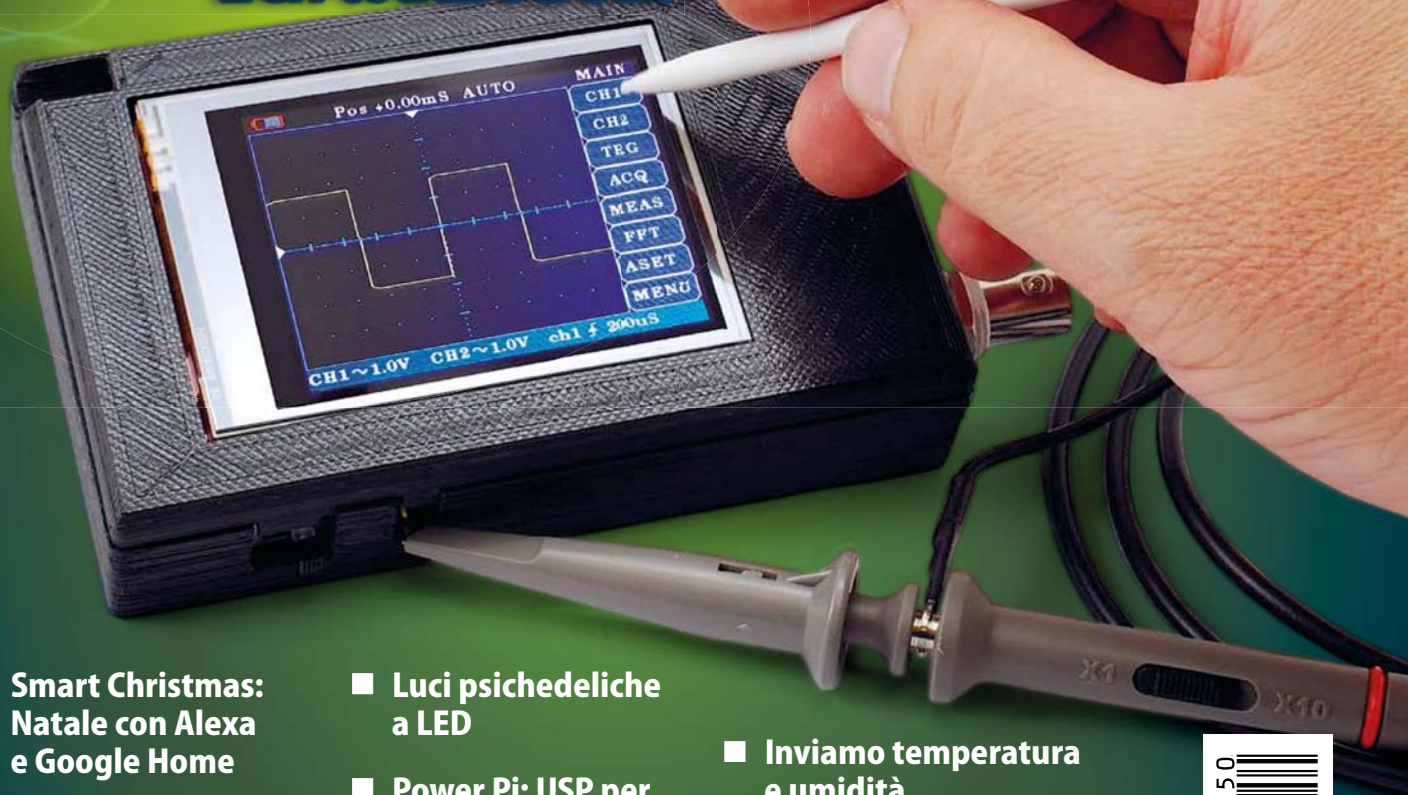


Elettronica In

www.elettronica.in.it

€ 6,00 - Anno XXVI - n. 250 - DICEMBRE 2020 / GENNAIO 2021

Oscilloscopio palmare 2CH touch screen



- Smart Christmas: Natale con Alexa e Google Home
- GPS Tracker
- Luci psichedeliche a LED
- Power Pi: USP per Raspberry Pi
- Inviamo temperatura e umidità via Bluetooth

Alberello di Natale SMD



Hai la mascherina? Raspberry Pi lo vede



LoRa™

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)



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

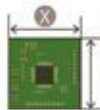
1 Layer

2 Layers

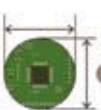
4 Layers

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.

250 volte innovativi

Siamo arrivati al numero 250: sono più di 25 anni che Elettronica In vi propone progetti e approfondimenti al passo con i tempi. In questi anni abbiamo seguito l'evolversi della tecnologia, abbiamo visto nascere insieme a voi schede come Arduino e Raspberry Pi, ma anche visto consolidarsi tecnologie come Sigfox, NB-IoT, LoRa. Abbiamo, mese dopo mese, cercato di proporvi soluzioni che potessero incontrare i vostri desideri e soddisfare la vostra sete di elettronica. Molte cose sono cambiate dalla prima pubblicazione e con il numero 250 introduciamo delle modifiche anche al piano editoriale.

La rubrica Rinnovabili si fonde con Scienze e Tecnologia, per dare maggior

spazio ai progetti pratici e ai tutorial, pur senza scordare di riportare le novità che la tecnologia applicata alle fonti energetiche rinnovabili

Sono più di 25 anni che Elettronica In vi propone progetti e approfondimenti al passo con i tempi

propone, Tecnologia per le rinnovabili, dunque.

La rubrica Q&A si trasforma in Tips&Tricks, dove non mancheremo di riportare quei "trucchetti" e "consigli" che vi permetteranno di realizzare i vostri progetti in maniera molto più semplice e veloce. Resta la possibilità di inviare le vostre domande alla Redazione o scriverle nel nostro forum (www.elettronica.in/webforum) dove è possibile ricevere risposte in tempi brevi e non vincolati alla periodicità della Rivista, il che è consono al fatto che viviamo nell'era dei Social Media, delle Chat e delle e-mail.

Il numero 250 segna anche l'avvicinamento alla Direzione della rivista. Da diversi anni do il mio contributo tecnico alla rivista e ho collaborato a stretto contatto con Arsenio Spadoni, imparando da lui il metodo per gestire una rivista tanto importante e tecnica come Elettronica In.

Permettetemi di utilizzare questo spazio per ringraziarlo pubblicamente per ciò che ha fatto in oltre un quarto di secolo, spronando me e tutti i collaboratori a dare il meglio per poter offrire a voi il meglio.

Con piacere prendo il timone di questo veliero che, mi auguro, solcherà il mare dell'elettronica per almeno altri 25 anni e vi invito subito ad aiutarmi, a consigliarmi ed anche a criticarmi (trovate qui sotto il mio indirizzo e-mail) per poter fare di Elettronica In la rivista che vorreste.

Buona lettura e Buone Feste!

ON LINE

elettronica.in

SI RINGRAZIA

Futura Elettronica
RM Elettronica



In copertina:
Un DSO portatile realizzato con un STM32, uno strumento indispensabile per un tecnico elettronico e non solo.


Boris Landoni
boris.landoni@elettronica.in

Elettronica In

Rivista mensile, anno XXVI n. 250
DICEMBRE 2020 / GENNAIO 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.

© 2020 Futura Group srl

Futura
Group
Editori

Contenuti

01

RUBRICHE

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

Progetto di Copertina

OSCILLOSCOPIO PALMARE 2CH TOUCH SCREEN



46

Raspberry Pi PowerPi

33



21

ARTICOLI

FISHINO

21 Natale con Alexa e Google Assistant

Gestiamo gli addobbi natalizi luminosi da remoto attraverso una scheda Fishino e Google Assistant o Alexa: chiediamo di accendere le luci dell'alberello e questo avverrà, da qualsiasi parte noi siamo.

RASPBERRY PI

33 PowerPi

Gruppo di continuità per la Raspberry Pi provvista di uscita USB capace di ricaricare batterie al litio e ai polimeri di litio. È provvista di display OLED e le impostazioni vengono caricate mediante USB.

Gadget Alberello SMD

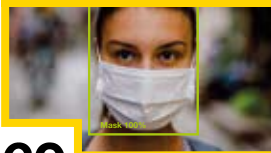
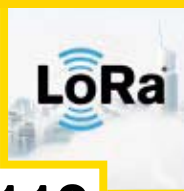


61

Fishino Natale con Alexa e Google Assistant



21

GPS-System**GPS Tracker****81****Visione****Hai la mascherina?
Raspberry Pi lo vede****69****Didattica****Corso LoRa****113****Sound & Light****Luci Psichedeliche a LED****94****Networking****Inviemo temperatura
e umidità via Bluetooth****105****STRUMENTAZIONE****46****Oscilloscopio palmare 2 CH
Touch-screen**

Costruiamo un valido strumento portatile per effettuare misure sul campo, come nel proprio laboratorio; compatto e di semplice utilizzo grazie al display tattile, si gestisce con un pennino.

GADGET**61****Alberello SMD**

Decorazione natalizia dal sapore amarcord di semplicissima costruzione e alla portata di tutti.

VISIONE**69****Hai la mascherina?
Raspberry Pi lo vede**

Individuare i volti all'interno di un video o di una ripresa live e riconoscere se indossano la mascherina: facile con il "lampo".

GPS-SYSTEM**81****GPS Tracker**

Realizziamo un localizzatore basato sul modulo SIM808 utilizzando, per l'invio dei dati di posizionamento, il servizio on-line ThingSpeak.

SOUND & LIGHT**94****Luci Psichedeliche a LED**

Centralina per luci psichedeliche a quattro canali rivisitata in chiave moderna con LED colorati ad alta efficienza al posto delle lampadine.

NETWORKING**105****Inviemo temperatura e umidità
via Bluetooth**

Interfacciamo via Bluetooth Arduino con uno smartphone per effettuare acquisizione e visualizzazione dati di temperatura e umidità.

DIDATTICA**113****Capire e usare LoRa**

Impariamo a utilizzare l'interfaccia di Node-RED per inviare i dati dal web al nostro dispositivo finale, sfruttando The Things Network e un gateway multicanale per implementare il downlink su LoRaWAN.

Costruisci e programma un ROBOT!



cod. KSR21
€ 42,00

Robot RICCIO - in kit

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



cod. KSR19
€ 35,00

Robot 5 in 1 con codifica meccanica - in kit

Robot didattico con codifica STEM (Science, Technology, Engineering, Mathematics). Ideale per principianti, permette di avvicinarsi all'ingegneria di costruzione delle macchine e della codifica manuale. Stimola i bambini a pensare e a sviluppare la loro capacità nella risoluzione dei problemi. Dispone di ruota di codifica con cui è possibile programmare i movimenti. Tanto maggiore saranno le capacità e l'esperienza dell'utente, quanto più interessanti e impegnativi diventeranno i movimenti del robot. Funziona con 2 batterie ministilo tipo AAA.

Robot TOBBIE II con MICRO:BIT in kit

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



cod. KSR20
€ 89,00



Robot TOBBIE II in kit

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

cod. KSR18
€ 37,00

Prezzi IVA inclusa

Simulare un circuito elettronico

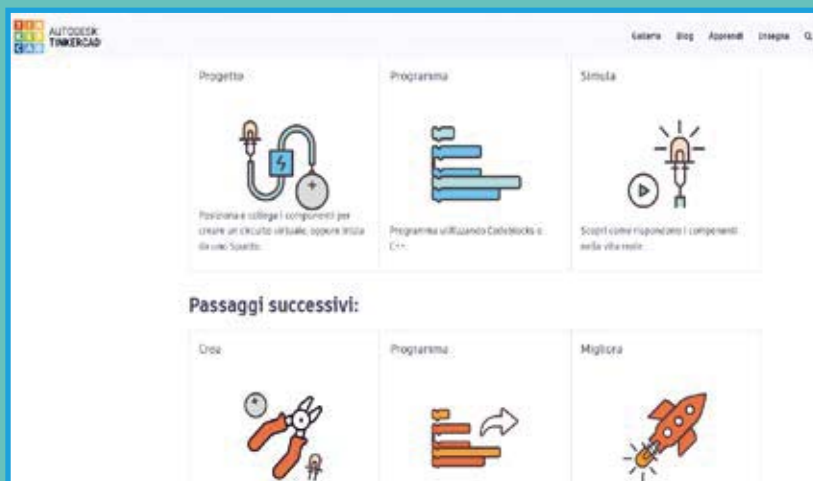
Anche quando conosciamo bene gli aspetti della progettazione e ci sentiamo sicuri che un circuito disegnato e progettato, sia esso basato su componenti digitali o su microcontrollore e quindi da programmare, funzionerà come atteso, in realtà la sorpresa è sempre dietro l'angolo. E sicuramente a nessuno piacerebbe trovarsi, dopo ore passate a disegnare il PCB, a inciderlo e forarlo, a montarne i componenti, ad accorgersi che qualcosa non va, che ha commesso un errore di distrazione o che ha scordato un elemento e deve rifare il circuito. In questo caso, poter disporre di strumenti di simulazione al computer aiuta molto, perché consente di realizzare il circuito virtualmente e di simularne il funzionamento. Tra questi strumenti segnaliamo quello reso disponibile su Internet da Autodesk e chiamato Tinkercad: lo trovate sul sito tinkercad.com/learn/circuits, la cui home-page (vedi immagine) riassume le funzionalità offerte che

Con questo numero cambia qualcosa in Elettronica In e, come qualche tempo fa avete visto la nostra "posta dei lettori" prendere la veste di Q&A (Question & Answers, ovvero domande e risposte), ora ci sembra doveroso fare un ulteriore passo avanti, sostituendo quell'appuntamento con questo: Tips & Tricks, vale a dire suggerimenti in pillole e piccoli trucchi che sperimentatori, tecnici e progettisti elettronici vorrebbero conoscere. Cercheremo di averne per tutti ogni mese e di non deludervi!

vanno dalla stesura del progetto (Progetto), all'eventuale creazione del programma (Programma), fino alla simulazione (Simula). Se la simulazione avrà dato esito positivo, Tinkercad mette a disposizione tre ulteriori tool per creare, programmare e ottimizzare (ingegnerizzare) il progetto.

Visualizzatore online di LCD

A volte scegliere il display da utilizzare in un progetto può risultare difficile, quantomeno se nella propria esperienza non si è provata una certa varietà di caratteri, colori, formati ecc. Che luce di retroilluminazione vogliamo? Che colore per i caratteri? Come viene un carattere o un simbolo grafico nel classico LCD a matrice di punti? E che dimensioni avere per ottenere l'effetto desiderato e la visibilità del caso? Sicuramente non è pensabile compere tanti display e provarli in pratica. Ecco quindi che il web ci viene in aiuto con un simulatore di display LCD, accessibile dal sito avtanski.net/projects/lcd. Qui troviamo un pannello di impostazione che permette di definire i parametri del display: in Display Type si sceglie il tipo (quindi matrice di punti ecc.) mentre in Text si scrive il testo o comunque i simboli e caratteri da visualizzare. Sotto c'è l'impostazione dello schema



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

di colori, riguardante lo sfondo (colore della retroilluminazione) e il colore dei caratteri. Infine in basso si possono definire le dimensioni del display, quelle dei caratteri, il tipo di cursore ecc. Impostato il tutto, cliccando sul pulsante Preview vedremo nella zona centrale dello schermo l'aspetto che avrà il tipo di visualizzatore selezionato; ogni volta che verrà fatta una modifica, l'aspetto andrà aggiornato con Preview. È anche possibile salvare sul proprio computer, sotto forma di immagine .PNG, la simulazione del display, cosa che è molto utile, ad esempio, per le illustrazioni di un manuale, una relazione per la scuola, delle istruzioni d'uso ecc.

Far aderire una stampa 3D al piatto

Uno dei problemi che più si riscontrano che più infastidiscono chi fa stampe 3D con materiali plastici quali l'ABS (specie quando si tratta di grandi oggetti che richiedono anche un giorno per essere completati), è la deformazione dei layer causata dal rapido raffreddamento del materiale deposto; un fenomeno che si

accentua con l'aumento della differenza di temperatura fra lo strato deposto e il piatto della stampante, quindi stampando con materiali ad alto punto di fusione o in locali non particolarmente caldi. Per mitigare questo inconveniente sono state messe in campo varie soluzioni, una delle quali consiste nel riscaldare il piatto di stampa mediante una serpentina alimentata elettricamente. Ma ce ne sono tante altre, che elenchiamo qui di seguito.

Nel caso di piatti di stampa in vetro (per esempio quelle delle 3D4040 della Futura Elettronica) il modo più semplice per migliorare l'aderenza del piano è mantenere pulito il vetro: a stampante spenta (e a piatto freddo, se di tipo riscaldato...), la pulizia può essere fatta con dell'alcool per rimuovere la polvere, la colla e le impronte delle dita.

Un'altra soluzione consiste nell'aggiungere un "brim" o un "raft": il primo è un singolo livello posizionato intorno al modello 3D e in grado di aumentare la superficie dell'oggetto che andrà ad aderire sul piatto. Il secondo costruisce una struttura rimovibile su tutta l'ampiezza del pezzo ed è utile se l'oggetto non è piatto, così da dargli una base d'appoggio uniforme. La maggior parte dei software di slicing consente di aggiungere "brim" o "raft": per esempio Cura Ultimaker.

La tradizionale lacca per capelli è una delle soluzioni più facili ed economiche per migliorare l'adesione al piatto di stampa: proteggendo con un foglio di giornale tutto quello che sta intorno al piatto, spruzzatela a circa 25 cm di distanza fino ad ottenere uno strato uniforme, lasciandolo poi asciugare. È possibile rimuovere la lacca semplicemente strofinando la superficie del piatto con un panno umido, perché le lacche

sono studiate per essere lavate via dai capelli.

Pellicola adesiva e nastro adesivo come ad esempio il popolare Blue Tape (o Blu Scotch), che va disposto a strisce ben accostate fino a coprire l'intera superficie d'appoggio. Dello stesso materiale sono fatti i fogli adesivi blu, anch'essi facilmente reperibili, nonché quelli Build Tak. Una soluzione molto utile stampando in ABS è l'applicazione al piatto di stampa di ABS liquido ottenuto lasciando sciogliere avanzzi di ABS in acetone all'interno di un barattolo in vetro o metallo, per qualche minuto, fino a quando l'ABS non si è completamente dissolto. Il composto può essere applicato usando un dischetto di cotone per struccare o un pennello, badando di non inalare l'aerosol durante l'applicazione, perché l'acetone è a dir poco inebriante e può anche portare allo svenimento.

Ultimo consiglio è quello che riguarda l'utilizzo e l'applicazione al piatto del nastro in Kapton, autoadesivo, che può rivelarsi molto utile e che va applicato come il Blue Tape.

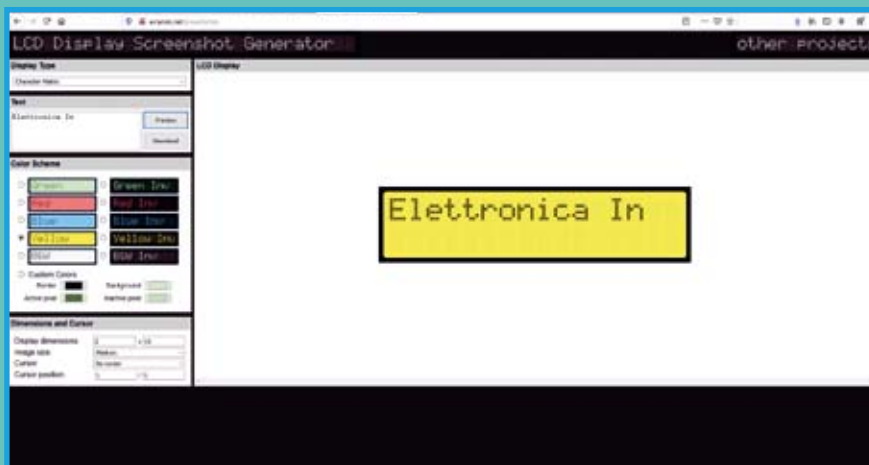
Gestire tanti LED con pochi pin

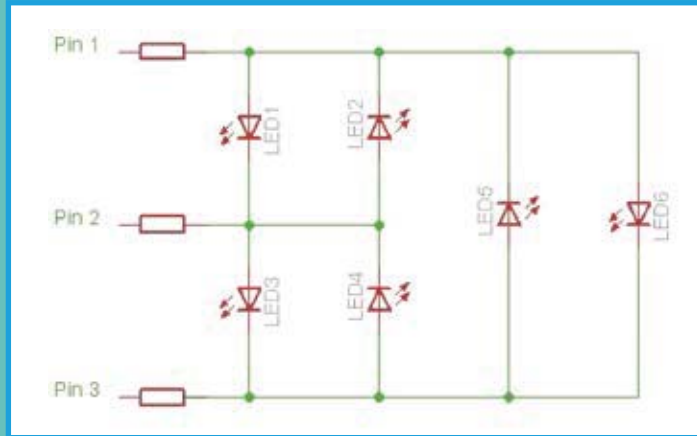
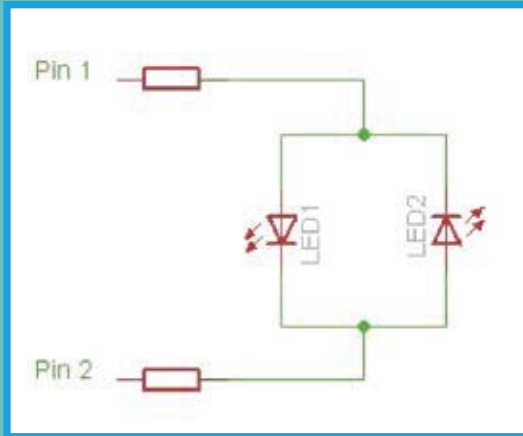
Una delle tante possibilità offerte dai LED, soprattutto dai minuscoli componenti in versione SMD, è quella di realizzare display a matrice di vario genere; il limite è che la quantità di punti è dettata dal numero di linee richieste per il pilotaggio di tutti i diodi, giacché anche ricorrendo a una matrice di righe e colonne servono sempre una linea per ogni riga e una per ogni colonna. Questo può essere un limite quando vogliamo sfruttare la più che sufficiente potenza di calcolo dei piccoli Arduino (il micro o il Nano, per esempio) per pilotare grandi quantità di LED.

Ma esiste una soluzione che incrementa la densità di diodi pilotabili, chiamata Charlieplexing, in grado di andare oltre le capacità della classica tecnica multiplexing. Si tratta di una soluzione circuitale proposta nel 1995 da Charlie Allen alla Maxim Integrated, capace di pilotare $(n * n) - n$ LED, dove n sta per il numero di pin del microcontrollore impiegato. Quindi con 10 piedini o linee di I/O si possono comandare:

$$(10 * 10) - 10 = 90$$

Questo un sistema che si basa sostanzialmente sul comando di LED connessi





in configurazioni a due a due in antiparallelo, a loro volta in serie e parallelo: ad esempio tre gruppi di antiparallelo, due in serie tra loro e il terzo tra gli estremi di questa serie.

Il tutto si pilota con appena tre linee. Il trucco sta nell'invertire e combinare opportunamente gli stati logici delle linee per sfruttare il fatto che i LED, come tutti i diodi, conducono in un verso soltanto, quindi pilotando un antiparallelo di due LED con due linee, quando una è a zero logico e l'altra ad uno si accende il primo LED, mentre invertendo la polarità si accende l'altro. In questo caso per accendere il LED 1 occorre scrivere le seguenti linee di codice:

```
digitalWrite(ledPin1, HIGH);
pinMode(ledPin2, INPUT);
digitalWrite(ledPin2, LOW);
```

Questo breve sketch farà in modo che inizialmente il LED1 si accenderà per effetto del livello logico alto sul pin 1 e basso sul pin 2; scrivendo il contrario, ossia ledPin1, LOW e ledPin2, HIGH si

illumina LD2 ed LD1 rimane spento. Aggiungendo un terzo I/O è possibile ampliare enormemente il numero di LED pilotabili con questa tecnica. Come vedete ora abbiamo a disposizione ben 6 LED con solo 3 PIN del nostro Arduino o qualsiasi altro microcontrollore.

Giocando quindi con la polarità di ogni singolo PIN e sfruttando la capacità dei LED di bloccare la corrente inversa è possibile accendere uno tra tutti i sei LED.

Per esempio per accendere il LED1 bisogna eseguire il seguente codice:

```
digitalWrite(ledPin1, HIGH);
pinMode(ledPin2, INPUT);
digitalWrite(ledPin2, LOW);
```

Se invece volessimo accendere il LED 6 bisogna eseguire le istruzioni:

```
digitalWrite(ledPin1, HIGH);
pinMode(ledPin3, INPUT);
digitalWrite(ledPin3, LOW);
```

In pratica per accendere LED1 e LED6 si pone pin 1 a 1 logico; il primo si accende con pin2 a zero e il secondo con pin3 a zero. Ponendo pin 1 a zero e pin2 a 1 si accende LD2, mentre con pin3 a 1 si accende LED5. Con pin2 a 1 e pin 3 a zero si accende LED3 e con pin3 a 1 e pin 2 a zero si accende LED4.

Naturalmente occorre attenzione nella stesura dello sketch, perché le istruzioni vanno eseguite in modo che non vengano a trovarsi attivi più I/O di quelli che servono per accendere ciascun LED alla volta. Ai vantaggi del Charlieplexing si contrappongono alcuni problemi da considerare, il primo dei quali è che aumentando il numero di pin e di conseguenza di LED, è necessario aumentare anche la velocità di calcolo del processore e di conseguenza anche il clock. Inoltre, all'aumentare del numero di LED si rallenta inevitabilmente il refresh rate dell'intera matrice, che non deve scendere sotto i 33,3 millisecondi per evitare che l'occhio percepisca la composizione; il rallentamento è dovuto alla complicazione del pilotaggio e al crescente numero di istruzioni da eseguire, che dipende dal numero di LED.



RM ELETTRONICA

Forniture per hobbisti, scuole e industria,
vendita di prodotti per la connettività in fibra ottica

Distributore autorizzato:

**FUTURA
ELETTRONICA**

Il tuo punto
di riferimento
per l'elettronica
a ROMA

Via Val Sillaro 38 • 00141 Roma • Tel: +39.068104753 • rmelettronica@libero.it

Nuove telecamere



Telecamera WiFi motorizzata

- Sensore d'immagine CMOS 2Mpx
- Pixel effettivi 1920x1080
- Ottica 3.6mm
- 4 LED IR
- 4 LED bianchi
- Portata LED 30 metri
- Funzione PAN/TILT con rotazione orizzontale di 320° e verticale di 90°
- Compressione video H.264/H.264+H.265X
- Microfono incorporato
- Speaker
- Funzione Motion Detection
- Supporto ONVIF
- Registrazione su SD-Card (da 8GB a 128GB)
- Potenza assorbita 6W max
- Alimentazione 12Vdc/2A (con adattatore di rete)
- Dimensioni 160x107x107 mm
- Peso 500 grammi

€ 79,00
cod. **FR769**



Prezzi IVA inclusa

Telecamera Wi-Fi con LED dual light

- Sensore d'immagine CMOS 2Mpx
- Pixel effettivi 1920x1080
- Ottica 3.6 mm
- 2 LED IR
- 4 LED bianchi
- Portata LED 30 metri
- Compressione video H.264/H.264+H.265X
- Microfono incorporato
- Speaker
- Funzione Motion Detection
- Supporto ONVIF
- Protocolli supportati RTSP, FTP, PPPOE, DHCP, DDNS, NTP, UPnP
- Registrazione su SD-Card (da 8GB a 128GB) 2.4835GHz
- Potenza assorbita 6W max
- Alimentazione 12Vdc/2A (con adattatore di rete)
- Dimensioni 230x130x115 mm
- Peso 400 grammi

€ 49,00
cod. **FR768**



Scopri la gamma completa
su www.futurashop.it



Telecamera WiFi IP Dome Fisheye 360° con IR e sensore di movimento

- Sensore d'immagine CMOS 2Mpx
- Sistema PAL/NTSC
- Pixel effettivi 1920x1080
- Portata 10 m
- Registrazione su SD-Card (max 128GB)
- Alimentazione 12Vdc/1A
- Potenza assorbita 5W max
- Dimensioni: 15,2x16,5x4 cm
- Peso 250 grammi

€ 42,00
cod. **FR766**

Telecamera WiFi da 2MPX, IR, con pannello solare e batteria ricaricabile

- Sensore d'immagine FHD 1080P CMOS 1/2.7 PS5230
- Risoluzione Video 1920x1080/15 fps
- Portata LED IR 10 metri
- Alimentazione con 2 batterie al litio tipo 18650 ricaricate tramite pannello solare integrato
- Registrazione su SD-Card (da 4GB a 64GB)
- Grado di protezione IP67

€ 119,00
cod. **FR767N**



Telecamera WiFi, PIR motion con memorizzazione dati su SD Card e batteria ricaricabile



€ 98,00
cod. **FR789**



- Sensore d'immagine CMOS 2Mpx
- Risoluzione Video 1920x1080/25 fps
- Angolo di ripresa 100° wide-angle
- Portata LED IR 6 metri
- Ottica 3.6mm
- Sensibilità 0,1 LUX
- Registrazione su SD-Card (max 64GB)
- Alimentazione 2 batterie 18650 3,7V 3000Mah (incluse)
- Grado di protezione IP66
- Peso 550g
- Dimensioni 95x53x80 mm

› Renesas alimenterà il futuro 5G

Renesas Electronics ha presentato la sua seconda generazione di controllori digitali multi-fase e 6 stadi e Smart Power Stage per le infrastrutture di rete dell'IoT.

Questi dispositivi nascono per soddisfare le crescenti necessità di alimentazione delle CPU, FPGA, GPU e degli ASIC per Intelligenza Artificiale utilizzati nei sistemi di infrastruttura per IoT. I controller multifase digitali ISL6822x, ISL6823x, RAA2282xx e gli "Smart Power Stage" ISL993xx, RAA2213xx risolvono le problematiche legate al rapido aumento della densità di alimentazione nei server dei data center, nell'archiviazione, nei link ottici, nei router, negli switch, nonché



nell'infrastruttura informatica e nell'infrastruttura wireless 5G.

A supporto dello sviluppo di stadi basati sui controller multifase, Renesas mette a disposizione il software PowerNavigator, che fornisce flessibilità, configurabilità e strumenti di debug in un'unica interfaccia utente grafica.

Con questa nuova soluzione i progettisti possono facilmente configurare il controller, ottimizzare le prestazioni impostando i diversi parametri, eseguire il completo monitoraggio grazie alla registrazione ad alta velocità dei comandi AVS e PMBUS, ma anche eseguire un approfondito debug grazie alla funzionalità Black Box.

www.renesas.com

› Maxim porta l'Intelligenza Artificiale nell'IoT

Intelligenza artificiale nei sistemi IoT embedded alimentati a batteria: questo è stato reso possibile da Maxim Integrated, che ha sviluppato allo scopo il proprio MAX78000 si tratta di un microcontrollore a basso assorbimento dotato di acceleratore di rete neurale che consente l'implementazione di algoritmi di intelligen-

za artificiale (AI) senza compromettere le prestazioni nei dispositivi per Internet of Things alimentati a batteria. Eseguendo inferenze di intelligenza artificiale con meno di 1/100 dell'energia richiesta dall'equivalente soluzione software, permette di migliorare drasticamente il comportamento real-time delle applicazioni

di Intelligenza artificiale alimentate a batteria. Peraltro i miglioramenti ottenuti a livello di assorbimento non compromettono la latenza o il costo. Il cuore del MAX78000 è rappresentato da un hardware specifico progettato per ridurre al minimo il consumo di energia e la latenza delle Reti Neurali Convolutionali (CNN).

Questo hardware funziona con un intervento minimo da parte del microcontrollore. Energia e tempo macchina sono utilizzati solo per eseguire le operazioni matematiche che implementano una CNN. Per trasferire in modo efficiente i dati dal mondo esterno al motore CNN, l'utente può utilizzare uno dei due core integrati nel microcontrollore: il core Arm Cortex-M4 ultra-low power (a bassissimo consumo) oppure il core RISC-V con consumi ancora più bassi. Il microcontrollore integra anche una prestante FPU con clock di 100 MHz.

www.maximintegrated.com

MAXIM INTEGRATED'S NEURAL NETWORK ACCELERATOR
SoC Enables Artificial Intelligence in Battery-Powered Devices



AI inferences
at less than 1/100th
the energy of
other embedded
solutions

 maxim
integrated.

› Sicurezza industriale? Ci pensa l'IoT

Nasce la protezione antiurto "intelligente" IoT, in grado di comunicare costantemente, tramite un'apposita App, la propria posizione all'interno dell'insediamento produttivo, lo stato di efficienza e il livello di gravità di eventuali impatti subiti.

Il sistema è stato realizzato dalla Stommpy, si chiama Smart e nasce per proteggere smart-panel e in generale pannelli di controllo fissi e mobili, ma la sua prerogativa è che non si tratta di una protezione passiva: tramite connessione wireless e sensori a bordo, può fornire anche indicazioni sulla sua condizione, quindi segnalare guasti causati, ad esempio, da carrelli in movimento nell'azienda, cadute e incidenti vari,

indicando anche il grado di efficienza residua e l'azione da svolgere in termini di manutenzione. Il tutto è ottenuto mediante un sensore inserito in una capsula a tenuta stagna.

La protezione antiurto Smart è in grado, in caso di urto, di registrare e di trasmettere in real-time la variazione di posizione assunta. Grazie all'interfaccia semplice ed intuitiva, l'Utente potrà visionare live, da remoto, direttamente da Smartphone, Tablet o PC numerose informazioni: la protezione antiurto che è stata colpita (grazie alla sua precisa mappatura riportata sul layout dello stabilimento del cliente), l'entità del danno eventualmente subito e lo stato di effi-

› Intel porta l'Intelligenza Artificiale sul satellite

PhiSat-1, un satellite che oggi si trova in orbita eliosincrona a 530 km di altitudine e a una velocità di oltre 27.500 km/h, è il primo ad essere gestito dall'Intelligenza Artificiale. Infatti è equipaggiato con una nuova camera termica iperspettrale dove l'elaborazione dei dati è affidata all'Intelligenza Artificiale grazie alla VPU (Vision Processing Unit) Intel Movidius Myriad 2. PhiSat-1 fa parte di una coppia di satelliti la cui missione è monitorare i ghiacci polari e l'umidità del suolo, oltre che testare i sistemi di comunicazione intersatellite in vista della creazione di una rete di satelliti

confederati. L'utilità, dell'Intelligenza Artificiale secondo l'ESA si può spiegare con un esempio: la capacità dei sensori di produrre dati aumenta molto più rapidamente di quanto i sistemi a Terra siano in grado di scaricare, per ragioni legate alla banda RF utilizzata; nel caso delle immagini, in ogni momento circa 2/3 della superficie della Terra sono coperti da nuvole e se lo scopo è riprendere la superficie terrestre, molte delle fotografie scattate sono inutili, ma ciò si scopre dopo che sono state salvate nella memoria di bordo e trasmesse a Terra, dove sono valutate, ore o giorni

dopo, da una persona o da un algoritmo. Questo comporta uno spreco di risorse che, potendo identificare e scartare le foto "annuvolate" a bordo del satellite eviterebbe l'occupazione di storage per il salvataggio e di banda per la trasmissione.

Questo compito viene svolto dal processore Intel del satellite grazie all'AI, che permette di identificare e scartare le foto disturbate dalle nuvole, risparmiando circa il 30% di banda.

La tecnologia per l'Intelligenza Artificiale di PhiSat-1 è stata sviluppata e testata da Ubotica, in collaborazione con Cosine (produttore della fotocamera di bordo) con l'Università di Pisa e con Singergise (specializzata in sistemi geospaziali). Una particolarità del progetto è che il processore Intel non è stato progettato per l'utilizzo nello Spazio né certificato "a prova di radiazioni cosmiche" quindi è stata necessaria un'attività di "caratterizzazione delle radiazioni" per addestrare l'algoritmo di AI a gestire eventuali errori di calcolo. Intel ha fornito supporto per abilitare le funzioni di intelligenza artificiale di PhiSat-1 utilizzando la tecnologia CVAI e c'è voluto circa un mese dalla messa in orbita per verificare la correttezza delle decisioni prese dall'AI.

ubotica.com



cienza del dispositivo, così da poter attuare le misure di ripristino e manutenzione del caso. Per rendere ancora più agevole ed immediata la verifica dello stato di salute di ciascuna protezione, è stato implementato l'utilizzo di un codice dei colori: il verde ne indica l'efficienza al 100%; il giallo richiede manutenzione ordinaria a causa di un'efficienza parziale e il rosso evidenzia l'inefficienza della protezione antiurto che deve essere subito gestita, mettendo in quarantena l'area fino al suo completo ripristino. La app consente di visualizzare anche le mappe dei locali dell'azienda in cui sono installate le protezioni antiurto, fornendo una visibilità completa ed immediata sullo stato di sicurezza dell'intero gruppo.

www.stommpy.it



› Schurter e U-Blox insieme per la connettività



Schurter, azienda nota per i suoi componenti elettromeccanici e per le tecnologie di alimentazione, ha iniziato una collaborazione con U-Blox (produttore di ottimi moduli GPS e per la connettività wireless) per sviluppare prodotti destinati ai distributori automatici e agli elettrodomestici del futuro. La combinazione di tre tecnologie wireless basilari (connettività cellulare, posizionamento basato su rete cellula-

re e Near Field Communication) a bordo dei moduli U-Blox consentirà lo sviluppo di soluzioni funzionali, complete, sicure e di lunga durata.

I moduli GPS U-blox offrono vantaggi essenziali grazie al supporto globale di cui godono ed alla connettività che li rende adatti alle applicazioni Internet of Things.

I vantaggi portati dall'adozione della LTE Cat M1 e della NB-IoT (LTE Cat

NB1) implementati nei moduli si concretizzano in un ampio ecosistema di prodotti e soluzioni, nel ridotto consumo energetico, nella buona copertura e nella possibilità di realizzare moltissime applicazioni dove la pur limitata velocità di trasferimento dei dati è sufficiente (ricordiamo che l'IoT mediante contempla trasmissioni poco frequenti di piccole quantità di dati).

I moduli si prestano alla realizzazione di sistemi di interfaccia uomo-macchina (HMI) e supportano la configurazione wireless tramite Bluetooth ed eventuali connessioni alla rete Wi-Fi esistente nell'edificio dove sono installati le macchine per il vending o gli elettrodomestici smart.

Con i moduli LARA e TOBY, U-Blox offre moduli cellulari adatti alle applicazioni di vending e alle esigenze di streaming, garantendo la protezione nelle applicazioni IoT end-to-end.

Il portafoglio di prodotti basati su host per applicazioni in esecuzione su un processore host è adatto anche per configurazioni Linux con EMMY-W1, ELLA-W1, LILY-W e JODY.

www.schurter.com/it



IMPARA L'ELETTRONICA FACILMENTE CON TOTEM MINI LAB



€ 108,00
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.

€ 34,00
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

HoriZone RA: trasferire in dati dei sensori IoT su Azure

Avnet Silica propone il kit di sviluppo HoriZone RA, creato per applicazioni Internet of Things (IoT) edge-to-cloud "proof-of-concept" che richiedono comunicazioni sicure. Basato sulla famiglia di microcontrollori Renesas Advanced per endpoint IoT e dispositivi edge, offre funzionalità per soluzioni Resource Constrained Embedded Systems connesse alla piattaforma cloud IoT Connect di Avnet basata sul Cloud Microsoft Azure. La soluzione include



sensori di umidità, temperatura, luce ambientale, qualità dell'aria e pressione atmosferica, oltre a microfono, accelerometro digitale, giroscopio digitale e sensore induttivo di prossimità per

il rilevamento della posizione. Integra componenti Renesas, incluso il microcontrollore RA6M3 (adatto alle applicazioni IoT che richiedono TFT, Ethernet, RAM embedded e USB high speed), il voltage supervisor ISL88014 e il DC/DC ISL80102 per la gestione dell'alimentazione. Avnet Silica ospiterà due webinar di presentazione del kit focalizzati sull'integrazione di funzioni di sicurezza di fascia alta, a beneficio dei progettisti.

www.avnet.com

MLX80302: un nuovo controller per motori BLDC trifase



-40°C e +150°C. Equipaggiato con driver da 0,25 ohm, il MLX80302 è un potente microcontrollore monolitico in package SOIC8 che supporta tutte le funzioni richieste dall'applicazione, aggiungendo una protezione dall'inversione di polarità e un singolo condensatore di disaccoppiamento. È dotato di funzioni di protezione complete e integrate che comprendono il blocco del rotore, il rilevamento delle sovratemperature e delle sovratensioni, oltre alla protezione ai cortocircuiti e la limitazione della corrente dei driver. Il tempo di avvio del controllo della pompa è di appena 50 ms sia in presenza di carico che a vuoto.

Il chip MLX80302 è programmabile via I²C-Bus e tutte le opzioni sono configurabili nella EEPROM di cui è dotato; è possibile selezionare diverse opzioni di ingresso PWM oltre a configurare la procedura di avviamento del motore per raggiungere in meno di 60 ms la velocità massima prevista per le pompe del carburante in pressione. Il controller MLX80302 è disponibile in package SOIC8 exposed pad. Sono anche disponibili le schede applicative plug-and-play e le schede di valutazione per la configurazione tramite bus I²C.

www.melexis.com

Melexis annuncia l'introduzione del suo nuovo dispositivo dedicato al controllo delle pompe carburante delle moto, che rappresenta un significativo progresso in termini di potenza controllabile e contenimento dei costi del sistema. Questo nuovo device consentirà ai produttori di rispettare le normative ambientali più severe, compresi gli standard di emissione BS-6 in vigore in India. Il MLX80302 è un controllore per motori BLDC (brushless in continua) trifase capace di erogare una corrente massima di 2,6 A.

È il dispositivo ideale per tutte le applicazioni legate al controllo delle pompe di alimentazione dei motori di autovetture e motocicli fino a 20 W con una tensione di batteria di 13 V, ma anche per una vasta gamma di elettrodomestici e pompe industriali con potenze fino a 40 W e alimentazione regolata a 24 Vcc.

Il controllore può operare in un intervallo di temperature fra

Sensore di profondità 3D ToF con portata fino a 10 metri e alta risoluzione

Infineon e pmdtechnologies hanno sviluppato un sensore di profondità 3D basato sulla tecnologia Time of Flight (ToF) REAL3 ToF. Il sensore consente una nuova generazione di applicazioni, mira a creare esperienze AR più coinvolgenti e intelligenti, nonché migliori risultati fotografici con un autofocus più veloce in condizioni di scarsa illuminazione o ritratti in modalità notturna. Il chip consente l'integrazione in moduli telecamera miniaturizzati, misurando accuratamente la profondità a breve e lungo raggio per la realtà aumentata, raggiungendo il 40% di risparmio energetico sull'imager. Grazie alla configurabilità flessibile, il sensore consente prestazioni della fotocamera differenziate in un'ampia varietà di inter-

valli, condizioni di luce utilizzi, ottimizzando la durata della batteria nei dispositivi mobili. Il sensore fornisce tecniche come la realtà aumentata in tempo reale, la scansione a lungo raggio, la ricostruzione di piccoli oggetti, l'autofocus veloce a bassa potenza e la segmentazione delle immagini. Effetti come la sfocatura dello sfondo nei video e nelle immagini delle scene in movimento sono facilmente attivabili senza la necessità di elaborazione post-fotografia e indipendentemente dalle condizioni di luce ambientale. Il supporto alla realtà aumentata consente l'acquisizione di dati di profondità 3D di alta qualità fino a una distanza di 10 metri, senza perdere la risoluzione.

www.infineon.com



Test molecolari rapidi ed economici per COVID-19



Alifax, produttrice di strumentazione diagnostica clinica, si avvale della tecnologia STM nello sviluppo di una soluzione portatile, rapida ed economica, per la diagnostica molecolare COVID-19 che utilizza la tecnica PCR (Polymerase Chain Reaction – reazione a catena della polimerasi) per amplifi-

care e quantificare il DNA (e l'RNA) contenuto nei campioni di pazienti. L'identificazione e il riconoscimento di specifici "codici" RNA e DNA permettono di rilevare gli agenti patogeni a cui appartengono.

Collegato ad un PC, il Molecular Mouse utilizza reagenti diagnostici per gestire il controllo

e il test di diversi target o campioni su una cartuccia monouso MEMS contenente 6 minuscole camere di reazione, ognuna delle quali opera il riscaldamento e il raffreddamento dei reagenti e dei campioni per amplificare il materiale genetico ricercato nel campione.

www.alifax.com/home-it-it

RA6T1, nuova famiglia MCU Renesas per controllo motore e manutenzione predittiva endpoint basata su AI

Renesas Electronics estende l'offerta di microcontrollori per controllo motore. Dotati di un ricco set di funzioni periferiche e rilevamento dei guasti basato su AI, i quattro nuovi micro della famiglia RA6T1 basati su core ARM sono progettati per il controllo motore in elettrodomestici, HVAC, inverter solari e AC drives. Man mano che gli elettrodomestici e i sistemi per l'automazione industriale e degli edifici diventano più intelligenti e complessi, i produttori si scontrano con l'aumento dei costi dei componenti per supportare le richieste di prestazioni del motore. Gli MCU RA6T1 combinano le prestazioni e la flessibilità della famiglia RA basata su ARM con l'esperienza di Renesas nel controllo motore. Inoltre Renesas integra le piattaforme supportate da TensorFlow Lite di Google con il controllo motore RA6T1 e la soluzione di manutenzione predittiva. Infatti l'intelligenza artificiale e l'apprendimento



automatico stanno portando la manutenzione predittiva verso la cosiddetta Manutenzione 4.0.

Basati sul core Arm Cortex-M4, i nuovi MCU a 32 bit RA6T1 funzionano a 120 MHz e integrano una FPU, 64 kB di RAM e da 256 a 512 KB di Flash, timer PWM a 32-bit con sette tipi di modalità PWM complementari per la generazione della portante, periodo di campionamento di 250 µs, ADC 12-bit high-speed e funzione di sample/hold che permettono acquisizione simultanea di 3 correnti di shunt. Le funzioni periferiche integrate

unite ad analogica ad alta velocità permettono un efficace controllo motore con pochi componenti: ad esempio, un singolo MCU RA6T1 può controllare contemporaneamente due motori. Inoltre, il framework Google TensorFlow Lite Micro per applicazioni TinyML aggiunge un rilevamento avanzato dei guasti agli MCU RA6T1, offrendo un sistema intelligente, facile da usare ed economico per la manutenzione predittiva su motori sensorless. A supporto della famiglia RA6T1 Renesas mette a disposizione il Renesas Solution Starter Kit (RSSK) che offre un facile debug del controllo motore, eseguendo analisi e sintonizzazione in tempo reale. Il kit include una scheda CPU RA6T1 e una scheda inverter compatibile 48 V, insieme a un'interfaccia grafica per workbench motore e un programma di esempio di controllo vettoriale sensorless con metodo a tre shunt.

www.renesas.com

Nuovi pulsanti piezo IP 67 da Schurter



L'uso di interruttori in ambienti umidi non riguarda solo la tecnologia, ma deve anche tenere conto del livello di isolamento e della connessione a grande distanza. Schurter soddisfa questo requisito con la famiglia di interruttori piezoelettrici pronti per l'installazione PSE, dotati di cavo pre-assemblato in poliuretano ben sigillato, che of-

fre la massima flessibilità e la massima protezione IP (IP67) durante l'installazione. Impermeabile dalla superficie dell'interruttore alla connessione, il cavo è lungo 15 metri.

Tutti gli interruttori piezoelettrici PSE sono realizzati in acciaio inossidabile 1.4462 ed offrono la massima affidabilità e una lunga durata (20 milioni di

azionamenti). Gli alloggiamenti sigillati ermeticamente rendono il PSE con cavo predestinato all'utilizzo in ambienti particolarmente difficili e inospitali.

Il pulsante di questi interruttori attivi è illuminato ad anello da LED RGB con 7 possibili colori. La tensione di alimentazione è tra 5 e 28 Vcc.

www.schurter.ch

Arriva il primo modulo Direct ToF a 64 zone, prodotto da ST

STMicroelectronics ha presentato VL53L5, il primo sensore Time-of-Flight (ToF) Flight Sense al mondo che suddivide la scena in 64 aree per aiutare il sistema di imaging a rappresentare spazialmente la scena col maggiore dettaglio possibile. Questo prodotto, unico nel suo genere, comprende una sorgente luminosa VCSEL (Vertical Cavity Surface Emission Laser) da 940 nm, un sensore System-on-Chip che integra un driver VCSEL, l'array di ricezione con tecnologia Single Photon Avalanche Diodes (SPADs) ed un core MCU a 32 bit con acceleratore e con un sofisticato firmware. Il VL53L5 mantiene la certificazione di Classe 1 di tutti i sensori Flight Sense. Alloggiato in un package da 6,4x3,0x1,5 mm, il VL53L5 ToF contiene elementi ottici nel foro di entrata che creano 64 zone di rilevamento.



Il VL53L5 FLIGHT Sense Time-of-Flight utilizza l'avanzato processo di produzione SPAD a 40 nm (su wafer da 12") per offrire una eccezionale portata di 4 metri. Il VL53L5 trova applicazione ad esempio nel rilevamento di presenza per dispositivi mobile e nell'auto-focus laser delle fotocamere degli smartphone. Integra entrambe le lenti di trasmissione e ricezione ed espande il campo visivo (FoV) a 61 gradi in diagonale, caratteristica che consente di rilevare oggetti fuori dall'inquadratura e migliorare la messa a fuoco. www.st.com

MOSFET a bassa Rdson da Toshiba

Toshiba Electronics ha presentato il nuovo MOSFET a canale N a basso consumo SSM6N951L in configurazione a drain comune da 12 V, destinato al circuito di protezione delle batterie incorporato nei pacchi batterie agli ioni di litio. La bassissima resistenza di ON (4,6 mohm massimi con una Vgs di 3,8 V) e i valori estremamente ridotti della corrente di dispersione gate-source (1µA massimi con una VGS di 8V) consentono all'SSM6N951L di avere una bassissima resistenza termica, cosa particolarmente importante durante i cicli di carica/scarica dei pacchi batterie, poiché consente di sviluppare soluzioni circuitali ad elevata densità di potenza che supportano velocità



di ricarica maggiori, oltre ad assicurare un'affidabilità molto più elevata e una durata operativa più lunga. Questi nuovi MOSFET sono disponibili in package con formato TCSP6A-172101 a basso profilo (con dimensioni di 2,14 mm x 1,67 mm x 0,11 mm). Di conseguenza, essi sono ottimizzati per l'applicazione nelle moderne apparecchiature alimentate a batterie soggette a vincoli di spazio. toshiba.semicon-storage.com

HiFive Unmatched: la board per testare e creare in modo nativo codice RISC-V

SiFive presenta la nuova piattaforma HiFive che consente agli sviluppatori di creare il software basato su RISC-V. Dai sistemi operativi in tempo reale alle distribuzioni Linux personalizzate, ai compilatori, alle librerie e alle applicazioni che accompagnano la progettazione del prodotto, gli sviluppatori possono utilizzare HiFive Unmatched per testare e creare in modo nativo codice RISC-V. La scheda HiFive è basata sul SoC SiFive FU740, un processore eterogeneo e coerente a cinque core (quattro SiFive U74 e uno SiFive S7). La serie U di SiFive comprende i core del processore applicativo a 64 bit compatibili con Linux, basati sull'ISA RISC-V mentre i core SiFive S-Series sono core embedded real-time a 64 bit progettati per consentire l'elaborazione deterministica mission-critical. Il complesso multicore eterogeneo integrato nel SiFive FU740 utilizzando la tecnologia SiFive Mix + Match offre una potente combinazione di applicazione ed elaborazione in tempo reale.

La scheda è la prima del genere ad avere connettori standard per alimentatori ATX, espansione PCI-Express, Gigabit Ethernet e porte USB come una motherboard da PC, pur essendo un sistema di sviluppo RISC-V single-board. HiFive Unmatched include 8 GB di memoria DDR4, saldata sulla scheda, e 32 MB di memoria Flash SPI.

Come espansione, dispone di quattro porte USB 3.2 Gen 1 (oltre a una porta di connessione console MicroUSB), una porta Gigabit Ethernet, uno slot di espansione PCI Express x16 (con connessione elettrica x8) e due slot M.2, uno per l'archiviazione NVMe e l'altro con chiave per adattatori Wi-Fi/Bluetooth. C'è anche uno slot di espansione MicroSD sulla scheda e l'alimentazione utilizza un ATX standard a 24 pin. Il sistema viene fornito con una scheda SD che include Yocto e OpenEmbedded (OE) Linux nonché altri pacchetti di sviluppo, con aggiornamenti disponibili per il download da SiFive.com.

www.sifive.com



Scopri il mondo di ARDUINO e le sue infinite potenzialità!

Progetta, costruisci e divertiti con i moduli per Arduino predisposti per essere inseriti sui mattoncini della LEGO®. Puoi utilizzarli per robotica, con Arduino o Raspberry Pi, ecc. Nella confezione sono comprese, oltre ai sensori e agli accessori, anche una board Arduino Uno compatibile e una scheda Shield sperimentale dove puoi collegare in tutta semplicità i moduli sensori e attuatori. Una dispensa a corredo vi accompagna con gli sketch per testare i vari moduli con Arduino e la Sensor Shield.

*29 MODULI PER ARDUINO
COMPATIBILI CON LEGO®*

€ 79,00

Cod. ARDULEGOKIT

I MODULI PRESENTI NELLA CONFEZIONE:

- Modulo con display 7 segmenti a 4 cifre
- Modulo con sensore di fiamma
- Modulo con LED RGB 10 mm
- Modulo sensore di temperatura e umidità
- Modulo touch sensor
- Scheda sensore umidità terreno
- Scheda sensore livello liquidi
- Modulo con LED rosso
- Modulo con LED verde
- Modulo con LED giallo
- Modulo con sensore effetto di Hall
- Modulo con LED RGB SMD
- Modulo con sensore luce ambiente
- Modulo con pulsante rosso
- Modulo con pulsante giallo
- Modulo con interruttore a pulsante
- Modulo con buzzer senza elettronica
- Modulo sensore di vibrazione
- Modulo con sensore di temperatura
- Modulo con potenziometro
- Modulo con fotoresistenza
- Modulo con interruttore reed
- Modulo con ricevitore IR a 3 pin
- Modulo con buzzer con elettronica
- Modulo con encoder rotativo
- Modulo sensore scala di grigi
- Modulo con microswitch

Prezzi IVA inclusa



Starter kit V6 con Arduino Uno REV3



**IL PIÙ
COMPLETO!**

€ 65,00

Cod. ARDUKITV6

Acquista anche il kit
completo con tutti i
componenti necessari per
realizzare gli esperimenti
descritti nel libro "l'ABC
di ARDUINO".

GLI APPUNTAMENTI DI DICEMBRE

Come sappiamo, molte manifestazioni fieristiche sono state sospese in tutto il mondo e rinviate o annullate. In questo periodo si avvicinano conferme e smentite di eventi a una tale velocità, che ci è davvero difficile poter pubblicare un prospetto che si sviluppi in un arco temporale come quello del numero doppio; vi aggiorneremo quindi costantemente sulle novità ed eventuali riprogrammazioni degli eventi attraverso il sito www.elettronicaipro.it, che vi invitiamo a visitare alla pagina **EVENTI > WEBINAR > Calendario**.

Di seguito riportiamo due eventi di formazione on-line per gli addetti ai lavori.

> 1-15 DICEMBRE 2020

SENSING DISCOVERY DAYS

AVNET[®] SILICA

Il mondo reale è analogico e per acquisirne le condizioni servono sensori con cui analizzare le informazioni e trasferirle a sistemi digitali. IIoT e IoT stanno diventando parte della nostra vita quotidiana, ad esempio nelle Utility e negli Smart Home Systems come Alexa, Google Home, Apple Home o altri. Gli investimenti nell'automazione dei Trasporti, Smart City e

Smart Industry sono in crescita. Avnet Silica offre un vasto portafoglio di soluzioni per la progettazione di moderni sistemi connessi che collegano il mondo reale e i servizi digitali e mette a disposizione le proprie risorse per supportare lo sviluppo allineato all'evoluzione dei sistemi. Per questo propone i Sensing Discovery Days, appuntamenti concentrati su diverse tecnologie di

sensori, sulla loro gestione con sistemi digitali e sulle applicazioni con alcuni esempi di laboratorio. Registratevi per essere aggiornati sui Sensori in diversi campi come Smart Industry, Smart City, BioSensing e Image!

Le prossime date in programma per l'evento sono 1 e 15 dicembre; l'orario è dalle 9.00 alle 12.45.

www.avnet.com

> 3 DICEMBRE 2020

LIVE DAYS: IN STRETTO CONTATTO CON ST

STMicroelectronics ha organizzato ST LIVE DAYS, un evento web articolato in tre giorni particolarmente ricco di contenuti, avviato il 19 novembre per condividere con i partecipanti i più recenti prodotti e soluzioni riguardanti Internet of Things e 5G, Smart Mobility e Power & Motor Control, i mercati da tutti indicati come i più interessanti per l'industria dei semiconduttori nei prossimi anni. A partire dal 19 novembre, ST condividerà con i partecipanti i più recenti prodotti e soluzioni riguardanti Internet of Things e 5G, Smart Mobility e Power & Motor Control, che sono le aree individuate come le più promet-



tenti degli anni a venire.

