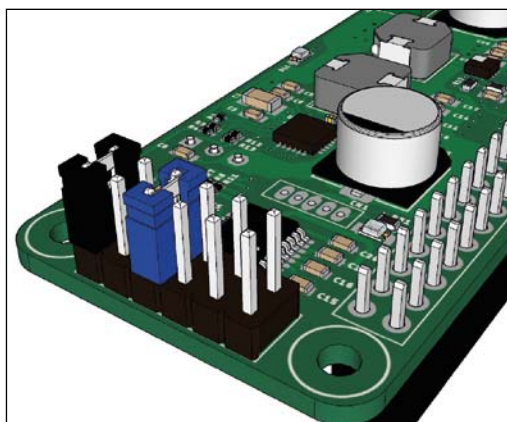


Fig. 7
Configurazione
della modalità
manuale.

le; in quest'ultimo caso si dovrà ricordare di premettere ai vari comandi riportati di seguito il comando `sudo` per permetterne l'esecuzione con privilegi di amministratore.

- 8) Si verifica se è stato attivato nel sistema in uso l'interfaccia I²C attraverso l'utility di configurazione del Raspberry *raspi-config* scrivendo nel terminale il seguente comando seguito dalla pressione del tasto *Invio*:

```
sudo raspi-config
```

Utilizzando i tasti freccia ci si posiziona sulla riga *Interfacing Options* (**Fig. 9**) e si preme il tasto invio; le immagini relative alla configurazione qui riportate si riferiscono ad una Raspberry Pi 4 model B rev 1.1. con sistema operativo *buster* e kernel Linux 4.19.75-v71+; per sistemi operativi differenti le opzioni potrebbero trovarsi in altri punti del menù di configurazione, ma saranno comunque sempre presenti e facilmente identificabili. Utilizzando i tasti freccia si ricerca l'opzione I²C e si preme

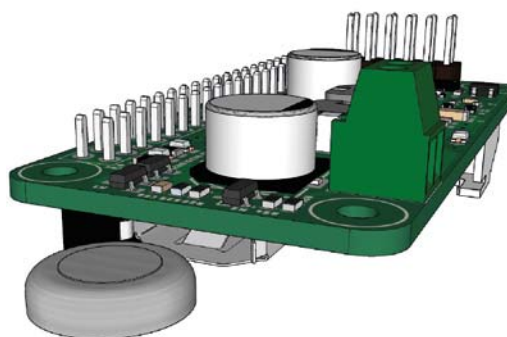


Fig. 8
Inserimento
della batteria
tampone per il
modulo RTC.

Invio (**Fig. 10**). Sempre mediante i tasti freccia si seleziona Yes e si preme il tasto Invio; si preme nuovamente il tasto Invio per confermare la selezione. Il sistema ripresenta la finestra iniziale, utilizzando il tasto Tab (tabulazione) ci si sposta sul pulsante Finish e si preme Invio per uscire dal programma di configurazione. Si procede con l'installazione del package "i2c-tools":

```
sudo apt-get update
sudo apt-get install i2c-tools
```

questo pacchetto permette di utilizzare una serie di strumenti utili per la gestione e la diagnostica del bus I²C della Raspberry Pi.

- 10) Lo scopo del modulo RTC è di fornire data ed ora esatti anche in assenza di connessione verso un server NTP, è quindi importante operare in modo che la lettura della data e ora venga effettuata al boot del sistema, questo è possibile andando a modificare il file **/boot/config.txt**. Nel caso delle versioni più recenti di sistema operativo basate su *systemd* (da Jessie in avanti – 25/04/2015), utilizzando l'editor *vi*, si scriverà il seguente comando:

```
sudo vi /boot/config.txt
```

Scorrendo fino alla fine del file si aggiunge la seguente riga:

```
dtoverlay=i2c-rtc,ds3231
```

A questo punto è possibile salvare le modifiche apportate al file ed uscire da *vi* premendo il tasto: e scrivendo *wq* seguito dalla pressione

del tasto *Invio*. Al prossimo riavvio l'ora sarà aggiornata automaticamente leggendola direttamente dal modulo RTC.

- 11) Si disabilita il modulo fake hwclock (utilizzato dal kernel per simulare il comportamento di un RTC reale nel caso in cui il sistema non sia dotato di una connessione di rete che permetta di avere data ed ora sempre aggiornate) per evitare possibili interferenze con il modulo **hwclock**:

```
sudo apt-get -y remove fake-hwclock
```

Si disattiva anche l'avvio automatico del servizio al boot del sistema, nel caso di sistemi operativi più datati basati ancora su *SysV* si utilizza il seguente comando:

```
sudo update-rc.d -f fake-hwclock remove
```

Per i sistemi più recenti basati invece su *systemd* (disponibile a partire dalla versione *Jessie*) si utilizza invece il seguente comando:

```
sudo systemctl disable fake-hwclock
```

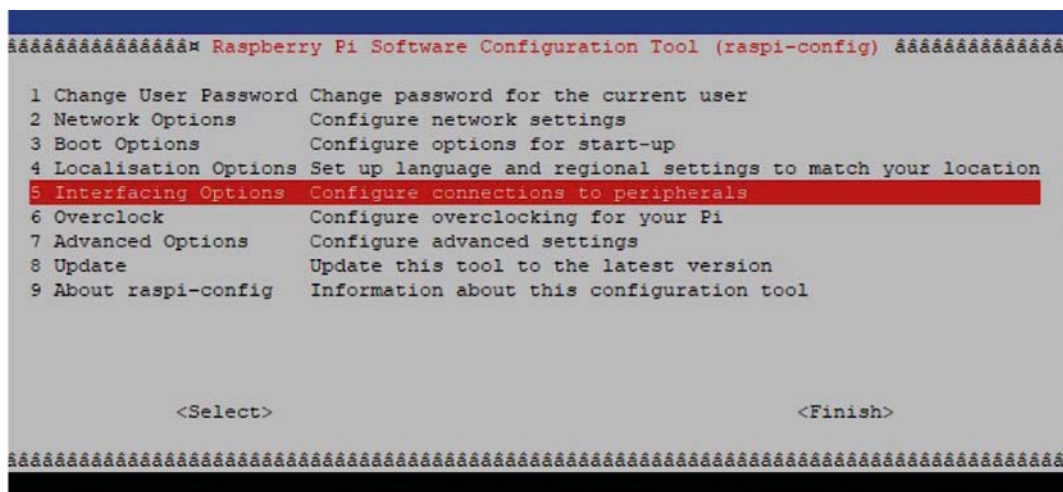
- 12) Utilizzando il proprio editor preferito (in questo esempio sarà utilizzato *vi*) aprire il file **/lib/udev/hwclock-set** quindi:

```
sudo vi /lib/udev/hwclock-set
```

scorrendo con i tasti freccia verso il basso si cercano nel file le seguenti righe:

```
if [ -e /run/systemd/system ] ; then
```

Fig. 9
Impostazione
delle opzioni
d'interfacciamento.



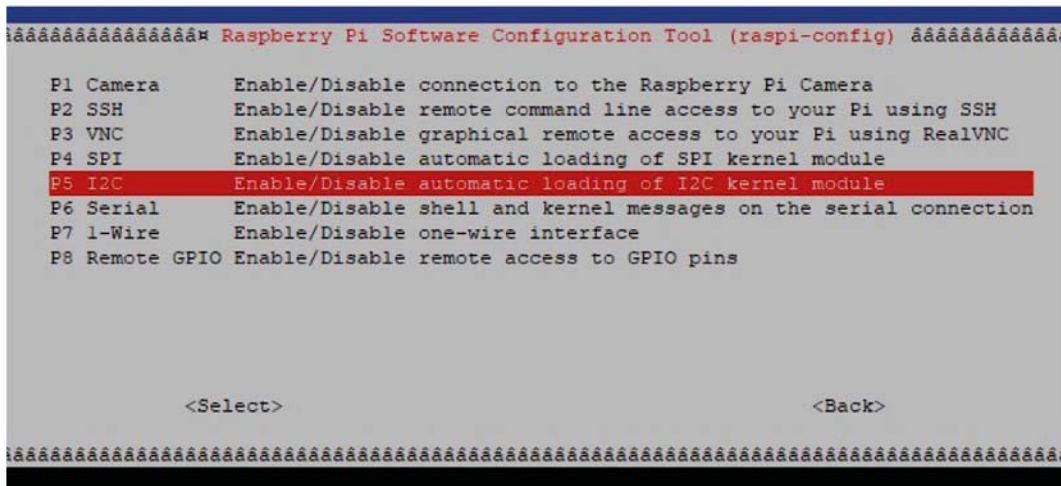


Fig. 10
Selezione
modalità I²C.

```
exit 0
fi
```

commentandole inserendo il carattere # ad ogni inizio riga, ottenendo quindi:

```
#if [ -e /run/systemd/system ] ; then
#exit 0
#fi
```

Scorrendo ancora verso la fine del file si commentano le righe come riportato nella **Fig. 11** (per maggiori informazioni vedere il manuale relativo allo strumento *hwclock*).

Si procede salvando in modo permanente le modifiche apportate al file ed uscendo da *vi* premendo il tasto: e scrivendo *wq* seguito dalla pressione del tasto *Invio*.

- 13) Si riavvia la Raspberry Pi con il seguente comando:

```
sudo reboot
```

seguito dalla pressione del tasto *Invio*.

- 14) Si effettua nuovamente il login e si verifica se il bus I²C risulta effettivamente abilitato dal kernel utilizzando il comando:

```
ls /dev/i2c*
```

Se si ottiene in risposta */dev/i2c-1* il bus I²C risulta correttamente configurato.

- 15) Per verificare la comunicazione con il modulo RTC si scrive nella finestra del terminale il seguente comando seguito dalla pressione del tasto *Invio*:

```
sudo i2cdetect -y 1
```

Se si ottiene la rappresentazione di **Fig. 12**, la quale indica che la scheda (0x08) e il modulo RTC (UU sull'indirizzo 0x68) sono correttamente identificati sul bus I²C.

- 16) Il modulo RTC è correttamente raggiungibile, con la Raspberry Pi connessa ad Internet si verificano le impostazioni di data ed ora utilizzando il comando:

```
sudo date
```

In assenza di connessione di rete, la data potrebbe non essere corretta; per impostarla manualmente si può utilizzare il comando:

```
sudo date -s "Wed Apr 28 10:27:00 CEST 2021"
```

Controllata la data si procede con lo scrivere nel modulo RTC la data ed ora corrente utilizzando il comando:

```
sudo hwclock -w
```

in seguito si rilegge la data memorizzata nel modulo RTC con il comando:

```
sudo hwclock -r
```

La data e l'ora devono corrispondere alla data e ora restituita attraverso l'utilizzo del comando **date**, potrebbero esserci alcuni secondi di differenza associati alla tempistica di esecuzione dei comandi.

