

Elettronica In

www.elettronica.in.it

€ 6,90 - Anno XXVIII - n. 269 - NOVEMBRE 2022



Cloud Energy
Meter

Light
Control

Interruttore
astronomico

Protezione da
sovraccarico

Regolatore per
pannelli solari

SPECIALE

Risparmio energetico

MIN

Demoboard ESP32

**Calcolatrice con
tastiera analogica**

Comando luce
con Telegram

Regolatore
di velocità
per ventole

Misura i consumi
con Blebricks



Consumi energetici sotto controllo!

Tieni sotto controllo i consumi della tua abitazione e risparmi sulla bolletta!



Misuratore di consumo di energia elettrica

€ 22,50
cod. NETBSEM5

Permette di conoscere il consumo ed il costo dell'energia consumata dall'apparecchiatura collegata ad esso. Alimentazione 230 Vac; massimo carico applicabile 3,6 kW / 16 A.



Termostato Wi-Fi touch screen da incasso



€ 59,00
cod. FR695

Ideale per gestire facilmente il clima domestico (riscaldamento o climatizzazione) anche lontano da casa grazie all'App gratuita per Android e iOS, risparmiando così sui consumi energetici.



Termostato Wireless



€ 62,00
cod. TP384



Sistema per il controllo dell'impianto di riscaldamento o condizionamento, composto da un termostato wireless (868,35 MHz) con ampio display retroilluminato e da un'unità ricevente in grado di controllare un carico resistivo max. 16 A / 230 Vac o induttivo max. 4 A / 230 Vac.

DISPONIBILE ANCHE
NELLA VERSIONE
CON IL COLLEGAMENTO
VIA CAVO

€ 32,00
cod. TP385



Timer digitale programmabile su guida DIN

Timer digitale con display LCD per montaggio su guida DIN. Ideale per fornire o togliere l'alimentazione a vari generi di apparecchiature elettriche secondo il tempo desiderato e ad orari prestabiliti siano essi giornalieri o settimanali. Carico max. 3680 W, 16 A, alimentazione 230 Vac.

€ 17,90
cod. TP238



Timer digitale settimanale da esterno (IP44)

Consente di accendere e spegnere apparecchiature elettriche nei giorni e nelle ore della settimana prestabilite. Dispone di 10 programmi di funzionamento per ciascun giorno della settimana e di 16 diverse combinazioni giornaliere. Potenza massima applicabile 3600 W / 16 A.

€ 18,90
cod. WT51301



Prezzi IVA inclusa.



Lettore, attore del Digital Green

Molti di voi ci seguono dal lontano 1995, quando un gruppo di "ragazzi" appassionati, reduci da precedenti esperienze editoriali, ha lanciato il nostro mensile di idee e applicazioni elettroniche. Tanti anni sono passati e il mondo ha conosciuto continui cambiamenti e rivoluzioni tecnologiche che hanno modificato radicalmente il nostro modo di vivere. Ma la storia del mondo non si ferma e un altro stravolgimento mondiale è in atto proprio in questi anni. La crisi climatica con le sue drammatiche conseguenze non è più solo argomento di letteratura, ma parte della vita di ognuno di noi. Tutela dell'ambiente e

Tutela dell'ambiente e risparmio energetico sono temi che coinvolgono etica, morale e, un po' prosaicamente, l'economia.

risparmio energetico sono temi che coinvolgono etica, morale e, un po' prosaicamente, l'economia. La carenza di materie prime e i costi sempre più elevati dell'energia hanno ricordato anche ai negazionisti del degrado ambientale, che qualcosa è cambiato per le loro tasche.

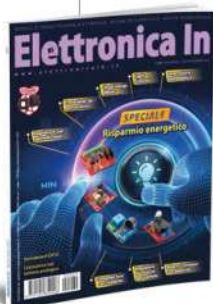
Sostenibilità è una parola che in futuro sarà sempre più al centro delle attività umane. Tra gli aumenti del 30/40/50% nell'ultimo anno c'è anche quello della carta, che sta rivoluzionando il mondo della stampa mettendo in crisi tutta la filiera, da chi la produce a chi la utilizza. E se da un lato potremo allietarci nel vedere sempre meno volantini incastrati nella buca delle lettere, dall'altro libri, riviste, giornali vedono inevitabilmente aumentare il prezzo di copertina come del resto è anche già successo a questa testata. Come mettersi al riparo da ulteriori aumenti? Un invito accorato è quello di unirsi ai numerosi abbonati: oltre a sostenere l'opera e il lavoro di progettisti, tecnici, grafici ecc. consente all'editore di approvvigionarsi in anticipo della carta, pianificando il consumo, ed evitando ulteriori aumenti del prezzo di copertina. E per chi si sente Digital Green, l'abbonamento digitale è un passo verso la sostenibilità ambientale e lo spirito ecologico del futuro.

ON LINE

elettronica.in

SI RINGRAZIA

Futura Elettronica Mouser



In copertina:
Otto progetti elettronici per aiutarci a risparmiare in questo periodo difficile per tutti.

Boris Landoni
 boris.landoni@elettronica.in



Elettronica In

Rivista mensile, anno XXVIII n. 269
NOVEMBRE 2022

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 49,00
Annuo 10 numeri digitale Euro 39,00
Le richieste di abbonamento vanno inviate a:
FUTURA GROUP srl
via Adige 11, 21013 Gallarate (VA)
 Tel. 0331-752668
oppure tramite il sito
<https://futuragroup.it/abbonamenti>

Distribuzione per l'Italia
S.O.D.I.P. Angelo Patuzzi S.p.A.
via Bettola 18 - 20092 Cinisello Balsamo (MI)
 Tel. 02-660301 Fax 02-66030320

Stampa

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

Tiratura
18.000 copie



Rivista mensile registrata presso il Tribunale di Milano
con il n. 245 il 03/05/1995.
Prezzo di copertina Euro 6,90.

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

Poste Italiane Spa - Spedizione in abbonamento Postale - D.L. 353/2003 (conv. in L. 27/02/2004) art. 1 comma 1 - DCB Milano.
Futura Group srl è iscritta al Registro Operatori della Comunicazione n. 23850 del 02/07/2013.

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

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

© 2022 Futura Group srl

Futura Group
Edizioni

Contenuti

01

RUBRICHE

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

Automazione

Interruttore astronomico



47

19

ARTICOLI

NETWORKING

19 Cloud Energy Meter

Contatore di energia e assorbimento basato sulla scheda di prototipazione ESP32, che ci consente di interfacciarci con il cloud.

HOME PROJECT

31 Protezione da sovraccarico AC

Tramite un sensore induttivo rileva la corrente in una linea elettrica e con un relé può attivare un avvisatore o sconnettere un carico se viene superata la soglia impostata mediante un trimmer.

DIDATTICA

37 Demoboard ESP32

Chiudiamo la panoramica sulla scheda di prototipazione basata sul popolare modulo WiFi, con ulteriori applicazioni. Terza puntata.

Progetto di Copertina

SPECIALE: RISPARMIO ENERGETICO



Networking Cloud Energy Meter

19



Didattica Demoboard ESP32

37



Home Project Protezione da sovraccarico AC



31

Lighting Light Control



57

Solare Regolatore per pannelli fotovoltaici



65

Misure Energy Meter con Blebricks



81

Home Project Calcolatrice con tastiera analogica



107

Networking Comando luce con Telegram



93

Automazione Regolatore velocità ventole



75

Didattica Programmiamo con RP2040



119

AUTOMAZIONE

47 Interruttore astronomico

Permette di attivare un impianto di illuminazione esterno in base agli orari di alba e tramonto calcolati automaticamente ogni giorno dell'anno.

LIGHTING

57 Light control

La luce della stanza te la spengo io! Un semplice ma efficace sistema di controllo per spegnere la luce.

SOLARE

65 Regolatore per pannelli FV

Stabilizza e ottimizza la carica di accumulatori alimentati da pannelli fotovoltaici a 12/24V, gestendo il carico.

AUTOMAZIONE

75 Regolatore velocità ventole

Adegua e stabilizza la velocità di rotazione di elettroventole in corrente continua in funzione della temperatura rilevata nell'area o sul dissipatore che deve essere raffreddato.

MISURE

81 Energy Meter con Blebricks

Utilizziamo la piattaforma di prototipazione a "mattoncini" per implementare un monitor di potenza assorbita ed energia consumata.

NETWORKING

93 Comando luce con Telegram

Controllo da smartphone di accensione e spegnimento di una lampada o di un carico elettrico, con timer di spegnimento automatico programmabile.

HOME PROJECT

107 Calcolatrice con tastiera analogica

Riscopriamo la classica calcolatrice con display a 7 segmenti dall'aspetto vintage. Seconda e ultima puntata.

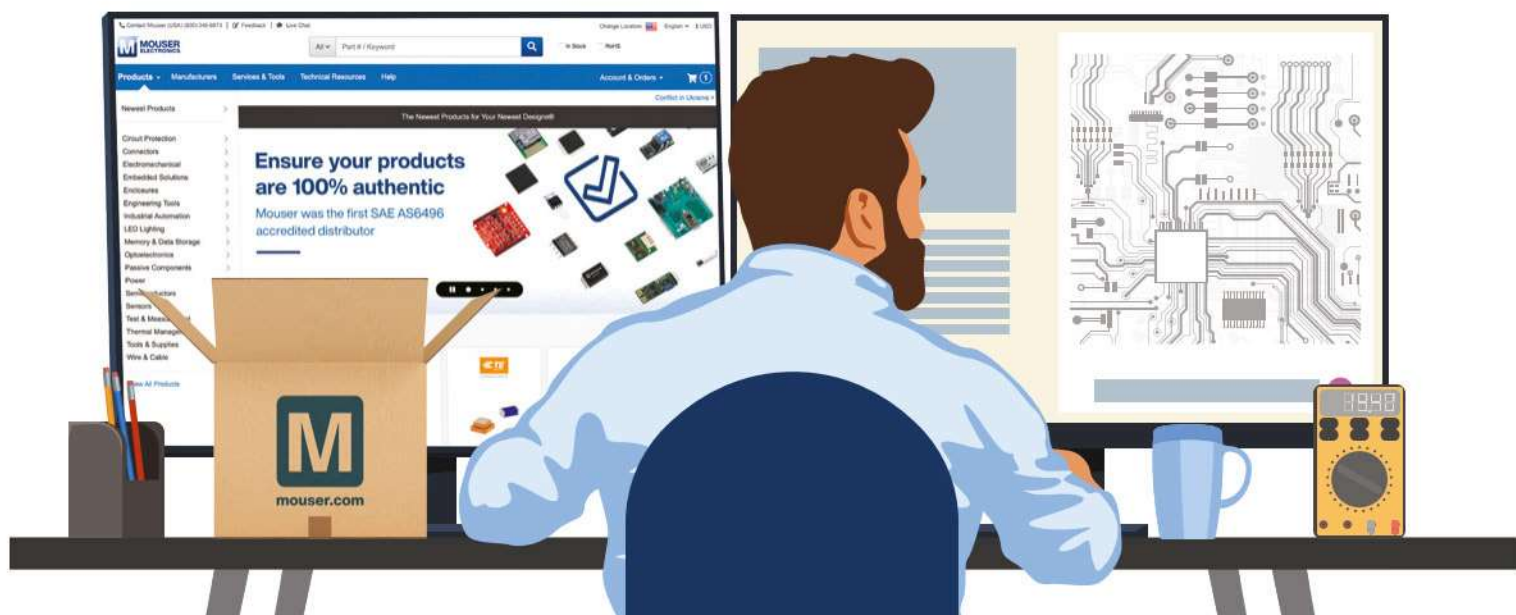
DIDATTICA

119 Programmiamo con RP2040

In questa puntata continuiamo a parlare di LoRa/LoRaWAN. Imparerete a sfruttare il cloud TTN per creare un dispositivo IoT con visualizzazione dei dati su Node-RED.

Tu progetti. Noi consegniamo

Prodotti d'avanguardia per progetti innovativi™



mouser.it/new



MOUSER
ELECTRONICS



Il plasma rivoluziona le antenne

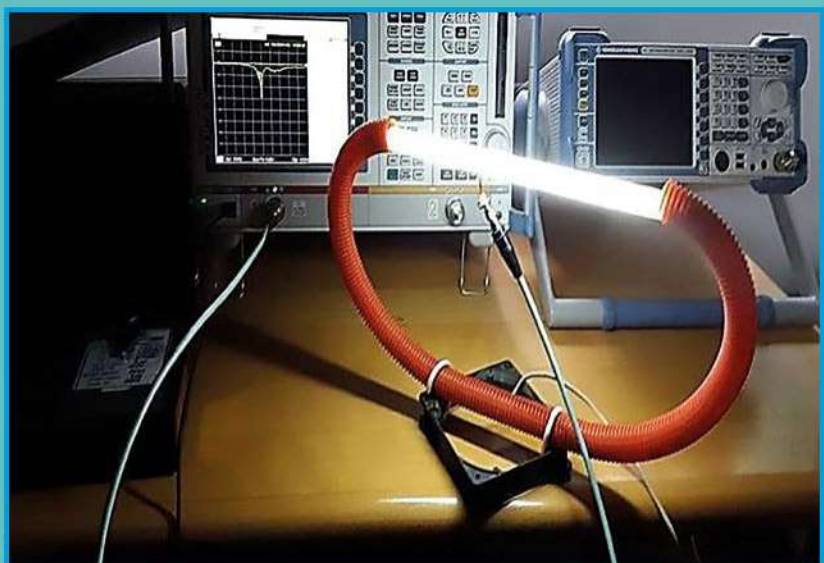
Quella al plasma è un tipo di antenna radio attualmente in fase di sviluppo, dove il plasma sostituisce gli elementi metallici che compongono le antenne tradizionali. Un esempio di antenna al plasma è l'antenna al gas plasma, che utilizza un tubo di scarico anziché elementi metallici. Mentre la corrente scorre attraverso il tubo, il gas viene parzialmente o completamente ionizzato al plasma, diventa conduttivo e funge da specchio, trasmettendo e ricevendo segnali RF. Al di sopra della frequenza del plasma, le antenne al plasma risultano virtualmente trasparenti per una grande larghezza di banda di onde elettromagnetiche, diventando invisibili quando l'impianto viene disattivato e il gas si deionizza. A differenza dei modelli di antenna tradizionali, è possibile riconfigurare elettricamente, piuttosto che meccanicamente, i modelli basati sul plasma per quanto riguarda impedenza, frequenza, larghezza di banda e direttività su scale temporali che spaziano dal microsecondo al millisecondo. Inoltre, è possibile impilare serie di antenne al gas plasma affinché funzionino a frequenze diverse. Il plasma possiede un'elevata conduttività elettrica, che contribuisce alla ricezione, alla direzione e alla trasmissione di vari tipi di segnali radio.

La tecnologia vanta numerosi vantaggi; tuttavia, sussiste sempre un margine di miglioramento relativo alle prestazioni, ed è proprio questo l'obiettivo conseguito dai ricercatori durante il progetto PATH, finanziato dall'UE. Il miglioramento delle prestazioni della sorgente di plasma richiede notevoli sforzi in termini di modellizzazione e tecnologia. Il superamento del limite della densità dell'attuale sorgente di plasma spia-

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

nerà la strada a nuove applicazioni in diversi ambiti tecnologici. I ricercatori hanno inizialmente sviluppato due tipi di sorgente di plasma: una compatta a radiofrequenza e un'altra a catodo cavo, che prevede elevate densità di plasma e antenne con un ciclo di vita più lungo. I modelli fisici dettagliati di entrambe le sorgenti di plasma illustrano le interazioni tra il plasma

e la radiazione elettromagnetica. Finora, varie attività di ricerca si sono incentrate sull'ottimizzazione delle sorgenti di plasma dal punto di vista dell'efficienza energetica, della riconfigurabilità, della dimensione, della densità o del ciclo di vita. Malgrado il successo delle sperimentazioni, non è stato possibile compiere progressi su tutti i fronti, perché occorre svolgere ulteriori ricerche, in particolare nell'ambito dei materiali al



↑ Nell'antenna al plasma, gli elementi metallici sono attivi e composti da plasma.

fine di raggiungere un livello di maturità tecnologica appropriato. In questo senso il progetto di ricerca europeo PATH (www.path-project.eu) ha gettato le basi per una nuova classe di tecnologie che coniuga le migliori proprietà della radiofrequenza e delle sorgenti di plasma a catodo cavo. Il modello di sorgente ibrida ha concorso all'aumento della densità del plasma, superando 1020 ioni per metro cubo e migliorando così le prestazioni dell'antenna. Grazie all'unione di competenze provenienti da settori diversi, nell'ambito del progetto PATH è stato scoperto il modo di sfruttare questo concetto di tecnologia ibrida nelle applicazioni delle telecomunicazioni e di navigazione. I risultati sperimentali dei prototipi di nuova concezione hanno dimostrato che il comportamento delle antenne al plasma attive assomiglia molto a quello delle antenne in metallo. La differenza più evidente quando si confrontano questi due tipi di antenne risiede nel guadagno, che risulta minore nelle antenne al plasma; quando gli elementi dell'antenna non sono attivi, non interferiscono con i campi elettrici attivi degli elementi delle antenne nelle vicinanze. In altre parole, quando sono spente, diventano trasparenti alle onde elettromagnetiche emesse da altri elementi. Il guadagno complessivo delle serie di antenne al plasma si è inoltre rivelato invariante rispetto alla scelta dell'elemento attivo dell'antenna, a condizione che la geometria degli elementi attivi rimanesse la stessa. Al contrario, ogni elemento dell'antenna nelle serie di antenne in metallo inciderebbe sul guadagno del canale in maniera differente.

I vantaggi offerti dalle antenne al plasma rispetto a quelle tradizionali in metallo sono soprattutto evidenti in applicazioni militari, in cui l'invisibilità e la guerra elettronica costituiscono le preoccupazioni principali. Inoltre è prevedibile che la tecnologia si affermerà

nel settore degli smartphone e di altri dispositivi dell'elettronica di consumo, nelle comunicazioni indossabili senza fili e nei sistemi biomedici a radiofrequenza.

Il diodo protegge ma non si vede

Quando bisogna proteggere un circuito dall'inversione di polarità, tipicamente si fa ricorso a due soluzioni, entrambe basate su diodi raddrizzatori: la prima consiste nel porre il componente in serie al ramo di alimentazione con l'anodo rivolto al positivo; questo implica che il diodo dovrà sopportare una corrente pari almeno a quella massima assorbita dal circuito da proteggere, ma anche la perdita di una potenza pari al prodotto della corrente per la caduta di tensione. Per esempio in un circuito che assorbe 5A con un diodo su cui a tale corrente cadono 0,8V, la perdita è di ben 4 watt. Ecco perché la soluzione a diodo serie è ideale per i circuiti a basso assorbimento. Per ovviare a questi problemi è possibile scegliere la seconda soluzione, che consiste nel porre il diodo in antiparallelo sulla linea di alimentazione, ovvero collegato col catodo al positivo e l'anodo al negativo; in questo modo, in caso di inversione della polarità il diodo mette la linea in cortocircuito; chiaro che a parte certe situazioni particolari, per proteggere la fonte di alimentazione occorre porre un fusibile a monte del diodo, meglio se è di tipo autoripristinante (ossia un classico Polyswitch). Anche in questo caso il diodo deve sopportare la massima corrente prevista nella linea.

Esiste in realtà una terza soluzione per proteggere un circuito dall'inversione di polarità, efficace ed energeticamente conveniente perché consiste nel commutare la linea di alimentazione tramite lo scambio di un relé, che abbia la portata pari almeno alla corrente assorbita dal circuito da proteggere: in pratica si utilizza un diodo in serie alla bobina del relé, collegato con l'anodo verso il polo positivo, secondo lo schema proposto in questa pagina. Il circuito implementa una protezione a "caduta zero" perché la corrente attraversa lo scambio del relé, il quale ha una resistenza d'inserzione praticamente nulla.

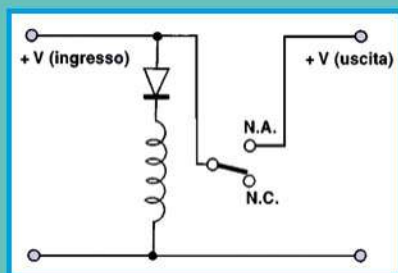
Quando la polarità dell'alimentazione è corretta, il diodo alimenta la bobina del relé e lo scambio la porta al resto

del circuito; se invece la tensione viene invertita, il relé non viene eccitato e la corrente non passa verso il circuito, che in tal modo viene protetto. Per il corretto funzionamento, il comune (C) ed il contatto normalmente aperto (N.A.) del relé devono interrompere la linea di alimentazione, mentre la bobina e il diodo vanno collegati in parallelo all'alimentazione. In questo modo, solo rispettando la polarità della tensione di alimentazione sarà possibile chiudere il relé e, nel contempo, dare alimentazione al carico senza nessuna caduta di tensione. Il relé va scelto con uno scambio capace di commutare la tensione e la corrente del circuito in questione.

Fotovoltaico... da polso

In tempi in cui si parla sempre più frequentemente di utilizzo delle rinnovabili non si possono trascurare le innovazioni nell'ambito dell'elettronica di consumo, come ad esempio nella tecnica degli orologi, dove spicca la soluzione Eco-Drive della Citizen; questa va incontro all'esigenza di coniugare il vantaggio offerto dagli orologi automatici di dimenticarsi della ricarica e la precisione dei movimenti elettronici al quarzo. In vero il problema è stato affrontato dall'industria giapponese in due modi: con il Kinetic della Seiko, in cui il volano aziona un generatore di corrente invece degli ingranaggi della molla di carica e con il fotovoltaico, tradotto nell'Eco-Drive Citizen.

Quest'ultimo sistema è un micro-fotovoltaico inserito nella cassa di un orologio, per fornire alimentazione perpetua all'elettronica che aziona il movimento. L'idea risale al 1973, quando le celle solari erano ancora una tecnologia innovativa. I primi orologi alimentati dalla luce avevano una cella solare posta sul quadrante che convertiva la luce in energia elettrica, a sua volta fornita di un circuito integrato quarzato che pilota il motore. Negli anni '80 anche altre aziende cominciarono a produrre orologi fotovoltaici ma Citizen aveva molti anni di vantaggio e la sua tecnologia è sempre stata più efficiente di quella di chiunque altro. Ad esempio, mentre gli orologi solari delle altre marche avevano una riserva di carica media di 2 giorni, Citizen - a metà anni '80 - presentò il modello Analogic Solar Cell con una riserva di carica quattro volte superiore grazie alla notevole effi-



↑ Come proteggere un circuito dall'inversione di polarità senza perdite nel diodo.



↑ Struttura di un orologio al quarzo Eco-Drive, ad alimentazione fotovoltaica.

cienza energetica che lo caratterizzava: necessitava un'ora e mezza di ricarica (un terzo rispetto al tempo necessario per i prodotti della concorrenza) per un'intera giornata di autonomia. Inoltre i progressi raggiunti nella tecnologia a carica solare permettevano la ricarica non solo con la luce del sole, ma anche con la luce artificiale. Negli anni '90 Citizen sostituì la tradizionale pila con un accumulatore, capace di fornire a carica completa un'autonomia di 6 mesi: questa scelta puntava anche a ridurre i rifiuti elettrici ed elettronici. Tipicamente un orologio Eco-Drive è composto da un sandwich in cui si trova, dall'alto sotto il vetro, il quadrante, realizzato in modo da far passare la luce attraverso di esso e raggiungere la cella solare che è posizionata nella parte sottostante, dal circuito elettronico che contiene il microprocessore che gestisce le funzioni e l'energia nella funzione di ricarica. In fondo si trova il

movimento, basato su micro-motore elettrico che aziona le lancette. Il miglioramento della tecnologia ha ridotto il fabbisogno di elettricità, consentendo di adottare quadranti con trasparenza dall'iniziale 80% (a scapito dell'estetica) ad appena il 20%.

Package per i transistor SMD

La componentistica SMD ha visto comparire nel tempo numerosi formati di package; tra questi, spiccano quelli dei transistor, che sono essenzialmente:

- SOT/TSOT;
- TO-251 (anche noto come IPAK); è un SMD con aletta di raffreddamento (tab) corta a saldare e tre terminali dritti;
- TO-252 (chiamato DPAK); è come il precedente, sempre con l'aletta a saldare e due terminali esterni ad ali di gabbiano (il centrale è cortissimo e risulta collegato all'aletta);
- TO-262 (noto anche come I²PAK); è un contenitore SMD con aletta a saldare e pin dritti, in numero variabile da 3 a 7;
- TO-263 (ovvero D²PAK); è come il precedente, però ha i terminali ad ali di gabbiano e nella versione a tre terminali è senza elettrodo centrale, che elettricamente è connesso all'aletta;
- TO-273; chiamato anche super-220 (perché è l'equivalente di un TO220 ma per montaggio superficiale) ha 2 terminali ad ali di gabbiano ed il centrale troncato;
- TO-274; anche detto super-247, è la versione SMD del TO-247, senza foro di montaggio e con tre terminali.

La serie SOT o TSOT comprende package a tre terminali disposti a triangolo, vale a dire uno da un lato e due da quello opposto; il contatto singolo è il collettore nei BJT e il drain nei MOSFET.

Si utilizzano anche per i doppi diodi a catodo comune (in questo caso gli anodi fanno capo ai due piedini di un lato e il catodo è l'unico sul lato opposto).

I piedini di questo package sono sempre piegati ad ali di gabbiano e piatti nel punto di appoggio alle piazzole del circuito stampato.

Un particolare contenitore è l'SOT, che esiste in molteplici varianti come la SOT-23, la quale in realtà può avere più di tre piedini: di solito fino a 6; ad esempio sono 5 in molti regolatori di tensione lineari LDO, con due piedini da un lato e tre dall'altro.

In alcuni casi il terminale centrale dei transistor o dei regolatori a tre terminali si prolunga sotto il corpo e va saldato in tutta la sua superficie, al pad corrispondente del circuito stampato, la cui sagoma sarà adeguata al componente da ospitare.

Alcuni package elencati sono utilizzati non solo per i componenti attivi discreti (tipicamente transistor bipolari e ad effetto di campo, soprattutto MOSFET) ma anche per gli integrati.

Il TO-262 (o D²PAK) viene utilizzato anche per circuiti integrati come i regolatori di tensione lineari e switching a più di tre terminali, ovvero per i MOSFET con circuito di protezione e sensing di corrente incorporato.

I package TOx prevedono un'aletta metallica che si prolunga sotto il corpo e che dev'essere saldata al PCB insieme all'aletta e ai terminali; in realtà non è sempre così, perché spesso questo tab metallico non viene saldato.



↑ Package per componenti a semiconduttore tipo TO-262 (a sinistra), TO-263 (in centro) e TO-274 (a destra).

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.elettronicaonline.it/webforum

Fai il pieno di energia con le batterie ai polimeri di litio!



€ 27,90



COD. PLIB6000

3,7V-6000 mAh.
Connettore 2 mm

€ 6,90



COD. PLIB1000-2MM

3,7V-1000 mAh.
Connettore 2 mm

€ 7,90



COD. PLIB1100-2MM

3,7V-1100 mAh
connettore 2 mm

€ 21,00



COD. 25C2000MAH7V4

7,4 V-2000 mAh.
Connettori JST-XH
Deans T

€ 34,00



COD. 25C2200MAH11V1

11,1 V-2200 mAh.
Connettori JST-XH

€ 4,90



COD. PLIB500NEW

3,7 V-500 mAh.
Connettore 2,54 mm

€ 6,90



COD. PLIB1000-1.25

3,7V-1000 mAh.
Connettore 1,25mm

€ 21,90



COD. PLIB5000MAH

3,7 V-5000 mAh.
Connettore 2 mm

€ 7,90



COD. PLIB1100

3,7V-1100 mAh.
Connettore 1,25mm

€ 21,90



COD. PLIB5000-2.54

3,7 V-5000 mAh.
Connettore 2,54 mm

€ 12,50



COD. PLIB430

3,7 V-500 mAh con
controllo temperatura.
Connettore JST 3 fili

€ 12,90



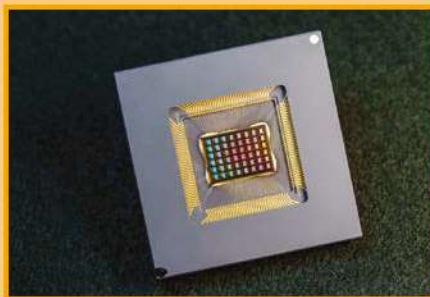
COD. PLIB2000

3,7 V-2000 mAh.
Connettore 2,54 mm

Prezzi IVA inclusa.

Replicare il cervello umano in software e silicio è da tempo lo scopo della ricerca sull'intelligenza artificiale. I chip neuromorfici hanno fatto ampi progressi nel calcolo simultaneo e possono sia calcolare che archiviare dati, ma sono lenti dall'efficienza energetica del cervello umano, perché il calcolo AI richiede molta energia, la maggior parte della quale è nel trasferimento dei dati tra la memoria e le unità di calcolo all'interno dei chip. Per affrontare questo problema, un team di ricercatori ha sviluppato un chip CIM (Compute-in-Memory) che ha 2 volte l'efficienza delle piattaforme di intelligenza artificiale esistenti. Chiamato NeuRRAM, perché utilizza un tipo di RAM chiamato memoria ad accesso casuale resistivo (RRAM), questo hardware RRAM-CIM ha 48 core. La RRAM ha molti vantaggi rispetto alla memoria convenzionale: ha

➤ RRAM: l'AI due volte più efficiente



più capacità a parità di area di silicio (il che consente di implementare modelli di intelligenza artificiale più grandi) ed è non volatile, il che significa che non c'è dispersione di potenza. Ciò rende i chip basati su RRAM ideali per i carichi di lavoro edge. L'innovazione principale è un nuovo schema di conversione A/D, basato sul rilevamento della tensione invece della corrente, consentendo maggiore parallelismo nell'array RRAM in un singolo ciclo di elaborazione. NeuRRAM ha ottenuto un'accuratezza del 99% su un'attività di riconoscimento delle cifre scritte a mano, dell'85,7% sulla classificazione delle immagini, dell'84,7% sul riconoscimento dei comandi vocali di Google e una riduzione del 70% dell'errore di ricostruzione dell'immagine.

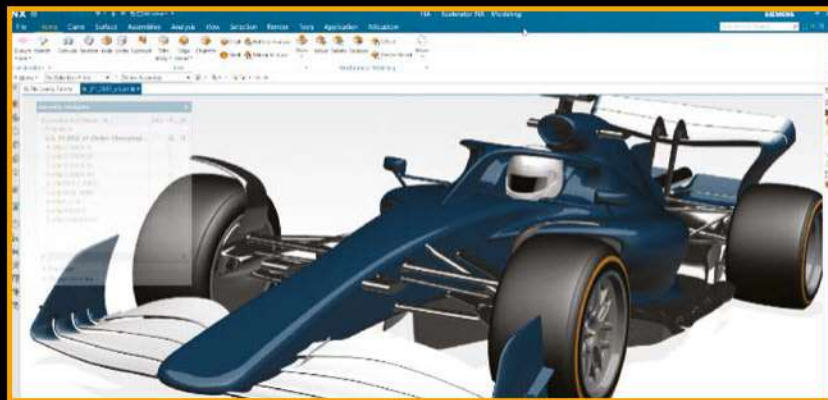
ucsdnews.ucsd.edu

➤ FIA e Siemens Xcelerator per auto più sostenibili

L'ente di governo mondiale degli sport motoristici, la FIA, ha selezionato Siemens Digital Industries Software come partner per aiutarla a migliorare la sostenibilità in F1 e in altri campionati di corse, ossia nella progettazione e nello sviluppo di auto da corsa più ecologiche con consumi energetici ed emissioni ridotti.

Il motorsport e la sostenibilità suonano come una contraddizione, ma con le giuste tecnologie è possibile perseguire l'obiettivo che da tempo si insegue, adottando soluzioni di recupero dell'energia in frenata e di ibridazione delle auto da corsa. Se il motorsport raggiunge la neutralità del carbonio, qualsiasi settore può farlo.

Dopo aver raggiunto la carbon neutrality e aver ottenuto la certificazione ISO 14001:2015 nel 2021, la FIA continua a esplorare nuove aree di sviluppo. La FIA inizierà a implementare il portafoglio Siemens Xcelerator per la progettazione di veicoli e normative che riducano il consumo di energia e



le emissioni. Attraverso questa collaborazione, Siemens supporterà la FIA nel raggiungimento dei suoi obiettivi di sostenibilità attraverso un pensiero innovativo, approcci pionieristici nello sport automobilistico e nuovi progressi tecnologici che potrebbero avere un impatto sull'industria automobilistica. Già nel 2019, la FIA e Siemens hanno collaborato per migliorare la sicurezza

nei rally esplorando come la tecnologia di rilevamento dei pedoni dei veicoli autonomi potrebbe essere utilizzata per identificare i luoghi pericolosi per gli spettatori dei rally. Poi, nel 2020, Siemens ha aiutato i club membri della FIA a fornire un approccio basato sull'evidenza per sostenere l'attuazione delle politiche di mobilità.

www.plm.automation.siemens.com

› Addestrare grandi reti neurali su piccoli dispositivi

La messa a punto dei modelli su dispositivi edge come gli smartphone consentirebbe la personalizzazione preservando la privacy sui dati sensibili. Tuttavia, l'edge training è stato storicamente limitato a modelli relativamente piccoli con architetture semplici, perché l'addestramento richiede sia memoria che energia, entrambi critici nei dispositivi mobile. Ma ora i ricercatori guidati dal prof. ShishirPatil dell'Università della California, hanno creato POET, un algoritmo capace di abilitare l'addestramento di reti neurali (NN) di grandi dimensioni su dispositivi edge alimentati a batteria e dotati di poca memoria. POET ottimizza congiuntamente gli spazi di

ricerca integrati di rimaterializzazione e paging, due algoritmi per ridurre il consumo di memoria della backpropagation. Dato un budget di memoria e un vincolo di runtime, è stato formulato un programma lineare misto intero (MILP) per l'addestramento ottimale dal punto di vista energetico. Tale approccio consente di addestrare modelli significativamente più grandi su dispositivi embedded, riducendo il consumo di energia senza modificare la correttezza matematica della backpropagation. POET è un progetto open source e dimostra che è possibile mettere a punto sia ResNet-18 che BERT entro i vincoli di memoria di un dispositivo embedded di classe

› Telecamera 5G AIoT di Milesight: 5G, IoT e LoRaWAN si incontrano



L'implementazione su larga scala della rete 5G rafforza l'ecosistema Internet of Things (IoT), estendendo gli scenari applicativi a svariati settori.

Milesight, uno dei partner dell'ecosistema LoRa di Semtech, ha lanciato la telecamera di rete 5G AIoT, che non solo integra in modo innovativo le tecnologie 5G, AI e IoT, ma può anche supportare contemporaneamente 5G e connettività di rete tramite LoRaWAN. La telecamera di Milesight incorpora un modulo gateway che sfrutta la connettività di rete LoRaWAN. Senza la

necessità di installare cavi complessi e fibre ottiche, può realizzare una trasmissione illimitata di dati video attraverso una rete 5G. La telecamera di rete adotta il Semtech LoRa Core SX1302 ad alte prestazioni, che vanta un'elevata capacità di elaborazione dei dati e un funzionamento stabile. Attraverso l'integrazione innovativa delle tecnologie 5G, AI e IoT, la Network Camera facilita il monitoraggio avanzato per un'ampia varietà di casi d'uso.

La telecamera di rete 5G AIoT HD può realizzare duplici applicazioni basate

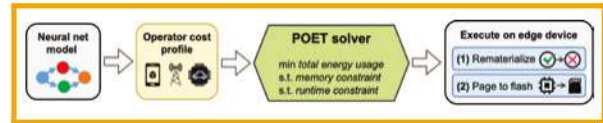
su un'unica macchina: ad esempio, può essere utilizzata come telecamera di rete HD per trasmettere video ad alta definizione e come gateway di rete che utilizza LoRaWAN per raccogliere i dati di sensori IoT corrispondenti. Inoltre, se i dati del sensore attivano un avviso preventivo, il sistema può effettuare una successiva conferma basata sul collegamento sincrono di immagini video correlate.

Il modulo gateway abilitato per LoRa supporta la raccolta dei dati dei sensori, mentre il modulo 5G può essere utilizzato per lo streaming video live, l'analisi rapida e il processo decisionale per realizzare l'implementazione e la trasmissione a tantum del flusso di dati e del flusso video, riducendo i costi di equipaggiamento e implementazione. Grazie alla tecnologia AI, è in grado di realizzare funzioni di analisi dei contenuti video diversificate, come intrusioni, rilevamento avanzato del movimento, o ingresso in zone, attraversamento di linee, rilevamento manomissioni, ecc. La telecamera di rete supporta più bande di frequenza tra cui CN470, RU864, IN865, EU868, US915, AU915, KR920, AS923, AS923-2 e modalità di connessione MQTT / HTTP / HTTPS.

www.semtech.com

Cortex-M, superando al contempo gli attuali metodi di training edge in termini di efficienza energetica.

I più potenti sistemi di Machine Learning utilizzano NN, funzioni complesse piene di parametri sintonizzabili. Durante l'addestramento, un network riceve un input, come un insieme di pixel, genera un output, lo confronta con la risposta corretta e regola i suoi parametri per fare meglio la prossima volta. Per sapere come calibrare questi parametri è necessario ricordare l'effetto di ciascuno, anche in miliardi di combinazioni, pertanto l'addestramento di una rete neurale può richiedere centinaia di volte la memoria del semplice utilizzo, dove è consentito dimenticare ciò che ha fatto ogni livello della rete non appena passa le informazioni al successivo. POET (Private Optimal Energy Training) acquisisce i dettagli tecnici di un dispositivo e l'architettura di una rete neurale da addestrare, specifica un budget di memo-



ria e uno di tempo, quindi gli chiede di creare un processo di formazione che riduca al minimo il consumo di energia. Potrebbe decidere di eseguire il paging di determinate attivazioni che sarebbero inefficienti da ricalcolare e di rimaterializzare altre che sono semplici da ripetere ma richiedono molta memoria. POET è stato testato su quattro diversi dispositivi, addestrando tre diverse architetture di NN: due di riconoscimento d'immagine (VGG16 e ResNet-18) e una rete di elaborazione del linguaggio (BERT).

poet.cs.berkeley.edu

› La GPU Hopper Nvidia mostra i muscoli all'AI

Nvidia strabilia ancora nei test di intelligenza artificiale con la sua GPU Hopper, dalle prestazioni fino a 4,5 volte superiori a quelle della precedente, conquistando la vetta nell'MLPerf Inference v2.1. MLPerf è un insieme di benchmark concordati dai membri del gruppo industriale MLCommons e rappresenta il primo tentativo di fornire confronti peer-to-peer utili a valutare la capacità dei computer nell'addestramento e nell'esecuzione (inferenza) delle reti neurali. Nei benchmark di inferenza di MLPerf, i sistemi costituiti

da combinazioni di CPU e GPU o altri chip acceleratori vengono testati su un massimo di sei reti neurali che svolgono una varietà di funzioni comuni: classificazione delle immagini, rilevamento di oggetti, riconoscimento vocale, imaging medico 3D, linguaggio naturale elaborazione e raccomandazione. Le reti erano già state addestrate su una serie standard di dati e hanno dovuto fare previsioni su dati a cui non erano state esposte prima. I computer testati sono classificati come destinati ai data center o "the

edge". I sistemi basati su data center disponibili in commercio sono stati testati in due condizioni: una simulazione dell'attività del data center reale in cui le query arrivano a raffica e un'attività "offline" in cui tutti i dati sono disponibili contemporaneamente.

I computer destinati a funzionare in loco anziché nel data center, quelli che MLPerf chiama edge, perché si trovano ai margini della rete, sono stati misurati nello stato offline; come se ricevessero un unico flusso di dati, ad esempio da una telecamera di sicurezza e come se dovessero gestire più flussi di dati, come farebbe un'auto con diverse telecamere e sensori.

Il test di performance è stato ulteriormente suddiviso in una categoria "chiusa", in cui tutti dovevano eseguire le stesse reti neurali "matematicamente equivalenti" e soddisfare le stesse misure di accuratezza, e una categoria "aperta", in cui le aziende potevano mostrare come con le modifiche allo standard neurale le reti fanno funzionare meglio i loro sistemi. Nella gara con i computer più potenti nelle condizioni più severe, il gruppo di data center chiusi è stato confrontato con chip di accelerazione AI.

nvidianews.nvidia.com





SUPPORTA
MICRO SD CARD
MAX 128 GB

cod. FR830

€ 38,00

**PENNA CON TELECAMERA
AD ALTA RISOLUZIONE**

Elegante penna dotata di telecamera nascosta in grado di effettuare riprese audio-video e di scattare fotografie in alta qualità. Alimentazione tramite batteria interna Li-Ion ricaricabile.



SUPPORTA
MICRO SD CARD
MAX 128 GB

cod. FR825

€ 80,00

**TELECAMERA A BOTTONE
CON TELECOMANDO**

Indossabile, di piccole dimensioni, facilmente occultabile con obiettivo a forma di bottone. Alimentazione tramite batteria interna Li-Ion ricaricabile.

Non farti sfuggire nulla. Riprendi ogni attimo con le mini telecamere.



Full HD
1080

WiFi

SUPPORTA
MICRO SD CARD
MAX 128 GB

HD
1080P HD
WiFi
Motion Detection
Up to 128GB
Built-in Mic
Night vision
Cloud Storage
Support Multi-view

cod. FR806

€ 79,00

**TELECAMERA WI-FI
IN LAMPADA A LED**

Telecamera Wi-Fi da 2 Megapixel nascosta in una lampada con 21 LED a luce bianca, Motion Detection, sensore crepuscolare, illuminatore con 8 LED IR con 15 m di portata.



SUPPORTA
MICRO SD CARD
MAX 32 GB

Full HD
1080

360°

90°

cod. CP756

€ 19,90

**MINI TELECAMERA
23 x 23 x 23 mm**

Registra video, scatta foto con una risoluzione fino a 12 Megapixel. Può essere utilizzata anche come Webcam. Alimentazione tramite batteria interna Li-Ion ricaricabile.

Prezzi IVA inclusa

**FUTURA
ELETTRONICA®**

www.futurashop.it

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

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

Nuovi relé DC compatti per automotive da Omron

I nuovi relé Omron G8K, G8G e G8PM sono orientati al controllo dei motori in continua, con ingombri fino al 95% inferiori rispetto ai relé simili attualmente sul mercato. I dispositivi sono tutti più piccoli dei relé DA standard e forniscono un'elevata pressione di contatto, grazie alla dimensione della bobina e a una capacità di isolamento di 500Vca.

I dispositivi della serie G8K sono rivolti alla commutazione di breve durata in applicazioni legate, ad esempio, al comfort automotive (come l'elettronica di controllo dei sedili, dei motorini delle serrature e dei motori per tettucci apribili). Una versione opzionale doppia permette di aumentare ulteriormente la densità di montaggio. La portata di corrente impulsiva è fino a 37A e quella continua fino a 5A, con una durata di 100.000 operazioni a carico nominale. Sostituti diretti dei relé plug-in micro-ISO, i prodotti serie G8G sono adatti



all'uso nei vani motore grazie alla loro elevata resistenza al calore; sono offerti in versione a 24V, proponendosi come unico prodotto sul mercato in grado di sostituire dai relé P-in a quelli per PCB.