L'evento prevede oltre 20 sessioni dal vivo con la possibilità di porre domande ad oltre 50 esperti di ST.

Inoltre è stata allestita un'area esposi-

va virtuale, nella quale saranno disponibili e visualizzabili on-line più di 40 soluzioni innovative, disponibili 24 ore su 24, 7 giorni su 7, durante e dopo i Live Days. Gli appuntamenti in programma, dopo quelli già tenuti in novembre (prodotti, soluzioni, strumenti ed ecosistemi che consentono di sviluppare oggetti connessi e sicuri per applicazioni wireless all'avanguardia, ma anche Smart Mobility) sono quello del 3 dicembre, dal tema Power & Motor Control (verterà sull'offerta di ST per i progetti di potenza con le nuove soluzioni wide-bandgap). L'inizio è per le ore 10.00.

www.st.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

10 NUMERI
A SOLI

€39

ANZICHÉ € 60,00

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"

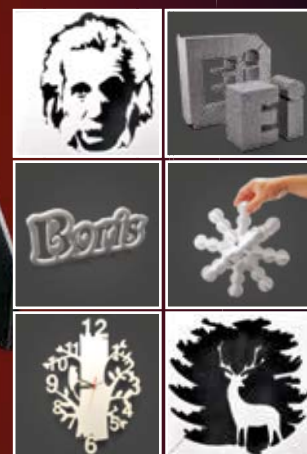
La possibilità di **LEGGERE GRATUITAMENTE**
la versione digitale di ELETTRONICA IN

10
NUMERI
A SOLI

€45

ANZICHÉ € 60,00

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

FISHINO

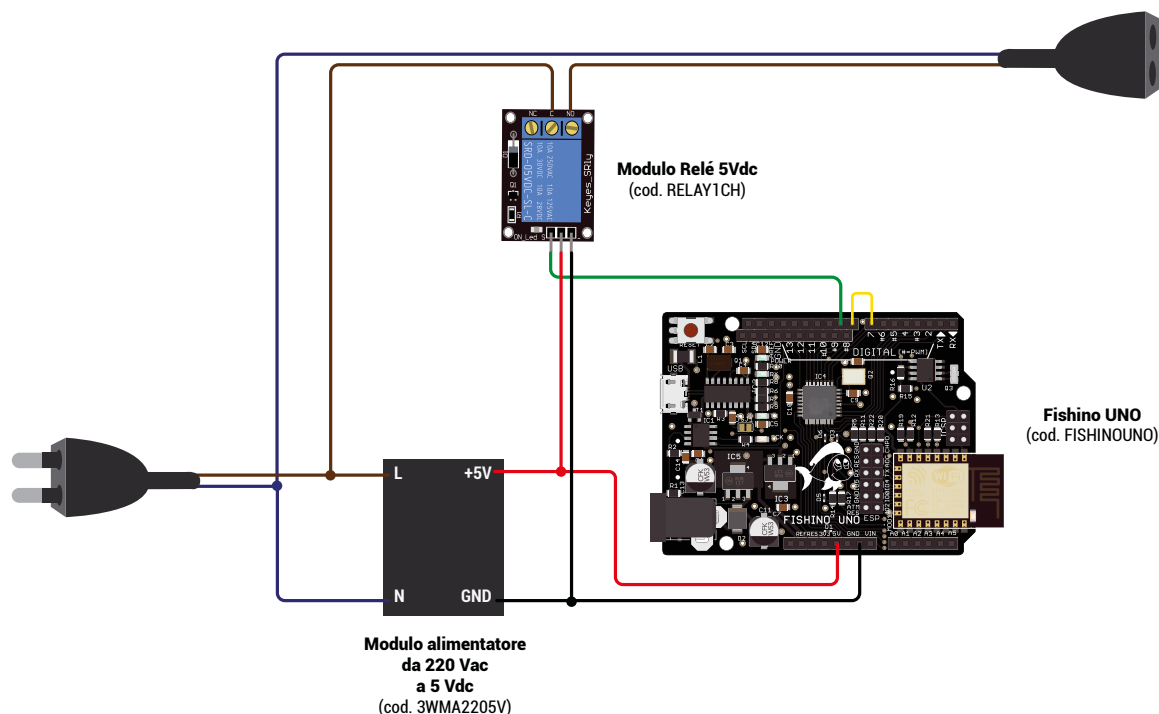
NATALE CON ALEXA E GOOGLE ASSISTANT



Gestiamo gli addobbi natalizi luminosi da remoto attraverso una scheda Fishino e Google Assistant o Alexa: chiediamo di accendere le luci dell'alberello e questo avverrà, da qualsiasi parte noi siamo.

di EMANUELE SIGNORETTA

Negli ultimi anni, grazie alla diffusione pervasiva di Internet e alla sempre attuale legge di Moore, troviamo dispositivi sempre più piccoli, economici e soprattutto capaci di connettersi in rete: viviamo l'era dell'IoT. Di pari passo stiamo assistendo alla comparsa di numerose piattaforme per sviluppatori e hobbisti che consentono di governare agevolmente luci, prese di corrente e altri dispositivi connessi al Cloud. Uno dei servizi più utilizzati è senza ombra di dubbio *IFTTT.com*, acronimo di **If This Then That** (se succede questo fai quest'altro). Possiamo descrivere efficacemente questo servizio come un aggregatore di tanti servizi basati sul Cloud: Google Assistant, Calendar, Amazon Alexa e tantissimi altri. Spetta a noi creare attraverso una GUI (Graphic User Interface) user-friendly le varie applet che vorremo per automatizzare la "nostra vita".



Le applicazioni sono svariate e spaziano dalla domotica ai servizi immateriali: basti pensare che si può creare un'applet che acquisti automaticamente una cassa di birra da Amazon, non appena ricevuta l'e-mail da parte della banca con la conferma dell'avvenuto accredito dello stipendio! In questo articolo tratteremo un'applicazione molto interessante e utile soprattutto nel periodo natalizio: quante volte vi è capitato di dimenticarvi le luci dell'albero di Natale accese quando siete via da casa? E quante altre vi siete dimenticati di accendere le luminarie che decorano la facciata o il cortile? Utilizzando il servizio offerto da IFTTT creeremo delle applet che ci permetteranno di comandare gli addobbi natalizi utilizzando gli assistenti vocali più diffusi: **Google Assistant** e **Amazon Alexa**.

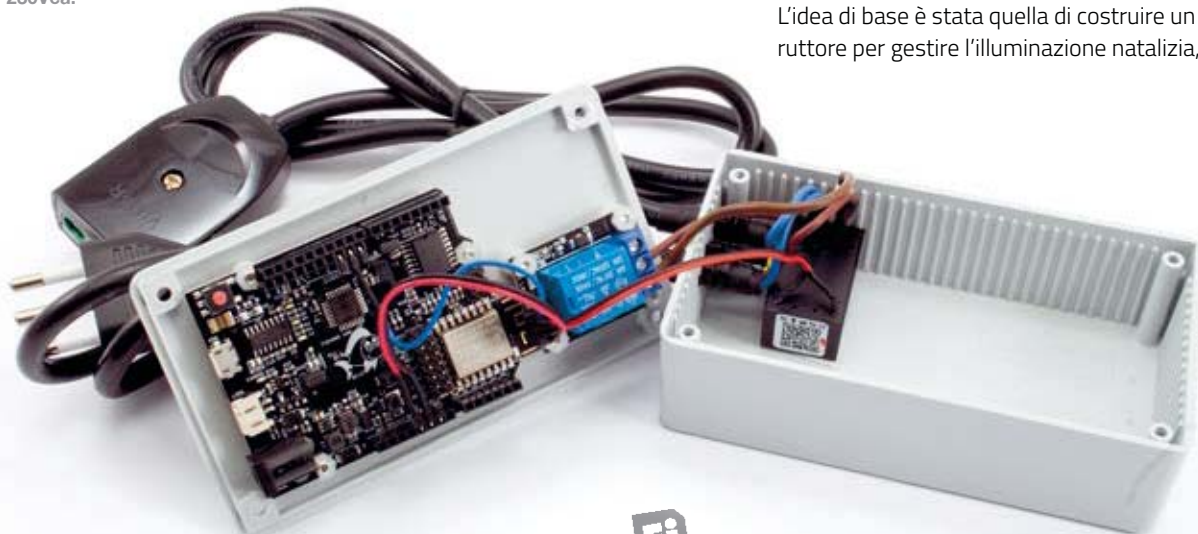
IL PROGETTO

Per questo semplice progetto abbiamo pensato di fare affidamento su una scheda Fishino che si connetterà ad un server MQTT. Utilizzando un qualunque dispositivo con Assistente Google integrato (Android/Google Home) o Alexa saremo in grado di impartire comandi vocali a Fishino oppure di creare delle vere e proprie routine. Ogni volta che l'Assistente Google o Alexa riceveranno il comando (per accendere o spegnere gli addobbi natalizi), questi comunicheranno con il server IFTTT; che a sua volta comunicherà con il server, **io.adafruit.com** il quale si prenderà carico di inviare i comandi in MQTT (Message Queue Telemetry Transport) al Fishino.

L'HARDWARE

L'idea di base è stata quella di costruire un interruttore per gestire l'illuminazione natalizia, quindi

→ Il prototipo inscatolato e dotato di cavi per l'alimentazione a 230Vca.



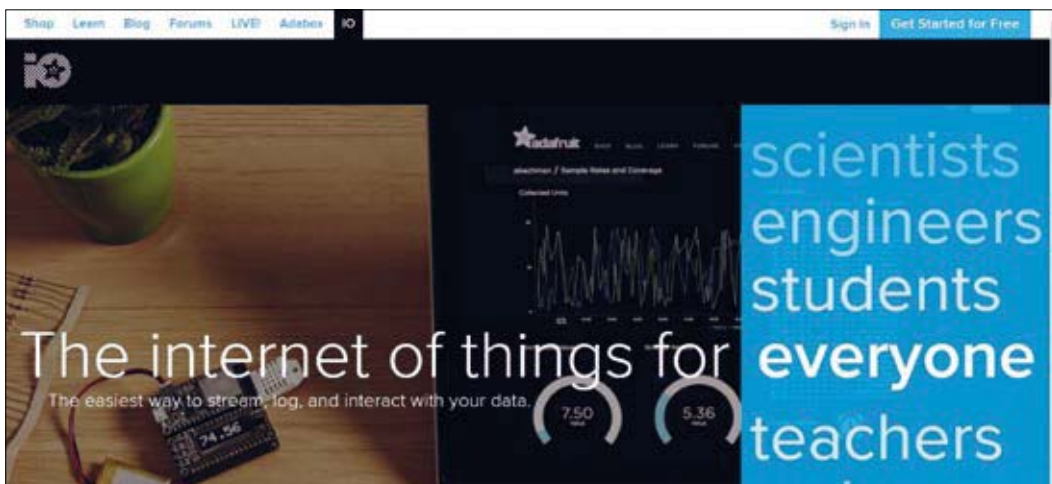


Fig. 1
Home page
del sito
io.adafruit.com.

il nostro circuito sarà composto da Fishino Uno (codice 7305-FISHINOUNO) che pilota una scheda relé comandata a 5V (codice RELAY1CH) cui è affidato il compito di aprire e chiudere il circuito di alimentazione di una presa volante comandata. Il dettaglio dei collegamenti è meglio descritto dallo schema di cablaggio visibile nella pagina accanto, che userete per interconnettere le varie parti componenti il circuito. Prenderete quindi una prolunga di alimentazione da rete (qualsiasi lunghezza va bene) e la taglierete in due parti, dopodiché farete passare le due estremità non intestate dentro un contenitore plastico (il classico COFFER2 della Teko) dentro cui collegherete l'elettronica. Al pin 9 della scheda sarà connesso un relé che verrà utilizzato per interrompere la fase del collegamento. Fishino è alimentato da un alimentatore switching in grado di convertire i 220Vca in 5Vcc; in particolare abbiamo usato il modello 3WMA2205V venduto da Futura Elettronica.

Vi chiediamo di prestare particolare **attenzione** a questi cablaggi. La presenza dell'alimentazione di rete chiede sempre particolare accortezza!

LA CONFIGURAZIONE DEL SERVER MQTT

Iniziamo con la configurazione del software andando ad impostare correttamente l'account sul sito *io.adafruit.com*: sarà proprio questo il server cui si conatterà il Fishino per ricevere i comandi. Come già detto il sistema si basa sul protocollo MQTT e per questo motivo ci serve un "broker" in grado di interpretare i dati ricevuti da IFTTT e inviare i comandi a Fishino. Per iniziare rechiamoci sulla home page e clicchiamo su "Sign in" se siamo già registrati o su "Get started for free" per registrarci. (**Fig. 1**). Una volta loggati ci comparirà la dashboard personale, da qui clicchiamo quindi su "Feeds" → "View all" → "**Create new feed**" e apparirà un pop-up (**Fig.2**) per creare un feed.

Diamo un nome al nostro feed che ci servirà per

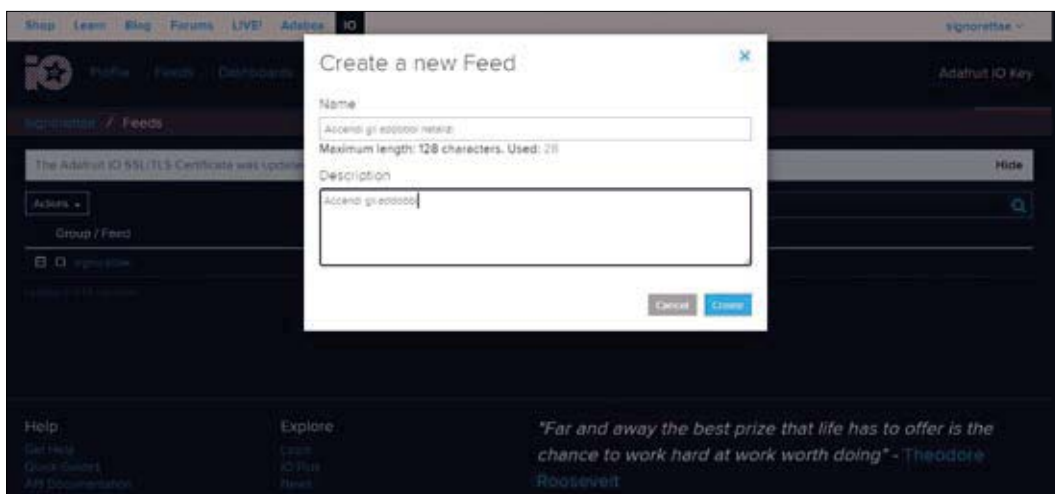


Fig. 2
Configurazione
di un feed sul
cloud Adafruit.

Fig. 3
Credenziali per
l'accesso dei
dispositivi mediante
protocollo MQTT



Fig. 4
Pagina di creazione
di un'applet.

accendere gli addobbi, ad esempio "Accendi gli addobbi natalizi", aggiungiamo una breve descrizione e infine clicchiamo su **"Create"**. Vedremo quindi apparire il feed per accendere gli addobbi nella nostra dashboard con a fianco la

"Key" relativa che ci servirà in seguito.

Ripetiamo la procedura per il feed che regola lo spegnimento delle luci e anche in questo caso vedremo il feed nella dashboard con la relativa "Key". A questo punto selezioniamo la voce **"ADAFRUIT IO KEY"** in alto a destra per visualizzare le nostre chiavi (Fig. 3) e appuntiamoci i valori **"Username"** e **"Active key"** che utilizzeremo per la personalizzazione del firmware. Bene, abbiamo appena completato la procedura di configurazione del server MQTT. Passiamo ora alla configurazione del servizio IFTTT.

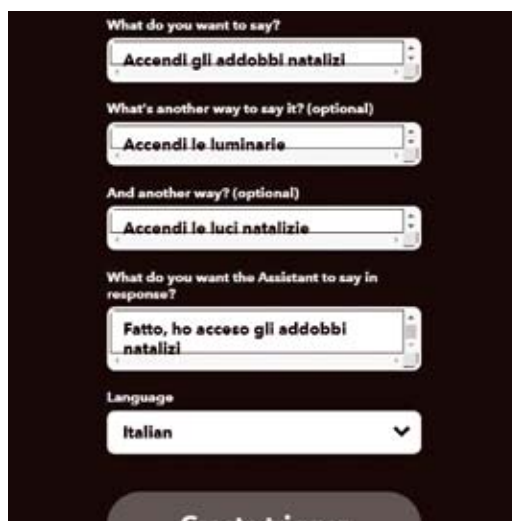
LA CONFIGURAZIONE DI IFTTT CON GOOGLE ASSISTANT

Come accennato precedentemente, IFTTT è un aggregatore di servizi in Cloud che permette di "innescare" dei comandi quando avvengono delle condizioni specifiche. Il servizio offre gratuitamente la creazione di tre applet e per chiunque fosse interessato a crearne di più è possibile sottoscrivere un abbonamento.

Per iniziare a configurare il servizio richiamoci sul sito www.ifttt.com. Clicchiamo in alto a destra sulla voce "Sign in" se siamo già registrati, "Sign up" per registrarci. È comunque possibile eseguire l'accesso social con i propri account Google, Facebook o Apple. Una volta loggati ci comparirà la dashboard del sito dalla quale potremo creare una nuova "applet", così sono chiamati gli automatismi che questo sito permette di configurare.

Clicchiamo quindi su "create" e ci verrà mostrata la pagina per la creazione degli applet (Fig. 4). Alla voce **"If this"** clicchiamo su **"add"** e apparirà un elenco con tutti i servizi disponibili: cerchiamo Google Assistant e autorizziamo IFTTT ad accedere ai dati dell'Assistente. Una volta fatto avremo a disposizione una serie di **"trigger"** tra cui scegliere:

Fig. 5
Possibile
configurazione di
un trigger basato
sul comando
vocale.



il nostro sarà del tipo **"Say a simple phrase"**. Nella nuova schermata dobbiamo completare una serie di campi: un esempio è quello illustrato nella **Fig. 5**. È possibile inserire un massimo di tre comandi vocali per innescare lo stesso trigger. Inseriamo la risposta che ci verrà data dall'Assistente per ogni comando ricevuto, selezioniamo la lingua e clicchiamo su **"Create trigger"** per confermare e tornare alla schermata precedente. A questo punto è possibile cliccare sul bottone **"Add"** della voce **"Then That"**. Selezioniamo Adafruit tra i servizi disponibili e **"Send data to Adafruit IO"** e come per Google Assistant diamo l'autorizzazione ad accedere ai servizi di Adafruit. Selezioniamo il feed "Accendi gli addobbi natalizi" e nel campo "Data to save" possiamo digitare un testo qualsiasi (**Fig. 6**). Una volta cliccato su **"Create action"** verremo portati di nuovo nella schermata principale (**Fig. 7**) dove vedremo un riepilogo delle nostre configurazioni e una volta cliccato su "Continue" avremo creato la nostra prima applet. Ripetiamo la procedura per l'applet che gestisce lo spegnimento degli addobbi e così avremo completato la configurazione di IFTTT. Ora basterà dire rispettivamente "Hey Google, accendi gli addobbi natalizi" oppure "Hey Google, spegni gli addobbi natalizi" per accendere o spegnere gli addobbi mediante comando vocale.

LA CONFIGURAZIONE DI IFTTT CON AMAZON ALEXA

È possibile configurare IFTTT per lavorare anche con l'assistente vocale Amazon Alexa. Poiché il limite massimo di applet gratuite per utente è tre, abbiamo deciso di creare un altro account utilizzando un altro indirizzo e-mail. Nella pagina di creazione dell'applet scegliamo il servizio Amazon

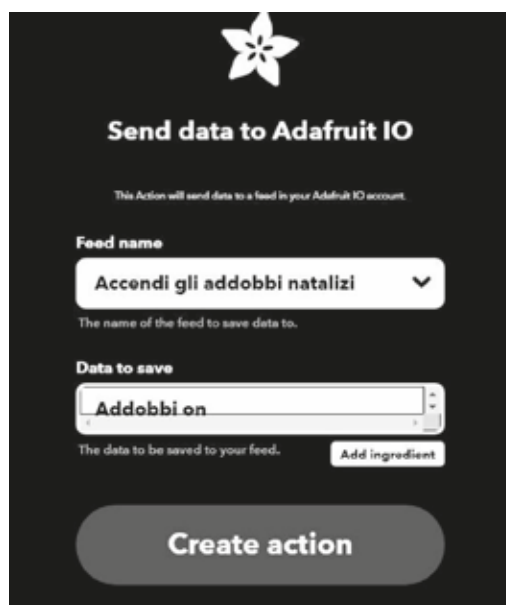


Fig. 6
Misure di
lavorazione
dei pannelli
anteriori.



Fig. 7
Riepilogo
dei dati
dell'applet
creata con
Google
Assistant.

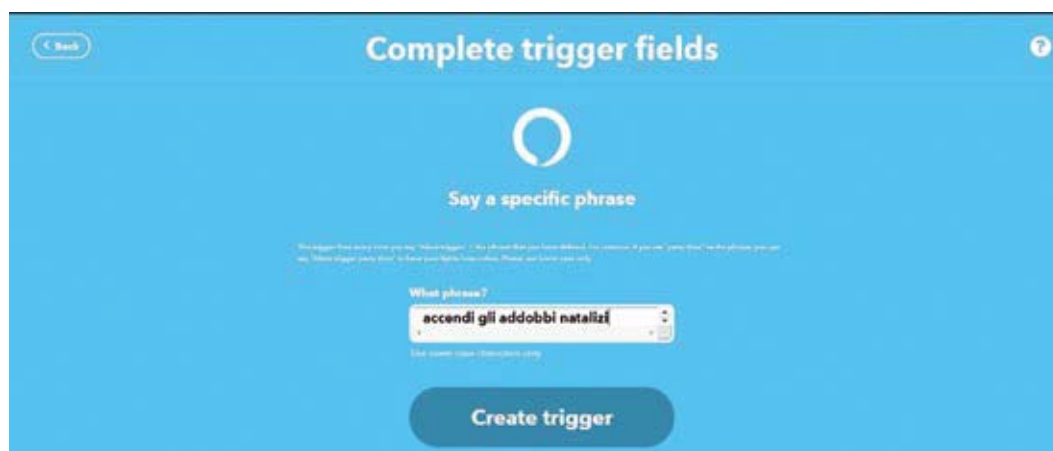


Fig. 8
Frase di
attivazione
del trigger.

➔ **Fig. 9**
Riepilogo
dell'applet creata
con Amazon Alexa.



Alexa e connettiamo IFTTT al nostro account Amazon. Per farlo è sufficiente consentire l'accesso di IFTTT ai nostri dati Amazon. Successivamente clicchiamo su **"Say a specific phrase"** tra i vari trigger disponibili e inseriamo la

frase che vogliamo utilizzare per accendere gli addobbi, come in (Fig. 8). Cliccato su "Create trigger" possiamo procedere alla configurazione dell'applet come già fatto per l'Assistente Google. Il risultato sarà come quello raffigurato in Fig. 9.

È possibile associare l'account Adafruit a due account IFTTT, perciò non è necessario crearne uno nuovo. Ripetiamo le procedure di cui abbiamo parlato nel paragrafo precedente e abbiamo creato la nostra prima applet con Alexa. Ripetiamo tutto per l'applet che gestisce lo spegnimento degli addobbi e la configurazione è completa.

CREAZIONE DI UNA ROUTINE CON GOOGLE ASSISTANT

Come di molti di voi sapranno è possibile anche creare delle routine per Google Assistant. Le routine sono state ideate per poter eseguire più azioni in alcune determinate condizioni come data e ora oppure con lo stesso comando vocale. In questa sezione realizzeremo una routine che verrà eseguita su diversi dispositivi. Il nostro scopo è creare un clima natalizio a tutti gli effetti: con un nostro comando verranno accesi gli addobbi natalizi e verrà riprodotto un brano da Spotify su tutti i dispositivi Google Home presenti a casa.

Per prima cosa scarichiamo l'app Home (disponibile sia per Android sia per iOS). Seguiamo le procedure guidate per la creazione della casa virtuale e per la configurazione dei dispositivi Google.

Se abbiamo a disposizione due o più Google Home è possibile creare un "gruppo". Si possono controllare tutti i dispositivi appartenenti a tale gruppo come se si trattasse di uno solo.

Per creare la routine avviamo l'app Home (Fig. 10) e selezioniamo la voce **"Routine"**.

➔ **Fig. 10**
Schermata
principale dell'app
Home di Google.



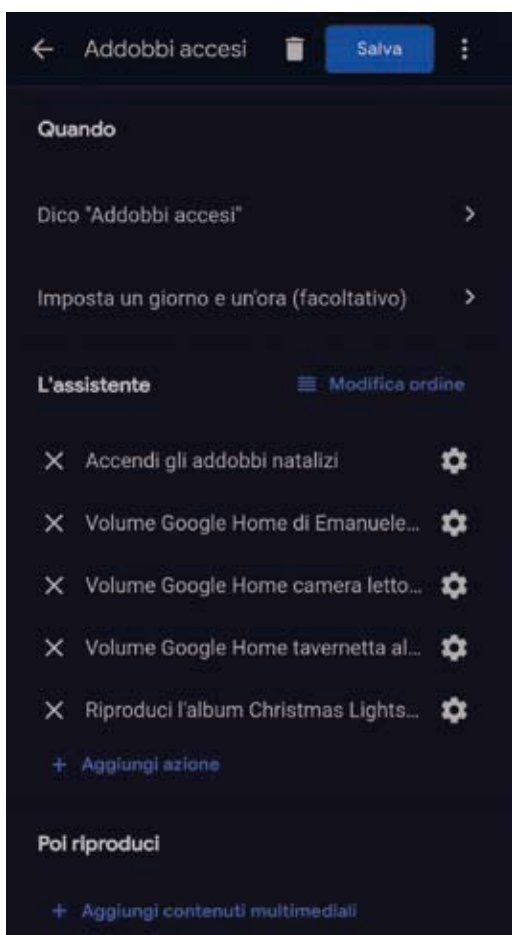


Fig. 11

Riepilogo di una routine Google appena creata.

Una volta nella nuova schermata facciamo un "tap" sul pulsante in basso a destra a forma di "+" per poter creare il trigger e inserire i comandi che verranno eseguiti. Durante i test abbiamo riscontrato un problema per quanto riguarda la gestione dei volumi. Se inseriamo il comando "Volume gruppo xx al 30%", questo non viene eseguito correttamente e i volumi saranno completamente diversi tra i vari dispositivi. Consigliamo di regolare ad uno ad uno il volume dei singoli dispositivi e far partire la musica come gruppo così da superare l'ostacolo. Volendo è possibile sostituire il brano con una playlist con un comando del tipo "Riproduci la playlist Christmas Hits in ordine casuale su gruppo xx". Nella **Fig. 11** è raffigurato il riepilogo della routine che abbiamo creato: una volta cliccato su "Salva" la nostra routine sarà pronta per essere avviata con un semplice comando vocale "Hey Google, addobbi accesi".

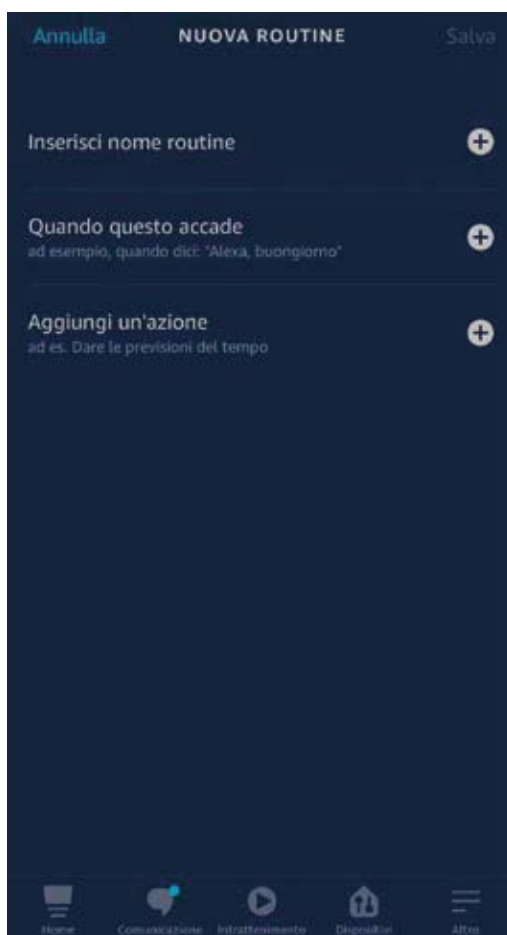


Fig. 12

Schermata di creazione di una routine Alexa.

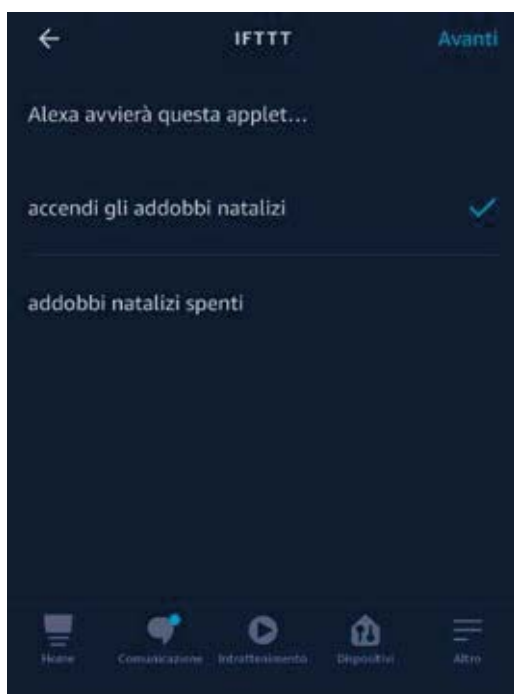
CREAZIONE DI UNA ROUTINE CON AMAZON ALEXA

Per creare una routine con Alexa è necessario scaricare l'app *Amazon Alexa* sul proprio smartphone. Una volta effettuato l'accesso potremo creare la nostra routine tappando sulla voce "Altro" → "Routine" e apparirà la schermata che permette la creazione di routine personalizzate (**Fig. 12**). Procediamo dunque ad inserire un nome alla routine e una volta fatto selezioniamo la voce "**Quando questo accade**". Andremo a creare un trigger: può essere un comando vocale, una sveglia o un evento generato da uno smart-device.

Noi utilizzeremo un comando vocale, perciò inseriremo la frase che farà avviare la routine: "*Addobbi natalizi accesi*".

Clicchiamo su "**Aggiungi azione**" → "IFTTT", come in figura. Selezioniamo l'applet che permette l'attivazione del relé (**Fig. 13**) e infine su "Avanti".

Fig. 13
Scelta dell'applet IFTTT per la creazione della routine.



dire "Alexa, addobbi natalizi accesi" e le decorazioni si accenderanno automaticamente e la musica natalizia riempirà la nostra casa, anche a distanza, utilizzando l'app da smartphone.

LO SKETCH

Lo sketch che governa l'intero progetto è molto semplice, ed è basato sul codice messo a disposizione da Adafruit, ma vediamo assieme nel dettaglio. Fishino Uno è stato programmato per connettersi al server di Adafruit con le credenziali univoche e controllare lo stato dei pin di output prima di azionare gli addobbi.

Dal sito mqtt.org/getting-started troverete una spiegazione molto dettagliata sul funzionamento del protocollo utilizzato.

Le eventuali migliorie apportate al codice saranno disponibili al link <https://github.com/signorettae/Smart-Christmas>.

Analizziamo passo dopo passo i vari listati.

Qui di seguito vediamo la parte riguardante l'inclusione delle librerie:

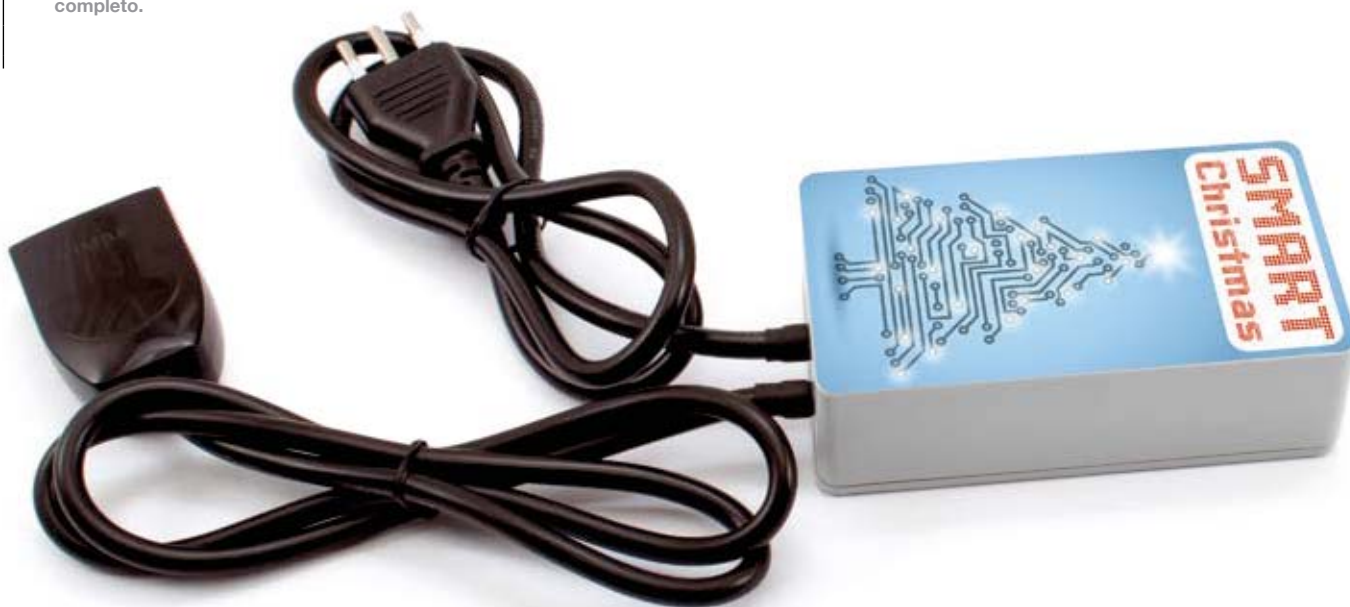
```
#include <SPI.h>
#include <Fishino.h>
#include <Adafruit_MQTT.h>
#include <Adafruit_MQTT_Client.h>
#include <EEPROM.h>

#define output 9
#define feedback_out 8
#define feedback_in 7
```

Nell'ordine troviamo le librerie *SPI.h* e *Fishino.h*.

È possibile scaricare l'ultima versione di quest'ulti-

Fig. 14
Il prototipo completo.



▼ Listato 1

```
#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT 1883           // usare la 8883 se si usa il Secure Sockets Layer (SSL)
#define AIO_USERNAME   "inserire qui l'username"
#define AIO_KEY        "inserire qui l'active key"

//Crea un client appartenente alla classe FishinoClient per la connessione al server MQTT
FishinoClient client;
// oppure usa un FishinoSecureClient per la connessione in SSL
//FishinoSecureClient client;

// Crea il client MQTT utilizzando il FishinoClient, i dati del server e le credenziali di accesso.
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);

/***** Feeds *****/
// Crea un feed chiamato 'accendi-gli-addobbi-natalizi'. Verrà utilizzato per ricevere dal server gli aggiornamenti inerenti lo stato.
Adafruit_MQTT_Subscribe addobbionFeed = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/accendi-gli-addobbi-natalizi");
// Crea un feed chiamato 'spegni-gli-addobbi-natalizi'. Verrà utilizzato per ricevere dal server gli aggiornamenti inerenti lo stato.
Adafruit_MQTT_Subscribe addobbioffFeed = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/spegni-gli-addobbi-natalizi");
/**** Sketch Code ****/
```

ma dal sito www.fishino.it. Subito dopo troviamo le librerie *Adafruit_MQTT.h* e *Adafruit_MQTT_Client.h*, scaricabili entrambe al link www.github.com/adafruit/Adafruit_MQTT_Library.

Queste due librerie ci permetteranno di interfacciarsi facilmente con il server di Adafruit (ma funzionano con un qualunque server MQTT) attraverso il protocollo MQTT. L'ultima libreria utilizzata è la *EEPROM.h*, ci permetterà di utilizzare l'EEPROM a bordo delle nostre schede per salvare e leggere i dati con lo stato dei pin.

Subito dopo vengono dichiarati rispettivamente i pin di output del relé, un pin di feedback settato come output e un altro come input. Durante la fase di sviluppo del progetto ci siamo resi conto che tentando di eseguire un *digitalRead* direttamente dal pin *output* le letture non risultavano non veritiere. Perciò abbiamo deciso di ovviare a questo problema leggendo direttamente dal pin *feedback_in* lo stato del pin *feedback_out* (lo stato è sincronizzato con quello del pin *output*).

Il **Listato 1** contiene alcuni dati di configurazione inerenti al server MQTT. Con la costante *AIO_SERVER* viene indicato il link del server a cui connettersi (in questo caso *io.adafruit.com*) mentre con *AIO_SERVERPORT* viene indicata la porta del server. Noi abbiamo utilizzato la porta 1883 ma è possibile utilizzare una connessione protetta utilizzando la porta 8883. I campi *AIO_USERNAME* e *AIO_KEY* contengono i nostri dati relativi al nostro account Adafruit. È sufficiente inserire i dati che abbiamo

appuntato prima tra le virgolette. È opportuno accertarsi che non siano presenti spazi in quanto poi Fishino non sarà in grado di autenticarsi presso il server. Di seguito viene creato l'oggetto *client* appartenente alla classe *FishinoClient*: verrà utilizzato per creare l'oggetto *mqtt* appartenente alla classe *Adafruit_MQTT_Client*.

▼ Listato 2

```
Serial.begin(115200);
delay(10);
Serial.println(F("Smart-Christmas"));
pinMode(output, OUTPUT);
pinMode(feedback_out, OUTPUT);
pinMode(feedback_in, INPUT);

int y;
EEPROM.get(0, y);
if (y == 0) {
  digitalWrite(output, LOW);
  digitalWrite(feedback_out, LOW);

  Serial.println("Mantengo gli affobbi spenti");
}
else if (y == 1) {
  digitalWrite(output, HIGH);
  digitalWrite(feedback_out, HIGH);
  Serial.println("Mantengo gli addobbi accesi");
}
[...]
```

```
// Inizializza le iscrizioni ai feed MQTT.
mqtt.subscribe(&addobbionFeed);
mqtt.subscribe(&addobbioffFeed);
}
```

Al termine di questo listato vengono creati gli oggetti *addobbionFeed* e *addobbioffFeed*. Entrambi appartengono alla classe *Adafruit_MQTT_Subscribe* e fanno riferimento all'oggetto *mqtt*.

Questi comandi salvano sul Fishino i topic di cui parlavamo prima. In questo caso il nome di ciascun topic è nella forma */feeds/nome-del-feed-assegnato-dal-server*. Il contenuto del **Listato 2** è parte del codice presente nel setup.

Nelle prime righe di codice vengono inizializzati i pin della scheda.

A seguire abbiamo previsto l'uso dell'EEPROM: può capitare che mentre gli addobbi sono accesi vi sia un'interruzione spiacevole della corrente elettrica. Essendo il relé monostabile, una volta ripristinata la corrente le luminarie rimarrebbero spente.

Per aggirare il problema abbiamo deciso di leggere all'avvio il valore su un indirizzo specifico dell'EEPROM: se il valore è 1 i pin digitali di output vengono attivati, se è 0 vengono mantenuti disattivati. Per questioni di praticità abbiamo ommesso la parte di codice inerente la connessione del Fishino alla

↓ Listato 3

```
void loop() {

    // Controlla che la connessione al server MQTT sia attiva(effettuerà la prima connessione
    // e si riconnetterà in automatico quando disconnesso). Controlla la definizione
    // della funzione funzione MQTT_connect di seguito.
    MQTT_connect();

    Adafruit_MQTT_Subscribe *subscription;

    while ((subscription = mqtt.readSubscription(5000))) {

        if (subscription == &addobbionFeed) {
            Serial.print(F("Ricevuto: "));
            Serial.println((char *)addobbionFeed.lastread);

            if (digitalRead(feedback_in) == LOW) {
                digitalWrite(feedback_out, HIGH);
                digitalWrite(output, HIGH);
                int z = 1;
                EEPROM.put(0, z);
                Serial.println("Addobbi accesi");
            }
            else if (digitalRead(feedback_in) == HIGH)
            {
                Serial.println("Addobbi gia' accesi");
            }
        }

        else if (subscription == &addobbioffFeed) {
            Serial.print(F("Ricevuto: "));
            Serial.println((char *)addobbioffFeed.lastread);

            if (digitalRead(feedback_in) == HIGH) {
                digitalWrite(feedback_out, LOW);
                digitalWrite(output, LOW);
                int z = 0;
                EEPROM.put(0, z);
                Serial.println("Addobbi spenti");
            }
            else if (digitalRead(feedback_in) == LOW)
            {
                Serial.println("Addobbi gia' spenti");
            }
        }
    }
    // Pinga il server per mantenere attiva la connessione
    if (! mqtt.ping()) {
        mqtt.disconnect();
    }
}
```



Listato 4

```
void MQTT_connect() {
    int8_t ret;

    // Stop se già connesso.
    if (mqtt.connected()) {
        return;
    }

    Serial.print("Connessione al server MQTT io.adafruit.com... ");

    while ((ret = mqtt.connect()) != 0) { // La funzione mqtt.connect restituisce il valore 0 se connesso
        Serial.println(mqtt.connectErrorString(ret));
        Serial.println("Ritento la connessione al server MQTT tra 5 secondi...");
        mqtt.disconnect();
        delay(5000); // attende 5 secondi
    }
    Serial.println("MQTT Connesso!");
}
```

rete WiFi. Il listato si conclude con la sottoscrizione del client ai topic.

Il **Listato 3** è senza ombra di dubbio quello più sostanzioso in quanto gestisce la ricezione e l'elaborazione dei comandi inviati dal server MQTT. La funzione `MQTT_connect()` si assicura che la connessione con il server MQTT sia attiva e in caso contrario provvede in automatico a connettersi. Con il comando `while ((subscription = mqtt.readSubscription(5000))) {...}` faremo in modo che il Fishino controlli ogni 5 secondi lo stato dei feed a cui è iscritto. All'interno delle parentesi graffe controllerà a quale feedback corrispondono i dati ricevuti. Se i dati corrispondono ad uno dei due feed il microcontrollore effettua dei controlli di stato. Visto e considerato che le EEPROM hanno il numero di cicli di scrittura limitato abbiamo deciso di controllare lo stato del relè prima di riscrivere lo stesso valore e consumare inutilmente dei cicli. Se viene ricevuto il comando per l'accensione degli addobbi e lo stato del pin `feedback_input` è a livello logico alto, il Fishino non compirà nessuna azione se non segnalarlo sulla porta seriale. La stessa procedura è stata effettuata per lo spegnimento degli addobbi se il pin `feedback_input` è a livello logico basso. In caso di scrittura dei valori sull'EEPROM abbiamo assegnato a seconda delle situazioni alla variabile `z` rispettivamente i valori 0 e 1 per indicare se gli addobbi sono spenti o accesi. Al termine del loop si tenta di pingare il server e, in caso di esito negativo, viene chiusa la connessione in modo che vengano avviati una serie di tentativi per ristabilire la connessione.

Il **Listato 4** fa sì che il client possa connettersi al server MQTT; questa funzione viene eseguita ripetutamente nel loop. Se Fishino è già connesso, esce dalla funzione stessa; in caso contrario tenta la connessione ogni cinque secondi, stampando sulla porta seriale gli eventuali errori che hanno impedito l'avvio della connessione.

CONCLUSIONI

Termina qui la descrizione del sistema per gestire da remoto gli addobbi natalizi che è un esempio dei numerosi automatismi che si possono creare con servizi quali *IFTTT* e *io.adafruit* e dell'hardware come Fishino. □



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. La board Fishino UNO (cod. FISHINOUNO) è in vendita a Euro 36,00, il modulo da un relè (cod. RELAY1CH) è disponibile a Euro 5,00, il modulo alimentatore da 220VAC a 5VDC (cod. 3WMA2205V) ha un prezzo di Euro 6,50, il contenitore utilizzato (cod. COFFER2) costa Euro 7,50. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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



Controlla tutto tramite App, Alexa, IFTTT e Google Assistant!



Interruttore Wi-Fi

Consente di attivare e disattivare un carico (luci o dispositivi elettronici) funzionante a 220 VAC con un assorbimento massimo di corrente di 10 A, come luci, ventilatori, stufe elettriche e tutto ciò che funziona con la rete domestica. Può essere controllato tramite smartphone o tablet, installando l'App "Smart Life" o "TUYA Life" per iOS o Android, oppure può anche tramite Amazon Alexa, Google Assistant e IFTTT.

€7,90

Cod. FR775

Installabile in scatole di derivazione o porta frutto



€19,90

Cod. FR776



Micro interruttore Wi-Fi a 1 canale

Modulo interruttore compatto Wi-Fi che consente di attivare e disattivare un carico (luci o dispositivi elettronici) funzionante a 220 VAC con un assorbimento massimo di corrente di 10 A. Può essere controllato sia tramite App (Smart Life o Tuya Smart) installata su smartphone, tablet, ecc. sia tramite Amazon Alexa, Google Assistant e IFTTT. Il modulo viene fornito completo di supporto rimovibile per guida DIN.

Disponibile anche a 2 canali con corrente massima di 5A per canale
cod. FR777 **€29,90**



Prezzi IVA inclusa.

€24,90

Cod. FR778



Telecomando Wi-Fi, IR

Dispositivo IR e Wi-Fi che permette tramite l'App "Smart Life o Tuya Life" di trasformare il vostro smartphone/tablet in un telecomando in grado di sostituire tutti i telecomandi IR utilizzati per controllare i dispositivi comunemente presenti in casa (condizionatori d'aria, TV, ventole, DVD e molti altri). Può essere controllato sia tramite App (Smart Life o Tuya Smart) sia tramite Amazon Alexa o Google Assistant.

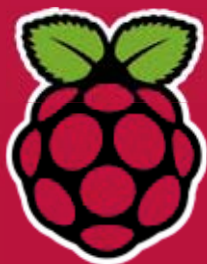
Presa smart Wi-Fi

Controllabile con App, Amazon Alexa, Google Assistant e ifttt, per accendere/spegnere qualsiasi dispositivo ad essa collegato. Dispone di funzione timer, sistema per il monitoraggio del consumo di energia elettrica. La presa è in grado di supportare un carico complessivo di 3400W e una corrente massima di 16A.



€21,90
Cod. FR779

POWERPi



di MAURIZIO ŠKERLIČ

Gruppo di continuità per la Raspberry Pi provvisto di uscita USB capace di ricaricare batterie al litio e ai polimeri di litio. È provvista di display OLED ed è configurabile mediante USB.

Con l'avvento dell'Internet of Things e della domotica, sempre più spesso i dispositivi devono rimanere accesi anche in caso di black-out. Sul mercato esistono innumerevoli tipi e modelli di gruppi di continuità o UPS (Uninterruptible Power Supply) utili a mantenere la tensione 220 Vac per ore, oppure per giorni, ma hanno il difetto di essere costosi, ingombranti e di consumare parecchia energia. Nel caso utilizzassimo una Raspberry Pi come end-device IoT, scegliere un UPS tradizionale per mantenere in funzione il trasformatore di alimentazione in caso di blackout, sarebbe non solo costoso ma anche controproducente. Quando l'autonomia della batteria è bassa, i gruppi di continuità si spengo-

no, e per evitare danni al sistema operativo, dovrebbero pure provvedere automaticamente allo spegnimento controllato della Raspberry Pi, ovvero a mandare in shutdown il sistema quando si arriva a un livello di carica che fa presagire l'insufficienza dell'alimentazione.

Una soluzione alternativa può essere rappresentata dall'utilizzo di un "Power bank" oppure di un dispositivo dedicato come il progetto descritto in queste pagine e che abbiamo chiamato PowerPi; si tratta sostanzialmente di un gruppo di continuità per Raspberry Pi, però di tipo "intelligente" in quanto governato da un microcontrollore ATSAM21G18 a 32 bit e 48 MHz di clock, nel quale gira un firmware capace di interagire con la scheda "del lampone" in modo da chiuderne l'attività se, in condizione di assenza dell'alimentazione

principale, rileva uno stato di carica della batteria che può comportare l'imminente spegnimento o comunque instabilità nel funzionamento. La nostra scheda ha caratteristiche di tutto rispetto, che elenchiamo qui di seguito:

- display OLED con risoluzione 128 x 64 pixel;
- alimentazione da 3,9 V a 14 V;
- corrente di carica massima 2,5 A;
- corrente massima in uscita 8 A;
- compatibile con batterie Li-Po, Li-ion o LiFePO₄;
- programmazione da terminale tramite USB;
- salvaschermo temporizzato del display OLED;
- spegnimento automatico quando la batteria è scarica;
- spegnimento controllato della Raspberry Pi;
- compatibilità con la Raspberry Pi 2, 3, 3 B+, 4 B, Zero e Zero W;
- indicatori a LED dello stato di carica e di allarme;
- alimentazione mediante connettore USB tipo C;
- protezione sovracorrenti sulle porte USB in entrata e in uscita;
- presa USB per la ricarica rapida compatibile con dispositivi Samsung e Apple;
- corrente massima in uscita sul connettore USB di uscita 1,7 A;
- Protezione da cortocircuito sulla porta USB in uscita.

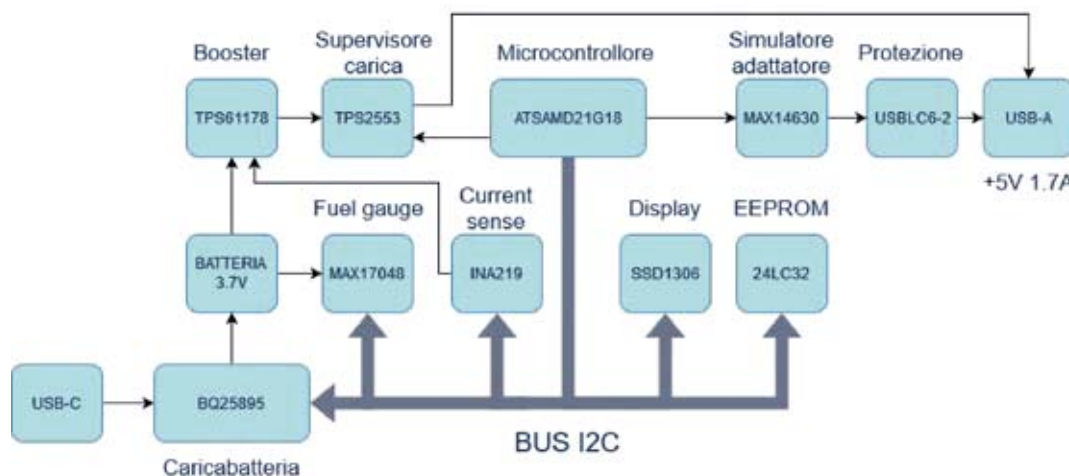
Ciò esposto, passiamo ora ad analizzare l'hardware da cui è composta la nostra scheda UPS.

SCHEMA ELETTRICO

Analizziamo dunque il circuito partendo dallo schema a blocchi di **Fig. 1**, che vi farà meglio comprendere la struttura dello schema elettrico.

La PowerPi può essere alimentata sia tramite la presa USB tipo C sia tramite il morsetto a vite a due vie (nello schema elettrico siglati rispettivamente USB1 e CON2) con una tensione continua compresa tra 3,9 V e i 14 V; tale range consente l'uso anche in auto. L'entrata è protetta da un fusibile autoripristinante (F1) da 3 A e dal varistore VR1. Le linee di bus D+ e D- della porta USB1 a loro volta sono protette dalle sovratensioni da ulteriori due varistori VR2 e VR3. La presenza della tensione in entrata viene segnalata dall'accensione del diodo LED di colore verde LD1, e poiché il range di tensione di alimentazione può variare da 3,9 a 14 V, non potevamo limitare la corrente del diodo LED con una semplice resistenza. Per questo abbiamo dovuto utilizzare un limitatore di corrente BCR402, nello schema elettrico siglato U4, che limita la corrente a 20 mA a prescindere dal voltaggio di alimentazione della PowerPi. La gestione della carica della batteria è affidata al collaudato chip BQ25895 prodotto dalla Texas Instruments, nello schema elettrico siglato U1, progettato per la ricarica delle batterie a cella singola. Può erogare una corrente massima di carica di 5 A e possiede una porta seriale di tipo I²C (abbreviazione di Inter Integrated Circuit) per l'interfacciamento e il controllo mediante un microcontrollore. Per il nostro progetto abbiamo scelto il microcontrollore ATSAMD21G18 (siglato U3), lo stesso montato sulle schede Arduino MO, un ARM CORTEX MO+ a 32 bit con frequenza di clock di 48 MHz, prodotto dall'Atmel. La scelta di questo microcontrollore ci ha consentito l'utilizzo delle innumerevoli librerie dell'ambiente di sviluppo Arduino IDE facilitandoci la fase di programmazione. Il microcontrollore è provvisto di 256 kB di

Fig. 1
Schema
a blocchi della
PowerPi.



memoria Flash per il firmware e 32 kB di memoria RAM, peccato sia sprovvisto di una memoria interna non volatile di tipo EEPROM (Electrically Erasable Programmable Read Only Memory) come nel più datato e meno prestante ATmega328. Per salvare i parametri delle impostazioni siamo stati costretti ad utilizzare una memoria esterna EEPROM I²C 24LC64, siglata U6, aumentando ulteriormente la complessità del circuito. Per monitorare i parametri di carica della batteria abbiamo aggiunto un display OLED da 0,96 pollici SSD1306, siglato LCD1 nello schema elettrico. La durata di vita di questi display, è nettamente inferiore rispetto ai comuni display LCD. Le sostanze organiche utilizzate per la loro produzione tendono a deperirsi, soprattutto gli OLED di tipo blu hanno attualmente un'autonomia breve che, se mantenuti accesi alla massima luminosità, si aggira attorno alle 20.000 ore, ossia all'incirca due anni di vita. Questo significa che il display con il tempo tenderà a perdere la sua luminosità iniziale e i pixel, a seconda della loro usura, presenteranno una differente luminosità, rendendo la visione del display sgradevole e difficoltosa. Per ovviare a questo inconveniente abbiamo implementato una funzione salvaschermo temporizzata, il cui intervento può essere impostato a piacimento oppure disattivato. Tutti i parametri possono essere impostati e modificati da terminale interfacciando la PowerPi ad un PC tramite la presa USB-C.

A tale scopo possiamo utilizzare il "monitor seriale" dell'ambiente di sviluppo Arduino IDE, avviabile dal menu "strumenti".