Qualora ci fossero dei messaggi di errore conse-

Fig. 11
Commento delle
righe.

```
if [ yes = "$BADYEAR" ] ; then
#   /sbin/hwclock --rtc=$dev --systz --badyear
#   /sbin/hwclock --rtc=$dev --hctosys --badyear
else
#   /sbin/hwclock --rtc=$dev --systz
#   /sbin/hwclock --rtc=$dev --hctosys
fi
"/lib/udev/hwclock-set" 37 lines, 898964 characters
```

guenti all'esecuzione dei comandi sopra riportati, è probabile che ci siano dei problemi hardware con il modulo RTC, in questo caso si suggerisce di utilizzare una scheda differente e ripetere le operazioni sopra elencate.

ORGANIZZAZIONE REGISTRI PER L'INTERFACCIAMENTO I²C-BUS

La scheda si interfaccia su bus I²C esponendo all'utente una serie di registri, il cui accesso può essere effettuato dal bus I²C della Raspberry Pi oppure da qualsiasi altro sistema che sia in grado di comunicare su bus I²C. I registri sono descritti nella **Tabella 4**. La scrittura in un'area non permessa oppure di sola lettura non eseguirà l'operazione richiesta ed il contenuto della cella indirizzata non sarà quindi alterato; per la lettura su aree riservate potrebbero essere restituiti dei valori che non riflettono il reale contenuto della cella indirizzata.

Per una descrizione dettagliata dei singoli registri e del loro significato (per alcuni i singoli bit devono essere decodificati secondo determinati criteri) si rimanda al manuale d'uso del MiniUPS, scaricabile direttamente dal repository GitHub per la scheda, il cui link è riportato all'inizio dell'articolo.

LETTURA REGISTRO DA BUS I²C

Per la lettura di un registro attraverso la shell bash della Raspberry Pi si può utilizzare il seguente comando (può essere eseguito anche senza privilegi di amministratore) seguito dalla pressione di *Invio*:

```
i2cget -y 1 0x08 <indirizzo_registro_hex> b
```

ottenendo in risposta il valore, espresso in esadecimale, del valore letto all'indirizzo specificato. Se si dovesse ottenere in risposta Error: Read failed; questo può essere sintomo di un problema hardware nelle connessioni oppure nel caso in cui la lettura venga richiesta in corrispondenza di un reset del microcontrollore.

SCRITTURA REGISTRO DA BUS I²C

Per la scrittura di un registro attraverso la shell bash del Raspberry Pi si utilizza il seguente comando (può essere eseguito anche senza privilegi di amministratore) seguito dalla pressione del tasto *Invio*:

```
i2cset -y 1 0x08 <indirizzo_registro_hex> <valore_da_scrivere_hex> b
```

Fig. 12
Scheda MiniUPS
e modulo RTC
identificati dal bus
I²C.

```
root@raspberrypi:~# sudo i2cdetect -y 1
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                -- -- -- -- -- 08 -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- UU -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
root@raspberrypi:~#
```

INDIRIZZO	NOME	ACCESSO	FUNZIONE
0x00 ÷ 0x0F	EE_USER	R/W	Area utente EEPROM interna
0x10	POWERON_HI	R	Numero reset microcontrollore scheda HI-BYTE
0x11	POWERON_LO	R	Numero reset microcontrollore scheda LO-BYTE
0x2E	FW_REV	R	Revisione del firmware
0x2D	FW_VER	R	Versione del firmware
0x34	IBCHGMAX	R	Corrente massima impostata per la carica della batteria LiPo
0x3E	BATTV	R	Tensione della batteria LiPo
0x41	VIN	R	Tensione di ingresso (presenza/assenza e valore)
0x42	IBCHG	R	Corrente di carica della batteria LiPo
0x50	FAULT_FLAG	R	Flag segnalazione FAULT
0x51	FAULT_RST	R	Azzeramento del flag di FAULT
0x52	FAULT_CURR	R	Legge lo stato corrente del registro di FAULT
0x53	FAULT_PREV	R	Legge lo stato del registro di FAULT precedente all'ultimo evento di FAULT.
0x54	VINSTATE	R	Stato tensione di ingresso (presente/assente)
0x55	BTCHGSTAT	R	Fase carica batteria LiPo

Tabella 4
Descrizione registri; per la modalità di accesso, R è lettura, W scrittura e R/W lettura/scrittura.

LETTURA E SCRITTURA SU REGISTRO AREA UTENTE DELLA EEPROM

Si riporta un esempio di lettura, scrittura e successiva lettura per verifica della prima locazione di memoria dell'area utente sulla memoria EEPROM. Tutti i comandi sono seguiti dalla pressione del tasto *Invio*. Si inizia con la lettura del contenuto della memoria all'indirizzo 0x00:

```
pi@raspberrypi:~ $ i2cget -y 1 0x08 0x00 b
0x00
```

Si prova a scrivere, nella stessa locazione, il valore 128 ovvero 0x80 in esadecimale:

```
pi@raspberrypi:~ $ i2cset -y 1 0x08 0x00 0x80 b
```

si rilegge il contenuto della locazione appena scritta:

```
pi@raspberrypi:~ $ i2cget -y 1 0x08 0x00 b
0x80
```

Se si ottiene in risposta 0x80, questo conferma che il dato era stato effettivamente memorizzato nella locazione 0 all'interno dell'area EEPROM utente; se invece si ottiene un valore differente o

un messaggio di errore è opportuno ricontrollare le connessioni e se i comandi sono stati inseriti con la sintassi corretta.

CONCLUSIONI

Bene, per il momento ci fermiamo qui; nella prossima e conclusiva puntata spiegheremo come lavorare sul complesso Raspberry Pi - MiniUPS tramite degli script Python e in che modo possiamo gestire la porta seriale.



Cosa occorre?

Il miniUPS presentato in questo articolo (cod. MINIUPSRPI) è in vendita a Euro 58,00, la batteria LiPo da 430 mAh con controllo temperatura (cod. PLIB430) è disponibile a Euro 12,50. 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>



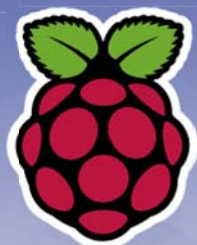
RASPBERRY PI 400!

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.



Utilizziamo il lettore RFID della nostra demoboard per creare un semplice database di tag per visualizzarne i dati in Azure IoT Central. Quinta puntata.

CONOSCERE

E USARE

Azure Sphere

5

di PIER CALDERAN

Prosegue la carellata sulle periferiche integrate nella nostra scheda di sviluppo per applicazioni IoT basate su piattaforma Microsoft Azure Sphere Futura MT3620; in questa puntata del corso andremo a proporre un'applicazione che permetterà di leggere i tag RFID tramite il lettore presente nella scheda di sviluppo e di inviarne i dati ad IoT Central. Il codice di esempio corrispondente si trova nella cartella Futura_samples\Futura_MT3620_RFID_IoT_Central, cui si accede clonando il repository GitHub https://github.com/piercalderan/futura_MT3620 in cui sono raccolti gli esempi di questo codice.

IL PROTOCOLLO RFID

RFID (Radio Frequency Identification) è una tecnologia per l'identificazione automatica di informazioni basata sulla capacità di memorizzare dati

digitali su particolari schede elettroniche dotate di memoria, chiamate PICC (Proximity Integrated Circuit Card) o, più comunemente, tag o transponder RFID. Queste card sono in grado di rispondere a distanza alle sollecitazioni elettromagnetiche, prodotte dai cosiddetti RFID Card Reader, ovvero lettori di card RFID. L'identificazione dei dati avviene mediante radiofrequenza, di solito sulla banda dei 13,56 MHz, con una sensibilità fino a 10 cm. La trasmissione dei dati arriva fino a 424 kbit/s, sufficienti a comunicare le informazioni contenute nei tag.

Il tag contiene un microchip con un identificativo univoco e una piccola memoria Flash ed è privo di alimentazione elettrica. Tramite l'antenna del ricevitore e il contatto ravvicinato di un lettore RFID, che emette il segnale radio, la radiofrequenza attiva il microchip e gli fornisce l'energia necessaria a



Fig. 1 - Il lettore RFID MFRC522.

rispondere al lettore, ritrasmettendo un segnale contenente le informazioni memorizzate nel chip. I lettori sono anche in grado di "scrivere", ovvero mandare piccole quantità di dati al tag, che li può memorizzare all'interno della sua memoria, la quale solitamente è da 1.024 byte (1 kB).

RFID MFRC522

Il lettore RFID che va montato nella nostra scheda è basato sul chip MFRC522 della NXP Semiconductors, progettato per la lettura/scrittura di tag e per l'identificazione a radiofrequenza a 13,56 MHz (Fig. 1).

Il modulo RFID può essere utilizzato per applicazioni sia professionali che amatoriali nell'ambito del controllo acces-

si, robotica, sistemi di pagamento, antifurto, eccetera. Il modulo supporta gli standard ISO/IEC 14443A e MIFARE e viene fornito con l'antenna di lettura/scrittura a 13,56MHz, con 2 tag RFID e due connettori con 8 pin dritti e piegati a 90°. La scheda supporta card Mifare1 S50, Mifare1 S70 Ultralight, Mifare Pro e Mifare DESFire. Per la connessione viene usata la porta SPI.

Dopo aver saldato il connettore con gli otto pin piegati a 90° si può inserire il lettore nel connettore femmina della scheda, come illustrato in Fig. 2.

ESEMPIO FUTURA MT3620 RFID IOT CENTRAL

Il progetto include una libreria open source MFRC522, simile a quella che di solito viene usata per Arduino, ma che è stata modificata per MT3620. Per i più curiosi, abbiamo aggiunto nella cartella del progetto anche il pdf ufficiale MFRC522.pdf, con tutte le informazioni sul chip NXP MFRC522 e sul protocollo RFID usato.

Si fa notare che i file mfr522.c e mfr522.h della libreria sono stati messi nella stessa cartella del progetto e importati localmente nel codice con la direttiva `#include "mfr522.h"`.

La libreria mfr522 comunica con l'interfaccia SPI Master della scheda identificata nel nostro file di definizione hardware come MT3620_ISU1_SPI, corrispondente ai pin fisici 22, 23, 24, 25 e 26 dell'MT3620.