Ultima ma non meno importante è la serie G8PM, ideale sostituta dei tradizionali relé mini ISO, con durata fino a 1 milione di operazioni.

components.omron.com

Optoisolatore per gate driver intelligenti con ripristino automatico da Toshiba

Toshiba presenta il primo fotoaccoppiatore per gate driver smart per MOSFET e IGBT, con ripristino automatico: il TLP5222 presenta una capacità di corrente in uscita di $\pm 2,5A$ e offre la protezione integrata dalle sovracorrenti con recupero automatico. Il TLP5222 monitora continuamente la tensione drain-source del MOSFET (VDS) o la tensione al collettore-emettitore dell'IGBT (VCE). In caso di sovracorrente, il circuito di protezione integrato del driver rileva l'aumento associato della VDS o della VCE e spegne gradualmente il MOSFET o l'IGBT. La funzione di ripristino automatico del driver riprende il normale funzionamento, in genere 25,5 μs dopo l'attivazione dello spegnimento di protezione, e ciò contribuisce a semplificare la progettazio-



ne del controllore dello stadio di potenza. Un pin isolato per lo stato di FAULT indica il rilevamento delle sovracorrenti anomale. Inoltre, il TLP5222 presenta un blocco di sottotensione (UVLO) e una funzione integrata di Miller clamp attivo che sopprime l'accensione indotta da dV/dt per prevenire il cortocircuito dei lati superiore e inferiore di un semiponte. Il TLP5222 è alloggiato in un package SO16L che garantisce distanze di dispersione di almeno 8 mm.

toshiba.com

Da TDK, nuovi DC-DC converter da 250W non isolati

TDK Corporation propone convertitori rinforzati DC-DC non isolati della serie TDK-Lambda RGA da 250W. Capaci di funzionare con una tensione di ingresso da 9 a 40V o da 9 a 53V, i convertitori step-down forniscono tensioni di uscita regolabili da 3,3 a 15V, da 3,3 a 24V o da 3,3 a 40V, con correnti fino a 20A. Con un fattore di forma di 1/16 brick (35,6x25,6x13mm), i convertitori sono incapsulati in alluminio pressofuso a cinque lati con linguette di montaggio, il che riduce il livello delle interferenze elettromagnetiche irradiate e consente il raffreddamento a conduzione per un funzionamento senza ventola. I converter supportano il funzionamento con fonti di alimentazione a 12V, 18V, 24V, 36V e 48V. Hanno efficienze fino al 98% e temperatura

del contenitore compresa tra -40°C e +110°C. Le funzionalità includono un pin di regolazione della tensione di uscita, una logica negativa on-off remota, rilevamento remoto, UVLO, protezione da sovracorrente e termica, Power Good, sequenziamento della tensione di uscita ecc. Tutti modelli sono certificati CE e UKCA per le Direttive Bassa Tensione nonché RoHS, e sono progettati secondo gli standard di sicurezza IEC/UL/CSA/EN 62368-1.

www.emea.lambda.tdk.com/it



Sensore CMOS 4,6 K e 120 fps da ON



ONsemi ha sviluppato il sensore CMOS custom per la fotocamera ALEXA 35 di ARRI; il sensore CMOS ALEV 4 Super 35 4,6K ha una risoluzione di 4608x3164 pixel (14,6 Mpixel) con un passo di 6,075 µm con filtri colorati RGB e un frame rate di 120 fps. È dotato dell'ultima tecnologia ON per l'ottimizzazione della ripo-

sta dei pixel, le tecniche High Dynamic Range (HDR) e il miglioramento dell'uniformità dei pixel durante la lettura ad alta velocità.

L'esclusivo sensore di immagine ON a pixel di grandi dimensioni ad alta risoluzione con tecniche di lettura ottimizzate si traduce in un'immagine di 4,6K, 17 stop di gamma di-

namica a 120 fotogrammi al secondo, producendo immagini con una qualità senza precedenti. La gamma dinamica estesa consente dettagli più elevati in una scena ad alto contrasto in un unico fotogramma, risultando un'immagine e un'esperienza cromatica straordinarie.

www.onsemi.com

SoM RM500Q 5G NVIDIA Jetson AGX Orin da Quectel

Quectel Wireless Solutions ha annunciato che i suoi moduli 5G avanzati, che connettono in modo sicuro e affidabile dispositivi e persone a reti e servizi, sono ora disponibili con il sistema su modulo (SOM) NVIDIA Jetson AGX Orin da 32 GB. Con la sua piattaforma Jetson edge AI, NVIDIA sta accelerando i progressi in robotica, automazione e intelligenza artificiale per l'IoT. Le funzionalità scalabili ed efficienti dal punto di vista energetico, includono modelli di intelligenza artificiale pre-addestrati, kit di sviluppo software e supporto per tecnologie cloud native. Quectel aggiunge i moduli 5G all'ecosistema Jetson, semplificando la gestione e la realizzazione di implementazioni in città intelligenti, produzione, IoT, applicazioni industriali e molte altre attività e processi connessi. La piattaforma Jetson AGX Orin 32GB SOM di NVIDIA è stata integrata con il modulo Quectel RM500Q serie 5G sub-6GHz,



che supporta più pipeline di applicazioni IA simultanee utilizzando una GPU con architettura NVIDIA Ampere. Con il deep learning, gli acceleratori di visione, l'I/O ad alta velocità e la larghezza di banda della memoria veloce, Orin consente a modelli di intelligenza artificiale complessi di risolvere problemi come la comprensione del linguaggio naturale, la

percezione 3D e la fusione multisensore. L'Orin SOM offre fino a 275 TOP di prestazioni AI con potenza configurabile tra 15 W e 60 W. Ciò fornisce oltre otto volte le prestazioni di Jetson AGX Xavier nello stesso fattore di forma, il che rende Orin ideale per la robotica e altri casi d'uso di macchine autonome.

www.quectel.com

PIC32CM JH: cybersecurity e compatibilità AUTOSAR da Microchip

Per soddisfare gli standard di ingegneria della sicurezza informatica ISO 26262 e ISO/SAE 21434, Microchip annuncia PIC32CM JH, ad architettura Arm Cortex-M0+ con supporto AUTOSAR, Memory-Built-In Self-Test (MBIST) e avvio sicuro. AUTOSAR è un'architettura software

open che offre la possibilità di passare a hardware di livello inferiore mantenendo il codice dell'applicazione originale, facilitando la migrazione tra progetti. Quando si utilizza AUTOSAR, Microchip offre livelli di astrazione del microcontrollore ASIL B (MCAL) per applicazioni di sicurezza funzionale,

fornendo l'interfaccia hardware di livello inferiore all'MCU. Il PIC32CM JH, se abbinato a uno dei circuiti integrati di sicurezza Trust Anchor TA100 CryptoAutomotive Microchip, è conforme ISO/SAE 21434, il nuovo standard di sicurezza informatica per automotive.

www.microchip.com



IMU con localizzazione GNSS ad alto SWaP da SBG Systems

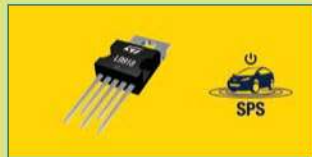
SBG Systems, produttore di sensori inerziali all'avanguardia come Inertial Measurement Units (IMU), Attitude and Heading Reference Systems (AHRS), Sistemi di navigazione inerziale con GNSS integrato (INS/GNSS) e soluzioni per la geo-referenziazione orientata agli UAV e alle soluzioni di rilevamento terrestre, presenta Quanta Micro, un sistema di navigazione inerziale assistito da GNSS progettato per applicazioni con vincoli di spazio. Quanta Micro offre prestazioni eccezionali, Incredibile SWaP ed è basato su una IMU Survey-grade con una calibrazione della temperatura da



-40 a +85°C e da un ricevitore GNSS multi-frequenza e multi-costellazione all'avanguardia. Quanta Micro è a prova di vibrazioni e i suoi giroscopi a basso rumore offrono un'elevata stabilità e la bassa distorsione (0,8°/h) mantiene un'intestazione ad antenna singola altamente precisa in condizioni difficili come la mappatura dei corridoi e i voli a bassa dinamica. Quanta Micro offre una navigazione continua durante le interruzioni del GNSS, con rollo/beccheggio di 0,015°, Rotta 0,035° (PPK) e performance di posizione RTK/PPK di 1 cm e profili di movimento per auto-veicoli, aerei e natanti. www.sbg-systems.com

Regolatore LIN per alternatori VDA-compliant da ST

ST introduce il regolatore per alternatori auto a 12V L9918. Il componente contiene un MOSFET per fornire la corrente di eccitazione e un diodo freewheel per gestire la corrente del rotore quando l'eccitazione è disattivata. Il funzionamento a circuito chiuso con controllo della risposta del carico (LRC) mantiene una tensione costante. Il MOSFET può fornire fino a 13A e grazie alla tecnologia di produzione, ha un RDS(on) minima, a vantaggio dell'efficienza energetica. Con la memoria non volatile (NVM) integrata, l'L9918 consente di programmare parametri tra cui il punto di setpoint della tensione e le caratteristiche dell'alternatore. Può essere utilizzato con alternatori che hanno fino a nove paia di poli.



Integrando un'interfaccia LIN, l'L9918 è facilmente collegabile alla ECU per le impostazioni, inclusa la tensione target, nonché la diagnostica. Le caratteristiche di sicurezza e failsafe dell'L9918 includono l'attivazione dell'avvio automatico tramite segnale di fase, che mantiene in modo indipendente il setpoint della tensione dell'alternatore in caso di perdita del collegamento LIN. Conforme VDA LIN Alternator Regulator, è qualificato AEC-Q 100, nonché a ISO26262 e ASIL-B. newsroom.st.com

MCU RISC-V per Controllo Motore da Renesas

Renesas presenta il primo MCU RISC-V ottimizzato per sistemi avanzati di controllo motore, che consente ai progettisti di beneficiare di una soluzione chiavi in mano pronta per l'uso per applicazioni di controllo motori, senza costi di sviluppo, con tempi di commercializzazione ridotti e riduzioni dei costi, grazie alla fornitura di un ASSP pre-programmato che elimina la necessità di approvvigionamento di tool e di investimenti software relativi a RISC-V. Le applicazioni target per la nuova soluzione includono domotica e building automation, dispositivi sanitari, elettrodomestici, droni e altro ancora.

Renesas ha assunto un ruolo di leadership nella fornitura di soluzioni di elaborazione embedded RISC-V, avendo recentemente introdotto le MPU RZ/Five a 64 bit per uso generale basate su una CPU RISC-V a 64 bit e annunciando inoltre lo sviluppo di soluzioni RISC-V per l'automotive.

Il nuovo ASSP di controllo motore R9A02G020 di Renesas si basa sull'IP di elaborazione RISC-V di Andes Technology Corp. Renesas consegnerà il nuo-

vo ASSP RISC-V pre-programmato con un codice applicativo specializzato sviluppato dalle principali società di progettazione indipendenti (IDH) BFG Engineering e DigiPower (HK) Technology Ltd.

Le caratteristiche della soluzione ASSP di Controllo Motore R9A02G020 sono:

- CPU a 32 bit, 32MHz basata su RISC-V ISA (senza costi di licenza);
- funzionalità IP analogiche ADC con tre PGA e S/H dedicati, due DAC, due comparatori e sensore di temperatura;
- timer con funzionalità avanzate per controllo motore e due timer watchdog;
- temperatura operativa fino a +125°C;
- controllo vettoriale sensorless per un motore BLDC, 1/3 shunt, PFC e ingressi per sensori di Hall;
- Flash di 48 KB e SRAM di 16 KB (4 KB con ECC);
- unità di protezione memoria per CPU e bus.

Renesas ha progettato una winning combination 3-Phase Motor Control con l'R9A02G020.

www.renesas.com





ABBONAMENTO
ANNUALE



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



Visita: www.elettronica.in.it

10 NUMERI
A SOLI

€39

ANZICHÉ € 69,⁰⁰

ABBONAMENTO ANNUALE PER RIVISTA DIGITALE

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

€49

ANZICHÉ € 69,⁰⁰

I tuoi PCB a partire da

1€

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

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

Prezzi IVA inclusa.



cod. PCBPRODUCTION

COLORE SERIGRAFIA



BIANCA



NERA

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

COLORE SOLDER



VERDE



ROSSO



GIALLO



BLU



BIANCO



NERO

SPESSORE (mm)

mm



0.2

0.4

0.6

0.8

1.0

1.2

1.6

2.0

2.4

2.8

3.2

NUMERO DI LAYERS

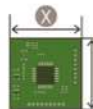
1 Layer

2 Layers

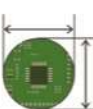
4 Layers

6 Layers

DIMENSIONI (cm)



Single Size



Pannel Size

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

ECCO QUALCHE ESEMPIO:

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

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

CLOUD ENERGY METER



di EMANUELE SIGNORETTA

Contatore di energia e assorbimento basato sulla scheda di prototipazione ESP32, che ci consente di interfacciarci con il cloud.

Il nostro Paese, non è un mistero, sul piano energetico è fortemente dipendente dall'estero perché l'autoproduzione, prettamente da fonti rinnovabili, è inadeguata a soddisfare il fabbisogno nazionale. Questo problema si acuisce ogni volta che fattori esterni causano la limitata disponibilità di fonti fossili e il conseguente incremento del loro prezzo. Sapendo un po' tutti quanto "salate" sono le bollette di luce, gas ci ritroviamo a ridurre i consumi, rinunciando ad alcuni "confort" e a tenerne comunque traccia, per evitare spiacevoli sorprese. Del resto l'attenzione al consumo energetico è mossa prima di tutto (e ce ne rammarichiamo...) dalla paura di spendere troppo, piuttosto che dalle preoccupazioni per i cambiamenti climatici. Comunque, per venire incontro a chi presta attenzione a quanta elettricità consuma, in questo articolo presenteremo un rilevatore di energia consumata, basato su un modulo

→ **Fig. 1**
La scheda ESP32.



ESP32 e su un sensore di tensione/corrente siglato PZEM004T, i cui dati verranno inviati in Cloud, nello specifico ai server del servizio on-line InfluxDb.

L'HARDWARE DEL PROGETTO

Per realizzare questo nostro contatore di energia utilizzeremo, per la parte hardware, un modulo WiFi ESP32, acquistabile sul sito www.futurashop.it con il codice HR0204 (**Fig. 1**), un sensore PZEM004T (lo vedete in **Fig. 2** incapsulato nel suo contenitore in plastica trasparente) un alimentatore switching con output 5V (se ne scegliete uno ad uscita microUSB potete alimentare con esso la scheda ESP e prendere da VIN l'alimentazione per il PZEM004T...) e dei jumper



→ **Fig. 2**
Il sensore PZEM004T con il trasformatore.



femmina-femmina per Arduino, morsetti, alcuni pezzi di cavo elettrico e un box di plastica (cod. KZ57) di www.futurashop.it. Il sensore di assorbimento è sostanzialmente un trasformatore di corrente dotato di case in plastica e toroide apribile, con un minimo di elettronica disaccoppiata mediante fotoaccoppiatori e interfaccia seriale. Le scelte degli elementi circuitali sono state fatte in virtù dei costi dell'hardware rimasti comunque contenuti nonostante la crisi dei semiconduttori e per le ottime specifiche dell'ESP32.

CONFIGURAZIONE DI INFLUXDB

Esistono diversi software che permettono di ricevere ed elaborare i dati trasmessi dai dispositivi IoT: non possiamo non citare Home Assistant, Grafana, Blynk, Thingspeak ecc. Per evitare di essere prolissi sull'utilizzo di alcuni software già trattati abbiamo deciso di introdurre l'utilizzo di InfluxDb, il cui logo è proposto nella **Fig. 3**.

InfluxDb è un software open source che permette di creare dashboard, eseguire query e inviare alert. È possibile utilizzarlo in diverse modalità: eseguibile (con numerose architetture supportate), container Docker, e Cloud. Per questioni di "proprietà dei dati" avremmo potuto installare un'istanza su Raspberry Pi, ma a causa dello shortage di componenti, ci siamo visti costretti a ricorrere al servizio **InfluxDb Cloud v2**.

Rechiamoci per prima cosa al link <https://cloud2.influxdata.com/signup> e registriamoci utilizzando o il social login di Microsoft e Google oppure compilando tutti i campi sottostanti (**Fig. 4**).

Una volta registrati al servizio ci apparirà una schermata come in **Fig. 5**, nella quale vengono richieste diverse informazioni, tra cui il provider per il servizio: per il nostro progetto abbiamo scelto Amazon Web Services (AWS). Nella schermata successiva (proposta nella **Fig. 6**) dovremo scegliere il piano di utilizzo: nel nostro caso abbiamo optato il piano free, che comprende archiviazione dati per una durata di 30 giorni e la possibilità di inviare le notifiche tramite Slack.

Confermata la nostra opzione ci apparirà la dashboard personale; da essa, per creare un bucket, andiamo su → **"Load Data" → Buckets" → "Create new bucket"**. Si aprirà una schermata come quella esemplificata nella **Fig. 7**, nella quale bisognerà compilare il campo relativo al nome e poi cliccare sul pulsante **"Create"**.

Una volta creato il bucket, quest'ultimo apparirà tra le data sources disponibili, come esemplificato nella **Fig. 8**.

↓ Listato 1

```
#include <WiFiMulti.h>
#include <ESPmDNS.h>
#include <WiFiUdp.h>
#include <ArduinoOTA.h>
#include <InfluxDbClient.h>
#include <InfluxDbCloud.h>
#include <PZEM004Tv30.h>
#include <Every.h>
```

Clicchiamo dunque su **"Add data" → "Client library"**. Nella nuova schermata che si aprirà selezioniamo la voce **"Arduino"** (Fig. 9) tra le possibili alternative per l'upload dei dati.

Nella nuova finestra che si apre (Fig. 10) comparirà una serie di snippet che andranno a comporre uno sketch dimostrativo. Prendiamo in considerazione il primo di tali snippet e copiamo i dati relativi a *INFLUXDB_URL*, *INFLUXDB_ORG* e *INFLUXDB_BUCKET*. Come ultimo passaggio è necessario generare un token di accesso al bucket; per fare ciò è necessario recarci sulla sidebar di sinistra e poi su **"Load Data" → "Api Tokens"**.

Nella nuova finestra che si aprirà clicchiamo su **"Generate API Token"** e una volta selezionato il bucket, scegliamo di abilitare i permessi in lettura e scrittura come in Fig. 11. Una volta generato il token, copiamolo e mettiamolo da parte poiché ci servirà tra poco.

LO SKETCH

Lo sketch del nostro progetto è frutto dell'unione di diversi sketch delle librerie utilizzate. È possibile scaricare i file dello sketch mediante il link del repository GitHub raggiungibile dall'indirizzo github.com/signorettae/ESP32-EnergyMeter. Procediamo adesso ad analizzare i nostri sorgenti, suddivisi in tre listati che esporremo, preceduti da una parte relativa all'inclusione delle librerie e delle dipendenze necessarie, proposta qui di seguito.

```
#include <WiFiMulti.h>
#include <ESPmDNS.h>
#include <WiFiUdp.h>
#include <ArduinoOTA.h>
#include <InfluxDbClient.h>
#include <InfluxDbCloud.h>
#include <PZEM004Tv30.h>
#include <Every.h>
```

A riguardo precisiamo che la libreria *WiFiMulti* serve a gestire la comunicazione WiFi tra l'ESP32 e l'ac-



Fig. 3
Logo di
InfluxDB.

cess point, mentre le librerie *ESPmDNS*, *WiFiUDP* e *ArduinoOTA* (<https://github.com/jandrassy/ArduinoOTA>) servono a gestire il caricamento OTA (Over The Air) degli sketch e il nome dell'host.

Facciamo presente che per fare uso di questa funzionalità è necessario avere Python v. 2.7.x installato sul PC che utilizzerete.

Le librerie *InfluxDbClient* e *InfluxDbCloud* servono

Fig. 4
Pagina di
registrazione
a InfluxDB.

↓ Listato 2

```
#define REFRESH_TIME 5000 // Delay interval for data upload
#define DEVICE "ESP32"
#define PZEM_RX_PIN 27
#define PZEM_TX_PIN 26
#define PZEM_SERIAL Serial2

/*****
  ESP32 initialization
  *****/

The ESP32 HW Serial interface can be routed to any GPIO pin
Here we initialize the PZEM on Serial2 with RX/TX pins 26 and 27

*/
PZEM004Tv30 pzem(PZEM_SERIAL, PZEM_RX_PIN, PZEM_TX_PIN);
WiFiMulti wifiMulti;
IPAddress local_IP(192, 168, 178, 154);
IPAddress gateway(192, 168, 178, 1);
IPAddress subnet(255, 255, 255, 0);
IPAddress primaryDNS(192, 168, 178, 1); //optional
IPAddress secondaryDNS(1, 1, 1, 1); //optional
// WiFi AP SSID
#define WIFI_SSID ""
// WiFi password
#define WIFI_PASSWORD ""
// InfluxDB v2 server url, e.g. https://eu-central-1-1.aws.cloud2.influxdata.com (Use: InfluxDB UI -> Load
Data -> Client Libraries)
#define INFLUXDB_URL "https://eu-central-1-1.aws.cloud2.influxdata.com"
// InfluxDB v2 server or cloud API token (Use: InfluxDB UI -> Data -> API Tokens -> Generate API Token)
#define INFLUXDB_TOKEN ""
// InfluxDB v2 organization id (Use: InfluxDB UI -> User -> About -> Common Ids )
#define INFLUXDB_ORG ""
// InfluxDB v2 bucket name (Use: InfluxDB UI -> Data -> Buckets)
#define INFLUXDB_BUCKET "Energy Meter @ ElettronicaIN"
// Set timezone string according to https://www.gnu.org/software/libc/manual/html_node/TZ-Variable.html
#define TZ_INFO "CET-1CEST,M3.5.0,M10.5.0/3"
// InfluxDB client instance with preconfigured InfluxCloud certificate
InfluxDBClient client(INFLUXDB_URL, INFLUXDB_ORG, INFLUXDB_BUCKET, INFLUXDB_TOKEN, InfluxDbCloud2CACert);
// Data point
Point sensor("EnergyMeter");
```

invece a gestire l'upload dei dati sul Cloud di Influx. La libreria PZEM004Tv30 (github.com/mandulaj/PZEM-004T-v30) viene utilizzata per la comunica-

zione seriale con il sensore e infine, la libreria Every (scaricabile dal sito della rivista) viene invece utilizzata per eseguire dei blocchi di codice ad intervalli regolari senza fare uso dei `delay()`.

Passiamo adesso ad analizzare le successive porzioni di codice, a iniziare da quella proposta nel **Listato 2**, la quale contiene la parte del preprocessore. Con `#define REFRESH_TIME 5000` impostiamo l'intervallo di caricamento in millisecondi dei dati sul cloud. Vengono successivamente definiti il nome del dispositivo (lo ritroveremo successivamente), i pin per la comunicazione seriale e quale porta seriale vorremo utilizzare. Vengono in seguito creati gli oggetti relativi alla rete WiFi e al sensore. Ancora a seguire, vengono definiti i vari indirizzi IP per configurare una connessione all'AP con IP statico, cambiamo i parametri a seconda della subnet della nostra rete. Procediamo infine a definire i parametri per la connessione WiFi e per l'accesso al cloud di Influx.

Fig. 5
Scelta del provider
e accettazione dei
termini di servizio.

The screenshot shows the InfluxDB Cloud setup wizard with three steps:

- 1 Choose a Provider & Region**: Three buttons for "Amazon Web Services", "Google Cloud", and "Microsoft Azure" are visible. The "Amazon Web Services" button is selected. Below it, a dropdown menu shows "EU Frankfurt" as the selected region.
- 2 Enter Company Name**: A text input field contains the name "Emanuele Signorella".
- 3 Read and agree to our service agreements**: A checkbox is checked, indicating agreement to the "InfluxDB Cloud 2.0 Services Subscription Agreement" and the "InfluxData Global Data Processing Agreement".

Listato 3

```
void setup() {
  Serial.begin(115200);
  // Uncomment in order to reset the internal energy counter
  // pzem.resetEnergy()
  // Setup wifi
  WiFi.mode(WIFI_STA);
  //Comment to use DHCP instead of static IP
  if (!WiFi.config(local_IP, gateway, subnet, primaryDNS, secondaryDNS)) {
    Serial.println("STA Failed to configure");
  }
  wifiMulti.addAP(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to wifi");
  while (wifiMulti.run() != WL_CONNECTED) {
    //Serial.print(".");
    //delay(100);
    Serial.println("Connection Failed! Rebooting...");
    delay(5000);
    ESP.restart();
  }
  Serial.println();
  // Port defaults to 3232
  // ArduinoOTA.setPort(3232);
  // Hostname defaults to esp3232-[MAC]
  ArduinoOTA.setHostname("ESP32-Energy Meter");
  // No authentication by default
  // ArduinoOTA.setPassword("admin");
  // Password can be set with it's md5 value as well
  // MD5(admin) = 21232f297a57a5a743894a0e4a801fc3
  // ArduinoOTA.setPasswordHash("21232f297a57a5a743894a0e4a801fc3");
  ArduinoOTA
    .onStart([]() {
      String type;
      if (ArduinoOTA.getCommand() == U_FLASH)
        type = "sketch";
      else // U_SPIFFS
        type = "filesystem";
      // NOTE: if updating SPIFFS this would be the place to unmount SPIFFS using SPIFFS.end()
      Serial.println("Start updating " + type);
    })
    .onEnd([]() {
      Serial.println("\nEnd");
    })
    .onProgress([](unsigned int progress, unsigned int total) {
      Serial.printf("Progress: %u%%\r", (progress / (total / 100)));
    })
    .onError([](ota_error_t error) {
      Serial.printf("Error[%u]: ", error);
      if (error == OTA_AUTH_ERROR) Serial.println("Auth Failed");
      else if (error == OTA_BEGIN_ERROR) Serial.println("Begin Failed");
      else if (error == OTA_CONNECT_ERROR) Serial.println("Connect Failed");
      else if (error == OTA_RECEIVE_ERROR) Serial.println("Receive Failed");
      else if (error == OTA_END_ERROR) Serial.println("End Failed");
    });
  ArduinoOTA.begin();
  // Add tags
  sensor.addTag("Dispositivo", DEVICE);
  // Accurate time is necessary for certificate validation and writing in batches
  // For the fastest time sync find NTP servers in your area: https://www.pool.ntp.org/zone/
  // Syncing progress and the time will be printed to Serial.
  timeSync(TZ_INFO, "pool.ntp.org", "time.nis.gov");\
  // Check server connection
  if (client.validateConnection()) {
    Serial.print("Connected to InfluxDB: ");
    Serial.println(client.getServerUrl());
  } else {
    Serial.print("InfluxDB connection failed: ");
    Serial.println(client.getLastErrorMessage());
  }
}
```


Listato 4

```
void loop() {
  ArduinoOTA.handle();
  EVERY(REFRESH_TIME) {
    // Print the custom address of the PZEM
    Serial.print("Custom Address:");
    Serial.println(pzem.readAddress(), HEX);
    // Read the data from the sensor
    float voltage = pzem.voltage();
    float current = pzem.current();
    float power = pzem.power();
    float energy = pzem.energy();
    float frequency = pzem.frequency();
    float pf = pzem.pf();
    // Check if the data is valid
    if (isnan(voltage)) {
      Serial.println("Error reading voltage");
    } else if (isnan(current)) {
      Serial.println("Error reading current");
    } else if (isnan(power)) {
      Serial.println("Error reading power");
    } else if (isnan(energy)) {
      Serial.println("Error reading energy");
    } else if (isnan(frequency)) {
      Serial.println("Error reading frequency");
    } else if (isnan(pf)) {
      Serial.println("Error reading power factor");
    } else {
      // Print the values to the Serial console
      Serial.print("Voltage: ");      Serial.print(voltage);      Serial.println("V");
      Serial.print("Current: ");      Serial.print(current);      Serial.println("A");
      Serial.print("Power: ");        Serial.print(power);        Serial.println("W");
      Serial.print("Energy: ");        Serial.print(energy, 3);      Serial.println("kWh");
      Serial.print("Frequency: ");    Serial.print(frequency, 1);  Serial.println("Hz");
      Serial.print("PF: ");           Serial.println(pf);
      ///UPLOAD DATA
      // Clear fields for reusing the point. Tags will remain untouched
      sensor.clearFields();
      // Store measured value into point
      sensor.addField("Tensione", voltage);
      sensor.addField("Corrente", current);
      sensor.addField("Potenza", power);
      sensor.addField("Energia", energy);
      sensor.addField("Frequenza", frequency);
      sensor.addField("Fattore di potenza", pf);
      // Print what are we exactly writing
      Serial.print("Writing: ");
      Serial.println(sensor.toLineProtocol());
      // Check WiFi connection and reconnect if needed
      if (wifiMulti.run() != WL_CONNECTED) {
        Serial.println("Wifi connection lost");
      }
      // Write point
      if (!client.writePoint(sensor)) {
        Serial.print("InfluxDB write failed: ");
        Serial.println(client.getLastErrorMessage());
      }
    }
    Serial.println();
  }
}
```

Ci basterà sostituire queste righe di codice con lo snippet copiato precedentemente e inserire i dati mancanti.

Con `#define WIFI_SSID` e `#define WIFI_PASSWORD` andremo a definire rispettivamente SSID e passphrase della nostra rete WiFi.

Passiamo adesso ad inserire i parametri per la connessione al cloud di Influx. L'unico parametro mancante è `INFLUXDB_TOKEN`: recuperiamo la chiave API creata precedentemente e inseriamola tra le virgolette.

Con `#define TZ_INFO "CET-1CEST,M3.5.0,M10.5.0/3"`

viene definito il fuso orario dell'Europa Centrale per il timestamp e con `InfluxDBClient client(INFLUXDB_URL, INFLUXDB_ORG, INFLUXDB_BUCKET, INFLUXDB_TOKEN, InfluxDbCloud2CACert)`; viene creato l'oggetto `client` che verrà utilizzato per la connessione al server di Influx.

Infine, con `Point sensor("EnergyMeter")`; viene creato l'oggetto `sensor`, soprannominato "EnergyMeter". A questo oggetto verranno associati tutti i dati raccolti dal sensore.

Il codice contenuto nel **Listato 3** è quello contenuto nel `setup` della scheda. Viene dapprima inizializzata la porta seriale a 115.200 baud/s.

Con il comando `WiFi.mode(WIFI_STA)`; il WiFi viene inizializzato in modalità "station", cioè per connettersi ad un access point. Invece con:

```
if (!WiFi.config(local_IP, gateway, subnet,
    primaryDNS, secondaryDNS)) {
    Serial.println("STA Failed to configure");
}
```

viene settato l'utilizzo di un IP statico, anziché fare uso del DHCP. In caso di errore viene scritto un codice di segnalazione sulla seriale. Commentate questa parte di codice per usare il DHCP. Con l'istruzione `wifiMulti.addAP(WIFI_SSID, WIFI_PASSWORD)`; vengono settati SSID e passphrase della rete WiFi.

Viene successivamente tentata la connessione all'AP e in caso di errore viene prima stampato un errore sulla seriale, per poi riavviare la scheda, dopo un'attesa di cinque secondi.

Con `ArduinoOTA.setHostname("ESP32-Energy Meter")`; viene impostato l'hostname che verrà trasmesso attraverso mDNS e che verrà visualizzato sull'IDE di Arduino per il caricamento dello sketch OTA. È presente anche del codice commentato che serve a limitare il caricamento dello sketch OTA a coloro i quali che saranno forniti di username e password. Decomentate queste righe di codice per aggiungere questa funzionalità di sicurezza. Con le istruzioni:

```
ArduinoOTA
.onStart([]() {
    String type;
    |
    |
    |
    if (error == OTA_AUTH_ERROR) Serial.println("--
Auth Failed");
    else if (error == OTA_BEGIN_ERROR) Serial.print-
```

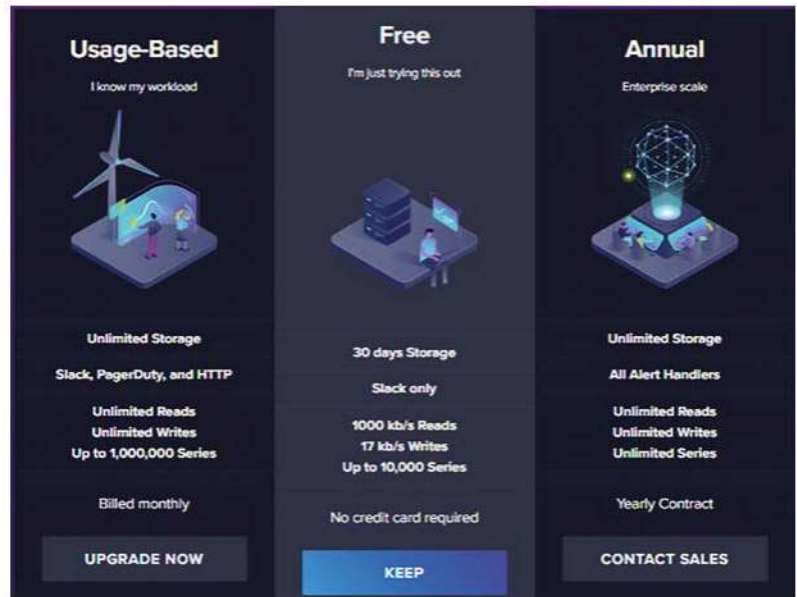
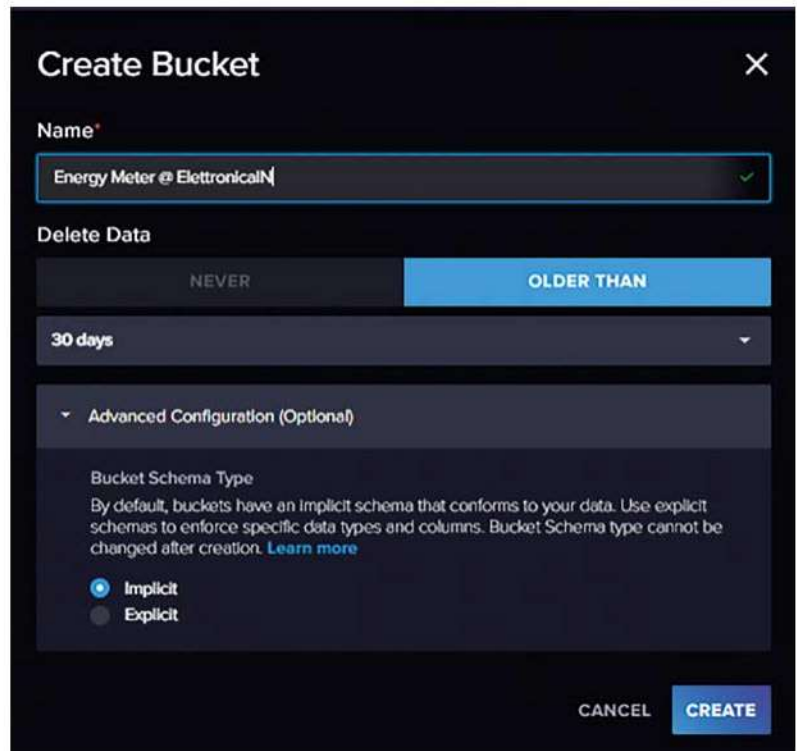


Fig. 6
Scelta del piano di iscrizione.

```
ln("Begin Failed");
else if (error == OTA_CONNECT_ERROR) Serial.
println("Connect Failed");
else if (error == OTA_RECEIVE_ERROR) Serial.
println("Receive Failed");
else if (error == OTA_END_ERROR) Serial.println("End Failed");
});
```

Fig. 7
Creazione di un nuovo bucket.



Energy Meter @ ElettronicaIN

Retention: 30 days Schema Type: Implicit ID: e885cc2422e3f2e0

+ Add a label

+ ADD DATA

SETTINGS

Fig. 8
Un bucket
appena
creato.

viene gestito il settaggio della modalità OTA. In particolare, vengono gestiti nell'ordine, l'avvio, la fine, il caricamento dello sketch e gli errori. Con `ArduinoOTA.begin()` viene avviata la gestione del caricamento OTA. Con `sensor.addTag("Dispositivo", DEVICE)` viene impostato un "tag" che in questo caso corrisponde al nome del dispositivo. Questa caratteristica potrebbe essere utile per distinguere uno o più sensori all'interno di una flotta. Con `timeSync(TZ_INFO, "pool.ntp.org", "time.nis.gov")`; l'ESP si collega ai due server NTP (Network Time Protocol) `pool.ntp.org` e `time.nis.gov`

per ottenere data e ora.
Infine, con queste istruzioni:

```
if (client.validateConnection()) {  
    Serial.print("Connected to InfluxDB: ");  
    Serial.println(client.getServerUrl());  
} else {  
    Serial.print("InfluxDB connection failed: ");  
    Serial.println(client.getLastErrorMessage());  
}
```

viene tentata la connessione ai server di Influx e,

Fig. 9
Selezione della
fonte di upload dei
dati.

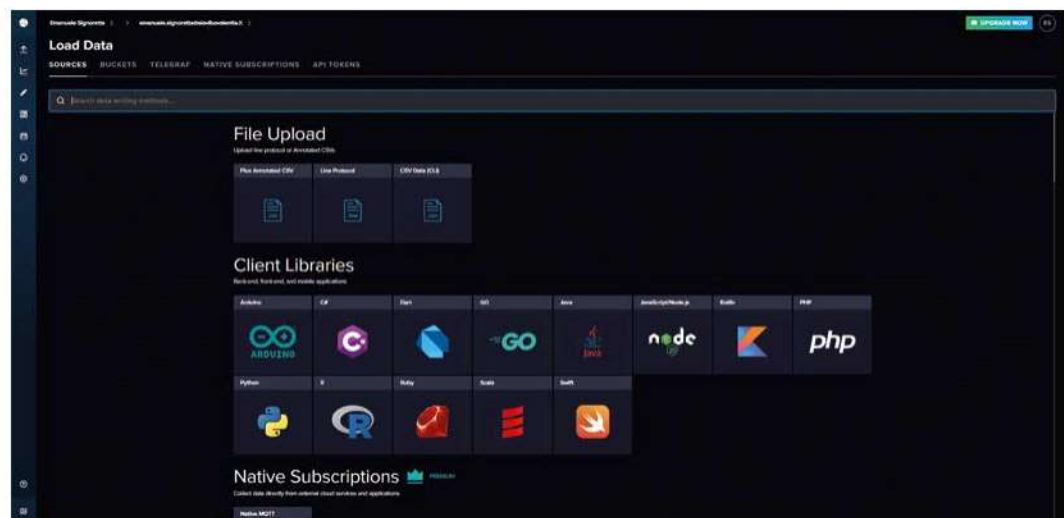


Fig. 10
Snippet per
uno sketch di
demo.

Initialize the Client

```
#include <InfluxDbCloud.h>  
  
// WiFi AP SSID  
#define WIFI_SSID "SSID"  
// WiFi password  
#define WIFI_PASSWORD "PASSWORD"  
// InfluxDB v2 server url, e.g. https://eu-central-1-1.aws.cloud2.influxdata.com (Use: InfluxDB UI -> Load Data -> Client Libraries)  
#define INFLUXDB_URL "https://eu-central-1-1.aws.cloud2.influxdata.com"  
// InfluxDB v2 server or cloud API token (Use: InfluxDB UI -> Data -> API Tokens -> Generate API Token)  
#define INFLUXDB_TOKEN "<INFLUX_TOKEN>"  
// InfluxDB v2 organization id (Use: InfluxDB UI -> User -> About -> Common Ids )  
#define INFLUXDB_ORG "emanuele.signoretta@aiavibovalentia.it"  
// InfluxDB v2 bucket name (Use: InfluxDB UI -> Data -> Buckets)  
#define INFLUXDB_BUCKET "Energy Meter @ ElettronicaIN"  
  
// Set timezone string according to https://www.gnu.org/software/libc/manual/html_node/TZ-Variable.html  
// Timezone
```

COPY TO CLIPBOARD

in caso di connessione favorevole viene stampato su seriale l'URL del server. In caso contrario viene stampato sulla seriale il messaggio di errore corrispondente.

Andiamo adesso al **Listato 4**, che contiene il codice del *loop()* del nostro sketch. Con il comando `ArduinoOTA.handle();` l'ESP32 esegue la routine per rimanere in attesa di ricevere OTA eventuali sketch compilati.

Con `EVERY(REFRESH_TIME){...}` viene eseguito il codice contenuto all'interno delle parentesi a intervalli regolari e senza fare uso dei `delay`. L'intervallo di tempo è stato definito precedentemente. All'interno del blocco `EVERY` viene prima letto e stampato su seriale l'indirizzo del PZEM. Successivamente vengono create e inizializzate tutte le variabili del sensore con i comandi della libreria dello stesso. Nell'ordine vengono create le variabili relative a tensione, corrente, potenza attiva (in Wh), energia attiva (in kWh), frequenza e fattore di potenza (il rapporto tra potenza attiva e potenza apparente, varia tra -1 e 1). Se almeno una delle variabili non dovesse risultare un numero, verrebbe stampato sulla seriale il relativo errore di lettura. In caso contrario, verrebbero stampate le letture del sensore. Fatto ciò, verrebbero puliti i campi e i timestamp inizializzati precedentemente e preparate le variabili per l'upload sul cloud. In caso di errore. Verranno infine controllate la connessione WiFi e l'esito favorevole dell'upload dei

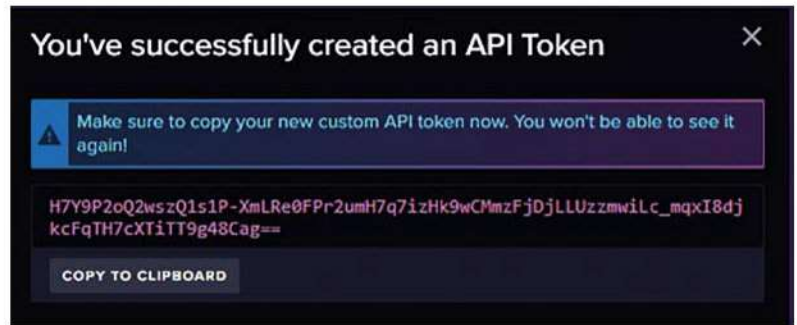


Fig. 11
Creazione
di una chiave API.

dati. Qualora quest'ultimo dovesse fallire, verrebbe stampato su seriale l'ultimo codice di errore.

INSTALLAZIONE

Abbiamo pensato di chiudere alimentatore, ESP32 e il modulo principale del sensore all'interno di un box di plastica, facendo entrare nella scheda solamente i fili per l'alimentazione e quelli che vanno al toroide sensore di corrente. L'alimentatore sarà collegato con l'ingresso direttamente all'uscita del contatore (magari al magnetotermico generale dell'impianto).

Per l'installazione dell'energy meter è sufficiente seguire lo schema riportato in **Fig. 12**.

In particolare, è necessario collegare due fili in parallelo all'ingresso dell'alimentatore e tra i pin N e L del PZEM004T, dunque in parallelo a tutto l'impianto elettrico. Ricordate di prestare sempre

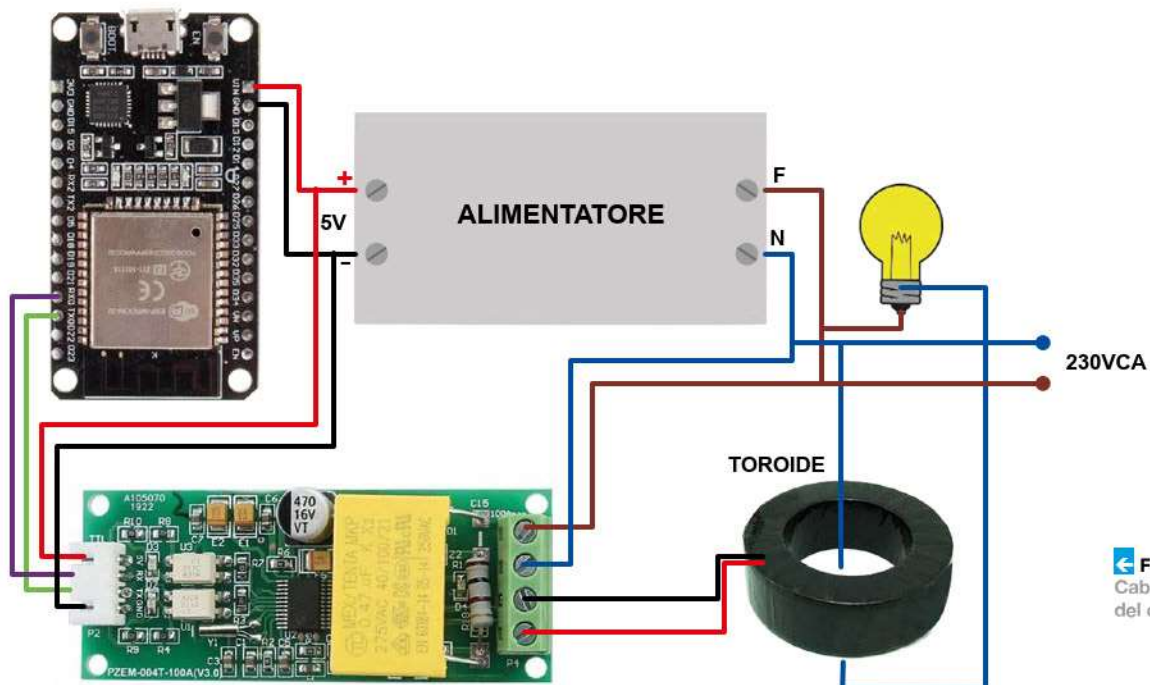


Fig. 12
Cablaggio
del circuito.