La tensione e la capacità delle batterie vengono monitorati dal chip "Fuel gauge" MAX17048, siglato U2 nello schema elettrico. Questo chip, grazie all'algoritmo proprietario denominato "ModelGau-

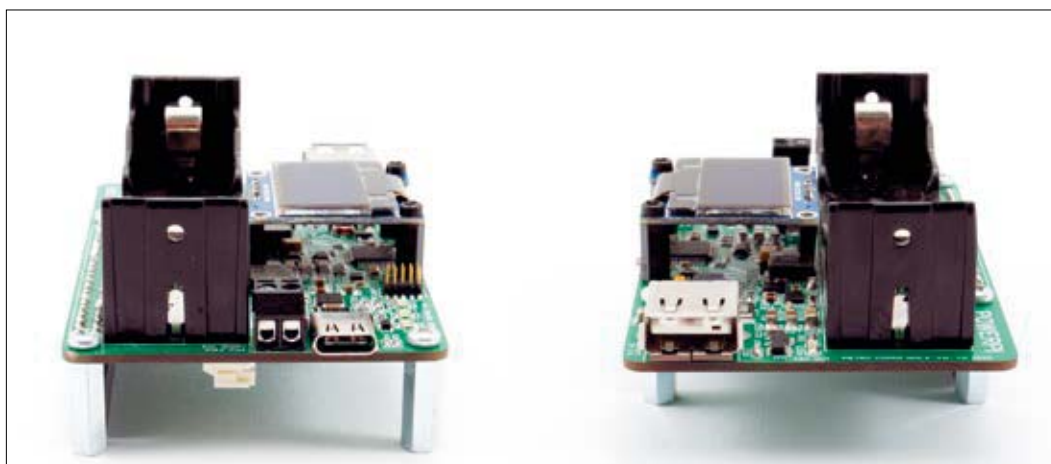
ge" riesce a calcolare la percentuale di capacità residua presente nella batteria, monitorando lo stato di carica relativa (Relative State of Charge), estrapolandola dalle ripetute variazioni di carica e scarica. Non preoccupatevi se, alla sostituzione della batteria, dopo la prima ricarica la capacità non raggiungerà il 100 %. Date tempo al chip di monitorare lo stato della batteria e dopo due o tre ricariche noterete che la misurazione sarà corretta. Per incrementare la tensione della batteria a 5 V abbiamo utilizzato un convertitore booster sincrono TPS61178 prodotto dalla Texas Instruments, nello schema elettrico siglato U7. Il MOSFET Q1 posto sull'uscita del convertitore ha il compito di isolare l'uscita dall'entrata in caso di cortocircuiti o quando il TPS61178 viene spento, altrimenti ci ritroveremo sull'uscita lo stesso voltaggio presente all'entrata. Questo chip è in grado di erogare una corrente di picco di ben 10 A che ci permette di avere una notevole riserva di potenza. La tensione di uscita viene impostato con le resistenze R20 e R21. Con i valori da noi forniti la tensione in uscita raggiunge i 5,15 V. Chi avesse il bisogno di modificare la tensione in uscita può calcolare i valori delle resistenze utilizzando la formula:

$$V_{out} = V_{FB} \left(1 + \frac{R_{20}}{R_{21}} \right) = 1,21 \text{ V} \left(1 + \frac{267000 \Omega}{82000 \Omega} \right) = 5,15 \text{ V}$$

Ad esempio per carichi gravosi potete aumentare la tensione in uscita a 5,25 V scegliendo per R21 un valore di 56 Kohm e per R2 un valore di 187 Kohm. Il costruttore consiglia di usare per la resistenza R21 un valore inferiore a 100 Kohm, in modo che la corrente, che fluisce attraverso questa resistenza, sia 100 volte maggiore della corrente di di-

CARATTERISTICHE TECNICHE

- **Tensione di alimentazione:**
3,9÷14 Vcc
- **Corrente di carica batteria:**
2,5 A
- **Corrente in uscita:**
max. 8 A
- **Batteria supportata:**
Li-ion, LiPo, LiFePO₄
- **Compatibilità:**
Raspberry Pi 2, 3, 3B+, 4B, Zero e Zero W
- **Funzioni:**
Shutdown Raspberry Pi quando la batteria si sta scaricando
- **Configurazione:**
Tramite porta USB e terminale
- **Interfaccia utente:**
Display OLED



La scheda si alimenta tramite connettore USB-C e fornisce corrente alla Raspberry Pi e alla presa USB-A.

schema ELETTRICO :

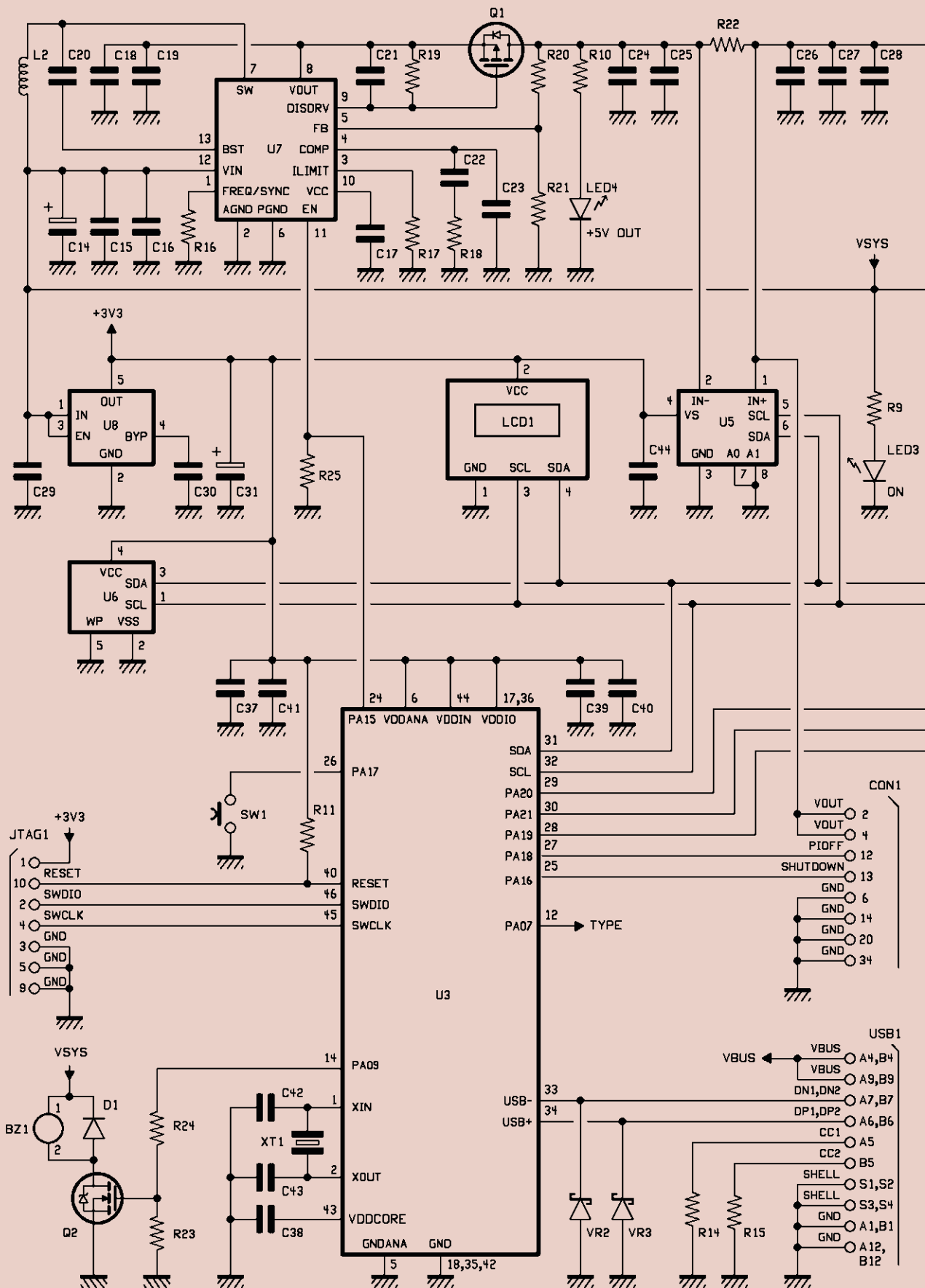


Tabella 1
Valori della
resistenza R12
per la limitazione
della corrente
nell'integrato
TPS2553.

LIMITAZIONE DELLA CORRENTE DESIDERATA (mA)	VALORE DELLA RESISTENZA IDEALE (KΩ)	VALORE DELLA RESISTENZA STANDARD 1% (KΩ)	CORRENTE MINIMA (mA)	CORRENTE NOMINALE (mA)	CORRENTE MASSIMA (mA)
75	Cortocircuitare i piedini 2 e 6		50	75	100
120	226,1	226	101,3	120	142,1
200	134	133	173,7	201,5	233,9
300	88,5	88,7	262,1	299,4	342,3
400	65,9	66,5	351,2	396,7	448,7
500	52,5	52,3	448,3	501,6	562,4
600	43,5	43,2	544,3	604,6	673,1
700	37,2	37,4	630,2	696	770,8
800	32,4	32,4	729,1	800,8	882,1
900	28,7	28,7	824,7	901,5	988,7
1000	25,8	26,1	908,3	989,1	1081
1100	23,4	23,2	1023,7	1109,7	1207,5
1200	21,4	21,5	1106	1195,4	1297,1
1300	19,7	19,6	1215,1	1308,5	1414,9
1400	18,3	18,2	1310,1	1406,7	1517
1500	17	16,9	1412,5	1512,4	1626,4
1600	16	15,8	1512,5	1615,2	1732,7
1700	15	15	1594,5	1699,3	1819,4

Grazie all'emulatore di carica MAX14630, nello schema elettrico siglato U10, la porta USB può essere usata anche come caricabatteria per la carica veloce degli smartphone compatibile con i modelli sia Samsung sia Apple. La compatibilità viene selezionata portando alto (per dispositivi Apple) o basso (per dispositivi Samsung) il bit di controllo (CB) piedino 3 dell'integrato U10. La selezione della compatibilità viene programmata da console tramite la porta USB-C. Terminiamo la descrizione dello schema elettrico analizzando l'ultimo integrato MIC5205 siglato U8, un regolatore lineare a basso rumore da 3,3 V.

IMPOSTAZIONE DEI PARAMETRI

Tutti i parametri d'impostazione possono essere visualizzati o modificati mediante terminale, collegando la PowerPi al PC tramite la presa USB-C. I parametri da modificare vengono selezionati digitando la lettera dell'alfabeto, sia in maiuscolo che minuscolo, scelta dal menu dalla A alla P. Analizziamo assieme ogni singolo parametro e vediamo come modificarlo.

Con la lettera "A" impostiamo la corrente massima con cui verrà caricata la batteria. La capacità della batteria viene indicata con la lettera "C"; per non accorciare la vita media della batteria, durante le ri-

cariche si consiglia di non applicare correnti di carica che in valore assoluto superino "C". Ad esempio, per una batteria da 1.500 mAh la corrente di carica non deve superare 1,5 A; in breve la batteria non dovrebbe essere ricaricata in meno di un'ora. Se volete ricaricare una batteria con una corrente superiore alla capacità "C", sappiate che questa si surriscalderebbe e con correnti superiori, a causa dello sprigionamento di gas al suo interno, può gonfiarsi e anche esplodere. Se non volete accorciare la vita della vostra batteria, cercate di ricaricarla con una corrente non superiore a 0,5 C. Le batterie ricaricabili dopo la carica, a causa della resistenza interna, con il tempo tendono a scaricarsi. Per ovviare a questo inconveniente le batterie nichel-cadmio (NiCd), nichel metallo idruro (NiMH) o al piombo (Pb) vengono mantenute sotto carica con una corrente di mantenimento pari o inferiori a 1/40 C. Con le batterie agli ioni e ai polimeri di litio questo non è possibile poiché non permettono sovraccariche, per questo motivo la carica viene gestita da circuiti integrati dedicati appositamente progettati, che terminano la ricarica della batteria quando la capacità ha raggiunto il 100 %.

Dopo questa lunga e doverosa spiegazione, per impostare la carica con una corrente di 1 A dobbiamo lanciare da terminale il comando "a = 1000".



Con la lettera "B" impostiamo la tensione di carica massima. Seppure le batterie Lipo o Li-Ion a riposo hanno una tensione nominale di 3,7 V, la loro ricarica deve invece avvenire con una tensione superiore, definita dal costruttore e riportata anche sull'etichetta posta sull'involucro della batteria oppure nei data sheet. Di solito questa tensione si aggira attorno ai 4,2 V. Nella PowerPi questo parametro può essere modificato da terminale lanciando il comando "b = 4200". Nel caso, terminata la carica, notiate che sul display la capacità della batteria non raggiunge il 100 %, potete incrementare questo parametro di alcuni millivolt. Nel nostro prototipo, utilizzando gli accumulatori "2001-ICR18650", "8240-LCR18650" oppure "7893-PLIB1100" della Futura Elettronica, abbiamo impostato questo parametro a 4.256 mV. Entrambe i parametri, corrente limite e tensione limite di carica, sono definiti dalla grandezza dei registri interni del chip BQ25895, rispettivamente il 0x04 per la corrente limite (valore a 7 bit) e il 0x06 per la tensione limite (valore a 6 bit). Vi diciamo questo, perché noterete che i valori inseriti da terminale verranno arrotondati in difetto in modo da adattarli alle dimensioni dei suddetti registri. Con la lettera "C" impostiamo la capacità minima della batteria, valore sotto al quale la PowerPi inizia la sequenza di spegnimento. Possiamo definire questo parametro come una protezione per non scaricare eccessivamente la batteria. Questo valore può essere impostato tra il 20 % e il 70 %; l'impostazione predefinita prevede il 20 % (ad esempio "c = 20").

Con la lettera "D" impostiamo il valore minimo di capacità, che la batteria deve raggiungere in fase di carica dopo uno spegnimento, prima di poter alimentare il carico in uscita. Questo parametro permette di consentire alla batteria di ricaricarsi dopo un blackout proteggendola soprattutto quando è troppo scarica. Questo valore può essere impostato tra 40 % e il 90 % e per impostazione predefinita è 40 % (per esempio "d = 40").

Con la lettera "E" impostiamo un ritardo "ΔT" prima dell'inizio della sequenza di spegnimento, impostata con il parametro "C".

Permette di evitare falsi spegnimenti in presenza di correnti variabili. Questo valore può essere impostato da 0 a 90 secondi, di default è impostato a 10 secondi (per esempio "e = 10").

Con la lettera "F" impostiamo il timer del screen saver, cioè il tempo di funzionamento del display. Questo valore può essere impostato da 0 a 60 minuti, ma per default è impostato a 1 minuto.

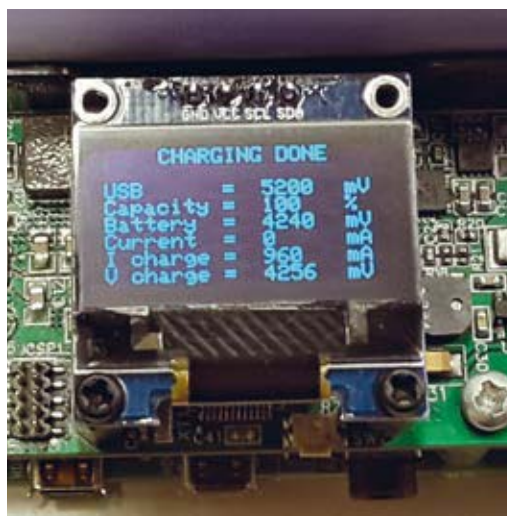


Fig. 2
Nella prima schermata sono visualizzati i parametri di carica della batteria.

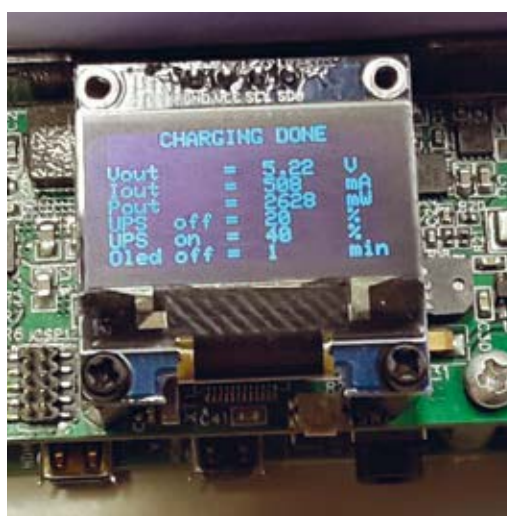
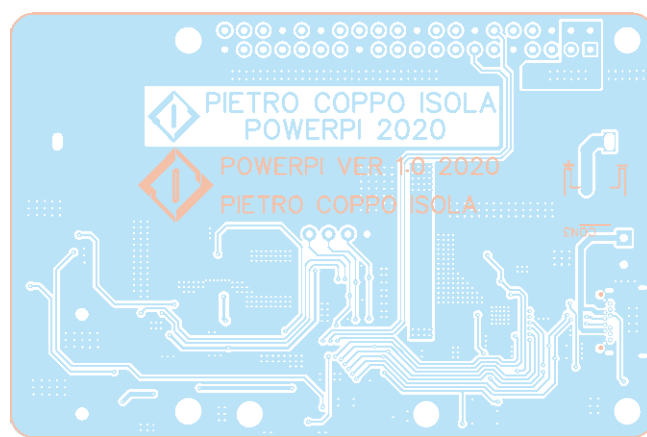
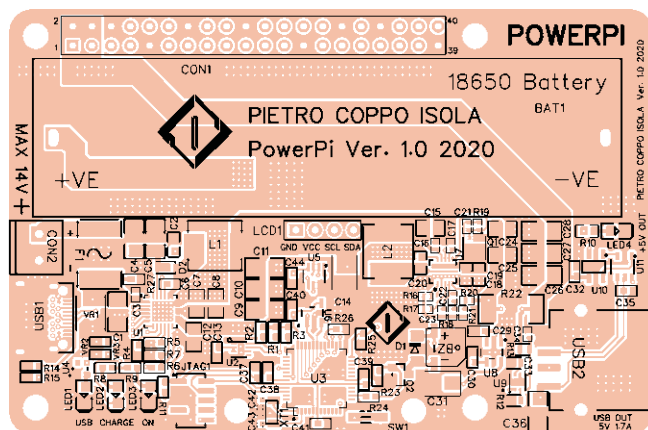


Fig. 3
Nella seconda schermata sono visibili il voltaggio, la corrente e la potenza assorbita in uscita.

Quando impostiamo il valore a zero il display rimarrà sempre acceso: per esempio "f = 1". Con la lettera "G" impostiamo il timer di commutazione delle schermate del display. Il display possiede due tipi di schermate, sulla prima sono visualizzati i parametri di carica della batteria, mentre sulla seconda i valori del voltaggio, della corrente e della potenza assorbita in uscita. Vedi **Fig. 2** e **Fig. 3**. Il passaggio della visualizzazione di una schermata all'altra avviene con una doppia pressione del pulsante di accensione. Questo valore può essere impostato da 0 a 300 secondi, di default è impostato a zero. Esempio "g = 0". Con la lettera "H" impostiamo il tempo di attesa per consentire alla Raspberry Pi di terminare la sequenza di shutdown, la quale termina quando va basso il piedino 12 sul connettore a 40 pin, nello schema elettrico siglato CON1. Questo valore può essere impostato da 10 a 300 secondi;



Elenco componenti:

R1, R2: 1,5 kohm
 R3, R4, R6, R7, R11, R23, R25,
 R26, R27: 10 kohm
 R5: 130 ohm
 R8, R9: 150 ohm
 R10: 68 ohm
 R12: 15 kohm 1%
 R13: 100 kohm
 R14, R15: 5,1 kohm
 R16: 348 kohm
 R17: 51,1 kohm
 R18: 15 kohm
 R19: 150 kohm
 R20: 267 kohm 1%
 R21: 82 kohm 1%
 R22: 01 ohm 1% 3W
 R24: 100 ohm
 C1, C4, C5: 22 μ F 25 VL
 ceramico
 C2: 4,7 μ F 16 VL ceramico
 C3: 1 μ F 6,3 VL ceramico
 C6: 47 nF ceramico
 C7: 100 nF ceramico
 C8, C12, C15: 10 μ F 6,3 VL
 ceramico
 C9, C10, C11, C26, C27: 22 μ F

6,3 VL ceramico:
 C13, C29, C32, C35, C37, C38,
 C39, C40, C41, C44 : 100 nF
 ceramico
 C14: 100uF 10V Tantalio
 Case B
 C16, C33, C34: 100 nF
 ceramico
 C17: 4,7 μ F 6,3 VL ceramico
 C18, C19: 10 μ F 6,3 VL
 ceramico
 C20: 22 nF ceramico
 C21: 22 nF ceramico
 C22: 6,8 nF ceramico
 C23: 10 pF ceramico
 C24, C25: 22 μ F 6,3 VL
 ceramico
 C28: 100 μ F 6,3 VL ceramico
 C30: 470 pF ceramico
 C31: 2,2 μ F 10 VL Tantalio
 Case A
 C36: 150 μ F 6,3 VL ceramico
 C42, C43: 22 pF ceramico
 LED1, LED3: LED verde
 LED2: LED giallo
 LED4: LED blu

D1: 1N4148
 D2: MSS2P3-M3/89A -
 Schottky
 Q1: CSD25402Q3A
 Q2: BSS138
 U1: BQ25895RTWR
 U2: MAX17048G+T10
 U3: ATSAM21G18
 U4: BCR402RE6327
 U5: INA219AIDCNR
 U6: 24LC64
 U7: TPS61178
 U8: MIC5205-3.3YM5-TR
 U9: TPS2553DBVR
 U10: MAX14630
 U11: Protezione ESD
 USBLC6-2:
 LCD1: Display OLED 0,96"
 VR1: TVS SMBJ5.0A
 VR2, VR3: TVS CG0603MLC-05E
 XT1: Quarzo 32,768 kHz
 L1: Induttanza 1 μ H
 L2: Induttanza 1,8 μ H
 BZ1: Trasduttore magnetico
 F1: Fusibile Ripristinabile 3A
 24V

BAT1: Portabatteria 18650
 USB1: Connettore USB-C
 USB2: Connettore USB-A
 SW1: Microswitch 3,5x2,9mm
 CON1: Connettore femmina
 2x20 pin lunghi 12,5 mm
 CON2: Morsetto a 2 vie P. 3,5
 mm
 CON3: Connettore JST 2 pin
 P. 2 mm
 JTAG1: Strip maschio 2x5 pin
 SMD P. 1,27 mm

Varie

- Strip femmina 4 poli
- Strip maschio 4 poli
- Distanziali esagonali F/F
M2,5x18
- Distanziali esagonali F/M
M2x12
- Vite M2,5x6
- Vite M2x6
- Dado M2
- Circuito stampato S1534
(86x57 mm)



La scheda
al completo,
equipaggiata
con la
batteria
18650.

l'impostazione predefinita è 10 secondi (ad esempio "h = 10"). Con la lettera "I" impostiamo il bit per informare il sistema che la PowerPi è connessa alla Raspberry Pi. Per impostazione predefinita, questa impostazione è attivata.

Con la lettera "L" impostiamo l'opzione che permette di spegnere la PowerPi mediante il pulsante dell'interfaccia grafica o attraverso il comando shutdown del sistema operativo Raspbian; con il riavvio del sistema operativo la nostra PowerPi non si spegnerà. Questo comportamento è attivo per impostazione predefinita.

Con la lettera "M" impostiamo l'opzione di riaccensione automatica della PowerPi dopo lo spegnimento successivo a un blackout; questa impostazione è attivata per default.

Con la lettera "N" impostiamo l'attivazione di un segnale acustico di allarme, che si attiva ogni 8 secondi, per avvertirci della mancanza dell'alimentazione. Di default questa impostazione è attivata.

Con la lettera "O" impostiamo l'attivazione del segnale acustico di sistema alla pressione del pulsante di accensione. Per impostazione predefinita, questa impostazione è attivata. Con la lettera

"P" impostiamo il tipo di emulatore della porta USB. Possiamo scegliere di simulare il caricabatteria per dispositivi "Samsung" oppure "Apple". L'impostazione predefinita è la ricarica di dispositivi "Samsung".

REALIZZAZIONE PRATICA

Ora che abbiamo analizzato lo schema elettrico e spiegato il funzionamento del circuito, passiamo alla realizzazione della PowerPi, partendo dal

circuito stampato a doppia faccia, ottenibile per fotoincisione partendo dai file Gerber disponibili per il download sul nostro sito web www.elettronica.in.it e ricavando da essi le pellicole mediante stampa a grandezza naturale su carta da lucido o acetato dei relativi file. Poiché il circuito è stato realizzato utilizzando componentistica a montaggio superficiale, in particolare per i semiconduttori U1, U2, U7, U10 e Q1, avrete bisogno di una stazione saldatrice ad aria calda simile a quella disponibile a catalogo Futura Elettronica con il codice "8220-VTSS210" oppure quella provvista anche di stilo codice "8220-VTSS230". Per il montaggio dei componenti restanti andrà benissimo un saldatore stilo a punta sottile con potenza non troppo elevata (20+25 watt) e del filo di lega saldatrice possibilmente non superiore a 0,7 mm di diametro. Consigliamo inoltre di usare un flussante di ottima qualità che non lascia scorie residue, utile soprattutto se usate leghe saldanti senza piombo che notoriamente aderiscono con difficoltà e possono richiedere del flussante aggiuntivo per ottenere buone saldature. Per posizionare i componenti aiutatevi con delle



Fig. 4

Collegamento
della batteria
LiPo da 1100
mAh codice
"6374-PLIB1100-
2MM".

Fig. 5
Modifica della
scheda Arduino
M0 pro.



pinzette a punta fine possibilmente amagnetiche (inox) altrimenti, magnetizzandosi i componenti vengono attratti, complicandovi il lavoro. Per saldare i semiconduttori con l'aria calda, in particolar modo quelli con case QFN, consigliamo l'utilizzo di una pasta saldante specifica per SMD come quella a catalogo. Spalmate un velo di pasta sulle piazzole, con le pinzette centrate il semiconduttore in posizione, rispettando l'orientamento come indicato nel piano di montaggio, e con il saldatore ad aria calda riscaldate il componente fino alla completa stagnatura. Durante questa fase state attenti che il flusso di aria non sposti il componente oppure lo surriscaldi rischiando di bruciarlo.

Per chi ha poca pratica con la saldatura ad aria calda, consigliamo di iniziare a scaldare, con movimenti circolatori, non solo il componente ma anche l'area circostante ad esso e quando si nota il flussante sciogliersi, di concentrare il calore sul componente fino a saldatura completata. Per eliminare i residui di flussante utilizzate un piccolo pennello a setole rigide o in alternativa uno spazzolino da denti inumidito con alcol isopropilico, oppure un prodotto più specifico come l'anti flussante spray F-02 codice "6935-F02". Per saldare il microcontrollore U3, consigliamo di spalmare le piazzole di flussante e centrare il componente in posizione saldando prima un solo piedino.

Dopo aver controllato la corretta centratura dei

rimanenti piedini alle piazzole saldate il piedino del lato opposto in modo da bloccarlo, dopodiché potete saldare tranquillamente i rimanenti piedini. Cercate di saldare i piedini alternando il lato, in modo da non surriscaldare eccessivamente il componente. Se non siete pratici, fate delle pause per permettere al componente di raffreddarsi. Lo stesso discorso vale per tutti i semiconduttori. Una volta stagnati i componenti attivi potete iniziare a saldare quelli passivi, ricordando di fare attenzione al corretto orientamento dei condensatori al tantalio C14 e C31, che andranno orientati come indicato nel piano di montaggio.

Una volta montati tutti i componenti SMD possiamo iniziare a saldare i componenti a foro passante, ossia quelli a montaggio tradizionale.

Raccomandiamo una certa attenzione nel saldare il connettore USB1, che possiede piedini SMD e foro passante, al fine di non surriscaldarlo. L'ultimo componente da saldare è il display OLED LCD1. Saldare una strip da 4 piedini maschio sul display e una strip femmina da 4 piedini sul circuito stampato, fate attenzione che i piedini dello strip entrino nei rispettivi fori e fissate il display al PCB con due distanziatori M2 da 11 mm e quattro viti da M2x5 mm. Seppure utilissimo per monitorare lo stato di carica della batteria e la corrente assorbita dal carico, il display in realtà è opzionale e la sua assenza non compromette il corretto funzionamento del circuito; potete quindi dedicare di farne a meno. Poiché la batteria 18650 per alcuni utilizzi risulta eccessivamente ingombrante e pesante, sul lato inferiore del circuito stampato (Fig. 4) abbiamo predisposto il montaggio di un connettore a montaggio superficiale JST passo 2 mm per collegare una batteria LiPo da 1.100 mAh che dovrà essere fissata a sua volta al PCB con del nastro biadesivo. Se invece avete bisogno di più autonomia potete usare quella da 5.000 mAh, per esempio quella a catalogo Futura Elettronica con il codice "7893-PLIB5000MAH".

Se la PowerPi verrà utilizzata per alimentare una Raspberry Pi 4, vi informiamo che rispetto alle versioni precedenti scalda parecchio, quindi vi consigliamo di acquistare un kit di dissipatori per il suo raffreddamento.

Fate attenzione a non confondere la presa USB-C, inserite il cavo di alimentazione nella presa TypeC del PowerPi con le versioni precedenti il problema non sussisteva perché il connettore della Raspberry Pi era microUSB, ma comunque non preoccupatevi, intervengono le protezioni e non succede niente, comunque è meglio non rischiare.

Fig. 6
Piedinatura
della porta JTAG.

VCC	1			2	SDWIO / TMS
GND	3			4	SWDCLK / TCLK
GND	5			6	SWO / TDO
KEY	7			8	NC / TDI
GNDDetect	9			10	nRESET

PROGRAMMAZIONE

Completato il montaggio è giunto il momento di programmare la PowerPi.

La prima cosa da fare è caricare il bootloader. Alla destra del display OLED, si trova il connettore JTAG1 a 10 piedini usato per la programmazione e il debug del microcontrollore AT91SAM7S128.

Questo connettore, la cui piedinatura è visibile nella **Fig. 5**, lo utilizzeremo solo una volta per caricare il bootloader nella memoria flash del microcontrollore per permetterci di caricare successivamente il firmware. A tale scopo ci servirà un programmatore hardware tipo Atmel ICE oppure PickIt 4.

In alternativa noi consigliamo l'utilizzo di una scheda Arduino M0 PRO sulla quale è montato il chip EDBG, modificarla e utilizzarla come programmatore a basso costo. La modifica è semplicissima basta dissaldare due resistenze siglate rispettivamente R2 ed R3 e saldare il connettore SWD strip da 2x5 pin passo 1.27 mm, se assente (**Fig. 6**).

Procuratevi un cavetto JTAG da 10 pin oppure realizzatelo utilizzando uno spezzone di cavetto flat riciclando il connettore IDE-ATA 2 a 40 pin passo 2x2,54 mm utilizzato per i vecchi dischi rigidi da computer, di cui utilizzeremo solo 10 conduttori. Alle due estremità monteremo crimpando due connettori IDC da 10 pin passo 1.27 mm, assicurandovi di posizionarli nel verso giusto cioè pin to pin. In alternativa, cercando nei negozi on-line potete acquistarlo già assemblato per pochi soldi. Collegate la scheda Arduino M0 PRO alla porta USB, come in **Fig. 7**, e dal menu a tendina "strumenti", dell'ambiente di sviluppo Arduino



Fig. 7
Programmazione della PowerPi mediante Arduino M0 PRO modificato.

IDE, selezionate la scheda Arduino M0 PRO (programming port) e la corretta porta seriale, successivamente sempre dal menu strumenti selezionate **Scrivi bootloader**.

Se non vengono visualizzati errori, il microcontrollore è stato programmato e l'ambiente Arduino IDE lo riconoscerà come una scheda Arduino M0.

A questo punto scaricate il firmware compilato dal sito web www.elettronica.in.it. Nel file compresso "PowerPi.rar" oltre al firmware "powerpi.hex", il programma "avrdude.exe", il file di configurazione "avrdude.conf", troverete lo script "powerpi.bat", nel

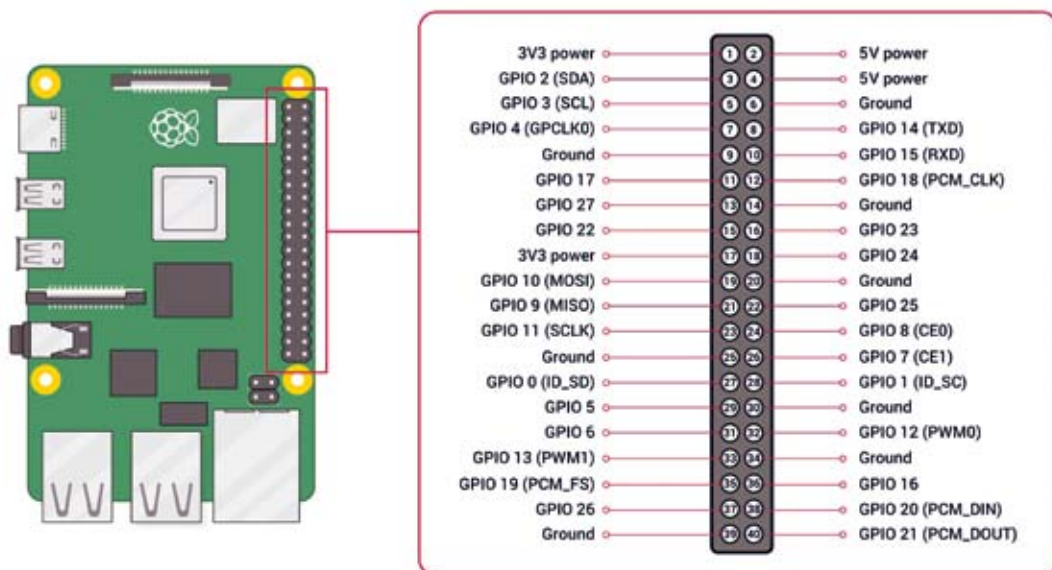


Fig. 8
Corrispondenza tra i piedini GPIO del connettore a 40 pin "BOARD" con quelli "BCM".



La scheda PowerPi applicata a Raspberry Pi mediante il connettore header.



quale dovrete modificare, nella seconda riga, il numero della porta seriale "comportA" con quello assegnato alla vostra scheda dal PC. Per scoprire il numero basta scegliere il sottomenu "porta" nel menu a tendina "Strumenti" di Arduino Ide; ad esempio, se la porta assegnata è la 5 nella seconda riga dovete scrivere "set comportA=5". Programmata la PowerPi passiamo a programmare la Raspberry Pi. Apriamo il terminale "LXterminal" e lanciamo il comando "sudo nano /boot/config.txt". Con il cursore ci spostiamo alla fine del file e aggiungiamo due righe di testo, nella prima riga scrivete "dtoverlay=gpio-poweroff, gpiopin=18, active_low="y", debounce=1500" e nella seconda "dtoverlay=gpio-shutdown, gpio_pin=27". Il primo comando definisce il piedino GPIO 18 come output, che andrà a livello logico basso quando la Raspberry Pi ha finito la sequenza di shutdown informandoci che è pronta per essere spenta. Il secondo comando definisce il piedino GPIO 27 come input, il quale se messo a livello logico basso fa partire la procedura di shutdown. Tenete presente che i comandi "gpio-poweroff" e "gpio-shutdown" non sono presenti in tutte le immagini. Il comando *gpio-shutdown* manca su alcune vecchie immagini, mentre dalla "2017-11-29-raspbian-stretch-lite" sono presenti entrambi. Fate attenzione perché il kernel fa riferimento ai pin del chip Broadcom, indicati come numeri "BCM", e non al numero dei piedini del connettore GPIO header a 40 pin indicati come piedini "BOARD". Ad esempio nel nostro progetto utilizziamo i piedini BOARD 12 e 13 che corrispondono rispettivamente ai piedini GPIO 18 e 27 "BCM", vedi **Fig. 8**. Se desiderate monitorare lo stato delle porte GPIO da terminale lanciate il comando "sudo gpio readall". Se avete una Raspberry Pi 4, probabilmente a monitor comparirà il seguente messaggio "Oops - unable to determine board type..."

model: 17". Per risolvere il problema, accertatevi che "wiringPi" sia aggiornato alla versione 2.52 lanciando da terminale il comando "gpio -v"; in caso contrario dovete aggiornarlo lanciando i seguenti comandi:

```
cd /tmp
wget https://project-downloads.drogon.net/wiringpi-latest.deb
sudo dpkg -i wiringpi-latest.deb
```

CONCLUSIONI

Terminiamo questo articolo ricordando, a chiunque sia interessato, che sul nostro sito www.elettronicaonline.it è possibile scaricare tutto il firmware, la BOM (Bill Of Material) dettagliata e i file per la stampa del PCB occorrente a realizzare il progetto. Per la realizzazione del circuito stampato vi consigliamo il servizio di realizzazione PCB (cod. PCBPRODUCTION) disponibile sul sito www.futurashop.it.

Abbiamo cercato di progettare un dispositivo versatile che implementi in un singolo circuito di dimensioni contenute, il massimo delle funzionalità, cercando per quanto possibile di semplificarne l'hardware. Speriamo, così, di aver soddisfatto le esigenze di tutti, non solo di quelli che hanno bisogno di un UPS per alimentare una Raspberry Pi, ma anche di chi ha bisogno di un'alimentazione sicura e senza interruzioni a 5V per realizzare progetti come antifurti, circuiti per la videosorveglianza, l'automazione sia in ambito domestico che in campo automobilistico.

Se desiderate ottenere ulteriori informazioni sul funzionamento potete visitare il sito informatici.pietrocoppo.net se invece per il vostro progetto avete bisogno delle modifiche del circuito, potete mandare la vostra richiesta tramite il sito design.futuragroupsrl.it. □



Cosa occorre?

Il progetto PowerPi (cod. POWERPI) è disponibile già montato a Euro 69,00.
I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

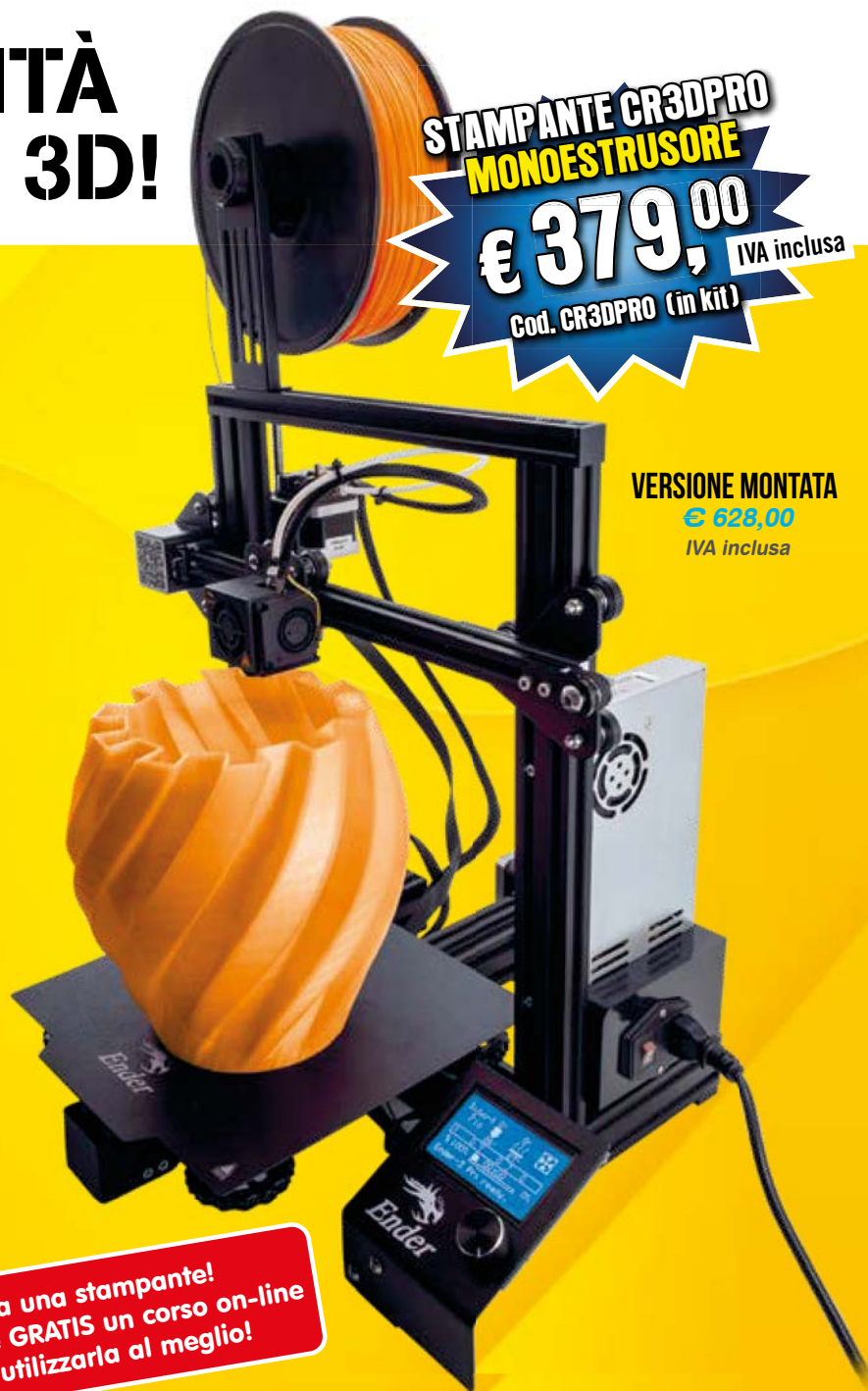
Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 - <http://www.futurashop.it>

ULTIME NOVITÀ DAL MONDO 3D!

CR3DPRO

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

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



VERSIONE MONTATA
€ 628,00
IVA inclusa

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



VERSIONE MONTATA
€ 1148,00
IVA inclusa

CRCP3IN1

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

**CRCP3IN1
MACCHINA 3 IN 1**

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

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

OSCILLOSCOPIO PALMARE 2 CH TOUCH-SCREEN

dell'ING. MIRCO SEGATELLO



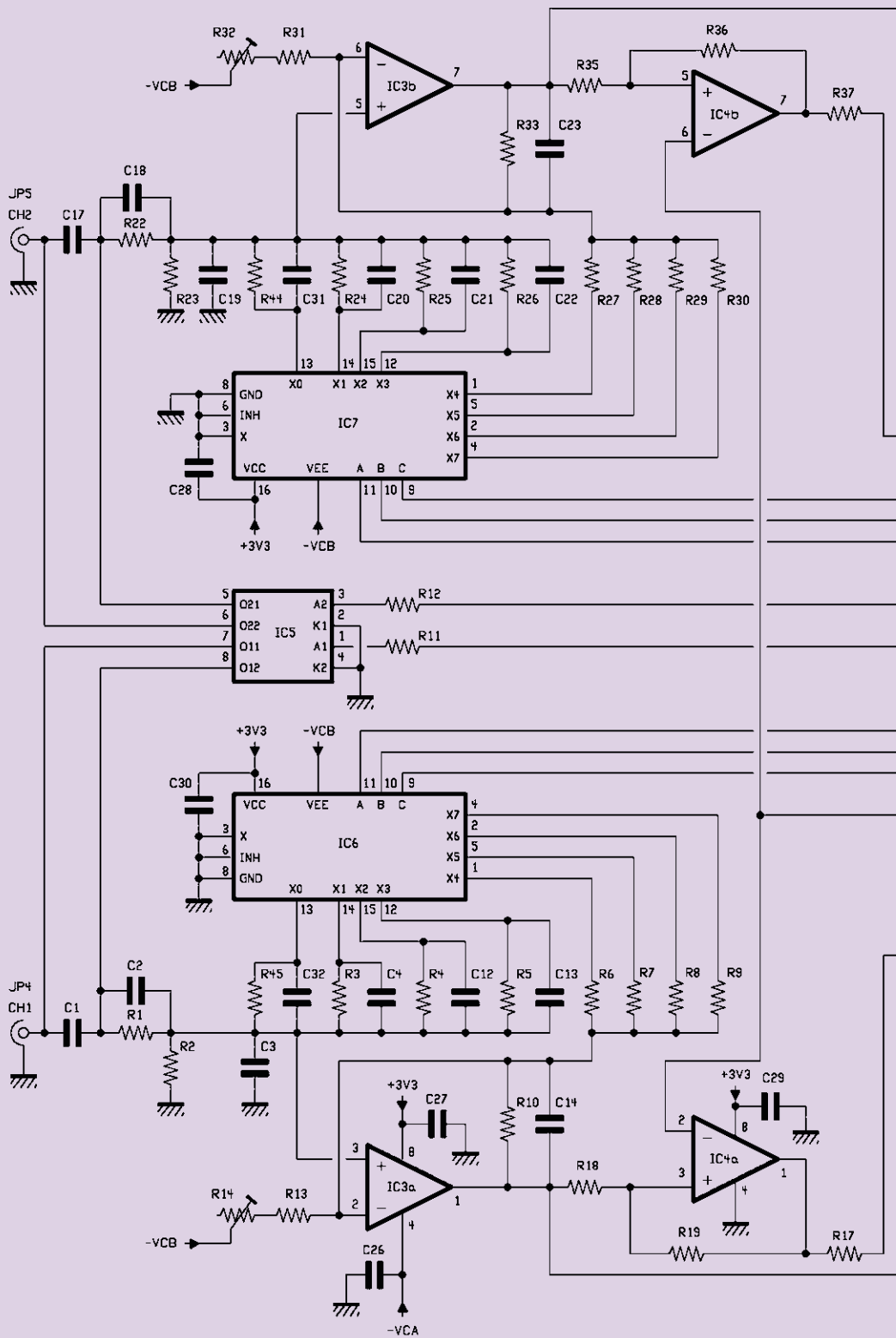
Costruiamo un valido strumento portatile per effettuare misure sul campo, come nel proprio laboratorio; compatto e di semplice utilizzo grazie al display tattile, si gestisce con un pennino.

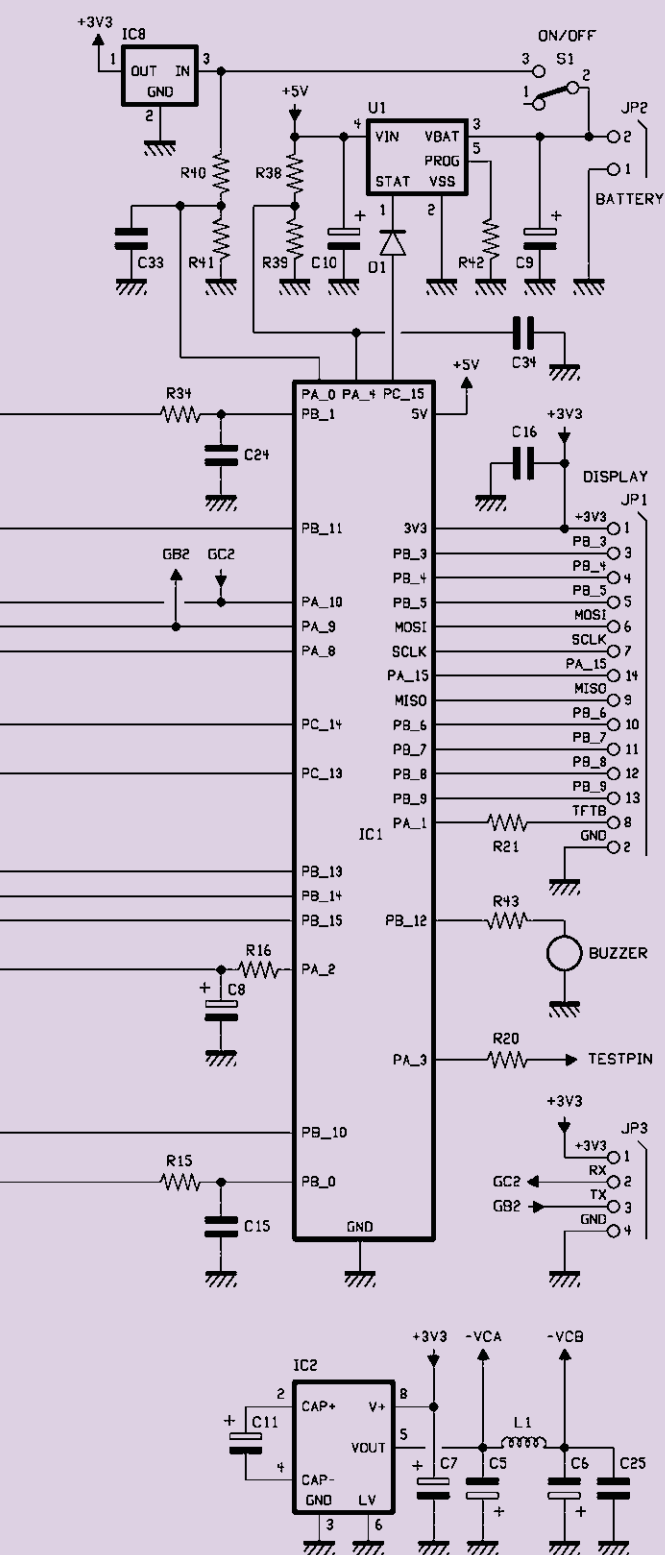
A close-up photograph of a person's hand holding a white probe, which is connected to a black oscilloscope. The oscilloscope is partially visible on the left side of the frame. Below the hand, another probe with a grey handle and a black cable is lying on a purple surface. The probe handle has 'X1' and 'X10' markings. The background is a solid purple color.

D

opo il multimetro, l'oscilloscopio è senz'altro uno strumento che non può mancare nel laboratorio e, complice un'elettronica sempre più a buon mercato, non è più uno strumento irraggiungibile. Infatti oggi, grazie sia alla riduzione dei costi di produzione derivanti

dalla produzione di massa (e orientale, soprattutto) è possibile, anche per un hobbista, attrezzarsi con un piccolo ma ben fornito laboratorio, completo di strumenti che le nuove tecnologie hanno reso alla portata di tutti. Questo lo abbiamo anche verificato personalmente quando, già nel 2011 (nel numero 160 della nostra rivista) presentammo il progetto di un oscilloscopio palmare, certo non un prodotto dalle grandi caratteristiche e





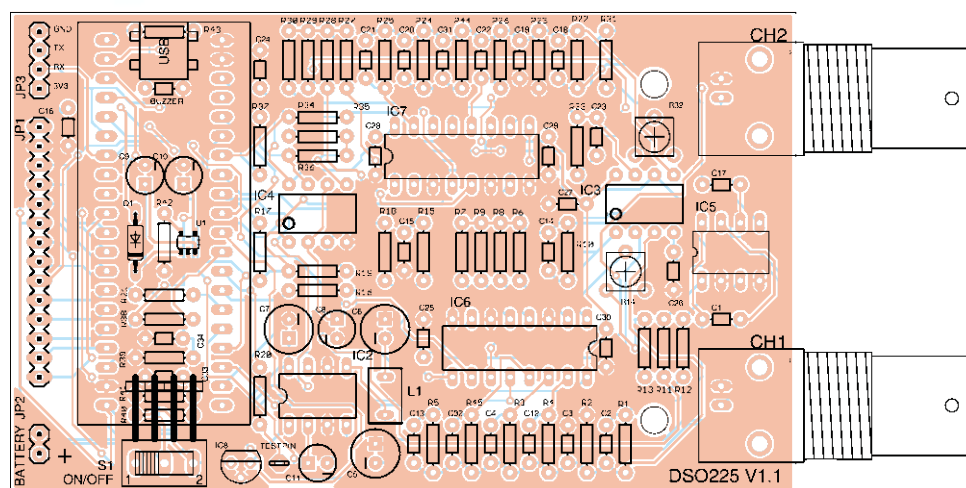
per questo pensato per un uso non professionale. E torniamo a dimostrarvelo in queste pagine, dove proponiamo una nuova versione di oscilloscopio digitale, rivisitato e dotato di caratteristiche decisamente migliorate. L'idea ci è venuta da quando è stato possibile disporre a buon mercato di un display a colori con funzionalità touch; fin da subito ci siamo chiesti che cosa ne avremmo potuto fare e l'idea è caduta proprio su di un oscilloscopio. Per la parte di acquisizione, inizialmente, avevamo preso in considerazione il microcontrollore usato nelle schede Arduino, magari interfacciato con un ADC esterno, viste le limitate caratteristiche del ADC interno dell'MCU Atmel 328P. Ma anche considerando altre board del mondo Arduino il risultato era sempre lo stesso: caratteristiche non adeguate o costi troppo elevati. Alla fine abbiamo ripiegato su una economica board basata su un microcontrollore a 32bit della famiglia ST, per la precisione un STM32F103 che, anche se di fascia bassa, ha delle caratteristiche di tutto rispetto, prima fra tutte il fatto di avere integrati due convertitori analogico-digitale ad alta velocità ed una CPU con architettura a 32 bit funzionante a 72 MHz. Scelti i due componenti principali tutto il resto è stato progettato di conseguenza, con particolare attenzione a scegliere componenti di facile reperibilità e a buon mercato. Dalle caratteristiche si capisce che il principale impiego di questo oscilloscopio è nella banda audio, spingendosi al massimo a frequenze sino a 100 kHz, più che sufficienti per la maggior parte dei vostri progetti basati su Arduino. Il pregio di questo oscilloscopio è senz'altro la portabilità, viste le dimensioni ridotte e la lunga durata con alimentazione a batteria, che può essere estesa riducendo al minimo la retroilluminazione del display, infatti la corrente assorbita dall'intero circuito è di 90 mA che però si riduce a soli 60mA con la retroilluminazione al minimo. Frequenza di campionamento superiori ci avrebbero obbligato all'utilizzo di un convertitore ADC esterno con memoria ad alta velocità ed uno stadio d'ingresso molto più sofisticato, tanto da potersi realizzare solo con procedimenti professionali.

SCHEMA ELETTRICO

Entriamo ora nel dettaglio della progettazione analizzando lo schema elettrico che, tutto sommato considerando il risultato finale, non è neppure troppo complesso a tutto beneficio di una sua facile realizzazione. Nel progettare abbiamo tenuto conto che dovrà essere realizzato da un

CARATTERISTICHE TECNICHE

- Numero di canali: 2
- Frequenza di campionamento: 2X2,5Mps
- Risoluzione: 12 bit
- Banda passante: >700KHz
- Impedenza di ingresso: >500Kohm
- Base dei tempi: da 10 s/div a 1 µs/div
- Trigger hardware: rising, falling (con pre e post trigger)
- Trigger modalità: AUTO, NORMAL, ONCE, ROLL
- Display: touch-screen 2,8"
- Dimensioni: 130x70x25mm



Elenco Componenti:

R1, R22: 560 kohm
 R2, R5, R23, R26: 470 kohm
 R3, R24: 39 kohm
 R4, R25: 120 kohm
 R6, R38, R39, R40, R41, R27:
 10 kohm
 R7, R28: 1,3 kohm
 R8, R29: 499 ohm
 R9, R43, R30: 220 ohm
 R10, R33: 4,7 kohm
 R11, R12: 680 ohm
 R13, R31: 8,2 kohm
 R14, R32: 2 kohm Trimmer M.O.
 R15, R20, R37, R17, R34:
 470 ohm
 R16: 47 kohm
 R18, R35: 100 ohm
 R19, R36: 100 kohm
 R21: 1 kohm

R42: 3,9 kohm
 R44, R45: 22 kohm
 C1, C17: 100 nF poliestere
 C2, C18: 47 pF ceramico
 C3, C19: 27 pF ceramico
 C4, C20: 820 pF ceramico
 C5, C6: 100 uF 16 VL
 elettrolitico
 C7: 47 uF 16 VL elettrolitico
 C8, C9, C10: 4,7 uF 16 VL
 elettrolitico
 C11: 22 uF 16 VL elettrolitico
 C12, C21: 270 pF ceramico
 C22, C13: 68 pF ceramico
 C23, C14: non montati
 C24, C15: 330 pF ceramico
 C16, C25, C26, C27, C28,
 C29, C30: 100 nF
 multistrato

C31, C32: 1,2 nF ceramico
 C33, C34: 220 nF ceramico
 D1: 1N4148
 U1: MCP7383T-2AT/OT
 IC1: STM32F103 bluepill
 board
 IC2: ICL7660CPA
 IC3, IC4: MCP6022
 IC5: ASSR-1228
 IC6, IC7: M74HC4051B1
 IC8: LP2950CZ 3,3V
 L1: Induttanza 1 mH 14 ohm
 S1: Mini deviatore a slitta
 da c.s.
 BUZZER: Buzzer passivo
 Ø 10 mm
 JP1: Strip femmina 14 poli
 JP2: Connettore maschio
 XH 2,54 mm da c.s.

JP3: Strip maschio
 4 poli
 JP4, JP5: Connettore BNC
 90° DA C.S.

Varie:
 - Batteria ai polimeri
 di litio 3,7 V 1100 mA
 - Display Touch 2,8"
 - Cavo con connettore
 JST 2 PIN 2,54 mm
 - Zoccolo 4+4
 - Zoccolo 4+4
 - Distanziale Nylon
 M3x15 F/F
 - vite nylon M3x6
 - Circuito stampato
 S1533
 (104x64 mm)

hobbista e quindi siamo venuti a compromessi pur di evitare di complicare troppo il circuito. Ribadiamo che si tratta però di un circuito la cui realizzazione necessita una certa competenza. Sulla destra il componente IC1, ovvero la scheda con STM32F103 disponibile già assemblata, che ci risparmia la fatica di inserire nello stampato il microcontrollore, il circuito di clock e la porta USB, che sarebbero tutti componenti SMD di non facile saldatura. Essendo un progetto pensato per gli sperimentatori, tutti i componenti utilizzati sono nel formato classico, tranne il circuito integrato per la ricarica della batteria, che viene fornito solo nel formato SMD ma fortunatamente ha soli 5 pin; si tratta dell'integrato di sigla U1 programmato tramite il resistore R42 per una corrente di ricarica di 270 mA. A questa corrente va sommata quella assorbita dal circuito, per un totale di circa 370 mA, valore conservativo per non sovraccaricare la porta USB dei PC limitata a 500 mA, in questo modo garantiamo l'utilizzo dello strumento per un tempo illimitato in sicurezza, anche durante la ricarica della batteria. Lo stadio di alimentazione è composto da un semplice stabilizzatore LDO che permette di ottenere i 3,3 volt dalla tensione della batteria. Il componente da noi previsto ha sigla LP2950 e ha una caduta di tensione interna di circa 400 mV pertanto l'alimentazione a 3,3 volt sarà fornita solo se la tensione della batteria sarà superiore a 3,7 volt, non consentendo di sfruttare al massimo la sua carica. In sostituzione consigliamo il componente MCP1700-33 che permette di sfruttare la batteria sino a 3,4 volt aumentando l'autonomia di circa due ore. La tensione negativa necessaria per gestire i segnali bipolari viene ricavata con l'integrato IC2 un invertitore di tensione molto comune, la cui tensione in uscita è filtrata dai componenti C5, C6 e L1. Abbiamo previsto due partitori di tensione per la misura della tensione della USB e della batteria utili per monitorare l'alimentazione del circuito. Il segnale di ingresso può essere accoppiato in AC o DC grazie al relè a stato solido di sigla IC5, mentre l'amplificatore operazionale IC3 in configurazione non invertente è adibito all'amplificazione del segnale. Non abbiamo previsto il vero e proprio accoppiamento GND che comunque è ottenibile scollegando il segnale di ingresso e selezionando GND nell'apposito menu, che in pratica porta semplicemente lo stadio di ingresso alla sua minima amplificazione. Gli integrati IC6 e IC7 sono degli interruttori a stato solido utilizzati per variare l'attenuazione del segnale e program-

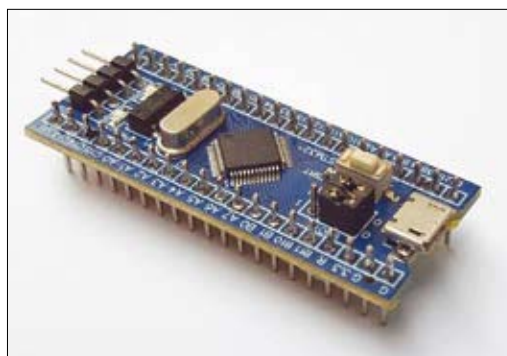


Fig. 1
La scheda
STM32
preparata per
l'installazione.

mare l'amplificazione dell'operazionale. Le resistenze di questo stadio sono calibrate per la classica sequenza 1-2-5, mentre i condensatori permettono la compensazione delle capacità parassite dello stampato e dei componenti. L'amplificatore operazionale IC4 viene usato per ricavare il segnale di trigger, confrontando la tensione proveniente dallo stadio di ingresso con una di riferimento generata dal microcontrollore tramite i componenti R16 e C8. Entrambi gli operazionali utilizzati sono del tipo rail-to-rail per assicurare di gestire segnali sino alla tensione di alimentazione come richiesto dall'ADC del microcontrollore. Gli unici due trimmer presenti R14 e R42 sono usati per la taratura dell'offset introdotto dallo stadio di amplificazione. Abbiamo anche previsto un buzzer per la gestione del classico beep quando si interagisce con il display ed un'uscita (TEST PROBE) per la taratura della sonda, utile anche per verificare "al volo" che tutto funzioni correttamente; si tratta di un semplice segnale ad onda quadra alla frequenza di 1 kHz con duty-cycle del 50% ed ampiezza di 3,3 volt. Il connettore JP2 accoglie la batteria al litio, JP1 serve per la connessione del display e JP3 è la porta di comunicazione seriale per il caricamento del firmware. Per semplificare al massimo lo stadio di ingresso e poterlo realizzare con un solo operazionale per canale l'impedenza di ingresso non è la classica da 1Mohm, bensì il suo valore cambia da portata a portata rimanendo comunque sempre superiore a 500Kohm (1Mohm solo nelle portate più basse). La cosa di per sé non è gravissima, ma di fatto impedisce di usare le sonde di attenuazione che sono calibrate per un'impedenza di 1Mohm. I resistori sono tutti della serie E12 di facile reperibilità, l'utilizzo di resistori all'1% di precisione oppure un'accurata selezione con un multimetro di precisione, permette di migliorare la precisione generale del dispositivo.