Come mostra la Fig. 3, i collegamenti tra i GPIO che nella scheda demo sono associati al bus SPI e il lettore sono i seguenti:

• GPIO31	SCK	22
• GPIO32	MOSI	22
• GPIO33	MISO	23
• GPIO34	CS	26
• GPIO35	RST	25

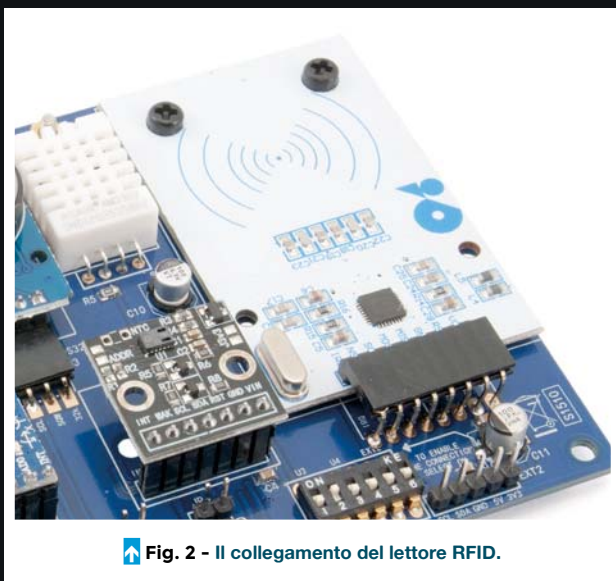


Fig. 2 - Il collegamento del lettore RFID.

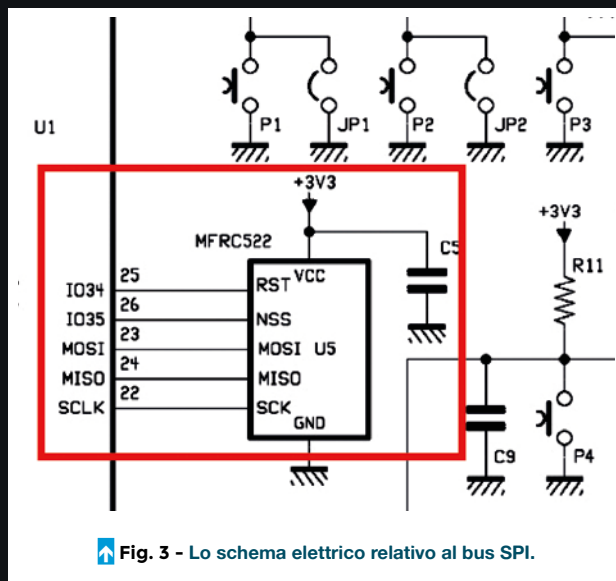


Fig. 3 - Lo schema elettrico relativo al bus SPI.

IL BUS SPI

Il bus SPI (Serial Peripheral Interface) è un sistema di comunicazione seriale tra un microcontrollore master e uno o più microcontrollori slave: il master controlla il bus, emette il segnale di clock e decide quando iniziare e terminare la comunicazione con le unità slave.

Il bus funziona solitamente su quattro segnali fra una periferica master e una o più periferiche slave in parallelo (Fig. 4):

- SCLK: Serial Clock (emesso dal master);
- MISO: Master Input Slave Output (ingresso per il master e uscita per lo slave);
- MOSI: Master Output Slave Input (uscita dal master);
- CS (o SS, NSS): Chip Select o Slave Select, emesso dal master per selezionare il dispositivo slave.

LA FUNZIONE DI LETTURA RFID

Dopo aver caricato in Visual Studio il file CMakeLists.txt dalla cartella dell'esempio, possiamo analizzare il nostro main.c per capire come funziona il tutto.

Il cuore dell'applicazione è nella funzione di lettura `RFIDTimerEventHandler`, che usa i metodi della libreria `mfr522` per leggere il registro 0x20 e 0x80 del chip del lettore. In questo modo tenta la lettura della card, che è un ID univoco composto da 4 cifre (riferitevi al riquadro dedicato in queste pagine).

Per i particolari della funzione riferitevi al **Listato 1**.

Quando si avvicina il tag all'antenna, se viene trovato l'ID, la funzione lo estrae e lo stampa nella finestra di Output:

```
[MFR522][INFO] Serial: 291C0EA3
```

LA STRUTTURA PRICEMAP

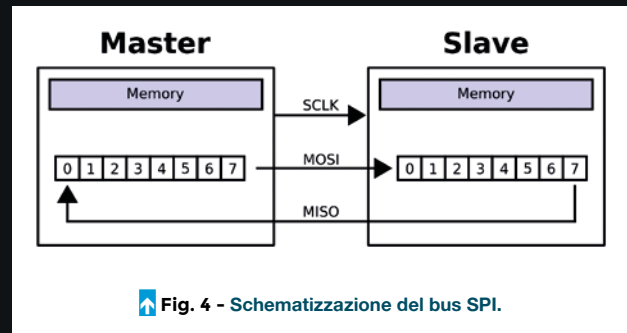
Per semplificare l'esempio, abbiamo pensato di creare una struttura di prezzi "priceMap" da abbinare rispettivamente al numero seriale di ogni card RFID. Siccome avete a disposizione solo due tag, che vengono dati insieme al lettore, abbiamo pensato di strutturare il database dei prezzi in questo modo:

```
<ID tag 1> = Modello 1: €100
<ID tag 2> = Modello 2: €200
```

Ovviamente, acquistando ulteriori tag potete "allungare" la vostra struttura priceMap in questo modo:

```
<ID tag 3> = Modello 3: €50
<ID tag 4> = Modello 4: €45
<ID tag 5> = Modello 5: €1200
<ID tag 6> = Modello 6: €300
```

... e via di seguito.



Per fare questo, abbiamo sfruttato la comoda libreria `map.c` (presente nella cartella del progetto) che troverete inizializzata nella funzione `main` così:

```
map_init(&priceMap);
map_set(&priceMap, "83B03F16 ", "Modello 1: €100");
map_set(&priceMap, "D098C71A", "Modello 2: €200");
map_set(&priceMap, "31AC6A1C", "Modello 3: €300");
```

Resta inteso che gli ID dei vostri tag saranno diversi e dovrete procurarveli con una prima lettura delle card, che

RFID UID

Ogni card RFID contiene un chip integrato con un numero di identificazione permanente univoco (UID, Unique Identifier), che viene incorporato durante il processo di produzione. Questo UID è spesso indicato come il numero di serie della carta (CSN, Card Serial Number). Il numero di serie della card non è crittografato e qualsiasi lettore conforme ISO può leggere il numero di serie della card.

Il numero di serie della carta MIFARE è l'identificativo univoco definito nella norma ISO 14443-3A. Esistono 3 tipi di UID definiti nello standard:

- singolo (4 byte)
- doppio (7 byte)
- triplo (10 byte)

Un UID a 4 byte è un identificatore assegnato dal produttore della carta utilizzando un database controllato. Questo database garantisce che un singolo identificatore non venga utilizzato due volte. C'è da dire che gli UID a dimensione singola (4 byte) hanno un limite poiché il numero di ID utilizzabili è limitato a soli 4 miliardi di pezzi circa. Si consiglia di utilizzare UID Double Size (7 byte) per garantire l'unicità della card RFID.

↓ Listato 1

```
static void RFIDTimerEventHandler(void)
{
    // Preparazione alla lettura dei tag
    uint8_t byte;
    byte = mfrc522_read(ComIEnReg);
    mfrc522_write(ComIEnReg, byte | 0x20);
    byte = mfrc522_read(DivIEnReg);
    mfrc522_write(DivIEnReg, byte | 0x80);

    // Prova a leggere la scheda RFID
    byte = mfrc522_request(PICC_REQALL, str_rfid); // trova l'ID

    if (byte == CARD_FOUND)
    {
        Log_Debug("[MFRC522][INFO] Trovata card: %x\n", byte);

        byte = mfrc522_get_card_serial(str_rfid);

        if (byte == CARD_FOUND)
        {
            Log_Debug("Card dumping: ");
            for (byte = 0; byte < 8; byte++)
            {
                Log_Debug("%x", str_rfid[byte]); // dump ID in
                byte array
            }
            // Converte l'array di byte in una stringa di byte
            char hexstr[8];
            bytetoHex(hexstr, str_rfid, 8);
            hexstr[8] = 0;
            Log_Debug("\nhex: %x\n", hexstr);
            int* val=0;
            if (map_get(&priceMap, hexstr)>0)
            {
                val=map_get(&priceMap, hexstr);
                Log_Debug("[MFRC522][INFO] Serial: %s\n",
                hexstr);
                Log_Debug("[MAP][INFO] Map Price: %s\n", *val);
            }
            else
            {
                Log_Debug("[MFRC522][INFO] Pre Serial: %s\n",
                hexstr);
                return;
            }

            static const char* EventMsgTemplate = "%s";
            int len = snprintf(eventBuffer, sizeof(eventBuffer),
            EventMsgTemplate, *val);
            Log_Debug("[IoTCentral][INFO] Sending IoT Central
            Message: %s\n", eventBuffer);
        }
    }
}
```

spiegheremo a breve. Dopodiché potrete scrivere gli ID rilevati nella suddetta mappa dei prezzi. Inutile dire che al posto di una mappa di prezzi potete abbinare agli ID qualsiasi altra cosa.

Una cosa importante da sottolineare è che l'ID del tag viene espresso in un array di byte, per cui si rende necessario convertire l'array in una stringa di testo.

In questo modo, la stringa potrà essere inviata a IoT Central tramite la consueta funzione SendTelemetry, assieme al prezzo assegnato.