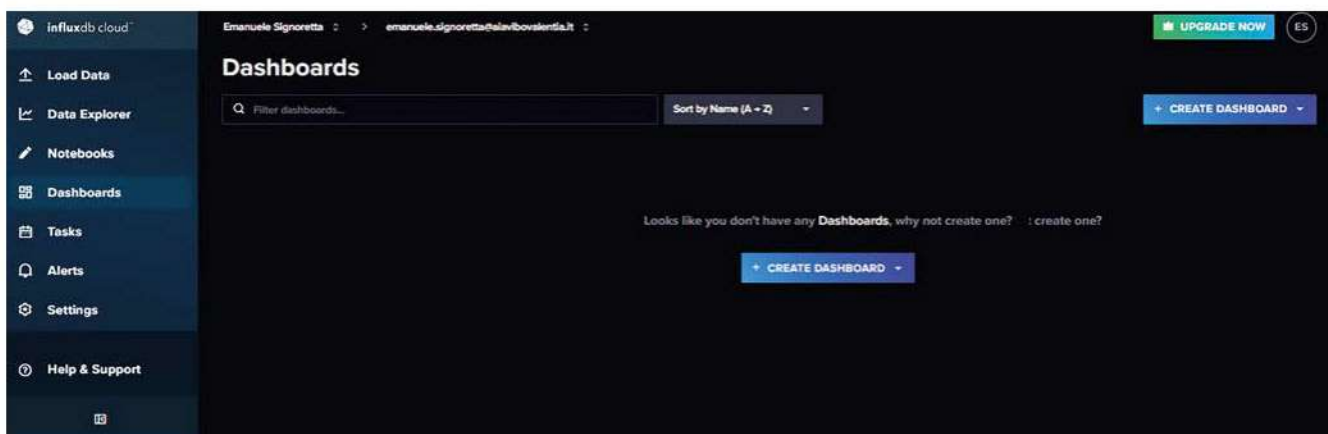


Fig. 13
Creazione della dashboard.

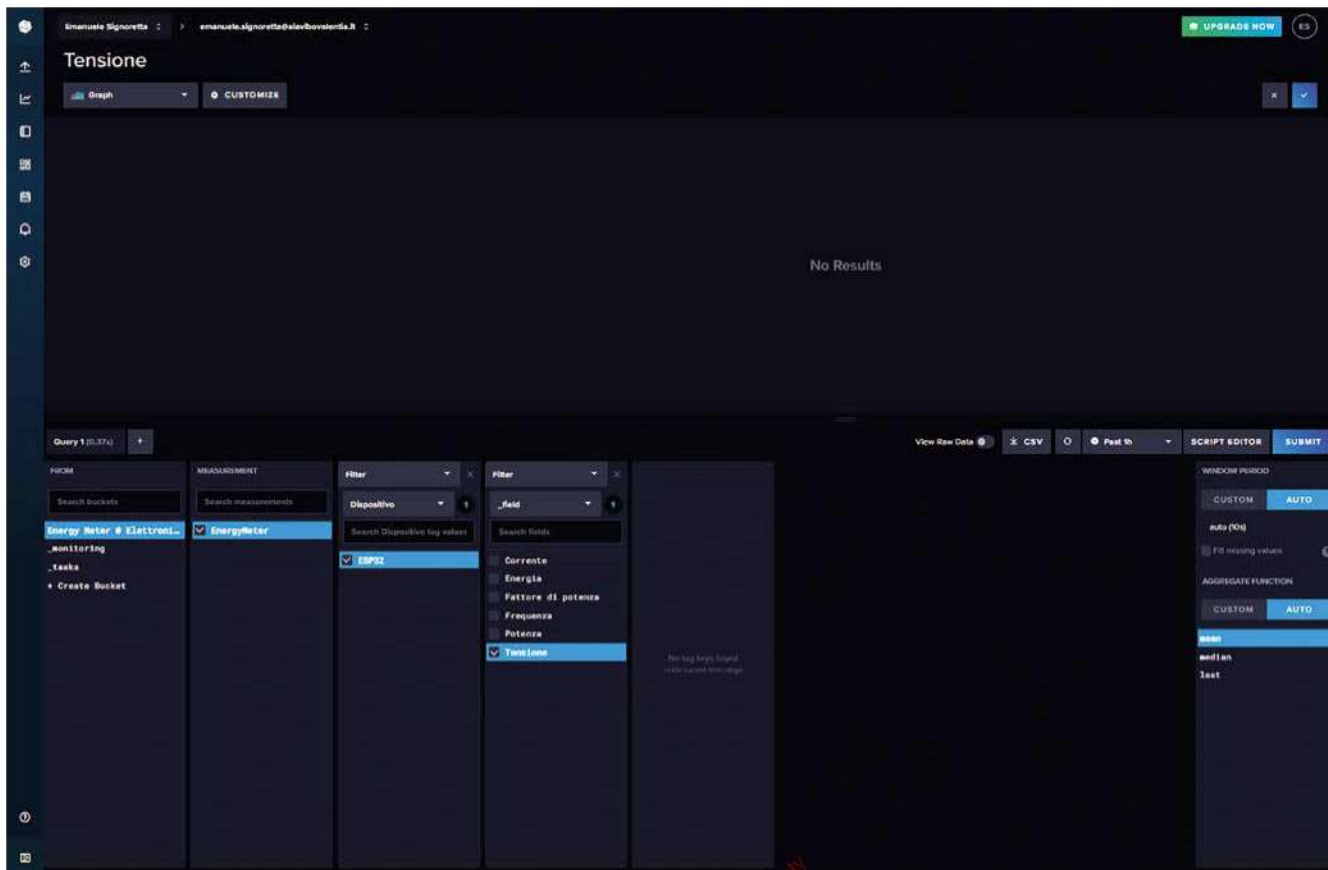
Fig. 14
Aggiunta dei grafici della dashboard.

la massima attenzione perché si lavora sui cavi della linea di distribuzione elettrica a 230Vca. Successivamente sganciamo la sicura del toroide e facciamo passare uno dei due fili che vanno all'impianto elettrico, quindi colleghiamo i fili del toroide ai pin CT del sensore. Colleghiamo invece il modulo principale del sensore all'ESP32, seguendo la corrispondenza riportata nella **Tabella 1**. Potete

utilizzare un alimentatore con uscita micro USB con cui alimentare l'ESP, prelevando i 5V per il sensore da VIN e GND. Una volta caricato lo sketch possiamo chiudere tutta l'elettronica nella scatola e avviare il sistema.

AVVIO E PREPARAZIONE DASHBOARD

Una volta avviata la scheda andiamo di nuovo sul



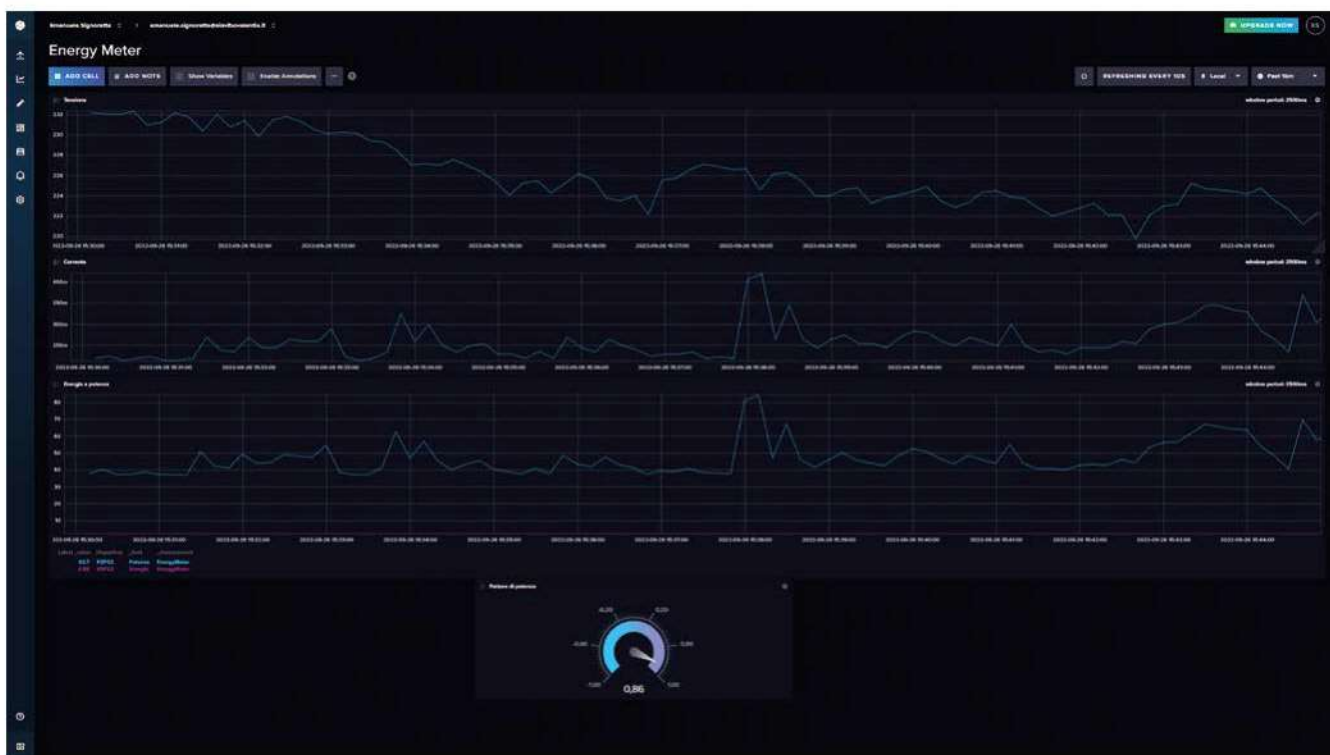


Tabella 1
Collegamenti fra modulo Energy Meter
e modulo ESP32.

PIN PZEM004	PIN ESP
VCC	V5
GND	GND
RX	26
TX	27

sito Internet di InfluxDb e dalla nostra dashboard andiamo su **"Dashboards" → "Create new dashboard"**. Facendo ciò apparirà una schermata come quella visualizzata in **Fig. 13**, nella quale andiamo ad assegnare un nome alla nostra dashboard e clicchiamo dunque su **"Add Cell"**. Nella nuova schermata che ci apparirà, andremo a selezionare quale parametri inserire in ciascuno dei grafici che costituiscono la dashboard. Selezioniamo in ordine, il bucket, le misurazioni, il dispositivo e infine i valori da visualizzare. È possibile scegliere quale veste grafica utilizzare per i valori: heatmap, gauge, grafico semplice, tabella, ecc. e anche quali valori di scala utilizzare. Personalizziamo le celle a nostro

piacimento e ripetiamo la procedura per ciascun valore che vogliamo visualizzare fino a quando non otterremo un risultato come quello visibile nella **Fig. 15**, che propone la nostra dashboard completa.

CONCLUSIONI

Termina qui la descrizione del nostro semplice ma potente energy meter. La duttilità della scheda ESP32 e le numerose funzionalità dei protocolli di rete e piattaforme supportati ci permettono di espanderne i campi di applicazione in base alle nostre necessità, magari integrando il misuratore in un sistema di gestione dei carichi e comunque capace di azioni autonome. □

Fig. 15
La dashboard
ultimata.



Cosa occorre?

Il modulo ESP32 (cod. HR0204) è in vendita a Euro 14,50, l'energy meter (cod. PZEM004T) costa Euro 25,00. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

Nuove Telecamere

Full color technology



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

TELECAMERA FULL COLOR 2 MPX - VARIFOCAL



€ 69,⁰⁰

cod. FR816VAR

TELECAMERA FULL COLOR 2 MPX



€ 52,⁰⁰

cod. FR816N

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

- ✓ Uscita Video (BNC): 4 in 1 - AHD / CVI / TVI / CVBS
- ✓ Sensore di immagine: 1/2.7" CMOS
- ✓ Risoluzione immagine: 2 MPX
- ✓ Pixel effettivi: 1920(H)*1080(V)
- ✓ Sistema TV: PAL
- ✓ Obiettivo: varifocal 2,8-12 mm
- ✓ Otturatore elettronico: Automatico

- ✓ Portata Illuminatore: 40 metri
- ✓ Grado di protezione: IP67
- ✓ Alimentazione: 12 VDC
- ✓ Consumo: IR OFF: <1W; LED ON: 3,6 W
- ✓ Temperatura di Lavoro: da -20 C a +50 C (RH95% Max)
- ✓ Dimensioni corpo telecamera: 138x75x67 mm
- ✓ Peso: circa 408 grammi

TELECAMERA DOME FULL COLOR - 2 MPX

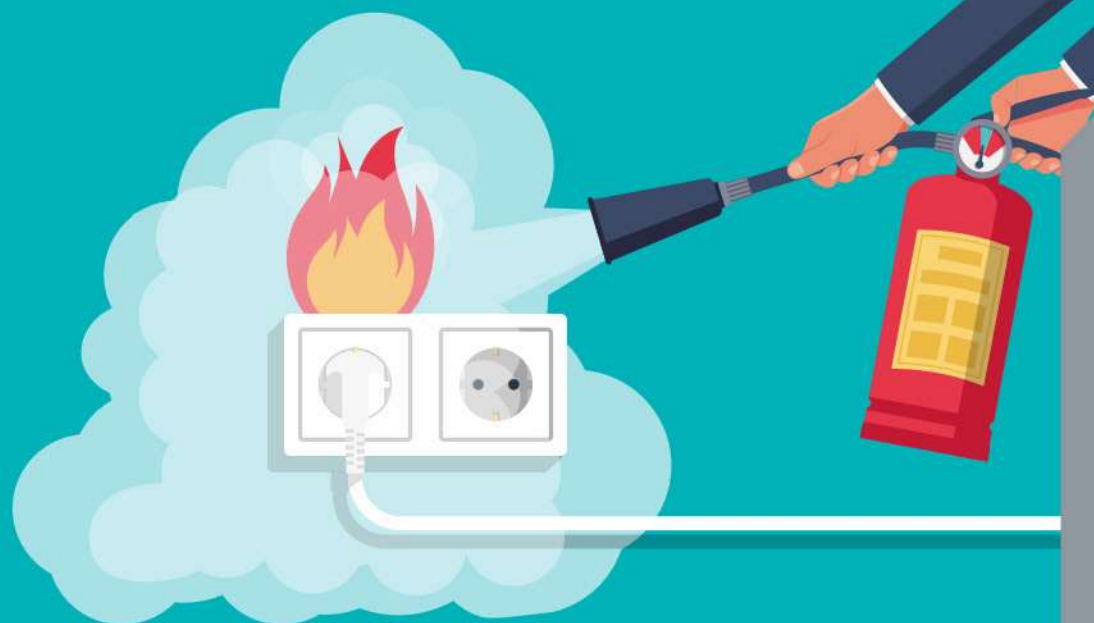
- ✓ Uscita Video (BNC): 4 in 1 - AHD / CVI / TVI / CVBS
- ✓ Sensore di immagine: 1/2.8" CMOS
- ✓ Risoluzione immagine: 2 MPX
- ✓ Pixel effettivi: 1920(H)*1080(V)
- ✓ Sistema TV: PAL
- ✓ Obiettivo: 3,6 mm
- ✓ Otturatore elettronico: Automatico
- ✓ Portata Illuminatore: 20-30 metri
- ✓ Grado di protezione: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

€ 44,⁰⁰

cod. FR817



Trasformano
la notte 
in  giorno!



PROTEZIONE DA SOVRACCARICO AC

di BORIS LANDONI

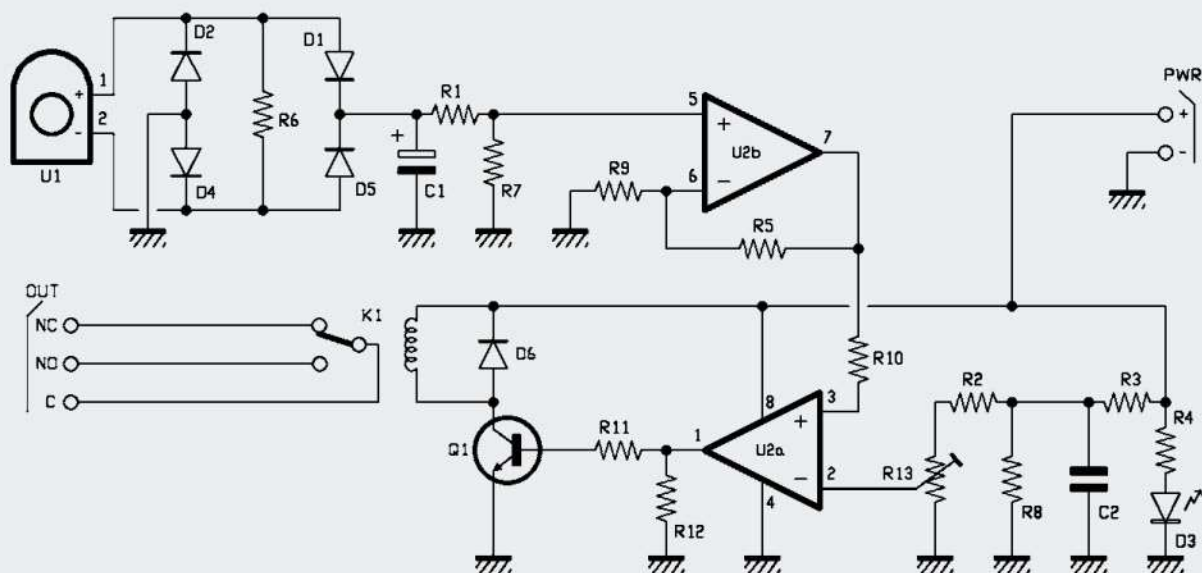
Tramite un sensore induttivo rileva la corrente in una linea elettrica e con un relé può attivare un avvisatore o sconnettere un carico se viene superata la soglia impostata mediante un trimmer.



N

on è infrequente trovarsi nella situazione in cui a un certo punto il contatore di casa si stacca perché abbiamo acceso contemporaneamente troppi elettrodomestici o utilizzatori elettrici energivori: quanto sarebbe stato utile avere un avvisatore o qualsiasi altro dispositivo

in grado di rilevare la situazione di imminente distacco o addirittura di provvedere a sconnettere in maniera automatica e autonoma la linea che porta il maggior carico al contatore? Ebbene, dispositivi del genere esistono e ce li offre l'elettronica: in commercio si trovano in varie esecuzioni, tipicamente da montare nei quadri elettrici per sconnettere, secondo una priorità fissa o programmabile, alcune linee di un impianto ed evitare che il distacco del contatore causato dal sovraccarico possa privare della tensione di rete gli utilizzatori che devono



rimanere in funzione, come ad esempio il frigorifero. Se amate l'autocostruzione, un dispositivo del genere ve lo proponiamo in queste pagine: si tratta di una scheda in grado di rilevare l'assorbimento su una linea elettrica a 230Vca (in generale, su qualsiasi linea in corrente alternata) e produrre un allarme in caso di superamento di una soglia di corrente. L'allarme può essere fornito da un segnalatore acustico o visivo gestito dal relé a bordo della scheda; lo stesso relé, del quale rendiamo disponibile l'intero scambio, costituisce un'uscita a contatti "puliti" che possono essere utilizzati per staccare il carico al verificarsi del sovraccarico, anche se tale applicazione è sensata solo quando si tratti di carichi che una volta sconnessi e ricollegati non si riaccendono automaticamente ma vanno riarmati manualmente.

SCHEMA ELETTRICO

Andiamo a vedere bene di cosa si tratta, studiando il circuito con la premessa che il rilevamento della corrente avviene per induzione elettromagnetica, introducendo uno dei fili della linea elettrica da monitorare nel foro che il sensore (prodotto dalla ZHT, www.lcsc.com) ha appositamente al centro del corpo; è necessario che passi un filo soltanto, perché se ci passasse una trecciola, il campo elettromagnetico prodotto dalla corrente entrante annullerebbe o contrasterebbe quasi totalmente quello della corrente uscente, ossia di ritorno.

Stabilito ciò, andiamo ad analizzare lo schema elettrico, nel quale il sensore è chiamato U1 ed essendo sostanzialmente un solenoide al cui interno passerà il filo elettrico del quale rilevare il flusso di corrente, produrrà una tensione alternata ed erogherà corrente alternata al circuito seguente; la tensione indotta ai capi del solenoide è direttamente proporzionale al flusso magnetico generato dal conduttore e questo flusso è direttamente proporzionale sia alla corrente che percorre il cavo (la quale determina l'induzione elettromagnetica B) sia all'angolo formato dalle linee di forza del campo elettromagnetico che essa genera (α) secondo la formula generica:

$$\Phi = B \times S \times \cos\alpha$$

dove Φ è il flusso magnetico ed S la superficie di ciascuna spira della bobina. La tensione derivante ai capi dell'intera bobina sarà la somma di quelle delle singole spire, il che nel nostro caso permette di disporre di un valore sufficiente da essere rad-drizzato e rilevato dal circuito.

Tutto questo per spiegare come il sensore induttivo sia sensibile all'induzione elettromagnetica determinata dalla corrente che fluisce nel cavo elettrico da monitorare.

Notate che per funzionare correttamente, il sensore richiede una resistenza di carico ai propri capi, giacché la tensione per una determinata corrente

nel conduttore che passa all'interno della bobina dipende dal valore di tale resistore; il trasduttore si comporta sostanzialmente da trasformatore di corrente, ossia è assimilabile a un trasformatore elettrico che permette di stabilire un rapporto di corrente fra il primario, che nello specifico è il cavo elettrico che verrà introdotto nel foro, ed il secondario, che è la bobina contenuta nel componente. Affinché venga rispettata la modalità di trasformazione di corrente, con un rapporto di trasformazione dichiarato di 1/1.000 (quindi per 5A nominali specificati dal costruttore si ottengono sul secondario 5 mA) bisogna applicare ai contatti 1 e 2 un resistore da 1000 ohm, che nel nostro circuito è R6.

Il componente garantisce una linearità dello 0,2% entro il campo di misura e una tensione di isolamento di 4.500V, a temperatura operativa compresa fra -40 e + 85 °C.

Siccome la tensione ai capi del sensore è alternata, abbiamo previsto alla sua uscita un ponte a diodi configurato come il classico raddrizzatore di Graetz, dove per ciascuna semionda conduce una coppia di diodi e rimane interdetta l'altra; nello specifico, in semionda positiva (polarità positiva sul contatto 1 del sensore) D1 conduce e porta la corrente a caricare il condensatore elettrolitico C1, entrando dal + e uscendo dall'elettrodo -, per poi chiudersi sull'estremo 2 della bobina attraverso D4. In semionda negativa (polarità positiva sul contatto 2 del sensore) C1 viene caricato con la stessa polarità (è questo il vantaggio del ponte di Graetz...) perché D1 e D4 sono ora interdetti, la corrente entra nell'anodo del diodo D5, percorre l'elettrolitico e ritorna al contatto 1 attraverso il diodo D2.

Per effetto di ciò, quando nel cavo monitorato fluisce una corrente di intensità sufficiente a determinare ai capi del sensore una tensione alternata che in valore picco-picco supera 1,2V (ossia la somma delle tensioni di soglia delle due coppie di diodi componenti il ponte di Graetz...), il condensatore elettrolitico C1 viene caricato e la tensione ai suoi capi, opportunamente ridotta dal partitore resistivo formato da R1 e R7 che ne dimezza l'ampiezza, raggiunge l'ingresso non-invertente dell'operazionale U2b, che funziona come amplificatore non-invertente senza polarizzazione centrale ed amplifica quindi solo le tensioni positive: per i nostri scopi, questo va più che bene, dato deve operare da amplificatore in continua.

La differenza di potenziale tra il piedino 7 dell'U2b e la massa è pari al doppio di quella presente ai capi della resistenza R7, giacché il guadagno in

tensione (Gv) dell'operazionale nella configurazione circuitale adottata è pari a:

$$Gv = (R5+R9) / R9$$

ed essendo il valore della R5 uguale a quello della R9, l'amplificazione è, pertanto, 2.

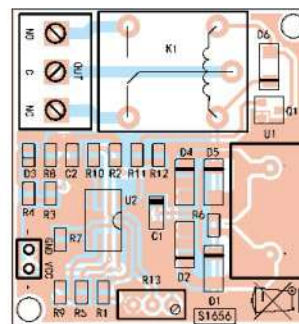
La tensione così amplificata viene confrontata nel comparatore U2a (l'altra metà dell'integrato U2, un comune LM358), cui arriva al pin 3, con il potenziale di riferimento che il trimmer multigiri R13 porta al piedino 2; questo potenziale è ricavato dalla linea di alimentazione in continua del circuito (morsetteria +/-PWR) e consente di stabilire la soglia di intervento del circuito, vale a dire l'assorbimento di corrente superato il quale il relé verrà fatto eccitare. Il trimmer è inserito in una cascata di partitori resistivi di tensione con filtro capacitivo (implementato dal condensatore C2) che ha lo scopo di ridurre la tensione e limitarne la dipendenza dalle fluttuazioni della tensione di alimentazione, allo scopo di evitare che il relé possa dissecarsi dopo essere stato attivato e viceversa, per una fluttuazione della tensione di alimentazione o l'introduzione di un disturbo nel circuito. In pratica l'accorgimento è finalizzato ad evitare il pendolarismo del relé intorno al valore di soglia della corrente assorbita e perciò a rendere stabile la commutazione, cosa utile perché il comparatore U2a non ha isteresi. Quando si porta a livello alto, l'uscita del comparatore polarizza la base del transistor NPN Q1 attraverso la resistenza R11 ed il collettore dell'NPN

CARATTERISTICHE TECNICHE

 Tensione di alimentazione: 12 Vcc	 Regolazione soglia potenza: 0+1.150 W @ 230Vca
 Corrente stabilita: 50 mA	 Uscita a relé con scambio completo
 Regolazione soglia corrente: 0+5 A	 Tensione e corrente d'uscita in ca 10 A / 250 Veff
	 Tensione e corrente d'uscita in cc: 10 A / 30 V

Elenco Componenti:

R1, R7, R8: 100 kohm (0805)
 R2, R3: 1 kohm (0805)
 R4: 4,7 kohm (0805)
 R5, R9, R10, R11, R12: 10 kohm (0805)
 R13: Trimmer multigiri 10 kohm
 C1: 10 μ F 25 VL tantalio (B)
 C2: 100 nF ceramico (0805)
 D1, D2, D4, D5, D6: LL4148
 D3: LED rosso (0805)
 Q1: BC817
 U1: ZHT103
 U2: LM358D
 K1: Relé 12V/10A SRD-12VDC-SL-C



Varie:

- Strip maschio 2 vie
- Morsetto 3 vie passo 5 mm

- Circuito stampato S1656
 (45x40 mm)

si porta a un potenziale circa uguale a quello di massa, alimentando la bobina del relé e facendo scattare l'equipaggio mobile di quest'ultimo. Notate il diodo D6 posto in parallelo alla bobina del relé, la che funge da protezione della giunzione base-collettore del transistor dall'inversione di polarità che si verifica ogni qualvolta il BJT viene lasciato interdire, allorché cesserebbe bruscamente il flusso di corrente nella bobina stessa; in questi casi, la natura induttiva farebbe in modo che la bobina tentasse di mantenere la corrente dell'istante precedente, generando un'extratensione di polarità inversa, che se non venisse spenta dal diodo (D6 conduce solo se il collettore diventa positivo rispetto alla linea di alimentazione del circuito), causerebbe il breakdown della giunzione di collettore e renderebbe il transistor inservibile, mandandolo perennemente in conduzione e vanificandone lo scopo.

Il relé ricadrà una volta che la corrente rilevata nel cavo passante scenderà al disotto della soglia impostata con il trimmer; riguardo alla soglia, va detto che con un'alimentazione di 12Vcc applicata ai contatti + e - PWR, è impostabile da 0, fino a 5 ampere, che in un impianto alimentato dalla rete-luce a 230Vca corrisponde a un tetto di 1,15 kW. La sensibilità sale portando il cursore verso massa, mentre in senso opposto, la protezione interverrà a potenze più elevate.

Concludiamo l'analisi del circuito con il LED D3, alimentato attraverso la resistenza limitatrice della corrente R4, che accendendosi indica quando il dispositivo è in funzione.

REALIZZAZIONE PRATICA

Bene, giunti a questo punto possiamo passare alla costruzione del circuito, per il quale abbiamo disegnato un PCB a doppia faccia del quale vi rendiamo disponibili per il download, nel nostro sito www.elettronica.in.it, le tracce lato rame affinché possiate autocostruirlo ricorrendo alla tecnica della fotoincisione.

Una volta inciso e forato, potete procedere al montaggio dei pochi componenti occorrenti. Trattandosi in buona parte di componentistica SMD (gli unici elementi a montaggio tradizionale sono il trimmer, la morsettiera, il relé ed il sensore induttivo U1) è richiesto un minimo di attrezzatura ed occorre prestare anche un po' di attenzione ed avere manualità; l'attrezzatura del caso consta almeno di un saldatore a punta molto fine, di filo di lega saldante (o pasta salda) del diametro massimo di 0,5 mm, della pasta flussante a media densità, una lente d'ingrandimento e una pinzetta per il corretto posizionamento dei componenti e per le verifiche finali delle saldature.

Bene, stabilito ciò possiamo procedere con la realizzazione: una volta procurato il tutto, prendete il circuito stampato, spalmate con la pasta flussante le piazzole destinate ai componenti SMD e saldate per prime le resistenze e i condensatori, poi il LED (quest'ultimo scaldatelo il meno possibile, ovvero tenete la punta del saldatore sui suoi elettrodi per il tempo strettamente necessario a sciogliere la lega saldante, altrimenti potrebbe deformarsi la resina trasparente da cui esce la luce prodotta) e il transistor Q1, che può essere saldato solo nel ver-

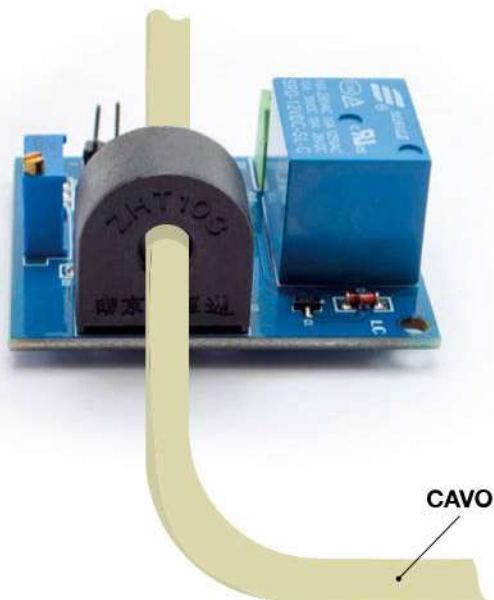
so giusto, per l'univoca disposizione dei terminali, mentre per il condensatore elettrolitico e il LED dovete fare riferimento all'orientamento indicato nel piano di montaggio che trovate in queste pagine. Procedete posizionando e saldando l'integrato LM358D, che andrà orientato come mostrato nel piano di montaggio, quindi inserite nei rispettivi fori e saldate la morsettiuera che renderà accessibili dall'esterno i contatti dello scambio del relé RL1. Inserite e montate quindi il relé, il quale entrerà nello stampato solamente nel verso previsto. L'ultimo componente elettronico che dovete montare è il sensore di corrente ZHT103, che va inserito dal lato componenti nelle piazzole previste allo scopo; non preoccupatevi più di tanto per l'orientamento, perché la disposizione dei fori è tale che il sensore entra soltanto nel verso giusto e in ogni caso nulla accade se invertite i piedini, perché il componente fornirà tensione e corrente alternate, raddrizzate dal circuito mediante il ponte a diodi già descritto.

Una volta completato il montaggio e verificatane l'esattezza, potete subito utilizzare il vostro circuito, rammentando che per farlo funzionare serve un'alimentazione stabilizzata per evitare imprecisioni nella commutazione e il pendolarismo del relé nell'intorno della soglia di corrente di commutazione. Ricordiamo che all'interno del sensore va fatto passare uno solo dei fili della linea elettrica da monitorare, perché altrimenti il segnale da esso ottenibile sarebbe pressoché nullo, giacché il verso della corrente che percorre uno dei fili della rete è opposto a quello della corrente che scorre nell'altro, in quanto in uno la corrente va e dall'altro torna.

UTILIZZO

Per l'uso, ricordiamo che il relé presente nel circuito può essere utilizzato per vari scopi: se vi basta una segnalazione e in generale un allarme che vi richiami a prestare attenzione al superamento della corrente (e quindi della potenza elettrica) potete pilotare con lo scambio del relé, utilizzando i contatti C ed NO dello scambio, un lampeggiante o una suoneria; se volete invece che il circuito intervenga sconnettendo uno o più utilizzatori, interrompete un filo della linea (può essere lo stesso che passa attraverso la bobina del sensore U1) e fatelo passare dallo scambio del relé, sempre utilizzando i contatti C ed NO.

Lo stesso scambio può tornare utile per implementare una particolare applicazione, vale a dire commutare un carico elettrico energivoro da una linea di alimentazione ad un'altra quando viene



Il cavo di cui rilevare la corrente passa nel foro del sensore.

superata la massima potenza assorbita: in tal caso dovete interrompere il filo di fase (L) dell'utilizzatore in questione e collegarlo al contatto C dell'uscita a relé, quindi portare al contatto NC il filo della fase della linea elettrica che lo deve alimentare in condizioni normali e all'NO la fase dell'alimentazione sostitutiva che interverrà al superamento del carico sulla rete.

Tenete comunque da conto che il circuito così com'è non ha temporizzazioni né isteresi, se non il ritardo determinato dalla presenza del condensatore elettrolitico C1, quindi se il distacco avviene per via dello stesso carico che il relé sconnette, l'interruzione inserisce poco dopo lo stesso utilizzatore, determinando un continuo commutare di RL1 fin quando non smette definitivamente il sovraccarico. Se invece dovete sconnettere utilizzatori che non ripartono da soli, il circuito va benissimo. Il relé può anche essere utilizzato per dare un segnale di distacco a vari apparati. □



Cosa occorre?

La scheda presentata in questo articolo è disponibile presso Futura Elettronica con il codice YM335 a Euro 9,00. Il prezzo si intende IVA compresa.

Il materiale va richiesto a:

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

Sicurezza e affidabilità senza compromessi!

Rendi più sicura la casa, il negozio e l'ufficio con questi set professionali per videosorveglianza e registrazione. Si installano facilmente e permettono la visione remota tramite PC e smartphone connessi a Internet.

SET COMPLETO VIA CAVO

Il set comprende un DVR ibrido a 4 ingressi che può funzionare sia in modalità HD (AHD / CVI / TVI) che in modalità analogica, completo di mouse, quattro telecamere Bullet da esterno da 2 Megapixel con illuminatori IR, due alimentatori da 12 VDC (uno per le telecamere e uno per il DVR), quattro cavi video da 14 metri e cavi di alimentazione per telecamere. L'hard disk è acquistabile separatamente.

VIA CAVO



€175,00

Cod. VIDEOCAMSET1

WiFi

App
gratuite
per Android
e iOS



€269,00

Cod. VIDEOCAMSET2

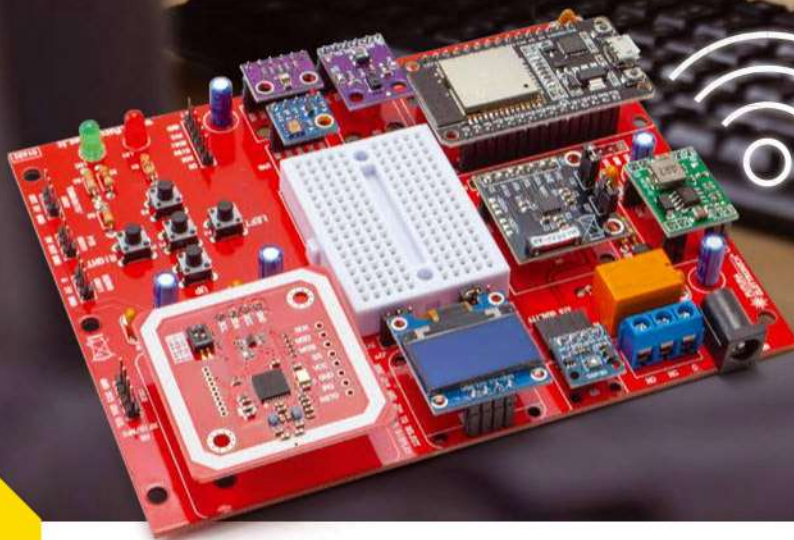


SET COMPLETO WIRELESS

Set composto da un NVR Wi-Fi a 4 canali, quattro telecamere IP IR wireless, un mouse USB, cinque alimentatori 12 Vdc (quattro per le telecamere e uno per l'NVR), quattro antenne per le telecamere, due antenne per l'NVR, un cavo di rete da 1 metro. L'hard disk è acquistabile separatamente.

DEMOBOARD ESP32

di DAVIDE SCULLINO



Chiudiamo la panoramica sulla scheda di prototipazione basata sul popolare modulo WiFi, con ulteriori applicazioni. Terza puntata.

N

elle due puntate precedenti abbiamo introdotto la demoboard per il famoso e diffusissimo SoC ESP32 della Espressif, che nello specifico è montato a bordo di una scheda con ESP32 commercializzata da www.futurashop.it con il codice YB555.

Oltre ad avervi proposto alcuni sketch scritti da noi per introdurvi nel mondo dello sviluppo con questo hardware, in ottica IoT; abbiamo descritto nel dettaglio firmware e creato degli esempi che abbiamo iniziato a presentarvi nella prima e nella seconda puntata. Adesso proseguiamo su questa strada proponendo gli ultimi esempi applicativi che utilizzano l'hardware di bordo e i moduli collegabili esternamente tramite gli appositi connettori.

Nello specifico andiamo a utilizzare questi componenti di bordo e moduli esterni:

- pulsanti UP, DOWN, LEFT, RIGHT, OK;
- Servomotore;
- Display OLED 0,96";
- Relé 5V (RL1);
- LED rosso (LD1);
- LED verde (LD2).

Ricordiamo che il cuore della demoboard è la scheda di sviluppo e prototipazione YB555 basata sull'ESP32 (ESP-WROOM-32) che integra un Wi-Fi 802.11 b/g/n, un Bluetooth dual-mode (classico e BLE) e 25 GPIO. Supporta una velocità di trasmissione dati fino a 150 Mbps, ha una potenza di uscita in antenna di 20,5 dBm e un consumo in modalità sleep di soli 5 μ A.

INTERAZIONE LED E PULSANTI

Iniziamo con il vedere un'applicazione nella quale utilizziamo i pulsanti a bordo della scheda e ne visualizziamo la condizione sul display OLED on-board (cod. OLED096C oppure cod. OLEDGVSCSD reperibili on-line su www.futurashop.it); questo esempio applicativo fornisce un'idea di come si possono utilizzare e integrare i pulsanti in un'applicazione vera e propria, ottenendo un feedback sul display oppure determinate azioni.

Lo sketch da caricare in questo caso si chiama OLED_Pulsanti.ino e una volta mandato in esecuzione sulla scheda ATmega2560 che governa la demoboard, sul display viene indicato di volta in volta quale pulsante è stato premuto fra quelli disposti a joystick, ossia UP, DOWN, LEFT, RIGHT e

OK (P1, P2, P3, P4, P5). Inoltre il codice di esempio permette di associare la pressione di uno dei predetti pulsanti all'accensione dei LED a bordo della scheda; per l'esattezza, se viene premuto LEFT o RIGHT si accenderà il LED verde (LD2), mentre se si preme UP o DOWN verrà comandata l'accensione del LED rosso (LD1). Nel caso venga premuto il pulsante OK si accenderanno entrambi i LED.

Questa applicazione e il codice che segue, proposto nel **Listato 1**, possono essere integrate in altre per ottenere un modulo di gestione dei pulsanti e per collegare i relativi input a determinate azioni, che possono essere non solo quelle descritte, ma anche l'attivazione di altre uscite, non ultima quella a relé.

Per l'utilizzo dello sketch dovete copiare la cartella "libraries" che si trova negli esempi scaricabili dal nostro sito www.elettronica.in fra i file di questo numero della rivista, sotto "/Documenti"/Arduino", quindi aprire l'IDE di Arduino e caricare il codice di esempio disponibile nella cartella "OLED_Pulsanti".

CONTROLLO SERVOCOMANDI

Passiamo adesso a un secondo esempio applicativo che è molto interessante perché riguarda la gestione dei servo analogici (da modellismo, robotica ecc.) attraverso la nostra demoboard; per implementarla utilizziamo l'uscita dedicata ai servocomandi, che sul circuito stampato è quella siglata CNU11, a tre poli: S (1) 5V (2) e GND (3). Si tratta della connessione standard per i connettori a passo 2,54 mm utilizzati nei servo, come ad esempio quello reperibile presso la Futura Elettronica (www.futurashop.it) con il codice prodotto SERVO206, con coppia di 1,5kg, rotazione: 160°, velocità a 4,8V: 0,12 s./60°.

Per la gestione dei servo abbiamo scritto lo sketch d'esempio OLED_Servo.ino, il quale, quando viene eseguito, imposta la demoboard in modo che il servocomando collegato al connettore CNU11 ruoterà prima in senso orario di 180° e poi di altrettanti gradi in senso antiorario; questo codice esegue quindi un test di funzionamento dei servocomandi, facendo compiere al loro albero una rotazione completa, divisa in due passaggi che corrispondono alle semirotazioni opposte rispetto alla posizione di riposo (0 gradi). Per l'utilizzo dello sketch dovete copiare la cartella "libraries" che si trova negli esempi scaricabili dal nostro sito www.elettronica.in fra i file di questo numero della rivista, sotto "/Documenti"/Arduino", quindi aprire l'IDE di Arduino e caricare uno dei codici di esempio disponibili nelle cartelle "OLED_Servo" oppure



→ La sezione pulsanti e LED utilizzata nell'esempio.

↓ Listato 1

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
// The pins for I2C are defined by the Wire-library.
// On an arduino UNO:      A4(SDA), A5(SCL)
// On an arduino MEGA 2560: 20(SDA), 21(SCL)
// On an arduino LEONARDO:  2(SDA),  3(SCL), ...
#define OLED_RESET      -1 // Reset pin # (or -1 if sharing Arduino reset pin)
#define SCREEN_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

#define NUMFLAKES      10 // Number of snowflakes in the animation example

#define LOGO_HEIGHT     16
#define LOGO_WIDTH      16

const int P_UP = 23;
const int P_DOWN = 4;
const int P_LEFT = 16;
const int P_RIGHT = 17;
const int P_OK = 5;
const int LEDV = 26;
const int LEDR = 25;

void setup() {
  Serial.begin(9600);

  // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
  if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }

  // Show initial display buffer contents on the screen --
  // the library initializes this with an Adafruit splash screen.
  display.display();
  delay(2000); // Pause for 2 seconds

  // Clear the buffer
  display.clearDisplay();
  display.setTextSize(2); // Draw 2X-scale text
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(5, 0);
  display.print("*** Test ***");
  display.setCursor(20, 30);
  display.print(" PUSH ");

  // Show the display buffer on the screen. You MUST call display() after
  // drawing commands to make them visible on screen!
  display.display();
  delay(2000);
  // display.display() is NOT necessary after every single drawing command,
  // unless that's what you want...rather, you can batch up a bunch of
  // drawing operations and then update the screen all at once by calling
  // display.display(). These examples demonstrate both approaches...