➔ **Fig. 2**
Circuito assemblato
senza display.



REALIZZAZIONE PRATICA

Veniamo ora alla realizzazione pratica, che richiede una certa manualità e attenzione, vista la densità dei componenti e la complessità dello schema. Per lo stampato dovete partire dai file gerber (PCB_DS0225V11.ZIP) scaricabili assieme ai file di questo numero della rivista usufruendo, ad esempio, del servizio stampa PCB offerto dalla Futura Elettronica alla pagina web <https://www.futurashop.it/produzionePCB>.

Non appena in possesso dello stampato procedete alla saldatura del componente SMD U1 e poi successivamente tutti gli altri partendo da quelli a più basso profilo come diodi e resistenze, per finire con le due prese BNC. Il componente denominato IC1 è la board con STM32 ed andrà preparata saldando i due strip marchio forniti a

corredo leggermente meno sporgenti ed abbinandoli a due strip femmina in versione ribassata da saldare sulla scheda principale. Per ridurre l'altezza della scheda dovrete anche ritoccare i due jumper, come visibile in **Fig. 1**, otterremo così un oscilloscopio con più basso profilo; nelle **Fig. 2** e **Fig. 3** potete vedere utili dettagli dell'insieme assemblato, prima di essere introdotto nel contenitore in plastica.

Il display dovrà potersi fissare sul connettore JP1 in modo che il suo stampato sia a 15 mm dallo stampato principale. Il fissaggio dal lato opposto sarà affidato a 2 distanziatori da 15 mm in plastica posizionati nei due fori predisposti, come visibile nella **Fig. 4**.

➔ **Fig. 3**
Circuito
assemblato.



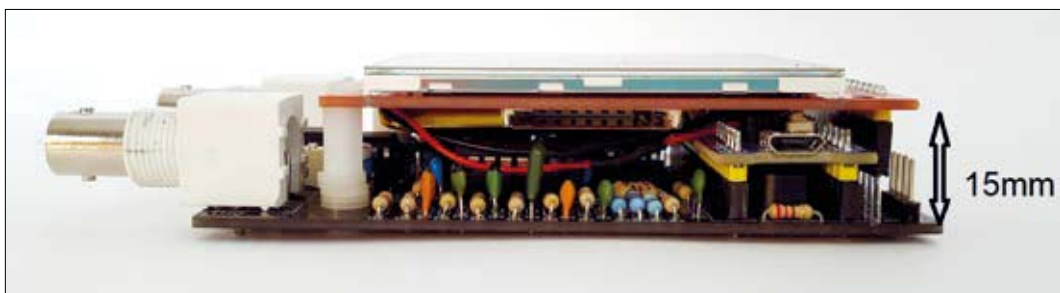


Fig. 4
Particolare
vista frontale
e posteriore.

Prestate la massima attenzione ai componenti saldati tra i connettori di IC1. La batteria è posizionata tra il circuito ed il display (come in **Fig. 2**), pertanto assicuratevi di scegliere dei condensatori elettrolitici sufficientemente bassi da assicurare abbastanza spazio. Per completare l'opera abbiamo previsto un contenitore dedicato da realizzare con una stampante 3D. Si tratta di un contenitore a guscio composto da due parti una per accogliere il circuito e l'altro che svolge la funzione di coperchio. Il fissaggio è ad incastro sul

lato sinistro, mentre sul lato destro sarà bloccato con i dadi delle boccole BNC. Abbiamo previsto un alloggiamento per accogliere il pennino fornito in dotazione con il display touch-screen, così si evita di perderlo quando l'oscilloscopio non viene usato. A seconda della precisione di stampa potrebbe essere necessario eseguire qualche limatura in corrispondenza delle sbavature o per aggiustare gli incastri, in quanto lo stampato deve inserirsi con precisione nel case; non abbiamo previsto alcuna antiestetica vite di fissaggio.

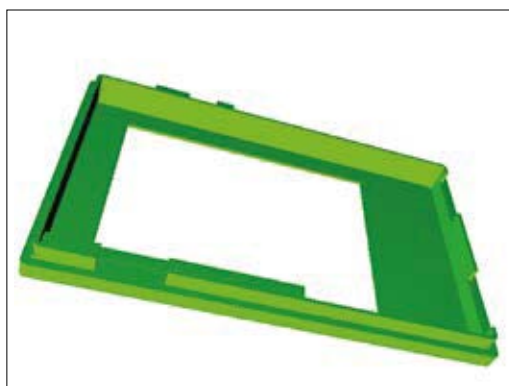
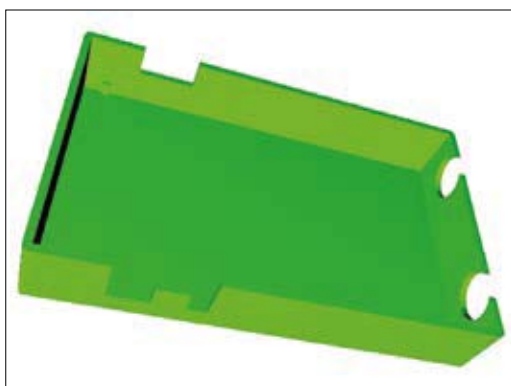


Fig. 5
Parte
inferiore e
superiore del
contenitore.



Fig. 6
Oscilloscopio
completato
ed inserito nel
contenitore.

Tabella 1
Collegamenti
elettrici per
aggiornamento del
firmware.

DSO225 JP1	Convertitore USB/seriale
GND	GND
TX	RX
RX	TX
3V3	Vcc (3,3V)

Per assicurare che il coperchio si inserisca correttamente è stata impostata una riduzione delle dimensioni di 0,1 mm in fase di realizzazione dello slicer. Assieme ai file di questa rivista troverete tutti i file del case in formato .stl adatti ad essere modificati secondo le vostre necessità, con le indicazioni per realizzare lo slicer, ed i file di stampa in formato .gcode; tutti i file sono forniti in due versioni con bordi arrotondati oppure ad angolo retto. Nella **Fig. 5** potete vedere i rendering delle due parti componenti il contenitore da stampare in 3D. In **Fig. 6**, invece, potete vedere il lavoro ultimato, cioè l'oscilloscopio inserito nel contenitore stampato a partire dai file .stl con del PLA nero. Un approfondimento merita la scelta del tipo di batteria utilizzata: si tratta di una batteria agli ioni di litio con circuito di sicurezza ovvero una particolare versione con presente al suo interno un circuito appositamente progettato per evitare che la batteria si danneggi in caso di uso errato. Questo circuito protegge la batteria da sovra

cariche, sotto scariche e corto circuiti, ovvero tutto quello che potrebbe succedergli. Come sapete le batterie al litio sono più delicate rispetto alle altre batterie, arrivando in alcune situazioni limite anche ad esplodere; il circuito integrato al suo interno protegge sia la batteria che il vostro circuito da eventuali danneggiamenti, questo è ancora più importante in un circuito autocostruito. Il fatto di usare il microcontrollore installato in una board non ci ha permesso di realizzare un sistema di spegnimento automatico nel caso la batteria si scarichi del tutto, lasciando l'utilizzatore con l'incarico di spegnerlo quando necessario. Questa particolare batteria ci assicura che in ogni caso la batteria ed il circuito non vengano danneggiati, anche nel caso di una dimenticanza di dispositivo acceso. La batteria viene fornita con il connettore staccato e, vista la dimensione "lillipuziana" di questo connettore, nell'elenco dei componenti abbiamo riportato un connettore alternativo di più generose dimensioni con relativo connettore maschio da saldare sulla scheda con il classico passo di 2,54 mm.

INSTALLAZIONE DEL FIRMWARE

Il software è particolarmente complesso ed ha richiesto l'utilizzo di diverse librerie spesso modificate per le particolari esigenze. Anche se il software è stato realizzato con l'IDE di Arduino l'installazione di tutte le dipendenze e delle librerie con la giusta versione potrebbe non essere agevole, pertanto la procedura di programmazione prevede di bypassare l'utilizzo dell'IDE di Arduino per la realizzazione del file compilato, che invece forniremo già pronto con i dovuti settaggi. Come vedremo in questo caso la procedura risulterà decisamente più snella, anche se sarà necessario disporre di un convertitore USB/seriale compatibile con MCU a 3,3 volt; qualsiasi prodotto andrà bene in quanto ci serviranno solo le linee TX ed RX. Prima di eseguire la procedura conviene rimuovere il display ed anche i vari circuiti integrati dai rispettivi zoccoli, per i collegamenti elettrici fate riferimento alla **Tabella 1**. Notate che il TX del DSO va sull'RX del converter e viceversa, come richiesto nella comunicazione seriale DTE-DTE, ovvero un collegamento incrociato. Su questa scheda sono presenti due jumper usati per selezionare il modo di funzionamento, per abilitare la funzione di boot da seriale dovrete posizionare il jumper di nome BOOT0 su 1 e BOOT1 su 0 e successivamente premere il pulsante di reset. Dovete scaricare e installare il software STM32 Flash loader

Fig. 7
Software STM32
Flash loader
demonstrator.





Fig. 8
Software STM32 Flash loader demonstrator riconoscimento MCU.

demonstrator disponibile sul sito ufficiale della ST e seguire la procedura seguente: avviate il software e selezionate la COM del convertitore (**Fig. 7**), verificate che il software riconosca il microcontrollore (**Fig. 8 e 9**), quindi aprite il firmware compilato con estensione .bin ed avviate la programmazione (**Fig. 10**), attendete il completamento delle operazioni, **Fig. 11**. Terminata la programmazione staccate il programmatore e riposizionate entrambi i jumper in posizione 0, inserite la batteria ed eseguite la prima accensione.

COLLAUDO E TARATURA

La taratura è molto semplice e consiste nel regolare i due trimmer per compensare l'offset presente nello stadio di ingresso. Non collegate nulla agli ingressi e selezionate l'accoppiamento GND quindi ruotate i due trimmer per avere le due tracce posizionate sulla linea dello zero. Visto l'impiego per il quale è stato pensato, l'obiettivo non era quello della massima precisione bensì di un circuito facilmente realizzabile a livello hobbistico. Mentre per i tempi è facile ottenere una buona precisione in quanto legati al clock di sistema che è basato su di un quarzo, per le ampiezze la cosa è differente. Se disponete di un generatore di fun-



Fig. 9
Software STM32 Flash loader demonstrator lettura memoria interna.

zioni e di un voltmetro di riferimento in alternata potete eventualmente ritoccare qualche valore dei resistori usati per selezionare il guadagno dello stadio di ingresso; comunque il microcontrollore STM32F103 usa come tensione di riferimento per lo stadio analogico la stessa di quella di alimentazione, che a sua volta è determinata dallo



Fig. 10
Software STM32 Flash loader demonstrator caricamento firmware.

Fig. 11
Software STM32
Flash loader
demonstrator
caricamento
completato.



stabilizzatore di tensione LDO, il quale, quindi, influisce sulla precisione. Particolare attenzione dovrà invece essere posta nei condensatori dello stadio di amplificazione che compensano le capacità parassite dello stampato e dei circuiti integrati utilizzati; questi condensatori influiscono sulla banda passante dello stadio d'ingresso. La verifica può essere effettuata utilizzando un generatore ad onde quadre visualizzando sul display il relativo segnale per ciascuna delle portate disponibili. In **Fig. 12** potete vedere i tre casi possibili: nella situazione B il condensatore ha valore troppo alto, nel caso C il valore è troppo basso e nel caso A il valore è corretto. Nella **Tabella 2** riportiamo le corrispondenze tra la portata ed i componenti coinvolti.

Tabella 2
Correlazione tra
componenti e
guadagno.

GAIN	R (CH1)	C (CH1)	R (CH2)	C (CH2)
10v/div	R45	C32	R44	C31
5V/div	R3	C4	R24	C20
2V/div	R4	C12	R25	C21
1V/div	R5	C13	R26	C22
0,5V/div	R6	-	R27	-
0,2V/div	R7	-	R28	-
0,1V/div	R8	-	R29	-
0,05V/div	R9	-	R30	-

IL FIRMWARE

lo sketch per il progetto è alquanto complesso e consta di oltre 2.000 righe di programma e più di 3000 righe di librerie, pertanto impossibile da riportare in queste pagine (vi invitiamo a scaricarlo direttamente dal sito della rivista); anche le librerie sono state modificate rispetto le originali al fine di ottimizzarle e renderle più efficienti in dispositivi STM32 e sono state integrate delle funzioni aggiuntive. Per ridurre all'essenziale la parte hardware abbiamo spostato quante più funzioni a livello software con il problema che l'unica CPU a disposizione deve svolgere tutte le funzioni di acquisizione, rilevamento del trigger, gestione del display con relativo touch ed analisi dell'alimentazione, tutto in real time e con tempi di reazione molto ridotti.

L'utilizzo di un display con touch-screen ci ha permesso di riprogettare da zero uno strumento che non avesse alcun pulsante se non l'interruttore di accensione. Ogni menu e ciascuna funzione è gestibile esclusivamente via touch-screen, risparmiandoci la progettazione e realizzazione di una tastiera che avrebbe comportato un rilevante ingombro ma, cosa più importante, possiamo ridefinire ed aggiungere funzioni semplicemente agendo via software, modificando i vari menu. L'accesso diretto ad una funzione agendo direttamente su un pulsante o un encoder rotativo fisico è sicuramente più immediato e preferibile in strumentazione professionale.

Per l'acquisizione è stata usata la tecnica del buffer circolare con DMA pertanto i due ADC a disposizione sono stati configurati per salvare i loro valori usando l'accesso diretto alla memoria (Direct Memory Access) continuativamente in un buffer circolare la cui scrittura viene bloccata solo in presenza di un evento di trigger, permettendo di

ottenere valori precedenti e successivi all'evento di trigger. L'uso del DMA ci ha permesso di non impegnare la CPU per l'acquisizione che invece è occupata nella gestione del display e del touch. Sempre nei riguardi dell'acquisizione abbiamo previsto un sistema di interpolazione dei campioni che incrementa in modo artificioso la frequenza di campionamento al fine di ottenere una base dei tempi che si spingesse sino a un microsecondo per divisione, a tal proposito potete vedere nelle **Fig. 13** e **Fig. 14** un'onda triangolare a 100 kHz e un'onda sinusoidale a 200 kHz. Avevamo provato anche un'interpolazione più sofisticata denominata $\sin(x)/x$ ma i tempi di elaborazione sono risultati troppo lunghi.

Per quanto riguarda la base dei tempi, oltre i 100 ms/div. abbiamo implementato la funzione ROLL Mode che non prevede l'utilizzo del DMA bensì di un campionamento diretto del segnale. Mentre con la tecnica di prima dovevamo attendere il completamento del riempimento del buffer di acquisizione, in questo caso il dato campionato è subito visualizzato, con un effetto di "rotolamento" come ad esempio avviene nei cardiogrammi.

Particolare attenzione è stata riservata alla sezione relativa alle misurazioni sul segnale sia in modalità automatica che manuale. Le misure automatiche permettono di ricavare i valori del valore medio, picco-picco, RMS, frequenza e duty-cycle sia per il canale 1 che 2, mentre le misure manuali di tempo e ampiezza sono gestite con dei merker che possono essere posizionati agendo direttamente sul touch del display. Siamo riusciti ad implementare la funzione di AUTOSET (solo per il canale 1) che se attivata permette di ricercare la corretta base dei tempi e i volt/div ottimali; questa funzione potrebbe fallire nel caso di segnali particolarmente complessi o nei quali la componente di offset è particolarmente rilevante. Sarà una delle funzioni che in futuro vorremmo cercare di migliorare. Per quanto riguarda la gestione dell'alimentazione il microcontrollore verifica in continuazione la tensione sulla USB e la tensione sulla batteria e se quest'ultima dovesse risultare insufficiente metteremo lo strumento in una modalità a basso consumo ed avviseremo l'utente con un beep che la batteria è da ricaricare; l'autoprotezione interna della batteria provvederà a staccare l'alimentazione prima che la batteria vada in sovra scarica.

UTILIZZO PRATICO

Se avete già dimestichezza con un oscilloscopio, l'utilizzo di questo strumento è praticamente

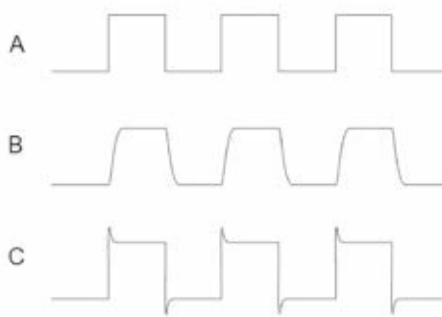


Fig. 12
Compensazione
stadio di
amplificazione.

immediato; chi volesse apprendere qualcosa in più potrà leggere l'articolo sull'oscilloscopio Hantek DSO4202 pubblicato sul numero 247 della rivista, oppure l'articolo sull'oscilloscopio palmare Hantek serie 2000 pubblicato sul numero 245 della rivista. Abbiamo inoltre predisposto un breve video, che meglio di tutto esplica il funzionamento di questo oscilloscopio. Le due principali funzioni di acquisizione sono chiamate AUTO e NORM e riguardano la modalità con la quale avviene l'avvio del campionamento, nel primo caso si attende l'evento di trigger e se questo non avviene entro un certo tempo si forza l'evento di trigger, nel secondo caso invece si attende necessariamente l'evento di trigger. La funzione ONCE permette di eseguire un'unica scansione in seguito ad un evento di trigger e di visualizzare i relativi valori sul display per un tempo indefinito. L'utilità è quella di catturare eventi non ripetitivi come ad esempio la comunicazione seriale tra una scheda Arduino ed un dispositivo esterno, ma anche il caso in cui si voglia analizzare il segnale elettrico generato da un pulsante nel momento in cui lo si preme. Per quanto riguarda il trigger possiamo scegliere sia il livello sia la pendenza oltre al canale.

In **Tabella 3** abbiamo riassunto tutti i menu con le relative funzioni svolte; in futuri sviluppi del software potremmo apportare alcune modifiche

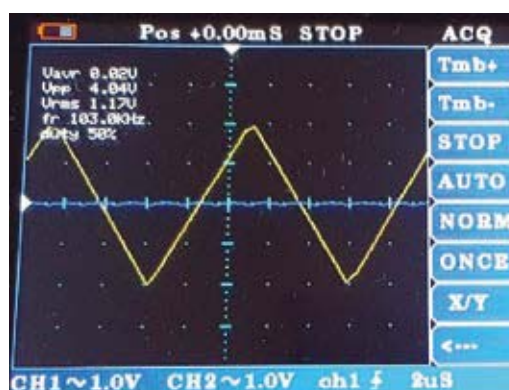


Fig. 13
Onda
triangolare
alla frequenza
di 100 kHz.

oppure aggiungere nuove funzioni. L'utilizzo di un display touch ci permette proprio questi aggiornamenti, senza dover poi agire a livello hardware

aggiungendo o togliendo pulsanti fisici. Una leggera riduzione della reattività dello strumento è rilevabile quando la base dei tempi è posizionata

Tabella 3
Menu e relative
funzioni.

MENU MAIN		
CH1	ON	Abilita la traccia
	OFF	Disabilita la traccia
	V/d+	Incrementa i Volt/div
	V/d-	Decrementa i Volt/div
	AC	Imposta accoppiamento AC
	DC	Imposta accoppiamento DC
	GND	Pseudo accoppiamento GND
	←	Ritorna al menu principale
CH2	ON	Abilita la traccia
	OFF	Disabilita la traccia
	V/d+	Incrementa i Volt/div
	V/d-	Decrementa i Volt/div
	AC	Accoppiamento AC
	DC	Accoppiamento DC
	GND	Pseudo accoppiamento GND
	←	Ritorna al menu principale
TRG	Lev+	Alza il livello del trigger
	Lev-	Disabilita la traccia
	Rise	Trigger su fronte di salita
	Fall	Trigger su fronte di discesa
	Pos>	Trasla la traccia a destra
	Pos-	Trasla la traccia a sinistra
	C1-2	Fonte del trigger CH1 o CH2
	←	Ritorna al menu principale
ACQ	Tmb+	Alza il livello del trigger
	Tmb-	Disabilita la traccia
	STOP	Trigger su fronte di salita
	AUTO	Trigger su fronte di discesa
	NORM	Trasla la traccia a destra
	ONCE	Trasla la traccia a sinistra
	X/Y	Funzione non ancora attivata
	←	Ritorna al menu principale
MEAS	Aut1	Misure automatiche su CH1
	Aut2	Misure automatiche su CH2
	MKV1	Merker di tensione su CH1
	MKV2	Merker di tensione su CH2
	MKT	Merker di tempo
	←	Ritorna al menu principale
ASET	Abilita funzione AUTASET	
MENU	EXPO	Abilita esportazione su USB
	BRG+	Aumenta illuminazione
	BRG-	Diminuisce illuminazione
	BEPP	Abilita/disabilita beep touch
	←	Ritorna al menu principale

su 50 o 100 ms/div a causa dei tempi necessari a rilevare il trigger e l'acquisizione.

In **Fig. 15** potete vedere un esempio di come si presenta la schermata dell'oscilloscopio con le relative indicazioni sui dati visualizzati. Abbiamo anche previsto la possibilità di esportare (in tempo reale) i dati acquisiti, in un formato facilmente leggibile con Serial Monitor di Arduino o altro software idoneo alla lettura di dati dalla seriale tipo PuTTY. È sufficiente connettere l'oscilloscopio tramite la porta USB ed installare gli opportuni driver ed attivare l'esportazione con il pulsante EXPO. Questi dati possono essere importati in un foglio di calcolo e post-elaborati per ulteriori informazioni.

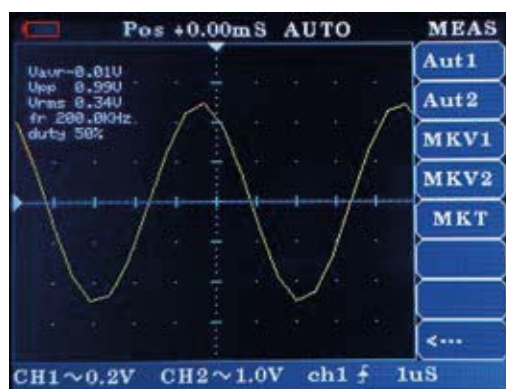


Fig. 14
Onda
sinusoidale
alla frequenza
di 200 kHz.

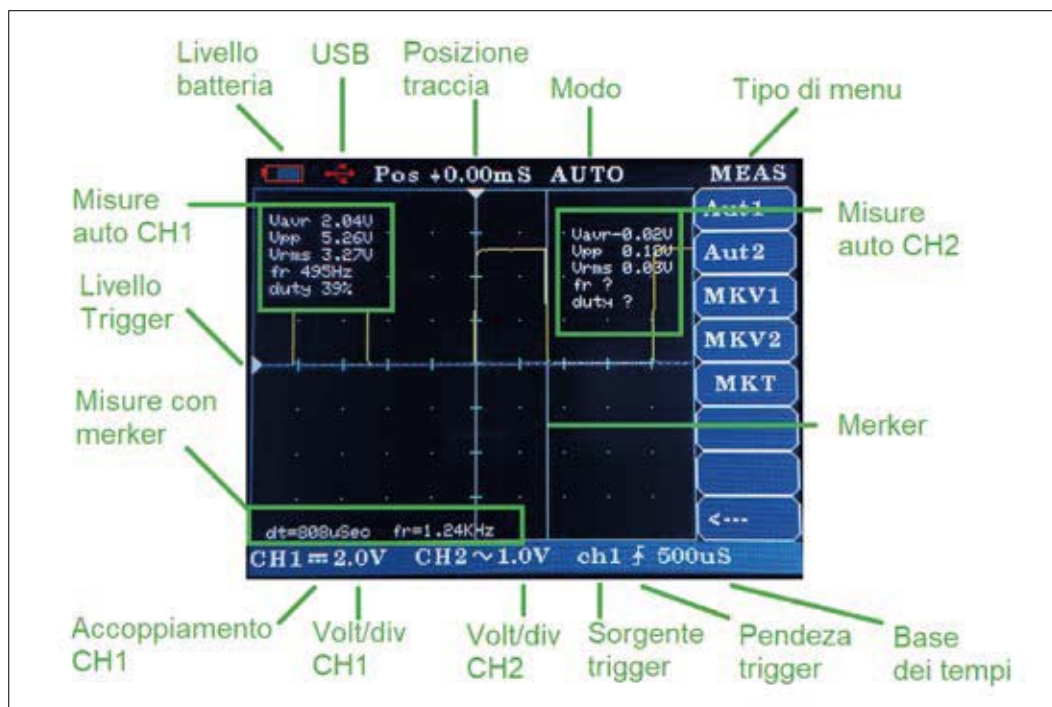


Fig. 15
Schermata
del display
con
descrizione
delle funzioni.

CONCLUSIONI

Pur con un hardware poco sofisticato e componenti a basso costo, siamo riusciti ad ottenere un oscilloscopio dalle interessanti caratteristiche, lo strumento è piccolo e compatto ottimo da tenere sempre a portata di mano per una misura veloce sui nostri circuiti, sia al banco, sia sul campo. Ci sono tutte le funzioni base di un oscilloscopio ma stiamo già lavorando per aggiungerne altre come la visualizzazione X/Y e la funzione matematica FFT praticamente introvabili in strumenti da laboratorio di questa categoria. □



Cosa occorre?

I componenti utilizzati in questo progetto sono di facile reperibilità. Il cuore del progetto, il modulo con STM32 (cod. STM32ARMCORE), è in vendita a Euro 7,50, il display touch da 2,8" (cod. LCDTOUCH28SER) è disponibile a Euro 19,50, la batteria litio da 1100 mA (cod. PLIB1100) costa Euro 7,90. Il circuito stampato può essere realizzato utilizzando il servizio di prototipazione dei PCB (cod. PCBPRODUCTION). I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

VASTA GAMMA DI STRUMENTI PER IL LABORATORIO

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

IL PIÙ VENDUTO



Disponibile anche
nella versione a 200MHz

COD. DS0202

€ 459,00

Oscilloscopio 2 Canali 100 MHz con generatore di forme d'onda

COD. DS0FG102

€ 420,00

- ✓ 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. DS0FG202

€ 549,00

Oscilloscopio a forma di penna - 1 canale

- ✓ Larghezza di banda 20 MHz
- ✓ Frequenza di campionamento 96 MSa/s
- ✓ Tensione max IN 50V
- ✓ Funzione FFT integrata
- ✓ Software per analisi in tempo reale con PC
- ✓ Alimentazione tramite porta USB
- ✓ Dimensioni 230 x 42 x 25 mm
- ✓ Peso 263 grammi



COD. PS02020

€ 95,00

Oscilloscopio USB per PC 2 Canali 20 MHz



COD. DS0PC22

€ 89,00

- ✓ Frequenza di campionamento 48 MS/s
- ✓ Funzione FFT integrata
- ✓ Interfaccia USB 2.0 per PC (alimentazione da USB)
- ✓ Software e memorizzazione delle forme d'onda

Disponibile anche
nella versione a 100MHz

COD. DS0PC102

€ 325,00

Generatore di funzioni e forme d'onda arbitrarie

USCITA FORME D'ONDA ARBITRARIE

- ✓ Frequenza onda DC~25 MHz
- ✓ 1 canale
- ✓ Frequenza di campionamento 200 MSa/s
- ✓ Risoluzione verticale 12 bit

€ 206,00

COD. FAW1025G

FREQUENZIMETRO

- ✓ Frequenza fino a 50 MHz
- ✓ Impedenza d'ingresso > 100 kohm
- ✓ Ampiezza segnale d'ingresso da 400 mVpp a 18 Vpp



Generatore di funzioni/forme d'onda arbitrarie fino a 10 MHz

€ 278,00

COD. TP0073



- ✓ 1 Canale
- ✓ Frequenza di campionamento 125 MS/s
- ✓ Frequenza max. di uscita 10 MHz
- ✓ Forme d'onda standard: sinusoidale, quadra, rampa, triangolare, impulso, noise, DC, arbitraria
- ✓ Supporta le modulazioni AM, FM, PM, PWM, ASK, FSK, PSK
- ✓ Ampiezza di uscita 1 mVpp ~ 10 Vpp (50 Ω); 2 mVpp ~ 20 Vpp (High Z)

Prezzi IVA inclusa.

FUTURA ELETTRONICA®

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

Caratteristiche tecniche
e vendita on-line su:

www.futurashop.it

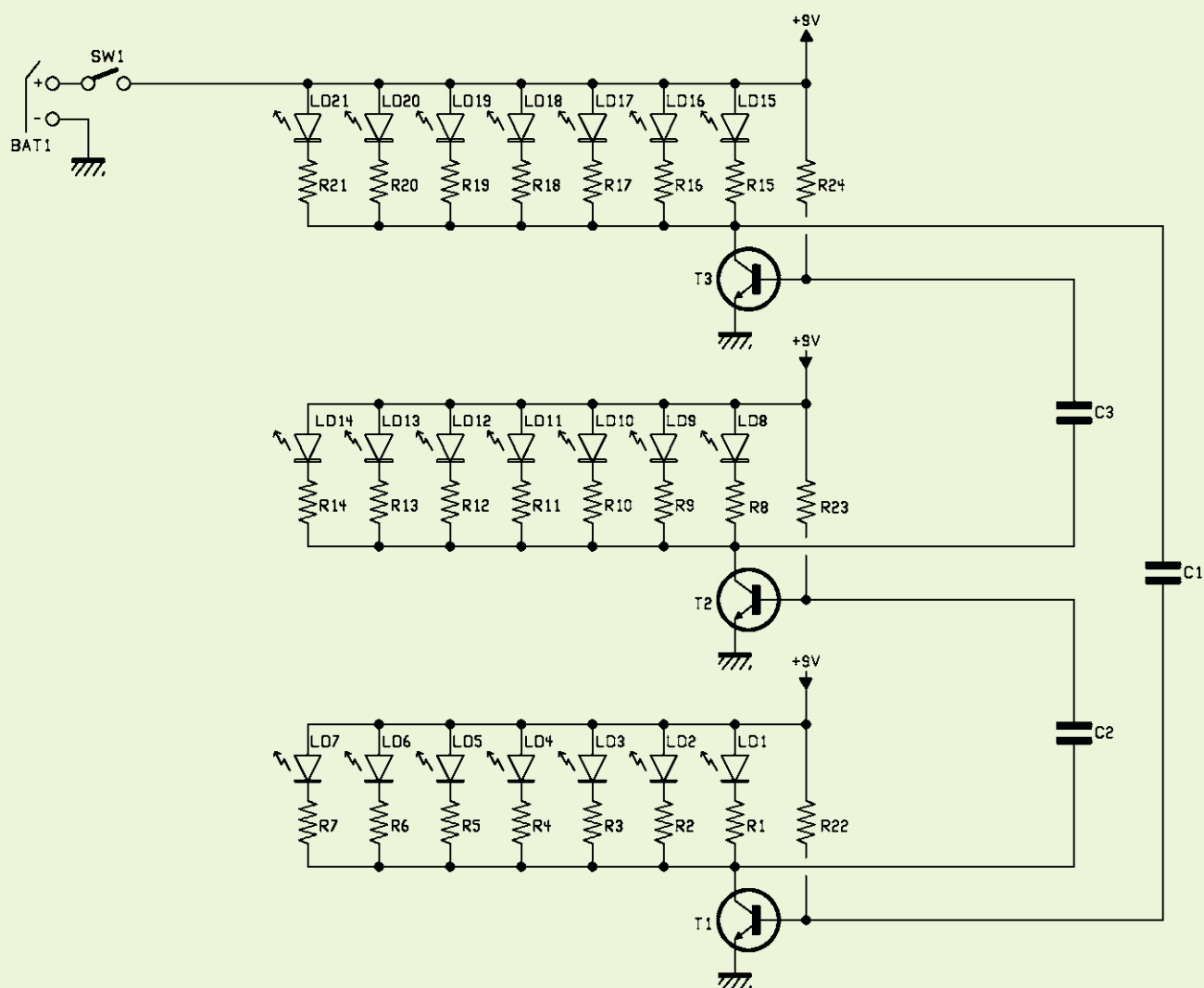
ALBERELLO SMD



di **DAVIDE SCULLINO**

Decorazione natalizia dal sapore amarcord di semplicissima costruzione e alla portata di tutti.

In questo Natale più che mai, c'è bisogno di spensieratezza, di allegria e di scaldarci con i simboli della festa e della tradizione, per allontanare la preoccupazione e i timori che il coronavirus -ospite indesiderato in queste feste- sta portando tornando a insinuarsi nella nostra vita e nella nostra libertà. Ecco quindi che vogliamo riproporvi, come ogni anno di questo periodo, una decorazione natalizia che vi allieterà guardando le sue luci lampeggiare e la cui costruzione e preparazione vi distoglierà dai cattivi pensieri, ancor più laddove siate costretti a rimanere a casa. Ciò che abbiamo preparato e vi proponiamo è un albero di Natale formato miniatura con tanto di circuito stampato sagomato (per farvi lavorare un po'...) a forma di pino e dotato di 28 lampadine composte da altrettanti LED, accesi a gruppi di sette per volta.



Per dare un tocco di modernità a quello che potremmo considerare un classico, abbiamo scelto di realizzare il tutto a montaggio superficiale, il che vi permetterà di ottenere un circuito praticamente piatto e quindi inseribile in buste, contenitori sottilissimi anche improvvisati, forme, pannelli ecc. Naturalmente potete metter quanti di questi gadget volete e in vari luoghi: non solo sull'albero di Natale ma anche sulla porta di casa, appeso a strenne e decorazioni passive o luminose, ma anche nelle vetrine di un negozio. Ma andiamo dunque a vedere di cosa si tratta.

SCHEMA ELETTRICO

Partiamo dall'analisi del circuito, per la quale faremo riferimento allo schema elettrico proposto in

queste pagine: si tratta, come vedete, di un circuito essenziale composto unicamente da componenti discreti, a dimostrazione di come ancora oggi possano essere svolte funzioni di sicuro effetto senza dover per forza scomodare logiche cablate o microcontrollori; infatti vediamo "in campo" tre transistor, altrettanti condensatori, 21 LED e 24 resistenze, più un interruttore per sezionare l'alimentazione (è prevista una pila da 9 volt). Il circuito propone il lampeggio alternativo di tre gruppi di LED, disposti sullo stampato dell'albero come le strisce di lampadine sarebbero in una configurazione a spirale attorno al vero albero di Natale; il gioco di luci risultante è ottenuto da quello che possiamo considerare un multivibratore astabile a transistor, però un po' inconsueto in

quanto formato da tre BJT (in questo caso NPN) e quindi da tre stadi invece dei 2 canonici. Analizziamone dunque il funzionamento immaginando di applicare l'alimentazione (quindi chiudere l'interruttore SW1) quando i condensatori C1, C2 e C3 sono completamente scarichi: in questa condizione, teoricamente tutti e tre i transistor hanno le basi alimentate dalle rispettive resistenze di polarizzazione, che sono R22 per il T1, R23 per il T2 ed R24 per il T3.

Questo però non basta a farli condurre, perché se è vero che ciascuno è polarizzato, è vero altrettanto che verrà portato in conduzione uno solo alla volta, dato che le basi degli altri due, ossia quelli che non avranno fatto in tempo a entrare in conduzione, saranno condizionate dalla connessione realizzata tramite i condensatori, che le cortocircuiteranno praticamente con i rispettivi emettitori (che sono a massa).

Ciò accade perché i transistor, pur essendo dello stesso tipo, in realtà presentano differenze costruttive (non a caso i costruttori di semiconduttori nei datasheet specificano un range di parametri come ad esempio l' h_{fe} , che va dal minimo al tipico, fino al massimo) tali che a parità di tensione di polarizzazione (tutte le resistenze di base sono alimentate dalla linea positiva comune +9V) uno dei tre inizierà a condurre per primo, ad esempio perché ha una V_{be} (tensione base-emettitore) di soglia inferiore a quella degli altri due e/o un guadagno in corrente maggiore. In questo contesto concorrono anche le tolleranze dei resistori componenti il circuito.

Bene, immaginiamo che ad entrare in conduzione per primo sia il T1: inizia a fluire corrente nel suo collettore, il quale assume un potenziale decrescente e tale da "tirare" verso massa la base di T2 attraverso il condensatore C2, il quale, ipotizzato che sia inizialmente scarico, si presenta come un cortocircuito. Per effetto di ciò, l'NPN rimarrà inizialmente interdetto finché C2 non si sarà caricato attraverso il predetto condensatore.

La tensione tra collettore ed emettitore di T2 sarà abbastanza alta da forzare la carica del C3, il quale però inizialmente scarico, sarà praticamente un cortocircuito e porterà il livello alto sulla base del T3, il quale fino a quel momento rimarrà in conduzione; infatti mentre la base di T2 è bloccata a livello basso dal collettore di T1, quella del T3 non ne segue le vicende e trovandosi polarizzata attraverso la resistenza R4, con C3 che dall'armatura opposta è a livello alto, avremo il predetto T3 in saturazione.



Tale condizione farà scendere praticamente a zero volt il potenziale dell'armatura di C1 connessa al collettore e ora tale condensatore verrà scaricato attraverso la R22, tanto da interdire in breve tempo il T1, giacché causerebbe un abbassamento della V_{be} di quest'ultimo al disotto di quella di soglia. Quindi determina ora la scarica del C2, che si stava caricando attraverso la resistenza R23, il quale poi verrà ricaricato con polarità opposta (avendo ora il positivo sul collettore del T1 e non più sulla base del T2) fin quando la V_{be} del T2 non sarà tale da portare quest'ultimo in conduzione, allorché il suo collettore scenderà a circa zero volt e trascinerà con sé l'armatura del C3 dal suo lato.

Adesso tale condensatore verrà scaricato, determinando un impulso di tensione a livello basso sulla base di T3, che passerà all'interdizione: il suo collettore si porterà a livello alto e tale condizione scaricherà C1 per poi avviare la carica con polarità opposta, ma nel frattempo la corrente in C1 sarà tale da portare in pochi istanti la base del T1 a livello alto, facendolo nuovamente condurre e ricominciando quindi il ciclo appena descritto. Abbiamo quindi questa sequenza di commutazione: inizialmente T1 e T3 in saturazione con T2 interdetto, poi T1 interdetto, T2 in saturazione, T3 interdetto, T1 nuovamente in saturazione, T2 interdetto, T3 in saturazione e via di seguito. Quindi il circuito parte praticamente con T1 in conduzione e immediatamente dopo T3, mentre T2 è interdetto, poi andrà in conduzione un



Il circuito stampato è sagomato come un albero per essere più realistico.

CARATTERISTICHE TECNICHE



Tensione di alimentazione:
9 Vcc



Corrente assorbita:
50 mA



Frequenza di lavoro:
3 Hz



LED utilizzati:
21



Sequenza luminosa:
LED accesi a 7 per volta



Elenco Componenti:

R1, R2, R3, R4, R5, R6, R7: 1 kohm
 R8, R9, R10, R11, R12, R13,
 R14, R15, R16, R17, R18, R19, R20, R21: 510 ohm
 R22, R23, R24: 10 kohm
 C1, C2, C3: 10 μ F ceramico
 LD1, LD2, LD3, LD4, LD5, LD6, LD7: LED rosso
 LD8, LD9, LD10, LD11, LD12, LD13, LD14: LED blu
 LD15, LD16, LD17, LD18, LD19, LD21: LED verde

LD20: LED giallo
 Q1, Q2, Q3: BC817
 SW1: Deviatore a slitta
 BAT1: Clip batteria 9V

Varie:
 - Circuito stampato S1538
 (96x75 mm)

solo transistor per volta; questa modalità di funzionamento fa sì che avremo accesi i sette LED collegati (ciascuno mediante una resistenza di limitazione) al collettore di ciascun transistor solo quando quest'ultimo andrà in saturazione, vale a dire LD1+LD7 (LED rossi) alla conduzione del T1, LD8+LD14 (blu) con T2 in saturazione ed LD15+LD21 (verdi) quando a condurre sarà T3. Il LED LD20 è posizionato proprio al centro della stella, per questo motivo abbiamo preferito utilizzare un colore giallo rispetto al verde. Naturalmente la sequenza potrebbe essere diversa, perché se a partire in conduzione fosse T2, avremmo inizialmente T3 interdetto e T2 in conduzione e via di seguito.

A questo punto è opportuno precisare che il circuito, più che un astabile a tre stadi è da vedersi come la connessione ad anello (in loop) di tre temporizzatori elementari (monostabili) retriggerabili: quello basato su T1 pilota quello realizzato intorno a T2, il quale è in cascata con il blocco facente capo a T3, il cui collettore pilota attraverso il condensatore C1 il transistor T1. Infatti le vicende di ognuno vanno a ripercuotersi su quello che segue e si può innescare una commutazione ciclica per il fatto che l'ultimo della catena interviene sullo stato del primo. Chiudiamo qui l'analisi del circuito con l'alimentazione, la quale deve essere a tensione continua applicata tra i contatti + e - BAT1; trattandosi di un circuito da appendere con una certa libertà,

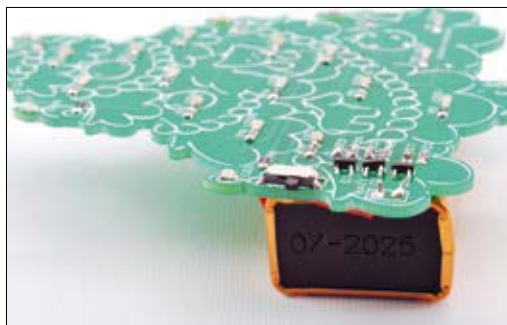
abbiamo ipotizzato di alimentarlo con una pila a secco di quelle da 9V (la classica nata per le radio a transistor) ma nulla vieta, ad esempio se l'alberello viene applicato all'albero di Natale come decorazione luminosa insieme a strisce di lampadine o di LED, di predisporre l'alimentazione da rete tramite un piccolo alimentatore con uscita a tensione continua di 9V, dato che l'alimentazione elettrica l'avete già per il resto delle luminarie.

La corrente assorbita dal circuito è molto ridotta, perché se è vero che ci sono ben 21 LED, è vero altrettanto che a regime (ossia dopo il brevissimo transitorio iniziale) se ne accende una fila per volta, determinando un prelievo dai contatti di alimentazione dell'ordine dei 60 milliamper di media. Quindi se usate un alimentatore da rete vi basta che eroghi 9V e 100 mA.

REALIZZAZIONE PRATICA

Bene, ciò detto passiamo alla costruzione dell'alberello, che al solito inizierà dal circuito stampato: si tratta di una basetta a doppia ramatura che dovrete ricavare per fotoincisione a partire dalle tracce lato rame, scaricabili dalla sezione Download del nostro sito www.elettronica.in.it insieme agli altri file di questo numero della rivista, che stamperete su carta da lucido o acetato per ricavare le due pellicole; fatto questo prendete una basetta doppia faccia presensibilizzata (che è la maniera più semplice per evitare di dover armeggiare con photoresist, pennelli, lampada di essiccazione al quarzo e roba simile...) tagliatela delle dimensioni esterne dello stampato che sono 96x75 mm e poi impressionate la prima faccia applicando una delle pellicole. Dopo fate alcuni fori con la punta da trapano da 0,8 mm in corrispondenza di due piazzole di interconnessione (stando attenti a non graffiare ed asportare il photoresist sul lato già impressionato) e sull'altra faccia centrate la seconda traccia, impressionando il lato mancante.

A questo punto passate alla soluzione di sviluppo, lavate il circuito stampato e incidetelo, dopodiché dovrete forarlo e iniziare a stagnare le piazzole di interconnessione da entrambe le facce dopo aver inserito in ciascuna un sottile spezzone di filo ricavabile, ad esempio, da una cordina multifilare da impianati elettrici (un pezzetto da 1 – 1,5 mmq va benissimo); passate quindi ai componenti, ricordando che sono tutti SMD in case 0805 e che per la loro saldatura vi serviranno del filo di lega saldante sottile (massimo 0,5 mm di diametro) e un saldatore con punta fine, oltre a una pinzetta



← Tutti i componenti si saldano su una faccia del PCB..

per il posizionamento e una lente d'ingrandimento per centrare bene gli elementi sulle piazzole e verificare come sono venute le saldature. Per facilitarvi il lavoro consigliamo di spalmare con un pennellino della pasta flussante sulle piazzole prima di applicare i componenti, il che agevolerà posizionamento e scioglimento dello stagno. Ricordate di seguire, in ogni fase, il piano di montaggio visibile in queste pagine, che vi guiderà nel corretto posizionamento dei 21 LED, che potrebbero anche essere "girati" e quindi una volta alimentato il circuito non si accenderebbero perché verrebbero polarizzati inversamente; per facilitarvi il compito, sappiate che normalmente sul corpo dei LED SMD, lateralmente e dalla parte del catodo, c'è un segno colorato o una tacca o una parte in rilievo).

Il montaggio è meno critico per i transistor, giacché si posizionano solo in un modo, ovviamente a



← La pila applicata con del biadesivo dietro al PCB.

Guarda l'alberello SMD in azione su YouTube!



Vuoi vedere in azione il nostro alberello SMD? Nulla di più semplice: collegati alla pagina <https://www.youtube.com/watch?v=OuMPawHJ5Yw> Potrai anche vedere i video di moltissimi altri nostri progetti proposti in passato.

patto di non metterli sottosopra, cosa però difficile perché i piedini sono sagomati in modo da non lasciare dubbi su quale è il lato su cui appoggiare il corpo dei componenti.

Completate le saldature, con la lente d'ingrandimento verificate che siano tutte ben fatte e che non siano rimasti "baffi" di stagno a cortocircuitare piazzole o terminali vicini, quindi date una pulita alla basetta con del "rimuovi fluxante" (si trova in commercio sotto forma di spray o flaconcino) o semplicemente con del diluente alla nitro, quindi il vostro alberello è pronto per essere installato.

Se pensate di alimentarlo a pila, applicate sul retro mediante del nastro biadesivo spesso un portapila idoneo allo scopo, dotato di doppiino rosso-nero e connettete i due fili ai contatti + (il rosso) e - (il nero) BAT1.

Volendo potete, come abbiamo fatto nel nostro prototipo, evitare il portapila e utilizzare per la connessione una presa volante per pile da 9V, collegandone i due fili rosso e nero come indicato, quindi applicare direttamente la pila con del biadesivo spesso al retro del PCB.

Se invece pensate all'alimentazione da rete, connettete i fili di uscita del vostro alimentatore da 9Vcc a + e - BAT1 seguendo sempre la polarità

indicata; collegatevi con una piattina, magari alla fonte di alimentazione di una centralina di controllo luci che già accende le luminarie dell'albero che andrete a decorare o strisce luminose di casa. Vi ricordiamo che la frequenza di lampeggio delle file di LED, pur essendo fissa non è obbligatorio tenerla com'è (circa 3 hertz): potete modificarla a piacimento, per esempio rallentando (se vi sembra troppo rapida) la pulsazione della luce mediante la sostituzione dei tre condensatori con elementi di valore maggiore; in teoria, raddoppiando le capacità dimezzate la frequenza del lampeggio e via di seguito.

È anche possibile aumentare la frequenza, però a nostro avviso è qualcosa che ha poco senso perché potrebbe arrivare a determinare un lampeggio e l'alternarsi dell'accensione dei tre colori, tanto rapido da diventare praticamente impercettibile, ma potete provarci ugualmente, fermo restando che per ottenere la cosa dovete diminuire i valori dei tre condensatori.

Se utilizzate valori differenti per C1, C2 e C3, differenziate i temi di accensione dei LED dei rispettivi colori: anche questa è una possibilità di personalizzazione del circuito.

Bene, con questo vi abbiamo detto proprio tutto e non ci resta che augurarvi buon lavoro, buon divertimento nell'addobbare i vostri locali e soprattutto... Buon Natale! 



Cosa occorre?

I componenti utilizzati in questo progetto sono facilmente reperibili. L'alberello di Natale SMD (cod. FT1538) è disponibile anche montato a Euro 9,00.

Per chi volesse stampare il PCB del proprio gadget natalizio, è disponibile il servizio PCBPRODUCTION sul sito www.futurashop.it

I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

Stop ai fili con i nuovi interruttori wireless a 433 MHz!

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

cavi di collegamento. La procedura di abbinamento tra ricevitore e pulsante è molto semplice e offre la possibilità di memorizzare fino a 5 dispositivi.



cod. FR790
€ 16,00

**Ricevitore wireless 433 MHz
1 Canale - 220 VAC**

**Interruttori wireless
a 433 MHz
autoalimentati**



cod. FR791
€ 9,00



cod. FR792
€ 16,00



RICEVITORE

2 CANALI 220VAC
cod. EMY220RX2 **€ 16,00**

2 CANALI 12VDC
cod. EMY12RX2 **€ 16,00**

4 CANALI 12VDC
cod. EMY12RX4 **€ 19,00**

**Istruzioni
in italiano!**

RICEVITORE

1 CANALE 220VAC
cod. EMY220RX1 **€ 13,00**

1 CANALE 12VDC
cod. EMY12RX1 **€ 13,00**

TRASMETTITORE

1 CANALE
cod. EMYTX1
€ 11,00

**TRASMETTITORE
2 CANALI**
cod. EMYTX2 **€ 12,00**

**TRASMETTITORE
4 CANALI**
cod. EMYTX4 **€ 13,00**

SISTEMA DI RADIOCONTROLLO A 433 MHz a 1-2-4 canali 220VAC o 12VDC

Trasmettitori e ricevitori a 1-2-4 canali operanti sulla banda dei 433 MHz. Forniscono un segnale stabile a lunga distanza e un bassissimo consumo di corrente in modalità stand-by. I ricevitori possono essere alimentati a 220VAC o a 12VDC in base al modello. Ai ricevitori va abbinato esclusivamente il trasmettitore con lo stesso numero di canali.



PER I TUOI PROGETTI UTILIZZA I SET DI COMPONENTI

Vasta scelta e grande risparmio!

Euro 14,90

cod. YM397



**SET 1000 PEZZI DI LED
3 MM - VARI COLORI**

Euro 4,90

cod. SETCONDELE



**SET CONDENSATORI ELETTROLITICI
12 VALORI**

Euro 8,00

cod. YM403



**SET 100 CONDENSATORI
POLIESTERE - VARIE CAPACITÀ**

Euro 7,50

cod. KITFUS5X30



**KIT FUSIBILI 82 PEZZI 6X30
ASSORTITI**

Euro 7,50

cod. YM406



**SET 300 PEZZI LED 3-5 MM
VARI COLORI**

Euro 9,00

cod. YM374



**SET 50 TRIMMER DI PRECISIONE
MULTIGIRI - MONTAGGIO VERTICALE**

Euro 9,00

YM398



**SET 500 PEZZI LED 5 MM
VARI COLORI**

Euro 6,00

cod. YM379



**SET COMPONENTI ELETTRONICI
ASSORTITI**

Euro 12,00

cod. YM377



**SET 300 TRANSISTOR ASSORTITI
TO-92**

Euro 21,90

cod. SETELEC



**SET COMPONENTI ELETTRONICI
VARI CON PIASTRA E ALIMENTATORE**

Euro 8,00

cod. SET24INDU



**SET INDUTTANZE 24 VALORI
10 PEZZI PER VALORE**

Euro 10,90

cod. SMDRES3K610M



**SET RESISTENZE SMD 0805
DA 3,6 KOHM A 10 MOHM**

Euro 6,00

cod. 140ZENER



**SET 140 DIODI ZENER
10 PEZZI PER VALORE**

Euro 15,00

cod. SMDRES03K6



**SET RESISTENZE SMD 0805
DA 0 A 3,6 KOHM**

Euro 3,90

cod. SETCERAMICI



**SET CONDENSATORI CERAMICI
22 VALORI - 10 PEZZI PER VALORE**

Euro 6,50

cod. SETDIODI



**SET DIODI RADDRIZZATORI
100 PEZZI**

Euro 5,50

cod. SETTRIMMERMO



**SET TRIMMER MONTAGGIO
ORIZZONTALE - ASSORTITI**

Euro 7,00

cod. SETMULTIST



**SET CONDENSATORI MULTISTRATO
VARI VALORI - 10 PEZZI PER VALORE**

Euro 16,00

cod. KRES-0603



**SET 1700 RESISTENZE SMD
(SERIE 0603) - ASSORTITE**

Euro 6,50

cod. HS2133



**SET 66 PZ ZOCCOLI A LAMELLA
ASSORTITI**

VISIONE

HAI LA MASCHERINA?

RASPBERRY Pi LO VEDE

Mask 100%



di DANIEL SCAINI

Individuare i volti all'interno di un video o di una ripresa live e riconoscere se indossano la mascherina: facile con il "lampone".

La diffusione del COVID-19 ha creato una crisi sanitaria globale che ha avuto un profondo impatto sull'umanità, sul modo in cui percepiamo i nostri spazi e la nostra vita quotidiana. A partire dal dicembre 2019 abbiamo assistito alla diffusione di questa "sindrome respiratoria acuta" comunemente denominata coronavirus (SARS-CoV-2) che ancora non ha una cura e che ci costringe a difenderci dal contagio adottando quelle precauzioni che ormai sono entrate nella vita di tutti i giorni: indossare una mascherina chirurgica o meglio ancora FFP2, disinfettarsi le mani dopo aver toccato oggetti e luoghi toccati da altri, mantenere una certa distanza tra le persone ed evitare di uscire di casa se si ha febbre. L'uso della mascherina è ormai condizione sine qua non per accedere in luoghi frequentati dal pubblico ed il rispetto della prescrizione è onere del gestore, che però può farlo quando

il flusso di persone lo permette; nei luoghi in cui l'afflusso è maggiore e incontrollato, come un supermercato, una stazione, un aeroporto, l'ostacolo può essere superato proprio attraverso la visione artificiale.

In questo articolo spiegheremo come farlo utilizzando Raspberry Pi, che ha tutte le risorse del caso e che può essere integrata in un sistema compatto stand-alone da collocare sui varchi dei luoghi da sorvegliare. Per farlo, in primis dobbiamo ottenere un sistema operativo nel quale siano presenti tutte le librerie e i framework stabili; poi è indispensabile riuscire ad avere un qualche algoritmo per riconoscere una persona all'interno di una serie/sequenza di immagini e più nel dettaglio il loro volto; ed infine un modo per poter discernere fra i vari volti chi ha il dispositivo di protezione personale e chi no.

In questo progetto coinvolgeremo la visione artificiale, anche chiamata Computer Vision (CV), cioè l'insieme dei processi che consentono di identificare il contenuto delle immagini, ma anche l'intelligenza artificiale, che mira a ricreare nei computer l'apprendimento umano.