Questa operazione viene svolta dalla funzione bytetoHex, che è molto semplice e serve solo a convertire ogni byte dell'array in una stringa esadecimale.

```
void bytetoHex(char* xp, const char* bb, int n)
{
```

```
    const char xx[] = "0123456789ABCDEF";
    while (--n >= 0) xp[n] = xx[(bb[n] >> 1] >> ((1 - (n & 1)) <<
    2)) & 0xF];
}
```

La stringa contenente il numero di serie della card viene memorizzata nella variabile "val" che restituisce il valore del metodo map_get della libreria map.c:

```
val = map_get(&priceMap, hexstr)
```

Per ottenere il numero seriale del tag, basta appoggiarlo sull'antenna del lettore e avviare il debug come al solito. Una volta che il lettore rileva il numero di serie, verrà stampato nella finestra di Output qualcosa di simile a questo:

```
[MFRC522][INFO] Pre Serial: 83B03F16
```

A questo punto, dovete fermare l'applicazione, copiare il numero di serie dalla finestra di Output (nel nostro caso 83B03F16) e incollarlo nella struttura priceMap che si trova nella funzione main:

```
map_set(&priceMap, "83B03F16", "Modello 1: €100");
```

Ripetete l'operazione per tutti tag che volete. Dopodiché, con un puntatore al valore della mappa potrete estrarre dalla mappa il testo abbinato, ovvero il prezzo (o qualsiasi altra cosa). Quello che verrà inviato a IoT Central sarà qualcosa di simile a questo, stampato nella finestra di Output:

```
Sending IoT Central Message: { "RFID": "Modello 1: €100" }
```

FUNZIONE SENDRFIDBUTTONHANDLER

I dati RFID vengono mandati a IoT Central solo quando viene premuto il pulsante 1 della scheda. La funzione sendRFIDButtonHandler legge lo stato del pulsante e a sua volta chiama la funzione RFIDTimerEventHandler per la lettura del tag. Viene quindi chiamata la funzione SendTelemetry che invia l'evento RFID e il valore abbinato, nel nostro caso il prezzo. Notare che l'evento da inviare a IoT Central nel formato {key:value} contiene il campo "RFID" tutto maiuscolo, che va inserito nel campo Nome dell'interfaccia del modello di dispositivo, e il testo (prezzo) abbinato all'ID del tag:

```
static void sendRFIDButtonHandler(void)
{
    if (IsButtonPressed(sendRFIDButtonGpioFd, &sendRFIDButton-
    State))
    {
        RFIDTimerEventHandler();
        SendTelemetry("RFID", eventBuffer); //manda il testo
        abbinato all'ID del tag nel campo RFID
    }
}
```



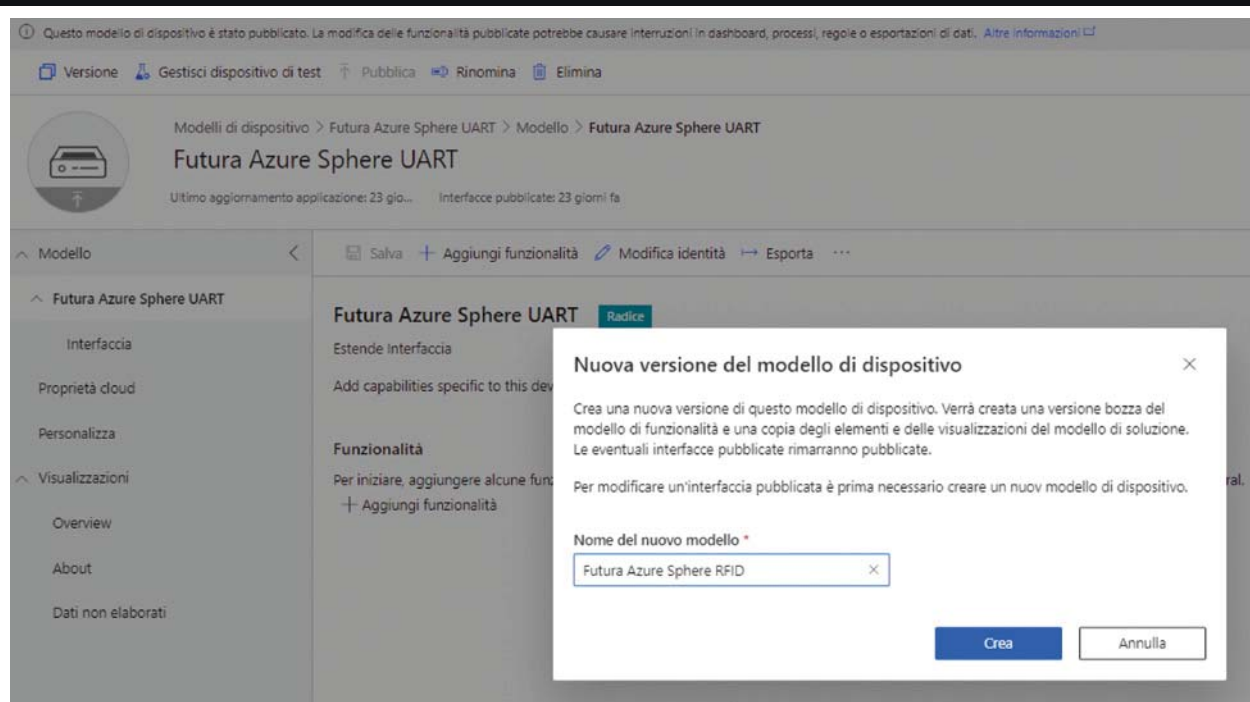


Fig. 5 - La finestra nuova versione del modello.

IL FILE APP_MANIFEST.JSON

Per l'invio dei dati tramite il pulsante 1 della scheda bisogna impostare nelle "Capabilities" del file app_manifest.json la variabile SAMPLE_BUTTON_1 della definizione hardware. Per la lettura dei dati bisogna attivare SAMPLE_ISU1_SPI che apre il bus SPI master. Gli argomenti "CmdArgs", "AllowedConnections" e "DeviceAuthentication" sono sempre gli stessi degli esempi delle scorse puntate:

```
"CmdArgs": [ "0ne001C85E4" ], // Ambito ID
"Capabilities": {
  "AllowedConnections": [
    "global.azure-devices-provisioning.net", "iotc-f18c6479-0fd2-
    4197-80f0-337ba9198b77.azure-devices.net" // connessione ammessa
  ],
  "Gpio": [ "$SAMPLE_BUTTON_1" ],
  "SpiMaster": [ "$SAMPLE_ISU1_SPI" ],
  "DeviceAuthentication": "df7ba4db-4a32-4bfd-b401-59f2846cc320"
```

VISUALIZZAZIONE RFID SU IOT CENTRAL

Come al solito vogliamo visualizzare i dati di telemetria su IoT Central. Per questo esempio, dovreste creare un modello di dispositivo con una finestra di testo per visualizzare il prezzo abbinato al tag.

Invece di creare un modello ex-novo, come spiegato nelle puntate precedenti, oggi impariamo a sfruttare il meccanismo di "Versione del modello".

Selezionando Versione nella pagina del modello, potete

creare interfacce e viste alternative per lo stesso dispositivo. Potete anche saltare tutti i passaggi preliminari per la creazione di un nuovo modello copiando la struttura in una nuova versione di un modello di dispositivo precedente. Ecco la procedura che permette di fare ciò.

1. Selezionate un modello di dispositivo creato in precedenza, per esempio il modello UART che avete creato nella puntata scorsa.
2. Fate clic su Versione nel menu in alto a sinistra: apparirà una finestra con il nome del modello con l'aggiunta di v2 alla fine, cioè versione 2.
3. Cambiate il nome per il modello, per esempio Futura Azure Sphere RFID.
4. Fate clic su Crea (Fig. 5).
5. Aprite l'interfaccia del modello dal pannello a sinistra.
6. Cambiate il nome visualizzato in RFID (Fig. 6), che è lo stesso del nome usato dalla coppia {key: value} nella funzione SendTelemetry (ricordate sempre questo particolare!).
7. Selezionate il tipo semantico Nessuno e lo schema String.
8. Salvate cliccando sull'icona Salva.
9. Selezionate Overview nel pannello Visualizzazioni, affinché appariranno i riquadri del modello precedente (nel nostro caso UART).
10. Eliminate i riquadri UART con l'icona X (elimina).
11. Selezionate nella casella telemetria la voce RFID.
12. Aggiungete un riquadro e scegliete il tipo di visualizza-

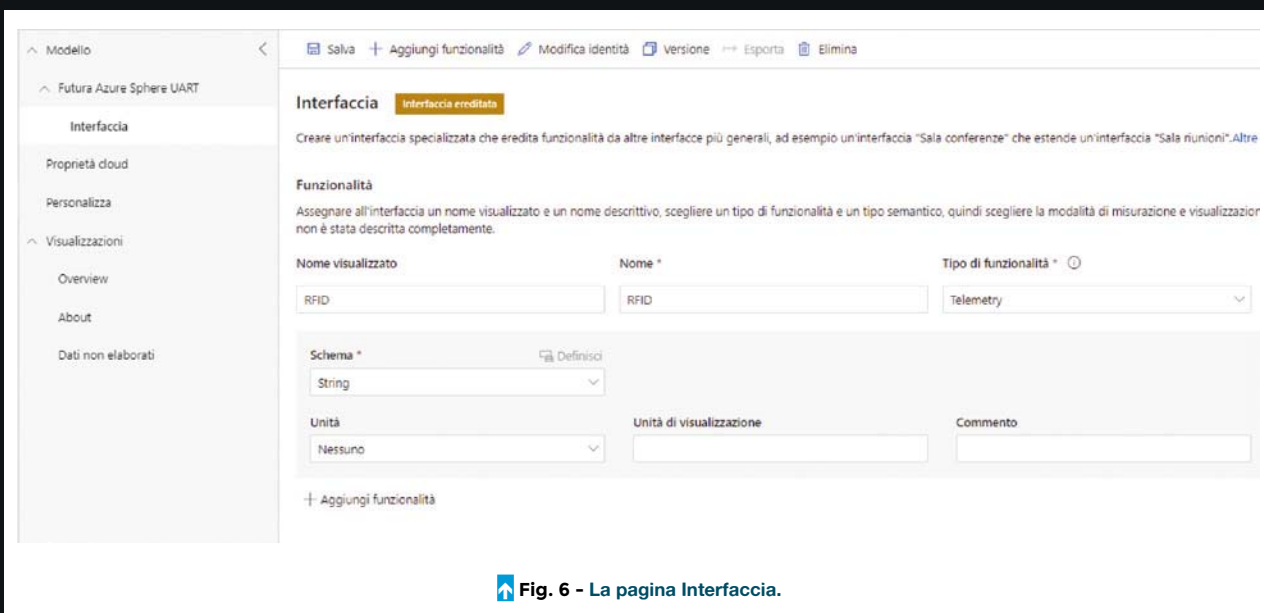


Fig. 6 - La pagina Interfaccia.

zione Ultimo valore noto (Fig. 7).