  /*
  // Invert and restore display, pausing in-between
  display.invertDisplay(true);
  delay(1000);
  display.invertDisplay(false);
  delay(1000);
  */

  pinMode(P_UP, INPUT_PULLUP);
  pinMode(P_DOWN, INPUT_PULLUP);
  pinMode(P_LEFT, INPUT_PULLUP);
  pinMode(P_RIGHT, INPUT_PULLUP);
  pinMode(P_OK, INPUT_PULLUP);
  pinMode(LEDV, OUTPUT);
  pinMode(LEDR, OUTPUT);
}

void loop()
{
  display.clearDisplay();
  display.setTextSize(2); // Draw 2X-scale text
  display.setTextColor(SSD1306_WHITE);
```


↓ Listato 1 – continua

```
switch (Ctrl_Push())
{
  case 1:
    Serial.println("Push UP");
    display.setCursor(50, 30);
    display.print("UP");
    digitalWrite(LED_R, HIGH);
    break;
  case 2:
    Serial.println("Push DOWN");
    display.setCursor(40, 30);
    display.print("DOWN");
    digitalWrite(LED_R, HIGH);
    break;
  case 3:
    Serial.println("Push LEFT");
    display.setCursor(40, 30);
    display.print("LEFT");
    digitalWrite(LED_V, HIGH);
    break;
  case 4:
    Serial.println("Push RIGHT");
    display.setCursor(40, 30);
    display.print("RIGHT");
    digitalWrite(LED_V, HIGH);
    break;
  case 5:
    Serial.println("Push OK");
    display.setCursor(50, 30);
    display.print("OK");
    digitalWrite(LED_R, HIGH);
    digitalWrite(LED_V, HIGH);
    break;
  default:
    display.setCursor(40, 30);
    display.print("----");
    digitalWrite(LED_V, LOW);
    digitalWrite(LED_R, LOW);
    break;
}

display.display();
delay(500);
}

int Ctrl_Push()
{
  int push = 0;
  if (digitalRead(P_UP) == LOW) push = 1;
  if (digitalRead(P_DOWN) == LOW) push = 2;
  if (digitalRead(P_LEFT) == LOW) push = 3;
  if (digitalRead(P_RIGHT) == LOW) push = 4;
  if (digitalRead(P_OK) == LOW) push = 5;
  return push;
}
```

Il connettore per collegare i servo.



"Servo". Il codice completo per la gestione dei servo-comandi è riportato nel **Listato 2**.

UTILIZZO DELLA CONNESSIONE WIFI

Il modulo ESP32 che governa l'intera demoboard si distingue per una caratteristica: l'avere a bordo la connessione WiFi ed è questa la ragione per cui l'abbiamo scelta: avere a bordo una connettività per l'accesso a Internet e per affacciare alla Rete le nostre applicazioni, in perfetta ottica IoT.

Per quello che riguarda la sezione ricevente WiFi, opera in un range di frequenza compreso fra 2,4 GHz e 2,5 GHz e le sue caratteristiche tecniche principali sono le seguenti:

- conformità IEEE 802.11 b/g/n HT40;
- stack in banda base e LwIP;
- modalità Sniffer / Station / SoftAP e Wi-Fi direct mode;
- massimo data-rate di 150 Mbps@11n HT40, 72 Mbps@11n HT20, 54 Mbps@11g e 11 Mbps@11b;
- massima potenza di trasmissione 19,5 dBm@11b, 16,5 dBm@11g, 15,5 dBm@11n;
- sensibilità in ricezione: -97 dBm.

Il transceiver è anche predisposto per supportare le modalità di protezione della connessione WEP, WPA/WPA2 PSK/Enterprise e la crittografia con accelerazione hardware AES / SHA2 / ECC / RSA-4096. Andiamo dunque a vedere l'esempio applicativo che utilizza questo hardware, il quale prevede che il dispositivo si conatterà alla propria rete WiFi per eseguire un test di connettività.

A connessione avvenuta, verrà mostrato l'indirizzo IP assegnato al modulo ESP sul display. Al termine il modulo rimarrà in attesa di una connessione da parte di un client interno alla rete, sulla porta TCP 8088. A connessione avvenuta un messaggio fornirà tale informazione e ogni carattere scritto sul terminale del client verrà mostrato sul display, nei limiti di capienza del display. Se a connessione avvenuta si preme uno dei cinque pulsanti a bordo della scheda, il client riceverà sul terminale l'indicazione del pulsante premuto.

Lo sketch di test si chiama OLED_WiFi.ino e si trova sotto la cartella degli esempi, ovvero sotto "Documenti"/Arduino; aprite l'IDE di Arduino e caricate lo sketch, quindi andate a modificare le variabili "apn" e "ssid" con i parametri della propria rete, perché queste ultime avranno parametri generici in quanto non possiamo sapere a quale rete il codice sarà applicato.

In particolare, nel codice di esempio, che vedete

↓ Listato 2

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

//for Servo
#include <Servo.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
// The pins for I2C are defined by the Wire-library.
// On an arduino UNO:      A4(SDA), A5(SCL)
// On an arduino MEGA 2560: 20(SDA), 21(SCL)
// On an arduino LEONARDO:  2(SDA),  3(SCL), ...
#define OLED_RESET     -1 // Reset pin # (or -1 if sharing Arduino reset pin)
#define SCREEN_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

#define NUMFLAKES 10 // Number of snowflakes in the animation example

#define LOGO_HEIGHT 16
#define LOGO_WIDTH 16

//for Servo
static const int servoPin = 18;
Servo servo1;

void setup()
{
  Serial.begin(9600);

  // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
  if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }

  // Show initial display buffer contents on the screen --
  // the library initializes this with an Adafruit splash screen.
  display.display();
  delay(2000); // Pause for 2 seconds

  // Clear the buffer
  display.clearDisplay();
  display.setTextSize(2); // Draw 2X-scale text
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(5, 0);
  display.print("** Test **");
  display.setCursor(10, 30);
  display.print(" Servo ");

  // Show the display buffer on the screen. You MUST call display() after
  // drawing commands to make them visible on screen!
  display.display();
  delay(2000);
  // display.display() is NOT necessary after every single drawing command,
  // unless that's what you want...rather, you can batch up a bunch of
  // drawing operations and then update the screen all at once by calling
  // display.display(). These examples demonstrate both approaches...

  /*
  // Invert and restore display, pausing in-between
  display.invertDisplay(true);
  delay(1000);
  display.invertDisplay(false);
  delay(1000);
  */
  servo1.attach(servoPin);
}

void loop()
{
  for(int posDegrees = 0; posDegrees <= 180; posDegrees++)
  {
    servo1.write(posDegrees);
    Serial.println(posDegrees);
    delay(20);
  }

  for(int posDegrees = 180; posDegrees >= 0; posDegrees--)
  {
    servo1.write(posDegrees);
    Serial.println(posDegrees);
    delay(20);
  }
}
```


↓ Listato 3

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

//for WiFi
#include <WiFi.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
// The pins for I2C are defined by the Wire-library.
// On an arduino UNO:      A4(SDA), A5(SCL)
// On an arduino MEGA 2560: 20(SDA), 21(SCL)
// On an arduino LEONARDO:  2(SDA),  3(SCL), ...
#define OLED_RESET    -1 // Reset pin # (or -1 if sharing Arduino reset pin)
#define SCREEN_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

#define NUMFLAKES 10 // Number of snowflakes in the animation example

#define LOGO_HEIGHT 16
#define LOGO_WIDTH 16

//for WiFi
const char* ssid      = "NOME APN";
const char* password = "PASSWORD APN";
/* create a server and listen on port 8088 */
WiFiServer server(8088);

const int P_UP = 23;
const int P_DOWN = 4;
const int P_LEFT = 16;
const int P_RIGHT = 17;
const int P_OK = 5;
const int LEDV = 26;
const int LEDR = 25;

void setup() {
  Serial.begin(9600);

  // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
  if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }

  // Show initial display buffer contents on the screen --
  // the library initializes this with an Adafruit splash screen.
  display.display();
  delay(2000); // Pause for 2 seconds

  // Clear the buffer
  display.clearDisplay();
  display.setTextSize(2); // Draw 2X-scale text
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(5, 0);
  display.print("** Test **");
  display.setCursor(5, 30);
  display.print("  WiFi  ");

  // Show the display buffer on the screen. You MUST call display() after
  // drawing commands to make them visible on screen!
  display.display();
  delay(2000);
  // display.display() is NOT necessary after every single drawing command,
  // unless that's what you want...rather, you can batch up a bunch of
  // drawing operations and then update the screen all at once by calling
  // display.display(). These examples demonstrate both approaches...

  /*
  // Invert and restore display, pausing in-between
  display.invertDisplay(true);
  delay(1000);
  display.invertDisplay(false);
  delay(1000);
  */
  pinMode(P_UP, INPUT_PULLUP);
  pinMode(P_DOWN, INPUT_PULLUP);
  pinMode(P_RIGHT, INPUT_PULLUP);
  pinMode(P_LEFT, INPUT_PULLUP);
}
```


⬇ Listato 3 - continua

```
pinMode(P_OK, INPUT_PULLUP);

Serial.begin(9600);
OLED_Message("Connecting to ", ssid);
Serial.print("Connecting to ");
Serial.println(ssid);
/* connecting to Wifi */
WiFi.begin(ssid, password);
/*wait until ESP32 connect to Wifi*/
while (WiFi.status() != WL_CONNECTED)
{
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("Wifi connected with IP address:");
");
String testo = WiFi.localIP().toString();
OLED_Message("Wifi connected", testo);
Serial.println(WiFi.localIP());
delay(2000);
/* start Server */
server.begin();
}

void loop()
{
    /* listen for client */
    WiFiClient client = server.available();
    uint8_t data[30];
    if (client)
    {
        OLED_Message("Client Connected", "");
        Serial.println("new client");
        display.setCursor(0, 20);
        /* check client is connected */
        while (client.connected())
        {
            switch (Ctrl_Push())
            {
                case 1:
                    Serial.print("Push UP");
                    client.write("\nUP");
                    break;
                case 2:
                    Serial.println("Push DOWN");
                    client.write("\nDOWN");
                    break;
                case 3:
                    Serial.println("Push LEFT");
                    client.write("\nLEFT");
                    break;
                case 4:
                    Serial.println("Push RIGHT");
                    client.write("\nRIGHT");
                    break;
                case 5:
                    Serial.println("Push OK");
                    client.write("\nOK");
                    break;
                default:
                    break;
            }
        }
        Attendi_Push();
        if (client.available())
        {
            int len = client.read(data, 30);
            if(len < 30)
            {
                data[len] = '\0';
            }
            else
            {
                data[30] = '\0';
            }
            Serial.print("client sent: ");
            Serial.println((char *)data);
            display.print((char *)data);
            display.display();
        }
    }
    OLED_Message("Client Disconnected", "");
}
```


Il servocomando
SERVO206.



riportato nel **Listato 3**, dovete modificare le righe seguenti:

```
const char* ssid      = "NOME APN";  
const char* password = "PASSWORD APN";
```

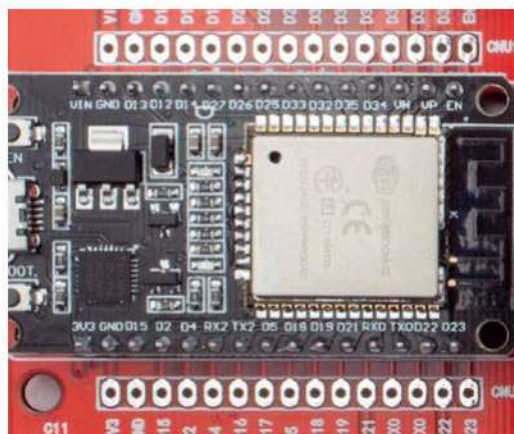
inserendo al posto del testo virgolettato NOME APN il nome del vostro access point e al posto di PASSWORD APN la password (o passkey) che utilizzate per l'accesso alla rete WiFi.

SCANSIONE DEL BUS I²C

L'ultimo esempio applicativo che proponiamo è in realtà un'utility, giacché il relativo codice permette di interrogare il bus I²C per rilevare cosa è affacciato ad esso e fornire sul display o tramite Serial Monitor l'indirizzo di tutti i dispositivi collegati ad esso.

Come vedete nel **Listato 4**, il codice fa uso della libreria wire.h, che permette di comunicare con i dispositivi I²C/TWI attraverso le due linee SDA (linea dati) e SCL (Serial Clock) che si trovano sia a bordo

Il transceiver
WiFi del modulo
ESP32 è una
periferica
utilizzabile a sè.



Listato 3 - continua

```
void OLED_Message(String text_display1,  
String text_display2)  
{  
  int lunghezza = 0;  
  display.clearDisplay();  
  display.setTextSize(1);  
  // Draw 2X-scale text  
  display.setTextColor(SSD1306_WHITE);  
  display.setCursor(0, 0);  
  lunghezza = text_display1.length();  
  lunghezza=21-lunghezza;  
  lunghezza=lunghezza/2;  
  for (int i=0;i<lunghezza;i=i+1)  
  {  
    display.print(" ");  
  }  
  display.print(text_display1);  
  display.setCursor(0, 20);  
  lunghezza = text_display2.length();  
  lunghezza=21-lunghezza;  
  lunghezza=lunghezza/2;  
  for (int i=0;i<lunghezza;i=i+1)  
  {  
    display.print(" ");  
  }  
  display.print(text_display2);  
  display.display();  
  delay(1500);  
}  
  
int Ctrl_Push()  
{  
  int push = 0;  
  if (digitalRead(P_UP)==LOW) push = 1;  
  if (digitalRead(P_DOWN)==LOW) push = 2;  
  if (digitalRead(P_LEFT)==LOW) push = 3;  
  if (digitalRead(P_RIGHT)==LOW) push = 4;  
  if (digitalRead(P_OK)==LOW) push = 5;  
  return push;  
}  
  
void Attendi_Push()  
{  
  while ((digitalRead(P_UP)==LOW) ||  
        (digitalRead(P_DOWN)==LOW) ||  
        (digitalRead(P_LEFT)==LOW) ||  
        (digitalRead(P_RIGHT)==LOW) ||  
        (digitalRead(P_OK)==LOW)) delay(10);  
}
```



Cosa occorre?

La demoboard per ESP32 (cod. FT1632) è disponibile in scatola di montaggio a Euro 25,00 e comprende tutti i componenti ad eccezione dei sensori, display e moduli. Questi sono venduti separatamente.

In particolare sono disponibili: il modulo con ESP32 (cod. YB555) a Euro 13,50, il display OLED (cod. OLED096C) a Euro 11,00, il sensore qualità dell'aria (cod. SGP30C02) a Euro 19,50, il sensore fulmini (cod. FT1324M) a Euro 25,00, il modulo gesture (cod. APDS9960BOUT) a Euro 7,50, il sensore UVB (cod. GYML8511) a Euro 8,00, il sensore di pressione (cod. BMP280) a Euro 4,50 e il modulo RFID (cod. PN532) a Euro 14,00.

I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

scheda, sia riportate sul connettore di espansione siglato I2C, che vale per i dispositivi connessi al bus esternamente alla scheda.

A partire da Arduino 1.0, la libreria si è allineata alle altre librerie di lettura/scrittura; per questo motivo, le funzioni `send()` e `receive()` originarie sono state sostituite con `read()` e `write()`.

Le versioni recenti della libreria `Wire` possono utilizzare i timeout per prevenire un blocco a fronte di determinati problemi sul bus, ma questi non sono abilitati per impostazione predefinita nelle versioni correnti (si consiglia di abilitare sempre questi timeout quando si utilizza la libreria `Wire`).

Caricate dunque l'ultima versione della libreria.

CONCLUSIONI

Bene, con questo esempio di utilizzo terminiamo la panoramica sulla demoboard ESP32, che in tre articoli vi ha presentato le possibilità di utilizzo dell'hardware, seppure non tutte quelle possibili, sia a bordo che connesso tramite i connettori dedicati. Ogni esempio è stato corredato di sketch per implementarlo e testarlo.

Ora lasciamo a voi e alla vostra inventiva l'implementazione di ulteriori applicazioni, che testerete e debuggerete sulla demoboard, per poi uscire dalla fase prototipale e realizzarle in maniera definitiva.

Buon lavoro!



↓ Listato 4

```
#include <Wire.h>

void setup()
{
    Wire.begin();

    Serial.begin(9600);
    while (!Serial);
    Serial.println("\nI2C Scanner");
}

void loop()
{
    byte error, address;
    int nDevices;
    address=0x76;

    Serial.println("Scanning...");

    nDevices = 0;
    for(address = 1; address < 127; address++ )
    {
        // The i2c_scanner uses the return value of
        // the Wire.endTransmission to see if
        // a device did acknowledge to the address.
        Wire.beginTransmission(address);
        error = Wire.endTransmission();

        if (error == 0)
        {
            Serial.print("I2C device found at address 0x");
            if (address<16)
                Serial.print("0");
            Serial.print(address,HEX);
            Serial.println(" !");

            nDevices++;
        }
        else if (error==4)
        {
            Serial.print("Unknown error at address 0x");
            if (address<16)
                Serial.print("0");
            Serial.println(address,HEX);
        }
    }
    if (nDevices == 0)
        Serial.println("No I2C devices found\n");
    else
        Serial.println("done\n");

    delay(5000);          // wait 5 seconds for next scan
}
```


OSCILLOSCOPIO DA LABORATORIO 2 CANALI-100 MHZ

cod. DS02C10
€ 298,00



CARATTERISTICHE TECNICHE OSCILLOSCOPIO

Display: TFT 7" - Frequenza di campionamento (in tempo reale): 1GSa/s (canale singolo), 500MSa/s (due canali)
Larghezza di banda del canale analogico: 100 MHz - Funzione FFT (Fast Fourier Transform) integrata - Profondità di memoria: 8MB
Software per l'analisi in tempo reale con il PC - Porta USB Host, USB Device

OSCILLOSCOPIO DA LABORATORIO 2 CANALI-100 MHZ CON GENERATORE FORME D'ONDA ARBITRARIE FINO A 25 MHZ



cod. DS02D10
€ 358,00

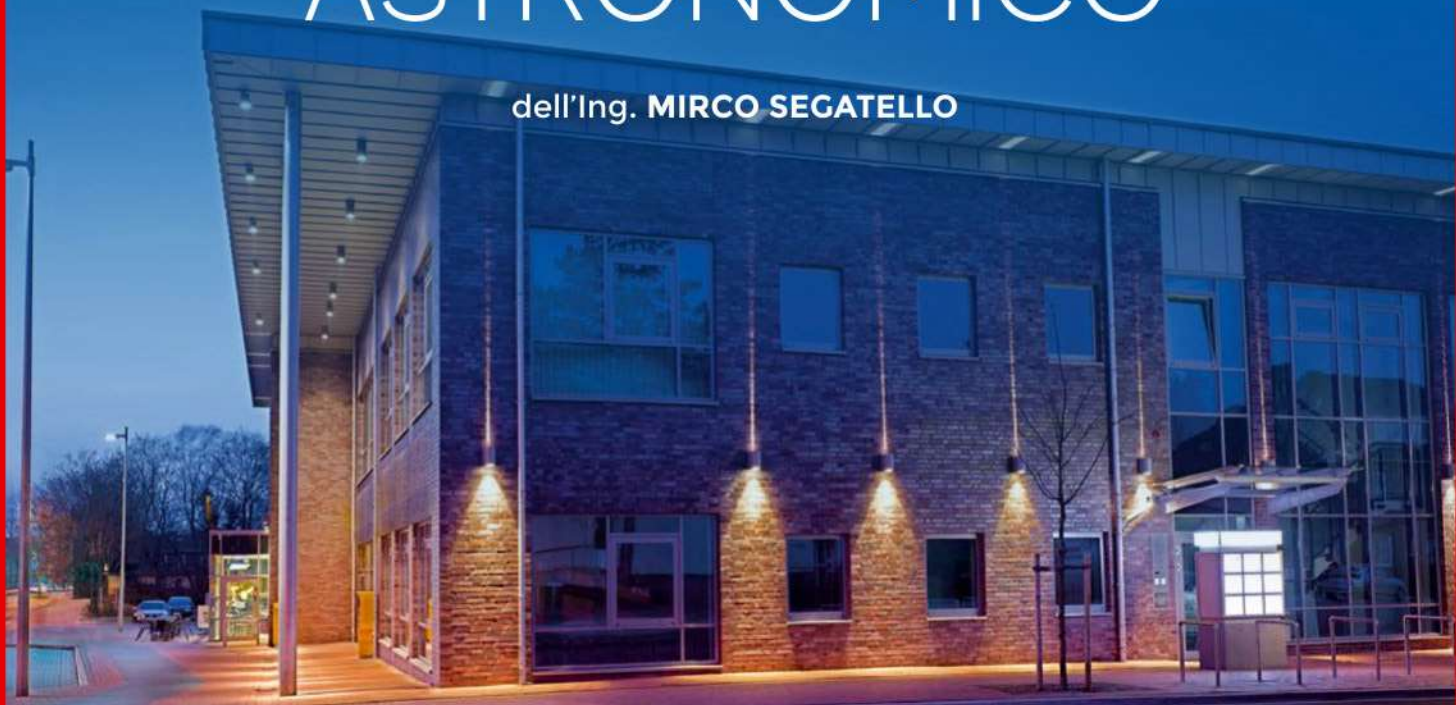
CARATTERISTICHE TECNICHE GENERATORE FORME D'ONDA

Numero canali: 1 - Sample Rate: 200 MSa/s - Risoluzione Verticale: 12 bit - Massima Frequenza di uscita: 25 MHz
Forma d'onda di uscita: sine, square, ramp, Exp, noise, DC - Forme d'onda arbitraria: Arb1, Arb2, Arb3, Arb4
Profondità forma d'onda: 4KSa - Ampiezza (range di uscita): 10mV~7Vp-p (alta impedenza) / 5mV~3.5Vp-p (50Ω)
DC offset: range $\pm 3,5$ V, alta impedenza / $\pm 1,75$ V, 50 Ω - Risoluzione: 100 μ V o 3 bits - Impedenza di uscita: 50 Ohm.

Le caratteristiche tecniche della parte relativa all'oscilloscopio sono le stesse del modello DS02C10.

INTERRUTTORE ASTRONOMICO

dell'Ing. MIRCO SEGATELLO



Permette di attivare un impianto di illuminazione esterno in base agli orari di alba e tramonto calcolati automaticamente ogni giorno dell'anno.

Non è inusuale gestire l'accensione e lo spegnimento automatico di un impianto di illuminazione artificiale per interni o esterni in determinate ore della giornata: basti pensare alle insegne di un esercizio pubblico, piuttosto che le luci esterne di un'abitazione o ancora l'illuminazione di spazi pubblici come strade, piazze e giardini. La prima soluzione che viene in mente è il classico interruttore orario, vale a dire un timer, sia di tipo meccanico che elettronico: esso consiste in un orologio che viene usato per attivare un contatto elettrico in determinate ore della giornata, prestabilite in base

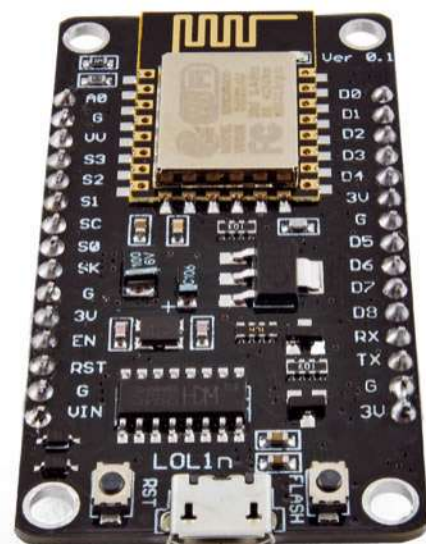


Fig. 1
Esempio di
interruttore orario e
crepuscolare.

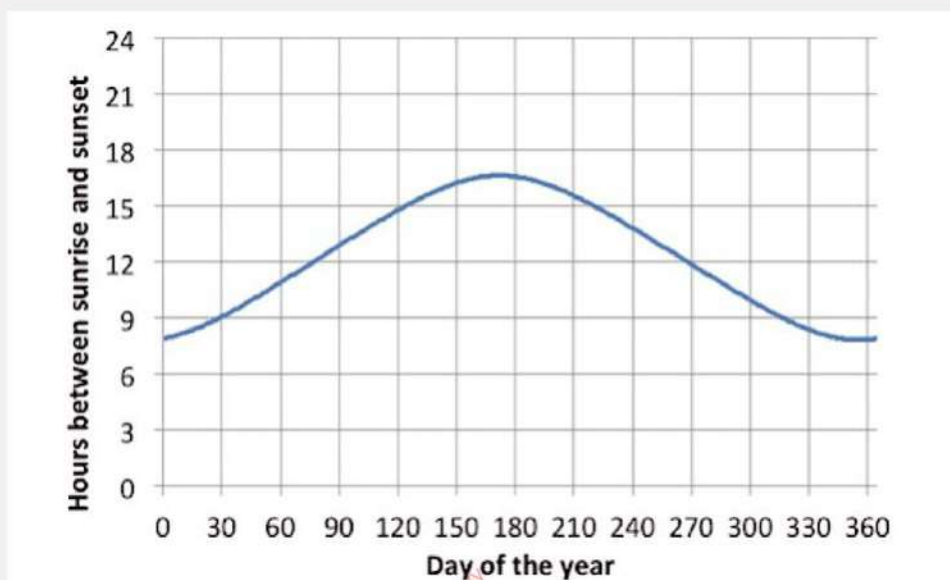


alle esigenze tecniche e funzionali, ma anche, vista la crisi energetica che ci attanaglia in questo periodo, sulla base di valutazioni economiche e strategiche finalizzate al contenimento dei consumi. Per come è stato concepito, l'accensione e lo spegnimento delle luci avvengono sempre alle stesse ore del giorno rendendolo di fatto poco adatto se l'impianto luce è esterno e quindi soggetto alla luce ambiente. Se accendessimo l'illuminazione esterna dalle ore 18 di sera alle ore 6 di mattina, potrebbe andare bene nel periodo invernale ma nel periodo estivo alle 18 il Sole è ancora alto rendendo inutile l'accensione delle luci.

In alcuni periodi dell'anno le luci sarebbero accese inutilmente mentre in altri rimarrebbero spente quando invece necessario.

Per superare questa limitazione sono stati inventati gli interruttori crepuscolari che gestiscono in automatico l'accensione/spegnimento in base alla reale intensità luminosa ambientale grazie ad un sensore di luminosità come ad esempio una fotoresistenza; vengono spesso usati nei condomini o nei negozi per l'accensione e lo spegnimento delle illuminazioni per lo più esterne e non necessitano di un orologio interno né di una batteria tampone per mantenere aggiornato l'orologio in assenza

Fig. 2
Andamento delle
ore di luce durante
l'anno.



di alimentazione elettrica. L'accensione delle luci avviene quando la luce solare inizia a scendere per poi spegnerle quando la luce dovesse ritornare sopra una determinata soglia, impostabile a piacere. Il risparmio in bolletta è evidente perché l'accensione e lo spegnimento avverrà sempre quando è effettivamente necessario. Il maggiore limite di questo interruttore è quello di non riuscire a discriminare eventi casuali come un temporale che, oscurando il cielo, comporta l'accensione indesiderata dell'impianto con conseguente aumento dei costi elettrici. La presenza di sporco o anche il solo naturale invecchiamento del sensore, possono influire significativamente sui periodi di accensione. La **Fig. 1** propone un esempio di interruttore orario e crepuscolare.

Per superare le limitazioni dei precedenti componenti si ricorre all'utilizzo di interruttori astronomici che sono in grado di gestire l'impianto luci in base alla situazione luminosa solare determinata matematicamente. Vediamo ora di capire in concreto di cosa si tratta e come possiamo costruire un semplice interruttore astronomico. Partiamo da una semplice constatazione: l'illuminazione solare diminuisce drasticamente al calare del Sole, ovvero al sopraggiungere del tramonto (in inglese sunset) mentre aumenta rapidamente al sorgere del Sole ovvero all'alba (in inglese sunrise). Conoscendo i momenti in cui avvengono l'alba e il tramonto saremo in grado di attivare le luci solo quando l'intensità luminosa sarà realmente bassa. Come tutti sapete, gli orari di alba e tramonto dipendono sia dal giorno dell'anno sia dalla zona della terra in cui ci troviamo, pertanto la loro determinazione non è così immediata. Il Sole sorge ad est e tramonta ad ovest descrivendo una traiettoria nella volta celeste più bassa nei periodi invernali e più alta nei periodi estivi, determinando assieme alla distanza dall'equatore le ore di luce e di buio a disposizione. L'andamento è pressoché sinusoidale ed è rappresentato nella **Fig. 2**.

A seconda di quanto il Sole si trova basso sull'orizzonte si definiscono tre diverse situazioni di alba e tramonto: civile, nautica e astronomica. Nell'alba civile il Sole è 6° sotto l'orizzonte, nell'alba nautica 12° sotto l'orizzonte e nell'alba astronomica 18° . Questo significa che se impostiamo il calcolo di alba e tramonto astronomico, le luci si attiveranno di meno rispetto l'alba e tramonto civile. In **Fig. 3** un grafico che esplica questo concetto.

DESCRIZIONE DELL'HARDWARE

Il nostro dispositivo dovrà essenzialmente disporre

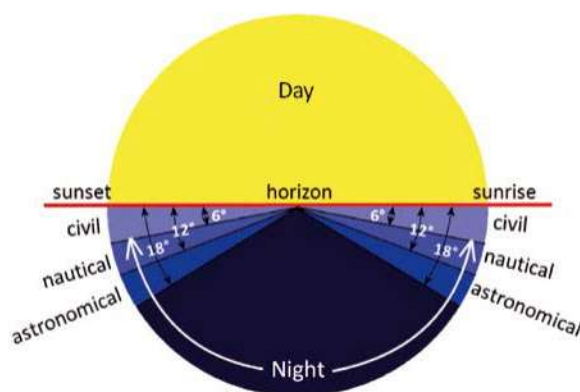
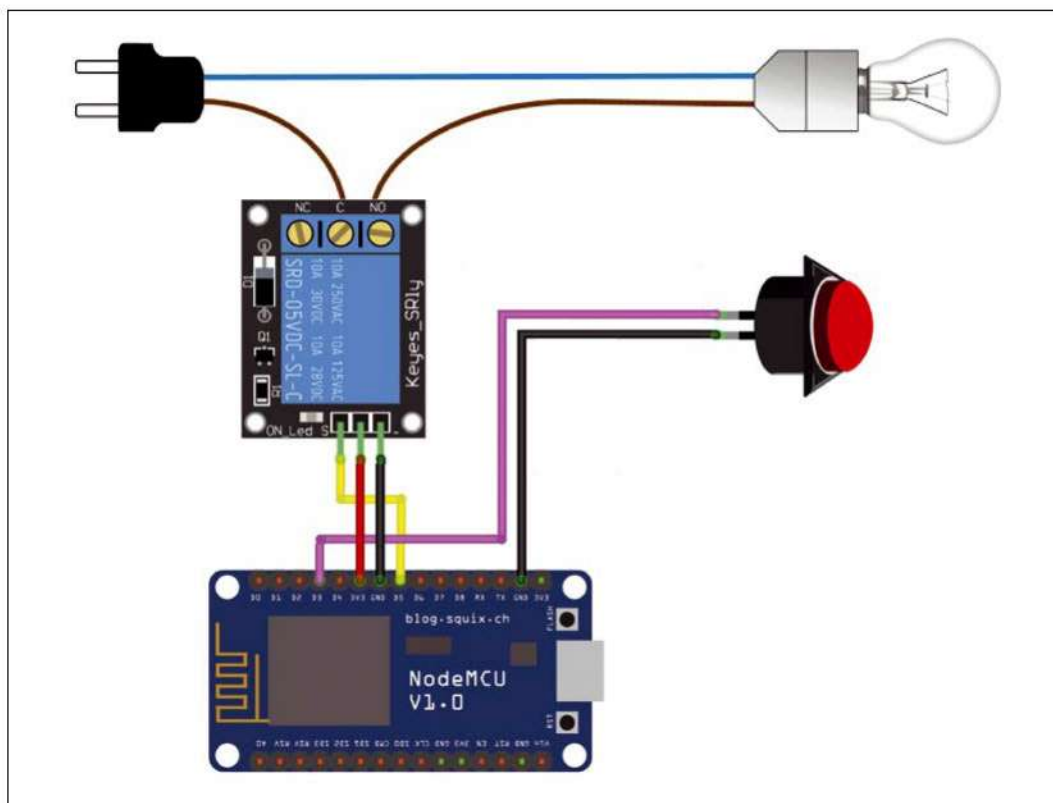


Fig. 3
Differenza
tra diversi
tipi di alba/
tramonto.

di un orologio in grado di tenere traccia dell'ora e della data, meglio con la possibilità di auto sincronizzarsi tramite uno dei tanti servizi online, in modo da evitare complicate funzioni di setup. Un altro vantaggio legato alla possibilità di accesso alla rete internet è che, in conseguenza di un blackout, alla riaccensione l'orologio si sincronizzerà in automatico senza la necessità di prevedere un sistema di backup a batteria. La sincronizzazione automatica garantirà anche un'ottima precisione nel lungo periodo. Tra le varie opzioni commerciali, una economica board con un microcontrollore ESP8266 fa al caso nostro: in particolare, per il nostro progetto abbiamo usato una scheda di nome NODE MCU fornita dalla Futura Elettronica e acquistabile on-line sul sito www.futurashop.it con il codice prodotto NODEMCUESP. Questa scheda dispone di un microcontrollore con buone prestazioni (ESP8266-12E) nel cui package (si tratta in realtà di un SoC=System On Chip) è incorporato un modulo transceiver WiFi con cui poter comunicare verso una rete esterna; inoltre può essere facilmente alimentata tramite il connettore micro USB attraverso il quale può anche essere programmata. Il modulo Wi-Fi è compatibile con lo standard 802.11 b/g/n a 2,4 GHz, dispone di stack TCP/IP integrato, potenza di trasmissione di 19,5 dBm, interfaccia dati (UART/HSPI/I²C ecc.) e antenna su PCB. Inoltre è programmabile con l'IDE di Arduino. Per completare il progetto è sufficiente aggiungere un qualunque pulsante normalmente aperto (che useremo per le funzioni di setup) ed un modulo a relé necessario per comandare l'impianto di illuminazione; anche quest'ultimo è disponibile presso la Futura Elettronica e si può acquistare on-line con il codice prodotto RELAY1CH oppure TRIGGERLL5V. Il collegamento elettrico tra i vari componenti del nostro sistema è riportato nella **Fig. 4**, che ne rappresenta lo schema di cablaggio.

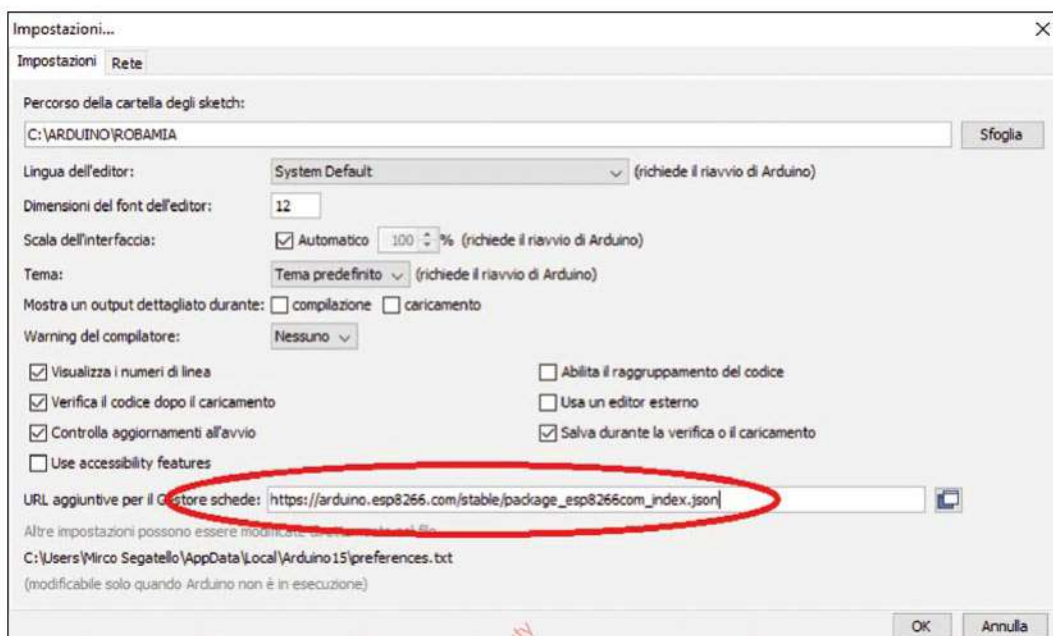
Fig. 4
Schema per il collegamento dei componenti.



Per l'alimentazione dell'insieme può essere usato un caricabatteria da smartphone con uscita micro USB. Fate molta attenzione al cablaggio dell'impianto di illuminazione, che normalmente alimen-

tato dalla tensione di rete a 230Vca, rammentando di operare esclusivamente dopo averlo privato della tensione e verificando attentamente i collegamenti prima di rialimentarlo.

Fig. 5
Installazione pacchetto aggiuntivo per ESP8266.



DESCRIZIONE DEL SOFTWARE

Occupiamoci ora del firmware necessario al corretto funzionamento del circuito; come ambiente di programmazione abbiamo scelto il classico Arduino IDE che è stato predisposto con l'installazione del pacchetto esp8266 by ESP8266 Community disponibile nel gestore schede dell'IDE.

Prima di avviare l'installazione occorre inserire l'URL del pacchetto (http://arduino.esp8266.com/stable/package_esp8266com_index.json) nel campo "URL aggiuntive per il Gestore Schede" nelle impostazioni di preferenza (**Fig. 5**).

Per implementare alcune funzioni ci siamo appoggiati a specifiche librerie facilmente recuperabili in rete che per comodità abbiamo allegato ai file di questo numero della rivista. La libreria **ESP8266WiFi** viene installata assieme al pacchetto aggiuntivo ESP e fornisce tutte le funzionalità di base per utilizzare il modulo WiFi interno al microcontrollore. Questa libreria, assieme alla libreria **WiFiManager** permette di implementare un sistema molto pratico per configurare l'accesso alla propria rete Wi-Fi senza dover preventivamente specificare le credenziali nel listato del software. Quando la scheda NODE MCU viene alimentata, se ci sono nella memoria interna del modulo ESP delle credenziali valide essa si conatterà alla rete WiFi, in caso contrario viene avviata una procedura di configurazione ed il modulo ESP si predispone in modalità di access point al quale potrete connettervi con il vostro smartphone.

Tramite un qualsiasi browser, inserendo l'indirizzo di default 192.168.4.1, avrete accesso ad un webserver che vi guiderà nella configurazione dell'accesso come visibile in **Fig. 6**; le credenziali saranno salvate in memoria per i successivi accessi. Nel caso vogliate connettervi ad un'altra rete potete azzerare la configurazione in memoria accendendo la scheda, mantenendo premuto il pulsante di setup.

Il nostro dispositivo dovrà connettersi alla rete internet per sincronizzare data e ora sfruttando i dati forniti dai server NTP (Network Time Protocol), operazione che avviene ad ogni accensione e ad intervalli regolari, anche se la mancata sincronizzazione per qualche settimana non influenza significativamente la precisione.

I dati forniti dal server NTP sono passati alla libreria TimeLib che si occupa, sfruttando il clock di sistema, di mantenere data e ora con estrema precisione, mentre la libreria Timezone permette di gestire il fuso orario ed il passaggio all'ora legale. L'orologio astronomico non tiene in considerazio-



Fig. 6
Procedura guidata di configurazione.

ne il cambio da ora solare a legale, ma abbiamo comunque voluto implementare questa funzione, per un'eventuale visualizzazione tramite il Serial Monitor dell'IDE di Arduino o un display esterno. Nelle righe di codice seguenti potete vedere i valori usati per impostare il corretto passaggio all'ora legale; ultima domenica di marzo (alle ore 2) e ultima domenica di ottobre (alle ore 3):

```
TimeChangeRule CEST = {"CEST", Last, Sun, Mar, 2, 120};  
TimeChangeRule CET = {"CET ", Last, Sun, Oct, 3, 60};  
Timezone CE(CEST, CET);
```

Le librerie **Wire** e **TM1650** sono state incluse nell'eventualità si volesse utilizzare un display a LED (basato sull'integrato TM1650) per la visualizzazione dell'ora. Per concludere useremo la libreria **Sunrise** per calcolare l'ora esatta dell'alba e del tramonto del Sole; quest'ultimo è un punto



Fig. 7
Determinazione coordinate GPS tramite Google Maps.

Fig. 8
Schermata di Serial Monitor all'avvio dello sketch.