IL LINGUAGGIO

Il linguaggio di programmazione utilizzato è Python: orientato agli oggetti, è utilizzabile per molti tipi di sviluppo software proprio per la sua facilità ed integrazione con altri tipi di linguaggio. Chiaro e potente, è paragonabile a Java o Perl e come questi utilizza un'elegante sintassi permettendo di leggere facilmente gli script e quindi di creare progetti complessi senza rinunciare alla manutenibilità.

Oggi, secondo diversi sondaggi, è utilizzato addirittura dall'84% degli sviluppatori come linguaggio di programmazione primario.

Per quanto riguarda invece gli ambiti, tra i più usati troviamo in testa Data Analysis e Web Development, ma non solo: stanno infatti cominciando ad emergere anche altri ambiti come il Machine learning, la Computer graphics e il Game development.

La versione più utilizzata è la 3.6 e spesso viene utilizzata con IDE/Editor come PyCharm Professional/Community Edition o VS Code.

Dato lo scarso numero di righe che andremo a scrivere è consigliato installare un editor solo se si è alle prime armi, altrimenti i pre installati Nano o Text Editor sono più che sufficienti.

Sebbene non sia ottimizzato come il C++, soprattutto per quanto riguarda l'uso di RAM e

processore, è stato scelto poiché open source, con un numero veramente alto di librerie anch'esse disponibili senza costi aggiuntivi e portabile su ogni tipo di piattaforma OS (Linux, Mac, Windows). Nel nostro caso infatti girerà su Raspbian, OS basato sui rilasci ufficiali di Debian adattati per una architettura ARM, ma potrebbe tranquillamente essere eseguito su Windows senza dover apporre troppe modifiche.

IL NOSTRO HARDWARE

L'hardware consigliato per questo progetto è Raspberry Pi 4 da 8GB, in modo da avere un minimo di RAM disponibile ed anche un sistema aggiornato in grado di poter supportare anche una maggiore elaborazione audio e video. Basata su un processore ARM, offre prestazioni paragonabili ad un pc entry-level ma può gestire anche flussi video 4k. Inoltre offre numerosi pin, sia di alimentazione a 3,3V, sia per poter interfacciare periferiche, relay o sensori di diversa natura e gestibili tranquillamente a livello di codice Python o C++ attraverso un insieme di librerie messe a disposizione dal produttore.

A corredo di questa andremo ad utilizzare una PI camera rev 1.3 da 5MP, che verrà collegata direttamente sulla Raspberry Pi attraverso il connettore CSI. Questo modulo può offrire un'immagine chiara e nitida con una risoluzione di 5MP o registrare video a 1.080p con 30 fps.

Il sensore installato sul modulo ha un connettore Ribbon a 15 pin e attraverso la connessione in seriale con il connettore CSI offre un'ottima velocità di trasferimento dati, rendendo questo strumento particolarmente adatto a progetti nell'ambito embedded/mobile (**Fig. 1**).

CONFIGURAZIONE DEL SISTEMA

Andiamo ora ad installare il sistema operativo all'interno della Raspberry PI, cosa che richiede una scheda microSD card con almeno 16/32GB, un software Raspberry Pi Imager ed il file zip con l'immagine vera e propria del sistema operativo. Per gli ultimi due il sito di riferimento è quello ufficiale: <https://www.raspberrypi.org/downloads/>: nella prima sezione vedremo Imager, ovvero il tool con cui creare una SD-card bootabile in grado di far partire il dispositivo. Più avanti, sempre all'interno del sito, troviamo lo zip invece che andremo a caricare, riporto il link diretto https://downloads.raspberrypi.org/raspbian_armhf_latest.

Una volta installato l'imager apriamolo, inseriamo la nostra SD-card, e clicchiamo CHOOSE OS.





Fig. 1
L'hardware
del progetto,
composto da
Raspberry
Pi e dalla
telecamera .

Qui selezioniamo l'ultima voce ovvero Custom OS e scegliamo quindi lo zip scaricato in precedenza. Successivamente clicchiamo su CHOOSE SD CARD e selezioniamo l'unità sulla quale desideriamo riportare il sistema operativo e infine WRITE. Dopo alcuni istanti avremo la nostra SD card e saremo pronti per partire con il nostro dispositivo. È possibile utilizzare l'immagine preconfigurata *rpi_32bit_facemaskdetector.zip* all'interno della quale sono già presenti tutte le librerie e i moduli per poter utilizzare correttamente i programmi. In tal caso le credenziali per accedervi sono: user **pi** e password **12345678**.

Inseriamo la card nel lettore della Raspberry Pi, colleghiamo quest'ultima a una rete con accesso ad internet e accendiamola. Una volta seguite le prime indicazioni sulle impostazioni generali come password, rete ecc, andiamo dentro alle impostazioni, cliccando sull'icona del lampone in alto a sinistra e scegliamo Preferenze>Configurazione Raspberry Pi (**Fig. 2**): apparirà una schermata come quella mostrata in **Fig. 3** e qui clicchiamo su Interfacce e abilitiamo le prime due voci, ovvero Camera e SSH. La prima per ovvi motivi, nel caso in cui volessimo appunto usare un modulo camera PI esterno, mentre la seconda per poterci connettere in remoto alla macchina attraverso tool come Putty o WinSCP.

Chiudiamo il menu cliccando su OK e ci verrà chiesto di riavviare il sistema: procediamo. Al riavvio apriamo un terminale e sempre per poter raggiungere la macchina da remoto installiamo il desktop remoto digitando:

```
sudo apt-get install xrdp -y
```

```
sudo apt-get update
sudo apt-get upgrade
```

Ora abbiamo tutti gli strumenti per raggiungere la macchina da remoto e quindi possiamo dimenticarci del monitor e agire direttamente dal nostro pc, a voi la scelta. Installiamo ora i due framework indispensabili per ogni progetto di visione artificiale e machine learning: OpenCV e Tensorflow. OpenCV è una libreria ed un insieme di algoritmi scritti in C per lo streaming e la manipolazione image/video real-time per la visione artificiale disponibile ormai per tutte le piattaforme OS. Questa permette la trattazione di immagini in maniera semplice, gestendole come matrici di pixel. Ci permette di effettuare quel processo alla base della computer vision, ovvero l'Image processing, andando a modificare/migliorare le immagini o sequenze di immagini (video). È molto utilizzata in ambito scientifico e di ricerca e può essere adoperata in diversi settori: controllo della qualità, ricerca, difesa e molti altri. Con poche righe di codice è possibile riconoscere le persone all'interno di una foto, leggere in maniera automatica delle targhe ecc. Tutte azioni che altrimenti sarebbe molto complicato implementare e integrare in progetti più ampi. Nel caso di dispositivi embedded come questo l'installazione non può essere lanciata con un solo comando, ma per avere un controllo puntuale della versione dovremmo andare ad installarla partendo direttamente dai sorgenti.

Fig. 2
Il sottomenu
Preferenze.



Per far ciò apriamo nuovamente il terminale e dal prompt **pi@raspberrypi: ~ \$** digitiamo i comandi mostrati in sequenza nel **Listato 1**. Questi ci permetteranno di installare tutti quei componenti di sistema che ci renderanno autonomi nel compilare i sorgenti e compiere l'installazione. Scarichiamo i sorgenti e decomprimiamoli mediante i seguenti comandi:

```
cd ~
wget -O opencv.zip https://github.com/opencv/opencv/archive/4.2.0.zip
wget -O opencv_contrib.zip https://github.com/opencv/opencv_contrib/archive/4.2.0.zip
unzip opencv.zip
unzip opencv_contrib.zip
mv opencv-4.2.0 opencv
mv opencv_contrib-4.2.0 opencv_contrib
```

Fig. 3
Schermata
impostazioni
Raspberry Pi..



Installiamo ora l'ambiente virtuale dove compilare senza fare danni o alterare le dipendenze dell'intero sistema operativo, mediante i comandi:

```
sudo pip3 install virtualenv
sudo pip3 install virtualenvwrapper
echo "export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3.7" >> ~/.bashrc
echo "export WORKON_HOME=$HOME/.virtualenvs" >> ~/.bashrc
echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~/.bashrc
source ~/.bashrc
```

Infine entriamo in questo ambiente di test e compiliamo i sorgenti (come vediamo in **Fig. 4**) e verifichiamo che la compilazione prosegua correttamente (**Fig. 5**).

I tempi di compilazione possono variare in base alla macchina su cui stiamo lavorando: nel nostro caso abbiamo utilizzato un Raspberry Pi 4 con 8 GB di RAM e ci abbiamo messo circa una mezz'ora. Ora installiamo il file compilato con i seguenti comandi:

```
sudo make install
sudo ldconfig
sudo apt-get update
cd ~
rm opencv.zip
rm opencv_contrib.zip
```

Chiudiamo il terminale su cui stiamo lavorando e apriamo una nuova finestra su cui digiteremo quest'ultime righe:

```
cd ~/.virtualenvs/cv420/lib/python3.7/site-packages
ln -s /usr/local/lib/python3.7/site-packages/cv2/python-3.7/cv2.cpython-37m-arm-linux-gnueabi.so
cd ~
cd ~/opencv/build/lib/python3
sudo cp cv2.cpython-37m-arm-linux-gnueabi.so /usr/local/lib/python3.7/dist-packages/cv2.so
cd ~
```

Una volta terminata l'installazione della prima tra le due macro librerie passiamo alla seconda TensorFlow.

Questa, riferimento per il machine learning, consente a principianti ed esperti di creare facilmente modelli di ai per desktop, dispositivi mobili, web e cloud. Verrà usata per creare il modello, ovvero una struttura basata su campioni appositamente

▼ Listato 1

```
sudo apt-get install build-essential cmake git unzip pkg-config -y
sudo apt-get install libjpeg-dev libpng-dev libtiff-dev -y
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev -y
sudo apt-get install libgtk2.0-dev libcanberra-gtk* -y
sudo apt-get install libxvidcore-dev libx264-dev libgtk-3-dev -y
sudo apt-get install python3-dev python3-numpy python3-pip -y
sudo apt-get install python-dev python-numpy -y
sudo apt-get install libtbb2 libtbb-dev libdc1394-22-dev -y
sudo apt-get install libv4l-dev v4l-utils -y
sudo apt-get install libjasper-dev libopenblas-dev libatlas-base-dev libblas-dev -y
sudo apt-get install liblapack-dev gfortran -y
sudo apt-get install gcc-arm* -y
sudo apt-get install protobuf-compiler -y
```

addestrati per discriminare le varie condizioni, e successivamente per utilizzarlo all'interno dell'algoritmo, così da poter riconoscere i volti e, soprattutto, se questi indossano o meno il dispositivo di protezione.

Sempre sul terminale digitiamo questi comandi per installare alcuni prerequisiti:

```
sudo apt-get install libhdf5-dev libc-ares-dev
libeigen3-dev -y
sudo apt-get install libatomic1 libatlas-base-dev -y
sudo pip3 install keras_applications==1.0.8 --no-
deps
sudo pip3 install keras_preprocessing==1.0.5 --no-
```

deps

```
sudo pip3 install h5py
```

```
sudo pip3 install keras_applications==1.0.8 --no-
```

deps

```
sudo pip3 install keras_preprocessing==1.0.5 --no-
```

deps

```
sudo pip3 install "h5py<3.0"
```

```
sudo pip3 install -U --user six wheel mock
```

Ora scarichiamo ed installiamo la versione 1.15.2 della libreria dedicata alla nostra architettura ARM, mediante il comando:

```
wget https://github.com/Qengineering/Tensorflow-Ra-
```

```
pi@raspberrypi: ~/op/opencv/build
File Modifica Schede Aiuto
-- Python 3:
--   Interpreter:           /home/pi/.virtualenvs/tst/bin/python3 (ver 3
.7.3)
--   Libraries:            /usr/lib/arm-linux-gnueabi/hf/libpython3.7m.s
o (ver 3.7.3)
--   numpy:                /home/pi/.virtualenvs/tst/lib/python3.7/site
-packages/numpy/core/include (ver 1.19.4)
--   install path:         lib/python3.7/site-packages/cv2/python-3.7
--
-- Python (for build):      /usr/bin/python2.7
--
-- Java:
--   ant:                  NO
--   JNI:                  NO
--   Java wrappers:        NO
--   Java tests:           NO
--
-- Install to:              /usr/local
-----
-- Configuring done
-- Generating done
-- Build files have been written to: /home/pi/op/opencv/build
(tst) pi@raspberrypi:~/op/opencv/build $
```

◀ Fig. 4
Compilazione
dei sorgenti
nell'editor.

Listato 2

```

mkvirtualenv cv420
pip3 install numpy
cd opencv
mkdir build
cd build
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules -D ENABLE_NEON=ON -D ENABLE_VFPV3=ON -D WITH_OPENMP=ON -D BUILD_TIFF=ON -D WITH_FFMPEG=ON -D WITH_GSTREAMER=ON -D WITH_TBB=ON -D BUILD_TBB=ON -D BUILD_TESTS=OFF -D WITH_EIGEN=OFF -D WITH_V4L=ON -D WITH_LIBV4L=ON -D WITH_QT=OFF -D WITH_VTK=OFF -D OPENCV_EXTRA_EXE_LINKER_FLAGS=-latomic -D OPENCV_ENABLE_NONFREE=ON -D INSTALL_C_EXAMPLES=OFF -D INSTALL_PYTHON_EXAMPLES=OFF -D BUILD_NEW_PYTHON_SUPPORT=ON -D BUILD_opencv_python3=TRUE -D OPENCV_GENERATE_PKGCONFIG=ON -D BUILD_EXAMPLES=OFF ..
make -j4
    
```

```

spberry-Pi/raw/master/tensorflow-1.15.2-cp37-cp37m-
linux_armv7l.whl
sudo -H pip3 install tensorflow-1.15.2-cp37-cp37m-
linux_armv7l.whl
sudo reboot
    
```

Dopo il riavvio, installiamo le ultime librerie per poter eseguire correttamente tutti gli script Python preparati:

```

pip3 install keras==2.3.1
pip3 install imutils==0.5.3
pip3 install numpy==1.18.2
pip3 install matplotlib==3.2.1
    
```

```

pip3 install argparse==1.1
pip3 install scipy==1.4.1
pip3 install scikit-learn==0.23.1
pip3 install pillow==7.2.0
pip3 install "picamera[array]"
sudo apt-get install alsa-utils mplayer -y
sudo apt-get install festival -y
    
```

e scarichiamo i sorgenti Python mediante i seguenti comandi:

```

git clone https://github.com/danisk89/Riconoscimen-
toMascherinaPythonOpenCV.git
cd RiconoscimentoMascherinaPythonOpenCV
    
```

 **Fig. 5**
 Progressione della
 compilazione dei
 sorgenti.

74

IL MODELLO

Come primo passo andremo ad implementare una serie di script in Python in modo da poter addestrare un nostro dataset usando Keras e Tensor-Flow. Queste ultime sono le librerie open source di riferimento per l'apprendimento automatico di reti neurali. Mentre della seconda abbiamo già ampiamente dibattuto della prima possiamo affermare che, scritta in Python anch'essa, permette una rapida prototipazione di reti neurali profonde e si concentra sulla facilità d'uso, la modularità e l'estendibilità. Nella fase 0 andremo a realizzare un modello in grado di determinare se il soggetto o i soggetti inquadrati hanno o meno la mascherina come dispositivo di protezione personale. Per semplicità e velocità il suddetto modello è già presente nella cartella e pre addestrato, ma per completezza verrà analizzato il flusso partendo da qui, cosa utile nel caso vogliate fare modifiche o ampliare il dataset con immagini più attuali e magari aggiungere falsi positivi per esempio attraverso l'uso di bandane o foulard. Per fare ciò avremmo quindi bisogno di un dataset, ovvero una serie di immagini classificate e suddivise in due macro gruppi: con mascherina o senza mascherina.

Nel nostro caso il sopraccitato dataset (https://drive.google.com/drive/folders/1XDte2DL2Mf_hw4NsmGst7QtYoU7sMBVG?usp=sharing) consta di 3835 foto suddivise in 1919 del primo tipo e 1916 del secondo, che dovrà essere preso in esame dallo script Python `train_mask_detector.py`.

```
ap = argparse.ArgumentParser()
ap.add_argument("-d", "--dataset", required=True,
                help="path to input dataset")
ap.add_argument("-p", "--plot", type=str,
                default="plot.png",
                help="path to output loss/accuracy plot")
ap.add_argument("-m", "--model", type=str,
                default="mask_detector.model",
                help="path to output face mask detector
model")
args = vars(ap.parse_args())

INIT_LR = 1e-4
EPOCHS = 20
BS = 32
```

Nella parte iniziale troviamo l'inizializzazione di alcune costanti come per esempio il percorso file `.model` in uscita dall'addestramento, il percorso del file dove verrà stampato il grafico report dell'ap-

prendimento, l'indice iniziale di apprendimento, le epoche, ovvero il numero di presentazioni alla rete di tutti i pattern del nostro training set e la dimensione del batch, ovvero il numero di campioni che verranno propagati attraverso la rete.

```
imagePaths = list(paths.list_images(args["dataset"]))
data = []
labels = []

for imagePath in imagePaths:
    label = imagePath.split(os.path.sep)[-2]
    image = load_img(imagePath, target_
size=(224, 224))
    image = img_to_array(image)
    image = preprocess_input(image)
    data.append(image)
    labels.append(label)

data = np.array(data, dtype="float32")
labels = np.array(labels)

lb = LabelBinarizer()
labels = lb.fit_transform(labels)
labels = to_categorical(labels)

(trainX, testX, trainY, testY) = train_test_
split(data, labels,
        test_size=0.20, stratify=labels, random_
state=42)

aug = ImageDataGenerator(
    rotation_range=20,
    zoom_range=0.15,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.15,
    horizontal_flip=True,
    fill_mode="nearest")

baseModel = MobileNetV2(weights="imagenet", inclu-
de_top=False,
    input_tensor=Input(shape=(224, 224, 3)))

headModel = baseModel.output
headModel = AveragePooling2D(pool_size=(7, 7))
(headModel)
headModel = Flatten(name="flatten")(headModel)
headModel = Dense(128, activation="relu")(headModel)
headModel = Dropout(0.5)(headModel)
```




```

headModel = Dense(2, activation="softmax")(headModel)
model = Model(inputs=baseModel.input,
outputs=headModel)

for layer in baseModel.layers:
    layer.trainable = False

print("[INFO] compiling model...")
opt = Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
model.compile(loss="binary_crossentropy",
optimizer=opt,
metrics=["accuracy"])

print("[INFO] training head...")
H = model.fit(
    aug.flow(trainX, trainY, batch_size=BS),
    steps_per_epoch=len(trainX) // BS,
    validation_data=(testX, testY),
    validation_steps=len(testX) // BS,
    epochs=EPOCHS)

```

Ora andiamo a caricare la rete MobileNetV2 ed inseriamola ai metodi forniti per l'appunto dalle due librerie open source (citare all'inizio). Queste ultime daranno in uscita un modello con il quale andremo a vagliare la sua capacità decisionale. Il MobileNetV2 è un'architettura di rete neurale di convoluzione il cui scopo è quello di essere molto ottimizzata per poter funzionare correttamente su dispositivi mobile. I comandi necessari sono.

```

print("[INFO] evaluating network...")
predIdxs = model.predict(testX, batch_size=BS)
predIdxs = np.argmax(predIdxs, axis=1)

```

Abbiamo così ottenuto il modello sotto forma di file che andremo a caricare all'interno del nostro algoritmo (**Fig. 6**).

L'ALGORITMO

Una volta ottenuto il .model responsabile della determinazione della predizione con/senza mascherina, andiamo quindi a focalizzarci su come poter acquisire e trattare agilmente un'immagine attraverso la libreria di OpenCV. Il primo passo è quindi estrarre da quello un'immagine che possa essere trattata:

```

import imutils
import time

```

```

import os
import cv2
import numpy as np
..
image = cv2.imread(Percorso dove trovare la foto/
la foto)

```

Con un'immagine da disco è molto semplice, ma anche con un video la cosa risulta altrettanto facile, ciò in quanto ogni video può essere visto come una serie di immagini, di frame. Ora sarà sufficiente inserire un ciclo while nel quale guarderemo lo stato del video e leggeremo frame per frame:

```

import imutils
import time
import os
import cv2
import numpy as np
..
cap = cv2.VideoCapture(Percorso dove trovare il
video/il video)

if (cap.isOpened()== False):
    print("Error opening video stream or file")
while(cap.isOpened()):
    ret, frame = cap.read()
    ..
    cv2.imshow('Frame',frame)

```

Infine arriviamo alla configurazione più idonea per il nostro obiettivo, ovvero integrare anche il sensore camera:

```

from picamera.array import PiRGBArray
from picamera import PiCamera
import imutils
import time
import os
import cv2
import numpy as np
..
camera = PiCamera()
camera.resolution = (640, 480)
camera.framerate = 32
rawCapture = PiRGBArray(camera, size=(640, 480))

for frame in camera.capture_continuous(rawCapture,
format="bgr", use_video_port=True):
    frame = frame.array
    ..
    cv2.imshow('Frame',frame)

```



Il frame estrapolato può essere anche scalato (ridotto o ingrandito) a seconda delle necessità. Dopo questa fase di introduzione alla libreria per la manipolazione video cominciamo a costruire quello che è l'algoritmo chiave per usare al meglio il nostro modello decisionale. Per far ciò come già fatto in precedenza carichiamo tutte le costanti indispensabili per la buona riuscita del lavoro:

```
ap = argparse.ArgumentParser()
ap.add_argument("-f", "--face", type=str,
                default="face_detector",
                help="path to face detector model directory")
ap.add_argument("-m", "--model", type=str,
                default="mask_detector.model",
                help="path to trained face mask detector
model")
ap.add_argument("-c", "--confidence", type=float,
                default=0.5,
                help="minimum probability to filter weak
detections")
args = vars(ap.parse_args())
print("[INFO] loading face detector model...")
prototxtPath = os.path.sep.join([args["face"], "de-
ploy.prototxt"])
weightsPath = os.path.sep.join([args["face"],
                                "res10_300x300_ssd_iter_140000.caffemodel"])
faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)
print("[INFO] loading face mask detector model...")
maskNet = load_model(args["model"])
```

Come si può notare abbiamo portato dentro anche un altro modello: questo è indispensabile per andare a capire e isolare i vari volti all'interno del frame che saranno poi sottoposti all'analisi del nostro modello. Pre addestrato con Caffe, noto framework per il Deep Learning sviluppato dal Berkeley AI Research, il nostro modello di Intelligenza Artificiale offre un'identificazione del volto quasi perfetta. Questo passo può essere affrontato anche attraverso il metodo di Haar, ovvero un metodo di rilevamento di oggetti efficace proposto da Paul Viola e Michael Jones nel loro articolo "Rilevamento rapido di oggetti utilizzando una cascata potenziata di funzionalità semplici" nel 2001. Si tratta di un approccio basato sull'apprendimento automatico in cui la funzione a cascata è addestrata da molte immagini positive e negative. Una volta caricati i nostri modelli andiamo a manipolare l'immagine creando un frame 300x300 per identificare i vari volti presenti nella scena.



```
(h, w) = frame.shape[:2]
blob = cv2.dnn.blobFromImage(frame, 1.0, (300, 300),
                              (104.0, 177.0, 123.0))
faceNet.setInput(blob)
detections = faceNet.forward()
```

Fig. 6
Rappresentazione grafica del modello di apprendimento.

Per ogni volto di cui sospettiamo andremo ad analizzare il valore attribuito è superiore rispetto alla soglia e solo in tal caso eseguiremo la ROI del viso con il modello per determinare se il soggetto inquadrato risulta con o senza mascherina. Una ROI ci permette di operare su un insieme rettangolare dell'immagine, per questo è molto utilizzata soprattutto per delineare quelli che sono i marcatori primari quali occhi, naso e bocca.

```
for i in range(0, detections.shape[2]):
    confidence = detections[0, 0, i, 2]
    if confidence > args["confidence"]:
        box = detections[0, 0, i, 3:7] *
np.array([w, h, w, h])
        (startX, startY, endX, endY) = box.
astype("int")
        (startX, startY) = (max(0, startX), max(0,
startY))
        (endX, endY) = (min(w - 1, endX), min(h - 1,
endY))

        face = frame[startY:endY, startX:endX]
        face = cv2.cvtColor(face, cv2.COLOR_BGR2RGB)
        face = cv2.resize(face, (224, 224))
        face = img_to_array(face)
        face = preprocess_input(face)
        faces.append(face)
        locs.append((startX, startY, endX, endY))
```

E infine, una volta che ci siamo assicurati di avere i volti, andiamo a sfruttare il modello per discriminare se questi hanno o meno il dispositivo di protezione personale e riportiamo tutto in una variabile.

```
faces = np.array(faces, dtype="float32")
preds = maskNet.predict(faces, batch_size=32)
```

A questo punto abbiamo tutto: nella variabile `locs` ci troviamo delle coordinate X, Y di dove sono i volti e della loro grandezza, mentre nella variabile `preds` possiamo notare se questi volti rispettano le misure di sicurezza imposte. Tuttavia, a livello video non abbiamo nessuna prova di ciò che abbiamo trovato, per cui è necessario un ulteriore step per poter marcare all'interno del frame quelli che sono i visi e i loro attributi. Proprio per questo per ogni volto andremo a fare un nuovo ciclo in cui modificheremo il frame aggiungendoci rettangoli e scritte in overlay laddove le coordinate estrapolate al punto precedente ci indicano un volto.

```
for (box, pred) in zip(locs, preds):
    (startX, startY, endX, endY) = box
    (mask, withoutMask) = pred
    label = "Mask" if mask > 0.85 else "No Mask"
    color = (0, 255, 0) if label == "Mask" else
    (0, 0, 255)
    label = "{: .2f}%".format(label, max(mask,
    withoutMask) * 100)
    if label=="Mask":
        faceMask=faceMask+1
    else:
        faceNoMask=faceNoMask+1
    cv2.putText(frame, label, (startX, startY -
    10),
                    cv2.FONT_HERSHEY_SIMPLEX,
    0.45, color, 2)
    cv2.rectangle(frame, (startX, startY), (endX,
    endY), color, 2)
```

Per quanto riguarda la quarta riga, teoricamente sarebbe possibile discriminare anche con una condizione come `mask>withoutMask`, ma è molto leggera e fragile, è quindi preferibile preferire il confronto con un valore percentuale. È anche possibile aggiungere un'ulteriore segnalazione dei risultati: il canale audio; infatti, come si può notare dalla porzione di codice precedente, è possibile contare i volti e quindi inserire una segnalazione diversa dal solo video. In particolare è preferibile un thread che si occupi di questo aspet-

Guarda Mask Detector in azione su YouTube!



Vuoi vedere in azione il nostro Mask Detector? Nulla di più semplice: collegati alla pagina <https://youtu.be/04cQTFmSB5Q>. Potrai anche vedere i video di moltissimi altri nostri progetti proposti in passato.

to, per non appesantire il già difficile compito di discernere e mostrare a video. Questo inoltre potrà essere invocato magari una volta ogni 10 frame in modo da avere la possibilità di raccogliere diversi fotogrammi su cui basare la sua segnalazione.

```
if index==10 :
    _thread.start_new_thread(speakTTS,
    (faceNoMask,faceMask,1))
    index = 0
    index = index + 1

def speakTTS(faceNoMask,a,b):
    if faceNoMask == 0 :
        if a==0 :
            os.system('echo "" | festival --tts')
        else:
            if a==1 :
                os.system('echo "Pass the check"
                | festival --tts')
            else:
                os.system('echo "All the
                '+str(a)+' persons pass the check" | festival --tts')
        else:
            if faceNoMask == 1 :
                os.system('echo "There is
                '+str(faceNoMask)+' person without mask protection" |
                festival --tts')
            else:
                os.system('echo "There are
                '+str(faceNoMask)+' persons without mask protection"
                | festival --tts')
```



Per eseguire il nostro algoritmo apriamo una nuova finestra di terminale e diamo il comando:

```
python3 cam_video.py
```

Il risultato ottenuto sarà tipo quello in **Fig. 7**, con il volto nel riquadro verde se il sistema avrà rilevato la mascherina e in rosso in caso contrario; il tutto in real-time per poter valutare ogni situazione nel più breve tempo possibile senza creare lunghe file. Il sistema giustifica l'esito con un'indicazione allegata circa la probabilità che la situazione mostrata sia quella vera, ovvero con Mask in verde e la percentuale di probabilità se ritiene che il volto abbia la mascherina, ovvero No mask in rosso e relativa percentuale di probabilità che il volto inquadrato sia privo della mascherina.

Insomma, ci viene restituito non solo un verdetto, ma anche un modo di giudicare quanto il sistema stia facendo la cosa giusta. E non è poco...

CONCLUSIONI

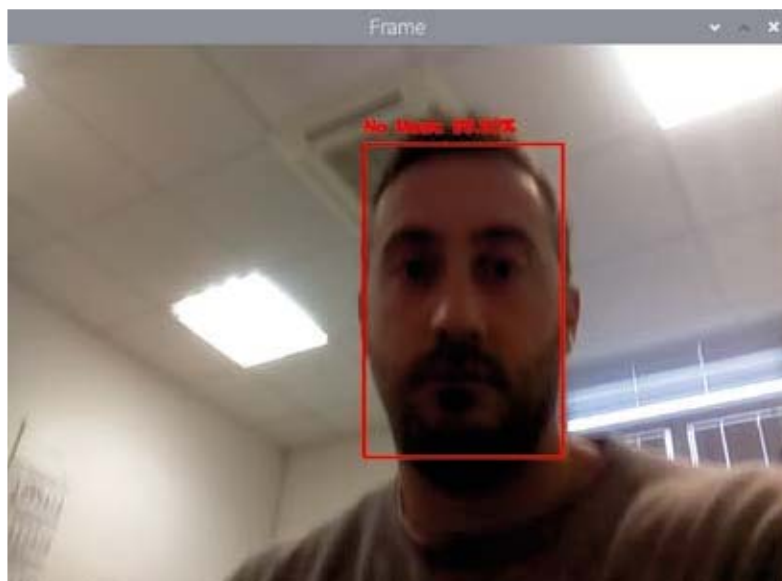
L'attività ovviamente non si ferma qui: infatti potremmo avere due ulteriori implementazioni che potrebbero portare questo progetto didattico ad un livello molto più alto. In primis si può rendere completamente fluido il flusso video in uscita e soprattutto normalizzare i tempi quando si hanno scene molto fitte, magari attraverso una parallelizzazione del tracciamento dei volti attraverso diversi thread opportunamente semaforizzati.

Vorremmo sottolineare l'aspetto "opportunamente semaforizzati" poiché tutti i worker dovranno poi convergere nella funzione di predizione globale, per cui dovranno svolgere il lavoro e attendersi uno con l'altro al fine di fornire un'unica risposta in uscita. Con pochi volti questa implementazione non migliora, se non di poco, l'uscita video, mentre in caso di 10/15 volti cambia radicalmente i tempi.

Un'altra realizzazione potrebbe essere migrare questo lavoro in un diverso sistema operativo come Ubuntu Server, sempre su Raspberry Pi.

In questo caso non si avrebbe alcuna UI sulla quale vedere i volti e il riconoscimento ma questa funzionerebbe solamente da camera intelligente.

Una volta elaborato il dato, la camera potrebbe avere un sistema di pushing delle informazioni in un Cloud dal quale verrebbe ripartito verso le unità collegate; ciò significa che, magari, attraverso un'app su smartphone possiamo vedere in tempo reale il video elaborato dal mio dispositivo ed avere informazioni sui volti e se questi indossano o meno una mascherina.



Sarà poi compito dell'app mostrare il flusso video, così come nella nostra versione, con volti marcati da quadrati di colore rosso e verde; oppure potremo ricevere solamente l'informazione base, tuttavia in questo caso verrebbero coinvolti elementi in più rispetto a una situazione di base come quella sviluppata in questo articolo: oltre alla capacità di sviluppare un'app (che richiede competenze del mondo Android o iOS) dovreste avere conoscenze di protocolli light di trasmissione dati/video come MQTT o RTMP e saper allestire un server per accogliere e ritrasmettere i dati. Sicuramente il compito è impegnativo ma stimolante e può rappresentare l'occasione per acquisire nuove competenze. □

Fig. 7
Il volto evidenziato con la percentuale di riuscita del rilevamento posta sopra il riquadro.



Cosa occorre?

Il materiale utilizzato in questo progetto è disponibile presso Futura Elettronica. La board Raspberry Pi 4 da 8GB (cod. RPI4-8GB) è in vendita al prezzo di Euro 99,00, ed è disponibile anche in un set con alcuni accessori (cod. RPI4FULLGB) a Euro 129,00. La telecamera da SMPx (cod. YGO16) è disponibile a Euro 14,00. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

UN TEAM DI PROFESSIONISTI PER IL TUO PROGETTO



oltre **1500** progetti sviluppati
26 anni di attività
più di **15** progettisti



L'area R&D di Futura Group studia, progetta e realizza prototipi per differenti settori utilizzando tecnologie innovative. Nel team di Futura Group fa parte il personale interno e un numero elevato di collaboratori esterni, ciascuno specializzato nel proprio settore (Disegno CAD – Raspberry Pi – Arduino – Audio – Sviluppo firmware – Linux – ecc). La disponibilità di oltre 5000 prodotti elettronici e componenti consente di ridurre i tempi di test e sviluppo e di realizzare facilmente il primo prototipo. La decennale esperienza nel settore e la necessità di ottimizzare i tempi e i costi, ha portato Futura Group a selezionare una serie di terzisti per la produzione di piccola o grande serie.

I SERVIZI CHE TI OFFRIAMO



Progetto Completo

Partendo dalla tua idea progettiamo per te la soluzione.



Sviluppo PCB

Ottimizziamo il tuo prototipo disegnando un PCB compatto secondo le tue specifiche.



Scrittura Codice

Scriviamo il codice del core per dare vita alla tua applicazione.

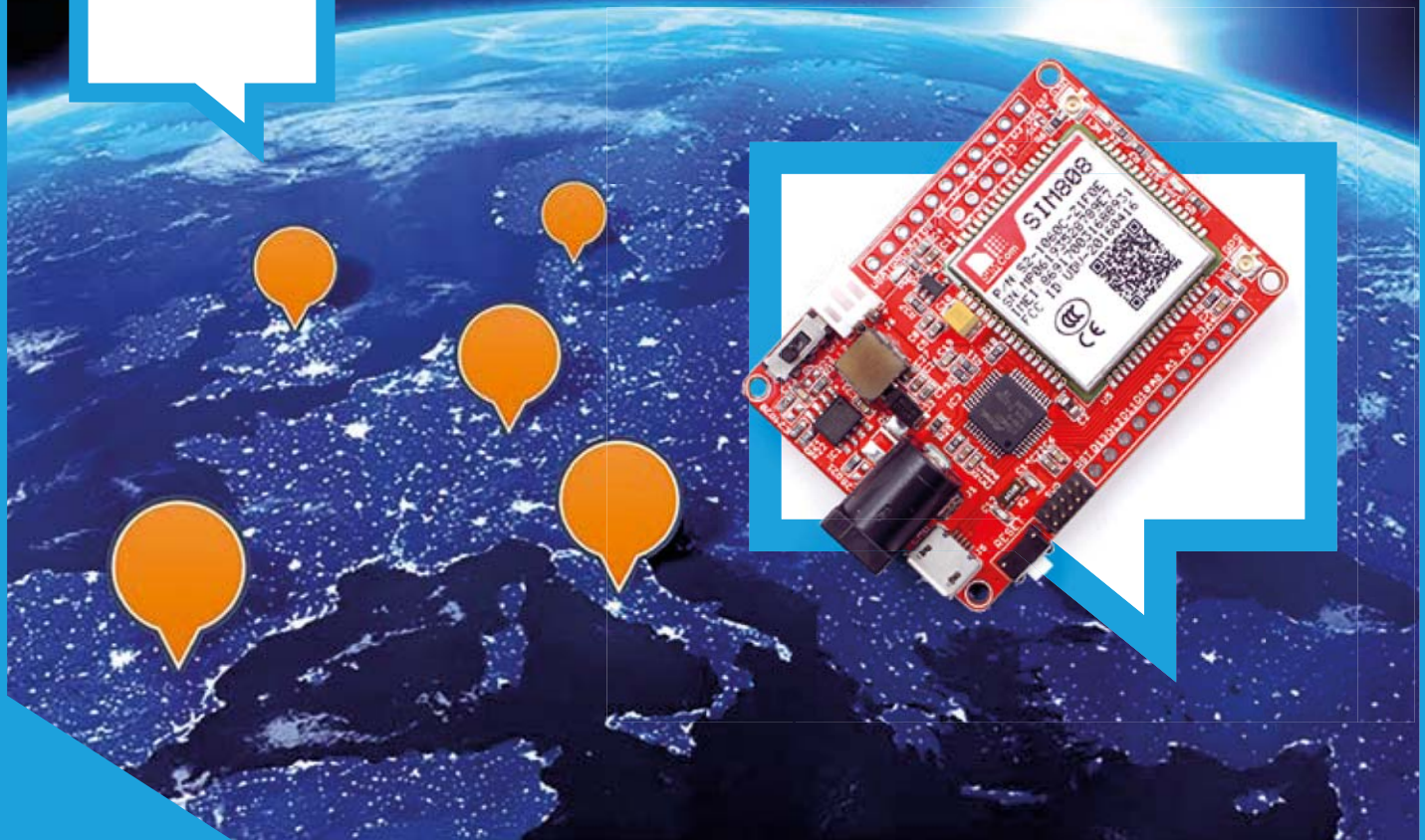


Produzione Progetto

Portiamo il prototipo alla fase di produzione seguendo per te tutte le fasi.

GPS TRACKER

dell'Ing. **MIRCO SEGATELLO**



Realizziamo un localizzatore basato sul modulo SIM808 utilizzando, per l'invio dei dati di posizionamento, il servizio on-line ThingSpeak.

C

on il termine "Tracker" si indica un dispositivo in grado di registrare e mantenere traccia della sua posizione tramite la lettura dei dati provenienti dalla rete satellitare e di inviare questi dati all'utente sfruttando un rete di comunicazione, normalmente la rete GSM. Capita infatti di avere la necessità

di mantenere traccia in tempo reale della posizione di mezzi o persone in diversi contesti. Si va dal classico tracciamento dei mezzi aziendali e/o commerciali al tracciamento dello spostamento di animali sia domestici che selvatici, ma anche come sistema di soccorso per rintracciare persone disperse durante escursioni, oppure per mantenere sotto controllo la spedizione di un carico particolarmente importante. Viste le piccole dimensioni del dispositivo in oggetto potrebbe essere usato per rilevare la posizione di una persona a sua insaputa, ma questo è un uso improprio vietato dalla legge.

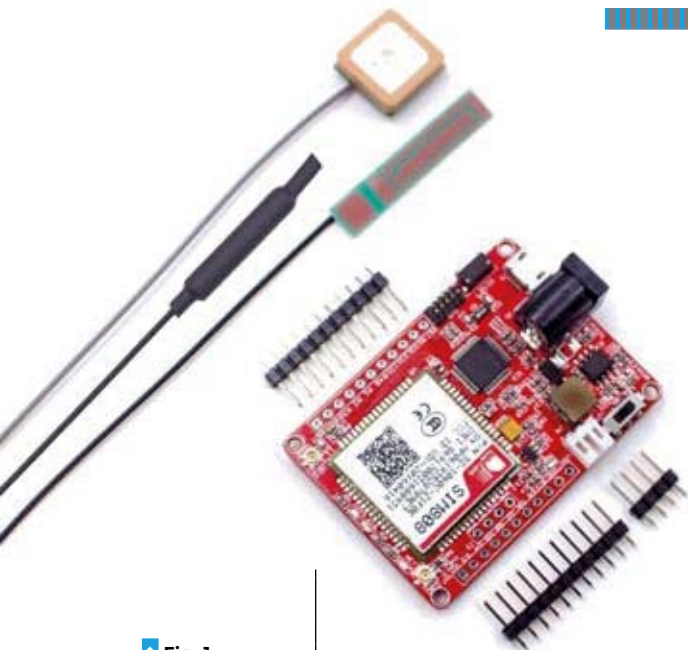


Fig. 1
Scheda
Maduino Zero
con SIM808.

IMPLEMENTAZIONE

Per raggiungere il nostro scopo necessitiamo di un microcontrollore interfacciato con un modulo GPS per leggere la posizione e di un modulo GSM/GPRS per inviare in tempo reale e da qualunque luogo i dati acquisiti. Fortunatamente possiamo disporre di un circuito che raccolga tutte queste funzioni, stiamo parlando della scheda Maduino Zero SIM808 distribuita da Futura Elettronica con codice GPSPADUINO. Questa scheda è stata appositamente progettata per svolgere la funzione di GPS Tracker integrando tutto il necessario, compresa la gestione dell'alimentazione.

Il nome stesso fa capire che questa scheda è equipaggiata con un microcontrollore ATSAM21G18 con core a 32bit e frequenza di clock di 48MHz, lo stesso utilizzato nelle schede Arduino Zero. Nella scheda è saldato il modulo SIM808 che integra al suo interno le funzioni GSM, GPRS, GPS e bluetooth, e giusto per non farsi mancare nulla anche un lettore di schede SD. Maduino SIM808 viene fornita completa di antenne (GSM, GPS e BT) e dei strip a saldare per favorire i collegamenti con dispositivi esterni (**Fig. 1**).

Le principali caratteristiche della scheda Maduino (arduino compatibile) SIM808 sono le seguenti:

- alimentazione a batteria 3,4÷4,2V;
- alimentazione da plug 6÷12V;
- ricarica batteria da plug di alimentazione;
- connettore microUSB solo per comunicazione
- microcontrollore ATSAM21G18, 32-Bit ARM Cortex M0+;
- clock speed 48 MHz;
- connettore microSIM;
- sistema integrato per la gestione della ricarica della batteria al litio;
- comandi AT con "auto baud";
- quad-band 850/900/1800/1900 MHz;
- invio e ricezione dati GPRS (TCP/IP, HTTP, ecc.);
- invio e ricezione di SMS;
- GPS/CNSS con 66 canali;
- cold start di 32 secondi;
- warm start di 1 secondo;
- accuratezza di circa 2,5 metri;
- supporto per Bluetooth 4.0;

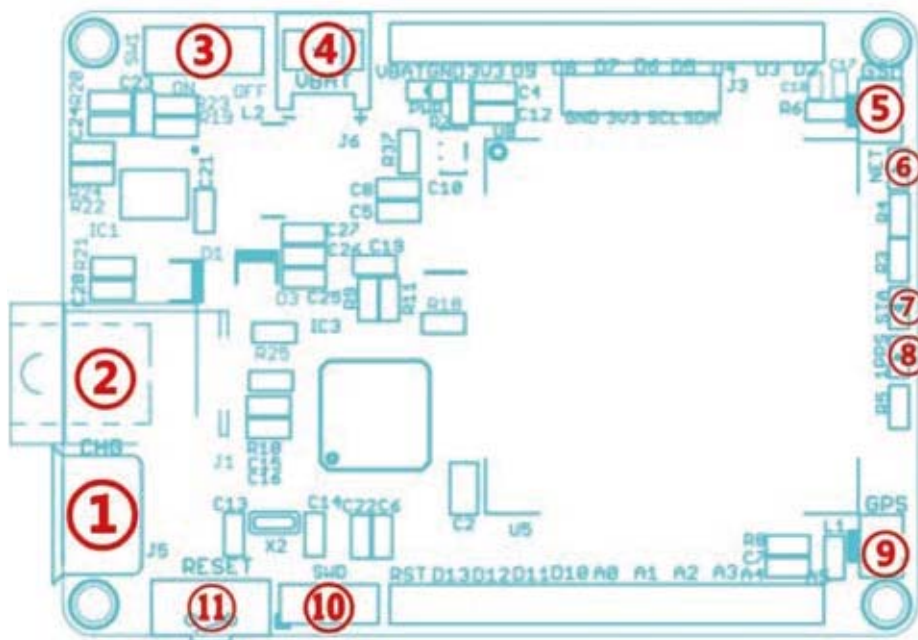


Fig. 2
Interfacce della
scheda Maduino
Zero.

Tabella 1
Funzione svolta dai LED.

LED	FUNZIONE
LED rosso sotto la scheda	Alimentazione da PLUG presente
LED rosso sopra la scheda	Scheda Alimentata
LED blu vicino SIM808	Modulo SIM808 alimentato
LED giallo vicino antenna GSM	Lampeggio veloce modulo attivo ma non connesso alla rete Lampeggio lento modulo attivo ma non connesso alla rete
LED giallo vicino antenna GPS	Lampeggio 1 sec segnale GPS fix e dati disponibili

- interfacce I²C/SPI/UART/GPIO;
- temperatura di funzionamento -40 ÷ + 85°.

La scheda Maduino è dotata di diversi connettori e LED di segnalazione, per la loro identificazione il loro utilizzo si faccia riferimento alla **Fig. 2** ed alla descrizione di seguito riportata:

1. Connettore microUSB: utilizzato per la comunicazione con il PC (non fornisce alimentazione).
2. Plug DC: ingresso alimentazione della scheda (6-12VDC).
3. Interruttore: per accendere e spegnere il sistema.
4. VBAT: per la connessione di una batteria al litio da 3,7 volt.
5. GSM: antenna del modulo GPRS/GSM.
6. NET: LED che indica lo stato della rete SIM808.
7. STA: LED che indica lo stato del modulo SIM808.
8. 1PPS: LED che indica il FIX del GPS.
9. GPS: antenna del GPS.
10. SWD: utilizzato per il caricamento del bootloader.
11. RESET: riavvia il microcontrollore ATSAM21G18.

Giusto per dimostrare la semplicità di interfacciamento esterno, useremo un modulo DHT11 per la misura della temperatura e umidità ambientale, nulla vieta di rilevare qualsiasi altra grandezza utile allo scopo. Il DHT11 funziona bene anche a 3,3V e comunica con un solo filo, inoltre la collaudata libreria ne semplifica l'utilizzo.

Per quanto riguarda l'alimentazione di Maduino SIM808 i progettisti hanno previsto due possibilità: una alimentazione esterna da connettere al plug che fornisca una tensione compresa tra 6 e 12V con almeno 2 ampere, oppure una batteria singola cella Li-Po da connettere all'apposito connettore JST passo 2 mm.

La commutazione tra alimentazione plug e batteria avviene in automatico, se presenti entrambe, la tensione dal plug verrà usata per alimentare la scheda e ricaricare la batteria con circa 270 mA.

UTILIZZO PRATICO

Per il primo test non inseriamo la SIM ma colleghiamo le due antenne GSM e GPS ed il sensore DHT11 come in **Fig. 3**. Il sensore DHT11 viene connesso all'alimentazione 3V3 e GND mentre il filo dei dati è connesso al pin D6.

Alimentiamo la scheda con un alimentatore o con la batteria, ed usiamo l'interruttore per accendere la scheda; è necessario assicurarsi una buona alimentazione perché il modulo SIM808 ha dei picchi di assorbimento notevoli che, se non forniti correttamente, ne compromettono il funzionamento. Colleghiamo la scheda al PC tramite il connettore microUSB, fate riferimento alla **Tabella 1** per conoscere la funzione svolta da ciascun LED presente nella scheda.

Per utilizzare Maduino Zero con l'IDE di Arduino è necessario configurare l'ambiente di lavoro ed installare i driver di comunicazione, per fare questo recatevi nel tab gestore schede ed installate il pacchetto Arduino SAMD Boards cercando la parola

Fig. 3
Maduino SIM808 con antenne e DHT11.

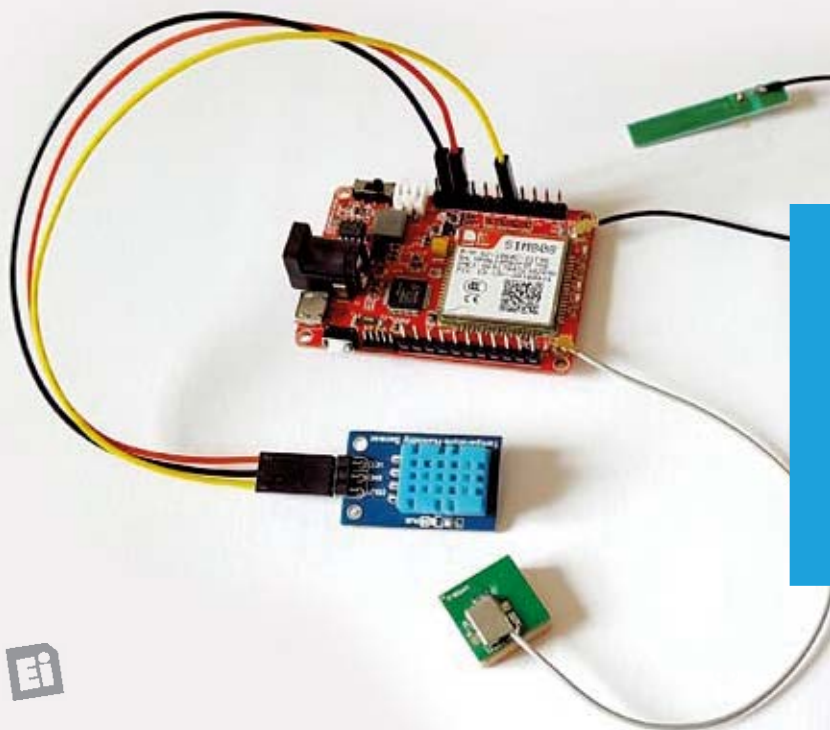
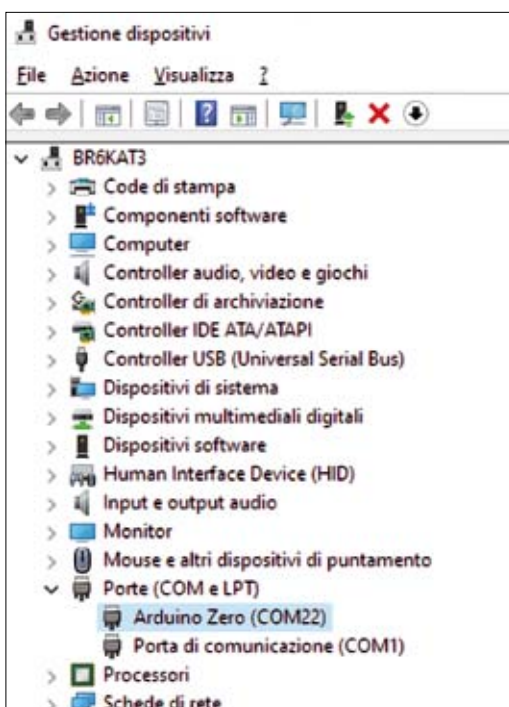


Fig. 4
Installazione
pacchetto SAMD
Boards.



Fig. 5
Maduino
riconosciuta come
scheda Arduino
Zero.



chiave "zero" (**Fig. 4**).

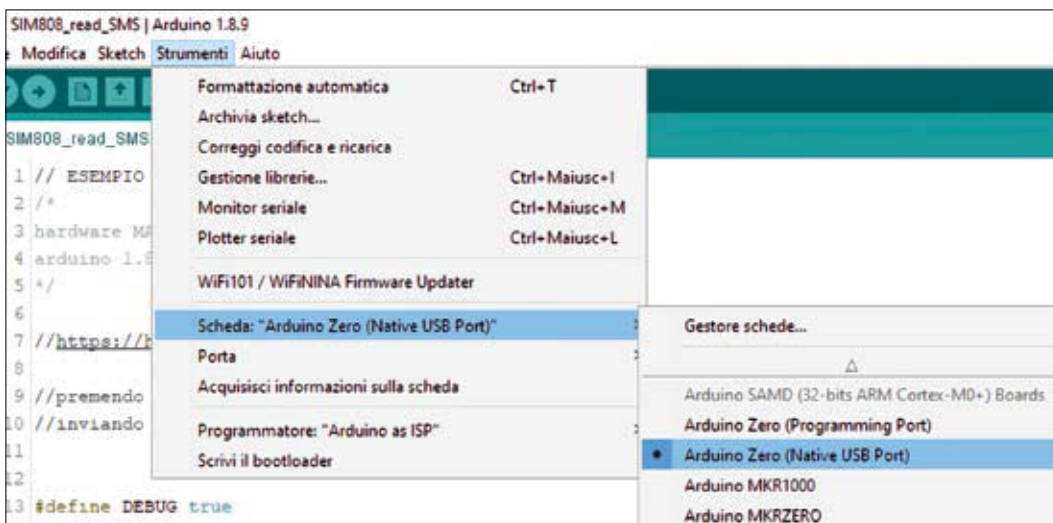
Questa installazione abiliterà l'IDE di Arduino a gestire le schede Arduino Zero ed installerà i relativi driver. Per assicurarvi che i driver siano correttamente configurati recatevi nella gestione dispositivi di Windows e verificate che nell'elenco delle COM sia presente la scheda Arduino Zero (**Fig. 5**).

Non vi resta che selezionare la scheda Arduino Zero (Native USB Port) nell'IDE di Arduino e caricare un programma di prova per verificare che tutto funzioni (il classico *blink* va benissimo).

Vediamo come gestire direttamente con comandi AT il modulo SIM808, pertanto carichiamo nella scheda lo sketch di nome *SIM808_AT_Commands.ino*; questo semplicissimo sketch convoglia i comandi di testo inviati da Serial Monitor di Arduino al modulo SIM808 e viceversa, ricordatevi di impostare Carriage Return (CR) nelle impostazioni in basso. In **Tabella 2** riportiamo i principali e più utili comandi AT.

In questo come anche in tutti gli altri sketch

Fig. 6
Selezione scheda
Arduino Zero per la
programmazione.



**Tabella 2**

Principali comandi AT per il modulo SIM808.

COMANDO	FUNZIONE
AT	Verifica se il modulo è attivo, risponde con OK
AT+GSV	Ritorna le caratteristiche del modulo SIM808
AT+CBC	Ritorna il valore di tensione della batteria
AT+CGNSPWR=1	Attiva il GPS
AT+CGNSSEQ=RMC	Imposta il tipo di uscita per i dati GPS in formato NMEA RMC
AT+CGNSINF	Richiesta dei dati GPS
AT+CGNSURC=1	Richiesta della lettura continua dei dati GPS (ogni secondo)
AT+CGNSURC=0	Ferma la lettura continua dei dati GPS
AT+CGPSSTATUS?	Verifica se il GPS ha il FIX
AT+CPIN?	Verifica se la SIM è correttamente installata
AT+COPS?	Ritorna l'operatore di telefonia
AT+CREG?	Verifica lo stato della connessione alla rete
AT+CSQ	Verifica dell'intensità del segnale

proposti troverete nella routine *Setup* la procedura *SIM808_powerON()* che è necessaria per attivare il modulo.

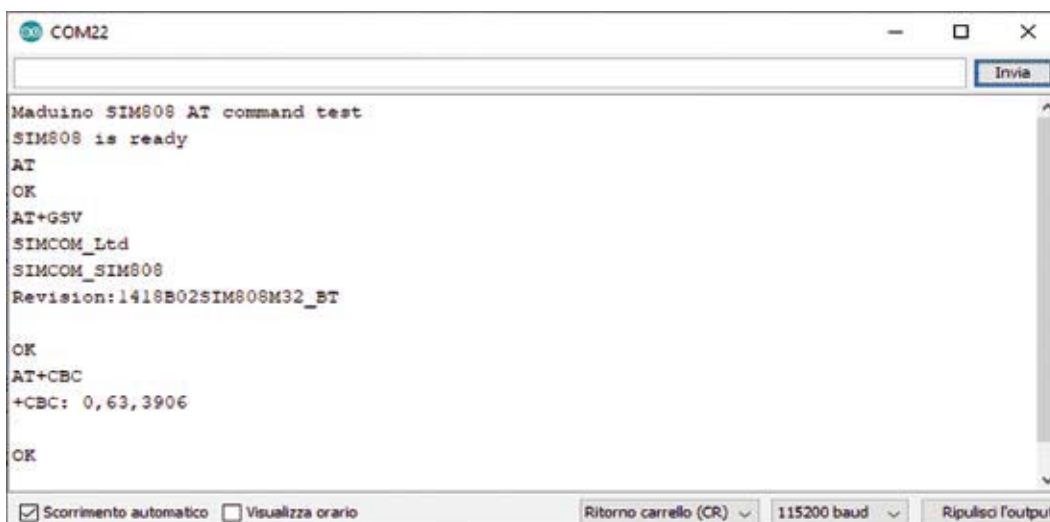
Cominciamo con l'invio del comando AT ed attendiamoci la risposta OK a significare che il modulo è operativo, inseriamo poi il comando AT+GSV per verificare la versione firmware e magari il comando AT+CBC per verificare la tensione di alimentazione, si veda la **Fig. 7**.

UTILIZZO DEL GPS

Giunti a questo punto possiamo occuparci di cose più serie e cominciamo con l'utilizzo del modulo

GPS per la lettura delle coordinate spaziali; assicuratevi che la piccola antenna sia vicina ad una finestra, meglio ancora se all'esterno. Iniziamo inviando il comando AT+CGNSPWR=1 per attivare il GPS, poi AT+CGNSSEQ=RMC per impostare i dati di uscita nella sequenza NMEA RMC (Time, date, position, course and speed data) e successivamente AT+CGNSINF per richiedere i dati (**Fig. 8**).