13. Andate in Dispositivi e selezionate il vostro dispositivo.
14. Fate clic su Esegui la migrazione, allorché apparirà una finestra in cui selezionare il modello di dispositivo, nel nostro caso Futura Azure Sphere RFID (Fig. 8).
15. Aprite il dispositivo per vedere il riquadro con l'ultimo

valore noto, nel nostro caso "Modello 1: €100" (Fig. 9). Una volta avviata l'applicazione, nella finestra di Output vedrete qualcosa di simile a questo:

```
Remote debugging from host 192.168.35.1, port 58508
SPI Initializing
```

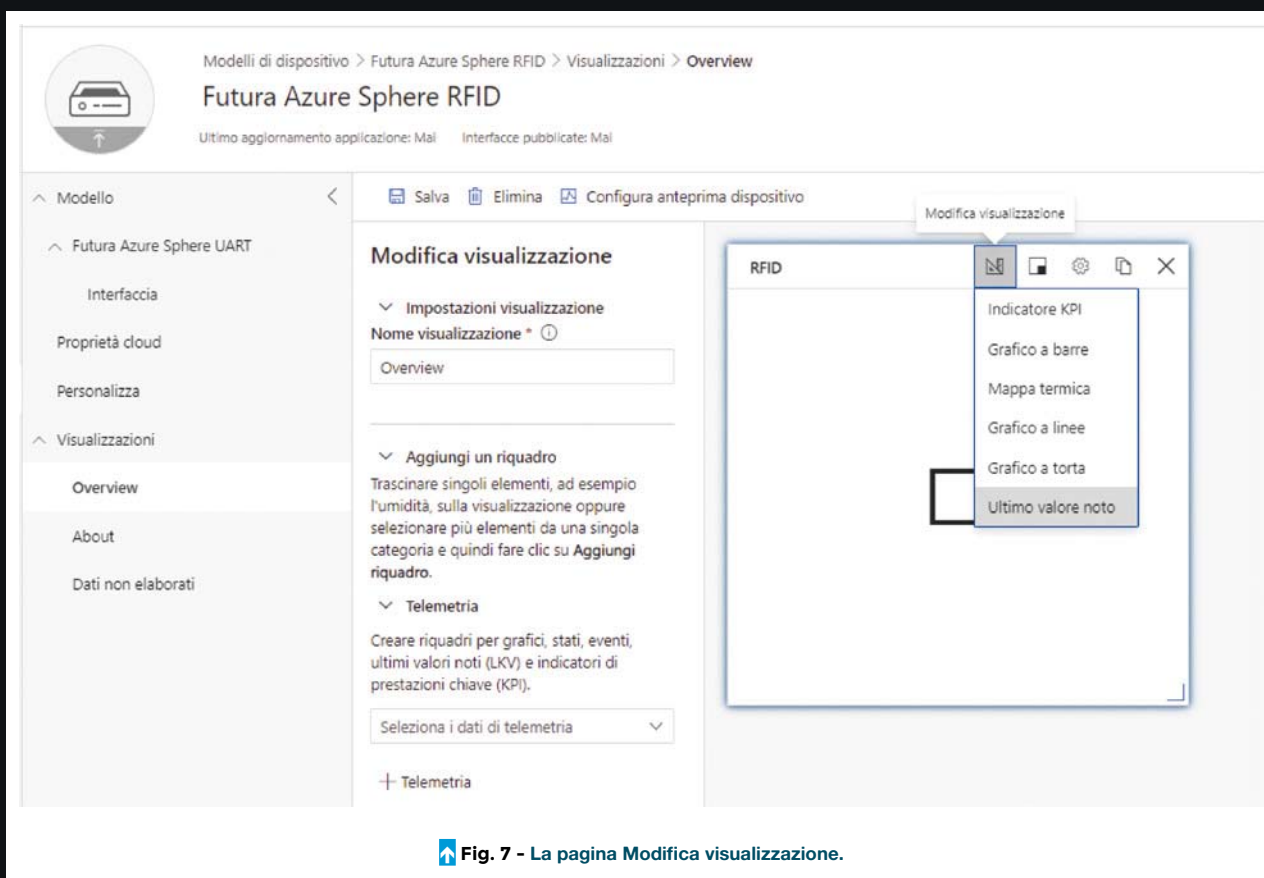


Fig. 7 - La pagina Modifica visualizzazione.

```
[SPI] Opened SPI Interface
[SPI] BusSpeed = 4000000
[SPI] SPIMode = 1
[SPI] BitOrder = SPI_BitOrder_MsbFirst
SPI FD Set on 3
[RFC522] Initialized
RFID Scanner trovato!
RFID application starting.
Setting Azure Scope ID 0ne001C85E4
Apertura di SAMPLE_BUTTON_1 come input
Provo a determinare la versione dello scanner
Version trovata 146 (Hex: 92)
[Azure IoT] Using HSM cert at /run/daa/df7ba4db-4a32-4bfd-b401-59f2846cc320
IoTHubDeviceClient_LL_CreateWithAzureSphereDeviceAuthProvisioning returned 'AZURE_SPHERE_PROV_RESULT_OK'.
IoT Central Authenticated: IOTHUB_CLIENT_CONNECTION_OK
```

Come si può vedere dal debug, il bus SPI viene correttamente inizializzato e il lettore viene riconosciuto con indirizzo 0x92 (146 in decimale) corrispondente alla versione 2.0 del chip MFRC522 (vedi datasheet del produttore NXP). Ogni volta che avvicinerete un tag all'antenna del lettore, ad ogni pressione del pulsante 1 vedrete qualcosa di simile a questo:

```
[MFRC522][INFO] Trovata card: 1
Card dumping: 83b03f161a000
hex: beffdbec
[MFRC522][INFO] Serial: 83B03F16
[MAP][INFO] Map Price: Modello 1: €100
[IoTCentral][INFO] Sending IoT Central Message: Modello 1: €100
INFO: IoTHubClient accepted the message for delivery
INFO: Message received by IoT Central. Result is: 0
INFO: Message received by IoT Central. Result is: 0
```

Come potete vedere, viene riconosciuta una card e il suo numero seriale.

L'ID associato al testo "Model 1: €100", come impostato nella struttura priceMap.

Il messaggio Sending IoT Central Message: { "RFID": "Model 1: €100" } indica che il testo abbinato al tag è stato riconosciuto e inviato a IoT Central.

I successivi messaggi indicano che tutto è stato ricevuto accettato correttamente da IoT Central.

ANALISI DEI DATI

Fino ad oggi abbiamo visto come mandare i dati di telemetria a IoT Central da vari dispositivi.

Una comoda funzionalità di analisi ci permette di analizzare i dati di qualsiasi dispositivo e di qualsiasi telemetria in una finestra temporale personalizzabile.

La pagina *Analisi* (Fig. 10) è costituita dai tre componenti principali di seguito descritti.

- **Pannello configurazione dati:** iniziate da qui per selezionare il gruppo di dispositivi per il quale desiderate analizzare i dati. Selezionate i dati di telemetria che volete analizzare e selezionate il metodo di aggregazione per ogni telemetria. Con "Raggruppa per" si possono raggruppare i dati usando le proprietà del dispositivo.
- **Controllo temporale:** il controllo della linea temporale viene usato per selezionare la durata per cui si desidera analizzare i dati. È possibile trascinare entrambe le estremità della linea temporale (timeframe) per selezionare l'intervallo di tempo. Il controllo dell'ora dispone anche di un dispositivo di scorrimento "Dimensioni intervallo" che controlla la dimensione dell'intervallo utilizzata per aggregare i dati.
- **Controllo Chart:** il controllo Chart "Visualizza" i dati sotto forma di grafico a linee. È possibile impostare la visibilità di linee specifiche interagendo con la legenda. Nel campo "Gruppi dispositivi" scegliere un gruppo, per esempio, "I miei dispositivi".

Nel campo "Telemetria" dovete scegliere una telemetria: per esempio "Temperature". Con "+Aggiungi" si può aggiungere la telemetria di altri parametri.

Nel campo "Raggruppa per" si può selezionare, per esempio, "ID dispositivo", "Modello", "Nessuno" ecc.

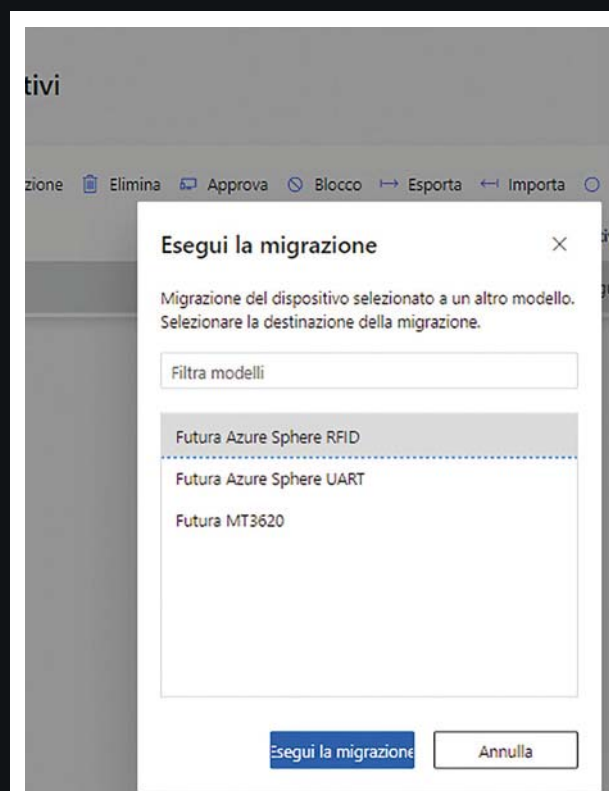
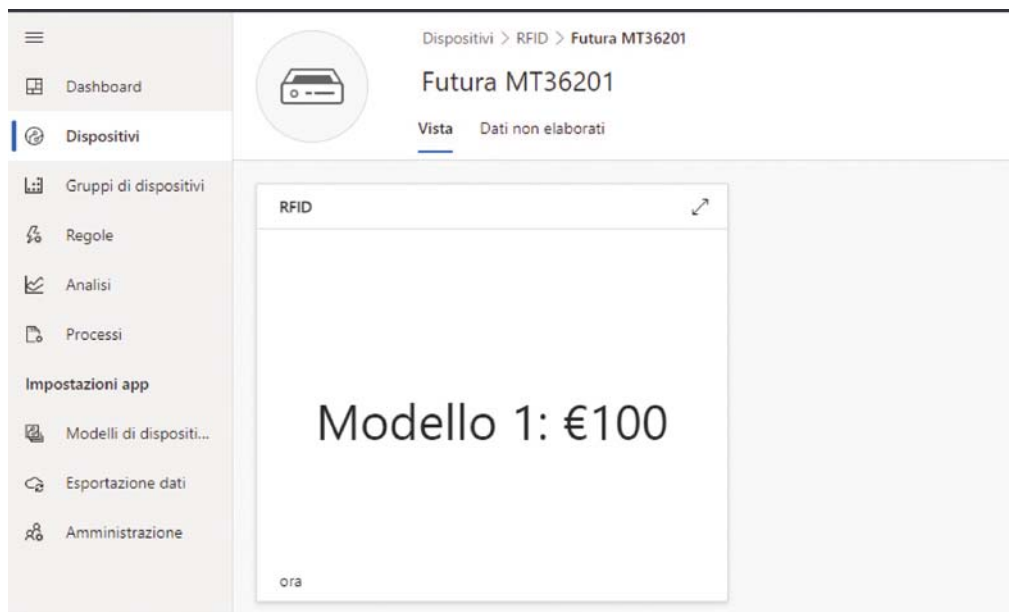


Fig. 8 - La finestra Esegui la migrazione.

Fig. 9 - La visualizzazione dell'ultimo valore noto.