```

COM6

Astronomic Timer Switch
WiFimanager manage the connection
*wm:[1] AutoConnect
*wm:[2] Connecting as wifi client...
*wm:[2] setSTAConfig static ip not set, skipping
*wm:[1] Connecting to SAVED AP: EOLO_2.4G
*wm:[1] connectTimeout not set, ESP waitForConnectResult...
*wm:[2] Connection result: WL_CONNECTED
*wm:[1] AutoConnect: SUCCESS
*wm:[2] Connected in 3462 ms
*wm:[1] STA IP Address: 192.168.178.38
Data in memory:
Sunset:0
Sunrise:0
MonFri:ON
SatSun:ON
Mode:2
Sync by NTP server
Modem ON
Connecting to local WiFi.....
Connected
Transmit NTP Request
Receive NTP Response
Modem OFF

2022-9-20      17:52.26
Sunrise=353  Sunset=1093  Now=1072
Mode=2 Auto OUT=0

2022-9-20      17:53.00
Sunrise=353  Sunset=1093  Now=1073
Mode=2 Auto OUT=0

☒ Scorrimiento automatico ☐ Visualizza orario
Entrambi (NL & CR) 115200 baud Ripulisci l'output

```

Fig. 9
WebServer per l'impostazione dei parametri di funzionamento.

cruciale e lo tratteremo più in dettaglio. Affinché la predetta libreria possa lavorare correttamente vanno impostati i parametri di latitudine, longitudine e fuso orario. Avremmo potuto dotare il nostro sistema di un ricevitore GPS affinché determinasse la posizione geografica in modo automatico, ma abbiamo ritenuto poco efficiente questo modo di procedere visto che il dato sarebbe stato letto una sola volta. Con le due semplici righe di codice sotto riportate, possiamo impostare i valori corretti:

```

const int timeZone = 1;
Sunrise mySunrise(Latitude, Longitude, timeZone);

```

Il valore `timeZone` è il fuso orario ovvero +1 (in Italia) rispetto il meridiano di Greenwich, mentre latitudine e longitudine sono facilmente deducibili tramite il servizio Google Maps, dopo aver cercato la località nella quale sarà installato e utilizzato l'interruttore, come visibile nell'esempio proposto nella **Fig. 7**. Lo sketch con il quale viene gestito il

progetto, il cui listato completo è disponibile assieme ai file di questo numero della rivista, prevede all'avvio di connettersi alla rete Internet, scaricare i dati aggiornati di data e ora ed impostare di conseguenza l'orologio interno; terminata questa operazione il modulo WiFi viene spento al fine di migliorare l'efficienza energetica. La sequenza delle operazioni è visibile tramite il Serial Monitor: un esempio di monitoraggio è proposto dalla Fig 8. Nel loop principale, ad intervalli regolari di un minuto si va a verificare se è necessario attivare o meno l'uscita, le righe di codice sono riportate nel Listato 1, dove possiamo vedere che utilizzando il

mezzo ed il giorno la libreria calcola rispettivamente quanti minuti dopo la mezzanotte ci sarà l'alba (**sunriseMin**) e quanti minuti dopo mezzanotte si verificherà il tramonto (**sunsetMin**); la variabile di nome flag sarà settata durante il periodo di buio. Abbiamo previsto la possibilità di variare entro $\pm$ un'ora le due soglie di alba e tramonto tramite le variabili **alarmData.sunset** e **alarmData.sunrise** che, come vedremo più avanti, saranno gestite tramite un piccolo Web Server. Queste due variabili definiscono di quanti minuti vogliamo anticipare o posticipare l'orario di attivazione, per meglio adattare l'accensione delle luci alle reali necessi-

↓ Listato 1

```
void verifySwitch()
{
    int sunriseMin, sunsetMin;
    sunriseMin = mySunrise.Rise(month(), day());
    sunsetMin = mySunrise.Set(month(), day());

    sunsetMin += alarmData.sunset;
    sunriseMin += alarmData.sunrise;
    int nowMin = hour()*60 + minute();

    Serial.printf("Sunrise=%d Sunset=%d Now=%d\n", sunriseMin, sunsetMin, nowMin);

    bool flag, output=false;

    if (nowMin>sunriseMin && nowMin<sunsetMin)
        flag=false;
    else
        flag=true;

    if (alarmData.aMonFri && weekday()>1 && weekday()<7)
    {
        if(flag) output=true;
    }

    if (alarmData.aSatSun && (weekday()==7 || weekday()==1) )
    {
        if(flag) output=true;
    }