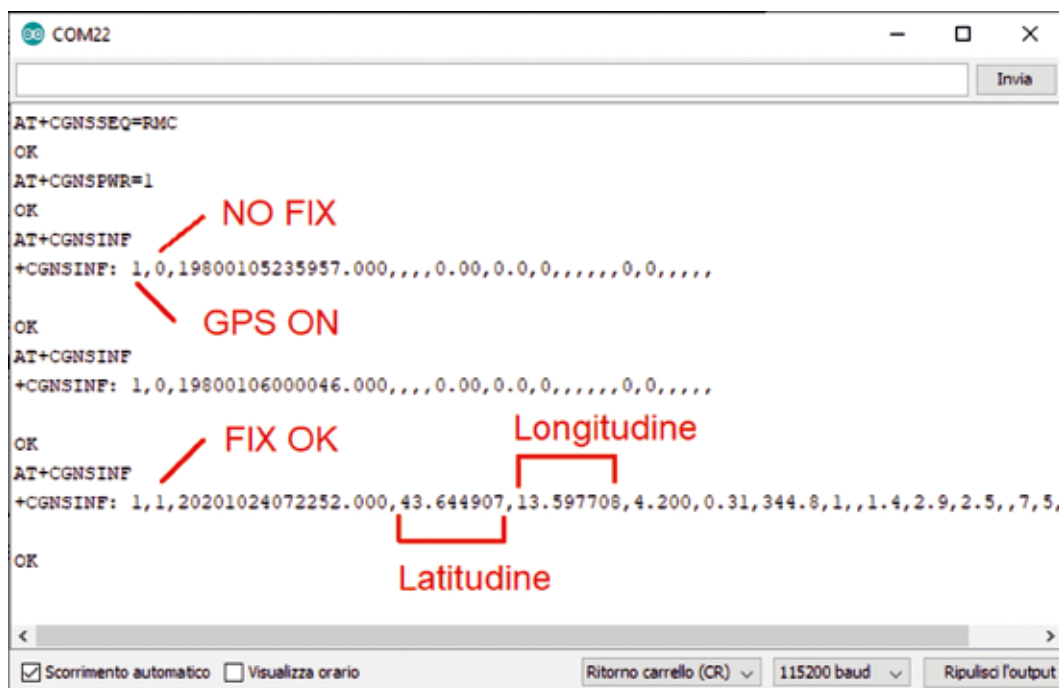
Per velocizzare tutte le operazioni abbiamo scritto un apposito sketch che svolge tutte le operazioni sopra indicate in modo automatico ed estrapola anche i principali dati quali data e ora, coordinate e velocità e direzione, si chiama *SIM808_Get_GPS_*

**Fig. 7**

Utilizzo dei comandi AT.



Fig. 8
Schermata del
Serial Monitor per
il controllo del GPS
con comandi AT



parse.ino ed è fornito assieme ai file di questo articolo scaricabili dal sito della rivista. In **Fig. 9** potete vedere la schermata di Serial Monitor con lo sketch in esecuzione.

UTILIZZO DEL GSM/GPRS

Il passo successivo è il collaudo della parte

GSM/GPRS e per fare questo dobbiamo disporre di una SIM con attivato un contratto per il traffico dati. Potete optare per apposite SIM per IOT nelle quali si paga a consumo e sono sicuramente vantaggiose visto l'esiguo traffico dati che andremo ad utilizzare. Per fare delle prove la SIM del vostro telefono andrà più che bene,

Fig. 9
Schermata di
Serial Monitor per
lo sketch SIM808_
Get_GPS_parse.ino.

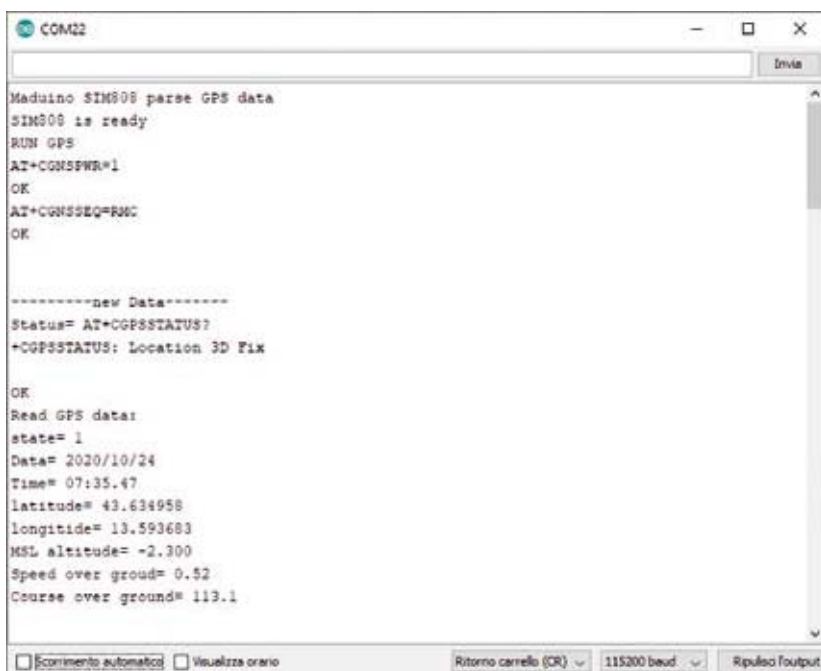


Fig. 10



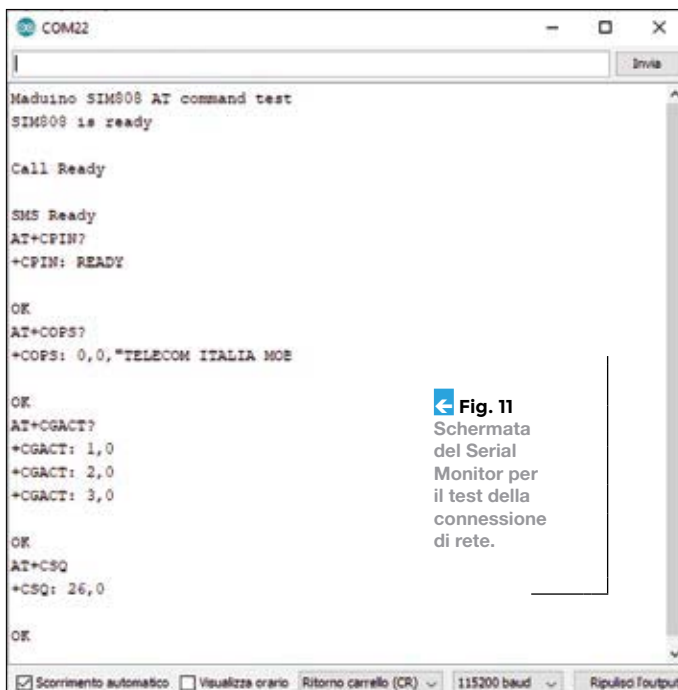


Fig. 11
Schermata
del Serial
Monitor per
il test della
connessione
di rete.

ma assicuratevi di aver disabilitato l'accesso alla SIM con codice. Inserite la SIM nell'apposito scomparto come visibile in **Fig. 10**, rigorosamente con la scheda Maduino spenta. Se il vostro smartphone utilizza una nano SIM utilizzate un piccolo adattatore per renderla compatibile con il formato micro-SIM di Maduino. Per evitare di ossidare i delicati contatti non toccateli mai con le mani, usate invece una pinzetta, la parte smussata è rivolta verso l'esterno ed i contatti elettrici verso la scheda. Non appena riavviato il modulo SIM808 cercherà già di registrarsi in rete, con l'utilizzo dello sketch *SIM808_AT_Commands.ino* possiamo eseguire delle interessanti interrogazioni utilizzando i comandi riportati nella **Tabella 2**; in **Fig. 11** la schermata di Serial Monitor di esempio.

IL CLOUD THINGSPEAK

Giunti a questo punto e dopo aver preso familiarità con questo modulo possiamo compiere il passo successivo, ovvero l'implementazione del nostro GPS tracker. Avremmo bisogno di un cloud sul quale poter salvare i dati inviati dal nostro dispositivo e poterli poi visualizzare in una dashboard, per questo ci appoggiamo al portale Thingspeak. Da Thingspeak create un nuovo canale ed inserite i field come riportato nella **Fig. 12**.

Nella sezione API Keys annotatevi la Write API Key e Read API Key che ci serviranno successivamente nello sketch per Arduino. Spostatevi nel tab Private View ed inserite due grafici uno per la temperatura ed uno per l'umidità impostando i parametri come in **Fig. 13**. Aggiungete due campi numerici per visualizzare latitudine e longitudine ed uno per

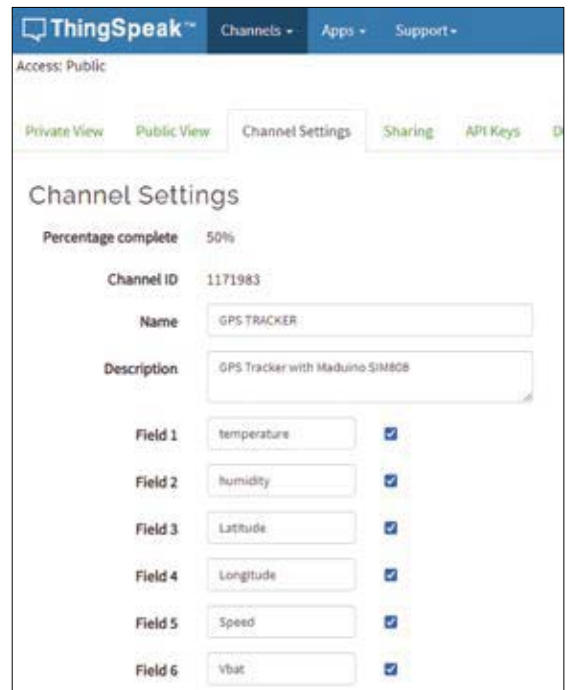


Fig. 12
Canale Thingspeak
configurato.

↓ Listato 1 (MATLAB Visualizations GPS Tracker)

```
% GPS data in a map
mapData = thingSpeakRead(1171983,'ReadKey','MIAREADKEY','numpoints',10,'outputformat','timetable');
geoscatteer(mapData.Latitude, mapData.Longitude,'r', 'fill');
geobasemap('street'); %seleziona tipo di vista 'satellite'
title('GPS Position');
latMin = min(mapData.Latitude)-0.01;
latMax = max(mapData.Latitude)+0.01;
lonMin = min(mapData.Longitude)-0.01;
lonMax = max(mapData.Longitude)+0.01;
geolimits([latMin latMax], [lonMin lonMax]);
```


Field 1 Chart Options

Title:	Temperature	Timescale:	
X-Axis:		Average:	
Y-Axis:		Median:	
Color:	#d62020	Sum:	
Background:	#ffffff	Rounding:	
Type:	line	Data Min:	
Dynamic?:	true	Data Max:	
Days:		Y-Axis Min:	0
Results:	20	Y-Axis Max:	50

Fig. 13
Impostazione
per il grafico della
temperatura.

visualizzare la tensione della batteria.
La parte più importante è la mappa con il tracciato che però non è disponibile nativamente in Thingspeak, pertanto va implementata con codice Matlab, questo è proprio il vantaggio di Thingspeak.

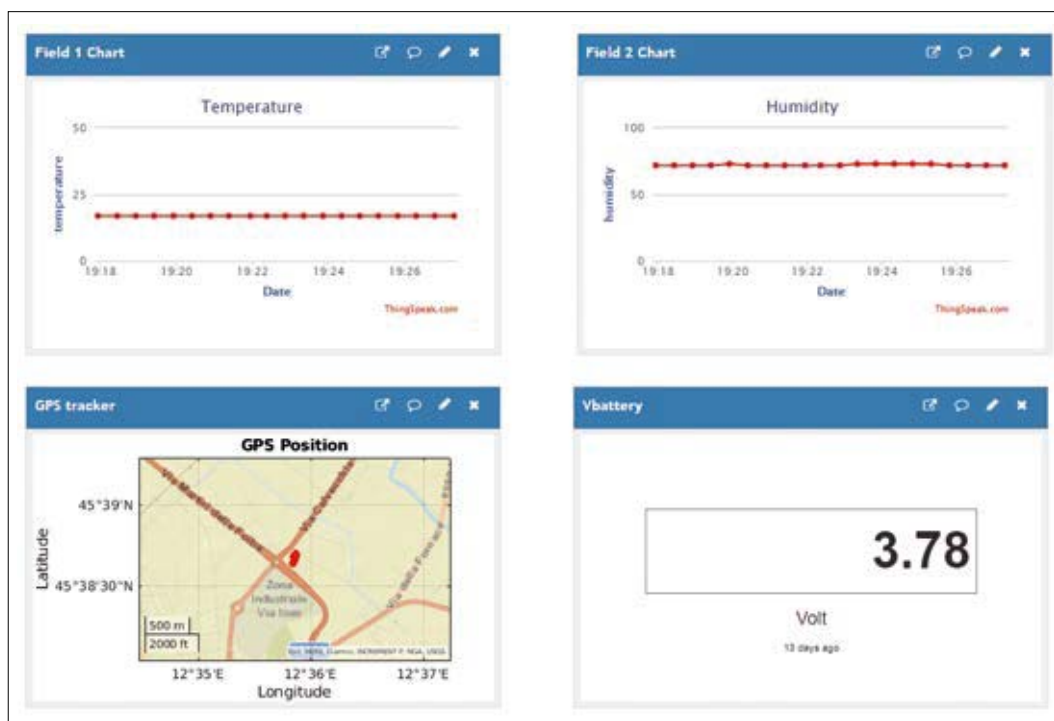
ak. In alto nel tab Apps selezionate MATLAB Visualizations e create una nuova di nome GPS Tracker ed inserite il **Listato 1**.

La prima riga permette di leggere gli ultimi 10 valori (valore modificabile a piacere) delle coordinate ed inserirle in un vettore, al posto di MIAREADKEY va inserito il codice Read API Key che trovate nel tab API Keys. Successivamente con il comando *geosscatter* viene creata la mappa con dei pallini 'r' riempiti 'fill' che con il comando *geobasemap* disegneremo sopra una cartina stradale. Dai dati acquisiti andremo a vedere quali sono le coordinate estreme e di conseguenza lo zoom della mappa. Non ci resta che aggiungere alla dashboard un riquadro nel quale fare girare lo script, si veda **Fig. 14**.

SKETCH ARDUINO

Vediamo ora la stesura dello sketch che dovrà girare all'interno dell'MCU di Maduino. Lo schema generale riportato in **Fig. 15** prevede che ad

Fig. 14
Dashboard di
Thingspeak con la
visualizzazione dei
dati.



Listato 2 (configurazione parametri)

```
const String apiKey="MIAWRITEKEY";           // Thingspeak write api key
const String APN="internet.coopvoce.it";      // provider APN
const float batCutOff=3.0;                    // battery volt level cutoff
const int UPDATE_INTERVAL_SECONDS = 20;       // Tempo di aggiornamento. Minimo è 20 sec
```

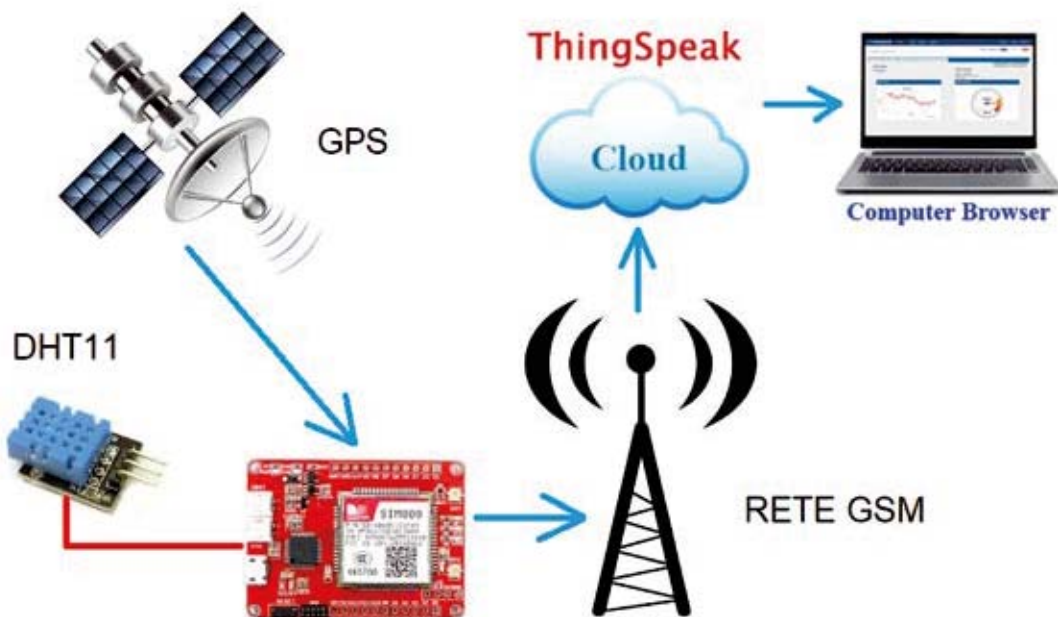


Fig. 15
Schema generale
dell'applicazione.

intervalli regolari (personalizzabili) la scheda Ma-
duino legge i dati dal sensore DHT11 e dal GPS
e li invia tramite la rete GSM al cloud Thingspeak
nel quale saranno visualizzati.

Il listato non è complicatissimo e prima di essere
caricato necessita di essere configurato con
alcuni parametri che riportiamo nel **Listato 2**. La
Write Key è quella fornita da Thingspeak, l'APN

del vostro provider è visibile nelle impostazioni
del vostro smartphone, a seguire l'intervallo
di invio dei dati. Per quest'ultimo dato consi-
derate che un veicolo che viaggia a 50Km/h in
20 secondi percorre poco meno di 300m e che
Thingspeak non accetta invii ad intervalli di tem-
po inferiori ai 20 secondi.

Nello sketch nella sezione Setup sono presenti

↓ Listato 3 (configurazione del modulo SIM808 per l'accesso alla rete GPRS)

```

sendData("AT+CPIN?", 1000, DEBUG); // Check if the SIM card is ready
sendData("AT+SAPBR=3,1,\"CONTYPE\",\"GPRS\"", 1000, DEBUG); // Bearer Settings for App. Based on IP
sendData("AT+SAPBR=3,1,\"APN\", \"\" + APN + \"\", 1000, DEBUG); // Set provider APN
sendData("AT+SAPBR=1,1", 1000, DEBUG); // Bearer Settings
sendData("AT+CGATT=1", 1000, DEBUG); // connected to the GPRS network

SerialUSB.print("GSM Status: ");
sendData("AT+CGREG?", 1000, DEBUG); // Network Registration Status
sendData("AT+CSQ", 1000, DEBUG); // signal quality
sendData("AT+CIFSR", 1000, DEBUG); // IP address
sendData("AT+CGATT?", 1000, DEBUG); // read if GPRS Service is attached
sendData("AT+HTTPINIT", 1000, DEBUG); // Initialize HTTP Service
  
```

↓ Listato 4 (invio dati a Thingspeak)

```

command = "AT+HTTPPARA=\\\"URL\\\",\\\"http://api.thingspeak.com/update.json?api_key=" + apiKey + "&field1="
+ (String)temperature + "&field2=" + (String)humidity + "&field3=" + Latitude + "&field4=" + Longitude +
"&field5=" + SOG + "&field6=" + VBAT + "\\\"";

sendData(command, 3000, DEBUG); //Set HTTP Parameters Value
sendData("AT+HTTPACTION=0", 1000, DEBUG); // HTTP POST //response: +HTTPACTION: 1,200,1 0=POST 1=GET
sendData("AT+HTTPREAD", 5000, DEBUG); // Read the HTTP server response
  
```

In questo articolo abbiamo usato parecchie sigle che potrebbero non essere note a tutti, pertanto riportiamo in questo riquadro il loro significato in modo esteso.

GNSS Acronimo di Global Navigation Satellite System, indica un sistema per la navigazione 3D utilizzabile sia da aerei sia da mezzi di superficie (navi, automobili ecc.), esso integra le funzioni dei satelliti di posizionamento del GPS con quelle di alcuni satelliti di telecomunicazioni e di stazioni di controllo terrestri.

GSM Acronimo di Global System Mobile, è lo standard di telefonia mobile più diffuso del mondo. La sua importanza ha indotto la maggior parte degli operatori internazionali di telefonia mobile a stipulare fra loro accordi per l'effettuazione del cosiddetto roa-

ming (commutazione automatica fra diverse reti).

GPRS Acronimo di Global Packet Radio Service. Si tratta della tecnologia che per prima ha permesso la navigazione in internet da dispositivi mobili e l'accesso a servizi multimediali.

SOG Acronimo di Speed Over Ground (Km/hour), indica la velocità di un mezzo che si muove nello spazio.

COG Acronimo di Course Over Ground (degrees), indica la direzione di marcia di un mezzo che si muove su di una superficie.

GPS FIX Indica che il ricevitore GPS ha catturato ed elaborato il segnale di un sufficiente numero di satelliti a determinare la posizione attuale ed i dati sono pronti per la lettura.

comando. La parte principale del listato che si occupa dell'invio dei dati è riportata nel **Listato 4**, essenzialmente viene creata una stringa che contiene il codice http necessario per andare a scrivere nei field di Thingspeak. La stringa viene quindi inviata con il comando *HTTPACTION* in modalità POST e successivamente si predispone la lettura della risposta con il comando *HTTPREAD*. Se la risposta è +HTTPACTION: 1,200,1 significa che l'operazione si è conclusa con successo. Con Maduino programmata ed operativa dovreste vedere su Thingspeak arrivare i dati ad intervalli regolari secondo quanto impostato, per i dati GPS ci vorrà più tempo perché è necessario che prima venga eseguito il FIX. Considerate che l'intero circuito assorbe circa 60mA che salgano a 130mA durante l'invio dei dati a Thingspeak, quindi con una batteria da 1100mah potrebbe reggere tranquillamente una decina di ore. Abbiamo previsto di monitorare la tensione ai capi della batteria e di intervenire spegnendo il modulo SIM808 qualora la tensione dovesse scendere sotto una certa soglia; la corrente scende a circa 10mA consentendo la salvaguardia della stessa. Questa operazione viene svolta dalla parte di programma riportata nel **Listato 5**. Ovviamente sarà necessario quanto prima connettere la scheda Maduino ad un alimentatore per la ricarica della batteria altrimenti finirà per danneggiarsi. Nella **Fig. 17** e nella **Fig. 18** potete

le righe di codice che configurano il modulo per l'accesso alla rete GPRS, si veda il **Listato 3**. Se il parametro DEBUG viene impostato su *true* avrete la possibilità di vedere su Serial Monitor la risposta del modulo ad ogni singolo

↓ Listato 5 (controllo alimentazione)

```
float VBAT= 2.0*3.3*analogRead(A1)/1024.0;
SerialUSB.println("VBAT= " + String(VBAT) + "V");
if (VBAT < batCutOff) //battery is empty POWER OFF
{
    SerialUSB.println("Battery LOW -> POWER OFF");
    sendData( "AT+CGNSPWR=0",1000,DEBUG); // POWER OFF GPS
    sendData("AT+HTTPTERM", 1000, DEBUG); // Terminate HTTP Server
    sendData("AT+CPWD=0", 1000, DEBUG); // power OFF SIM808
    while(true); // switch off/on for restart
}
```

↓ Listato 6 (Matlab analysis)

```
alert_subject = 'GPS Tracker battery LOW';
alert_body = 'Battery too low';
alert_api_key = 'ALERTAPIKEY';
alert_url= "https://api.thingspeak.com/alerts/send";
jsonmessage = sprintf(['{"subject": "%s", "body": "%s"}'], alert_subject,alert_body);
options = weboptions("HeaderFields", {'Thingspeak-Alerts-API-Key', alert_api_key; 'Content-Type','application/json'});
result = webwrite(alert_url, jsonmessage, options);
```


vedere le schermate di Thingspeak relative ad un piccolo traghetto in auto.

ALLARMI

Potrebbe essere utile ricevere delle notifiche nel caso si verificano determinati eventi durante il funzionamento del tracking. Visto che tutti i dati acquisiti vanno a finire nel cloud di Thingspeak vediamo ora come configurare la notifica degli eventi. Sicuramente avremmo bisogno di un avviso di batteria scarica, ma nulla vieta di creare avvisi anche per superamento limiti di velocità o anche solo per sapere se il veicolo è in movimento o fermo. Partendo dal tab Apps selezionate Matlab Analysis e create un nuovo elemento dal nome "GPS Tracker Send email Alert" ed inserendovi all'interno il **Listato 6**. Per completare il listato vi serve una Alerts API Key che potrete generare andando nella sezione My Profile disponibile cliccando sull'icona tutta a destra nella barra in alto.

In alto cliccate sul tab Apps e poi su React, si aprirà una finestra di configurazione che dovrete completare come visibile in **Fig. 19**. Il funzionamento è molto semplice, quando si verifica l'evento impostato (batteria sotto i 3,3V) viene inviata una mail di allerta (**Fig. 20**) all'indirizzo riportato durante la fase di regi-

```
COM22
Send
Start GPS
AT+CGSPWR=1
OK
AT+CGSSSEQ=PM0C
OK
GPS Status:AT+CGPSSTATUS?
*CGPSSTATUS: Location 3D Fix
OK
----> NEW DATA
Temperature= 17.00      humidity= 78.00
Read GPS data:
GPS status= 1
Data= 2020/10/08
Time= 07:52.05
latitude= 48.644917
longitude= 12.897722
MSL altitude= -8.400
Speed over ground= 0.28
Course over ground= 198.3
Write data to Thingspeak
AT+HTTPPARA="URL","http://api.thingspeak.com/update.json?api_key=EXDUVIZYL6Dl1MA4field1
OK
AT+HTTPACTION=0
OK
AT+HTTPREAD
OK
*HTTPACTION: 0,200,260
<
[ ] Scombinato automatico [ ] Visualizza orario Ritorno carrello (CR) 115200 baud Ripulisci l'output
```

strazione al portale. Tra le varie opzioni di Action in react è presente anche l'opzione ThingTweet che permette l'invio di un'allerta direttamente sul

Fig. 16
Serial Monitor per il listato SIM808_GSM_DHT11_GPS.ino



Fig. 17
Grafico di latitudine, longitudine e velocità per un breve tratto.

→ **Fig. 18**
Mappa relativa
ad un percorso
stradale.



→ **Fig. 19**
Configurazione
di un evento in
Thingspeak.

proprio account Tweeter, previa condivisione del vostro account Twitter con Thingspeak.

CONCLUSIONI

Si tratta di un progetto che necessita di un pochino di attenzione per portarlo a termine perché è necessario utilizzare più tecnologie tutte insieme, dall'uso di una scheda Arduino all'invio di dati tramite rete GPRS all'utilizzo di un portale per raccogliere i dati insomma, c'è proprio tutto, pertanto consigliamo a chi volesse cimentarsi nella realizzazione di affrontarlo per gradi, con una certa caparbieta nella risoluzione di eventuali problemi dovessero insorgere. □



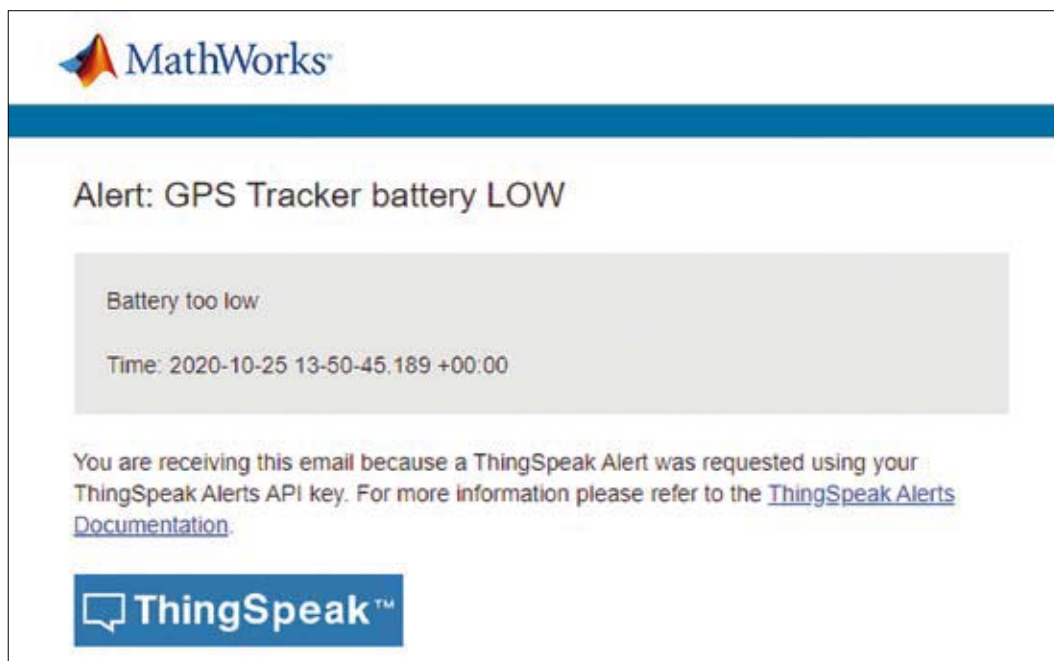
Cosa occorre?

La board **Maduino Zero** con **SIM808** (cod. **GPSMADUINO**) è disponibile presso **Futura Elettronica** a Euro 37,50, la batteria litio utilizzata nel progetto (cod. **PLIB1100-2MM**) è disponibile a Euro 7,90, il sensore di temperatura e umidità (cod. **DHT11**) è venduto a Euro 3,50
I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

→ **Fig. 20**
E-mail di allerta
inviata da
Thingsspak.



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

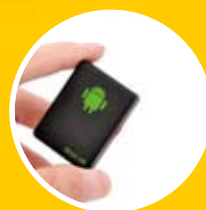
Tutto sotto controllo con il nuovo micro localizzatore GPRS/GSM. Per attivarlo basta inserire una SIM di qualsiasi operatore; inviando un SMS potrai attivare l'ascolto ambientale oppure ricevere un link con le coordinate che indicano la posizione in tempo reale del localizzatore. Dispone di batteria 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

LUCI PSICHEDELICHE A LED

di **CRISTIANO RUGGERI**

Centralina per luci psichedeliche
a quattro canali rivisitata
in chiave moderna con LED
colorati ad alta efficienza al
posto delle lampadine.

Q

ualche mese fa, quando ancora si ignorava che di lì a poco saremmo stati investiti dallo tsunami Covid19, sedevamo attorno ad un tavolino di un bar e ricordavamo di come in gioventù

festeggiassimo i nostri compleanni, capodanni e carnevali in compagnia, organizzando feste improbabili in locali altrettanto improbabili quali cantine ammuffite, soffitte polverose, garage (solo i più fortunati potevano avere un salotto o una taverna) messi a disposizione da qualcuno della "compagnia" col benessere di nonni o genitori. Ricordavamo i nostri impianti audio e luce, realizzati praticamente con niente, mettendo insieme in qualche modo i pezzi di impianto che si potevano recuperare tra i vari componenti della compagnia. In quegli anni, per generare un effetto luci semplice, ma per noi più che sufficiente, bastava acquistare una centralina di luci psichedeliche a 1, 2 o 3 canali più un po' di faretto. All'epoca (siamo a cavallo tra l'80 e il 95) vendevano dei kit piuttosto economici costituiti da una centralina a tre canali (toni bassi, medi e acuti) e da altrettanti faretto colorati.

Oggi, all'alba dei nostri 50 (talvolta anche più) anni, cosa ci rimane di quegli anni spensierati? E che fine avranno fatto i nostri "pezzi" di impianti luce e stereo? Ed ecco le solite tristi risposte: perso o rotto durante un trasloco, finito in discarica perché inutilizzato da troppi anni, o peggio... non ne abbiamo più alcun ricordo. Così, cavalcando un po' l'onda dei nostri ricordi,

abbiamo pensato di realizzare una centralina per luci psichedeliche che funzioni come quelle di allora, ma rivisitata con le tecnologie a LED attuali.

Se anche voi avete un po' di nostalgia, oppure semplicemente vi siete incuriositi perché più giovani di noi (amanti delle elettroniche di vecchia scuola) e non usi al concetto di luci psichedeliche, probabilmente troverete molto interessante proseguire la lettura di questo articolo.

SCHEMA DI PRINCIPIO

Prima di analizzare la nostra versione datata 2020 di centralina luci psichedeliche, è necessario vedere brevemente come funziona una centralina classica (anni '80 del secolo scorso) facendo riferimento alla Fig. 1 che mostra uno schema di principio.

L'effetto luci viene comandato dal segnale audio, che può essere prelevato direttamente da un microfono a bordo dell'apparato, oppure in parallelo ai morsetti rosso e nero di una cassa acustica. Il segnale audio quindi, a prescindere dalla sorgente, passa attraverso un primo stadio di ingresso che, con l'ausilio di un potenziometro, svolge la funzione di controllo di



Il colibrì è un driver per LED a 4 canali, che può pilotare diodi RGBW (Red, Green, Blu, White) agendo opportunamente sui quattro ingressi.

Eroga fino a 1A per canale e per il pilotaggio dell'uscita accetta sia un segnale PWM che una tensione analogica compresa tra 0,4 e 2,5V. Si alimenta con una tensione compresa tra 6 e 20V.

L'elevata tensione di alimentazione consente di pilotare più LED in serie; per esempio, con una tensione di 20V è possibile pilotare un massimo di $20V/3,4V = 5$ LED bianchi in serie per ogni canale.

Come è noto, i LED vanno pilotati a corrente costante e non a tensione costante, per ottenerne il massimo rendimento ed evitare di "bruciarli".

Per spiegare come mai ai fini della valutazione di un LED RGBW si debba prendere in considerazione la corrente diretta, prendiamo un LED quadricolor da 4 watt: potenza distribuita tra quattro LED che solitamente hanno una corrente di pilotaggio di 300mA (corrente diretta).

Siccome la caduta di tensione sui LED varia in base al colore, abbiamo potenze differenti per i vari LED.

In particolare:

$$P_{\text{rosso}} = (2,2 \div 2,5V) \cdot 300mA = 0,66 \div 0,75W$$

$$P_{\text{verde}} = (2,5 \div 2,8V) \cdot 300mA = 0,75 \div 0,84W$$

$$P_{\text{blu}}, P_{\text{bianco}} = (3,2 \div 3,4V) \cdot 300mA = 0,96 \div 1,02W$$

dove tra parentesi sono indicate le cadute di tensione minima e massima sui diodi. Considerando i valori massimi, abbiamo una potenza totale pari a:

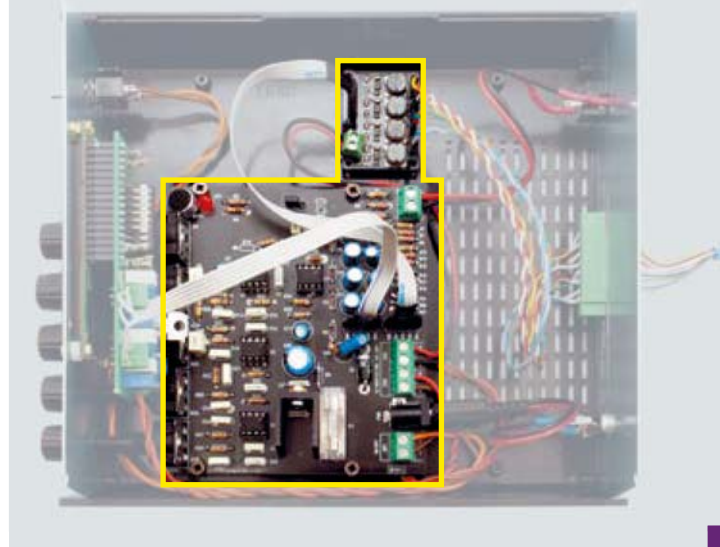
$$P_{\text{tot}} = 0,75 + 0,84 + 2,04 = 3,63W$$

quindi vicina a quanto dichiarato. Per i LED da 12W le cose cambiano un po', visto che la corrente di funzionamento dichiarata ammonta a 700mA; ora le potenze diventano 1,75W per il LED rosso, 1,96W per il LED verde e 2,38W per il bianco e blu. Con questi valori, la potenza massima complessiva diviene: $P_{\text{tot}} = 8,47W$. Questo valore è ben inferiore a quanto dichiarato, anche se rimane una potenza notevole. La potenza dichiarata per i LED RGBW è quindi da intendersi quella massima pilotando ciascun LED alla massima potenza, perciò con correnti di valore differenti.

livello principale (attenuazione) comune a tutti i canali, ma non obbligatoriamente presente in tutte le centraline. Una volta regolato per un'ampiezza adeguata, il segnale giunge contemporaneamente a uno o più filtri in bassa frequenza che di norma sono passa basso, passa banda e passa alto, sebbene alcune centraline avessero anche un quarto canale a banda piena o intera mentre altre avevano due soli canali, alti e bassi.

Ogni canale dispone di una propria regolazione di ampiezza così da poter dosare il segnale filtrato in uscita. A questo punto ogni uscita è convertita da alternata a continua tramite un rudimentale raddrizzatore quindi il segnale continuo così ottenuto viene usato per pilotare il gate di un tiristore (il più delle volte risultava essere un TRIAC ma non era raro trovare delle soluzioni con SCR). Regolando opportunamente i livelli di ogni canale, anche se in realtà si dovrebbe dire filtro, si può trovare la combinazione che determina il miglior effetto luci al ritmo della musica che si sta ascoltando. Il sistema è semplice ed efficace e non presenta grossi problemi quando usato col microfono in dotazione. Se collegato invece direttamente alle casse acustiche, occorre essere certi di disporre di centraline di livello superiore poiché le centraline molto economiche erano spesso sprovviste di disaccoppiamento galvanico tra sezione audio e sezione di potenza quindi, se non si prestava adeguata attenzione, poteva capitare di collegare la fase della tensione di rete al negativo delle casse, con conseguenze disastrose per l'impianto audio. In ogni caso, quando il cablaggio era corretto, il neutro della tensione di rete risultava collegato al negativo delle casse acustiche, anche se erano comunque presenti delle protezioni di qualche tipo. Ovviamente, ai giorni nostri, esistono i sistemi di illuminazione a led molto efficienti, meno dispendiosi in termini di potenza elettrica e disponibili in forme e colori decisamente soddisfacenti.

Di contro i led devono essere comandati con un segnale PWM (Pulse Width Modulation, ossia modulazione a larghezza di impulso: è un'onda quadra a frequenza costante e duty-cycle variabile) invece che continuo, e devono anche essere limitati in corrente pena la rottura irrimediabile. Pertanto il nostro progetto è partito "un po' al contrario" ossia dalla ricerca del pilotaggio dei LED sulla base di diverse ipotesi e circuiti di complessità differente. Dopo alcune ricerche e valutazioni, abbiamo deciso di utilizzare un apparato già comparso nelle pagine della rivista ossia il driver di potenza per LED RGBW noto come Colibrì (acquistabile presso



la Futura Elettronica). Ad esso abbiamo dedicato un riquadro specifico, pertanto in queste righe ci limiteremo alle caratteristiche principali per cui lo abbiamo scelto.

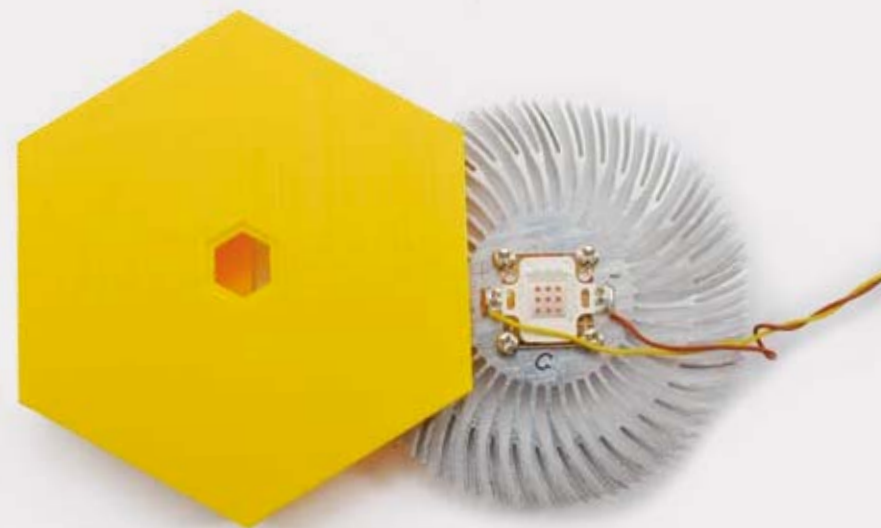
Innanzitutto si tratta di un apparato in grado di gestire quattro differenti led di potenza, comandati con quattro ingressi di controllo distinti; inoltre integra tutti i circuiti necessari per pilotare da solo in PWM i LED mantenendone controllata la corrente. Ha un range di alimentazione piuttosto ampio che arriva fino a 20 V con 1 A disponibile per ogni canale led (80 W in totale). Ma la caratteristica che lo ha reso più appetibile è il modo in cui i quattro ingressi possono essere comandati: ogni ingresso, infatti, può accettare sia un segnale PWM da 0 a 100% per il massimo della potenza in uscita, sia un segnale analogico compreso tra 0,4 e 2,5 V.

A questo punto è sufficiente pilotare gli ingressi con dei segnali in continua derivati da dei filtri di banda audio, sovrapposti ad un offset di poco meno di 0,4 V e da solo il colibrì può rimpiazzare i quattro circuiti di uscita a tensione di rete di **Fig. 1**.

SCHEMA ELETTRICO

Vediamo ora lo schema elettrico della sezione di pilotaggio e filtri del nostro progetto, partendo dal punto di entrata del segnale audio.

Tale segnale può essere prelevato direttamente dalla capsula microfonica MIC1 (polarizzata da R4) sia da un cavetto proveniente direttamente da una cassa acustica e collegato al morsetto CN1, associato al



partitore resistivo R1/R3/R5 (tutte resistenze da 220 ohm) che limita l'ampiezza del segnale in ingresso e lo rende idoneo al pilotaggio dell'elettronica già con potenze nell'intorno di 1 watt audio. Ovviamente, aumentando il valore di R1, è possibile entrare con potenze superiori a discrezione vostra.

A questo punto è possibile selezionare quale delle due fonti giungerà al circuito attenuatore di ingresso (l'amplificatore operazionale U2A, LM358) tramite il ponticello J1, posto direttamente sul circuito stampato. A prescindere dalla sorgente prescelta, il segnale passa attraverso la rete costituita da C1, R10 e R13, il cui scopo è disaccoppiare il segnale dalla continua e riferirlo a massa virtuale ossia 5 V circa. Questa operazione è necessaria poiché il circuito è alimenta-

Ogni LED è applicato a un dissipatore di calore e montato in un contenitore stampato in 3D che ne farà un faretto.

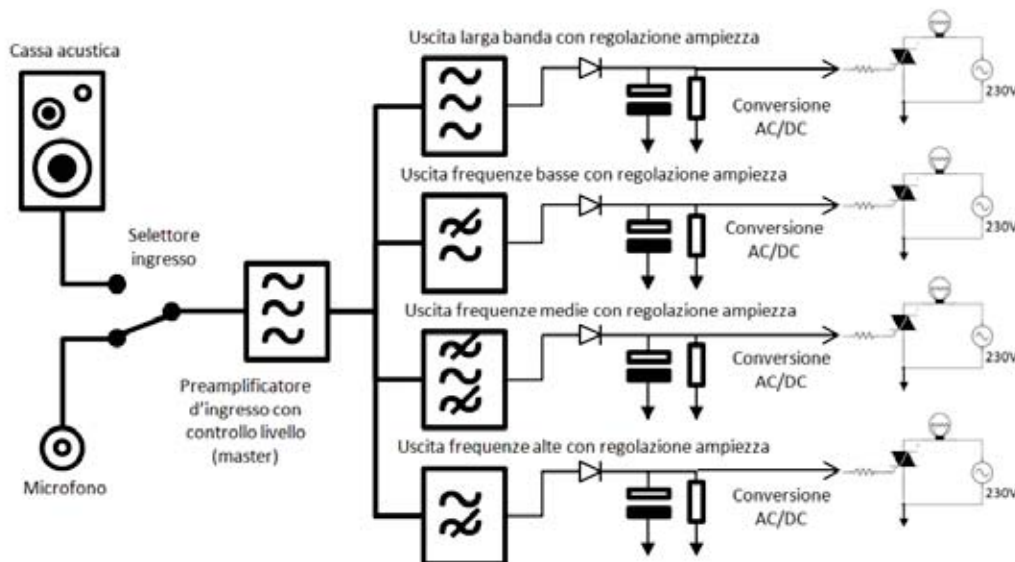
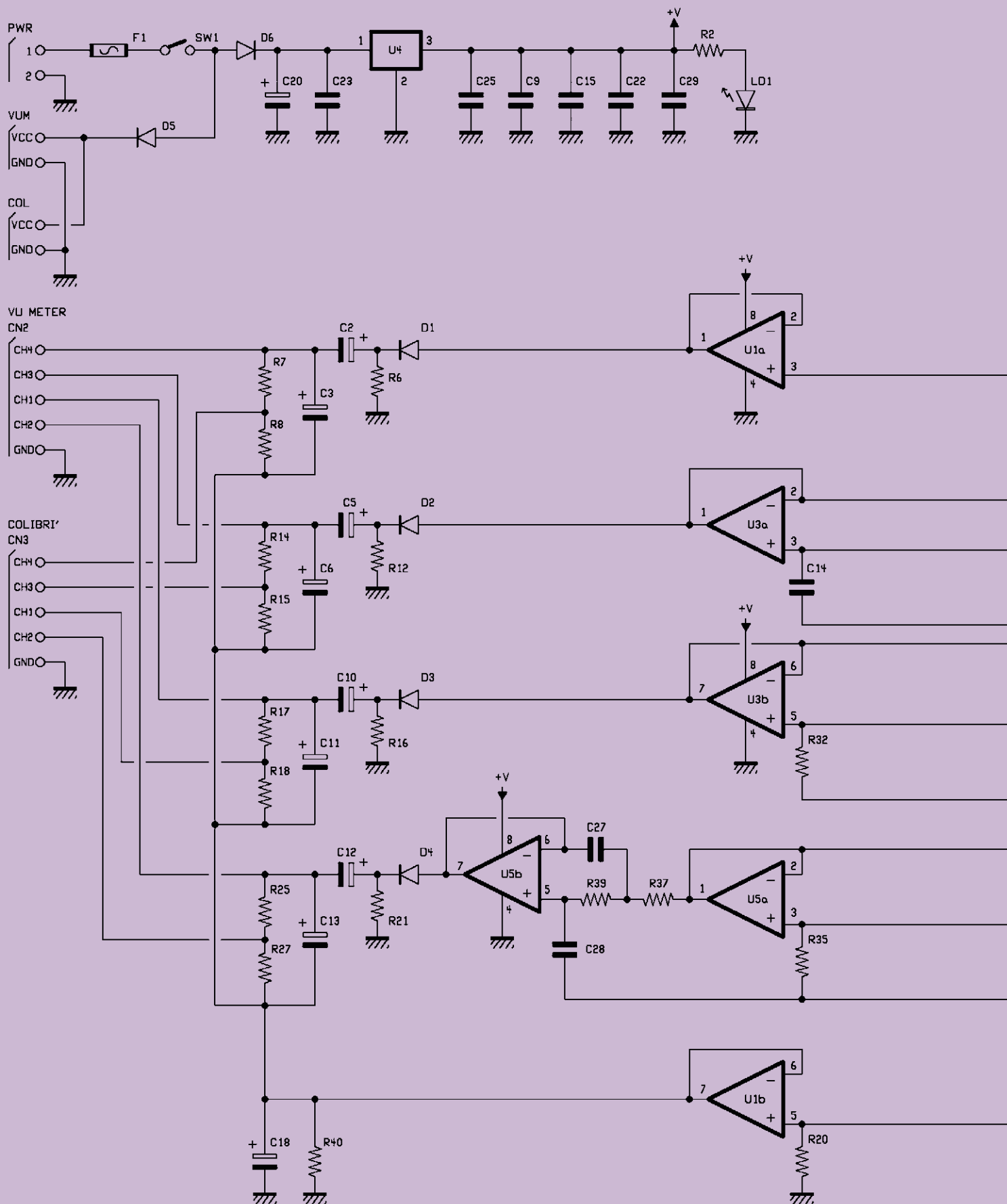
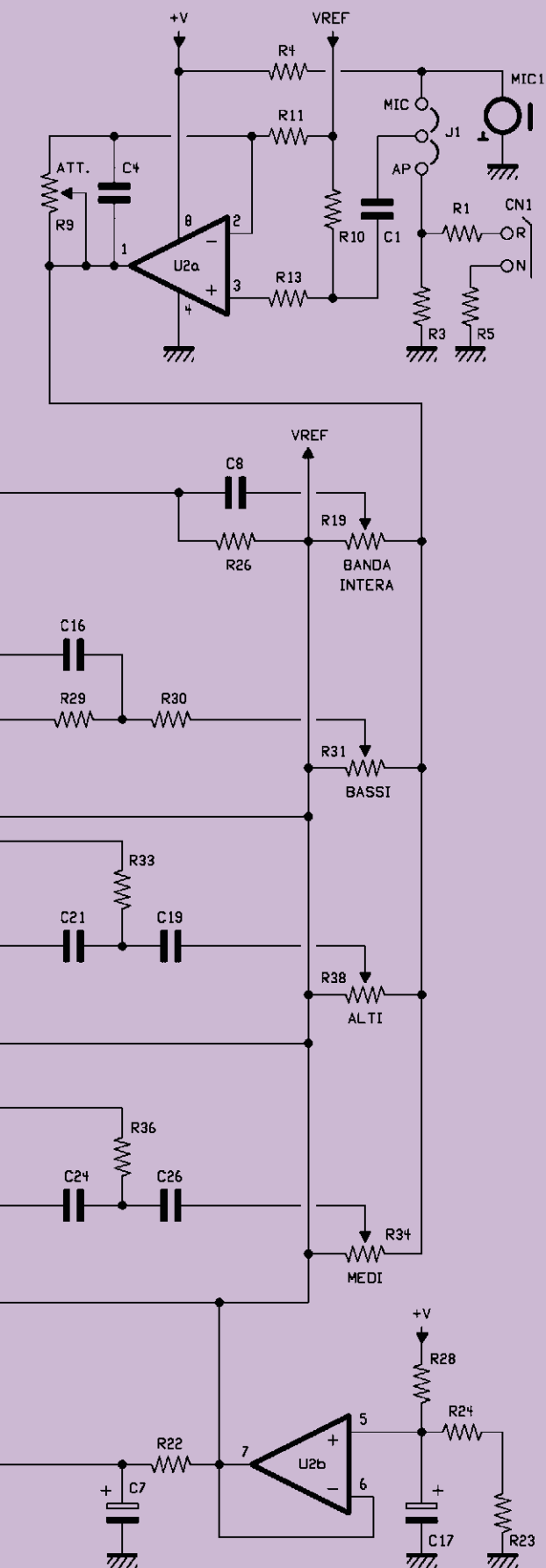


Fig. 1 Schema di principio di una centralina luci psichedeliche analogica a tensione di rete.





to a tensione singola e gli amplificatori operazionali usati necessitano di una massa virtuale a circa metà tensione di alimentazione. L'operazionale U2A dispone di un potenziometro (R9) con cui è possibile regolare l'attenuazione del segnale di ingresso: infatti insieme alla resistenza R11 definisce un guadagno da un minimo di 1 a un massimo di 100: ciò garantisce ampio margine di manovra sia con segnali molto deboli captati dal microfono che con segnali molto forti prelevati direttamente dalle casse acustiche. C4, di capacità bassissima, serve ad evitare auto-oscillazioni dell'operazionale con guadagno molto alto. L'uscita di U2A giunge contemporaneamente a un capo di quattro potenziometri (R19/R31/R38/R34) mentre l'altro capo dei potenziometri è collegato alla massa virtuale di 5 V, generata dall'operazionale U2B insieme alla rete passiva C17/R23/R24/R28. I quattro cursori dei potenziometri vanno agli ingressi dei filtri di banda. R19 regola il livello del filtro a banda intera, rappresentato da U1A (LM358). R31 regola il livello del segnale in entrata al filtro passa basso rappresentato da U3A. Analogamente R38 e R34 giungono al filtro passa alto (U3B, alti) e passa banda (U5 A e B, medi).

Tutti i filtri impiegati sono realizzati secondo il circuito Sallen Key, al quale abbiamo dedicato un riquadro a parte, e sono tutti riferiti a massa virtuale. Il filtro passa basso utilizza un solo operazionale e ha una frequenza di taglio di circa 190 Hz. Analogamente il filtro passa alto è realizzato con un singolo operazionale, calcolato per la frequenza di taglio di 1900 Hz circa; il filtro passabanda, invece, altri non è che una serie "passa alto – passa basso" tagliati a 190 e 1.900 Hz, come i precedenti.

In questo modo abbiamo ottenuto:

- un'uscita a banda larga, a cui giunge tutto lo spettro audio in ingresso;
- un'uscita frequenze basse che taglia gradualmente tutte le frequenze oltre i 190 Hz;
- un'uscita frequenze alte che lavora a partire da 1.900 Hz;
- un'uscita per frequenze medie, che lavora solo tra 190 e 1.900 Hz.

Questa suddivisione, per quanto eseguita con un filtraggio piuttosto blando, ci consente di avere un effetto luci centrato sulla ritmica, tipicamente a bassa frequenza, un effetto legato quasi completamente alla voce maschile e femminile, uno legato agli strumenti acuti, ed uno sempre attivo con qualsiasi frequenza.

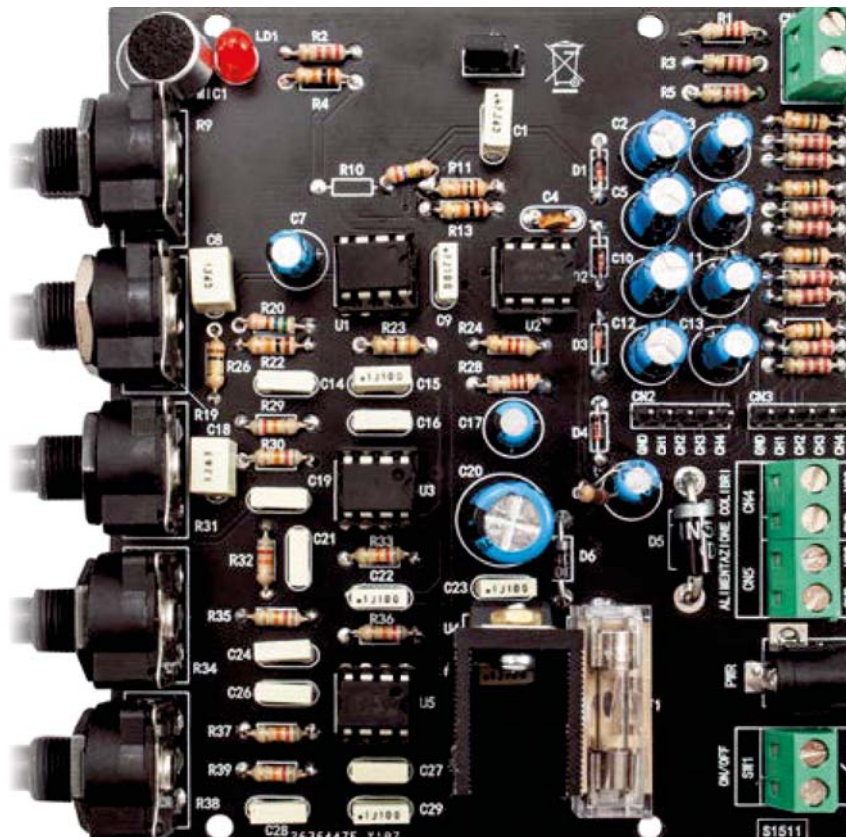
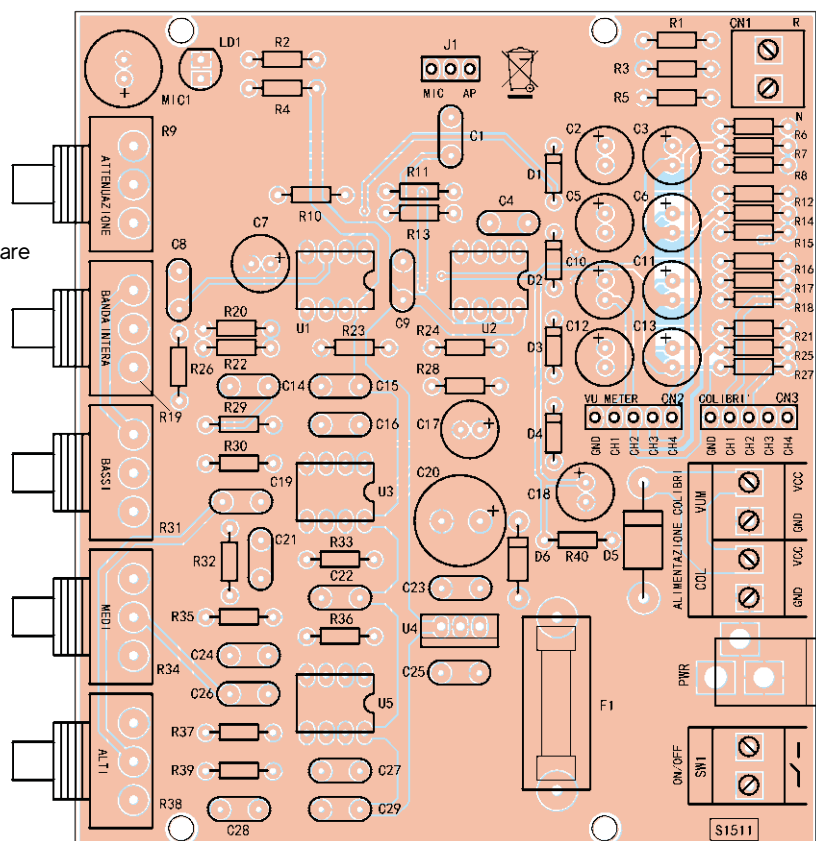
Le uscite dei quattro filtri giungono ognuna ad un circuito di conversione da alternata a continua costi-

Elenco Componenti:

R1, R3, R5: 220 ohm
 R2, R7, R8: 2,2 kohm
 R14, R15, R17, R18, R25, R27: 2,2 kohm
 R4, R13, R22: 10 kohm
 R6, R12, R16, R21: 15 kohm
 R9, R19: 100 kohm Potenziometro lineare
 R31, R34, R38: 100 kohm Potenziometro lineare
 R10: 270 kohm
 R11: 1 kohm
 R20: 560 ohm
 R23: 390 ohm
 R24: 8,2 kohm
 R26: 100 kohm
 R28: 12 kohm
 R29, R30, R32, R33: 82 kohm
 R35, R36, R37, R39: 82 kohm
 R40: 100 ohm
 C1: 470 nF 63 VL poliestere P5
 C2, C5, C10, C12: 47 μ F 35 VL elettrolitico
 C3, C6, C7: 1 μ F 35 VL elettrolitico
 C11, C13, C17: 1 μ F 35 VL elettrolitico
 C4: 12 $\pm$ 22 pF ceramico
 C8: 1 μ F poliestere P5
 C9, C15, C22, C23, C25, C29:
 100 nF 63 VL poliestere P5
 C14, C16, C24, C26: 10 nF poliestere P5
 C18: 10 μ F 25 VL elettrolitico
 C19, C22, C27, C28: 1 nF poliestere P5
 C20: 470 μ F 35 VL elettrolitico
 D1, D2, D3, D4: 1N4148
 D5: 1N5408
 D6: 1N4007
 U1, U2, U3, U5: LM358
 U4: UA7812
 LD1: LED 5 mm rosso
 MIC1: Capsula microfonica
 preamplificata
 F1: Fusibile 2 A rapido
 SW1: Interruttore da pannello
 COL: Morsettiera 2 Poli P 5,08 mm
 VUM: Morsettiera 2 Poli P 5,08 mm
 CN1: Morsettiera 2 Poli P 5,08 mm
 SW1: Morsettiera 2 Poli P 5,08 mm
 CN2, CN3: Strip maschio 5 poli
 J1: Strip maschio 3 poli
 PWR: Plug di alimentazione
 5,5x2,1 mm

Varie:

- Zoccolo 4+4
- Portafusibile 5x20 da c.s.
- Jumper
- Manopole per potenziometri
- Circuito stampato S1511
(111x99 mm)



tuito da un diodo, tre resistenze e due condensatori. I quattro circuiti raddrizzatori di conversione sono identici e svolgono la semplice funzione di trasformare il segnale sinusoidale proveniente dai circuiti analogici in un segnale pseudo-continuo di ampiezza adatta per pilotare gli ingressi del driver RGBW. Vediamone uno solo (quello del banda larga) a titolo didattico.