Con il tasto "Analizza" si avvia l'analisi nell'arco temporale impostato dai cursori della linea temporale. Dopo aver eseguito una query sui dati, è possibile iniziare a visualizzarli sul grafico a linee. È possibile visualizzare/nascondere la telemetria, modificare la durata, visualizzare i dati di telemetria in una griglia di dati.

- Finestra temporale: per impostazione predefinita, i dati sono relativi all'ultimo giorno. È possibile trascinare una delle estremità del dispositivo di scorrimento per modificare la durata dell'ora. È anche possibile usare il controllo Calendar per selezionare uno dei contenitori temporali predefiniti oppure

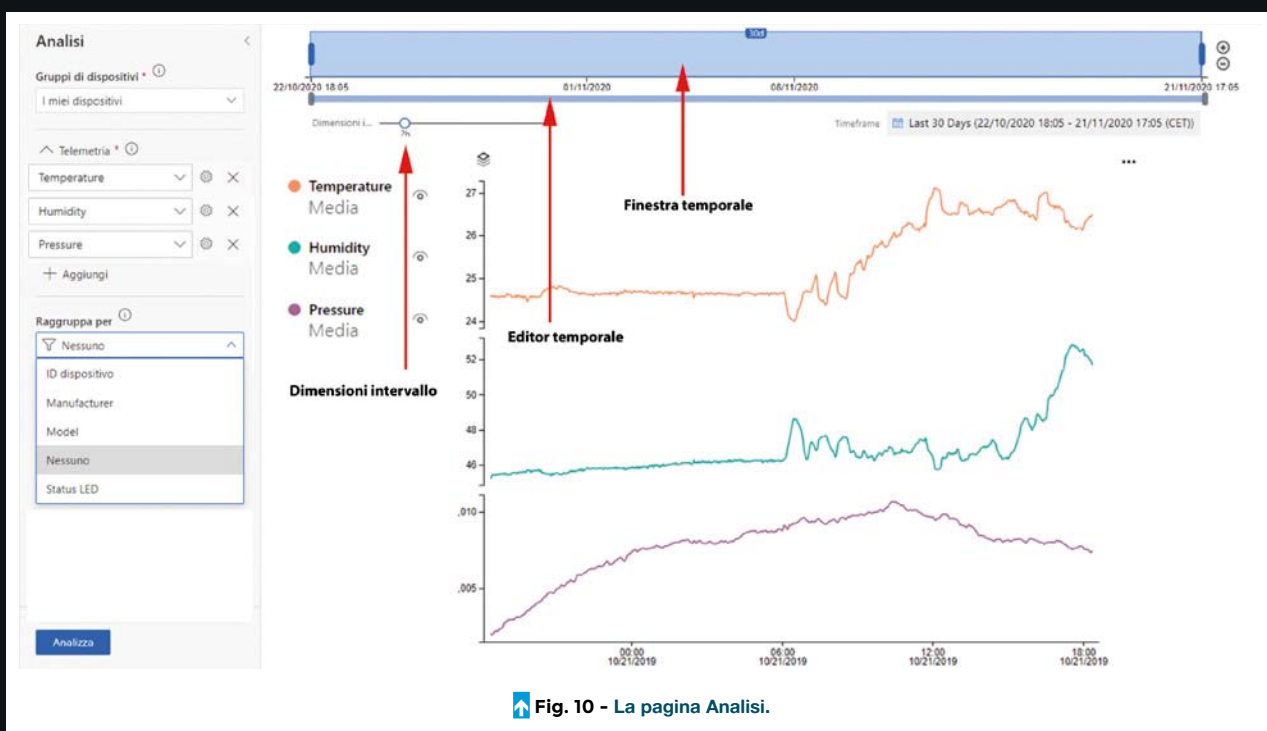


Fig. 10 - La pagina Analisi.

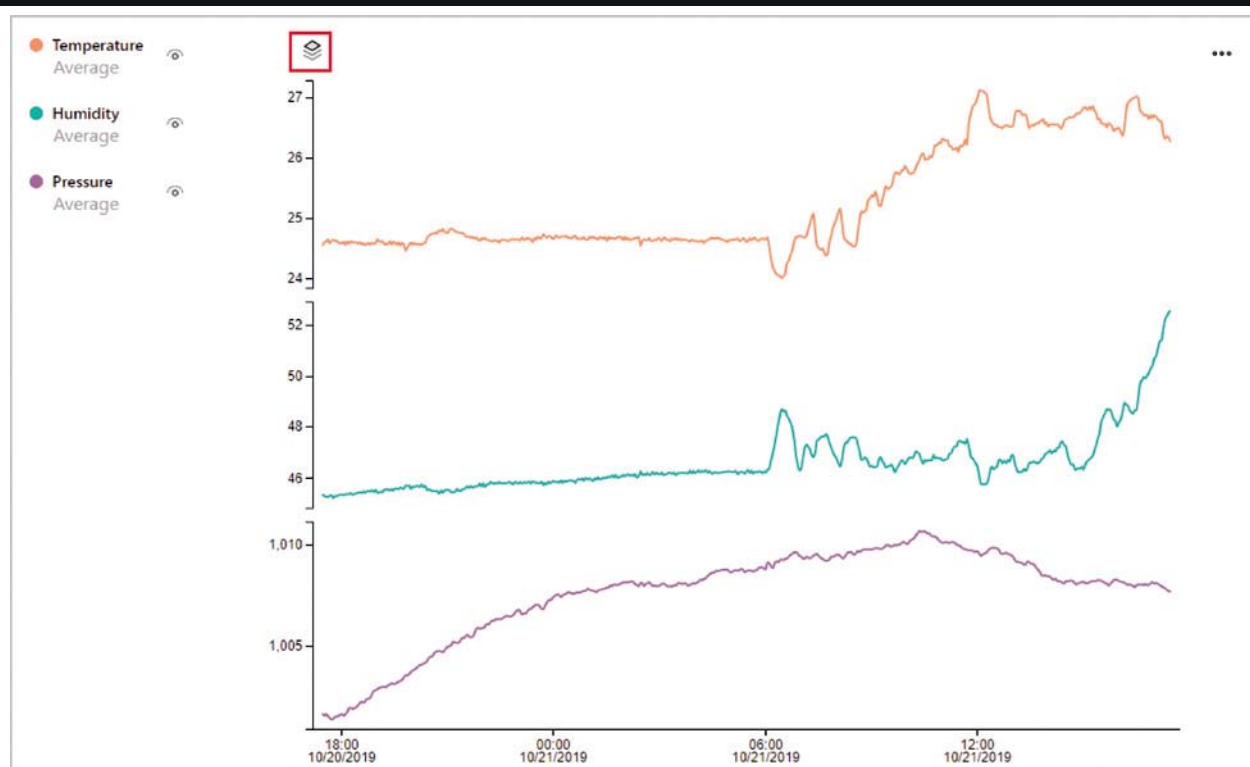


Fig. 11 - Il Controllo del formato dell'asse y.

selezionare un intervallo di tempo personalizzato.

Il controllo dell'ora dispone anche di un dispositivo di scorrimento Dimensioni intervallo che controlla la dimensione dell'intervallo utilizzata per aggregare i dati.

- Editor temporale: è uno strumento di scorrimento per l'intervallo di date interno. Usate i due controlli finali trascinandoli nell'intervallo di tempo desiderato. Questo intervallo di date interno è vincolato dalla linea temporale dell'intervallo di date esterno. Usate i controlli per selezionare l'intervallo di date esterno, che sarà disponibile per il controllo dell'intervallo di date interno. Aumentate o riducete i pulsanti per l'intervallo desiderato.
- Dimensioni intervallo: lo si può usare per ingrandire e ridurre gli intervalli nello stesso intervallo di tempo. Questa azione garantisce un controllo più preciso dello spostamento tra intervalli di tempo di particolare entità. È possibile usarlo per visualizzare viste granulari e ad alta risoluzione dei dati, anche fino ai millisecondi. Il punto di partenza predefinito del dispositivo di scorrimento è impostato come visualizzazione ottimale dei dati della selezione, che bilancia la risoluzione, la velocità delle query e la granularità.
- Selezione intervallo di date: con questo controllo è possibile selezionare gli intervalli di data e ora desiderati. È anche possibile usare il controllo per passare da una fascia oraria a un'altra.

Dopo avere apportato le modifiche da applicare all'area di lavoro corrente, selezionate Salva.

- Legenda grafico: la legenda del grafico mostra i dati di telemetria selezionati sul grafico. È possibile passare il mouse su ogni elemento della legenda per attivarlo nel grafico. Quando si usa "Raggruppa per", i dati di telemetria vengono raggruppati in base ai rispettivi valori della dimensione selezionata. È possibile impostare la visibilità di ogni telemetria specifica o l'intero gruppo facendo clic sul nome del gruppo.
- Controllo del formato dell'asse y (Fig. 11): la modalità dell'asse y scorre le opzioni di visualizzazione asse y disponibili. Questo controllo è disponibile solo quando vengono visualizzate diverse telemetrie. È possibile impostare l'asse y selezionando una delle tre modalità seguenti:
 - In Pila: un grafico per ogni telemetria è in pila e ognuno dei grafici ha un proprio asse y (questa modalità è impostata come predefinita);
 - Condiviso: un grafico per ogni telemetria viene tracciato con lo stesso asse y;
 - Sovrapposizione: serve per eseguire lo stack di più righe sullo stesso asse y, con la modifica dei dati dell'asse y in base alla riga selezionata.
- Controllo zoom: quando si trova un periodo di tempo in cui si vuole concentrare l'attenzione nel set di risultati, usare il puntatore del mouse per selezionare l'area e