    switch (alarmData.aMode) {
        case 0:
            Serial.println("Mode=0 always OFF");
            if (OUTINVERT) digitalWrite(relePin, HIGH); else digitalWrite(relePin, LOW);
            break;
        case 1:
            Serial.println("Mode=1 always ON");
            if (OUTINVERT) digitalWrite(relePin, LOW); else digitalWrite(relePin, HIGH);
            break;
        case 2:
            Serial.printf("Mode=2 Auto OUT=%d", output);
            if (OUTINVERT) digitalWrite(relePin, !output); else digitalWrite(relePin, output);
            break;
        default:
            if (OUTINVERT) digitalWrite(relePin, HIGH); else digitalWrite(relePin, LOW);
            break;
    }
}
```

Fig. 10
Avvio del Web
Server evidenziato
dal Serial Monitor
di Arduino.



tà, ottimizzando al massimo i consumi. Abbiamo previsto anche due variabili di nome **alarmData.aMonFri** e **alarmData.aSatSun** che permettono di selezionare in quali periodi della settimana attivare le luci, ovvero solo da lunedì a venerdì, solo il fine settimana oppure tutta la settimana. Abbiamo anche previsto la variabile **alarmData.aMode** per definire la modalità di funzionamento. Se questa variabile vale zero il relé rimarrà sempre spento, è il caso in cui vi sia la necessità di mantenere l'impianto luci spento. Se la variabile **alarmData.aMode** vale uno, il relé sarà sempre alimentato e le luci risulteranno sempre accese, mentre il valore due indica il funzionamento automatico in cui accensioni e spegnimenti degli utilizzatori avvengono secondo l'orario impostato nel sistema. Qualora si desideri modificare questi valori è sufficiente premere il pulsante del circuito per più di tre secondi: così facendo verrà attivato il modulo

WiFi e si avvierà il Web Server, raggiungibile da qualsiasi browser connesso alla stessa rete, come visibile in **Fig. 9**; dall'interfaccia utente disponibile sarà possibile effettuare le impostazioni. In **Fig. 10** vedete l'apertura in Serial Monitor.

CONCLUSIONI

Siamo giunti alla fine di questo articolo in cui vi abbiamo spiegato il funzionamento degli interruttori orari astronomici che, anche se poco noti, sono disponibili in quasi tutti i cataloghi delle principali aziende fornitrici di materiale elettrico. A titolo di esempio riportiamo in **Fig. 11** l'immagine del modello GWD6785 prodotto dalla Gewiss, ma analogamente sono presenti prodotti simili nei cataloghi Bticino, Finder ecc. Alcuni di questi dispositivi dispongono, oltre ad un modulo per l'accesso alla rete Internet, anche di un transceiver Bluetooth per essere programmati tramite app per smartphone. La recente necessità di efficientamento energetico può essere soddisfatta da diverse soluzioni, di cui una è quella presentata in questo articolo. □

Fig. 11
Interruttore orario
astronomico
Gewiss GWD6785.



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. La board con ESP8266 (cod. NODEMCUESP) è in vendita a Euro 9,50 mentre il modulo relè (cod. RELAY1CH) è disponibile a Euro 5,00. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

Contatore Geiger



€169,00
Cod. FR822

Misura l'intensità di raggi X, raggi gamma (γ) e raggi beta (β). Utilizza un microprocessore a 32 bit e dispone di ampio display LCD con retroilluminazione, tastiera a membrana, allarme acustico con soglia regolabile.



Air Quality

CONTROL

Purificatore d'aria da tavolo
a raggi ultravioletti

MicroUVAIR è un sistema di sanificazione aria con ventilazione forzata a ciclo chiuso basato su led UVC. L'aria in ingresso viene purificata attraverso un filtro ad alta densità (30 PPI), quindi viene sterilizzata tramite luce ultravioletta emessa da LED UVC in grado di effettuare la massima azione germicida. Una volta purificata l'aria viene reimpressa nell'ambiente.

€199,00
Cod. MICROUVAIR-T



Prezzi IVA inclusa.

Misuratore di formaldeide e
monitor della qualità dell'aria



Fornisce in tempo reale la quantità di formaldeide (HCHO) presente nell'ambiente circostante e monitora il livello di TVOC (Composti Organici Volatili Totali). È dotato di display per la lettura dei valori rilevati.

€21,90
Cod. FR832

Il buongiorno si sente dal mattino

NEW

Wooden Clock - Sveglia digitale con temperatura e calendario

Orologio digitale con display a LED rossi in un elegante contenitore effetto legno. Dispone di allarme sonoro, batteria di backup (due LR44 incluse), visualizzazione temperatura ambiente e calendario. Alimentazione tramite 3 batterie minitipo AAA o tramite alimentatore USB (acquistabile separatamente) e cavetto da USB a plug DC (compreso). Dimensioni 180x80x80 mm, peso 270 grammi.



€15,00
cod. WC233

Prezzi IVA inclusa.

Sveglia digitale a campanelli meccanici con doppio allarme

- » Involucro verde fluo e bianco
- » Possibilità di impostare due differenti orari di allarme
- » Retroilluminazione display verde
- » Dimensioni 110x105x74 mm
- » Alimentazione tramite 3 batterie minitipo AAA (acquistabili separatamente)
- » Peso 200 grammi



€11,90
cod. WC206

Sveglia digitale a specchio con display a led bianchi

- » Display a specchio a led bianchi con visualizzazione 12/24 h
- » Allarme con snooze
- » Regolazione di luminosità (normale/notte)
- » Visualizzazione temperatura (°C - °F)
- » Alimentazione tramite 3 batterie minitipo AAA o alimentatore (acquistabili separatamente)
- » Dimensioni 82x82x35 mm
- » Peso 88 grammi

Attenzione! Necessità di presa USB.



€12,90
cod. WC214

Radiosveglia con proiezione dell'ora e radio FM

- » 2 sveglie
- » Allarme con radio o segnale acustico
- » Radio FM 87,5 - 108 MHz con 10 stazioni memorizzabili
- » Proiezione dell'ora con regolazione manuale dell'angolazione e della nitidezza
- » Distanza di proiezione da 0,9 a 2,7 metri
- » Alimentazione 220 V AC ~ 50 Hz, 5 W
- » Dimensioni 160x65x57 mm
- » Peso 367 grammi



€32,90
cod. CRP717BLACK

Sveglia radiocontrollata

- » Display LCD con retroilluminazione BLU
- » Data e orario radiocontrollati tramite segnale radio DCF77
- » Mostra le fasi lunari
- » Funzione sveglia con buzzer
- » Funzione termometro per temperatura interna
- » Alimentazione tramite due batterie da 1,5 volt tipo AAA (acquistabili separatamente)
- » Dimensioni 82x2,50x82 mm
- » Peso 90 grammi



€11,90
cod. REC34BLACK

LIGHT CONTROL

del Prof. **MICHELE MENNITI**



La luce della stanza
te la spengo io!
Un semplice ma
efficace sistema
di controllo per
spegnere la luce.

S

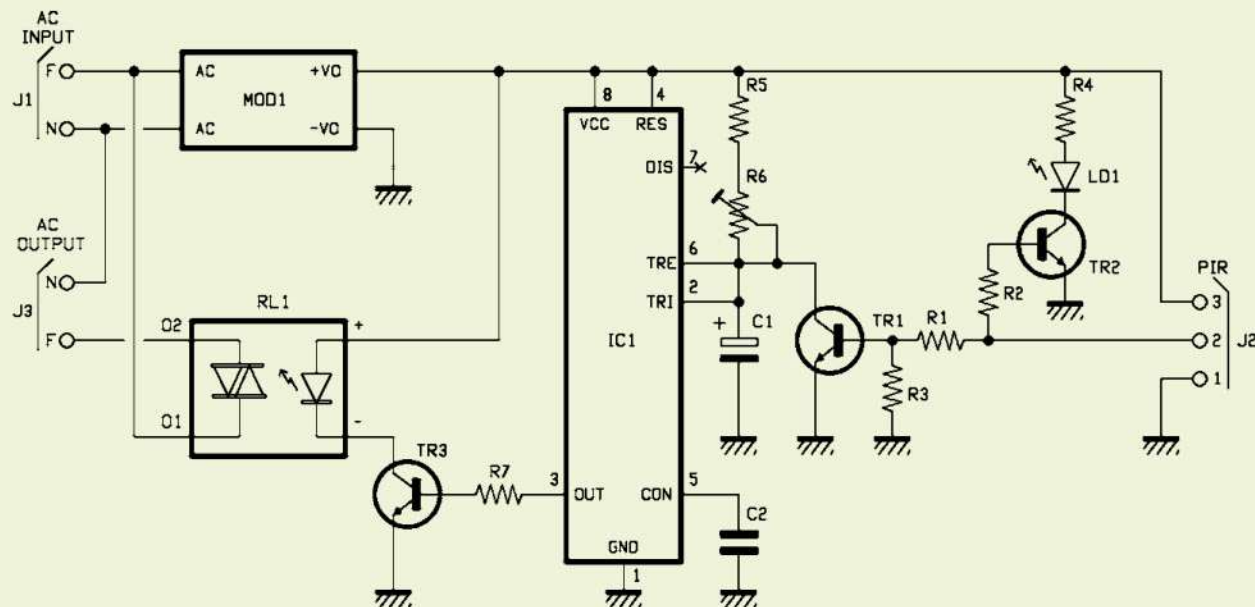
tiamo affrontando la più grande crisi energetica di questo secolo e stiamo imparando a comprendere come un occhio ai consumi energetici possa farsi sentire sulla bolletta.

Tra i tanti errori che molti commettono giornalmente sicuramente c'è quello di lasciare le luci accese quando

si spostano da una stanza all'altra, con un inutile spreco di energia. Questo semplice progetto, facilmente realizzabile, ci permetterà di controllare la presenza di persone nella stanza e spegnere la luce quando sarà vuota.

DESCRIZIONE GENERALE

L'idea per questo progetto nasce dal lavoro che abbiamo presentato sui numeri precedenti della rivista, riguardante l'Analizzatore per circuiti basati sul Timer 555; abbiamo pen-



sato di sfruttare la configurazione monostabile per realizzare un temporizzatore regolabile controllato da un sensore PIR a infrarossi, in grado di rilevare la presenza del corpo umano.

Poiché il sensore dovrà essere fissato sul soffitto o comunque nei pressi della lampadina da controllare, sarà necessario poterlo alimentare direttamente con la tensione alternata di 230 V presente in

quel punto. In tal modo il circuito può essere interposto tra i due fili della rete elettrica e la lampadina, interrompendo l'alimentazione quando necessario. In **Fig. 1** abbiamo riportato il nostro prototipo montato.

SCHEMA ELETTRICO

Per comprendere bene il funzionamento del circuito consigliamo di dare un'occhiata all'articolo presentato sul numero 255 della rivista (Introduzione al Timer 555). Il cuore del nostro Light Control è infatti

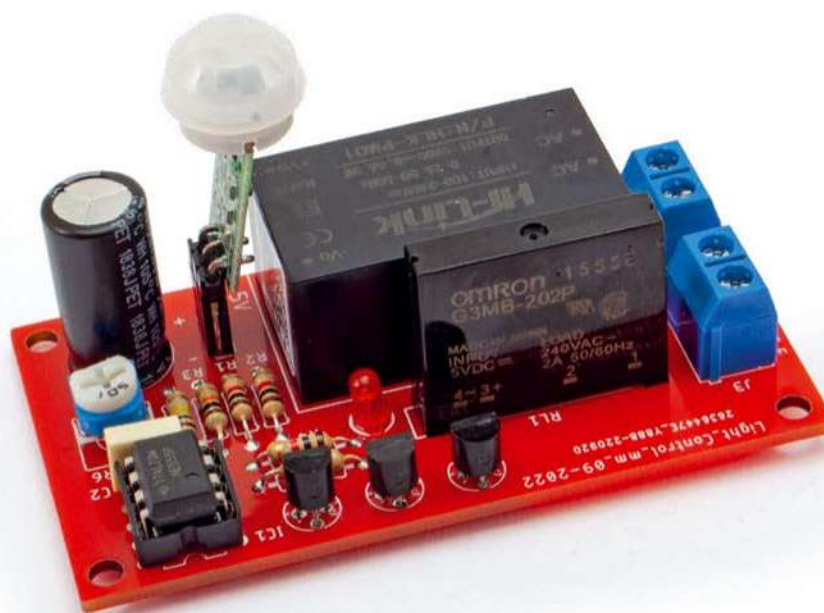


Fig. 1
Il Light Control.

il circuito integrato NE555 (IC1) in configurazione monostabile, tipicamente usata per funzioni di timer. La temporizzazione è ottenuta mediante il gruppo R5, R6 e C1, collegati agli ingressi TRIGGER (pin2) e SOGLIA (pin 6) dell'integrato.

Il resistore R5 ed il trimmer R6, configurato come resistenza variabile, in serie tra loro, permettono di ottenere una resistenza totale regolabile tra 100 kΩ (trimmer al minimo) e 1,1 MΩ (trimmer al massimo). Il gruppo R5-R6 serve per dosare la corrente di carica del condensatore C1 e quindi la durata della temporizzazione, in base alla formula:

$$t(s) = 1,1 * (R5 + R6(k\Omega)) * C1(\mu F) / 1000$$

il range temporale può variare quindi tra:

$$t1 = 1,1 * 100 * 1000 / 1000 = 110s$$

quindi poco meno di 2 minuti e:

$$t2 = 1,1 * 1100 * 1000 / 1000 = 1210s$$

circa 20 minuti. Volendo aumentare la durata della temporizzazione sarà sufficiente aumentare la capacità di C1; per esempio con 2200 μF, invece dei 1000 μF previsti da noi, il tempo massimo arriva a circa 44 minuti.

Se poi si sostituisce il trimmer con uno da 2,2 MΩ si può arrivare a circa un'ora e mezza.

La seconda parte del circuito è costituita dalla sezione di controllo da parte del PIR; il sensore a riposo ha l'uscita (pin 2) a livello LOW, quindi prossimo a 0 V, nel momento in cui rileva una presenza la tensione aumenta e, tramite il resistore R1, manda in saturazione il transistor NPN TR1, il cui collettore si viene a trovare collegato alla massa mentre C1 viene scaricato istantaneamente.

In assenza di ulteriore rilevazione, dopo circa 6 secondi, il sensore torna in condizione di riposo, TR1 si interdice e C1 ricomincia a caricarsi.

Lo stesso sensore pilota anche TR2, tramite R2, e fa accendere il LED LD1, la cui corrente è limitata da R4, dando conferma dell'attivazione del PIR.

Il circuito di pilotaggio della lampadina, costituito da R7, TR3 e RL1, viene attivato dalla condizione HIGH dell'USCITA (pin 3) del 555, che manda in conduzione TR3, portando a massa il pin 1 del relè a stato solido RL1 (catodo del fotodiodo interno); poiché il pin 2 di RL1 (anodo del fotodiodo interno) è collegato stabilmente a +5 V, il fototriac interno (pin 3 e 4) va in conduzione e provoca l'accensione della lampadina.

Il modulo MOD1 è un microalimentatore step-down, che riceve in ingresso la tensione elettrica di 230 Vac e fornisce in uscita 5 Vdc 600 mA, più che sufficienti per alimentare l'intero circuito, che assorbe circa 15 mA.

DESCRIZIONE DEL FUNZIONAMENTO

Prima di tutto dobbiamo aver chiaro che i due fili provenienti dall'interruttore delle luci della stanza vanno scollegati dalla lampadina (o dall'insieme di lampadine) e collegati al connettore a vite J1 (AC INPUT), mentre la lampadina va collegata al connettore a vite J2 (AC OUTPUT).

Anche se abbiamo indicato espressamente i collegamenti auspicabili di fase e neutro della rete elettrica, il comportamento del circuito non cambia in caso di inversione.

Come si vede dallo schema, uno dei due poli è collegato direttamente al connettore di uscita, mentre l'altro è interrotto dal contatto del relè RL1.

Nel momento in cui il circuito viene alimentato, mediante la pressione dell'interruttore della luce della stanza, il modulo alimentatore eroga +5 V (Vcc), l'uscita del sensore va a livello HIGH, il condensatore C1 viene scaricato e di conseguenza entrambi i pin TRIGGER e SOGLIA si trovano a circa 0 V, cioè nella condizione < 1/3 Vcc.

Come si evince dalla **Tabella 1** (prima riga), in questo stato l'USCITA si trova a livello HIGH, quindi TR3 attiva il relè RL1 e la fase viene collegata al connettore J2, provocando l'accensione della lampadina.

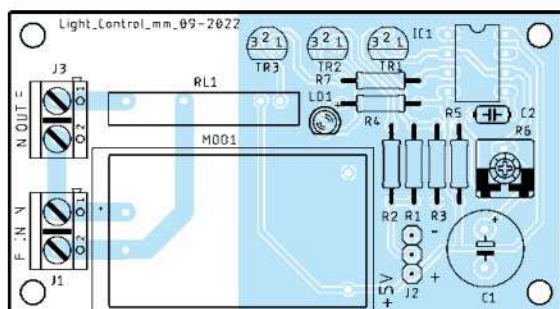
Contemporaneamente il condensatore C1 inizia a caricarsi, e la durata della carica dipende naturalmente dalla regolazione del trimmer R6, potendo variare, come già detto, tra 2 e 20 minuti.

Durante la fase della carica di C1, la tensione ai suoi capi ad un certo punto sarà > 1/3 Vcc, ma finché essa sarà anche inferiore di 2/3 Vcc (ricordiamo che i pin TRIGGER e SOGLIA sono collegati tra loro), la Tabella 1 ci dice che non interverranno variazioni (terza riga).

Se in questo periodo di tempo il sensore non viene attivato da una presenza nella stanza, nel momento in cui la tensione di carica di C1 raggiunge i

TRIGGER 2	SOGLIA 6	USCITA 3
< 1/3 VCC	Irrilevante	H
> 1/3 VCC	> 2/3 VCC	L
> 1/3 VCC	< 2/3 VCC	non cambia

Tabella 1
Tabella
funzionale
del 555.



Elenco Componenti:

R1, R2, R4: 1 kohm
 R3, R7: 10 kohm
 R5: 100 kohm
 R6: Trimmer 1 Mohm 5x5 mm MO
 C1: 1000 µF 16 VL elettrolitico
 C2: 10 nF 63 VL poliestere
 TR1, TR2, TR3: BC547
 IC1: NE555
 LD1: LED 3 mm rosso
 RL1: Relè allo stato solido
 G3MB-202P

MOD1: Modulo alimentatore
 AC-DC HLK-PM01

Varie:

- Morsetto 2 poli passo 5 mm (2 pz.)
- Pin strip femmina 3 vie passo 2,54 mm
- Zoccolo 4+4
- Circuito stampato S1657

in carica. Per riaccendere la luce occorre portare l'interruttore in posizione "spento" e poi di nuovo in posizione "acceso", iniziando nuovamente il ciclo. Questo passaggio naturalmente non è necessario se si entra nella stanza permettendo la rilevazione da parte del sensore PIR che, attivandosi, manderà in conduzione TR1 e scaricherà C1, ricreando di nuovo la condizione della prima riga della **Tabella 1** e facendo accendere la luce.

Se durante la fase di carica del condensatore, con la luce accesa, il sensore PIR rileva un movimento, si attiva e manda in conduzione TR1, facendo scaricare C1; questo significa che, in presenza di persone nella stanza la luce non si spegnerà mai, com'è giusto che sia, a meno di non agire sull'interruttore.

Ogni attivazione del sensore PIR sarà chiaramente indicata dall'accensione temporanea del LED LD1.

IL MONTAGGIO DEI COMPONENTI

Per il montaggio dei componenti del nostro Light Control abbiamo predisposto un circuito stampato a singola faccia, per la cui realizzazione troverete i file gerber nella sezione download del sito della rivista www.elettronica.in.

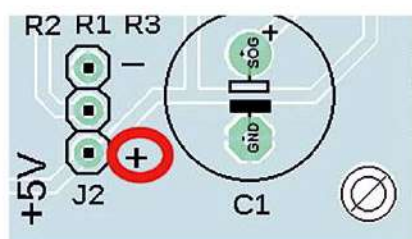
In alternativa alla preparazione artigianale, ricordiamo che è disponibile il servizio di realizzazione PCB (codice 7200-PCBPRODUCTION) sul sito www.futuranet.it.

Seguendo il piano di montaggio, come sempre conviene iniziare dai componenti più piccoli e più bassi: resistori e zoccolo per l'integrato, trimmer, transistor, condensatore non polarizzato, morsetti, connettore header, relè (reperibile su www.futuranet.it col codice YL051), modulo alimentatore (codice 3WMA2205V), condensatore elettrolitico, diodo LED. Il modulo PIR (codice MINIPIRMOD), ha un coperchio che integra una lente di Fresnel che deve fuoriuscire completamente dal contenitore per funzionare correttamente.

Nel caso di quello da noi proposto (codice COFFER1) il modulo dovrà essere saldato direttamente sul PCB; in caso di contenitore con mag-

2/3 Vcc, ovviamente anche i pin TRIGGER e SOGLIA si troveranno allo stesso stato, quindi l'USCITA sarà portata a livello LOW (seconda riga), il relè verrà disattivato, scollegando la corrente elettrica e spegnendo la lampadina. Questa condizione resta immutata nel tempo in quanto C1 verrà costantemente mantenuto

Fig. 2
 Particolare del collegamento del modulo PIR al PCB





giore altezza, occorrerà adottare un connettore header, in modo che dal coperchio del contenitore fuoriesca solo il cappuccio plastico.

NOTA BENE: il modulo PIR termina con 3 pin strip maschi, bisogna fare molta attenzione alla polarità, in caso contrario si danneggerà alla prima accensione!

Sul suo circuito stampato è serigrafato il simbolo "+" che deve combaciare con quello posto vicino ad uno dei tre contatti di J2. In **Fig. 2** è possibile vedere i particolari dei due PCB.



Fig. 3
Prototipo
Light Control
montato nel
contenitore
plastico.

schema di CABLAGGIO



Anche il LED dovrà spuntare dal coperchio quindi, se i suoi reofori non sono abbastanza lunghi, sarà necessario usare due spezzoni di filo di rame, bloccando poi il LED mediante un portaled da pannello. L'ultima operazione da fare sarà quella di inserire l'NE555 nello zoccolo, facendo attenzione al verso (fare riferimento al piano di montaggio): la tacchetta presente sul corpo dell'integrato deve corrispondere con la tacchetta presente sullo zoccolo e con la serigrafia del circuito stampato.

Nella **Fig. 3** è visibile il nostro prototipo montato nel contenitore COFFER1.

Oltre ai due fori per la testa del sensore PIR e per il LED, lateralmente andrà aperta una fessura per il passaggio dei 4 fili, due provenienti dall'impianto e due destinati alla lampadina. Si possono praticare anche 4 fori con diametro idoneo a far passare i fili. Il Light Control, per svolgere correttamente la sua funzione, dovrà essere fissato al soffitto (o all'eventuale lampadario), in una posizione tale da permettere al sensore di captare i movimenti delle persone che sono nella stanza, in caso contrario una qualsiasi zona d'ombra potrebbe causare lo spegnimento indesiderato della luce.

I COLLEGAMENTI

ATTENZIONE: Il collegamento del nostro Light Control richiede la necessaria esperienza in materia di operazioni con la corrente elettrica quindi è indispensabile chiamare un professionista del settore. Nello schema di cablaggio riportiamo lo schema grafico dei collegamenti da rispettare rigorosamente.

Da considerare che i due contatti interni ("N") dei due connettori J1 e J3 sono collegati direttamente tra loro tramite una pista del PCB, quindi, volendo, sarebbe possibile interrompere solo uno dei fili destinati alla lampadina e collegare il capo proveniente dall'interruttore al contatto esterno di J1 ("F") e quello destinato alla lampadina sul contatto esterno di J3 ("F"). In pratica i due fili blu nello schema di cablaggio possono restare collegati tra loro esternamente al nostro circuito, lasciando i corrispondenti contatti dei connettori liberi. Per comodità e per semplicità di cablaggio abbiamo comunque previsto due morsetti separati. Rispettate le indicazioni e non sbaglierete.

Il relè è dichiarato compatibile con una corrente di 2 A sui contatti, ciò significa che può reggere senza problemi un carico di circa 400W, pertanto di molto superiore al consumo delle tipiche lampadine a LED da stanza. Chiaramente potrete collegare anche lampade a LED.

IMPOSTAZIONE DELLA TEMPORIZZAZIONE

Una volta effettuati i collegamenti, prima di procedere con il collaudo, conviene regolare il trimmer ruotando il cursore in base alla temporizzazione che si vuole ottenere.

Considerando che il comportamento è di tipo lineare e che ruotando il cursore tutto a sinistra (in senso antiorario) si ha un tempo di 2 minuti, mentre tutto a destra (in senso orario) il tempo sarà di 20 minuti, per ottenere un valore all'interno del range occorrerà posizionarlo in un punto idoneo: per esempio a metà corsa saranno circa 10 minuti.

COLLAUDO FINALE

A questo punto sarà possibile procedere con il collaudo, operazione estremamente semplice in quanto sarà sufficiente premere l'interruttore della luce per vedere la lampadina accendersi normalmente.

Contemporaneamente si farà partire un cronometro, restando fuori dalla stanza, per non entrare nel raggio di azione del PIR, e si misurerà il tempo necessario per lo spegnimento della lampadina, quindi si potrà posizionare l'interruttore su spento ed effettuare la regolazione fine del trimmer, ripetendo l'intera operazione fino ad ottenere la temporizzazione desiderata.

Una volta regolata la temporizzazione si potrà chiudere il coperchio del contenitore, avendo la certezza che, da questo momento in poi, non ci saranno più sprechi di energia elettrica a causa di una lampadina inutilmente accesa in una stanza dove non c'è nessuno! 



Cosa occorre?

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

Il cuore del circuito, il timer 555, (cod. 555) è in vendita a Euro 0,60, il sensore di presenza PIR (cod. MINIPIRMOD) a Euro 5,50, il relè allo stato solido (cod. YL051) a Euro 3,50, il modulo alimentatore (cod. 3WMA2205V) a Euro 6,50, il contenitore utilizzato (cod. COFFER1) è disponibile a Euro 6,50. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

Fai il pieno di energia!

€ 12,⁹⁰
Cod. UL45-6



6V-4,5AH

Batteria ricaricabile ULTRACELL al piombo senza manutenzione

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

12V-1,3AH



€ 13,⁹⁰
Cod. UL13-12



12V-7AH

€ 26,⁹⁰
Cod. UL7-12

6V-1,3AH



€ 7,⁹⁰
Cod. UL13-6



6V-3,4AH

€ 11,⁹⁰
Cod. UL34-6



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



€ 12,⁰⁰

Cod. HR03AAA800



€ 13,⁰⁰

Cod. HR6AA2100



€ 13,⁸⁰

Cod. HR14C3000

Prezzi IVA inclusa.

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



€ 35,90

Cod. **ARDUKITLEVEL1**



€ 45,90

Cod. **ARDUKITLEVEL2**



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

Disponibili anche....

SET PER ARDUINO UNO REV3

€ 65,00

Cod. **ARDUKITV6**



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

SET CON ARDUINO UNO REV 3 CON CAVO USB

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



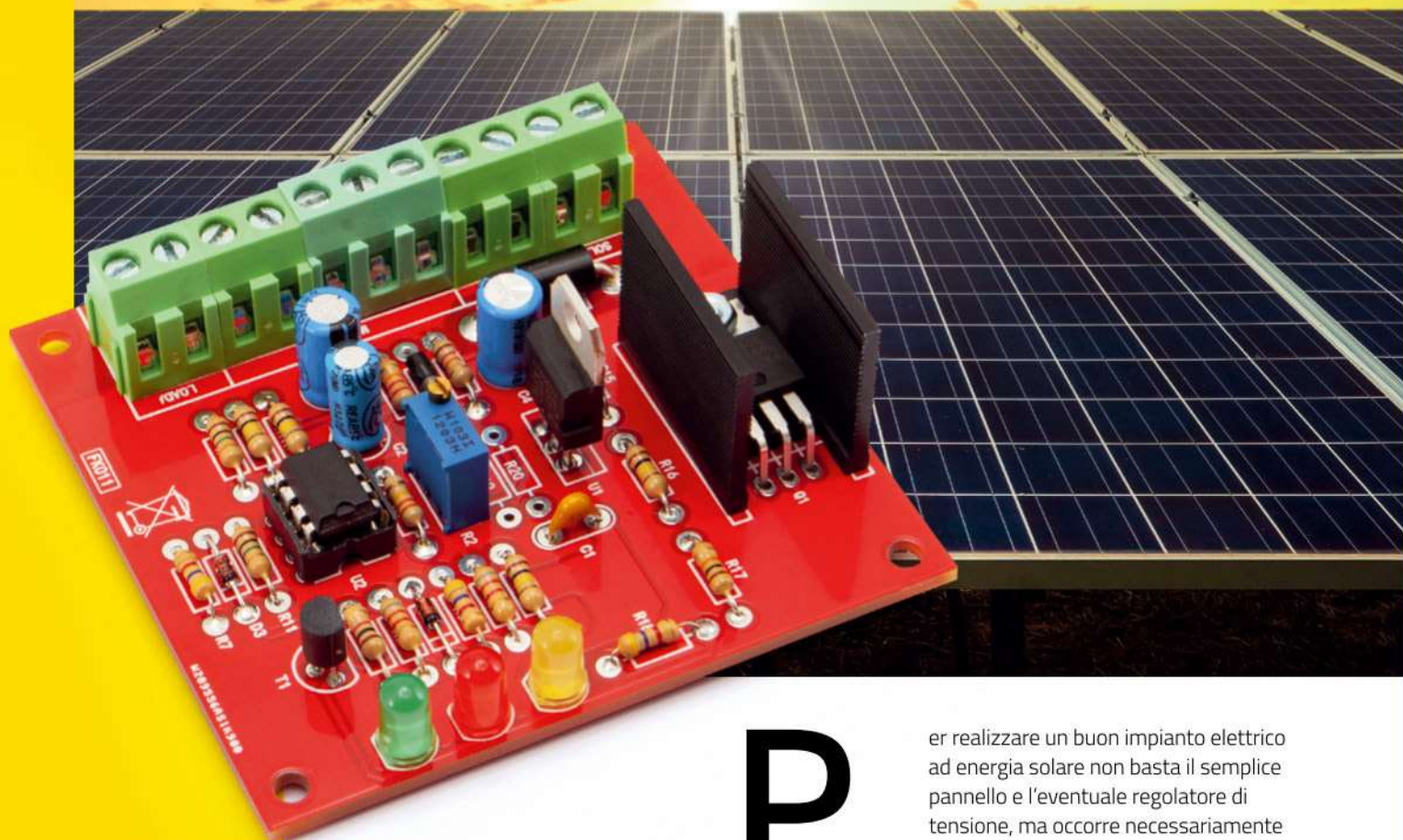
€ 29,00

Cod. **COMB06G**

Prezzi IVA inclusa

REGOLATORE PER PANNELLI FV

di **BORIS LANDONI**

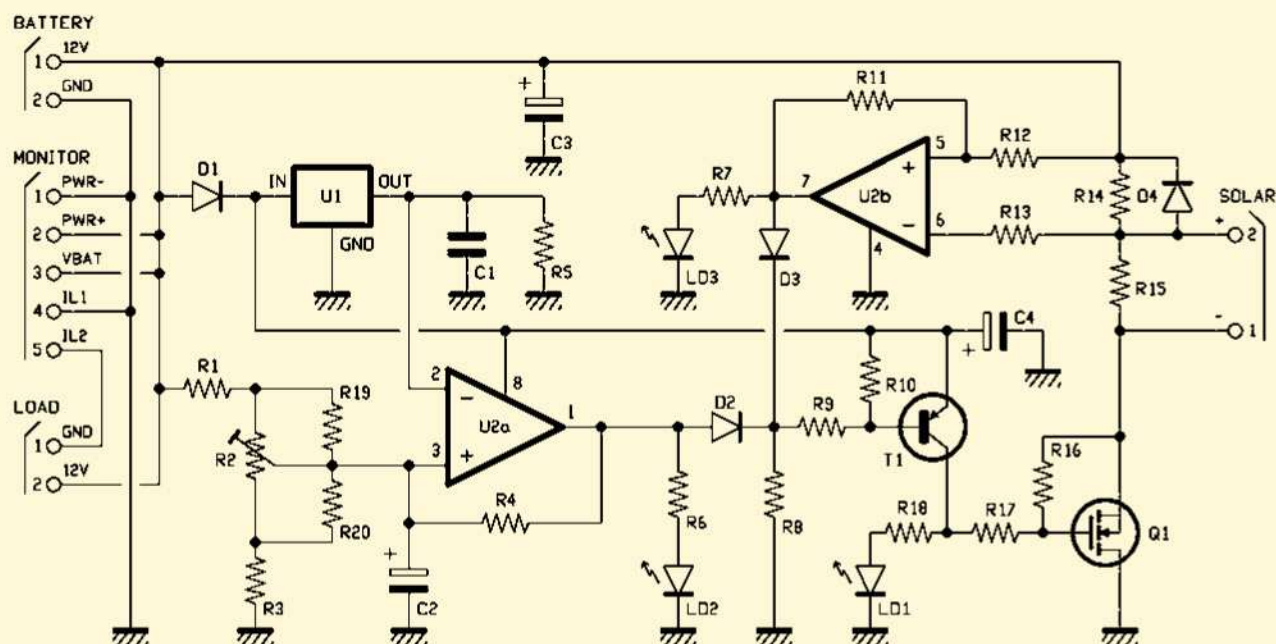


Stabilizza e ottimizza la carica di accumulatori alimentati da pannelli fotovoltaici a 12/24V, gestendo il carico.

P

er realizzare un buon impianto elettrico ad energia solare non basta il semplice pannello e l'eventuale regolatore di tensione, ma occorre necessariamente disporre di una o più batterie che possano accumulare energia nelle ore diurne per poi restituirla quando c'è poca insolazione oppure fa buio; diversamen-

te gli utilizzatori possono funzionare soltanto in presenza della luce del sole e a regime, soltanto con un buon livello di irraggiamento solare. La gran parte dei dispositivi alimentati con pannelli fotovoltaici deve poter funzionare indipendentemente dalla presenza o dall'assenza della luce del sole, in maniera stabile a prescindere dal livello di irraggiamento, quindi la batteria (che ormai usualmente chiamiamo "accumulo") è sicuramente necessaria per dare corrente quando



non la dà più il pannello. Ma non solo; spesso le batterie servono anche per un motivo un po' meno intuibile ma tangibile: potendo immagazzinare una certa quantità di elettricità, più o meno grande a seconda delle caratteristiche, possono erogare correnti ben più elevate di quanta se ne possa chiedere ai pannelli solari, pur essendo caricate sempre da questi ultimi.

Volendo fare un esempio, un pannello solare in grado di erogare 1 A di corrente non può, da solo, alimentare un dispositivo che assorbe 2 ampere; però, se con il pannello si va a caricare una batteria che può fornire in scarica 2, 3 ampere o più, il problema è risolto: il pannello carica la batteria che,

quando serve, alimenta l'utilizzatore. Chiaramente in questo caso è impensabile che il carico funzioni a regime continuo, dato che la corrente fornita dal pannello solare è minore di quella che esso richiede; per la precisione, l'utilizzatore può essere alimentato per un periodo di tempo rapportato alla quantità di corrente data nello stesso periodo dal pannello solare.

Riprendendo l'esempio fatto in precedenza, se il pannello può erogare 1 A ed il carico richiede 2A, ammettendo che la batteria possa essere caricata per 10 ore al giorno, il carico potrà funzionare per poco meno di 5 ore, salvo eccezioni.

Ma a parte i numeri, per gestire un impianto fotovoltaico per fare in modo che questo possa caricare la batteria con una corrente pressoché costante, interrompendo la corrente stessa quando la batteria è a piena carica, bisogna disporre di un particolare regolatore che solitamente è allo stato solido: un circuito elettronico del tipo di quello proposto in queste pagine.

Il nostro, oltre a regolare la carica della batteria sospendendola quando si arriva al valore limite, permette di gestire il carico, collegandolo quando la tensione della batteria è sufficiente a tenerlo in funzione, e scollegandolo quando invece la tensione di batteria è troppo bassa. Insomma, il nostro regolatore di carica è un completo gestore del funzionamento di un impianto a pannelli solari fotovoltaici, e può lavorare controllando carichi la cui corrente complessiva non superi i 5A; è comun-

CARATTERISTICHE TECNICHE

-  **Tensione di ingresso (pannello solare):**
12 ÷ 28 V
-  **Massima corrente di ingresso (pannello solare):** 5 A
-  **Corrente massima in uscita (carico):** 25 A
-  **Assorbimento massimo del circuito di controllo:**
20 mA
-  **LED di segnalazione dello stato di ricarica, insolazione insufficiente e batteria carica.**
-  **Possibilità di inserzione misuratore di energia consumata**

que possibile, sostituendo semplicemente il diodo D4, aumentare la corrente gestibile. Esprimendoci con i termini tecnici dell'attualità, quello che vi proponiamo è ciò che, negli impianti fotovoltaici, è chiamato "sistema di accumulo". Ma vediamo bene di cosa si tratta analizzando il circuito, il cui schema elettrico è visibile in queste pagine: il regolatore è in sostanza un regolatore di corrente realizzato con l'ausilio di un circuito a comparatori e di un MOSFET che assorbe la parte di corrente in eccesso; il tutto è completato da un comparatore che permette di connettere e disconnettere il pannello dalla batteria quando questa è a piena carica.

SCHEMA ELETTRICO

Per comprendere il funzionamento del circuito è utile guardare lo schema elettrico, dalla quale si vede lo schema del regolatore collocato nell'impianto tipo: il polo positivo del pannello solare è collegato al polo positivo della batteria mediante il diodo D4, mentre il polo negativo è collegato a massa tramite il MOSFET Q1 che funge da interruttore statico pilotato. Quando il MOSFET è in conduzione il circuito si chiude e la batteria

viene ricaricata. In virtù della sua ridotta resistenza ($R_{ds(on)}$) in stato di conduzione, dell'ordine di un decimo di ohm, il MOSFET si comporta come un vero e proprio interruttore statico.

Notate che il circuito ha due linee di alimentazione negativa: GND è quella collegata al negativo della batteria e alla massa contrassegnata dal tradizionale simbolo di massa dei circuiti elettronici, che fa da riferimento al circuito di controllo ed entra in contatto con GND quando il MOSFET va in stato di ON. C'è poi il negativo del pannello fotovoltaico, che fa da riferimento negativo al source del MOSFET con il quale chiudiamo il circuito del pannello stesso sull'accumulatore e sul resto del ramo di massa del circuito.

Per fare condurre il MOSFET è necessario applicare al suo gate una tensione positiva ovvero fare condurre il transistor T1 che a sua volta pilota Q1. Quando T1 ed il MOSFET sono in conduzione, il LED giallo LD1 risulta acceso. Questo componente indica dunque che il pannello sta ricaricando la batteria. Il transistor T1 (un PNP tipo BC557) è normalmente in conduzione perché la sua base è collegata a massa tramite la resistenza R8. Per bloccare la conduzione è necessario che una

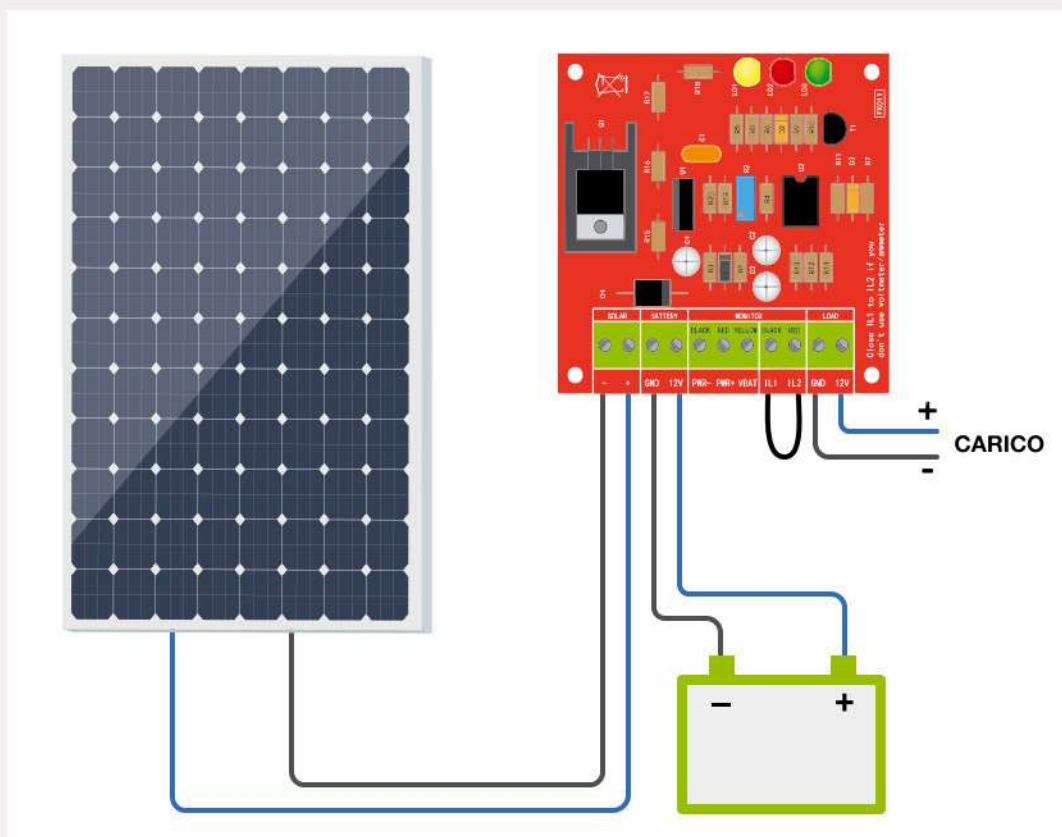
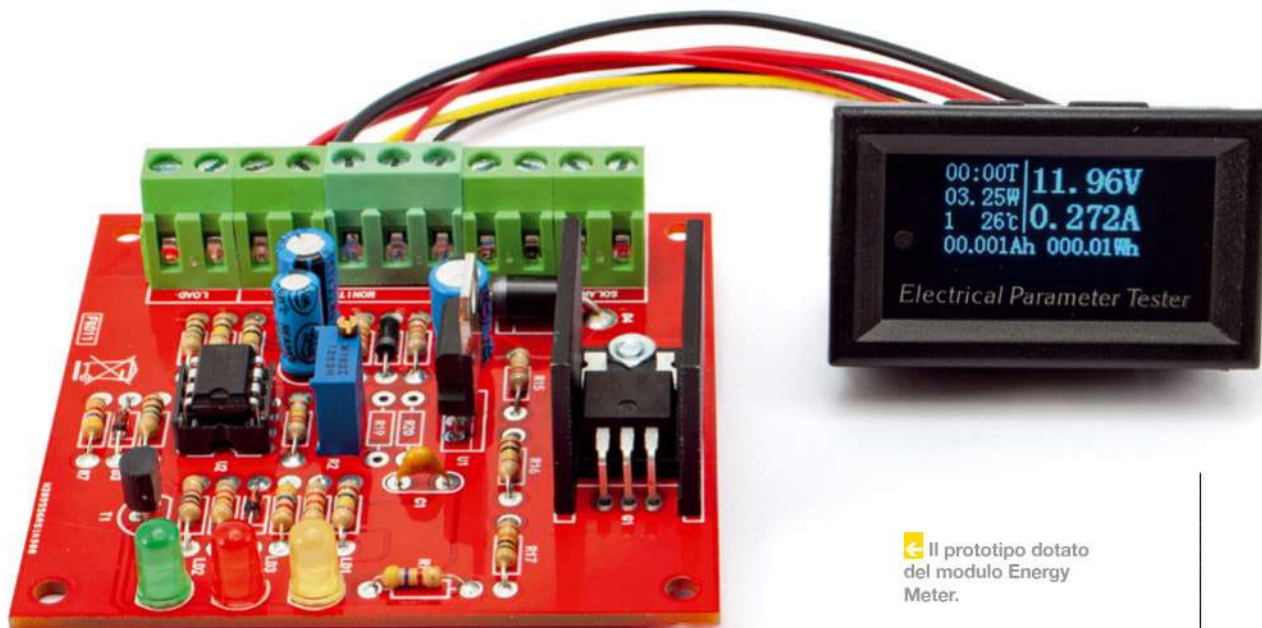


Fig. 1
Piano di cablaggio generale dell'impianto fotovoltaico. Il regolatore blocca la carica della batteria quando è completamente carica.



Il prototipo dotato del modulo Energy Meter.

qualsiasi delle uscite dei due operazionali presenti un livello alto, livello che viene applicato alla base di T1 tramite i diodi D2 e D3. Vediamo quando ciò avviene, studiando innanzitutto il circuito che fa capo al primo operazionale (U2a). All'ingresso invertente (piedino 2) di questo operazionale viene applicata la tensione a 5 volt fornita dal regolatore U1 mentre all'ingresso non invertente (pin 3) viene applicata, mediante il trimmer R2, parte della tensione presente ai capi della batteria. Quando questa tensione supera quella applicata sul pin 2, l'uscita dell'operazionale (normalmente a livello basso) si porta ad un livello alto bloccando il MOSFET ed attivando il LED verde LD2. È evidente che il trimmer va regolato in modo da ottenere la commutazione quando la tensione della batteria raggiunge un livello di 14,2÷14,4 volt (batteria completamente carica). La resistenza R4 introduce una leggera isteresi in modo da modificare la soglia di commutazione ed evitare che

l'operazionale entri in oscillazione. In pratica, per ottenere la commutazione dell'operazionale da 0 a 1, è necessario che la batteria raggiunga un potenziale di 14,4 volt mentre, con l'uscita a livello alto, per ottenere la commutazione da 1 a 0, la tensione della batteria deve scendere a 13,5 volt. Dunque questo stadio blocca la corrente quando la batteria è completamente carica.

Vediamo quindi cosa succede nella sezione relativa all'altro operazionale, il quale presenta un livello alto in uscita quando la tensione del pannello solare è inferiore a quella della batteria; in questo caso si accende il LED rosso LD3 ed il MOSFET viene posto in interdizione. Al contrario, quando il pannello è illuminato e la sua tensione è sufficiente per caricare la batteria, il LED è spento ed il MOSFET è in stato di ON.

Ricapitolando, dunque, il LED giallo (LD1) segnala che il pannello sta ricaricando la batteria, quello rosso (LD3) che l'insolazione è insufficiente per

Tabella 1
Connessioni del
modulo Energy
Meter utilizzato.

FILO	CONTATTO CONNETTORI SU PCB	FUNZIONE
Rosso sottile	+Vcc	Positivo di alimentazione dello strumento
Nero sottile	GND	Negativo di alimentazione dello strumento
Giallo	VIN	Ingresso per la misura della tensione
Rosso spesso	I+	Ingresso della corrente da misurare
Nero spesso	I-	Uscita della corrente da misurare

IL DIMENSIONAMENTO DEL SISTEMA

Quando si va devono abbinare pannelli solari e batterie al circuito è necessario fare prima alcune considerazioni. Innanzitutto è necessario conoscere il consumo complessivo del carico. Per fare un esempio ipotizziamo di illuminare un locale con quattro lampade a LED da 12 volt - 10 watt con un funzionamento continuo di 4 ore, il che significa una potenza oraria di 40W; per quattro ore, questo significa un consumo di energia pari a 0,16 kWh. A questo punto possiamo scegliere la batteria a 12V, che dovrà avere una capacità di almeno $160W/12V = 13,3Ah$. Nella pratica, per avere dell'energia di riserva, è consigliabile utilizzare una batteria con una capacità almeno doppia. Ora dobbiamo scegliere il pannello necessario per ricaricare la batteria: qui entra in gioco un parametro che si chiama ESH (Equivalent Sun Hours) ovvero le "ore di sole equivalente". Questo parametro indica qual'è mediamente il numero di ore di sole equivalenti al massimo irraggiamento per ciascuna zona del nostro paese. Solitamente questo dato si riferisce al periodo invernale: quello estivo si ottiene raddoppiando il valore. Nel caso dell'Italia il valore dell'ESH è compreso tra 1,5 e 3; ciò significa che nell'Italia settentrionale i nostri pannelli funzioneranno mediamente a pieno regime per 1,5 ore d'inverno e 3 ore d'estate mentre nel meridione avremo 3 ore di funzionamento alla massima potenza d'inverno e 6 ore d'estate. Considerando un valore medio di 3 ore per sette giorni e tenendo conto che abbiamo bisogno di circa 180 watt per ogni fine settimana, il nostro pannello deve essere in grado di fornire una potenza massima di circa 8,5 watt ($180W:7$

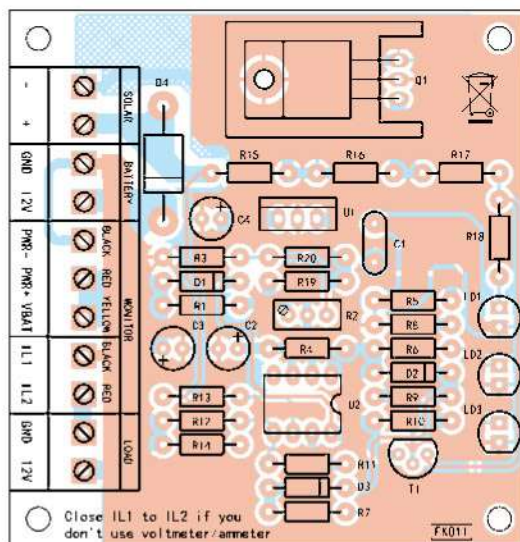
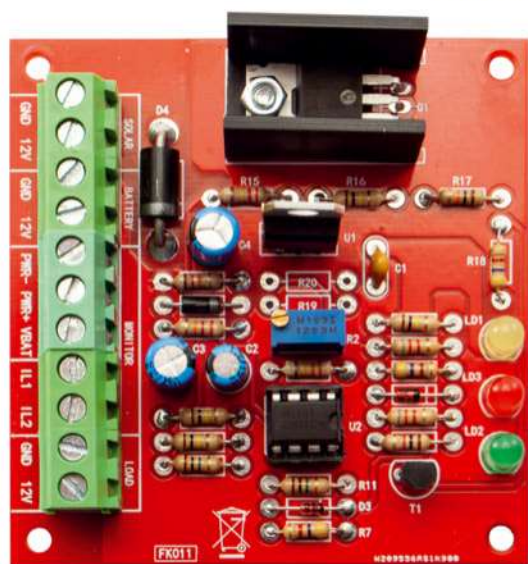
giorni:3 ore di ESH). Per le considerazioni fatte in precedenza, è consigliabile che la potenza massima da considerare sia almeno il doppio. Nel nostro caso abbiamo utilizzato due pannelli da 12 watt ciascuno per complessivi 24 watt. Non resta ora che scegliere il tipo di pannello fotovoltaico tra quelli disponibili in commercio: amorfo o cristallino. I pannelli del primo tipo presentano un rendimento decisamente più basso (attorno al 5-6 per cento) ma funzionano bene anche con scarsa illuminazione mentre quelli cristallini se non sono ben illuminati non erogano neppure un milliampère. Essendo il costo per watt molto simile, è consigliabile utilizzare i pannelli del primo tipo al Nord mentre al Sud vanno decisamente meglio i pannelli mono o policristallini. Questi ultimi, presentando un rendimento migliore, sono indicati anche nei casi in cui ci siano problemi di ingombro. Mettendo in pratica questi semplici concetti, potremo modificare a nostro piacere le caratteristiche dell'impianto. A questo punto, dopo i pannelli e la batteria, bisogna spendere due parole su un altro componente indispensabile per realizzare un sistema che si rispetti: il regolatore di carica. Questo circuito controlla il livello di carica della batteria e quando quest'ultima risulta completamente carica interrompe il collegamento con i pannelli. In questo modo si evita il surriscaldamento dell'accumulatore e la conseguente riduzione del numero di cicli di lavoro. Resta inteso che non appena il livello della batteria scende sotto un valore prefissato, il circuito ripristina il collegamento con i pannelli.

poter attivare il circuito di ricarica ed infine, quello verde (LD2), che la batteria è completamente carica. Questo regolatore di carica può essere utilizzato con pannelli che erogano una corrente massima di 3-5 ampère; è possibile aumentare la portata montando per D4 un diodo di maggiore corrente diretta, fino a 10-20.

Il regolatore facente capo all'integrato a tre terminali U1, adjuvato dal condensatore di filtro C1 e dal diodo di protezione D1, permette di alimentare

con una tensione stabilizzata il piedino d'ingresso invertente dell'operazionale U2a; la parte del circuito che deve provvedere al rilevamento dello stato di carica della batteria è invece alimentata a monte, ossia dalla tensione presente tra il catodo del diodo D1 e massa.

A questo punto è opportuno spiegare il significato della morsettiere MONITOR che vedete nella parte di sinistra dello schema elettrico: l'abbiamo prevista per consentire l'inserzione nel circuito di



Elenco Componenti:

R1: 22 kohm
R2: Trimmer 10 kohm multigiri
R3: 12 kohm
R4: 330 kohm
R5, R12, R13, R16: 100 kohm
R6, R7, R18: 4,7 kohm
R8, R9: 8,2 kohm
R10: 1 kohm
R11, R14: 1 Mohm
R15: 220 kohm
R17: 10 kohm
R19, R20: -

C1: 100 nF ceramico
C2: 1 μ F 63 VL elettrolitico
C3: 22 μ F 25 VL elettrolitico
C4: 47 μ F 25 VL elettrolitico
D1: 1N4007
D2, D3: 1N4148
D4: 1N5408
LD1: LED 5 mm giallo
LD2: LED 5 mm verde
LD3: LED 5 mm rosso
U1: 7805
U2: LM358

T1: BC557
Q1: IRF540

Varie:

- Zoccolo 4+4
- Dissipatore ML26
- Vite 10 mm 3 MA
- Dado 3 MA
- Morsetto 2 vie passo 5 mm (4 pz.)
- Morsetto 3 vie passo 5 mm (1 pz.)
- Circuito stampato FK011 (69x72 mm)

un misuratore di energia (Energy Meter, se preferite...) commerciale, allo scopo di mostrare su un display tensione e corrente erogate dalla batteria all'utilizzatore. Nel caso specifico, dopo varie prove abbiamo optato per il prodotto EPT33V3AOLED della Futura Elettronica, un piccolo strumento da pannello con display OLED blu. È in grado di visualizzare una serie di parametri elettrici come tensione, corrente, potenza, consumo energetico, capacità, temperatura ambiente e tempo grazie a una serie di fili per collegarlo in parallelo al circuito di cui misurare la tensione e per inserirlo nel ramo di cui rilevare l'assorbimento. L'alimentazione del modulo è a tensione continua compresa fra 4 e 20 Vcc e può essere in comune con la tensione da misurare o distinta.

Questo modulo è fornito con due cavetti già intestati nei connettori sul suo PCB visibili dal retro, il cui significato è dettagliato nella **Tabella 1**.

I cavetti in dotazione sono due: uno tripolare con

un connettore a passo 2,54 mm ed uno bipolare con connettore a passo 5,08 mm; il primo ha un filo rosso ed uno nero per alimentare l'elettronica del modulo ed uno giallo da collegare al positivo della fonte di alimentazione di cui misurare la tensione, che nel nostro caso sarà la batteria. Il secondo ha due fili di maggiore sezione perché sono quelli da cui entrerà (rosso) ed uscirà (nero) la corrente dello shunt utilizzato per la misura di corrente. Siccome il circuito di misura della corrente è sollevato dalla massa del circuito, quando l'alimentazione del modulo è derivata da quella di cui misurare la tensione e la corrente il costruttore suggerisce di non collegare a massa il filo nero sottile, ma di lasciarlo sconnesso e isolato con del nastro isolante.

Resta inteso che l'Energy Meter non è strettamente indispensabile, ma costituisce un utile accessorio; in base al fatto che sia o meno inserito, cambiano le connessioni della morsettiera MONITOR.

Se lo inserite nel circuito, collegate così i contatti della morsettiera MONITOR:

- 2 (PWR+) al filo rosso sottile (+Vcc) dello strumento;
- 3 (VBAT, che è collegato a PWR+) al filo giallo dello strumento;
- 4 (IL1) al filo nero spesso dello strumento;
- 5 (IL2) al filo rosso spesso dello strumento.

Il tutto, secondo lo schema di cablaggio proposto nella **Fig. 2**.

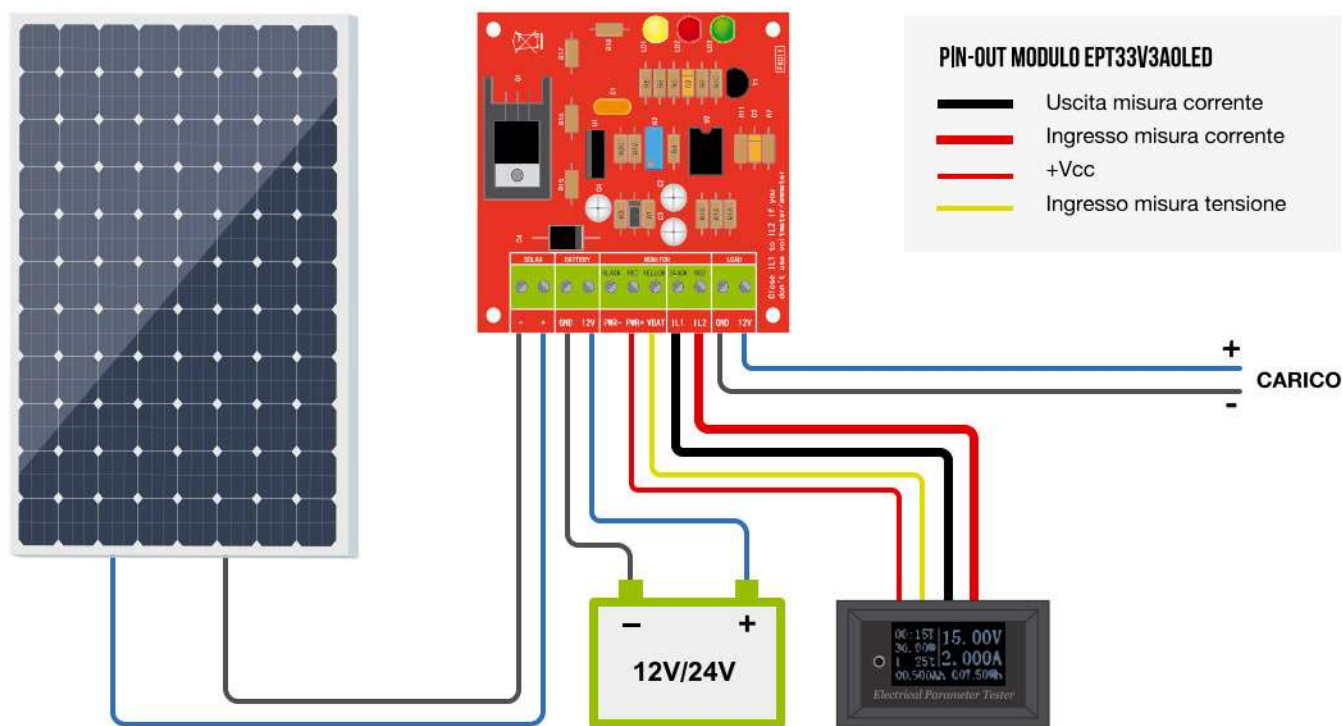
Se invece non volete montare l'Energy Meter dovete applicare lo schema di connessione già esposto nella **Fig. 1**, che rappresenta l'installazione base del sistema; quindi dovete ponticellare i morsetti IL1 e IL2 tra loro perché altrimenti non si chiude il circuito di alimentazione del carico (uscita LOAD) sul lato di massa.

REALIZZAZIONE PRATICA

Passiamo adesso ad occuparci della realizzazione del circuito, per il quale abbiamo disegnato un apposito PCB di dimensioni particolarmente contenute sul quale abbiamo montato tutti i componenti occorrenti, del quale sul nostro sito Internet www.elettronica.it trovate la traccia lato

rame da scaricare per ricavare la pellicola necessaria a procedere con la fotoincisione. Inciso e forato lo stampato, montate su di esso le resistenze e i diodi; nell'inserire i diodi rammentate che il terminale vicino alla fascetta colorata è quello di catodo. Infilate e saldate lo zoccolo a 8 piedini per il doppio operativo, quindi i due trimmer e il transistor T1, ricordando di orientare quest'ultimo (il suo lato piatto deve stare rivolto verso la resistenza R10) come indicato nel piano di montaggio visibile in queste pagine. È poi la volta dei condensatori, dando la precedenza a quelli non polarizzati e rispettando la polarità indicata per gli elettrolitici; montate il MOSFET di potenza, che va disposto adagiato con la parte metallica appoggiata ad un piccolo dissipatore ad "U" tipo ML-026 (molto comune...) dopo aver interposto della pasta termica al silicone, fissandolo poi al dissipatore mediante una vite 3MA con il suo dado. Il regolatore U1 va invece montato in piedi, mantenendolo con la parte metallica rivolta verso il dissipatore del MOSFET. Fatto ciò il circuito è pronto; introduce l'integrato U2 nel proprio zoccolo, facendo attenzione al fine di farlo entrare con il riferimento dalla parte indicata nel piano di montaggio e verificate che durante l'operazione non si sia piegato qualche

Fig. 2
Cablaggio del sistema di alimentazione fotovoltaica con accumulo ed Energy Meter.



piedino, quindi date un'occhiata generale al circuito in modo da controllare se è tutto in ordine. Fatto ciò, il dispositivo è pronto all'uso.

La basetta dispone di una morsettiera alla quale fanno capo tutti i collegamenti; per le connessioni conviene utilizzare dei robusti capicorda che dovete saldare in corrispondenza dei punti marcati SOLAR + e -, BATTERY + e - (12V), e LOAD + e -. Ai morsetti SOLAR + e - dovete collegare, rispettivamente, il morsetto positivo ed il negativo del pannello solare, utilizzando cavo di sezione adeguata (mediamente 1 mmq per 2,5 ampere); l'accumulatore va collegato ai contatti + e - BATTERY, rispettando la polarità indicata e utilizzando filo di sezione pari ad 1 mmq ogni 2,5 A di corrente. Le stesse considerazioni valgono per i fili elettrici che dovete impiegare per collegare l'utilizzatore desiderato ai punti + e - LOAD del circuito stampato. La verifica del funzionamento del circuito e la taratura si possono effettuare al banco; a tale scopo, senza connettere né la batteria né il pannello fotovoltaico (o i pannelli), collegate al posto della batteria un alimentatore in grado di erogare una tensione continua di 14,4 volt esatti. In questo stato il LED rosso LD3 deve risultare acceso, in quanto la tensione del pannello (che non c'è) è nulla. Lo stadio corrispondente inibisce il funzionamento del MOSFET per cui il LED giallo LD1 è sicuramente spento. Il LED verde può invece essere acceso o spento a seconda di come è regolato il trimmer R2. Se il LED è acceso, ruotate il cursore a vite del trimmer R2 sino ad ottenere lo spegnimento del diodo luminoso, quindi ruotate lentamente il cursore nella direzione opposta sino all'accensione. Se invece il LED è già spento, ruotate il cursore dell'R2 sino ad ottenerne l'accensione. A questo punto collegate la batteria ed utilizzate l'alimentatore al posto del pannello solare collegando il positivo ed il negativo rispettivamente ai morsetti + e - SOLAR.

Aumentate a poco a poco la tensione sino ad ottenere lo spegnimento del LED rosso e l'accensione di quello giallo. Quest'ultimo non si accende se anche LD2 è acceso. L'alimentatore utilizzato per questa prova deve essere munito di protezione amperometrica per evitare un'eccessiva corrente di carica (la protezione va regolata a 1+2 ampere). Ricordiamo infine di utilizzare dei cavi di diametro adeguato alle correnti in gioco, allo scopo di ridurre al minimo le perdite dovute ai collegamenti elettrici. Il nostro circuito è in grado di gestire la ricarica da pannello fotovoltaico ed alimentare direttamente tutti i dispositivi funzionanti in c.c. a 12 volt. Nel

caso di apparecchiature funzionanti a 230 Vca è necessario ricorrere ad inverter che consentono di ottenere, dai 12 volt continui, una tensione alternata adatta ad alimentare gli utilizzatori previsti per funzionare con la rete-luce domestica. Come inverter, se utilizzate apparecchi sensibili alle forme d'onda o ai disturbi radioelettrici, vi suggeriamo di preferire uno di quelli a sinusoide ricostruita, invece del tradizionale (ed oggi un po' superato) inverter con uscita ad onda quadra.

Rammentiamo che la batteria ed il pannello solare devono "andare d'accordo" per quanto riguarda la tensione: in definitiva se il pannello fornisce 12+20 volt (tipica escursione di un pannello da 12V nominali...) bisogna utilizzare una batteria da 12V; la capacità va rapportata alla corrente erogabile dal pannello e al tempo per cui potrà farlo, perché non ha senso prendere batterie di capacità elevata se poi non verranno mai caricate a pieno.

CONCLUSIONI

Il progetto qui proposto, ossia il circuito descritto, è un completo gestore di un piccolo impianto fotovoltaico con accumulo in batteria da 12V; abbinato ad uno o più pannelli fotovoltaici e ad una batteria, fornisce autonomia energetica ad un impianto elettrico di piccola potenza in continua oppure, aggiungendo un inverter 12Vcc/230Vca, in alternata. Riguardo alla possibilità di collegare due o più pannelli solari in parallelo, precisiamo che per compensare le differenze di tensione fra essi è opportuno porre in serie a ciascuno un diodo di potenza, capace di reggere una corrente diretta (If) superiore alla massima erogabile da ciascun pannello; suggeriamo di connettere il diodo sul cavo positivo di ciascun pannello, con l'anodo verso il filo + e il catodo sul morsetto +SOLAR. I negativi andranno uniti sul morsetto - SOLAR. 



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. Il regolatore di carica (cod. FK011) è in vendita a Euro 18,00 mentre il modulo energy meter (cod. EPT33V3AOLED) è disponibile a Euro 26,00. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

La TV cambia? Non cambiarla!

Arriva la nuova TV digitale, acquista il decoder senza cambiare la TV!

Decoder TV digitale terrestre DVBT, DVBT2, H.265/HEVC



DVBT/T2
TERRESTRIAL

**H.265
HEVC**

**FULL HD
1080**

USB

HDMI

mpeg
VIDEO

XVID
VIDEO

COD. GS240T2
€ 26,90

**NUOVO
MODELLO**

GoSat GS240T2 è un decoder digitale terrestre di nuova generazione compatibile con il nuovo standard di trasmissione DVBT, DVBT2 H.265/HEVC. L'apparecchio può ricevere comunque trasmissioni nei vecchi standard DVB-T H.264 e MPEG-4. Il dispositivo offre una perfetta qualità delle immagini in Full HD 1080p. Grazie alla porta USB, disponibile sulla parte frontale, e alla funzione PVR (Personal Video Recorder), permette la registrazione di programmi TV e radio su chiavetta USB o su hard disk esterno. Può essere inoltre connesso a internet tramite il connettore RJ45 per utilizzare servizi come YouTube e guardare canali IPTV (liste con estensioni M3U). Il telecomando in dotazione, oltre ad avere tutte le funzioni per il controllo del ricevitore, ha la funzione di apprendimento e può essere utilizzato anche per altri dispositivi (TV, ecc.). Il pannello frontale ha un display che mostra l'ora o il numero del programma selezionato.



Prezzi IVA inclusa.

Il tuo telecomando TV è rotto?

**PRONTI
ALL'USO!**

Sostituiscilo con i telecomandi universali a infrarossi compatibili con le TV delle maggiori marche prodotte dal 2000 ad oggi. Dispongono di tutte le funzioni del telecomando originale. Subito utilizzabili!

€ 5,90
OGNI PEZZO



Cod. TVIR-SAMSUNG



Cod. TVIR-SONY



Cod. TVIR-PHILIPS



Cod. TVIR-LG



Cod. TVIR-PANASONIC

DA PROGRAMMARE

Per comandare fino a due dispositivi audio/video.

€ 9,00
cod. TVFREEDOM2



DISPONIBILE ANCHE 1 in 1
cod. TVFREEDOM1

€ 8,50

**FUTURA
ELETTRONICA®**

www.futurashop.it

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

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

MAKER UNO

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

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

+

50
video
tutorial



€ 58,00

cod. MAKERKITBASE

**FUTURA
ELETTRONICA®**

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

Prezzo IVA inclusa

Caratteristiche tecniche
e vendita on-line su:

www.futurashop.it

REGOLATORE VELOCITÀ VENTOLE

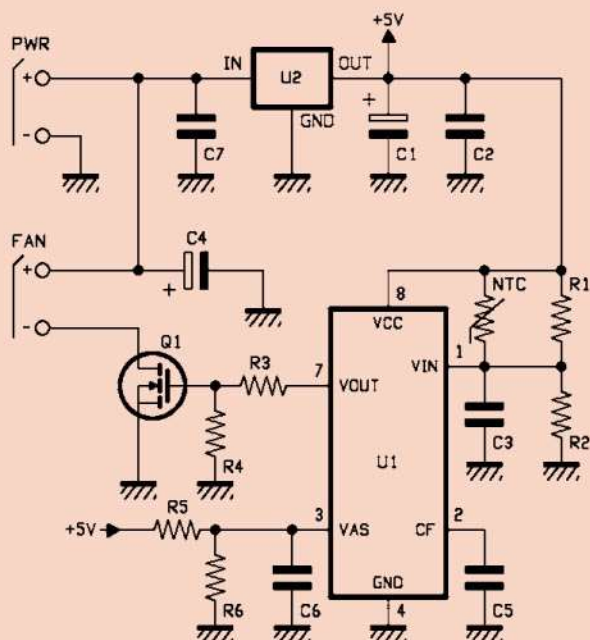
di DAVIDE SCULLINO



Adeguare e stabilizza la velocità di rotazione di elettroventole in corrente continua in funzione della temperatura rilevata nell'area o sul dissipatore che deve essere raffreddato.

S

iamo un po' tutti abituati a vedere le ventole di raffreddamento dei sistemi elettronici come qualcosa che si accende, fornendo la tensione nominale, quando ci serve e si spegne quando non occorre più; in quest'ottica vengono realizzati i classici controlli termostatici, basati su interruttore termostatico tradizionale o elettronico allo stato solido, che operano una gestione di tipo on/off asservita alla temperatura rilevata da una lamina bimetallica o da un termistore NTC. Questo tipo di controllo è però poco efficiente sotto l'aspetto energetico e poco versatile, ragion per cui abbiamo pensato di progettare un controller che fosse in grado di adeguare il flusso d'aria della ventola e quindi il suo numero di giri (velocità angolare) alla temperatura dell'elemento



ventilato, in maniera lineare. Otteniamo così una velocità della ventola proporzionale alla temperatura ed un controllo rispondente alle necessità di raffreddamento, per un controllo acustico e una maggiore durata della ventola.

SCHEMA ELETTRICO

Il progetto qui descritto è un regolatore di velocità di elettroventole in modalità switching per l'uso con motori CC a spazzole (brushed) o senza spazzole (brushless). Il controllo della velocità proporzionale alla temperatura viene eseguito utilizzando la modulazione della larghezza degli impulsi, anche nota come tecnica PWM. L'elemento sensore utilizzato per rilevare la temperatura è un termistore NTC da 10 kohm a 25 °C.

Il circuito è stato realizzato utilizzando il chip TC648 della Microchip, configurato in modalità di spegnimento automatico, sulla quale torneremo tra breve.

Il controller ventola

Il TC648 è un controller di velocità delle ventole switching PWM, del quale vedete lo schema a blocchi in **Fig. 1**.

Nel TC648, il controllo della velocità angolare proporzionale alla temperatura viene eseguito attraverso un modulatore PWM, composto da un comparatore che confronta una tensione continua

applicata al piedino Vin con una forma d'onda a dente di sega prodotta dal blocco Clock Generator. Un termistore (o altro sensore di temperatura con uscita in tensione) collegato all'ingresso Vin del comparatore fornisce la tensione di controllo richiesta, che è tipicamente compresa fra 1,25V e 2,65 V, escursione cui corrisponde un duty-cycle che spazia dallo 0% al 100%.

Il TC648 può essere configurato per funzionare in modalità di spegnimento automatico o in modalità velocità minima. In modalità di spegnimento automatico, il funzionamento della ventola viene sospeso automaticamente quando la temperatura misurata (attraverso la tensione che raggiunge Vin) è inferiore a quella programmata dall'utente come impostazione minima (Vas). La ventola è automaticamente riavviata e il controllo proporzionale della velocità ripristinato, quando Vin supera Vas (sommando l'isteresi del comparatore). Durante lo spegnimento, la corrente di alimentazione scende a 25 µA (valore tipico)

L'operatività in modalità di velocità minima è simile a quella a spegnimento automatico, con l'eccezione che la ventola, una volta raggiunta la temperatura minima (ossia quando viene misurata la temperatura più bassa) viene mantenuta in funzione ma all'impostazione minima programmata dall'utente. Un timer di avvio integrato garantisce un affidabile avviamento del motore della ventola all'accensione e quando si esce dalla modalità di spegnimento o spegnimento automatico.

L'integrato dispone anche di una linea di uscita di

CARATTERISTICHE TECNICHE

Alimentazione operativa:
da 12V a 18V

Ventola
12V - 18VCC

Carica fino a 3A
(da 12V a 18V);

Ciclo di lavoro PWM
dal 42% al 100%

Spegnimento della ventola
quando la temperatura
scende a 25 °C

Ventola a bassa velocità
(ciclo di lavoro del 42%)
quando la temperatura
è di 25 °C

Velocità massima della ventola
(ciclo di lavoro al 100%)
quando la temperatura
supera circa i 45 °C

Frequenza PWM
30 Hz.

allarme (OTF = Over Temperature Failure) di guasto per sovratemperatura, la quale viene attivata (ovvero si porta a zero logico) quando il segnale PWM raggiunge un duty-cycle del 100%, indicando una possibile situazione di deriva termica; in breve, questa uscita indica che la ventola potrebbe non essere sufficiente a raffreddare l'apparato cui è applicata.

Come accennato poc'anzi, nel nostro progetto utilizziamo l'integrato nella modalità di spegnimento automatico, dove il funzionamento della ventola viene sospeso automaticamente quando la temperatura misurata è inferiore a 25 gradi centigradi. La ventola si riavvia automaticamente e il controllo proporzionale della velocità viene ripristinato quando la temperatura supera i 25 gradi centigradi, fatta salva l'isteresi prevista nel comparatore per evitare il pendolarismo del controllo, ovvero che l'uscita del comparatore stesso continui ad invertire il proprio stato quando VIN presenta un potenziale nell'intorno di quello presente sul piedino VAS.

Bene, spiegato il funzionamento dell'integrato, vediamo in che modo si applica al nostro circuito, analizzando lo schema elettrico: come vedete il TC648 (siglato U1) è l'elemento circuitale principale e il segnale PWM presente al suo piedino di uscita (VOUT) pilota, tramite il partitore resistivo formato da R3 ed R4, il gate di un MOSFET enhancement-mode a canale N (Q) la cui funzione è pulsare la tensione ai capi dell'elettroventola, in modo da impostarne la velocità angolare.

Il MOSFET funziona da interruttore statico e ad ogni impulso ricevuto sul gate chiude praticamente a massa il polo negativo del motore della ventola, il cui positivo è collegato alla linea +PWR dell'alimentazione del circuito.

Il riferimento di temperatura viene fornito all'integrato dal termistore NTC, collegato all'ingresso VIN, sul quale arriva anche la tensione ottenuta dal partitore resistivo R1/R2 che stabilisce la massima escursione sul predetto piedino; per l'esattezza, NTC fa partitore con la R2, mentre la resistenza R1 serve ad evitare che la tensione su VIN scenda eccessivamente quando la temperatura scende al disotto della soglia o salga troppo a temperature più elevate di quella prevista.

Il termistore è da 10 kohm alla temperatura di riferimento di 25 °C. Il condensatore C3 è stato inserito per filtrare la tensione su VIN in modo da evitare che fluttuazioni nella linea di alimentazione o disturbi captati dai collegamenti, nonché la pulsazione del PWM sulla ventola e il carico che

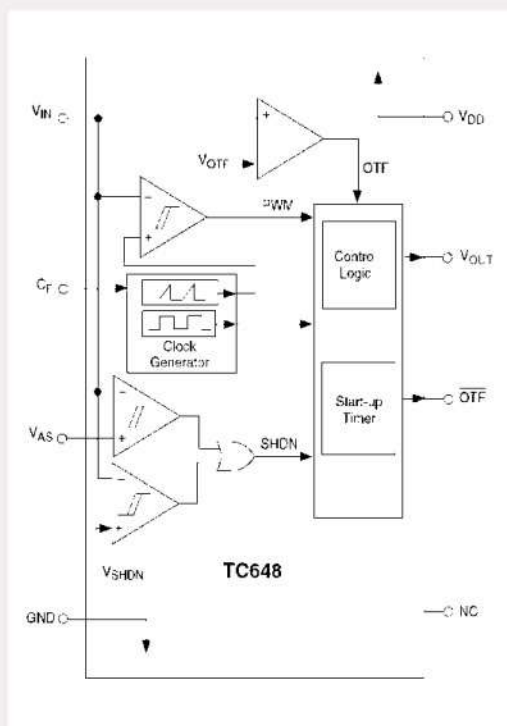
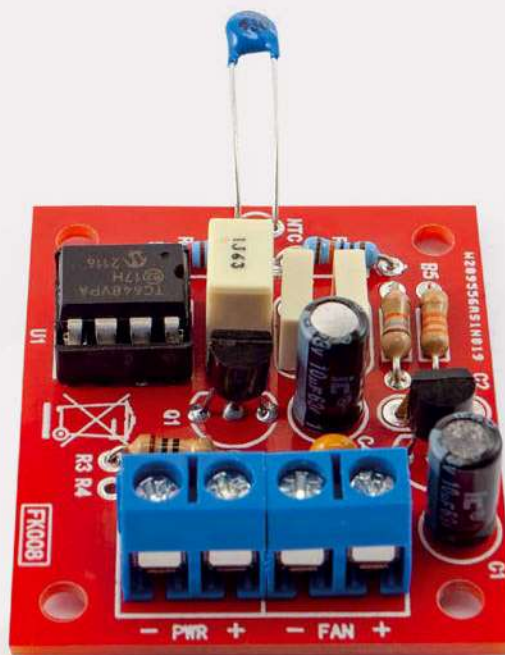


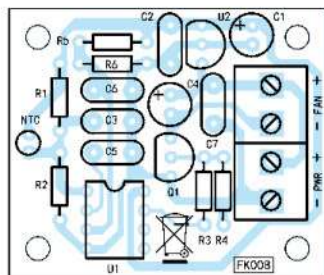
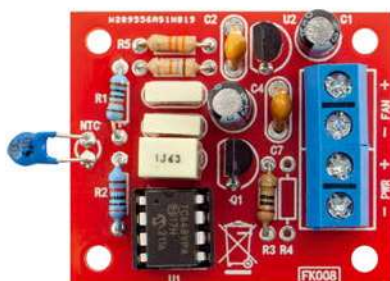
Fig. 1
Schema
a blocchi
dell'integrato
TC648.

ne deriva, possano alterare in funzionamento del modulatore, bloccandolo o determinando false commutazioni.

Procediamo con l'analisi dello schema elettrico analizzando il partitore resistivo formato da R3



Le
morsettiere
riportano
l'uscita di
comando e
l'ingresso di
alimentazione.



Elenco Componenti:

R1: 75 kohm	U1: TC648VPA
R2: 1 kohm	U2: 78L05
R3: 10 ohm	NTC: NTC 10 kohm
R4: -	
R5: 33 kohm	Varie:
R6: 18 kohm	- Zoccolo 4+4
C1, C4: 10 µF 63 VL elettrolitico	- Morsetto 2 poli passo 5 mm (2 pz.)
C2, C7: 100 nF ceramico	- Circuito stampato FK008 (36x42 mm)
C5: 1 µF 63 VL poliestere	
C3, C6: 10 nF 63 VL poliestere	
Q1: BS170	

e R5, collegato all'ingresso VAS, il quale imposta la soglia di spegnimento automatico. Lo spegnimento automatico si verifica quando la tensione sull'ingresso VIN è minore o uguale a quella su VAS. La ventola si riavvierà automaticamente quando il potenziale su VIN diverrà maggiore o uguale a quello sul VAS, sempre considerando l'isteresi del comparatore.

Per come sono stati dimensionati i partitori, il duty-cycle del segnale PWM varierà dal 42% al 100%; il 42% è il setpoint per la velocità minima dell'elettroventola e oltre questo valore, cioè se il comparatore dovesse portarsi al disotto di esso, avverrebbe lo spegnimento della ventola stessa. La frequenza del PWM è determinata da quella del segnale a dente di sega prodotto internamente all'U1 e dipende dal valore del condensatore collegato al pin CF e quindi, nel nostro circuito, dal C5;

nel nostro caso la frequenza impostata è di 30 Hz e costituisce anche la base dei tempi per il timer di avvio (startup della ventola).

Il timer di avvio serve per garantire un avvio affidabile della ventola, nel senso che provvede ad un soft-start, quindi avvia la ventola facendola accelerare gradualmente per evitare sollecitazioni sia meccaniche del suo motore, sia elettriche del circuito, oltre che spike di corrente che possono generare disturbi al circuito stesso e ad altri apparati elettronici nei dintorni.

Il timer di avvio attiva l'uscita VOUT per 32 cicli del PWM ogni volta che la ventola viene avviata dallo stato spento; ciò si verifica all'accensione e quando si esce dalla modalità di spegnimento o spegnimento automatico. Il tempo di avvio è di circa un secondo, con frequenza del PWM di 30Hz.

Vediamo ora come si effettua il calcolo della modalità di spegnimento automatico del TC648, iniziando con il partitore R1 e R2, dimensionato sull'utilizzo di un NTC avente una resistenza di 10 kΩ a TMIN (25°C) e 4,65 kΩ a TMAX (45°C) per rientrare nella finestra di tensione suaccennata. I valori che ne risultano sono: R1 = 20 kΩ R2 = 3K6 kΩ.

Per impostare le temperature di accensione e di spegnimento automatico, il costruttore rende disponibili queste due formule:

$$V(T1) = (VDD \times R2) / R_{temp}(T1) + R2$$

e

$$V(T2) = (VDD \times R2) / R_{temp}(T2) + R2$$

In queste formule, T1 e T2 sono le temperature scelte e Rtemp è il parallelo tra il resistore R1 e il termistore NTC. Le due equazioni semplificano la risoluzione delle due variabili incognite R1 e R2. Per quanto riguarda il pin VAS, l'impostazione della soglia di 1,8 V si ottiene con R3 = 33 kΩ R5 = 18 kΩ. È possibile ottenere vari intervalli di velocità (duty-cycle) modificando il valore del resistore.

Ciò detto, passiamo allo stadio di alimentazione del nostro controller, che parte dalla morsettiera di ingresso ±PWR cui si applica la tensione principale, la quale opportunamente filtrata dal condensatore C7 viene portata a 5 volt ben stabilizzati tramite il regolatore integrato U2, che è un classico 78L05, ossia il 7805 in case TO-92. L'uscita di questo componente viene ben filtrata da C1 (per ridurre l'eventuale ripple dell'alimentatore in alternata) e da C2 (per sopprimere eventuali interferenze impulsive e ad alta frequenza) quindi i 5 volt raggiun-

TERMOSTATO SMART: RIMEDIO ALLA CRISI ENERGETICA

Una dei modi per difendersi dalle conseguenze della crisi energetica è rendere smart la climatizzazione degli ambienti con un termostato smart; un valido esempio è Black della Tado, che consente un risparmio 22%; ad esso viene abbinato l'abbonamento al servizio Tado Balance, che attingendo ai dati in Rete permetterà di scegliere l'orario più adatto in cui riscaldare la casa, consentendo anche un risparmio sui costi



del 38%. Tutto quello che serve è un contratto di fornitura elettrica che segua i prezzi orari del mercato, invece che le fasce orarie. Tado Balance prende le informazioni sui prezzi dalla stessa fonte dei fornitori di energia, quindi ha bisogno di conoscere l'area geografica in cui si trova il cliente. L'app consente di impostare la flessibilità del termostato.

gono l'alimentazione (piedini 8 e 4) del TC648. Il circuito di comando della ventola è invece derivato direttamente dall'ingresso PWR. A proposito di ventola, il MOSFET Q1 e le piste del circuito stampato del nostro progetto sono calcolati per pilotare elettroventole con assorbimento fino a 3A, un valore che sicuramente soddisfa la gran parte delle applicazioni e che permette di gestire gruppi di elettroventole per raffreddare server, apparati di potenza ed anche vani, giacché 3 ampere a 12 volt significa disporre di una potenza di 36W; se consideriamo che un'elettroventola da PC assorbe tipicamente una cinquantina di milliampere, capite cosa si può fare con il nostro controller. Notate che alcune ventole richiedono un segnale PWM per funzionare; in questo caso potete utilizzare il gate del MOSFET per prelevare il segnale PWM diretto.

REALIZZAZIONE PRATICA

Bene, esaurita la teoria possiamo passare alla parte costruttiva di questo progetto: tutti i componenti necessari prendono posto su un circuito stampato a singola lamina e pertanto facilmente realizzabile anche da chi è alle prime armi, per fotoincisione (dopo aver scaricato dal nostro sito web www.elettronica.in.it e stampato su un foglio di acetato o carta da lucido la traccia lato rame) o tramite il servizio PCBPRODUCTION di Futura Elettronica. Una volta ottenuto il PCB, potete montarvi i pochi componenti che servono, tutti a montaggio tradizionale: anche questa caratteristica rende adatto il montaggio anche ai principianti e a chi non ha l'attrezzatura per saldare gli SMD. I primi elementi da saldare sono quelli a più basso

profilo, quindi resistenze e zoccolo DIP a 4+4 pin per l'integrato TC648, quindi si procede con i condensatori, prestando attenzione alla polarità di quelli elettrolitici, con il regolatore 78L05 (da orientare con il lato piatto rivolto al condensatore C2) e con le due morsettiere a passo 5 mm per l'alimentazione (+ e - PWR) e l'uscita di controllo della ventola (+ e - FAN). Completate il circuito inserendo nello zoccolo l'integrato U1, badando di orientarlo come mostrato nel piano di montaggio e verificando che nell'inserzione non si pieghi alcuno dei terminali sotto al corpo.

Fatto ciò il controller è pronto per l'uso, perché non ha alcun elemento da tarare: alimentate la morsettieria + e - PWR con una tensione continua idonea a quella della ventola (tipicamente 12Vcc) mediante un alimentatore in grado di erogare la corrente richiesta dalla ventola stessa, più una ventina di milliampere occorrenti al controller per funzionare. La ventola va connessa alla morsettieria FAN, rispettando la polarità indicata, altrimenti girerà al contrario o non girerà affatto. ■



Cosa occorre?

Il progetto presentato in questo articolo è disponibile in scatola di montaggio (cod. FK008) a Euro 12,00. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

HDMI
↓
VGA

€ 11,50

Cod. **BOXHDMITOVGA**

CONVERTITORE DA HDMI A VGA 1080P PLUG AND PLAY

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

CONVERTITORE DA AV A HDMI PLUG AND PLAY

AV
↓
HDMI

€ 14,00

Cod. **BOXAVTOHDMI**

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

Convertitori di segnale

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

VGA
↓
HDMI

€ 11,50

Cod. **BOXVGATOHDMI**

CONVERTITORE DA VGA A HDMI PLUG AND PLAY

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

CONVERTITORE DA HDMI AD AV PLUG AND PLAY

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

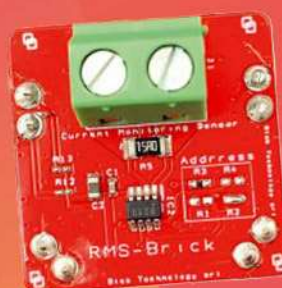
HDMI
↓
AV

€ 11,00

Cod. **BOXHDMITOAV**

ENERGY METER CON BLEBRICKS

di BORIS LANDONI



Utilizziamo la piattaforma di prototipazione a "mattoncini" per implementare un monitor di potenza assorbita ed energia consumata.

Da anni il problema della consapevolezza dei consumi energetici e del loro impatto ambientale è un tema di primo piano, tanto che anche noi, adeguandoci di volta in volta alle esigenze e cercando di soddisfarle via-via con i componenti del momento, proponiamo progetti a tema, tra i quali spiccano i misuratori di consumo (kWh) e assorbimento (kW) e i limitatori di carico delle linee elettriche domestiche. Di Energy Meter ne abbiamo proposti diversi, basati su hardware più o meno convenzionale, però ora vogliamo provare a rifare la stessa cosa in chiave Maker, ricorrendo a una piattaforma di prototipazione "istantanea" chiamata Blebricks e caratterizzata da un sistema hardware modulare facilmente interfacciabile anche con MIT App Inventor e di semplice assemblaggio.

Fig. 1
L'insieme
dell'hardware
basato su Blebricks



I moduli Blebricks sono mattoncini realizzati con una tecnologia brevettata grazie alla quale, combinando ad incastro quelli desiderati, si possono creare nuovi dispositivi elettronici che comunicano in wireless. Questa tecnologia permette di creare dispositivi anche senza avere competenze tecniche di elettronica o di programmazione.

La prototipazione è istantanea e, per questo motivo, incontra l'interesse di chi vuole realizzare un dispositivo in modo semplice e veloce, senza dover o saper fare saldature, né realizzare circuiti stampati, né programmare!

I "mattoncini" sono così detti perché si tratta di moduli che si incastrano tra loro realizzando le necessarie interconnessioni tramite appositi connettori incorporati, senza bisogno di collegamenti esterni. La base di questo sistema è il modulo BLE-B, delegato alla gestione operativa del sistema e delle comunicazioni wireless, che avvengono tramite Bluetooth (portata di circa 20 metri) grazie al transceiver Bluetooth Low Energy BLE 4.2 integrato nel modulo stesso. Proprio utilizzando questo modulo ed una scheda adattatrice cui applichiamo un modulo d'interfaccia per sensori di corrente, abbiamo implementato il progetto descritto in queste pagine, che è uno speciale Energy Meter con il quale cogliamo l'occasione per farvi vedere come, con grande facilità, sia possibile costruire un progetto del genere, interfacciato ad uno smartphone che farà da interfaccia utente, appoggiandosi alla piattaforma Blebricks. Più esattamente, il sistema trasmetterà tramite Bluetooth informazioni sullo stato dell'assorbimento

di corrente e potenza elettrica, nonché del consumo istantaneo, visualizzando allarmi quando il consumo supera la soglia che potremo impostare sempre tramite lo smartphone.

L'HARDWARE DEL SISTEMA

Dal punto di vista dell'elettronica occorrente, il nostro progetto si realizza incastrando tra loro i seguenti moduli Blebricks e interconnettendoli attraverso una scheda base Blebricks IB2:

- BLE-B, come componente di base del sistema e per la comunicazione BLE;
- RMS, utilizzato per la misura della corrente alternata assorbita dal carico e per il calcolo della relativa potenza;
- due batterie AA in serie o singola batteria 3,6V per alimentare tutto il sistema.

Il modulo IB2

Il Blebrick IB2 è una base di montaggio, ossia una docking station a due slot che consente il collegamento e il fissaggio di due mattoncini Blebricks su una piattaforma orizzontale.

Il modulo RMS

Il mattoncino RMS Blebricks utilizza il noto integrato INA219 della Texas Instruments, che consente la misura istantanea della corrente alternata campionando a 200 Hz tensione e corrente, quindi calcola il valore efficace e lo invia al modulo BLE-B per la successiva elaborazione e la trasmissione via Bluetooth al dispositivo target, che nel nostro caso è uno smartphone.

L'integrato INA219 effettua fisicamente la misura della corrente nella sezione di circuito di cui siamo interessati, tramite l'utilizzo di una resistenza di shunt di piccolo valore, posta in serie al carico elettrico, nella quale scorre la corrente da misurare. L'integrato misura inoltre la caduta di tensione ai capi del resistore di valore noto e sufficientemente accurato (almeno 1%) da cui determina tramite la Legge di Ohm, il valore della corrente I in esso circolante, essendo $I = V/R$, dove V rappresenta la caduta di tensione ai capi del resistore di shunt ed R la sua resistenza.

L'INA219 è un dispositivo di current sense bidirezionale con power monitor incorporato, funzionante con un'interfaccia I²C o SMBus compatibile, che garantisce un'accuratezza dello 0,5% su tutto il range di temperatura entro il quale può funzionare (-40°C - +125°C). È disponibile in due versioni, "A" e "B" di cui la seconda garantisce maggior accuratezza di misura (±0,5%) ed è realizzato in package per montaggio SMT sia di tipo

SOIC-8 che SOT-23, entrambi ad 8 pin. L'integrato fornisce, oltre al valore della corrente circolante nello shunt, il valore della tensione V presente sulla linea che alimenta il carico, ma anche la caduta di tensione ai capi della resistenza di shunt; avendo a disposizione il valore puntuale di queste grandezze l'integrato può facilmente calcolare e fornire l'entità della potenza fornita al carico, essendo questa pari a $P=V \cdot I$. Come si può vedere dallo schema a blocchi, riportato in **Fig. 2**, all'interno dell'INA 219 si trovano un PGA (Programmable Gain Amplifier), un convertitore A/D, i tre registri relativi alle misure di corrente, tensione e potenza (quest'ultima ottenuta come prodotto delle altre due grandezze) ed un'interfaccia I²C/SMBUS compatibile. La tensione di alimentazione dev'essere compresa tra 3 e 5,5V.

Tramite il registro "MODE" è possibile configurare l'INA219 per funzionare in modalità "continua", in cui acquisisce e campiona i valori corrispondenti alle grandezze di misura e li converte in uscita senza interruzione, oppure in modalità "triggerata", in cui misura i valori di tensione e di corrente solamente in corrispondenza del verificarsi di un particolare evento di trigger. Il PGA (un amplificatore a guadagno programmabile) collocato all'ingresso della catena di acquisizione permette di impostare un guadagno di 2, 4 oppure 8 (320mV) per il valore della tensione di shunt, e due valori di fondo scala per la tensione del BUS, pari a 16 o 32V (pur restando la massima tensione applicabile pari a 26V). L'integrato è in grado di misurare sia la tensione differenziale tra IN+ e IN- sia la tensione di modo comune su IN-.

L'ADC è di tipo delta-sigma con frequenza di campionamento di 500 kHz ($\pm 30\%$); il costruttore suggerisce di applicare un semplice filtro all'ingresso dello stadio facente capo al PGA, consistente in due resistenze in serie a IN+ e IN- di valore pari a 10 Ω ed un condensatore ceramico in parallelo agli input, se ci sono transienti da misurare che hanno frequenze multiple (armoniche) della frequenza di campionamento di 500 kHz, quindi a partire da 1 MHz e superiori. La presenza delle due resistenze ha un effetto trascurabile sull'accuratezza della misura finale e fornisce una pur minima protezione degli ingressi dell'integrato nel caso di dV/dt causati principalmente da picchi di tensione derivanti da effetti induttivi legati a particolari condizioni di funzionamento.

Affinché l'INA219 sia in grado di effettuare correttamente la conversione delle grandezze che si vogliono misurare, bisogna impostare adeguatamente il valore di risoluzione nel "current register" così come il valore del resistore di shunt "Rshunt". Nel nostro caso, avendo l'intento di monitorare una linea in corrente alternata a 220V, al modulo RMS deve essere poi collegata una sonda di corrente per disaccoppiare il sistema a bassa tensione dalla rete 220V.

Nel progetto abbiamo utilizzato un trasformatore di corrente costituito da un nucleo in metallo (composto da lamierini di ferro al silicio) nel cui foro si fa passare uno dei fili della linea; la sonda in questione è un sensore amperometrico dotato di pinza che permette di misurare correnti alternate, senza dover interrompere il cavo per misurare la corrente assorbita.

Fig. 2
Schema a blocchi dell'INA219.

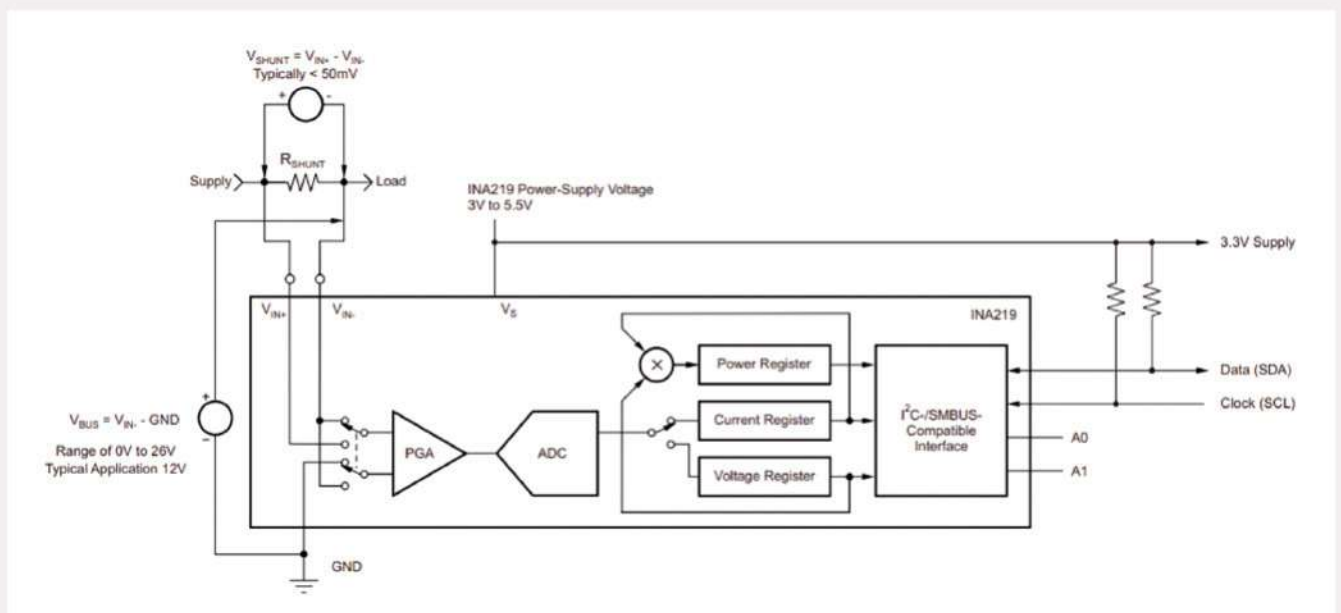


Fig. 3
Schermata della
MakeApp for
Blebricks relativa
alla gestione del
mattoncino RMS.

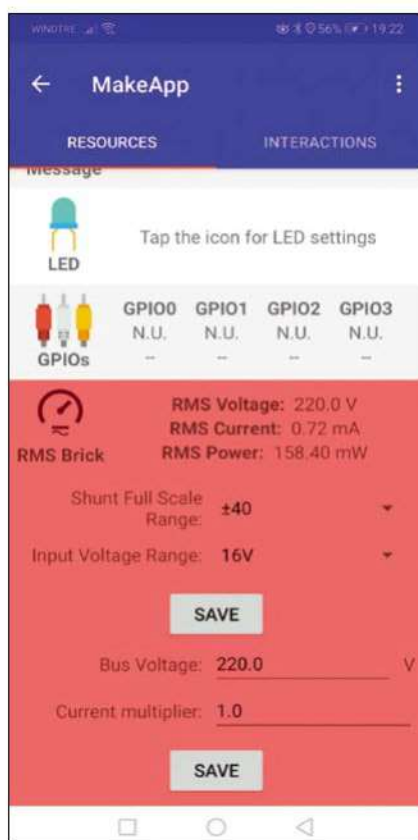
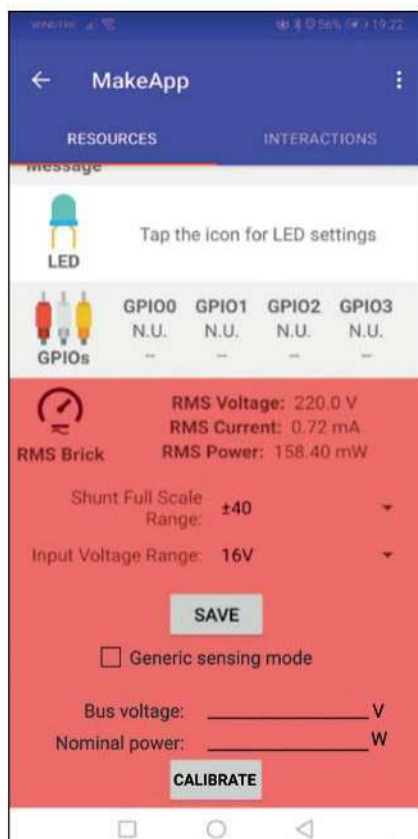


Fig. 4
Impostazione delle
attività correlate
allo stato del
modulo RMS.



In uscita fornisce una tensione alternata proporzionale alla corrente rilevata. Questo elemento è prodotto dalla YMDO e si può acquistare dalla Futura Elettronica (www.futurashop.it) con il codice prodotto 7300-THM104C. Comunque, in linea generale si può utilizzare qualsiasi altro sensore induttivo dal funzionamento similare, perché l'app che utilizzeremo permette di calibrarne il funzionamento.

La sonda da noi utilizzata offre, rispetto ad altre, il vantaggio di essere "apribile" e quindi facilmente collegabile ad un cavo di alimentazione senza doverlo sconnettere dal carico.

Il modulo BLE-B

Andiamo adesso al mattoncino BLE-B, il quale è un dispositivo intelligente che comunica tramite Bluetooth Low Energy che si configura automaticamente per lavorare insieme agli altri Blebricks. Può inviare notifiche allo smartphone in base a come viene programmato il sistema.

Il BLE-B è il mattoncino alla base di tutto il sistema e costituisce il cuore di ogni applicazione; è l'unico che deve sempre essere presente e ad esso si deve poi aggiungere un mattoncino di alimentazione ed eventuali moduli dedicati. Il BLE-B richiede un'alimentazione che può provenire da un mattoncino Blebricks ovvero da altra fonte esterna. Per impostazione predefinita, il modulo BLE-B trasmette pacchetti di advertising in modo da essere rilevato anche da altri dispositivi, permettendo di creare una rete di Blebricks.

Il blocco di alimentazione

Per alimentare l'insieme dei mattoncini abbiamo usato due batterie AA in serie collegate direttamente all'alimentazione del modulo BLE-B che garantiscono mesi e mesi di lavoro continuativo. Ma va bene anche un qualsiasi altro Blebrick di alimentazione, come ad esempio l'RPS che integra una batteria al litio ricaricabile.

Il consumo medio dell'hardware così realizzato è di circa 2 mA con il LED di segnalazione del BLE-B attivo. Tale consumo può essere ridotto disattivando tale LED attraverso la MakeApp for Blebricks, che è l'applicazione per smartphone con la quale interagire col mondo Blebricks attraverso la connessione Bluetooth.

Una volta montato il prototipo si fa passare un solo cavo della linea di alimentazione a 220V attraverso la sonda di corrente e il circuito di misura è pronto all'uso.

Passiamo ora a vedere come visualizzare i dati e

quindi all'indispensabile, quanto immediata, configurazione software.

IL SOFTWARE DEL SISTEMA

L'interazione con l'hardware e quindi la visualizzazione e gestione del funzionamento dell'Energy Meter qui proposto possono avvenire attraverso due soluzioni, entrambe valide:

- MakeApp for Blebricks, scaricabile dal Play Store di Android;
- BLEB PowerMonitor, applicazione realizzata con App Inventor (Coding).

Vediamo per prima l'operatività in associazione con la MakeApp for Blebricks, applicazione generica di Bleb Technology: da un dispositivo Android rechiamoci nel Play Store ed installiamola.

Non appena avviata, essa ricercherà i dispositivi BLE nelle vicinanze, quindi non dovremo far altro che selezionare il modulo di interesse e si aprirà la relativa pagina di configurazione.

Vengono mostrare subito le informazioni essenziali, come la distanza a cui si trova, il livello della batteria ed eventuali moduli connessi.

Tra i vari parametri possiamo impostare la potenza in trasmissione, la modalità Low Power Mode (utile a prolungare la durata della batteria) e la Data Logger, che permette di leggere e salvare in memoria i valori dei sensori ad intervalli regolari, i cui dati saranno poi leggibili tramite app.

Relativamente al modulo RMS, la MakeApp for Blebricks mostra il valore efficace istantaneo (in inglese RMS) della corrente assorbita e del relativo consumo. In tal caso basta lanciare la App e osservare i valori che appaiono sullo schermo.

La taratura avviene a livello di App, inserendo il valore di potenza nominale assorbita dal carico (spesso indicata sul dispositivo stesso), la tensione di rete (220V in Italia) e premendo il pulsante "Calibrate" mentre il carico è sotto tensione. In questo modo la corrente misurata sarà associata a quella assorbita dal carico noto.

Sempre utilizzando la MakeApp for Blebricks si possono anche impostare delle Direct Interaction nella schermata corrispondente, proposta nella

Fig. 4; tali interazioni sono sostanzialmente le azioni che l'App eseguirà in corrispondenza di determinati stati di funzionamento dell'hardware, quindi si possono definire azioni come ad esempio far suonare lo smartphone, inviare un messaggio telegram, una e-mail se il valore monitorato supera una certa soglia di potenza che avrete impostato.



Fig. 5
Schermata
dell'app
BLEB Power
Monitor.



Fig. 6
Schermata
delle
impostazioni
dell'app
BLEB Power
Monitor.