Il segnale audio, simmetrico riferito a 5 V di offset (massa virtuale) giunge dal pin 1 di U1A al diodo D1 che ne lascia passare la sola parte positiva. Questa, attraverso il condensatore C2 da 47 μ F, caricato dalla semionda positiva e scaricato durante la semionda negativa tramite R6 da 15kohm, carica e mantiene carico il condensatore C3 da 1 μ F. Durante la parte negativa della semionda, C3 si scarica sul partitore resistivo R7/R8 (entrambe da 2k2 ohm), i cui nodi alto e centrale vanno ai connettori CN2 e CN3. Da CN2 è possibile prelevare un segnale compreso tra 0,3 e 5 V massimi circa, con cui verrà pilotato un VU Meter che vedremo in un altro articolo, mentre da CN3 si piloteranno gli ingressi del colibrì, con un segnale di ampiezza compresa tra 0,3 e circa 2,5 V massimi.

Notate che i quattro raddrizzatori sono tutti riferiti a 0,3 V circa invece che massa, segnale questo generato dall'operazionale U1B opportunamente polarizzato in ingresso (R22/R20/C7) e conforme alle specifiche del driver colibrì. In assenza di segnale in ingresso, le uscite dei raddrizzatori sono tutte al valore di 0,3V circa, valore questo che mantiene il driver colibrì in stato di riposo. Con l'aumentare del segnale in ingresso tali uscite possono raggiungere un valore massimo di circa 2,5 V, come richiesto dal driver. Completa la centralina un alimentatore stabilizzato basato sul classico 7812, corredato dei pochi componenti che necessita. Tutto il circuito può essere alimentato con una tensione compresa tra 15 e 20 V; allo scopo abbiamo previsto un plug da circuito stampato seguito da un interruttore generale che a sua volta giunge a un fusibile da 2 A rapido. Quest'ultimo alimenta la sezione analogica e la sezione del driver di potenza tramite due diodi di dimensioni adeguate, D5 e D6.

REALIZZAZIONE PRATICA

Come sempre, occorre realizzare il circuito stampato a doppia faccia: potete scaricare il master dal sito della rivista e realizzarlo, con la tecnica che preferite. Tenete però presente che è un circuito piuttosto complesso, per cui la realizzazione artigianale è destinata prettamente solo a quanti siano in possesso della dovuta esperienza.

Guarda le Luci Psichedeliche a LED in azione su YouTube!



Vuoi vedere in azione le nostre Luci Psichedeliche a LED? Nulla di più semplice: collegati alla pagina <https://youtu.be/peewa59CfT8>. Potrai anche vedere i video di moltissimi altri nostri progetti proposti in passato.

È altresì possibile ottenere dei circuiti stampati già pronti attraverso il servizio di realizzazione PCB (cod. PCBPRODUCTION) che trovate sul sito www.futurashop.it.

Ora che siete in possesso del PCB, potete procedere col montaggio, avendo cura di partire prima con i componenti più piccoli e bassi, quali zoccoli, resistenze, diodi e condensatori, lasciando via via per ultimi i componenti di dimensioni maggiori, come i potenziometri.

A montaggio ultimato occorre verificare le tre tensioni di sistema ossia i 12Vcc di alimentazione, e i due riferimenti a 5 e 0,3 V. Alimentate il sistema con un alimentatore da laboratorio stabilizzato, impostato a 18V - 100 mA.

Accertatevi che il fusibile sia inserito sul suo portafusibile e che l'interruttore SW1 sia chiuso: il led rosso LD1 si deve accendere immediatamente. Tramite un tester impostato per misurare 20 V fondo scala, verificate quanto segue:

- circa 17,5 V tra + e - dei connettori COL e VUM;
- circa 12 V tra aletta regolatore U4 e il pin 8 di ognuno degli zoccoli degli integrati;
- circa 5 V tra aletta regolatore U4 e pin 5 dello zoccolo U2.

Spegnete ora il circuito e montate gli integrati sui rispettivi zoccoli quindi verificate che ci siano:

- circa 5 V tra aletta regolatore U4 e i pin U1/1, U3/1, U3/7, U5/1, U5/7, U2/7;
- circa 0,26V tra aletta regolatore U4 e U1/pin7, CN2/pin1-2-3-4, CN3/pin1-2-3-4.

IL FILTRO SALLEN KEY

È una soluzione circuitale molto semplice, introdotta da R.P. Sallen e E. L. Key del laboratorio Lincoln del MIT nel 1955, con la quale è possibile creare dei filtri passa basso e passa alto aggiungendo solo 4 componenti passivi ad un amplificatore operazionale. Nella configurazione minima, l'amplificatore operazionale viene utilizzato come inseguitore di tensione, quindi il sistema complessivo ha guadagno unitario in banda e va poi ad attenuare sempre più quanto più ci si allontana dalla frequenza di taglio impostata. Quest'ultima è data dalla formula:

$$F_c = 1 / \{2\pi * [\sqrt{(R1 * R2 * C1 * C2)}]\} \text{ per entrambi i circuiti}$$

Mentre il fattore Q è dato dalla formula:

$$Q = [\sqrt{(R1 * R2 * C1 * C2)}] / [C2 * (R1 + R2)] \text{ nel passa basso}$$

$$Q = (R2 * Cx) / [\sqrt{(R1 * R2 * C1 * C2)}] \text{ nel passa alto}$$

$$\text{Dove } Cx = (C1 * C2) / (C1 + C2)$$

Si possono ottenere filtri passa banda mettendo in cascata un passa alto calcolato per la frequenza di taglio inferiore e un passa basso dimensionato per la frequenza di taglio superiore.

Prendiamo ad esempio i valori utilizzati nel nostro filtro passa basso delle frequenze basse, dove abbiamo scelto 82kohm per le resistenze e 10 nF per i condensatori, quindi riscriviamo le formule:

$$F_c = 1 / \{2\pi * [\sqrt{(82000 * 82000 * 0,00000001 * 0,00000001)}]\} = 194 \text{ Hz}$$

$$Q = [\sqrt{(82000 * 82000 * 0,00000001 * 0,00000001)}] / [0,00000001 * (82000 + 82000)] = 0,00082 / 0,00164 = 0,5$$

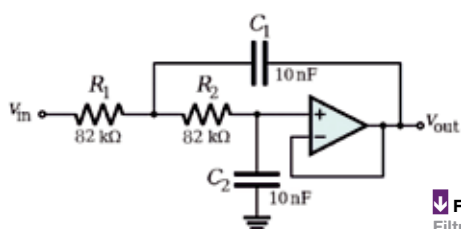
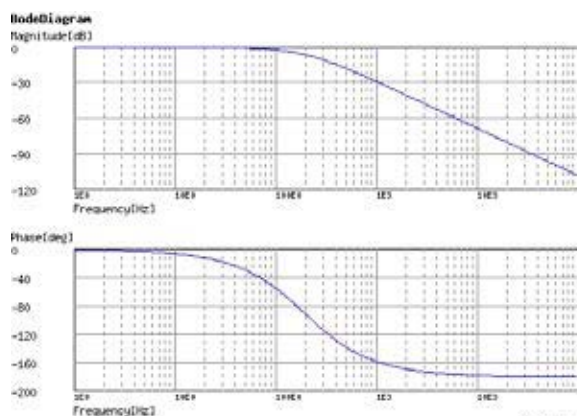


Fig. A
Filtro Sallen
passa basso.



(c)okawa-denritchi .jap

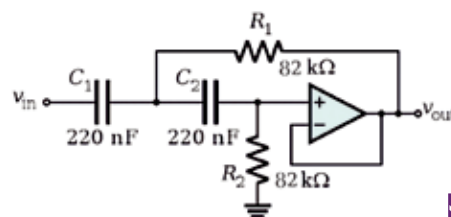
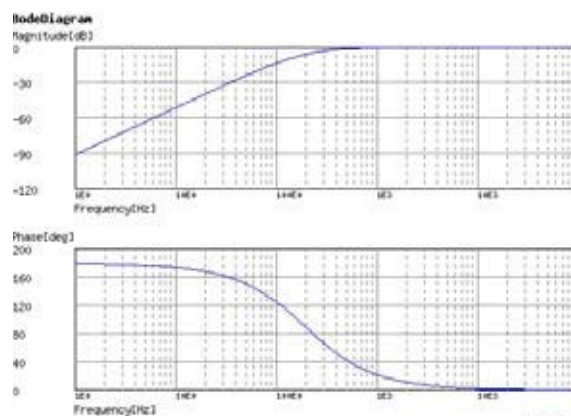


Fig. B
Filtro Sallen
passa alto.



(c)okawa-denritchi .jap

A questo punto potete spegnere nuovamente il circuito, collegare il Colibrì al circuito, alimentandolo dal connettore CN4 e pilotandolo dal connettore CN3 (fate attenzione alla polarità e alla corretta disposizione degli ingressi di pilotaggio) e collegare alle uscite del colibrì i LED di potenza che avete

scelto di utilizzare.

Vista la versatilità di questo driver, non è possibile spiegare tutte le possibili connessioni con i singoli LED, poiché in commercio ne esistono parecchi e non possiamo sapere che tipo di diodi avrete adottato. Noi per comodità e per impatto visivo,

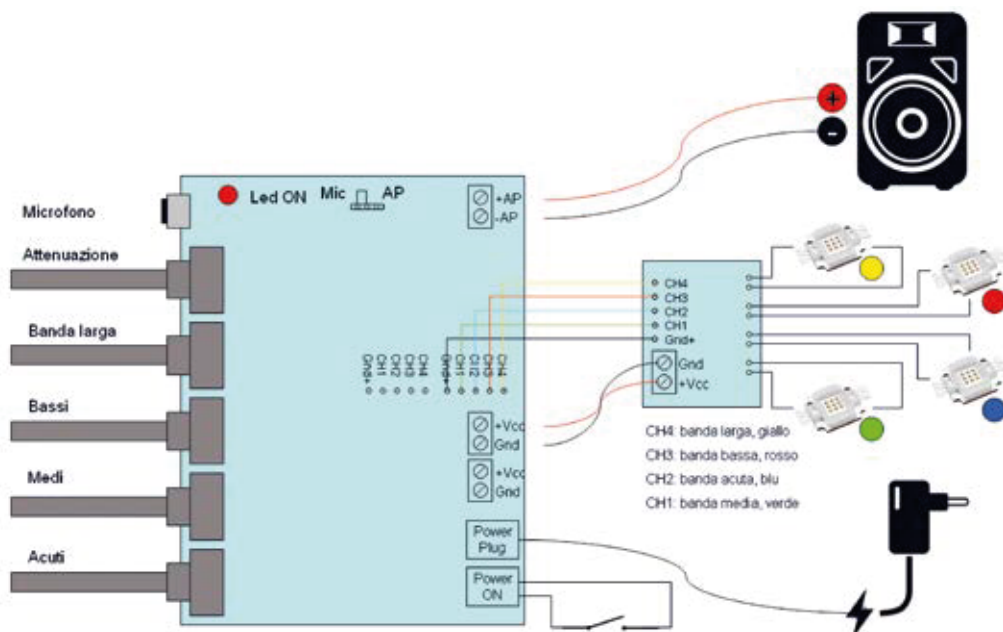


Fig. 2
Connessioni della scheda, alla fonte BF, all'alimentatore, al driver colibri e da questo ai power LED.

abbiamo scelto i led della serie 2846-LED10Wxxxx, da 10 W, disponibili in varie colorazioni ed acquistabili presso www.futurashop.it insieme ad un dissipatore adeguato (2846-ROUNDISS). Per il cablaggio, occorre fare riferimento alla **Fig. 2**. Fatto anche questo, gettate un'ultima occhiata per essere certi dei collegamenti e siete quindi pronti per il primo utilizzo delle luci psichedeliche.

COLLAUDO E UTILIZZO

Come prima prova, utilizzeremo la centralina in modo autonomo, ossia il suono verrà captato dalla capsula microfonica di cui dispone. Impostate correttamente il jumper J1, portate il potenziometro dell'attenuazione (R9) al massimo, quello della banda larga (R19) quasi al massimo e tutti gli altri potenziometri al minimo. Fate suonare qualcosa a vostro piacimento, quindi accendete la centralina: ruotate lentamente il potenziometro R9 attenuazione in senso antiorario, in modo da ridurre l'attenuazione fino a che i LED collegati al canale banda intera non inizieranno a lampeggiare; a questo punto diminuite leggermente il livello della banda intera ruotando R19 a sinistra e compensate con un leggero ritocco di R9. Ora che il livello di partenza è impostato, potete regolare gli altri potenziometri (bassi medi e alti) in modo da far accendere i led relativi.

Verificato che la sezione microfonica funziona, potreste provare con il collegamento diretto (effettuato sempre a macchine spente!) in parallelo ad un altoparlante tramite un doppino rosso-nero

anche piccolo (0,25 mmq è più che sufficiente) rispettando la polarità sull'altoparlante e sul connettore CN1. Dalle nostre prove, con un amplificatore dotato di misuratori di livello e collegato a casse acustiche di dimensioni ed efficienza importanti, abbiamo rilevato che la centralina funziona correttamente con una potenza audio di picco misurata tra 0,05 e 0,5 W: tale potenza è più che adeguata per un ascolto domestico anche allegro, però ovviamente i nostri indicatori di livello potrebbero differire leggermente da altri in altri amplificatori di altre marche. Se vi va, potete inserire tutto il sistema in un contenitore adeguato: noi abbiamo pensato di utilizzare il Teko AUS22, per il quale sono previsti i fori di fissaggio corretti per il passo dei fori di ancoraggio meccanico del circuito stampato, ma potete optare per ciò che preferite. Non ci resta che augurarvi buon divertimento con la vostra centralina luci psiche a LED. ■



Cosa occorre?

I componenti principali utilizzati in questo progetto sono disponibili presso Futura Elettronica. Il driver LED (cod. COLIBRI) è in vendita a Euro 19,90, i led di vari colori da 10W (cod. LED10Wxxx) sono disponibili a Euro 3,80 caduno. Il circuito stampato può essere realizzato utilizzando il servizio di prototipazione dei PCB (cod. PCBPRODUCTION). I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

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

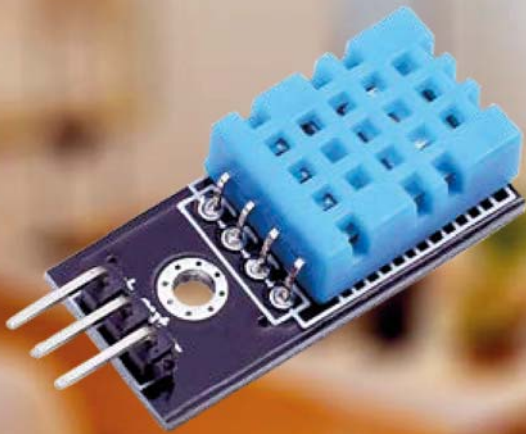
€20,00

Cod. MW10012



ARDUINO CHIAMA SMARTPHONE

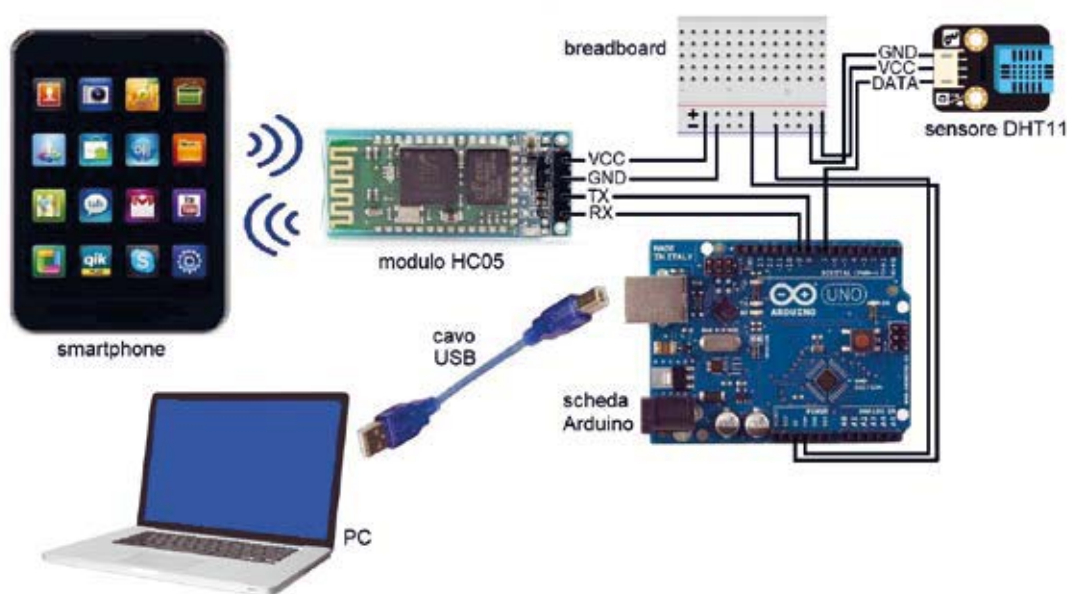
di STEFANO GARAVAGLIA



Interfacciamo via Bluetooth Arduino con uno smartphone per effettuare acquisizione e visualizzazione dati di temperatura e umidità.

I molti articoli presentati sulle nostre pagine vi avranno ormai chiarito come sia facile, oggi, rilevare parametri ambientali e trasferirli sul web, grazie ad hardware dedicati e a basso costo, ma anche a servizi on-line che supportano la pubblicazione e l'elaborazione dei dati. Tra i vari hardware ormai vi sarete abituati a vedere le schede basate sul chip ESP8266, che è un completo transceiver WiFi con il quale, passando da un hot-spot WiFi, si può andare in Internet e inviare i dati desiderati. Probabilmente un po' meno, tra voi, immagineranno che cose del genere possono essere svolte tranquillamente anche con hardware davvero essenziali, come dimostreremo, in questo articolo, con un esempio applicativo dove andremo a effettuare l'acquisizione di due parametri ambientali e la trasmissione dei relativi dati tra una scheda Arduino e uno smartphone. In questo caso, però, pur avvalendoci di un link

Fig. 1
Schema di principio
della nostra
applicazione.



wireless non ci appoggeremo ad alcun servizio sul web, ma stabiliremo una connessione diretta con uno smartphone, sfruttando il protocollo Bluetooth implementato da uno specifico modulo. Il software che gestisce questa applicazione deve, in particolare, consentire ad un utente:

- di richiedere, attraverso il suo smartphone, i dati di temperatura o umidità ambientale che la scheda Arduino acquisisce periodicamente attraverso un sensore;
- di vedere i dati richiesti visualizzati sullo schermo dello smartphone.

Il trasferimento di dati tra una scheda Arduino e uno smartphone può avvenire nelle due direzioni (smartphone-scheda e scheda-smartphone) grazie alla presenza di un modulo Bluetooth che, interposto tra di essi, trasforma una porta seriale in una connessione wireless Bluetooth, consentendo in pratica un controllo da remoto attraverso lo smartphone.

Con il termine Bluetooth si è soliti definire una tecnologia di scambio dati senza fili (via radio, tipicamente intorno ai 2,4 GHz di frequenza) utilizzata per il trasferimento di dati da punto a punto senza fili su breve distanza, che risulta particolarmente adatta per l'impiego con dispositivi mobili come smartphone e tablet. Peraltro con il link Bluetooth è possibile realizzare, oltre che comunicazioni punto-punto, anche reti WPAN (Personal Area Network in wireless).

Rispetto ad altre tecnologie di trasferimento dati come USB, LAN o WLAN, il Bluetooth è caratte-

rizzato da velocità di trasmissione inferiori che lo rendono poco adatto per l'invio di pacchetti dati di notevoli dimensioni, ma ideale per applicazioni come la nostra. Peraltro le recenti versioni di Bluetooth, dalla 4.2 in poi, rendono possibili connessioni a basso consumo, sicuramente inferiori a quello del WiFi.

Il trasferimento di dati tramite tecnologia Bluetooth viene reso possibile da un software specifico di controllo del trasferimento dati e da un dispositivo hardware chiamato transceiver, dotato quindi di unità trasmittente e unità ricevente, che rappresenta il livello fisico dello stack Bluetooth. Una connessione può essere avviata da un qualunque dispositivo che assume il ruolo di master rispetto a uno slave creando una rete piconet; lo slave rimane attivo fino a quando il master non disattiva la funzione Bluetooth; la connessione di due o più dispositivi tramite Bluetooth viene denominata pairing.

Lo schema circuitale (di cablaggio, se preferite) utilizzato per la realizzazione di questa semplice ma interessante esperienza viene rappresentato in **Fig. 1**. La scheda Arduino acquisisce i dati di temperatura e umidità ambientale utilizzando un sensore DHT11; la comunicazione tra scheda e smartphone nelle due direzioni è affidata al modulo bluetooth HC05.

Come si può notare i componenti necessari per la realizzazione di questa esperienza sono sostanzialmente pochi e tutti acquistabili direttamente sul sito di Futura Elettronica (www.futurashop.it).

Di seguito riportiamo le specifiche dei prodotti interessati unitamente al codice:

- Arduino Uno REV3 (microcontrollore Atmega328) codice 7300-ARDUINOUNOREV3;
- Modulo Bluetooth Transceiver con Pin Strip HC05 (2846-HC05STRIP)
- scheda con sensore di temperatura e umidità DHT11 (2850-TEMPHUM);
- Breadboard 400 contatti con base autoadesiva dimensioni 82x55 mm (7300-BB400).

Serve poi, per realizzare le connessioni elettriche, una confezione con 65 jumper maschio-maschio 7300-JUMPER65.

IL MODULO BLUETOOTH

Andiamo ora a fare una panoramica sui dispositivi utilizzati nel progetto, partendo dal Modulo bluetooth HC05: viene utilizzato quando si vuole far comunicare un microprocessore (e di conseguenza anche una scheda Arduino) con dispositivi forniti di connessione Bluetooth (Personal Computer, smartphone, tablet). Il modulo è basato su chipset CSR BlueCore4 e dispone di 8 Mbit di memoria Flash per l'archiviazione di software; supporta due modi di funzionamento: tramite comandi AT e in modalità trasparente. Nella modalità trasparente può essere configurato come Master, Slave o Loopback, e può connettersi ad altri dispositivi che supportano il protocollo SPP. Nella modalità di comando AT, si può configurare il modulo e inviare i comandi. In ogni momento si può scegliere tra la modalità di configurazione via comandi AT, oppure l'invio trasparente dei dati attraverso il pin EN. I quattro contatti evidenziati nella **Fig. 1** vengono rispettivamente utilizzati per l'alimentazione (VCC e GND) e per la comunicazione seriale con Arduino (RX per la ricezione, TX per la trasmissione). Il pin VCC deve essere collegato ai 5 V della scheda, il pin GND al corrispondente della scheda. Il pin di ricezione RX del modulo Bluetooth deve essere connesso al pin di trasmissione TX della scheda Arduino (viene di seguito scelto il pin 8); viceversa il pin di trasmissione TX deve essere collegato al pin di ricezione RX della scheda (viene di seguito scelto il pin 9).

SCHEDA CON SENSORE DHT11

Viene utilizzata per effettuare misure di temperatura e umidità ambientale fornendo un segnale digitale che rappresenta i valori assunti da queste due grandezze fisiche. La scheda in questione è

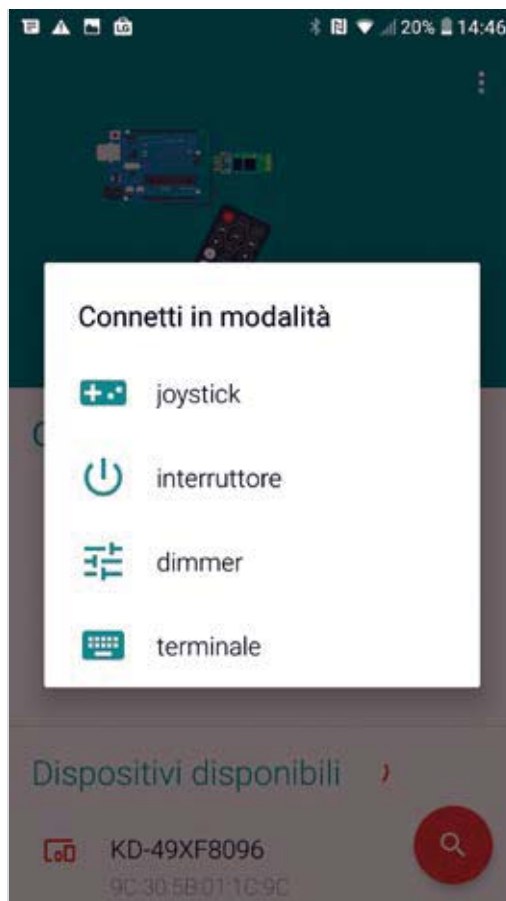
una breakout board che ospita il popolare sensore DHT11, il quale contiene due elementi resistivi, uno sensibile all'umidità e l'altro (un NTC) alla temperatura. Il DHT11 è un sensore di temperatura (da 0°C a +50°C) e umidità relativa (da 20% a 90%) calibrato in modo estremamente preciso in fase di costruzione, i cui coefficienti di calibrazione sono memorizzati nella memoria OTP del microcontrollore interno e vengono richiamati durante il processo di rilevamento. In questo modo non è necessario ricalibrare il sensore prima di ogni utilizzo. La tensione generata dai due trasduttori viene convertita in formato digitale da un convertitore A/D a 8 bit e resa disponibile lungo una connessione seriale one-wire. La risoluzione della misura è di 1°C sulla temperatura (tolleranza $\pm 2^\circ\text{C}$) e dell'1% sull'umidità relativa (tolleranza $\pm 5\%$).

La scheda sulla quale viene montato il sensore dispone di tre pin (VCC per la tensione di alimentazione, GND per la massa e DATA per la trasmissione del segnale); il pin DATA viene di seguito collegato al pin 7 della scheda Arduino, come si vede nella **Fig. 1**.



Fig. 2
Connessione del
dispositivo.

➔ **Fig. 3**
Modalità di
connessione
Bluetooth.



⬆ **Fig. 4**
Dati rilevati dal sensore.

Il componente può essere alimentato con tensioni comprese fra 3 V e 5,5 V; la corrente assorbita vale circa 1,5 mA se sta misurando il dato di temperatura o umidità, circa 120 μ A se è in standby.

PROGRAMMAZIONE

I dati forniti dal sensore possono essere visualizzati sullo schermo dello smartphone solo se è

stata scaricata sul medesimo un'applicazione open source che consente di inviare delle stringhe via Bluetooth. Facendo riferimento a dispositivi che utilizzano il sistema operativo Android si può ad

➔ **Fig. 5**
Visualizzazione
dei valori letti.



Listato 1

```
#include<SoftwareSerial.h> //si include la libreria SoftwareSerial
#include<dht11.h> //si include la libreria dht11.h
char dato; //dato=carattere digitato sulla tastiera dello smartphone
int u,t; //u=umidità,t=temperatura
dht11 DHT; //viene definito l'oggetto DHT
SoftwareSerial seriale(8,9); //viene definito l'oggetto denominato seriale
// pin 8=RX deve essere collegato al pin TX del modulo
// pin 9=TX deve essere collegato al pin RX del modulo
void setup(){
  serial.begin(9600); //velocità di comunicazione seriale
}
void loop(){
  DHT.read(7); //i dati di umidità e temperatura vengono acquisiti dal pin 7
  u=DHT.humidity;
  t=DHT.temperature;
  dato=seriale.read(); //lettura dato da smartphone
  if(dato=='u'){ //digitando il carattere u sulla tastiera dello smartphone
    seriale.println("umidità %"); //si legge "umidità %" sullo schermo dello smartphone
    seriale.println(u); //si legge il valore di umidità sullo schermo dello smartphone
  }
  if(dato=='t'){ //digitando il carattere t sulla tastiera dello smartphone
    seriale.println("temperatura °C"); //si legge "temperatura °C" sullo schermo dello smartphone
    seriale.println(t); //si legge il valore di temperatura sullo schermo dello smartphone
  }
}
```

esempio utilizzare l'applicazione *Arduino bluetooth controller* scaricabile sullo smartphone da Play Store (sul web sono comunque reperibili numerose applicazioni simili).

Segnaliamo che i normali moduli Bluetooth vengono bloccati da Apple e che pertanto i possessori di iPhone non possono realizzare questa esperienza con le modalità con cui è descritta in questo articolo.

Dopo aver scaricato l'applicazione l'icona corrispondente si trova sullo schermo dello smartphone. Di seguito si deve procedere alimentando il circuito (collegandolo al PC tramite USB) e aprendo l'applicazione (facendo clic sull'icona).

Con l'applicazione aperta si deve infine:

- procedere alla connessione del dispositivo (in questo caso il modulo HC05) tra quelli che si trovano disponibili (**Fig. 2**);
- scegliere la modalità di connessione (**Fig. 3**) tra le quattro consentite (joystick, interruttore, dimmer o terminale).

In questo caso la scelta più idonea è la modalità terminale in quanto, selezionandola, compare una tastiera che consente l'introduzione di caratteri alfanumerici che vengono visualizzati sullo schermo dello smartphone insieme con i dati acquisiti dal sensore (**Fig. 4**).

Il programma (vedere listato 1) viene strutturato in modo tale che digitando il carattere **u** si legga

sullo schermo dello smartphone il valore di umidità espresso in percentuale, digitando il carattere **t** si legga sullo schermo il valore di temperatura espresso in °C (**Fig. 5**). Le librerie richiamate nel programma sono la dht11.h (che gestisce il funzionamento del sensore) e la SoftwareSerial.h (che gestisce la trasmissione dei dati). Entrambe le librerie prevedono la definizione di due oggetti ai quali deve essere assegnato un nome (si scelgono in questo caso gli oggetti denominati DHT e seriale). Le istruzioni tipiche di queste librerie vengono ben evidenziate con dei commenti opportuni nel **Listato 1**, che riporta il firmware al completo.



Cosa occorre?

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

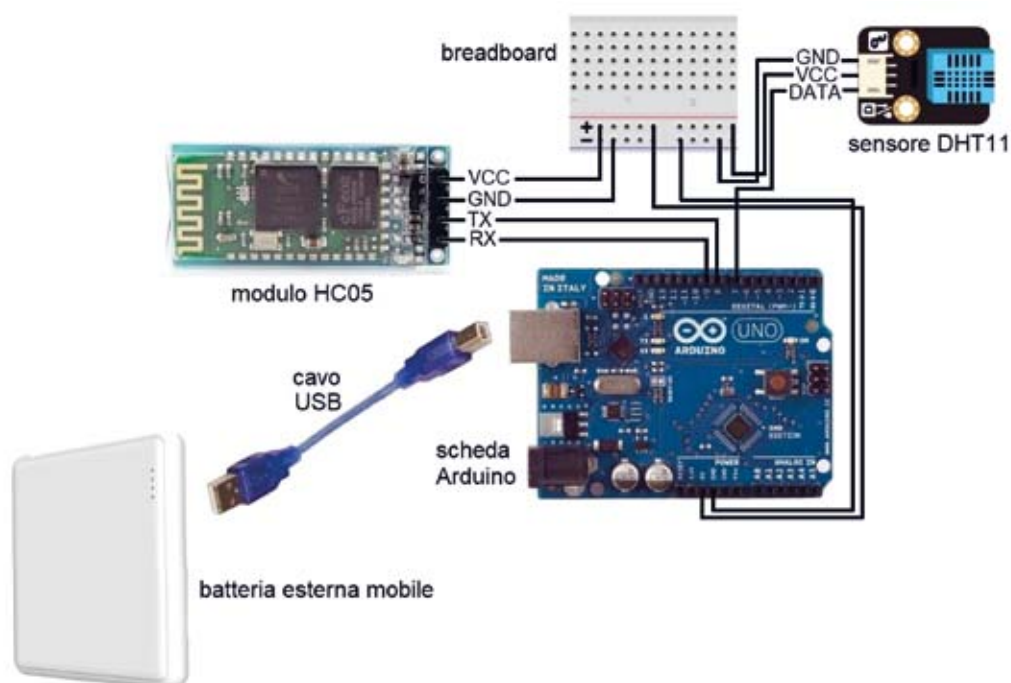
La board Arduino UNO (cod. ARDUINO UNO REV3) è in vendita al prezzo di Euro 24,50, il modulo Bluetooth (cod. HC05STRIP) costa Euro 9,00, il sensore DHT11 (cod. TEMPHUM) è disponibile a Euro 5,40, la breadboard (cod. BB400) costa Euro 4,00, i jumper per i collegamenti (cod. JUMPER65), sono in vendita a Euro 5,50.

I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

→ **Fig. 6**
Circuito per
l'alimentazione
a batteria.



VERIFICHE DI FUNZIONALITÀ

Alimentando il circuito direttamente dal PC si possono controllare i valori di temperatura e umidità che si registrano nelle immediate vicinanze del PC stesso. Invece per realizzare il controllo da remoto al quale si era in precedenza accennato si deve sostituire il PC con una batteria esterna, ad esempio un power bank, che alimenta la scheda Arduino UNO, il modulo Bluetooth e il sensore (**Fig. 6**). Due esempi concreti vengono suggeriti in **Fig. 7**.

I valori di temperatura e umidità non sono più quelli misurati nelle vicinanze del PC ma quelli che si registrano ponendo il circuito con il sensore in pieno sole (ad esempio su un terrazzo) o all'interno di un frigorifero (tenendo presente che il sensore è in grado di rilevare soltanto temperature sopra lo zero e che quindi il freezer non è il luogo indicato). Lasciamo a voi la possibilità di sperimentare questo circuito sostituendo, ad esempio, il DHT11 con un altro sensore. □

→ **Fig. 7**
Esempio
applicativo.





Sfoggia on-line i nostri cataloghi

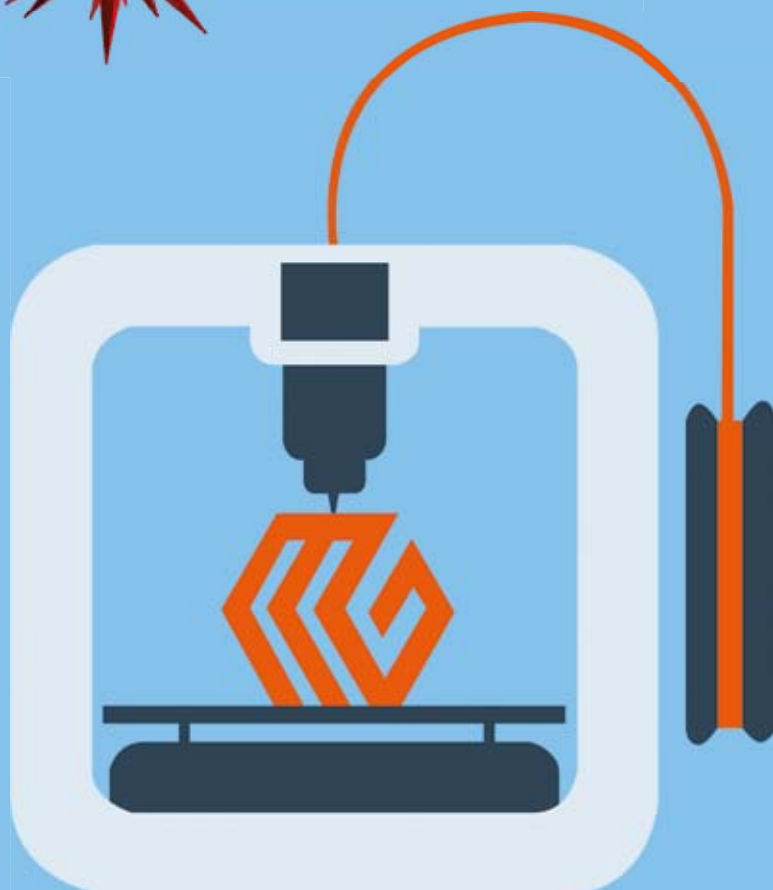
**FUTURA
ELETTRONICA®**



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



di PIER CALDERAN

Impariamo a utilizzare l'interfaccia di Node-RED per inviare i dati dal web al nostro dispositivo finale, sfruttando The Things Network e un gateway multicanale per implementare il downlink su LoRaWAN.

Nella scorsa puntata abbiamo spiegato come ricevere i dati provenienti dai dispositivi finali LoRaWAN e in che modo visualizzarli con Node-RED.

In queste pagine ci cimenteremo con l'esatto contrario, cioè mandare i controlli da Node-RED al dispositivo finale, attraverso The Things Network. Nell'esempio di oggi spiegheremo come creare dei pulsanti per accendere e spegnere un LED collegato al dispositivo finale.

MODULO NODE-RED UI LED

Oltre alla dashboard che abbiamo già installato nella scorsa puntata per aggiungere i misuratori di temperatura, umidità e luminosità all'interfaccia di Node-RED, abbiamo pensato che fosse utile installare un modulo che simuli le funzioni di un LED. Fra i tanti presenti in rete, abbiamo scelto il modulo **Node-RED UI LED**, il cui sito di riferimento è il seguente: <https://flows.nodered.org/node/node-red-contrib-ui-led>. Qui potete trovare le istruzioni per usarlo così com'è ma

anche i comandi per personalizzare il colore del LED. Per installare il modulo, procedete come al solito con il comando *npm* dal terminale:

```
npm install node-red-contrib-ui-led
```

Per impostazione predefinita il LED ha tre stati, evidenziati da tre colori di base: uno rosso, uno verde e uno grigio, come illustrato in **Fig. 1**. Per approfondire l'uso di questo modulo si consiglia di caricare nel flow gli esempi preimpostati usando la voce *Import* dal menu di Node-RED.

ARCHITETTURA DI RETE LORAWAN

Prima di affrontare l'argomento di oggi, ci sembra doveroso spiegare la logica che sta alla base dell'architettura di rete LoRaWAN. Come già detto fin dalla prima puntata, il protocollo LoRaWAN non è stato introdotto per controllare "velocemente" i dispositivi finali.



Fig. 1 - L'interfaccia grafica del modulo Node-RED UI LED.

È un sistema efficace, ma abbastanza lento. In altre parole, se pensate di usare un gateway come un radiocomando per l'apricancello, il protocollo LoRaWAN non ve lo permette. Abbiamo spiegato in questo corso come usare la tecnologia LoRa per la ricetrasmisione punto-punto a lunga distanza, ma senza utilizzare gateway e il protocollo LoRaWAN. Deve essere chiara una cosa: il protocollo LoRaWAN è stato pensato per l'utilizzo dei canali LoRa in modo tale da evitare collisioni, attacchi o connessioni fallite. Quindi, se volete sfruttare il cloud per avere il controllo totale dei vostri dispo-

sitivi finali, continuate a leggere. I sistemi di backend di The Things Network sono responsabili dell'instradamento dei dati tra i dispositivi e le applicazioni. Se per esempio guardiamo una normale rete IoT, questa richiede un gateway come ponte tra specifici protocolli radio (per esempio Wi-Fi) e Internet. Nei casi in cui i dispositivi stessi supportino lo stack IP, questi gateway devono solo inoltrare i pacchetti a Internet. Invece, i protocolli non IP, come LoRaWAN, richiedono una forma di instradamento ed elaborazione prima che i messaggi possano essere consegnati a un'applicazione. La rete TTN è posizionata tra i gateway e le applicazioni e si occupa di queste fasi di instradamento ed elaborazione (vedere la Fig. 2).

Il compito di The Things Network è quello di eseguire tutte queste funzioni di routing in modo decentralizzato e distribuito. L'obiettivo è quello di essere molto flessibile in termini di opzioni di distribuzione pubblica tramite le applicazioni create nel backend. Ovviamente è possibile anche distribuire reti private, eseguendo tutti questi componenti in un ambiente privato. In questo modo, tutti i dati rimarranno all'interno dell'ambiente privato, ma sarà comunque sempre possibile utilizzare l'account ospitato da TTN per l'autenticazione e l'autorizzazione.

FUNZIONALITÀ DI BASE DI TTN

The Things Network fornisce un server di rete LoRaWAN ovvero un protocollo cosiddetto *network-intensive*. "Intensivo"

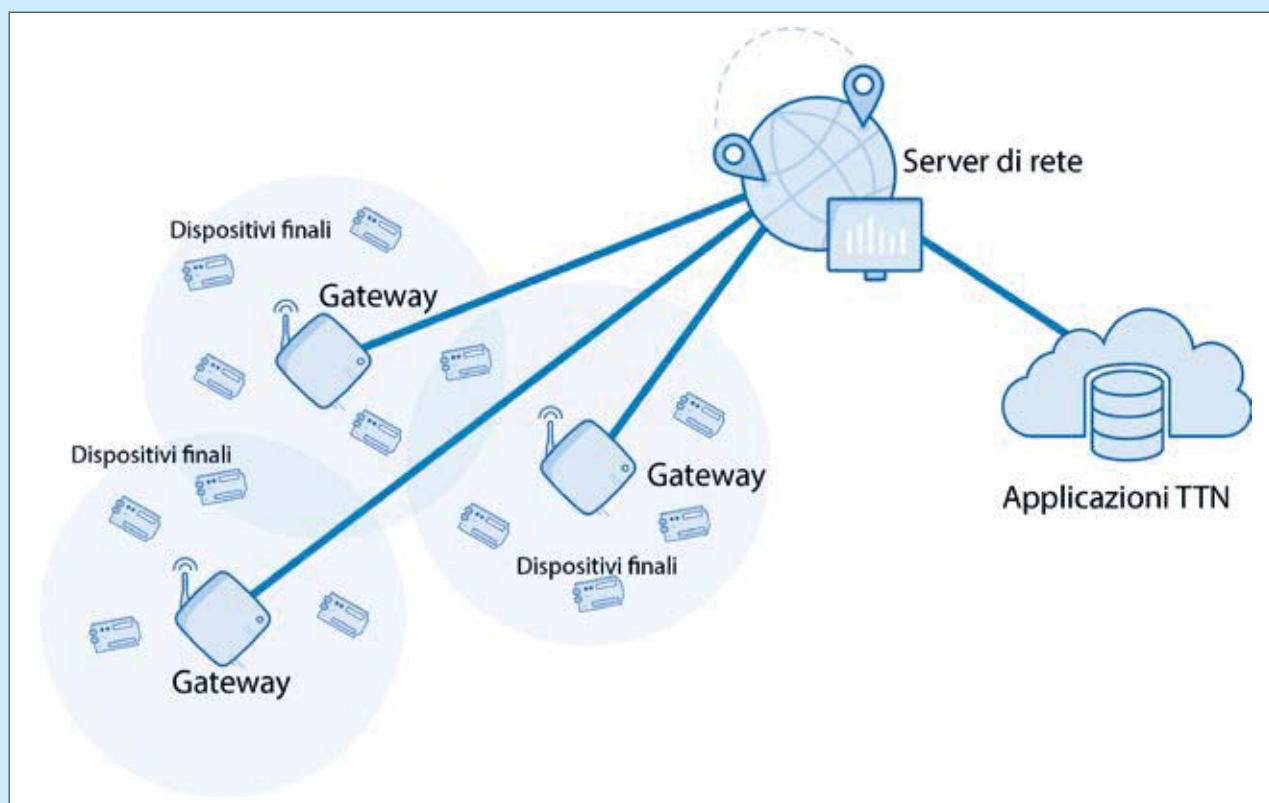


Fig. 2 - Architettura di rete LoRaWAN.

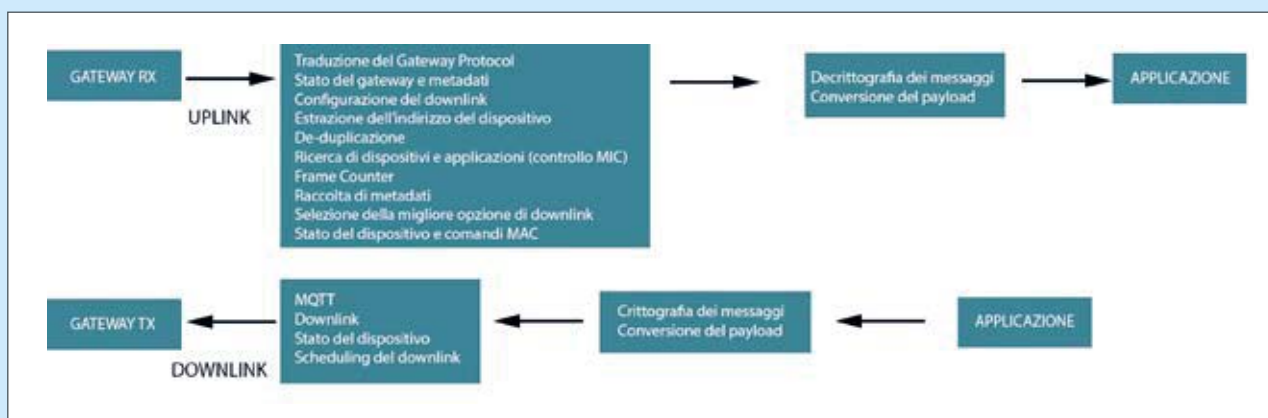


Fig. 3 Flusso di elaborazione dei messaggi di uplink e downlink.

nel senso che a causa dell'approccio dei milioni di dispositivi connessi, i sistemi di backend sono responsabili della maggior parte della logica LoRaWAN. Quindi per funzionare su un'infrastruttura distribuita come The Things Network, è stato necessario seguire alcune funzioni. Nel backend di The Things Network si possono distinguere una serie di diverse funzioni principali. Vediamo quali.

- **Scheduling** (pianificazione): le funzioni di *scheduling* servono all'utilizzo dei gateway. Lo *scheduling* è necessario perché un gateway possa eseguire solo una trasmissione o ricezione alla volta. Le informazioni sull'utilizzo vengono usate per distribuire uniformemente il carico su diversi gateway e per essere conformi ai cicli di lavoro (*duty cycle*) europei.
- **Stato dei dispositivi finali**: è necessario gestire lo stato dei dispositivi nella rete, poiché l'indirizzo del dispositivo non è univoco. Il server deve tenere traccia di quali indirizzi vengono utilizzati da quali dispositivi per mappare un messaggio al dispositivo corretto e all'applicazione. Altre cose di cui la rete deve tenere traccia sono le chiavi di sicurezza e i contatori di frame.
- **Funzionalità delle applicazioni**: per esempio, i broker MQTT e i gestori devono sapere a quale server deve essere inoltrato il traffico per un'applicazione specifica. I gestori devono sapere come interpretare i dati binari e creare collegamenti a protocolli di livello superiore, come AMQP e MQTT.
- **Funzionalità della distribuzione**: la funzionalità di rilevamento dei servizi aiuta i componenti a determinare dove deve essere instradato il traffico. Attualmente, questo è implementato come un server di rilevamento centralizzato, dando a The Things Network il controllo su quali componenti sono autorizzati ad annunciare servizi specifici.

Per rendere tutto questo lavoro possibile, si è reso necessario separare i compiti. L'idea di base è che il router sia responsabile di tutte le funzionalità relative al gateway. Un broker gestisce un intervallo di indirizzi dei dispositivi ed è

responsabile di trovare il gestore giusto a cui inoltrare ogni messaggio. Il server di rete è responsabile del mantenimento dello stato di tutti i singoli dispositivi.

Il gestore TTN è responsabile della crittografia, decrittografia e della conversione dei messaggi e dell'inoltro dei messaggi alle applicazioni.

ELABORAZIONE DEI MESSAGGI DI UPLINK E DOWNLINK

Sulla base di questa separazione dei compiti, il backend di The Things Network deve eseguire una serie di attività durante l'elaborazione dei messaggi di *uplink* e *downlink*. Una panoramica di questo flusso è illustrata nella Fig. 3 e spiegata qui di seguito:

- traduzione del Gateway Protocol;
- stato del gateway e metadati;
- configurazione del downlink;
- estrazione dell'indirizzo del dispositivo;
- deduplicazione;
- ricerca di dispositivi e applicazioni (controllo MIC);
- Frame Counter;
- raccolta di metadati;
- selezione della migliore opzione di downlink;
- stato del dispositivo e comandi MAC;
- decrittografia dei messaggi;
- conversione del payload;
- MQTT;
- downlink.
- Stato del dispositivo
- Scheduling del downlink

TRADUZIONE DEL GATEWAY PROTOCOL

Quando un gateway riceve un messaggio che è stato trasmesso su LoRa, viene incapsulato e inoltrato a TTN.

La maggior parte dei protocolli gateway (*Gateway Protocol*) ha la stessa struttura ovvero quando vengono ricevuti uno o più messaggi. Il loro payload binario viene inoltrato al backend, insieme ai metadati.

Periodicamente il gateway invia anche alcune informazioni di stato *alive* (vivo) sul gateway stesso: per esempio, coordinate GPS, potenza del segnale RSSI, rapporto segnale/rumore SNR, il numero di pacchetti ricevuti/trasmessi e altre metriche.

STATO DEL GATEWAY E METADATI

Le coordinate GPS di un gateway possono essere particolarmente rilevanti per l'applicazione, pertanto il backend memorizza l'ultimo messaggio di stato inviato dal gateway e inietta le informazioni GPS nei metadati di ogni messaggio di *uplink*.

CONFIGURAZIONE DEL DOWNLINK

In LoRaWAN la risposta di *downlink* a un messaggio di *uplink* dipende fortemente dalla regione geografica del gateway e sono descritte nella specifica "LoRaWAN Regional Parameters". Poiché il router è responsabile di tutti i dettagli relativi al gateway e specifici della regione, il router deve determinare come inviare una risposta in *downlink* a un dispositivo. Dopo ogni messaggio di *uplink*, ci sono due finestre di ricezione, una esattamente 1 secondo dopo l'*uplink*, l'altra dopo 2 secondi. Pertanto, per ogni gateway che ha ricevuto il messaggio di *uplink*, un router crea due configurazioni di *downlink*. Per selezionare l'opzione migliore in un secondo momento, il router deve inoltre calcolare un tempo per ciascuna opzione. Questo tempo è influenzato da una serie di fattori. Al momento consideriamo il tempo di trasmissione, la potenza del segnale, l'utilizzo del gateway e le trasmissioni già programmate (*scheduled*), in quanto un gateway non può fare due trasmissioni contemporaneamente. La pianificazione (*scheduling*) di un messaggio di *downlink* su un gateway che aveva una migliore potenza del segnale rende anche più probabile che un nodo riceva correttamente il *downlink*. La combinazione del tempo di trasmissione di un messaggio e l'utilizzo di un gateway viene utilizzata per ottimizzare la rete nel suo complesso. Poiché ogni trasmissione blocca per qualche tempo i ricevitori di un gateway, è meglio inviare messaggi in un tempo più breve. Pertanto, i messaggi di *downlink* a una velocità dati maggiore sono preferiti rispetto ai messaggi con una velocità dati inferiore.

ESTRAZIONE DELL'INDIRIZZO DEL DISPOSITIVO

Il primo passaggio nell'instradamento di un pacchetto è basato sull'indirizzo del dispositivo. Come abbiamo avuto modo di vedere, questo *DevAddr* è un indirizzo a 32 bit impostato dall'utente (modalità ABP) o dal gestore (modalità OTAA). TTN ha scelto di distribuire il traffico ai broker in base al prefisso dell'indirizzo del dispositivo.

DEDUPLICAZIONE

LoRaWAN è un protocollo che rende probabile che il messaggio venga ricevuto da più di un gateway. Ciò significa che il backend deve eseguire una sorta di deduplicazione per recapitare un messaggio solo una volta all'applicazione.

Ciò non significa che i duplicati non siano importanti. Anche i metadati di questi messaggi potrebbero essere preziosi. Per esempio, quando si combinano le posizioni dei gateway che hanno ricevuto il messaggio con l'ora di ricezione e la potenza del segnale, potrebbe essere possibile determinare la posizione del dispositivo che ha inviato il messaggio. L'attuale implementazione del backend deduplica localmente i messaggi di *uplink*. Il payload del messaggio sarà lo stesso per tutti i duplicati e la possibilità che durante il periodo di deduplicazione (solitamente un paio di secondi) arrivi un messaggio diverso è estremamente bassa.

Per raccogliere i metadati aggiunti al messaggio da ciascun gateway, il broker deve memorizzare nel buffer i duplicati per un certo periodo di tempo, abbastanza lungo da raccogliere il maggior numero possibile di duplicati, ma abbastanza breve da dare all'applicazione il tempo sufficiente per rispondere a un messaggio nella finestra di ricezione che verrà aperta 1 secondo dopo la trasmissione.

Le nostre misurazioni nell'attuale distribuzione di The Things Network hanno dimostrato che il ritardo medio tra il primo e l'ultimo duplicato è di poco inferiore a 100 ms, con il ritardo massimo di circa 300 ms.

RICERCA DI DISPOSITIVI E APPLICAZIONI

Poiché gli indirizzi dei dispositivi non sono univoci, è necessario determinare il dispositivo esatto che ha inviato il messaggio e l'applicazione a cui appartiene. Per fare ciò, il backend deve eseguire una serie di controlli MIC (*Message Integrity Code*), uno per ogni dispositivo che utilizza lo stesso indirizzo del dispositivo. Il broker richiede pertanto un elenco di dispositivi con l'indirizzo del dispositivo specificato dal server di rete e controlla se il MIC può essere convalidato utilizzando la chiave di sessione di rete. Se non viene trovata alcuna corrispondenza, il messaggio viene ignorato.

FRAME COUNTER

Il contatore di frame nei messaggi LoRaWAN è una misura di sicurezza utilizzata per rilevare gli attacchi di replay. Dopo aver convalidato il MIC, il broker controlla se il contatore di frame è valido. Poiché i contatori di frame possono solo aumentare, un messaggio con un contatore di frame inferiore all'ultimo contatore di frame noto dovrebbe essere eliminato. Inoltre, il broker deve verificare che lo spazio tra l'ultimo contatore di frame noto e il contatore nel messaggio non sia troppo grande.