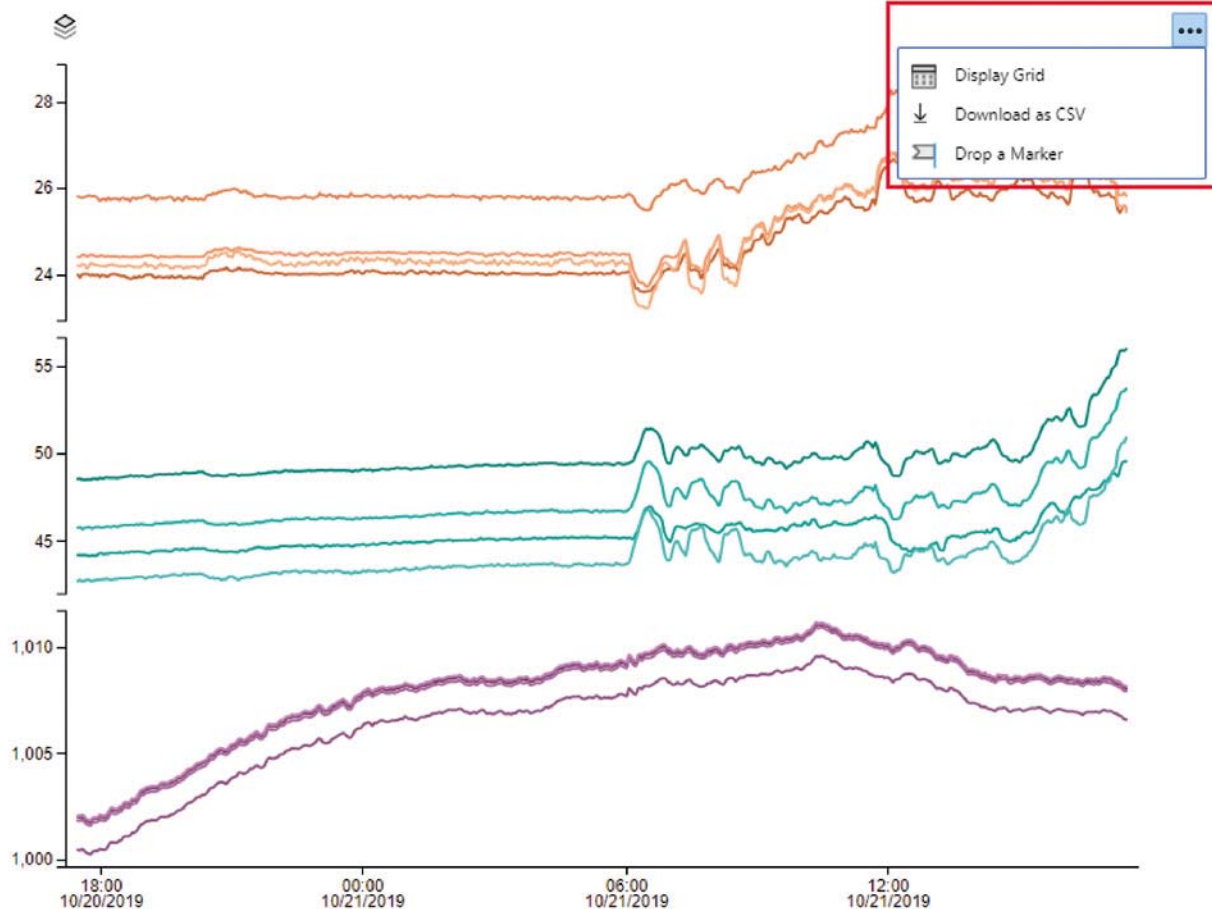


Fig. 12 - Il menu contenente i controlli Chart.

trascinarla nell'endpoint desiderato. Fate quindi clic con il pulsante destro del mouse sull'area selezionata e scegliere Zoom. Sotto i tre puntini ... si apre un menu con i controlli "Chart" per interagire con i dati (Fig. 12).

- Griglia di visualizzazione: i risultati sono disponibili in formato tabella, consentendo di visualizzare il valore specifico per ogni punto dati.
- Scarica come CSV: i risultati sono disponibili per l'esportazione come file con valori delimitati da virgole (CSV). Il file CSV contiene i dati per ogni dispositivo. I risultati vengono esportati utilizzando l'intervallo e l'intervallo di tempo specificati.
- Rilasciare un marcatore: il controllo "drop marker" fornisce un modo per marcare determinati punti dati nel grafico. È utile quando si tenta di confrontare i dati di più righe tra periodi di tempo diversi.

CONCLUSIONI

Per ora ci fermiamo qui. Con la speranza di essere stati sufficientemente chiari, vi diamo appuntamento alla prossima puntata, nella quale spiegheremo come creare un'interfac-

cia di controllo della nostra scheda da IoT Central, con tanto di pulsanti e altro del genere.



Cosa occorre?

La demoboard di Futura Elettronica con MT3620 (cod. FEAZURESPH) è disponibile a Euro 128,00 e include i componenti base per poter sperimentare con la tecnologia Azure Sphere. Dal sito futurashop.it è possibile acquistare anche le breakout board correlate alla demoboard. La board USB/TTL in grado di creare quattro porte seriali virtuali (cod. FT1497K) è in vendita a Euro 35,00, mentre la board adattatrice necessaria per programmare la demoboard (cod. FT1498K) è disponibile a Euro 25,00. I prezzi si intendono IVA compresa.

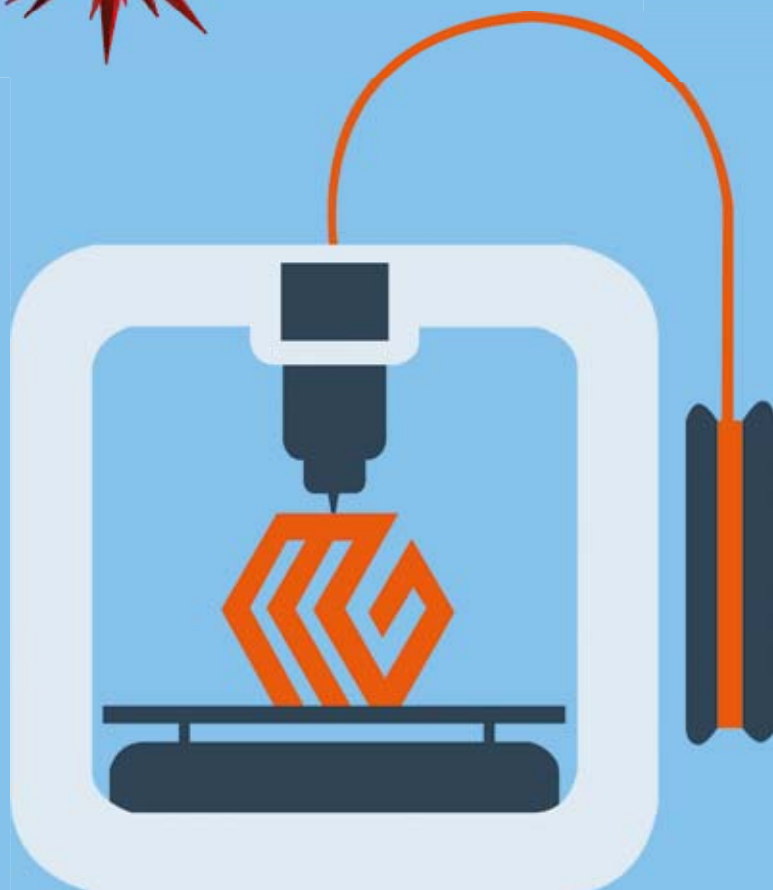
Il materiale va richiesto a:

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

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



Adesso il robot fa le scale

I robot bipedi sono costosi, complicati, fragili ed hanno un equilibrio precario, tuttavia sono interessanti perché rappresentano gli automi che meglio simulano la camminata degli umani. Le scale sono sempre state una sfida per i robot di tutti i tipi (bipedi, quadrupedi, robot cingolati e così via) per le abilità che richiedono. Ad esempio far salire e scendere dalle scale un robot bipede richiede il rilevamento di molti parametri da parte di sensori come accelerometri triassiali, molti calcoli e una velocità di elaborazione capace di correggere in tempo reale ogni errore. Questo, comunque, non assicura stabilità, tantomeno in presenza di imprevisti.

Ma l'approccio presentato alla conferenza Robotics Science and Systems dell'Oregon State University e da Agility Robotics permette di far camminare un robot bipede, chiamato Cassie, senza necessità di tutto ciò. Cassie non ha informazioni sulle scale che sta salendo o scendendo, ma riceve un feedback propriocettivo, il che significa che sa che tipo di contatto stanno facendo i suoi piedi con le scale e si adatta.

I ricercatori hanno utilizzato l'apprendimento per rinforzo per addestrare il robot su codici di costruzione tipici della città, con serie di scale fino a otto gradini. Per trasferire efficacemente le strategie apprese per salire le scale (politiche) dalla simulazione al mondo reale, la simulazione includeva una serie di disturbi progettati per rappresentare gli imprevisti, tra cui l'attrito del suolo. Cassie è stato volutamente addestrato su scale irregolari: una piccola quantità di rumore (± 1 cm) viene aggiunta all'alzata e alla corsa di ogni gradino in modo che le scale non siano mai uniformi, cosicché la politica deduca le dimensioni precise via proprioccezione e successivamente si adatti a esse.

www.automata.tech

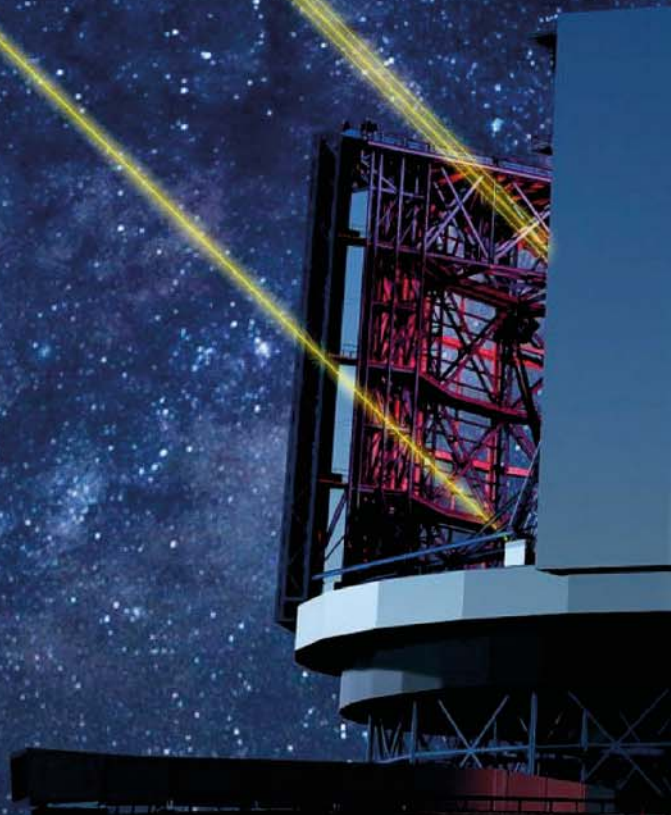
Laser per eliminare i rifiuti spaziali

Da un po' di tempo stiamo percependo l'effetto che il lancio di razzi, satelliti ecc. nello spazio ha provocato: l'esistenza di quella che viene chiamata "spazzatura spaziale" che orbita intorno alla Terra e che è composta da satelliti fuori servizio, manufatti dell'umanità e loro pezzi. Uniti ai detriti che naturalmente si trovano nello Spazio, hanno già danneggiato i satelliti o costretto la stazione spaziale a modificare la propria orbita. Ora i ricercatori in Australia vogliono usare i laser per tracciare le particelle ribelli e persino spingerle via prima che causino problemi.