RACCOLTA DI METADATI

Quando tutti i controlli hanno avuto esito positivo, il broker può continuare a elaborare il messaggio. Innanzitutto, unisce i duplicati ricevuti da tutti i diversi router e gateway. In questo passaggio è importante distinguere tra metadati uguali per ogni gateway che ha ricevuto il messaggio e metadati specifici per ogni ricezione. Per esempio, la frequenza, la modulazione e la velocità dei dati saranno le stesse per tutti



i gateway, quindi deve essere inoltrato solo una volta. D'altra parte, la potenza del segnale, l'ora di ricezione e le coordinate GPS di ciascun gateway dovrebbero essere tutte incluse quando si inoltra il messaggio. In questa fase vengono raccolte anche le diverse configurazioni di *downlink* per selezionare l'opzione migliore nella fase successiva.

SELEZIONE DELLA MIGLIORE OPZIONE DI DOWNLINK

Il broker deve selezionare l'opzione migliore per una risposta in *downlink* a un messaggio. Poiché il broker non dispone di alcuna informazione sul gateway che ha ricevuto un messaggio, è molto difficile farlo. Pertanto il router ha già calcolato un tempo per ciascuna configurazione di *downlink*. Se questo calcolo del tempo viene eseguito in modo standard, il broker deve solo ordinare tutte le possibili opzioni di *downlink* e utilizzare l'opzione migliore.

STATO DEL DISPOSITIVO E COMANDI MAC

Prima di inoltrare il messaggio di *uplink* al gestore, viene prima inviato al server di rete in modo che lo stato del dispositivo possa essere aggiornato. Il server di rete aggiunge anche un template di *downlink* al messaggio. Questo template può essere utilizzato dal gestore per inviare un messaggio di *downlink* al dispositivo. Contiene tutti i valori necessari (come il contatore di frame, il tipo di messaggio e gli indicatori di opzione) in modo che il gestore debba solo aggiungere al messaggio il payload dell'applicazione. Inoltre, ciò offre al server di rete la possibilità di aggiungere comandi MAC al messaggio. Per esempio, in base al numero di gateway che hanno ricevuto un messaggio e alla loro potenza del segnale, il server di rete può aggiungere comandi MAC che istruiscono il dispositivo a trasmettere a una velocità dati maggiore.

DECRITTOGRAFIA DEI MESSAGGI

Poiché i messaggi sono crittografati end-to-end, il backend è anche responsabile della decrittografia dei messaggi. Dopo aver decrittografato il payload del messaggio, il gestore può passare il messaggio all'applicazione. Tuttavia, per molte applicazioni, sono necessarie alcune semplici operazioni di decodifica e conversione del payload.

CONVERSIONE DEL PAYLOAD

Dopo la decrittografia, il gestore è in grado di decodificare e convertire il payload in un formato facilmente accessibile dall'applicazione. L'implementazione include quindi le cosiddette funzioni di formato del payload. Queste funzioni sono semplici funzioni JavaScript che possono essere utilizzate per decodificare, convertire e convalidare i dati (vedere la scorsa puntata). Il decodificatore (*decoder*) viene utilizzato per decodificare il payload binario in un formato più appropriato. Per esempio, la seguente funzione decodifica un valore di

temperatura inviato come due byte a un oggetto JSON:

```
function (bytes) {
  var data = (bytes[0] << 8) | bytes[1];
  return { temperature: data / 100.0 };
}
```

Il convertitore (*converter*) facoltativo può convertire i valori nell'oggetto JSON decodificato. Per esempio, potrebbe trattarsi di una conversione da una temperatura in Celsius a una temperatura in Fahrenheit.

Il validatore (*validator*) facoltativo può essere utilizzato per verificare la validità dei dati ed eliminare i valori anomali.

MQTT

L'implementazione del gestore predefinito pubblica su un broker MQTT semplicemente una rappresentazione JSON dei messaggi di *uplink* a un topic di questo tipo:

```
<app_id>/devices/<dev_id>/uplink
```

Ciò consente alle applicazioni di iscriversi semplicemente allo stesso topic MQTT ed elaborare i dati in qualsiasi modo.

DOWNLINK

Dopo aver pubblicato il messaggio di *uplink* su MQTT, il gestore determinerà se è necessario rispondere al dispositivo con un messaggio di *downlink*. Esistono tre situazioni in cui è necessario inviare un messaggio di *downlink*:

- il primo caso è quando l'applicazione ha un payload disponibile da inviare al dispositivo. In questo caso, il payload viene aggiunto al template di risposta generato dal server di rete;
- il secondo caso è quando il messaggio di *uplink* richiede una conferma. Indipendentemente dal fatto che un payload di *downlink* sia disponibile o meno, deve essere inviato un *acknowledgement* (riconoscimento);
- la terza situazione è quando il server di rete deve inviare comandi MAC al dispositivo.

Analogamente a quanto avviene con i messaggi di *uplink*, il gestore è responsabile della crittografia del payload del messaggio. Se non è disponibile alcun payload di *downlink*, il gestore può scegliere di attendere un breve periodo per consentire all'applicazione di preparare un messaggio di *downlink* in base al messaggio di *uplink* appena ricevuto. Trascorso tale termine, il gestore dovrà inviare nuovamente il messaggio di *downlink* al broker.

STATO DEL DISPOSITIVO

Dopo che il broker riceve un messaggio di *downlink* da un gestore, invia il messaggio che aggiornerà lo stato del dispositivo (in particolare, i contatori di frame) nel database e genererà il MIC del messaggio.

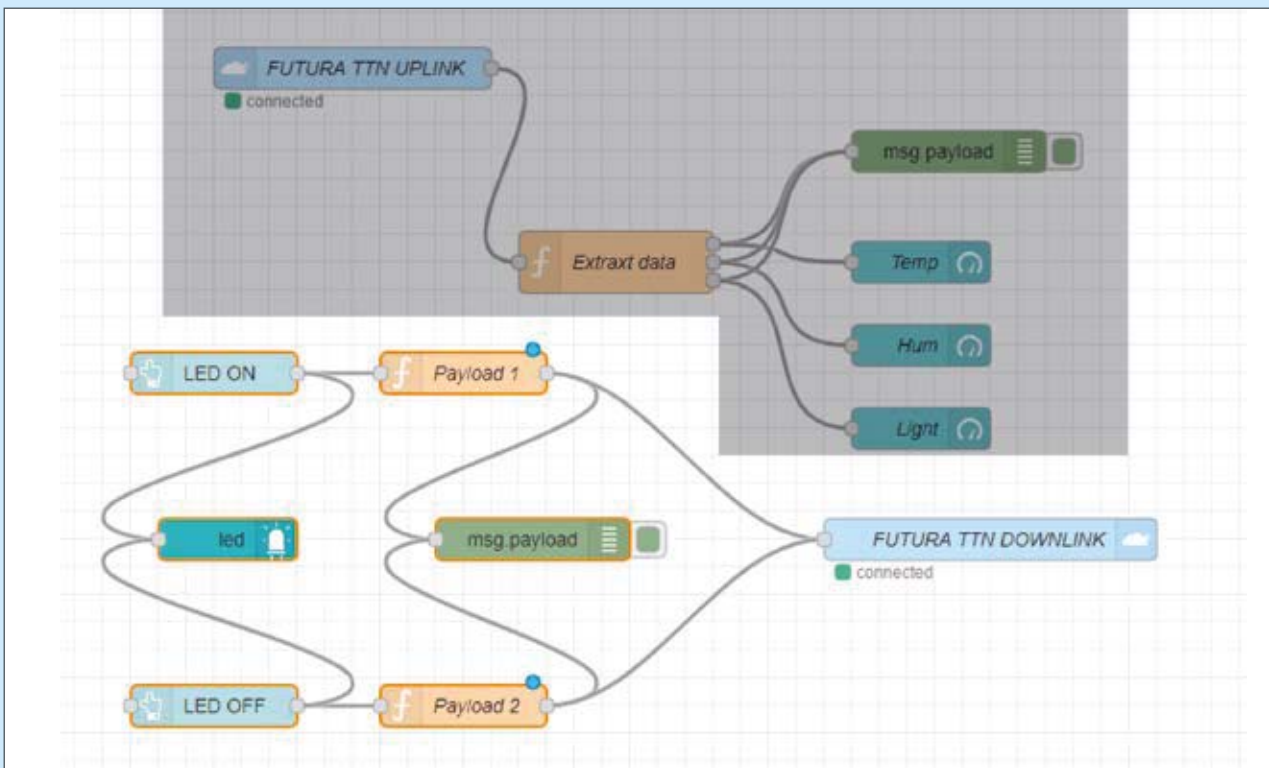


Fig. 4 - Il flusso utilizzato per leggere temperatura, luce e umidità, che qui riutilizziamo.

Successivamente, il broker inoltra il messaggio di *downlink* al router responsabile del gateway che deve trasmettere il messaggio di *downlink*.

SCHEDULING DEL DOWNLINK

Come accennato all'inizio di questo capitolo, il router è responsabile della gestione della pianificazione (*scheduling*) del gateway. Poiché la maggior parte dei gateway ha solo un buffer di un messaggio di *downlink*, il router deve memorizzare nel buffer i messaggi pianificati (*scheduled*) fino all'ultimo momento, quindi inviare ogni messaggio appena in tempo al gateway.

AGGIUNGERE DUE PULSANTI E UN LED

Dopo la spiegazione dell'architettura di rete LoRaWAN, passiamo alle cose più semplici e vediamo come aggiungere due pulsanti alla nostra interfaccia di Node-RED. Questi pulsanti manderanno dei messaggi di *downlink* a TTN, che a sua volta li manderà al nostro gateway tramite l'applicazione *futura_devices*. L'applicazione convertirà questi messaggi per il *device_1*, al quale avremo collegato un LED. Se guardate la **Fig. 4** potete notare nella parte grigia lo stesso flow della scorsa puntata, usato per l'*uplink* dei dati di temperatura, umidità e luminosità.

A questo flow abbiamo aggiunto i seguenti nodi:

- nodo downlink **FUTURA TTN DOWNLINK**
- nodo button **LED ON**

- nodo function **Payload 1**
- nodo button **LED OFF**
- nodo function **Payload 2**
- nodo **led**
- nodo **debug** (opzionale)

Una volta inseriti tutti i nodi e collegati fra loro, è necessario aprire il rispettivo editor e inserire tutti i dati di ciascun nodo nel modo seguente. Il risultato sarà simile alla **Fig.5.**, nella quale è proposto il **Nodo downlink**, i cui parametri sono:

Name: FUTURA TTN DOWNLINK

App: futura_devices

Device ID: 2601164B

Port: 1

Schedule: replace

Abbiamo poi i **Nodi button (Fig. 6)** così impostati.

Group: DOWNLINK

Size: 4x2

Label: LED ON (LED OFF per il secondo nodo button)

Port: 1

Payload: true (false per il secondo nodo button)

Topic: led

Invece il **Nodo led (Fig. 7)** ha queste impostazioni.

Group: DOWNLINK

Size: auto

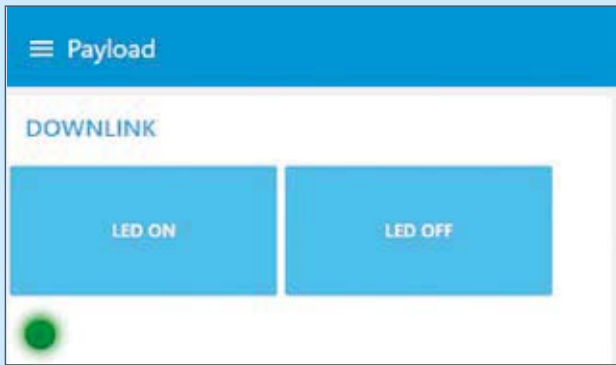


Fig. 5 - Nodo downlink.

Msg.payload: false (color red)

Msg.payload: true (color green)

Andiamo quindi ai **Nodi function (Fig. 8)**.

Name: Payload 1 (Payload 2 per il secondo nodo function)

Function:

```
return {
  dev_id: "device_1",
  port: 1,
  payload: "31"
}
```

Function (per il secondo nodo function):

```
return {
  dev_id: "device_1",
  port: 1,
  payload: "32"
}
```

È importante notare che impostando il payload dei due nodi button rispettivamente a *true* e *false* si può controllare lo

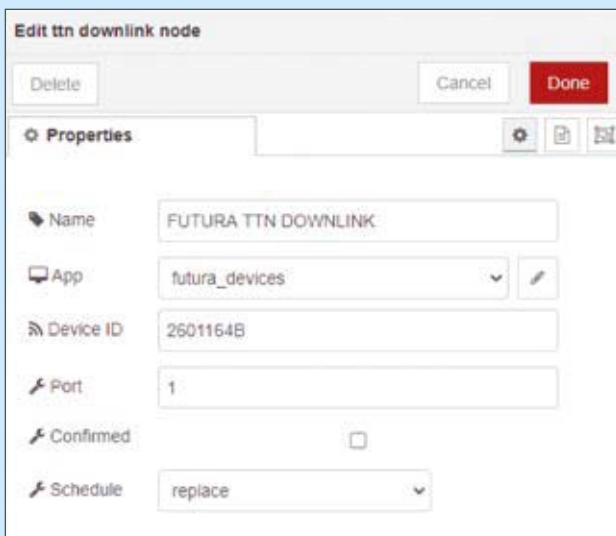


Fig. 6 - Editor del nodo downlink

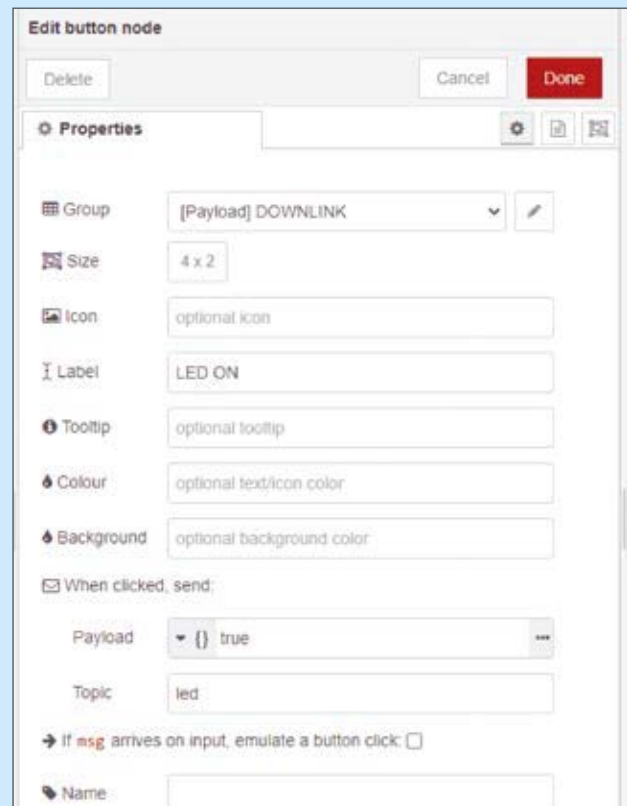


Fig. 7 - Editor del nodo button.

stato del nodo *led*: questo diventa verde se il *msg.payload* del primo button è *true* e diventa rosso se il *msg.payload* del secondo button è *false*. Ovviamente questo procedimento serve solo di esempio e non verifica lo stato effettivo del LED fisico. Per fare questo bisognerebbe creare una serie di operazioni

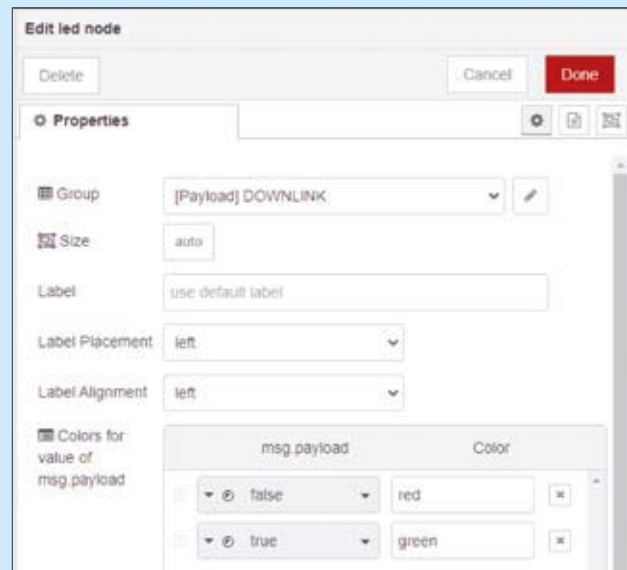


Fig. 8 - Editor del nodo led.

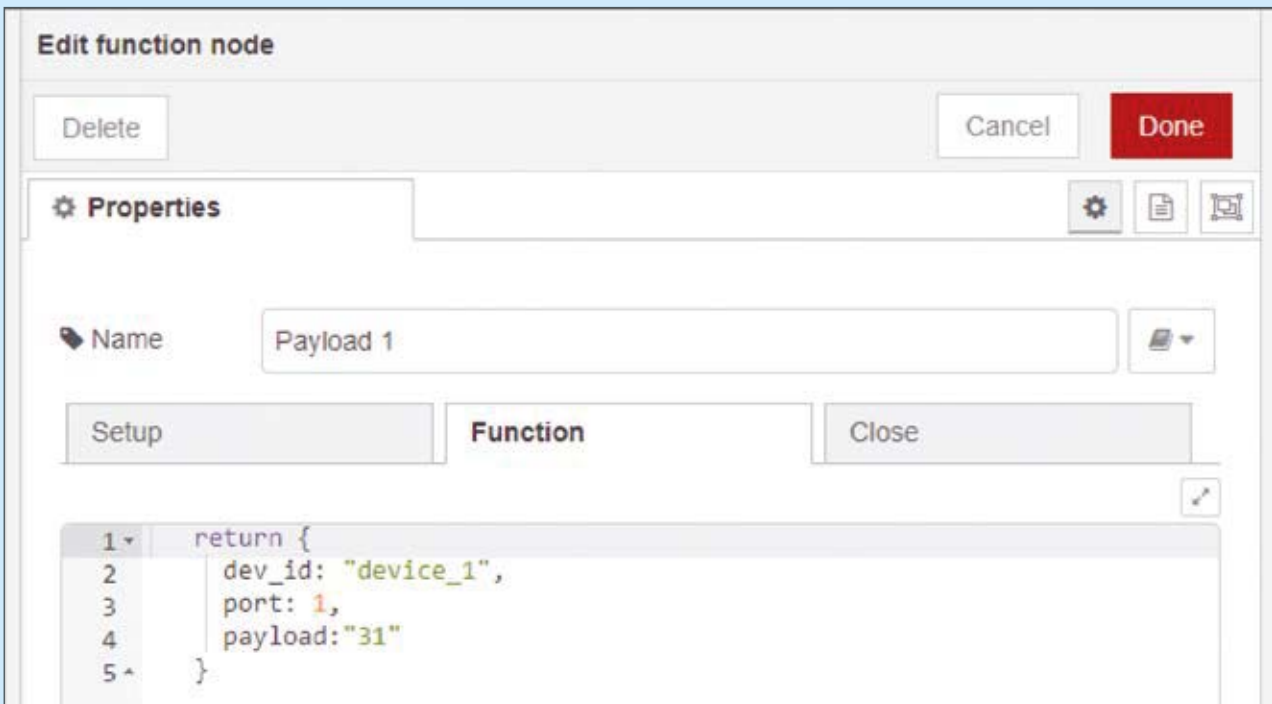


Fig. 9 - Editor del nodo function.

di *uplink* e *downlink* molto complesse che non trattiamo qui. Nel primo nodo function abbiamo inserito nel messaggio JSON i seguenti parametri che TTN leggerà e "schedulerà" sotto forma di messaggi di *downlink* al *device_1*:

- **dev_id**: "device_1", ovvero il nome del nostro dispositivo finale;
- **port**: 1, la porta disponibile al momento;
- **payload**: "32", ovvero il valore in esadecimale equivalente 49 decimale, che in ASCII corrisponde al numero 1.

Nella seconda funzione abbiamo inserito nel messaggio JSON gli stessi parametri eccetto quello per il payload: **payload**: "32", ovvero il valore in esadecimale corrispondente al decimale 50 che nella tabella ASCII corrisponde al numero 2. Riportiamo le due funzioni complete nel **Listato 1**.

In questo modo, cliccando su LED ON il nodo *led* diventerà

verde e verrà spedito un messaggio di *downlink* con il payload 0x31 (decimale 49, simbolo ASCII 1). Cliccando su LED OFF, il nodo *led* diventerà rosso e verrà spedito un messaggio di *downlink* con il payload 0x32 (decimale 50, simbolo ASCII 2). Quello che si vedrà nella sezione **Application Data** del *device_1* sarà simile alla **Fig. 10**.

Come si può vedere, insieme ai messaggi di *uplink* di temperatura, umidità e luminosità, ci sono anche i messaggi di *downlink* "31" e "32" marcati come *scheduled*. Questi messaggi verranno eseguiti dopo il successivo messaggio di *uplink*. La scelta di questi valori per il payload è arbitraria e ovviamente si possono mandare messaggi di qualsiasi tipo, sempre in esadecimale.

Per esempio, se volete provare a mandare un payload con il testo "Hello world", dovrete scrivere il payload in questo modo:

"48656C6C6F20776F726C64"

Che corrisponde ai caratteri ASCII e relative lettere:

72 (H)
101 (e)
108 (l)
108 (l)
111 (o)
32 (spazio)
119 (w)
111 (o)
114 (r)
108 (l)
100 (d)

Listato 1

```
Function payload 1
return {
  dev_id: "device_1",
  port: 1,
  payload: "31"
}

Function payload 2
return {
  dev_id: "device_1",
  port: 1,
  payload: "32"
}
```



LO SKETCH PER IL DOWNLINK

A questo punto, tutto è pronto per modificare lo sketch che avete usato nella scorsa puntata per l'*uplink*. In pratica, basta aggiungere all'evento **EV_TXCOMPLETE** il codice per l'analisi della finestra di ricezione **RX**: se questa finestra conterrà il payload 0x31 o 0x32 di *downlink* da TTN, che sono gli stessi valori 0x31 o 0x32 mandati da Node-RED, potremo leggerli e abbinare a questi il codice per l'accensione e lo spegnimento del LED. Nel **Listato 2** potete vedere le modifiche apportate che riguardano sostanzialmente l'accensione e lo spegnimento di un LED collegato al pin digitale 5 del nostro dispositivo finale. Il circuito elettronico andrà modificato di conseguenza, come illustrato in **Fig. 11**. Il codice aggiunto si riferisce alla definizione della variabile **PIN_LED** come pin 5 e la sua modalità come **OUTPUT**. La parte di codice per il *downlink* è stata inserita nella funzione **onEvent**. All'interno dello switch della funzione, in caso l'evento sia **EV_TXCOMPLETE** verrà analizzata la finestra di **RX** che viene ricevuta dopo l'*uplink* dei dati avvenuto con successo. All'interno della finestra di ricezione del *downlink* si potranno analizzare i dati con le seguenti funzioni della libreria LMIC:

- **LMIC.dataLen**: può essere 0 (nessun dato) oppure maggiore di 0 (byte di dati);
- **LMIC.dataBeg**: può essere 0 (nessun dato) o inizio dei dati

Se **LMIC.dataLen** è maggiore di zero allora si entra nel ciclo *for* per memorizzare i dati nella variabile *mydata*. Sappiamo che i dati ricevuti possono essere 0x31 o 0x32, per cui li memorizzeremo come *char* nel primo elemento dell'array **mydata[0]**. A questo punto il gioco diventa facile:

```
if (mydata[0] == '1') digitalWrite(PIN_LED,1);
if (mydata[0] == '2') digitalWrite(PIN_LED,0);
```

Significa che se il primo elemento dell'array *mydata* ha il valore ASCII 1 allora il LED si accende. Se il valore ASCII è 2, lo spegnerà. Si fa notare che nello sketch abbiamo impostato l'intervallo di **TX_INTERVAL** a 10 secondi per ottenere un *downlink* in questo lasso di tempo. Questo significa che il LED potrà venire spento o acceso entro l'intervallo di 10 secondi, per cui non aspettatevi la contemporaneità degli eventi.

Listato 2

```
#define PIN_LED 5

void onEvent (ev_t ev) {
  switch(ev) {
    ...
    ...
    case EV_TXCOMPLETE:
      Serial.println(F("EV_TXCOMPLETE (includes waiting for RX windows)"));

      char mydata[2]={0};
      if (LMIC.dataLen)
      {
        Serial.println(F("Received "));
        Serial.print(LMIC.dataLen);
        Serial.println(F(" bytes of payload"));
        Serial.print("txCnt :");
        Serial.println(LMIC.txCnt);
        Serial.print("txrxFlags :");
        Serial.println(LMIC.txrxFlags);
        Serial.print("dataBeg :");
        Serial.println(LMIC.dataBeg);
        for (int i = 0; i < LMIC.dataLen; i++)
        {
          if (LMIC.frame[LMIC.dataBeg + i] < 0x10) {
            Serial.print(F("0"));
          }
          Serial.print(LMIC.frame[LMIC.dataBeg + i], HEX);
          mydata[i]=char(LMIC.frame[LMIC.dataBeg + i]);
        }
        Serial.println();
        Serial.println(mydata);

        if (mydata[0]=='1') digitalWrite(PIN_LED,1);
        if (mydata[0]=='2') digitalWrite(PIN_LED,0);

        // Schedule next transmission
        os_setTimedCallback(&sendjob, os_getTime()+sec2osticks(TX_INTERVAL), do_send);
        break;
      }
    }

  void setup() {
    pinMode(PIN_LED,OUTPUT);
    ...
    ...
    ...
  }
}
```

APPLICATION DATA

|| pause

■ clear

Filters

uplink

downlink

activation

ack

error

time

counter

port

▼

14:13:58

1

payload: 32

▲

14:13:56

461

1

payload: 54 18 48 28 4C 1F 00

h1: 72

h2: 40

f1: 76

f2: 31

t1: 84

t2: 24

▼

14:13:47

1

scheduled

payload: 32

▲

14:13:32

459

1

payload: 54 18 48 28 4C 20 00

h1: 72

h2: 40

f1: 76

f2: 32

t1: 84

t2: 24

▼

14:13:20

1

payload: 31

▲

14:13:19

458

1

payload: 54 18 48 28 4C 21 00

h1: 72

h2: 40

f1: 76

f2: 33

t1: 84

t2: 24

▼

14:13:09

1

scheduled

payload: 31

▲

14:13:07

457

1

payload: 54 17 48 28 4C 21 00

h1: 72

h2: 40

f1: 76

f2: 33

t1: 84

t2: 23

Fig. 10 - I dati di uplink e downlink dell'applicazione.

In altre parole, quando premete LED ON o LED OFF posso-
no passare da 2 a 10 secondi prima di vedere accendersi o
spegnersi il LED. Questo dipende dal tempo di *scheduling* dalla
ricezione del messaggio del successivo *uplink*.
Quello che abbiamo proposto qui è solo un semplice esem-
pio per iniziare a sperimentare. Per chi vuole dilettarsi con le
API della libreria LMIC, consigliamo di visitare la pagina web
<https://lora-developers.semtech.com/resources/tools/basic-mac/programming-model-and-api/>.

UPLINK E DOWNLINK CON IC880A

Il gateway che abbiamo usato fino ad oggi per le opera-
zioni di *uplink* è a canale singolo e per usarlo abbiamo fru-
ito del *single channel packet forwarder* di Thomas Telkamp
(https://github.com/tftelkamp/single_chan_pkt_fwd).
Purtroppo, come dice lo stesso autore, con questo software
non è possibile effettuare il *downlink* da TTN. Come abbiamo
avuto modo di sottolineare, l'uso di un gateway a canale sin-
golo può essere considerato come un modo veloce ed econo-

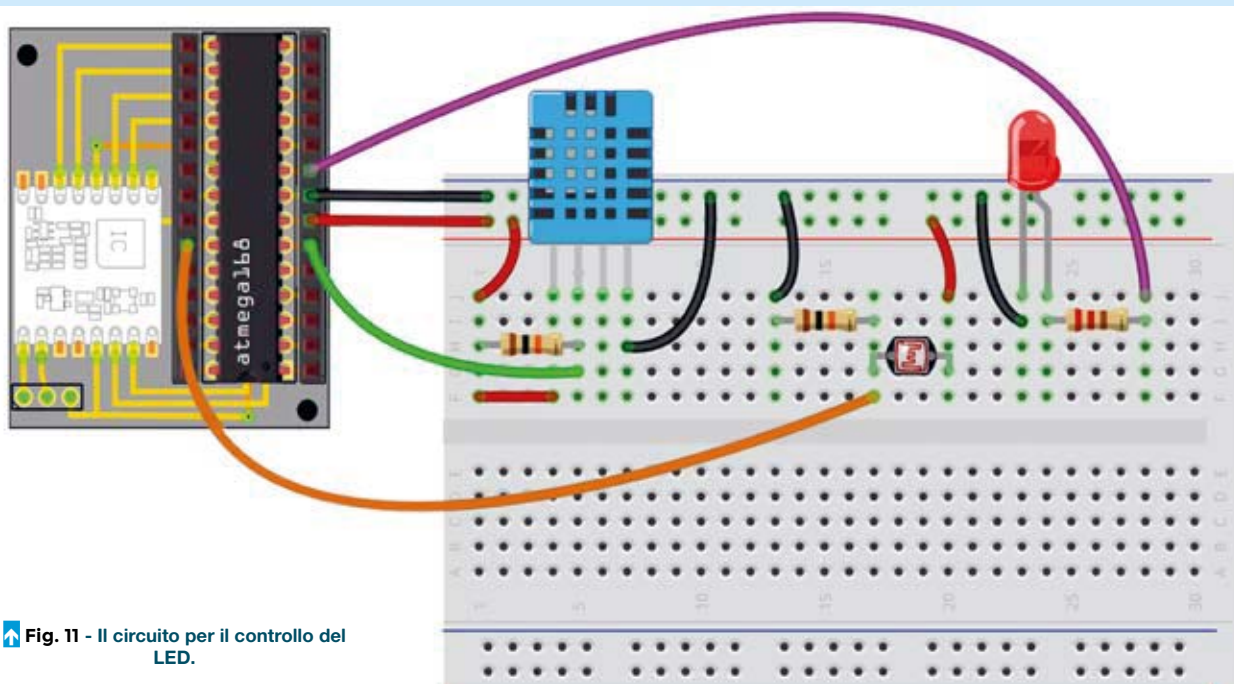


Fig. 11 - Il circuito per il controllo del LED.

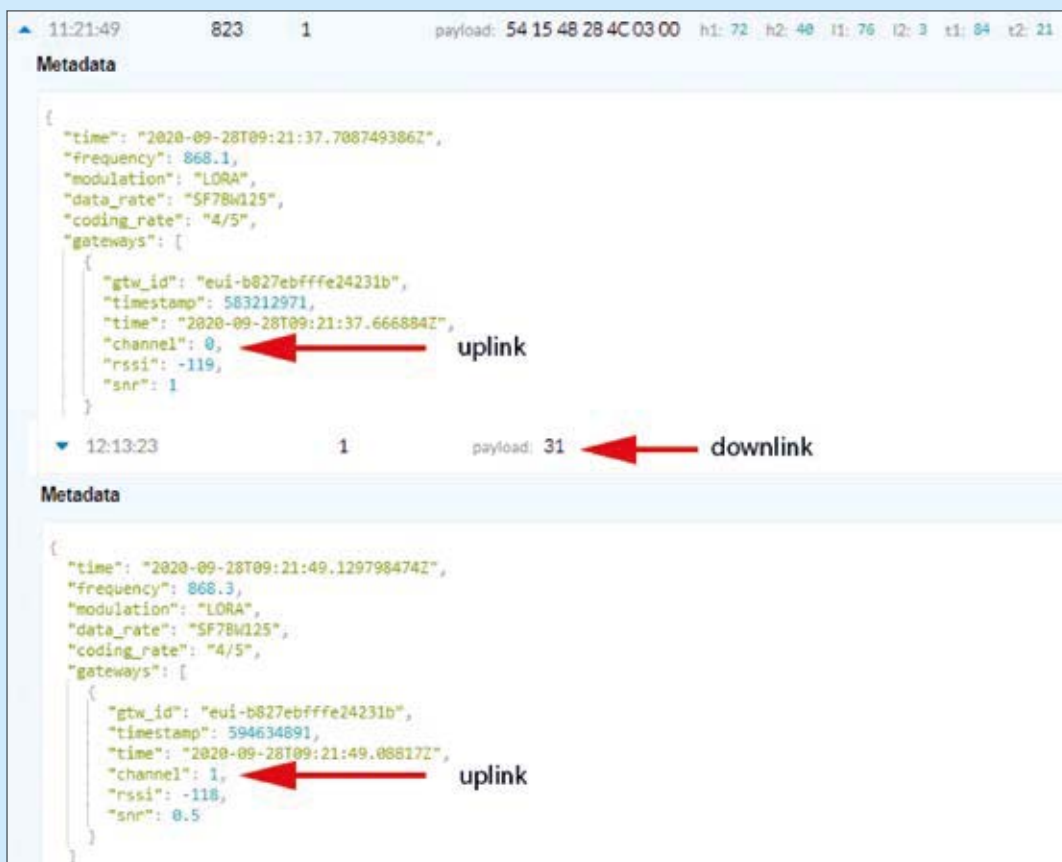


Fig. 12
I due uplink
prima e dopo un
downlink.

mico per sperimentare con LoRaWAN, ma per applicazioni professionali diventa obbligatorio usare un gateway a 8 canali. Per esempio il gateway iC880A della Wireless Solutions (<https://wireless-solutions.de/products/long-range-radio/ic880a.html>). Ricordiamo che questo gateway è basato sul chip multicanale SX1301 e pertanto è adatto per il *packet forwarder* bidirezionale di Semtech (https://github.com/Lora-net/packet_forwarder). Per il collegamento del gateway a Raspberry Pi rimandiamo alla seconda puntata di questo corso. Facciamo notare che non è possibile utilizzare il *single channel packet forwarder* con il gateway iC880A. Perché non sia possibile usare un gateway single channel per l'*uplink* e il *downlink* è spiegabile osservando la **Fig. 12**, dove vedete due *uplink* avvenuti prima e dopo un *downlink*: notare che vengono usati due canali di *uplink* diversi ogni volta.

THE THINGS NETWORK ZURICH

Oltre al già citato *packet forwarder* di Semtech (che comunque vi invitiamo a provare) è disponibile un *packet forwarder* di The Things Network Zurich. Questa è una comunità di hacker/maker svizzeri che ha modificato il *packet forwarder* originale di Semtech, apportando molte migliorie. La comunità ha al suo attivo circa 200 gateway sparsi nel territorio svizzero ed è in continua espansione. Oltre al *packet forwarder*, il GitHub della comunità mette a disposizione una serie di utility molto

interessanti che vi invitiamo a provare a visitare <https://github.com/ttn-zh>.

Al link <https://github.com/ttn-zh/ic880a-gateway/wiki> trovate un'ottima guida "From zero to LoRaWAN in a weekend" per costruire il vostro gateway con Raspberry Pi e iC880A in un weekend, come viene dichiarato nel titolo. Seguendo le istruzioni passo passo, capirete come installare facilmente il *packet forwarder* ed essere operativi in pochi minuti.

La **Fig. 13** illustra il nostro gateway su TTN che abbiamo chiamato ttn zh. Per completezza di informazione, vi diamo alcune indicazioni per la corretta installazione e l'uso di questo *packet forwarder* della comunità svizzera.

Dal terminale di Raspberry Pi digitate questi comandi per clonare il repository e installare il *packet forwarder*:

```
git clone https://github.com/ttn-zh/ic880a-gateway.git ~/ic880a-gateway
cd ~/ic880a-gateway
sudo ./install.sh spi
```

L'installazione chiederà di inserire alcune informazioni; cliccate enter ad ogni opzione in modo da accettare le impostazioni predefinite. Per impostazione predefinita, il programma di installazione cambia il nome host del Raspberry Pi in ttn-gateway (per evitare collisioni con altre Raspberry Pi eventualmen-

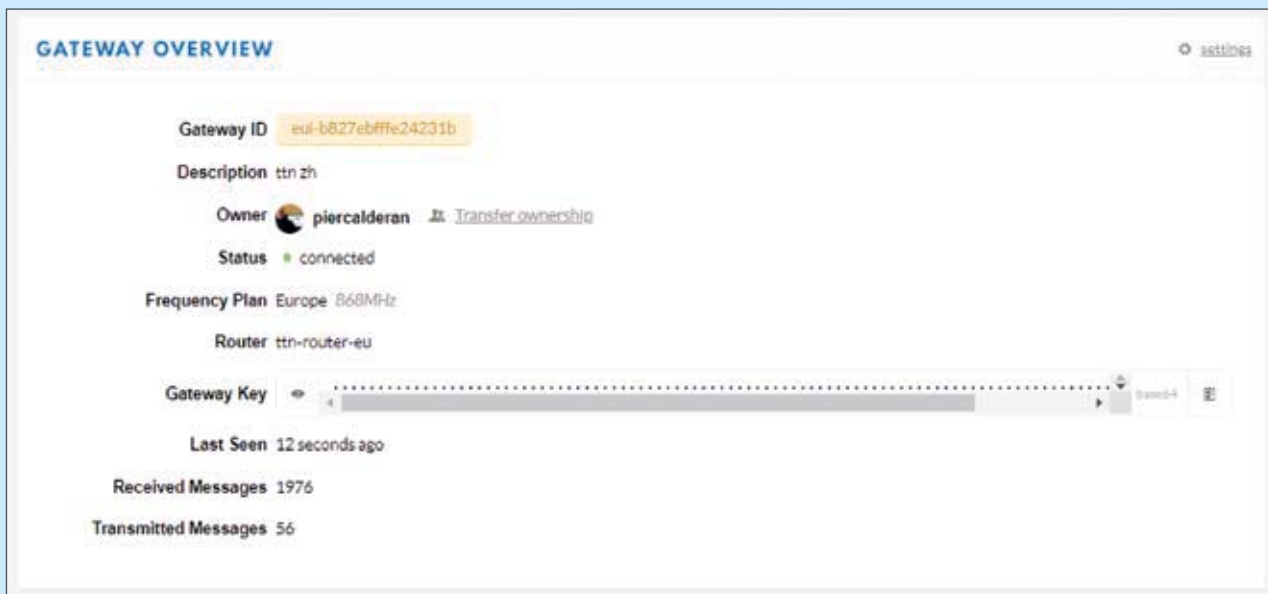


Fig. 13 - Il nostro gateway ttn zh.

te connessi a Internet). Dopo l'installazione viene riavviato il sistema; dopo il riavvio, dal terminale digitate questi comandi per entrare nella directory dove si trova l'eseguibile ed avviarlo con uno script:

```
cd /opt/ttn-gateway/bin
sudo ./start.sh
```

Se aprite lo script **start.sh** con un editor di testo, noterete che contiene già le istruzioni per il reset del gateway ic880A il cui pin di reset va collegato al pin GPIO25 di Raspberry Pi. Una

volta avviato il *packet forwarder* si vedrà qualcosa di simile alla Fig. 14.

CONCLUSIONI

Bene, termina qui questa puntata dedicata alla tecnologia LoRa e alla rete LoRaWAN, nella quale vi abbiamo descritto come usare l'interfaccia di Node-RED per inviare i dati dal web al vostro dispositivo finale, sfruttando The Things Network e un impiegando gateway multicanale per implementare il downlink sulla rete LoRaWAN. Non ci resta che darvi appuntamento alla prossima puntata!

Fig. 14
La finestra del packet forwarder di The Things Network Zurich.

```
pi@ttn-gateway: /opt/ttn-gateway/bin
File Modifica Schede Aiuto
INFO: [up] PUSH_ACK for server router.eu.thethings.network received in 58 ms
INFO: [down] for server router.eu.thethings.network PULL_ACK received in 57 ms
INFO: [down] for server router.eu.thethings.network PULL_ACK received in 53 ms

##### 2020-09-27 19:01:47 GMT #####
### [UPSTREAM] ###
# RF packets received by concentrator: 2
# CRC_OK: 50.00%, CRC_FAIL: 50.00%, NO_CRC: 0.00%
# RF packets forwarded: 1 (20 bytes)
# PUSH_DATA datagrams sent: 2 (442 bytes)
# PUSH_DATA acknowledged: 100.00%
### [DOWNSTREAM] ###
# PULL_DATA sent: 3 (100.00% acknowledged)
# PULL_RESP(onse) datagrams received: 0 (0 bytes)
# RF packets sent to concentrator: 0 (0 bytes)
# TX errors: 0
### [GPS] ###
# Invalid gps time reference (age: 1601233307 sec)
# Manual GPS coordinates: latitude 0.00000, longitude 0.00000, altitude 0 m
##### END #####
INFO: [up] PUSH_ACK for server router.eu.thethings.network received in 57 ms
INFO: [down] for server router.eu.thethings.network PULL_ACK received in 56 ms
INFO: [down] for server router.eu.thethings.network PULL_ACK received in 56 ms
```




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. FISHINO UNO
€ 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
€ 49,90

Fishino32



cod. FISHINO32
€ 59,00

- 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

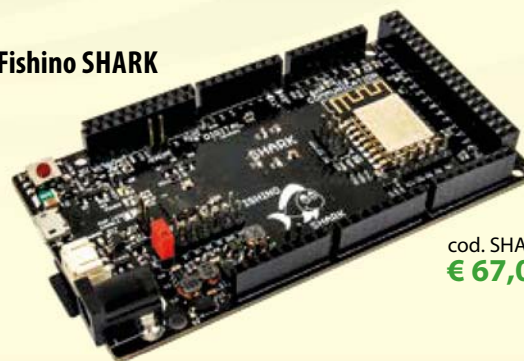
Fishino GUPPY



cod. GUPPY
€ 33,90

- 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
€ 67,00

- 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
€ 44,90



- 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.



Arriva il LASER Terabit "freddo"

Le onde Terahertz (radiazioni submillimetriche o luce del lontano infrarosso, tra 0,1 e 10 THz) si trovano nella porzione di spettro delle radiazioni elettromagnetiche fra le onde ottiche e le microonde radio. Potrebbero trovare molte applicazioni nell'imaging, come il rilevamento di molti esplosivi e droghe illegali, la scansione di tumori, l'identificazione di strutture proteiche, test non distruttivi e controllo di qualità. Potrebbero anche essere la chiave per le future reti wireless ad alta velocità (terabit al secondo). Tuttavia l'utilizzo dei raggi terahertz è impraticabile per la mancanza di sorgenti potenti e compatte. I dispositivi a semiconduttore convenzionali possono generare onde con frequenza inferiore a 1 o superiore a 10 THz, ma non tra queste (gap terahertz). Una soluzione è l'utilizzo di laser a cascata quantistica, basati su wafer costituiti da strati alternati di materiale semiconduttore con diversi livelli di conduttività elettrica: quando sono caricati elettricamente, gli elettroni rimangono confinati negli strati più elettricamente conduttivi ed emettono luce in una "cascata quantistica". Tuttavia tali laser richiedono temperature inferiori a -63 °C, perché oltre, gli elettroni confinati sfuggono attraverso gli strati più elettricamente resistenti.

Ora gli scienziati del MIT hanno ideato un laser quantistico a cascata ad alta potenza che emette a circa 4 THz, capace di operare a -23 gradi, che quindi potrebbe funzionare raffreddato da celle di Peltier. Il wafer consiste in un foglio di GaAs elettricamente più conduttivo, inserito tra strati di arseniuro di gallio e alluminio più resistenti elettricamente. Probabilmente si può regolare il laser per emettere anche tra 2 e 7 THz, ma i ricercatori puntano ad arrivare a 10 THz ricorrendo a wafer di nitrato di gallio (GaN).

news.mit.edu

Il LED che abbatte

Da quando ci siamo imbattuti nel COVID-19, sono state messe in campo svariate soluzioni tecnologiche per sanificare gli ambienti, ciascuna basata su una precisa logica di funzionamento: ozono, luce UVC e quant'altro. E sono state avviate sperimentazioni, spesso in accordo con Istituzioni ed Enti Pubblici. Un esempio è la sperimentazione in corso nel torinese, per l'esattezza a Rivoli, dove si sta utilizzando una tecnologia innovativa per depurare l'aria nelle aule dell'Istituto di Istruzione Secondaria Giulio Natta. Primo in Italia, l'Istituto Scolastico adotta dispositivi per la sanificazione dell'aria che sfruttano un processo di fotocatalisi a luce visibile per abbattere i betacoronavirus, famiglia cui appartiene il tristemente noto Sars-CoV-2 (o COVID-19, come è stato classificato). Il trattamento biologico dell'aria verrà applicato in tutte le 27 aule dell'istituto scolastico per decomporre le sostanze organiche presenti.

La fotocatalisi con luce LED messa in campo da WivActive è una tecnologia sviluppata dal centro di ricerca TechLab creato da Wiva Group, azien-

da specializzata nella progettazione e produzione di sorgenti luminose e apparecchi di illuminazione. Gli studi e le ricerche effettuate hanno portato allo sviluppo di un nanomateriale formato da biossido di titanio drogato con azoto e con nanoparticelle di argento insieme a particolari elementi chimici, che permette di svolgere il processo di fotocatalisi senza bisogno degli ultravioletti che tipicamente sono alla base di processi simili impiegati per agire su DNA ed RNA dei virus.

Infatti i processi fotocatalitici mediati dal biossido di titanio (il più usato composto chimico fotocatalizzatore), che operano effetti biocidi e di disinfezione attaccando batteri, spore, funghi e virus, sinora sono stati realizzati combinando la luce ultravioletta, che può andare bene in contenitori chiusi e comunque che non può essere irradiata negli ambienti frequentati dalle persone a causa dei danni che causerebbe agli occhi.

L'innovazione introdotta dalla tecnologia WivActive, invece, sta nel fatto che è stato adottato biossido di titanio additivato con azoto e argento, con-



il coronavirus

sentendo di implementare fotocatalizzatori efficaci e attivati già con la luce visibile. Il composto a base di TiO_2 così ottenuto viene applicato in forma coating superficiale anche sui filtri ceramici degli apparati, incrementando la capacità di filtrazione grazie alla combinazione con la luce a LED.

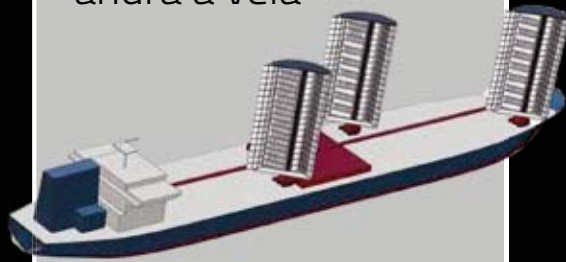
In totale, nel plesso scolastico sono stati installati 112 dispositivi per la sanificazione quotidiana e continua dell'aria, che utilizzano la tecnologia WivActive, chiamati Air Panel. Ciascuno di questi, attraverso la fotocatalisi con luce visibile prodotta da LED e l'impiego di biossido di titanio (TiO_2) drogato, riesce ad abbattere le particelle virali in sospensione.

Gli Air Panel installati sono dotati di sensori che permettono il monitoraggio sia dell'aria che del proprio funzionamento. I sensori verificano sistematicamente i dati di temperatura, umidità e qualità dell'aria, per la quale viene misurata la concentrazione dei principali inquinanti e dei composti organici volatili. L'insieme è introdotto nei corpi illuminanti a LED utilizzati per l'illuminazione degli ambienti.

Il sistema verifica anche tutti i parametri di funzionamento, allo scopo sia di pianificare la manutenzione preventiva degli apparati per mantenerne l'efficienza dei sistemi (per esempio la rigenerazione dei filtri) sia per programmare eventuali interventi di assistenza tecnica che si rendessero necessari. Infine, il sistema trasferisce i dati su un Cloud dove una specifica applicazione li storicizza e li rende disponibili per la consultazione. Il Dirigente Scolastico o l'eventuale Autorità preposta potrà così verificare costantemente lo stato di salubrità dei locali e l'azione dei dispositivi di sanificazione e impostare la programmazione che rispecchi le reali esigenze d'uso dei locali, ottimizzando anche costi e consumi. La certificazione antivirale ottenuta dalla tecnologia WivActive attesta la capacità antivirale di agire sulla disgregazione e abbattimento del Betacoronavirus grazie ai radicali liberi generati dal sistema fotocatalitico attivato con luce visibile a spettro calibrato.

www.wivagroup.com

Il Cargo del futuro andrà a vela



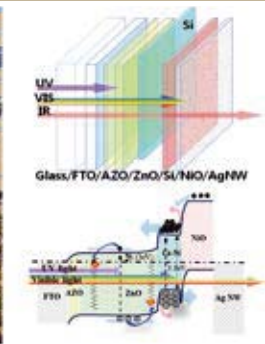
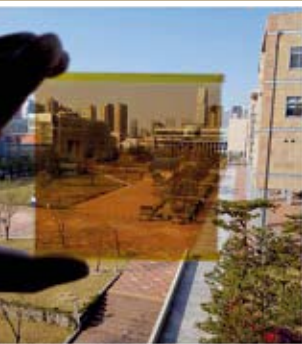
Le esigenze di contenimento dei consumi energetici e dell'inquinamento stimolano costantemente nuove idee, talvolta innovative, ma in qualche caso ispirate dalle lezioni del passato, come nel caso di un progetto di Cargill Incorporated, il principale commerciante di materia prima agricola del mondo, che unisce innovazione e tradizione. L'idea è di rinnovare il parco nautico con nuove navi cargo capaci di sfruttare l'energia del vento grazie a enormi vele alari applicate sul ponte, com'era sui velieri di un tempo. Non si tratterebbe di sostituire la tradizionale propulsione diesel o elettrodiesel, ma di complementarla nell'ottica di raggiungere l'obiettivo di diminuire le emissioni generate dal settore dei trasporti marittimi del 50% entro il 2050.

Le vele saranno del tipo ad ala solida, alte fino a 45 metri, realizzate in materiale composito; assomiglieranno ad ali mobili degli aerei e saranno pieghevoli, per evitare che, data l'altezza, urtino gru dei porti, ponti o altre strutture in navigazione o in manovra. Gli scafi saranno studiati in modo da far risparmiare fino al 30% in combustibile. Considerando che la società ha in media circa 600 navi, sostituendole tutte il risparmio di combustibile sarebbe notevole e varrebbe sia per il tipo più inquinante in uso al momento sia per quelli più ecosostenibili (come idrogeno o gas naturale liquefatto).

Il progetto di Cargill è finalizzato a mettere in mare entro il 2022 navi cisterna MR2 dotate di tre vele alari; partirà da queste navi di medie dimensioni che trasportano olio, biocarburante o etanolo e se avrà successo verrà esteso a natanti più grandi come le portarinfuse che trasportano cereali e semi.

La parte tecnica e tecnologica del progetto è stata affidata a Bar Technologies, società britannica che discende dalla Ben Ainslie Racing, la nota squadra di vela vincitrice della Coppa America.

www.cargill.com



Il fotovoltaico diventa trasparente

sparente arriva dalla Incheon National University, in Corea del Sud e rappresenta un'innovazione perché combina le due tecnologie usate nei pannelli fotovoltaici: è formata da uno strato ultrasottile di silicio amorfo tra due semiconduttori trasparenti, l'ossido di zinco e l'ossido di nichel. Questo consente l'utilizzo degli UV, la cattura di una gran quantità di fotoni e il trasporto più veloce delle particelle cariche agli elettrodi. Inoltre, può potenzialmente generare elettricità anche in situazioni di scarsa illuminazione (ad esempio, nei giorni nuvolosi o piovosi). Sulla base dei risultati ottenuti in laboratorio, i ricercatori ritengono che questa tecnologia sarà presto applicabile.

www.sciencedirect.com

L'ideale per un pannello fotovoltaico è che abbia il minor impatto visivo possibile, tanto da essere applicato anche alle vetrate. È questo il concetto che da sempre sta dietro alle celle solari trasparenti, tecnologia nata per facilitare l'integrazione delle rinnovabili nell'edilizia. I pannelli sono progettati per catturare solo alcune lunghezze d'onda e lasciar passare le altre: nello specifico, trattengono gli ultravioletti. L'ultimo ritrovato nell'ambito della tecnologia a film tra-

Soyuz MS-17: da Bajkonur all'ISS in sole 3 ore

La navicella spaziale Soyuz MS-17 non smette di stupire e nella sua ultima missione è riuscita a raggiungere la Stazione Spaziale Internazionale attraccando (realizzando quindi un rendez-vous) dopo appena tre ore. Lanciata la mattina del 14 ottobre scorso attraverso un vettore Soyuz 2.1a dal Cosmodromo di Bajkonur, in Kazakistan, ha trasportato il proprio equipaggio nello spazio per un rapidissimo rendez-vous di tre ore a due orbite con la ISS. L'evento è unico ed ha segnato una pietra miliare perché la Soyuz è stata la prima ad utilizzare un nuovo schema di rendez-vous "ultraveloce" con la ISS. Dopo un'ascesa impeccabile verso l'orbita prevista, la russa Soyuz MS-17 ha raggiunto il laboratorio orbitante in sole tre

ore, praticamente metà del tempo richiesto per raggiungere la stazione spaziale. Il successo dell'operazione è stato tale che il metodo collaudato sostituirà il precedente che ha visto la navicella Soyuz trascorrere quattro orbite in volo libero per raggiungere la ISS in circa sei ore. Le immagini della grande impresa sono state prontamente condivise dalla NASA sui canali social ufficiali. Questa missione ha avuto un equipaggio internazionale, guidato dal comandante Sergey Ryzhikov e composto dal cosmonauta Roscosmos, da Sergei Koud-Svertchikov e dall'astronauta americana Kathleen Rubins della NASA.

www.nasaspacelife.com

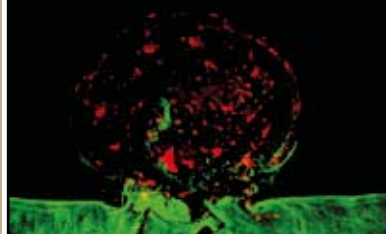


Stampare in 3D aneurismi per curarli meglio

Gli aneurismi cerebrali sono prolassi delle pareti dei vasi sanguigni, che in caso di rottura possono portare a gravi patologie come ictus, danni cerebrali e morte; il loro trattamento è chirurgico ma l'esito può variare notevolmente da persona a persona e in tal senso è utile avere dei modelli su cui simulare le cure. In quest'ottica al Lawrence Livermore National Laboratory hanno stampato in 3D un aneurisma vivente bioprintato per sperimentare una procedura medica su di esso e osservare l'esito come se fosse nel cervello umano.

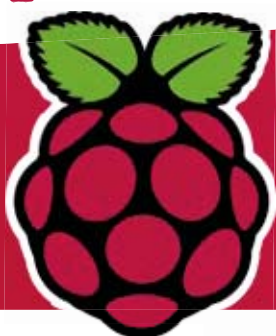
L'aneurisma in vitro è stato replicato stampando in 3D vasi sanguigni con cellule cerebrali umane e su di esso è stata eseguita una procedura di riparazione endovascolare inserendo un catetere nel vaso sanguigno e avvolgendo le bobine di platino all'interno del sacco dell'aneurisma. Dopo il trattamento di impaccamento, i ricercatori hanno introdotto plasma sanguigno nell'aneurisma e hanno osservato la formazione di un coagulo di sangue dove si trovavano le bobine; sono stati anche in grado di osservare il processo di guarigione post-operatorio delle cellule endoteliali all'interno dei vasi. La piattaforma, se combinata con la modellazione al computer, rappresenta un passo significativo nello sviluppo di cure specifiche per il paziente per gli aneurismi sulla base della geometria dei vasi sanguigni, della pressione sanguigna e di altri fattori di un individuo e potrebbe aiutare a superare il problema del tempo necessario perché le nuove tecniche e tecnologie chirurgiche passino dal laboratorio alla clinica.

www.llnl.gov



Raspberry Pi 4

**UTILIZZALA
SUBITO CON
I NUOVI KIT**



Starter kit completo con Raspberry Pi 4 da 2 GB

Prezzi IVA inclusa.



€ 89,00

Cod. RPI4FULLG2

- » Board Raspberry Pi 4 modello B con 2GB di RAM.
- » Alimentatore 5 VDC / 3 A uscita type C e interruttore.
- » Contenitore plastico per Raspberry Pi 4 tipo B, colore bianco e rosso.
- » Cavo da HDMI (tipo A) a micro HDMI (tipo D), lunghezza 1,5 metri.
- » Micro SD card da 32GB con sistema operativo NOOBS precaricato.



Disponibile anche con Raspberry Pi 4 modello B da 4GB di RAM **COD. RPI4FULLG4 € 105,00**
e con Raspberry Pi 4 modello B da 8GB di RAM **COD. RPI4FULLG8 € 129,00**

Starter kit con Raspberry Pi 4 da 2 GB



€ 75,00

Cod. RPI4STARTERG2

- » Board Raspberry Pi 4 modello B con 2GB di RAM.
- » Alimentatore 5 VDC / 3 A uscita type C e interruttore.
- » Contenitore plastico per Raspberry Pi 4 tipo B, colore bianco e rosso.



Disponibile anche con Raspberry Pi 4 modello B da 4GB di RAM **COD. RPI4STARTERG4 € 89,00**
e con Raspberry Pi 4 modello B da 8GB di RAM **COD. RPI4STARTERG8 € 115,00**

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