Se lasciati nello spazio, tali detriti alla fine entreranno in collisione tra loro e si verificherà un effetto valanga in cui vengono creati più detriti e ad un certo punto ci sarebbero così tanti detriti a certe altitudini che non sarà più possibile compiere attività a tali quote. La NASA afferma che c'è più di mezzo

milione di detriti, da satelliti in disuso a macchie di vernice, che sfrecciano intorno alla Terra a velocità fin oltre 30.000 kmh.

La tecnica australiana consiste nell'applicare l'ottica adattiva, ampiamente utilizzata in astronomia, per il tracciamento dei frammenti più piccoli, di dimensioni comprese tra 1 e 10 cm, che possono essere difficili da vedere. L'ottica adattiva consente agli astronomi di superare la turbolenza atmosferica causata dai cambiamenti di pressione e temperatura che fa brillare le stelle. Per fare ciò, gli astronomi creano una stella guida laser sparando un raggio di luce giallo attraverso un telescopio. Il raggio colpisce uno strato di particelle di sodio in alto nell'atmosfera, al di sopra della turbolenza, e rimbalza indietro. I fotorilevatori misurano le distorsioni nel raggio riflesso per calcolare la quantità di turbolenza, quindi istruiscono una serie di attuatori su



uno specchio deformabile per spingere e sollevare diverse parti dello specchio per compensare la distorsione. Ciò provoca un notevole miglioramento della messa a fuoco.

Utilizzando il sistema si possono emettere serie di impulsi laser a infrarossi sui detriti in orbita per misurare dove si trovano e dove sono diretti. Senza l'ottica adattiva, non sarebbe possibile mettere a fuoco il laser a infrarossi abbastanza bene da trovare piccoli oggetti. Conoscendo il percorso dei detriti è possibile, ad esempio, avvertire un satellite per comunicazioni di alterare leggermente la sua orbita per evitare di essere colpito.

Ma il team di ricercatori vuole andare oltre e utilizzare un altro laser per spingere via alcuni detriti nel caso l'analisi rilevi dei minuscoli frammenti di spazzatura spaziale diretti verso un insediamento spaziale o verso un altro cumulo di detriti, il che potrebbe provocare detriti più difficili da rintracciare. La tecnica consiste nel dirigere un raggio laser a infrarossi da 20 kW su uno dei frammenti: la leggera pressione del laser basterebbe, dopo pochi impulsi, a spingere delicatamente la spazzatura spaziale su una traiettoria diversa, anche senza distruggerla.

Per distruggere la spazzatura spaziale si può spingerla in un'orbita più bassa, finché non cade nell'atmosfera e brucia come una meteora. Ma escludendo lo sviluppo di qualsiasi tipo di forza di trazione simile a un "raggio trattore", il laser dovrebbe probabilmente essere posizionato nello spazio, dove potrebbe spingere più facilmente i detriti verso il nostro pianeta.

L'idea è stata sviluppata presso lo Space Environment Research Center dell'ANU (Australian National University). Tutte le apparecchiature laser e ottiche adattive sono disponibili presso il Monte dell'università. Osservatorio di Stromlo fuori Canberra.

Naturalmente l'istituto vuole rassicurare chiunque si preoccupi che il laser utilizzato potrebbe colpire gli aeroplani nel cielo o distruggere costosi satelliti del Governo: esistono regolamenti per evitare di interferire con il cielo e il traffico orbitale e inoltre il laser non ha una potenza tale da distruggere aeromobili o satelliti. Serve solo a generare una forza capace di cambiare l'orbita di pochi centimetri alla volta e non a fare quel che vediamo in alcuni film di fantascienza.

www.anu.edu.au



Vaccinare con l'elettricità

I vaccini a mRNA sono entrati nella nostra vita perché rappresentano la difesa contro il dilagare del COVID-19, però richiedono una catena del freddo per conservarli e presentano significativi effetti collaterali. Alcuni di questi potrebbero essere superati passando ai più evoluti vaccini a DNA, che hanno i vantaggi dei vaccini a mRNA ma non causano effetti collaterali significativi e, soprattutto, non hanno bisogno di essere refrigerati. Potrebbero quindi essere somministrati più facilmente e con dispositivi automatici come quello rivoluzionario proposto da Inovio Pharmaceuticals, che utilizza una tecnica nota come elettroporazione, in cui un impulso elettrico applicato sulla pelle apre brevemente i canali nelle cellule per consentire l'ingresso del vaccino. Lo strumento di elettroporazione è essenziale per il vaccino Inovio, ma richiede persone che sappiano come usarlo. Però risolve il problema che ostacola la diffusione dei vaccini a DNA, ossia quello di far penetrare il DNA, che è una molecola grande, non solo attraverso gli strati esterni della cellula, ma anche attraverso la membrana nucleare. A differenza di un vaccino a mRNA, che può funzionare in parti della cellula al di fuori del nucleo, un vaccino a DNA deve penetrare il nucleo.

Tra tutti i metodi provati, l'elettroporazione è promettente, giacché i ricercatori l'hanno utilizzata per decenni per trasferire materiale genetico nelle cellule in laboratorio.

Il dispositivo Inovio si chiama Celectra 3PSP, funziona a batteria e può erogare circa un centinaio di dosi con una singola carica e ha una durata di circa 5.000 utilizzi. Ogni utilizzo richiede una punta usa e getta; gli elettrodi di circa 3 millimetri di lunghezza erogano una serie di quattro impulsi elettrici ad onda quadra che durano 42 millisecondi ciascuno, a 0,2 ampere.

ir.inovio.com



L'interfaccia cervello-computer batte il record di velocità di digitazione

Immaginare di scrivere a mano è più veloce che scrivere: lo dimostra la ricerca sulle interfacce BCI (Brain to Computer Interface) che consente all'uomo di far scrivere al computer ciò che pensa. I ricercatori della Stanford University hanno condotto lo studio su un uomo di 65

anni con una lesione al midollo spinale a cui era stato impiantato un array di elettrodi nel cervello. Sinora gli esperimenti di digitazione BCI erano basati sull'immaginare di spostare un cursore su una tastiera digitale per selezionare le lettere. Nel frattempo, gli elettrodi registrano l'attività cerebrale e gli algoritmi di apprendimento automatico decifrano i modelli associati a quei pensieri, traducendoli nelle parole digitate.



Il più veloce di questi precedenti esperimenti di digitazione cerebrale consentiva di digitare circa 40 caratteri, o 8

parole, al minuto. Nel più recente esperimento, la persona ha comunicato circa 90 caratteri (18 parole, al minuto)! I ricercatori di Stanford hanno compiuto l'impresa riproponendo un algoritmo di apprendimento automatico originariamente sviluppato per il riconoscimento vocale. L'algoritmo di deep learning, chiamato rete neurale ricorrente, si è addestrato in poche ore a riconoscere l'attività neurale del partecipante quando ha immaginato frasi scritte a mano.

nptl.stanford.edu/



Vento in poppa per la nave a Rinnovabili

Le navi trasportano, oltre alle persone, la gran parte delle merci nel mondo e sono quindi una rilevante fonte di inquinamento ambientale, se si considera che vanno ad olio combustibile, un derivato del petrolio poco raffinato. Per questo si lavora alla ricerca di propulsioni a zero emissioni come quella proposta da Eco Marine Power, chiamata Aquarius Marine Renewable Energy with EnergySail. Si tratta di un sistema integrato avanzato di vele rigide, pannelli solari marini, moduli di accumulo di energia, caricatori e computer, che consente di attingere all'energia eolica e fotovoltaica. La schiera di vele rigide viene orientata da un sistema informatico per adattarsi al meglio alle condizioni meteorologiche prevalenti e può essere ammainata e riposta quando non in uso o in caso di maltempo. Le vele rigide si basano sulla tecnologia EnergySail di EMP e possono essere utilizzate anche quando una nave è all'ancora o in porto. Ogni EnergySail può essere configurata con un mix di sensori, pannelli fotovoltaici o altri dispositivi di generazione di energia elettrica.

Il sistema ha ricevuto la certificazione AiP di ClassNK per i due sottosistemi principali: L'EnergySail: un sistema di propulsione assistita dal vento del tipo a vela rigida che include la struttura della vela, il sistema di guida e il sistema di controllo automatizzato.

Diversi armatori stanno collaborando con Eco Marine Power in termini di pianificazione delle prove in mare del sistema completo e di studi sull'uso di sistemi di energia rinnovabile a bordo delle navi.

www.ecomarinepower.com

Primo volo suborbitale per VSS Unity

Virgin Galactic ha completato il suo terzo volo spaziale, primo in assoluto dallo Spazioporto America nel Nuovo Messico: un passo fondamentale nella direzione dei voli turistici diretti nello Spazio e missione compiuta per Virgin Galactic, dopo il rinvio dello scorso dicembre causato dallo spegnimento del motore a causa di un problema di comunicazione rilevato dal computer dell'aereo madre e dopo l'ulteriore rinvio di febbraio. La navicella VSS Unity ha raggiunto una velocità di Mach 3 dopo essere stata rilasciata dall'aereo madre a doppia fusoliera VMS Eve, ed ha raggiunto lo spazio a un'altitudine di 55,45 miglia (circa 90 km) dove è rimasto per un po' prima di ruotare gli stabilizzatori

e puntare verso la Terra, dove è tornato planando fino alla pista di atterraggio allo spazioporto. Nella cabina di pilotaggio della VSS Unity c'erano CJ Sturckow e Dave Mackay ed il volo è servito a condurre esperimenti di ricerca scientifica come parte del Flight Opportunities Program della NASA e a raccogliere dati da utilizzare per gli ultimi due rapporti di verifica richiesti per ottenere la licenza di operatore di veicoli spaziali commerciali da parte della FAA. Ha inoltre permesso di testare gli stabilizzatori orizzontali e i controlli di volo, nonché l'immunità alle EMI che avevano causato il rinvio della missione.

www.virgingalactic.com



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

Purificatore d'aria a raggi ultravioletti

VERSIONE
DA PARETE

€179,00

Cod. MICROUVAIR-P

Alimentazione: 12 Vcc

Potenza assorbita: @ 3.6 W

Durata sorgente UV: @20000 ore

Capacità sterilizzazione: 60 m³/h

Dimensioni: altezza 25 cm x 10.5 cm x 7 cm

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

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

Rende gli ambienti batteriologicamente sani eliminando anche le spore.

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

Può funzionare continuativamente anche in presenza di persone.

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



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



VERSIONE
DA TAVOLO

€199,00

Cod. MICROUVAIR-T



Prezzi IVA inclusa.

SANIFICATORE UVC PER OGGETTI

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



€80,00

Cod. UVCSAN