```

when SetupButton - Click
do
  if SetupFrame - Visible -
  then
    set SetupFrame - Visible - to false -
    set OperationalScreen - Visible - to true -
    call Calibrate -
  else
    set SetupFrame - Visible - to true -
    set OperationalScreen - Visible - to false -

```

```

when ExitSetupButton - Click
do
  set SetupFrame - Visible - to false -
  set OperationalScreen - Visible - to true -

```

```

when SetAlarmButton - Click
do
  call AlarmPowerValue - HideKeyboard -
  call TinyDB1 - StoreValue -
    tag AlarmPowerValue -
    valueToStore AlarmPowerValue - Text -

```

```

when CalibrateButton - Click
do
  call Calibrate -
  call AlarmPowerValue - HideKeyboard -

```

```

when BasicInteractions1 - OnOneShotCommandDone
address command
do
  call BasicInteractions1 - StartGlobalScan -

```

```

when BasicInteractions1 - OnScanFailed
do
  set ServiceLabel - Text - to Scan Restarted -
  call BasicInteractions1 - StartGlobalScan -

```

```

when Screen1 - Initialize
do
  call BasicInteractions1 - StartGlobalScan -
  call ChartMaker1 - DrawLineGraph -
    chartTitle Power Consumption -
    hAxisTitle Time -
    vAxisTitle Watts -
    labels make a list Watts -
    Amp -
    values get global GraphData -
    webView WebViewer1 -
  set CalibrateWattValue - Text - to call TinyDB1 - GetValue -
    tag WattCalibration -
    valueIfTagNotThere 0 -
  set CalibrateVoltValue - Text - to call TinyDB1 - GetValue -
    tag VoltValue -
    valueIfTagNotThere 220 -
  set AlarmPowerValue - Text - to call TinyDB1 - GetValue -
    tag AlarmPowerValue -
    valueIfTagNotThere 3000 -
  set global CurrentMultiplier - to call TinyDB1 - GetValue -
    tag CurrentMultiplier -
    valueIfTagNotThere 0.061 -

```

```

when BasicInteractions1 - OnAdvertiseReceived
address name rssi manufacturerData battery rawData
do
  if not is in list? thing get address -
    list get global AddressList -
  then
    add items to list list get global AddressList -
    item get address -
    add items to list list get global AddressAndNamesandRSSIList -
    item join get name -
    get address -
    get rssi -
    dB -
  set ListPicker1 - Elements - to get global AddressAndNamesandRSSIList -
  if get address - = get global ActualBLEBAddress -
  then
    set BlebNameLabel - Text - to get name -
    set AddressLabel - Text - to get address -
    set RSSILabel - Text - to get rssi -
    set BatteryLabel - Text - to get battery -
    call DecodeBLEBData -
    BLEBData get manufacturerData -

```

La selezione della soglia si effettua mediante il menù a tendina collocato sotto la voce "If RMS Power" visibile nella **Fig. 4** e l'azione corrispondente nella sezione "then, on".

Qui abbiamo un menù a tendina che permette di selezionare dove avverrà l'azione ed uno sottostante contenente una serie di azioni correlate al device o ambito con cui avverrà l'interazione con il sistema. Notate che la visualizzazione dell'app in questo caso propone una struttura che ricalca il software che le sta dietro, perché si tratta di qualcosa che funziona come un costrutto If – then.

BLEB POWER MONITOR

Bene, stabilito ciò passiamo a vedere il secondo approccio, ossia quello basato su un'applicazione implementata mediante MIT App Inventor: si tratta di una app per ambiente Android realizzata ad hoc per interagire con il nostro hardware, la quale propone un'interfaccia utente come quella esemplificata nella **Fig. 5**.

Tale App, una volta selezionato il BLE-B di interesse (la scelta è indispensabile perché in uno stesso ambiente possono essercene più di uno, associati a diverse utenze e individuabili con nomi differenti)

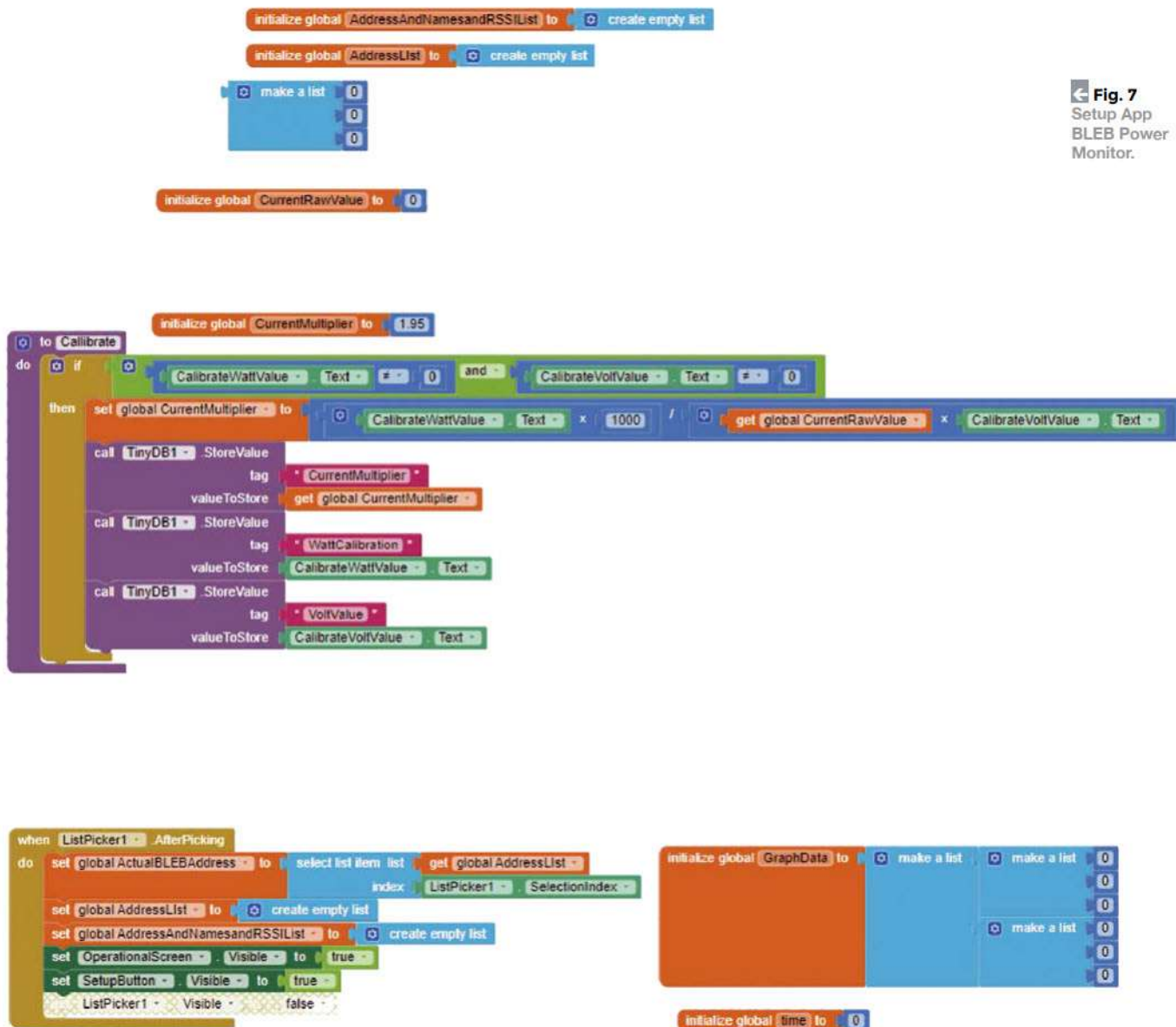
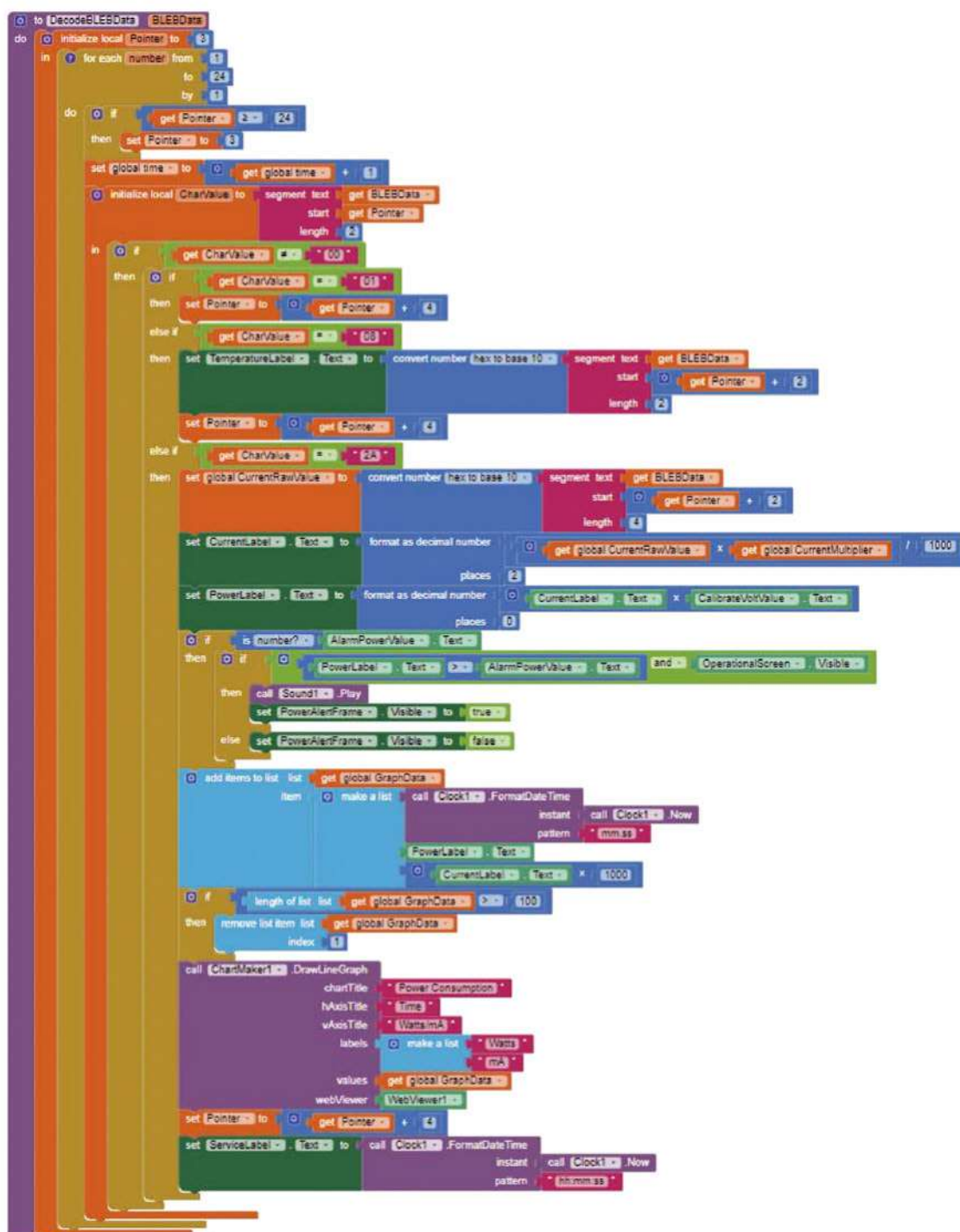


Fig. 7
Setup App
BLEB Power
Monitor.

attraverso il menù a tendina raggiungibile dalla voce "Please select your BLE-B" mostra i valori istantanei della corrente efficace, il relativo consumo istantaneo e propone un grafico (l'aggiornamento del grafico avviene ogni volta che viene ricevuto il dato, con frequenza impostabile anche da MakeApp for Blebricks). Un esempio è proposto nella **Fig. 5**. Entrando nel menù dei settings accessibile premendo sull'icona a "rotellina" (in alto a sinistra) compare la schermata a lato che è composta da tre aree principali, come proposto dalla **Fig. 6**. Nella prima area vengono visualizzati i dati generici

che mostrano il nome assegnato al BLE-B selezionato, che ad esempio può essere impostato (anche tramite la MakeApp for Blebricks) per indicare l'elettrodomestico monitorato e magari selezionarlo tra altri simili che stanno trasmettendo i dati di altri elettrodomestici, l'indirizzo BLE del BLE-B, la RSSI (signal strength indicator) il valore di temperatura ed il valore di batteria. Nella seconda area, a sfondo di colore rosso, si può settare la potenza massima oltre la quale si vuole generare e inviare la notifica e quindi la soglia di allarme.

Fig. 8
Blocco decodifica
Dati advertising.



Nella terza area, quella evidenziata in verde, si possono settare i valori di tensione (220 V è l'impostazione predefinita), oltre che il valore del carico noto su cui si vuole effettuare la taratura. Se ad esempio come carico ci troviamo un asciugacapelli da 1.000 watt, con il carico inserito, nel Load Value scriviamo 1.000. La nostra app calcola quindi

il fattore moltiplicativo da associare alla corrente misurata al fine di far corrispondere tale valore di corrente con la potenza nota, per la particolare tensione utilizzata.

Ovviamente questo è solo un possibile esempio realizzativo: i sorgenti forniti potranno ulteriormente essere personalizzati ed elaborati per svol-

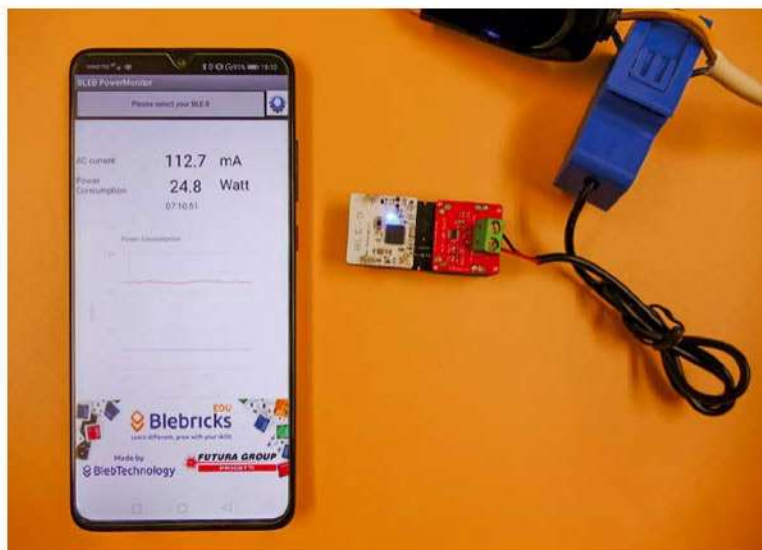
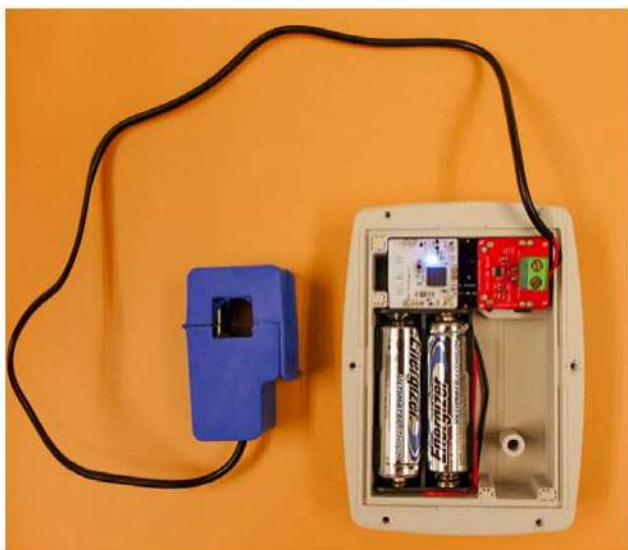


Fig. 9
Il sistema cablato e inserito in un contenitore in plastica.

Fig. 10
Test con l'app BLEB Power Monitor.

gere funzioni quali ad esempio, generare allarmi di varia natura, memorizzare file di log sul telefono o sul BLE-B, trasmettere dati a broker MQTT per visualizzare i dati online, per esempio utilizzando Node-Red. Nella **Fig. 7** proponiamo il codice completo dell'applicazione scritta in MIT App Inventor, del quale la **Fig. 8** dettaglia il blocco di decodifica dei dati di advertising Bluetooth.

I sorgenti della App mostrati in entrambe le figure (quelli relativi a MIT App Inventor) sono resi disponibili nella sezione Download del nostro sito Internet www.elettronica.in.it, insieme agli altri file di questo numero della rivista.

INSTALLAZIONE DEL SISTEMA

L'hardware, dopo essere stato assemblato, deve venire alloggiato ed installato in un contenitore in plastica, preferibilmente ignifuga, che può essere anche semplicemente una scatola di derivazione per impianti elettrici o un piccolo quadro elettrico a 2 moduli DIN da esterno o da incasso.

Al mattoncino RMS bisogna collegare, utilizzando l'apposita morsettiera bipolare, i due fili provenienti dal cavetto del trasformatore di corrente, senza curarsi della polarità, giacché il sensore funziona in corrente alternata e quel che ci interessa sono i valori di tensione indotta presenti tra i due fili, non la loro polarità. Un esempio di assemblaggio è quello proposto nella **Fig. 9**, dove vedete un portatile che fa da batteria al modulo di alimentazione in caso di mancanza dell'alimentazione ricavabile dalla rete. Per testare il sistema, dopo aver scelto quale app utilizzare nello smartphone per interfacciarsi con esso, potete semplicemente applicare il sensore

di corrente (quello che vedete in blu nelle foto del prototipo) ad un filo di un cavo di rete che alimenta, per esempio, una lampadina a incandescenza funzionante a 220Vca.

La **Fig. 10** propone un esempio di test eseguito con abbinamento all'app BLEB Power Monitor: vedete il sensore /trasformatore di corrente applicato ad uno solo dei fili del cavo tripolare di rete. Nel posizionare il circuito nel suo alloggiamento definitivo, ricordate che non deve essere chiuso in strutture metalliche o di cemento armato, perché ostacolerebbero la ricetrasmisione radio del collegamento Bluetooth, impedendo l'interazione con lo smartphone e il corretto funzionamento del nostro sistema Energy Meter.

Una volta installato il sistema, mettetelo sotto tensione e avviate l'app sul vostro smartphone; ora non dovrete fare altro che configurare i parametri e guardare di volta in volta quando vi viene inviata una notifica.



Cosa occorre?

I moduli Blebricks utilizzati in questo progetto sono disponibili presso Futura Elettronica. La main board (cod. BLE-B) è in vendita a Euro 29,00, il modulo con INA219 (cod. BLEBRMS) a Euro 25,00, il modulo adattatore (cod. BLEBIB2) a Euro 8,00, la pinza amperometrica (cod. THM104C) a Euro 11,00. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

Entra nel mondo



cod. KSR19
€ 42,00



cod. KSR18
€ 45,00



cod. KSR21
€ 42,00

Robot con codifica meccanica 5 in 1 - in kit

Robot didattico in scatola di montaggio con codifica STEM (Science, Technology, Engineering, Mathematics) particolarmente adatto per principianti, pensato per stimolare l'utente ad avvicinarsi all'ingegneria di costruzione delle macchine con codifica manuale.

Robot TOBBIE II - in kit

Può funzionare in due differenti modalità: "Seguimi" e "Esplora". Nella modalità "Seguimi" è in grado di seguire qualsiasi cosa gli si avvicini. Quando funziona in modalità "Esplora", è in grado di evitare gli ostacoli che trova davanti a sé trovando sempre un nuovo percorso alternativo.

Robot RICCIO - in kit

Robot 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.



cod. KSR10
€ 68,00



cod. KSR13
€ 33,00

Braccio robotico

Braccio meccanico, dotato di cinque motori e cinque articolazioni con unità di controllo via cavo a cinque comandi.

Interfaccia USB
per braccio robotico

cod. KSR10/USB | € 47,00

Educational Solar Robot 14 in 1 - in kit

Kit di montaggio che consente di realizzare 14 diverse configurazioni di robot. Grazie alle istruzioni dettagliate e ricche di illustrazioni, l'assemblaggio delle parti che lo compongono, risulterà estremamente facile e divertente. Il pannello solare gli permette di funzionare senza batterie!

della robotica

cod. KSR20
€ 92,00



Robot TOBBIE II con MICRO:BIT - in kit

Robot a 6 gambe con corpo libero di ruotare, dotato di tutti i componenti elettronici necessari 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 PC portatile o smartphone.

cod. KSR14
€ 22,00



Robot SOLAR ROVER - in kit

Particolarmente indicato come gioco didattico per bambini (da 8 anni in su). Questo kit permette di assemblare un mini rover solare con 6 ruote. Dispone di un sistema di trasmissione a ingranaggi planetari, piccoli bracci robotici, un pannellino solare e un piccolo motorino.

cod. KSR17
€ 38,00



Robot solare e idraulico 12 in 1 - in kit

Permette di realizzare fino a 12 differenti modelli di robot con movimento idraulico e alimentati tramite energia solare. Il kit viene fornito completo di un manuale ricchissimo di illustrazioni.

cod. KSR12
€ 56,00



Braccio robotico idraulico

Kit di montaggio di braccio robotico idraulico a 6 assi facile e divertente. Il movimento viene azionato agendo sui controller a leva tramite il sistema idraulico ad acqua.

cod. KSR16
€ 28,00



Kit per auto con motore ad aria

Kit di montaggio per auto con motore alimentato ad aria compressa. È composto da 102 parti di montaggio estremamente facili da assemblare tra di loro. La macchinina è in grado di percorrere fino 50 metri di distanza.

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

www.futurashop.it

Ora i componenti non hanno più segreti!

DISPOSITIVI PROFESSIONALI PER MISURE E ANALISI PRECISE

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



COD. DSO138MINI
€ 68,00

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

**TESTER PER
COMPONENTI
ELETTRONICI**



COD. JT-LCRT7
€ 34,00

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

Prezzi IVA inclusa.

**Analizzatore di
componenti DCA55**



€ 73,00
cod. DCA55

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

**Analizzatore di
semiconduttori DCA75**



€ 158,00
cod. DCA75

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

**Misuratore di induttanza,
capacità, resistenza
e impedenza LCR45**



€ 129,00
cod. LCR45

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

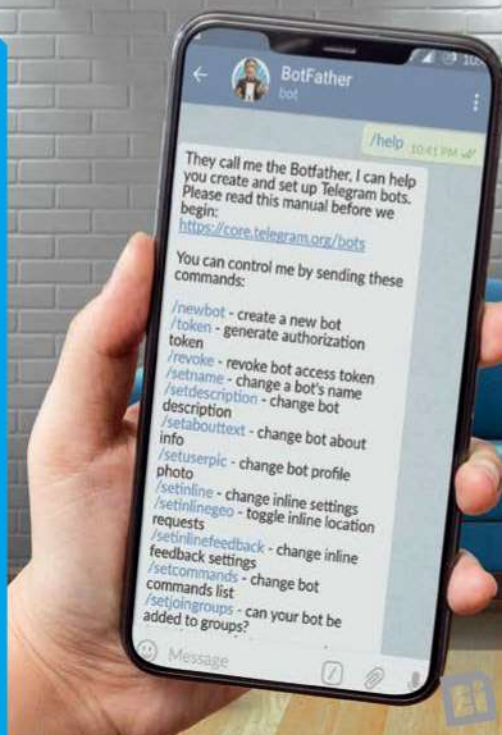
COMANDO LUCE CON TELEGRAM

di DANIEL SCAINI

Controllo da smartphone di accensione e spegnimento di una lampada o di un carico elettrico, con timer di spegnimento automatico programmabile.

Il tema forse più discusso in questi mesi è sicuramente quello della crisi energetica, la quale ci obbliga a un po' più di attenzione ai consumi per farvi fronte. Ci troviamo a dover fare i conti non solo con l'aumento esorbitante dei costi di elettricità e gas, ma anche al rischio che l'energia, cui eravamo abituati ad attingere senza preoc-

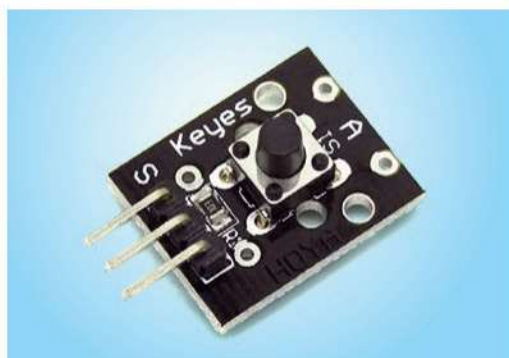
cuparci di quanta ne consumassimo, ci venga razionata. Una soluzione per proteggersi è ricorrere a dispositivi in grado di consumare sempre meno oppure dotati di un'intelligenza particolare che tende a preservare proprio tale aspetto; basti pensare alla funzione timer del televisore, che se attivata, andrà a spegnere l'apparecchio dopo un tempo preimpostato. In quest'ottica nasce il progetto proposto in



→ **Fig. 1**
Il modulo
nodeMCU.



→ **Fig. 2**
La breakout board
pulsante.



questo articolo, consistente nel creare, con un nodeMCU, un relé intelligente in grado di ricevere comandi remoti ed essere in grado di spegnere tutto se viene superata una certa soglia temporale impostabile.

Il nodeMCU sarà connesso alla nostra rete WiFi di casa. Invieremo e comanderemo il nostro dispositivo attraverso uno dei servizi di messaggistica più conosciuto ed utilizzato: Telegram; infatti attraverso una chat colloquieremo direttamente con il nostro relé, impartendo comandi di ON, OFF oppure SET, per impostare un tempo soglia oltre il quale l'utilizzatore verrà spento.

Per realizzare questo progetto dovremo approntare un hardware basato sul nodeMCU e scriveremo un firmware per programmare quest'ultimo in modo che consenta di fare ciò che abbiamo appena spiegato.

IL LINGUAGGIO DI PROGRAMMAZIONE

Sulla nostra piattaforma, il nodeMCU con ESP8266, è possibile sviluppare sia in Python che in C++ o comunque un suo "dialetto".

Il primo è un linguaggio molto conosciuto, corredato da una vastissima gamma di librerie "ready to use" e utilizzato oramai da circa l'84% degli sviluppatori ormai paragonabile al Java o Perl.

Nonostante quanto appena descritto utilizzeremo il secondo, rimanendo su quello che inizialmente era stato il linguaggio di base per sviluppare su piattaforme come Arduino o similari. Per supportare lo sviluppo con questo linguaggio c'è a disposizione Arduino IDE: un Integrated Development Environment, ovvero un ambiente dedicato, che mette a disposizione diversi tool per poter scaricare sul device i nostri progetti e diverse librerie per supportarne la creazione.

L'HARDWARE

Come accennato, l'hardware su cui ci basiamo è incentrato sul chip ESP8266; nel dettaglio, utilizzeremo la demo board Nodemcu V3 sulla quale è montato il chip ESP8266 della Espressif System, la quale deve la sua fortuna e la sua rapida espansione all'introduzione e al porting all'interno del suo core della libreria MQTT di cui esporremo più avanti. La board è reperibile sul sito www.futurashop.it con il codice prodotto NODEMCUESP.

Questa piattaforma, grazie alle sue ridottissime dimensioni, è l'ideale per la fase di prototipazione di applicazioni IoT (Internet of Things) perchè al suo interno vi è già un modulo WiFi.

Il modulo WiFi è compatibile con lo standard 802.11 b/g/n a 2,4 GHz, dispone di stack TCP/IP integrato, potenza di uscita di 19.5 dBm, interfaccia dati (UART / HSPI / I²C / I²S / IR Remote Control GPIO / PWM) e antenna su PCB.

Questa scheda (**Fig. 1**) è equipaggiata con un microcontrollore Tensilica 32-bit RISC CPU Xtensa LX106, con un clock di 80 MHz, al quale si affiancano 4MB di Flash memory e 64K di SRAM.

A differenza di una classica Arduino, non possiede una vera e propria EEPROM, ma questa viene simulata alla perfezione sfruttando la Flash, il che rende il modulo in grado di memorizzare alcune impostazioni del programma o dati esterni.

La board lavora ad una tensione di 3,3V ma in ingresso accetta una tensione che può variare da 3 a 12V. Mette a disposizione, oltre che ad una microUSB per poterlo programmare agevolmente, diversi pin digitali (DIO), un pin analogico (ADC), un pin per la SPI ed un pin per I²C-Bus.

Oltre al device principale appena descritto dovremo dotarci anche di un key switch module (**Fig. 2**), ovvero un modulo composto da un pulsante semplice a pressione e spesso una resistenza, tipicamente da 10 kohm.

Alla pressione dell'interruttore il circuito si chiude, mentre al rilascio si riapre. Data la funzione che svolge, all'occorrenza questa breakout board può

essere sostituita da un qualsiasi interruttore normalmente aperto.

Se vogliamo lavorare con una lampadina o comunque un carico più importante del LED prototipale, usato come esempio, sarà necessario anche un modulo con a bordo un relé (**Fig. 3**) come il RELAY1CH della Futura Elettronica (www.futurashop.it).

Le connessioni fra i componenti del sistema sono essenziali, come si vede dallo schema in **Fig. 4** oppure, nel caso in cui vogliamo lavorare con carichi più importanti, quelle riportate nella **Fig. 5**, che comprendono il modulo a relé.

UTILIZZIAMO TELEGRAM

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

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

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

Un altro fattore importante da considerare nella vicenda è la sicurezza di WhatsApp, più volte finito sotto attacco dai suoi detrattori, i quali spingono



Fig. 3
La breakout board a relé.

molto sulle polemiche legate alla composizione dell'impero Zuckerberg (WhatsApp, Instagram, Facebook).

Una delle peculiarità più interessanti del servizio di messaggistica Telegram è costituita dai gruppi, che possono raggiungere dimensioni enormi, anche di oltre 100.000 membri.

Inoltre Telegram ci dà la possibilità di creare dei veri e propri canali di trasmissione che possono avere un numero illimitato di utenti e possono essere sia privati che pubblici. Tra quelli privati ci sono i bot ed è proprio quest'ultima caratteristica che possiamo sfruttare per creare un filo diretto fra noi e la nostra scheda NodeMCU. Un bot è un "utente" di Telegram non umano, fatto da righe di codice e capace di svolgere azioni più o meno complesse in modo automatico, ricevendo i comandi dell'utente sotto forma di richieste HTTP.

Esistono due tipologie di bot Telegram:

- bot stand-alone, dotato di una finestra di chat autonoma in cui l'utente può impartire i comandi; per raggiungere un bot stand alone basta aggiungerlo attraverso un link;

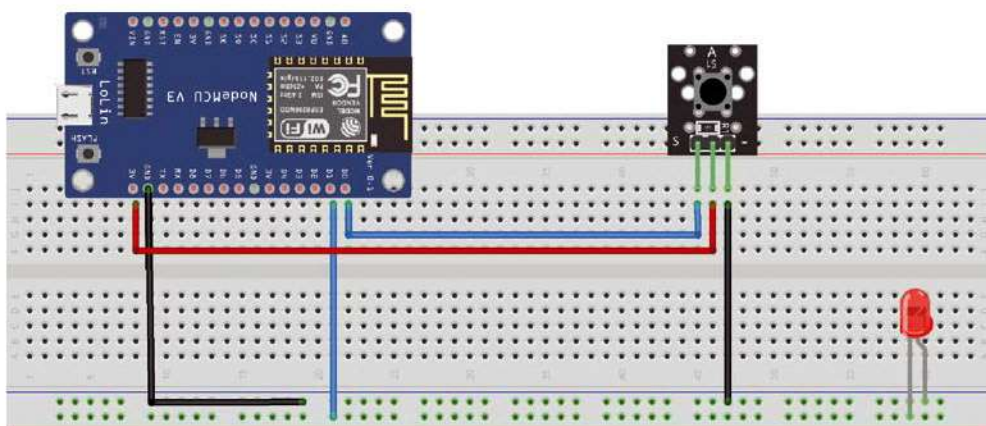
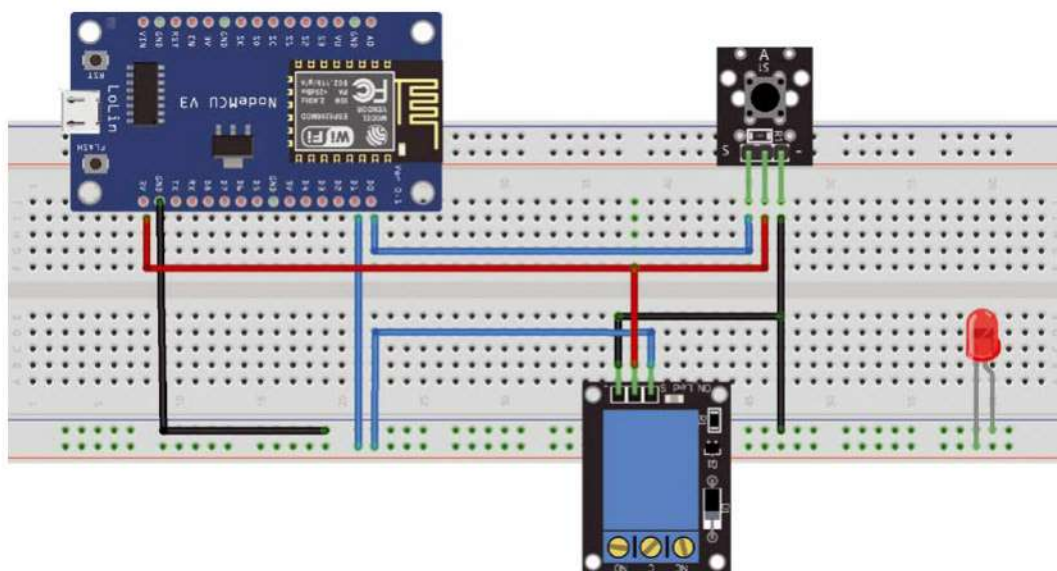


Fig. 4
Schema di cablaggio dell'hardware base.

fritzing

Fig. 5
Schema di
cablaggio con il
modulo relé.



- bot inline, che può essere richiamato da altre finestre di chat (singole o di gruppi) tramite il comando @nomebot.

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

Vediamo dunque come creare il nostro bot: apriamo l'app e clicchiamo sulla lente di ricerca nella barra superiore, dove digitiamo BotFather, allorché ci appare quanto mostrato nella **Fig. 6**. Clicchiamo su questa e apriamo la chat nella quale andremo a scrivere questo messaggio: /newbot, e seguiamo le istruzioni a video. Dopo questa procedura copiamo il token rilasciato da BotFather. Questa è una delle due variabili, SECRET_BOT_TOKEN, che ci servirà per comunicare con le API di Telegram. La seconda invece la possiamo ottenere in diversi modi. Consigliamo di mandare un messaggio con scritto "Ciao" all'interno della chat appena creata e inserire all'interno di un browser questo link: <https://api.telegram.org/bot<BOTToken>/getUpdates>. In esso <BOTToken> è il token del passo precedente. Gli step dovranno esser eseguiti esattamente in questo ordine. In risposta, sul browser apparirà un json, ovvero una struttura dati formattata in un determinato modo, simile al seguente:

```
{
  "ok": true,
  "result": [
```

```
{
  "update_id": YYYYYY,
  "message": {
    "message_id": 18,
    "from": {
      "id": XXXXXXXX,
      "is_bot": false,
      "first_name": "NOME COMPLETO TELEGRAM",
      "username": "IL TUO USERNAME TELEGRAM",
      "language_code": "it"
    },
    "chat": {
      "id": XXXXXXXX,
      "first_name": "NOME COMPLETO TELEGRAM",
      "username": "IL TUO USERNAME TELEGRAM",
      "type": "private"
    },
    "date": 1600324371,
    "text": "Ciao"
  }
}
```

Questa struttura rappresenta le interazioni con il bot da parte di utenti.

Ovviamente noi abbiamo interagito con lui precedentemente con il messaggio "Ciao". L'ultima variabile, quella chiamata SECRET_CHAT_ID, si trova proprio all'interno di questo json nel campo in grassetto.

IL FIRMWARE

Prima di procedere alla programmazione dovremo impostare ArduinoIDE, ovvero il nostro ambiente di sviluppo, per poter scrivere correttamente sul modulo. Apriamo quindi ArduinoIDE e andiamo in File->Impostazioni e aggiungiamo negli URL aggiuntivi per i gestori delle schede questo link: http://arduino.esp8266.com/stable/package_esp8266com_index.json; facciamo quindi clic sul pulsante OK.

Se disponiamo già di uno o più URL, si usa la virgola per concatenarli.

Ora scarichiamo il gestore della nostra scheda NodeMCU: rechiamoci in *Strumenti->Scheda: XXXX->Gestore schede..*, cerchiamo "8266" e installiamo l'unico risultato che la ricerca proporrà.

Uscendo, selezioniamo nuovamente *Strumenti->Scheda: XXXX* e noteremo una nuova voce ESP8266 Modules, la quale al suo interno ha diverse altre scelte; da qui selezioniamo NodeMcu 1.0 (ESP-12E Module).

Ora andiamo a caricare la libreria per la decodifica dei JSON. Il JSON, Javascript Object Notation, è un formato testuale per la strutturazione di dati che, una volta compresso, ti permetterà di padroneggiare le informazioni in qualsiasi ambito dell'informatica di oggi. È un formato che deriva dal web, dove si è avuta l'espansione più importante negli ultimi anni. Questo ci permette di continuare a lavorare ad oggetti anche durante lo scambio di dati con altre entità.

Per caricare la libreria all'interno dell'IDE sarà sufficiente andare in "Sketch", "#include libreria" e infine "Gestione librerie"; proprio come indicato nella Fig. 7.

Nella schermata successiva clicchiamo nella barra di ricerca e digitiamo "ArduinoJson" e installiamo il primo risultato.

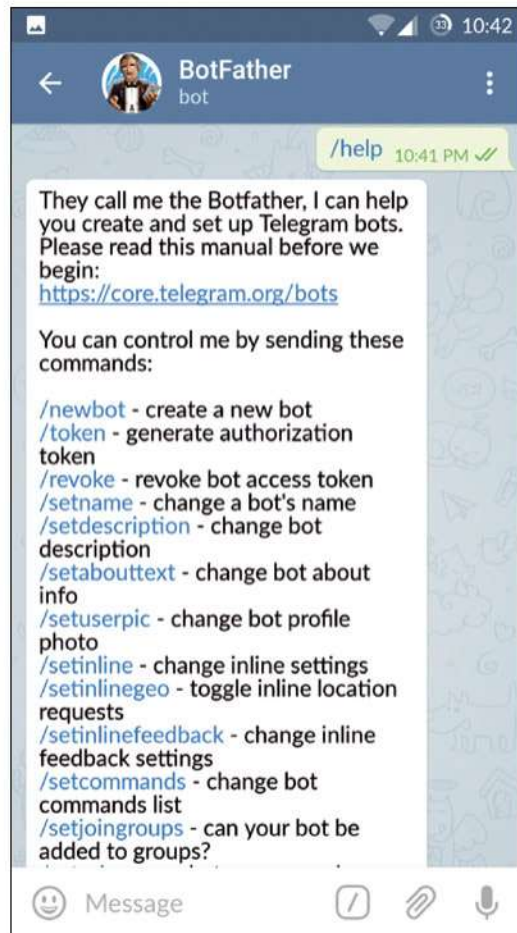


Fig. 6
Schermata di Botfather.

Di questa libreria è importante installare una versione superiore alla 6 altrimenti sarà necessario cambiare alcune parti del codice. Cominciamo ora a scrivere il firmware del nostro device principale. In generale possiamo scriverlo in due modi: procedurale oppure ad oggetti.

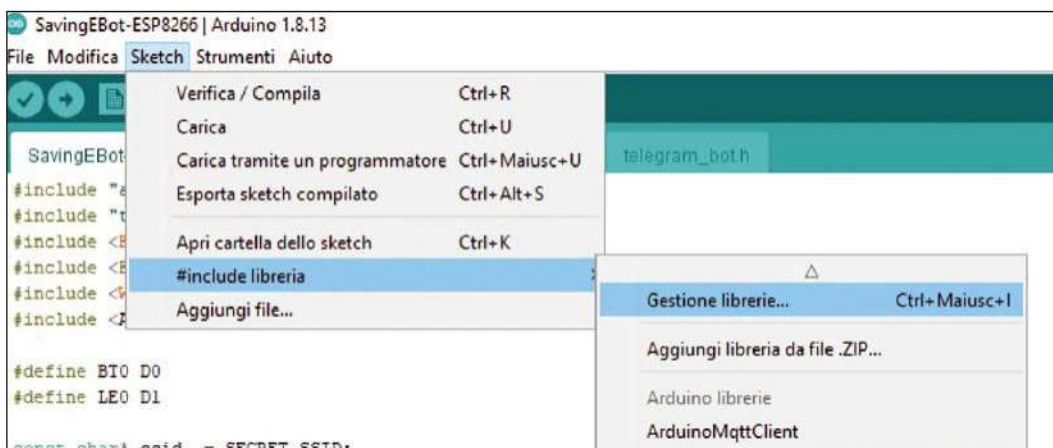
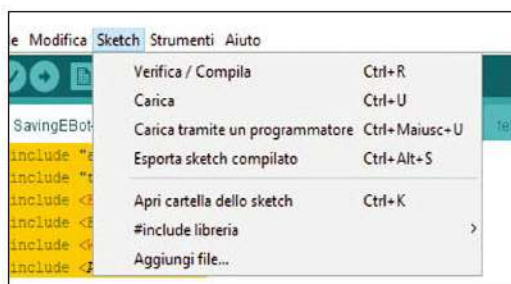


Fig. 7
La libreria.

Fig. 8
Importazione
file.



Nel primo caso si ha una stesura unica, su un unico file. Spesso questo approccio è caratteristico di programmi semplici, ma viene adottato anche nel caso in cui si faccia una prima stesura veloce, prima di capire gli elementi in gioco. Noi utilizzeremo il secondo, il quale presenta numerosi vantaggi, tra cui la facilità di lettura e modifica, anche da parte di chi non ha mai visto il codice e la modularità, che rende riutilizzabili i vari moduli anche all'interno di altri progetti. Nel foglio nuovo all'interno di Arduino IDE, inseriamo quindi i primi import:

```
#include "arduino_secrets.h"
#include "telegram_bot.h"
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <ArduinoJson.h>
```

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

Come possiamo notare, le ultime quattro sono le librerie cui abbiamo già accennato. I primi due invece sono due file .h, header, i quali, in ordine, conterranno SSID, password e TokenTelegram e le dichiarazioni dei metodi e delle variabili della nostra classe responsabile della comunicazione con il servizio di chat.

Il primo file è semplice e contiene cinque parametri: SECRET_SSID, SECRET_PASS, SECRET_BOT_TOKEN, SECRET_CHAT_ID, SECRET_TIME_REFRESH. Questi permettono di cambiare rete e configurazione senza dover necessariamente modificare il .ino.

Ovviamente oltre al file .h, per quanto riguarda la nostra classe ci sarà anche un .cpp, che contiene l'implementazione delle varie funzioni, il vero core dell'oggetto. Creiamo dunque questi file vuoti: *arduino_secrets.h*, *telegram_bot.h* ed infine *telegram_bot.cpp*; per semplicità li collocheremo sul Desktop.

A questo punto importiamo i tre file all'interno del progetto Arduino: nella barra degli strumenti dell'IDE clicchiamo su Sketch e poi su Importa file (**Fig. 8**).

Selezioniamo un file alla volta e dovremmo vederli come proposto nella **Fig. 9**.

Dopo queste prime configurazioni iniziali, possiamo dunque iniziare a scrivere il nostro codice. All'interno del primo dei file importati, *arduino_secrets.h* copiamo ed incolliamo la sezione di codice proposta qui di seguito.

```
#define SECRET_SSID "Wifi_ssid"
#define SECRET_PASS "Wifi_password"
#define SECRET_BOT_TOKEN "TelegramBOT_Token"
#define SECRET_CHAT_ID "TelegramBOT_ChatID"
#define SECRET_TIME_REFRESH 10
```

Come primo passo, invece, all'interno del file principale dovremo istanziare le solite due macro funzioni necessarie nel ciclo di vita di Arduino, e le costanti:

```
#include "arduino_secrets.h"
#include "telegram_bot.h"
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <ArduinoJson.h>

const char* ssid = SECRET_SSID;
const char* password = SECRET_PASS;
String botToken = SECRET_BOT_TOKEN;
String CHAT_ID = SECRET_CHAT_ID;
const int TIME_REFRESH = SECRET_TIME_REFRESH;
long unsigned startTime = 0;
long unsigned settedTime = 0;
void setup() {
}
void loop() {
}
```

Fig. 9
I file
importati.



Così facendo ora abbiamo dato i valori presenti all'interno del file .h alle nostre costanti. Concentriamoci ora sulla funzione di setup. Come prima cosa, oltre naturalmente a istanziare la Serial per permetterci di debuggare, dobbiamo collegare il nostro device alla nostra rete Wifi. Utilizzeremo quindi alcuni metodi della Wifi-Client:

```
Serial.begin(115200);
while (!Serial);
Serial.print("Connecting Wifi: ");
Wifi.begin(ssid, password);
while (Wifi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("Wifi connected");
```

Se proviamo a compilare e scaricare il progetto all'interno del dispositivo in questo istante e apriamo il Serial monitor presente nell'IDE noteremo una serie di "." finché non ha stabilito la connessione. Solitamente, se la rete non ha particolari limitazioni o particolarità, il tempo impiegato è davvero poco. Introduciamo ora il LED e il modulo switch contenente il pulsante; allo scopo dichiariamo, attraverso due define in alto, appena sotto le costanti, i due pin responsabili della loro gestione:

```
#define BT0 D0
#define LE0 D1
```

Fatto ciò, inizializziamoli all'interno del setup:

```
pinMode (BT0, INPUT);
pinMode (LE0, OUTPUT);
```

Andiamo quindi a gestire all'interno dell'altra macro funzione vitale, il loop, il comportamento alla pressione del pulsante; lo facciamo con questa porzione di codice:

```
if(digitalRead(BT0)==0){
    Serial.println("Click");
    if(startTime==0){
        turnON();
    }else{
        turnOFF();
    }
}
```

Come si può notare da questo stralcio di codice, è stata aggiunta anche un'altra variabile che ci dirà se è acceso o spento il LED, in modo da dare correttamente il comando di ON o di OFF. In realtà oltre a questo ci darà un feedback sul tempo di attività in secondi. Nel dettaglio le due funzioni saranno così:

```
void turnON(){
    if(startTime==0){
        startTime = millis()/1000;
    }
    digitalWrite(LE0, HIGH);
}

void turnOFF(){
    startTime = 0;
    digitalWrite(LE0, LOW);
}
```

Passiamo ora invece ad interfacciarci con il servizio di messaggistica.

Noi gestiremo tre tipi di messaggio: ON, per accendere il dispositivo, così come il click del pulsante, OFF, per poter disattivare ed infine set seguito da un numero di secondi per impostare il limite di tempo in cui il dispositivo può rimanere nello stato di ON.

Per fare ciò cominciamo a scrivere la classe partendo dal file .h, inserendo la dichiarazione della classe:

```
#define telegram_bot_h

#if ARDUINO >= 100
#include "Arduino.h"
#else
#include "WProgram.h"
#endif
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <ArduinoJson.h>
//EXTRA

#define JSON_SIZE 1024
#define DEBUG 0

class telegram_bot{
public:

private:
};
```



↓ Listato 1

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

#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <ArduinoJson.h>
//EXTRA

#define JSON_SIZE 1024
#define DEBUG 0

class telegram_bot{
public:
    telegram_bot();
    telegram_bot(String tk, String chat_id);
    void getMessages();
    void sendMessage(String data);
    int checkNow(long unsigned startTime,
        long unsigned settedTime);
    long unsigned getSettedTime();
    int getType();
    int TURN_ON = 1;
    int TURN_OFF = 2;
    int SET = 3;
    int TOO_MUCH_STATE = 1;

private:
    void debug(String data);
    long unsigned offsetMessages;
    String botToken;
    String chatID;
    long unsigned settedTime = 0;
    StaticJsonDocument<JSON_SIZE> doc;
    String PATH_TELEGRAM = "https://api.telegram.org/bot";
};
```

Questa avrà, come suggerito dal codice una parte pubblica, metodi e variabili richiamabili dall'esterno ed una parte privata, spendibile solo all'interno della classe. Nella parte pubblica andiamo ad inserire il costruttore così:

```
telegram_bot();
telegram_bot(String tk, String chat_id);
```

Questa funzione è necessaria in una classe ed è importantissima, perché costituisce quel metodo di una classe il cui compito è proprio quello di creare nuove istanze, oltre ad essere il punto del programma in cui un nuovo elemento (quindi una nuova identità) viene creato ed è reso disponibile per l'interazione con il resto del sistema. Per impostazione predefinita viene sempre richiesto un costruttore che non accetti parametri in ingresso; nel nostro caso abbiamo fatto

un override per potergli passare i due parametri fondamentali quali *token* e *chatid*.

Dichiariamo ora le funzioni. In ordine, la prima si occupa di recuperare i nuovi messaggi, la seconda dell'invio di questi ad una chat specifica ed infine la terza valuta se è stata verificata la condizione settata:

```
void getMessages();
void sendMessage(String data);
int checkNow(long unsigned startTime,
    long unsigned settedTime);
```

Poi altre due per recuperare alcune variabili private:

```
long unsigned getSettedTime();
int getType();
```

Ed infine le costanti. Anch'esse come public in modo tale da poterle usare anche fuori dalla classe.

```
int TURN_ON = 1;
int TURN_OFF = 2;
int SET = 3;
int TOO_MUCH_STATE = 1;
```

Un esempio dell'intero file è possibile trovarlo all'interno del **Listato 1**.

Passiamo all'altro file, il .cpp, nel quale troviamo le implementazioni vere e proprie delle funzioni. Tralasciando la parte iniziale di include che è molto simile alla precedente, troviamo i costruttori. Tutti i metodi dovranno essere accompagnati dal prefisso della classe, ovvero:

```
tipo classe::metodo(argomenti)
```

I nostri costruttori sono quindi:

```
telegram_bot::telegram_bot(){
    offsetMessages = 0;
    botToken = "";
    chatID = "";
}
telegram_bot::telegram_bot(String tk, String chat_id){
    offsetMessages = 0;
    botToken = tk;
    chatID = chat_id;
}
```

La funzione responsabile della ricezione di nuovi

messaggi è una GET, alla quale inseriremo una logica per poter discriminare i messaggi vecchi da quelli nuovi. Tale logica è basata su un argomento passato alla richiesta REST detto offset.

Attraverso questo valore è possibile andare a ricevere solo i messaggi nuovi tralasciando quelli già letti (**Listato 2**).

È importante instaurare un client Secure perché la chiamata è effettuata in https. Inoltre nella parte finale inseriamo il controllo nel caso ci trovassimo al primo giro, quindi con un offset = 0.

Il metodo per la lettura dei messaggi invece è un pò più complicato.

Infatti anche qui c'è una richiesta web, ma questa volta si tratta di una POST, quindi dovremmo inserire anche un header nella chiamata per poterla rendere efficace.

Un esempio completo è presente all'interno del **Listato 3**.

Torniamo ora al nostro codice principale, quello nel .ino, che avevamo abbandonato all'inizio dell'implementazione per creare e popolare i vari file .h e .cpp.

Qui inseriamo la classe appena creata.

Come prima cosa è necessario dichiararla, magari in modo globale, in questo modo:

```
long unsigned startTime = 0;
long unsigned settedTime = 0;
telegram_bot myBot;
void setup() {
```

Poi dobbiamo inizializzarla, andando a richiamare il costruttore già ampiamente descritto; le righe di codice corrispondenti sono:

```
Serial.println("WiFi connected");
myBot = telegram_bot(botToken, CHAT_ID);
}
void turnON(){
```

La nostra variabile myBot contiene tutto, i metodi per andare a recuperare, inviare e capire in che stato sono i nostri messaggi verso il servizio di Telegram.

Nella parte loop, dove si svolgono già i controlli per verificare se sto premendo il tasto per accenderla o spegnerla, andremo ad aggiungere la lettura e il

Listato 2

```
WiFiClientSecure client;
client.setInsecure();

HTTPClient https;
String offSet = "";
if(offsetMessages>0){
    offSet = "?offset=";
    offSet.concat(offsetMessages);
}
String fullUrl = PATH_TELEGRAM+botToken+"/getUpdates"+offSet;
debug("Requesting " + fullUrl);
if (https.begin(client, fullUrl)) {
    int httpCode = https.GET();
    debug("===== Response code: " + String(httpCode));
    if (httpCode > 0) {
        debug(https.getString());
        deserializeJson(doc, https.getString());
        if(doc["result"][0]["update_id"]>0){
            offsetMessages = doc["result"][0]["update_id"];
            offsetMessages = offsetMessages + 1;
        }
    }
    https.end();
} else {
    debug("[HTTPS] Unable to connect\n");
}
if(offsetMessages==0){
    offsetMessages = doc["result"][0]["update_id"];
    offsetMessages = offsetMessages + 1;
}
```

▼ Listato 3

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

#include "telegram_bot.h"
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <ArduinoJson.h>

telegram_bot::telegram_bot(){
    offsetMessages = 0;
    botToken = "";
    chatID = "";
}

telegram_bot::telegram_bot(String tk, String chat_id){
    offsetMessages = 0;
    botToken = tk;
    chatID = chat_id;
}

void telegram_bot::debug(String data){
    Serial.println(data);
}

void telegram_bot::getMessages(){
    WiFiClientSecure client;
    client.setInsecure();

    HTTPClient https;
    String offSet = "";
    if(offsetMessages>0){
        offSet = "?offset=";
        offSet.concat(offsetMessages);
    }
    String fullUrl = PATH_TELEGRAM+botToken+"/getUpdates"+offSet;
    debug("Requesting " + fullUrl);
    if (https.begin(client, fullUrl)) {
        int httpCode = https.GET();
        debug("===== Response code: " + String(httpCode));
        if (httpCode > 0) {
            debug(https.getString());
            deserializeJson(doc, https.getString());
            if(doc["result"][0]["update_id"]>0){
                offsetMessages = doc["result"][0]["update_id"];
                offsetMessages = offsetMessages +1;
            }
        }
        https.end();
    } else {
        debug("[HTTPS] Unable to connect\n");
    }
    if(offsetMessages==0){
        offsetMessages = doc["result"][0]["update_id"];
        offsetMessages = offsetMessages +1;
    }
}

void telegram_bot::sendMessage(String data){
    WiFiClientSecure client;
    client.setInsecure();

    HTTPClient https;
    String fullUrl = PATH_TELEGRAM+botToken+"/sendMessage";
    debug("Requesting " + fullUrl);
    if (https.begin(client, fullUrl)) {
        https.addHeader("Content-Type", "application/x-www-form-urlencoded");
        data.replace(" ", "%20");
    }
```



```

String httpRequestData = "text="+data+"&chat_id="+chatID;
int httpCode = https.POST(httpRequestData);
debug("===== Response code: " + String(httpCode));
if (httpCode > 0) {
    debug(https.getString());
    debug("Sent");
}
https.end();
} else {
    debug("[HTTPS] Unable to connect\n");
}
}

int telegram_bot::checkNow(long unsigned startTime, long unsigned settedTime){
    if(settedTime > 0 && startTime > 0 && ((millis()/1000)-startTime) >= settedTime){
        debug("Too much time");
        String sendMsg = "Superata la durata massima impostata (" +
            sendMsg.concat(settedTime);
        sendMessage(sendMsg+"s). Spengo il dispositivo.");
        return TOO_MUCH_STATE;
    }
    return 0;
}

String getValue(String data, char separator, int index)
{
    int found = 0;
    int strIndex[] = {0, -1};
    int maxIndex = data.length()-1;

    for(int i=0; i<=maxIndex && found<=index; i++){
        if(data.charAt(i)==separator || i==maxIndex){
            found++;
            strIndex[0] = strIndex[1]+1;
            strIndex[1] = (i == maxIndex) ? i+1 : i;
        }
    }

    return found>index ? data.substring(strIndex[0], strIndex[1]) : "";
}

long unsigned telegram_bot::getSettedTime(){
    return settedTime;
}

int telegram_bot::getType(){
    if(sizeof(doc["result"])>0){
        String text = doc["result"][0]["message"]["text"];
        if(text!=NULL){
            debug(text);
            if(text=="on"){
                if(DEBUG==1){
                    debug("Turn on");
                }
                return TURN_ON;
            }else if(text=="off"){
                debug("Turn off");
                return TURN_OFF;
            }else if(text.startsWith("set")){
                settedTime = getValue(text, ' ', 1).toInt();
                return SET;
            }
        }
    }
    return 0;
}

```

controllo dei messaggi in questo modo:

```

void loop() {
    myBot.getMessages();

```

```

int tp = myBot.getType();
if(tp==myBot.TURN_ON){
    turnON();
}else if(tp==myBot.TURN_OFF){

```



↓ Listato 4

```
#include "arduino_secrets.h"
#include "telegram_bot.h"
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <ArduinoJson.h>

#define BT0 D0
#define LE0 D1

const char* ssid = SECRET_SSID;
const char* password = SECRET_PASS;
String botToken = SECRET_BOT_TOKEN;
String CHAT_ID = SECRET_CHAT_ID;
const int TIME_REFRESH = SECRET_TIMEREFRESH;

long unsigned startTime = 0;
long unsigned settedTime = 0;

telegram_bot myBot;

void setup() {
    Serial.begin(115200);
    while (!Serial);

    pinMode (BT0, INPUT);
    pinMode (LE0, OUTPUT);
    Serial.print("Connecting Wifi: ");
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");

    myBot = telegram_bot(botToken, CHAT_ID);
}

void turnON(){
    if(startTime==0){
        startTime = millis()/1000;
    }
    digitalWrite(LE0, HIGH);
}

void turnOFF(){
    startTime = 0;
    digitalWrite(LE0, LOW);
}

void interrControl(){
    Serial.println("Inter");
}

void loop() {
    myBot.getMessages();
    int tp = myBot.getType();
    if(tp==myBot.TURN_ON){
        turnON();
    }else if(tp==myBot.TURN_OFF){
        turnOFF();
    }else if(tp==myBot.SET){
        settedTime = myBot.getSettedTime();
    }
    if(digitalRead(BT0)==0){
        Serial.println("Click");
        if(startTime==0){
            turnON();
        }else{
            turnOFF();
        }
    }
    int _toom = myBot.checkNow(startTime, settedTime);
    if(_toom==myBot.TOO_MUCH_STATE){
        turnOFF();
    }
    delay(TIME_REFRESH);
}
```

```
turnOFF();
} else if(tp==myBot.SET){
    settedTime = myBot.getSettedTime();
}
if(digitalRead(BT0)==0){
```

Come specificato all'interno dell'introduzione, è fondamentale impostare anche un controllo per poter permettere al dispositivo di potersi settare in OFF se sfioriamo la soglia di tempo che abbiamo impostato in precedenza.

Questo controllo, che abbiamo già creato nella classe, è da inserire alla fine della funzione di loop, dopo le varie attivazioni/disattivazioni.

```
int _toom = myBot.checkNow(startTime, settedTime);
if(_toom==myBot.TOO_MUCH_STATE){
    turnOFF();
}
```

Tutto il codice di questo file è all'interno del **Listato 4**.

CONCLUSIONI

Dal semplice prototipo si possono creare diverse applicazioni o evoluzioni, basti pensare al protocollo di comunicazione: pensiamo che Telegram non faccia al caso nostro, possiamo rimpiazzarlo con una tecnologia come MQTT semplicemente agendo sulla classe telegram_bot o magari creando una classe parallela, ma senza modificare troppo la struttura alla base del progetto. 



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. Il modulo nodeMCU con ESP8266 (cod. NODEMCUESP) è in vendita a Euro 9,50, il relè (cod. RELAY1CH) è disponibile a Euro 5,00. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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



IMPARA L'ELETTRONICA FACILMENTE CON TOTEM MINI LAB



€ 112,00
cod. TOTEMMINILAB

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

COMPLETALO CON LE SCHEDE DI ESPANSIONE!



ESPANSIONE AUDIO

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

€ 45,00
Cod. TOTEMAUDIOSIDE



ESPANSIONE I/O

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

€ 37,00
Cod. TOTEMIOSIDE



ESPANSIONE SENSORI

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

€ 39,50
Cod. TOTEMSENSORSIDE

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.

Maggiori informazioni e richiesta di preventivi su design.futuranet.it

CALCOLATRICE CON TASTIERA ANALOGICA

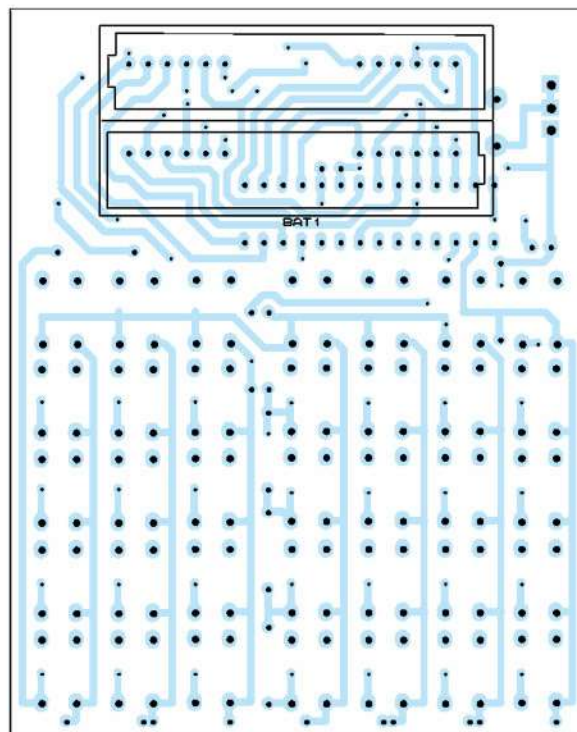
di STEFANO PURCHIARONI



Riscopriamo la classica calcolatrice con display a 7 segmenti dall'aspetto vintage. Seconda e ultima puntata.

N

ella puntata introduttiva di questo articolo vi abbiamo presentato il progetto di una calcolatrice da tavolo, fatta come quelle di una volta e quindi con tasti fisici e display a diodi luminosi, caratterizzata dalla particolarità di avere la tastiera non organizzata a matrice, ma di tipo analogico; per analogico intendiamo che la pressione dei tasti (come si vede nello schema elettrico proposto nella prima puntata) viene rilevata come tensione prodotta da partitori resistivi multipli, di cui ogni resistore è in parallelo a un pulsante e viene cortocircuitato da esso quando lo si preme. Bene, il mese scorso ci siamo lasciati dopo aver spiegato la tecnica per la lettura di tastiere e le relative implicazioni.



Elenco Componenti:

RC1: 820 ohm 1%
RC2, RC3: 2,2 kohm 1%
RC4: 4,9 kohm 1%

RC5: 4,7 kohm 1%
RK1, RK2, RL13: 15 kohm 1%
RL11, RL12, RL21, RL22: 1,5 kohm 1%

SW111+SW115, SW121+SW125,
SW131+SW135, SW141+SW145,
SW211+SW215, SW221+SW225,

Come abbiamo visto la calcolatrice è basata su un comune microcontrollore Microchip dotato di A/D converter, la cui funzione è leggere i tasti della tastiera, eseguire i calcoli corrispondenti e visualizzare gli operandi e i risultati pilotando in multiplex un display ad 8 cifre ottenuto affiancando due moduli display a LED 7 segmenti da 4 digit ciascuno. Ora è il momento di riprendere il discorso da dove lo abbiamo lasciato.

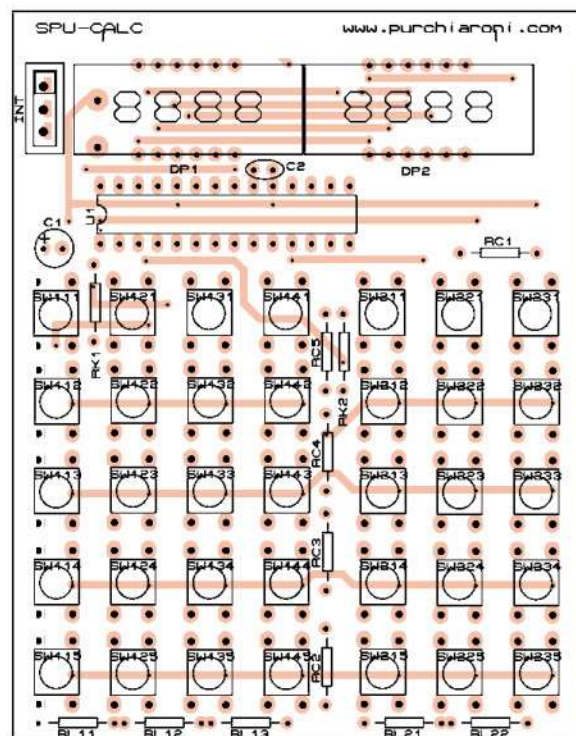
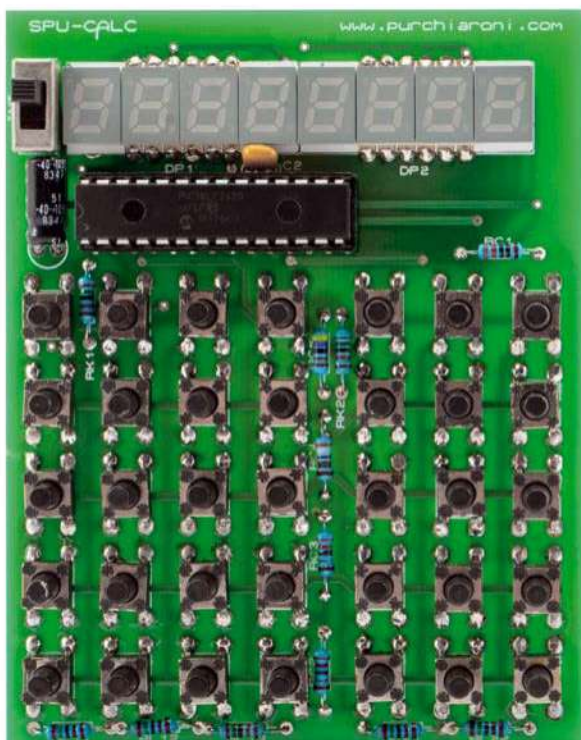
REALIZZAZIONE PRATICA

Passiamo quindi alla costruzione della nostra calcolatrice, iniziando dal circuito elettronico, per il quale abbiamo previsto e disegnato un circuito stampato a doppia faccia, del quale trovate le tracce nella sezione Download del nostro sito www.elettronica.in.it, insieme agli altri file di questo numero. Il PCB è progettato per ospitare componenti SMD, come ad esempio i moduli del Display, ma anche componenti a montaggio tradizionale,

vale a dire a foro passante. I pulsanti, se in versione SMD potranno comunque essere piazzati senza problemi sul lato "Componenti" del PCB, che offre le piazzole di montaggio su entrambi i lati. Sopra i pulsanti va posizionata la gabbia contenente i premi-tasti, anch'essi stampati in 3D, che consentono di creare una superficie livellata sulla quale applicare una stampa plastificata con il layout della tastiera (**Fig. 1**); prima di inserire i copripulsanti rovesciate questa gabbia, quindi inseriteli nelle rispettive finestrelle, magari aiutandovi con una pinzetta.

Vi suggeriamo di utilizzare del nastro adesivo in carta per unire il circuito alla gabbia con i premi-pulsanti, in modo da ottenere un insieme stabile e facilmente lavorabile.

Per sapere come assemblare le varie parti della calcolatrice riferitevi alla **Fig. 2**, che propone un esploso della calcolatrice e tutti i dettagli riguardanti il posizionamento del portatasti, dei tasti e



SW231÷SW235: Microswitch
C1: 22 μ F 63 VL elettrolitico
C2: 100 nF ceramico
U1: PIC18F2620-I/SP (MF1646)

INT: Deviatore a slitta verticale
DP1, DP2: Display 4 digit 7 segmenti
OSK-4028A-LR
BAT1: Porta batteria 2xAA

Varie:
- Zoccolo 14+14
- Circuito stampato S1646 (76x97 mm)

del plexiglass da collocare davanti ai display. La tastiera dovrà essere creata dall'immagine mostrata in **Fig. 3** che andrà stampata e plastificata, per essere poi applicata con del biadesivo sulla superficie superiore della gabbia.

Sul nostro sito Internet www.elettronica.in.it è disponibile il file PowerPoint con il layout, che è modificabile come descritto nel riquadro dedicato nel caso vogliate aggiungere o variare le funzioni della calcolatrice.

Sul lato posteriore del circuito andrà applicata la batteria, come mostrato nella **Fig. 3** nel caso di utilizzo di due pile AA o nella **Fig. 4**. Se si utilizza una batteria LiPo (batteria ricaricabile a polimeri di litio) o Li-ion.

Nel prototipo abbiamo scelto una Li-Po da 3,7V, ma è utilizzabile qualsiasi batteria con tensione compresa tra 2,7 e 3,9 volt, quindi nulla vi vieta di impiegare una serie di due stilo per complessivi 3V o di tre AA ricaricabili NiMH (3,6V).

Per l'ancoraggio vi consigliamo di utilizzare del nastro biadesivo gommoso (spesso) o della colla a caldo. Nel caso di una batteria ricaricabile, come la Li-Po, tagliare la pista che connette il piedino

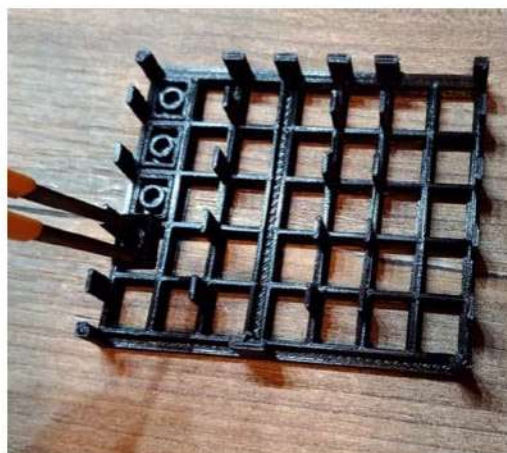


Fig. 1
Posizionamento
dei premi-
pulsanti nella
gabbia.

Per creare e rendere operativa una nuova funzione è possibile modificare il codice seguendo le semplici indicazioni fornite di seguito. Naturalmente la nuova versione di codice andrà compilata, con MikroC-Pro 7.6, ed in seguito caricata a bordo del microcontrollore con un apposito Programmatore, sostituendo quella già presente.

Si consiglia preventivamente di scaricare e mettere da parte la versione originale del codice dal microcontrollore stesso, tramite l'apposita funzione presente nel proprio kit di programmazione.

Questo consentirà un rapido ripristino, in caso di problemi.

Aggiungiamo per esempio la funzione →M per rimpiazzare il contenuto della memoria con il valore correntemente visualizzato, funzione da associare poi ad un tasto. Iniziamo con l'inserire il prototipo della funzione come mostrato di seguito. Il prototipo è solo la Dichiarazione della funzione, che sarà completata più avanti ma, per evitare errori durante la compilazione, va dichiarata prima delle funzioni che la utilizzeranno. Chiamiamo f_mst la funzione da aggiungere.

```
void f_log(FUNVARTYPE);
void f_10x(FUNVARTYPE);
```

```
void f_In (FUNVARTYPE);
void f_ex (FUNVARTYPE);
void f_mst(FUNVARTYPE);
void ExitFrac();
void ExitExp();
void Clr();
```

Associamo ora la funzione f_mst al tasto collocato all'intersezione della colonna 5 e della riga 7. La parte sinistra della seguente linea di codice associa la funzione alla pressione diretta del tasto. Mentre la parte destra permette di definire la "seconda funzione", da chiamare se prima del tasto si preme "2nd". Le etichette tra virgolette servono solamente in caso di debugging, che si abilita decommentando la definizione della flag di compilazione DEBUG_HW presente all'inizio del listato. Questo rende possibile l'uso del terminale virtuale nel debugger MikroC, per il cui impiego vengono però sottratte due linee dal display, che mostrerà due cifre oscurate.

```
/* 31 2 7 */ {{"TAN", f_tan, 0}, {"ATN", f_atn, 0}},
/* 32 3 7 */ {{"POW", f_pow, OP_POW}, {"", f_nop, 0}},
```

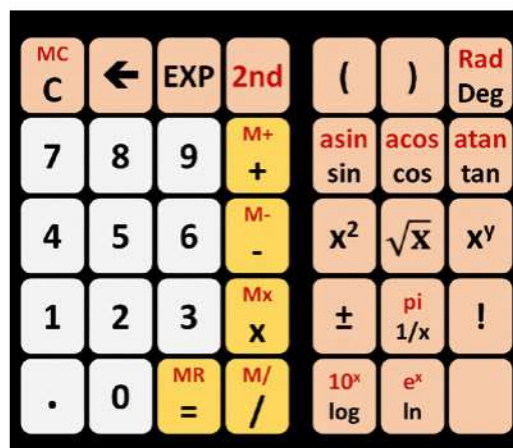
centrale con il lato "Off" del deviatore di accensione, ed eseguire i collegamenti come proposto nella Fig. 6 per consentire la ricarica solo a calcolatrice spenta. Una volta assemblato il circuito, chiuderlo nel contenitore come quello mostrato in Fig. 7, usando elementi da stampare in 3D a partire dagli STL forniti.

Come si vede, è prevista una gabbia intermedia che ospita dei "Premipulsanti", i quali mediano la pressione diretta verso ogni pulsante, offrendo una superficie uniforme sul lato esterno, sulla quale va applicata come già detto l'immagine della tastiera. Chi volesse creare una propria versione della Calcolatrice, dovrà definire le nuove



Fig. 2
Assemblaggio del circuito con le parti stampate in 3D.

Fig. 3
Immagine della tastiera.




```

/* 33  4  7*/ {{ "FAC", f_fac, 0}, {"", f_nop, 0}},
/* 34  5  7*/ {{ "->M", f_mst, 0}, {"", f_nop, 0}}
};

```

In seguito troviamo la definizione del corpo delle funzioni. Inseriamo `f_mst` con le istruzioni necessarie. Si osservi la struttura delle altre funzioni per comprendere come programmarla. Nel caso specifico ci limitiamo ad agire sul contenuto della variabile `MemVal` che rappresenta il valore presente in memoria, impostandolo col contenuto di `CurVal`, che è il valore correntemente visualizzato. Chiamando la funzione `Store` il nuovo valore di `MemVal` verrà anche salvato nella EEPROM del microcontrollore, in modo da ritrovarlo ad una futura accensione.

```

void f_sqr(FUNVARTYPE par) {CurVal = sqrt(CurVal); DoCalc();}
void f_pow(FUNVARTYPE par) {manage_oper(par);}
void f_mul(FUNVARTYPE par) {manage_oper(par);}
void f_mst(FUNVARTYPE par) {MemVal = CurVal; Store();}
void f_inv(FUNVARTYPE par) {ChangeSign();}
void f_rec(FUNVARTYPE par) {if (CurVal!=0) CurVal=1/CurVal; else

```

```

ShowMessage("ERR"); DoCalc();}
void f_pi (FUNVARTYPE par) {CurVal = _pi; DoCalc();}

```

Come ciliegina sulla torta, modifichiamo la membrana aggiornando la grafica per includere la nuova funzione $\rightarrow M$ utilizzando il file PowerPoint scaricabile dal sito. Una volta stampata la nuova tastiera adattandola ad una larghezza di 74 mm, essa va plastificata, ritagliata ed inserita in luogo della precedente tra la gabbia dei tasti e la cover superiore come mostrato nella Fig. A.

Fig. A
Aggiunta dell'etichetta per la nuova funzione " $\rightarrow M$ ".



funzioni in accordo col disegno da stampare, vale a dire che dovrà farsi una propria tastiera contenente tali funzioni.

La finestra sui display va chiusa con un pezzo di plastica trasparente rossa. Le due parti dell'involucro vanno chiuse con una piccola vite autofilettante.

IL SOFTWARE

Il codice di gestione, da caricare a bordo del microcontrollore, è disponibile per il download. Esso gestisce il display in multiplexing, e la pressione dei tasti. La struttura del codice è di per se un esempio didattico di gestione della pila di calcoli, che tiene conto della precedenza degli operatori,

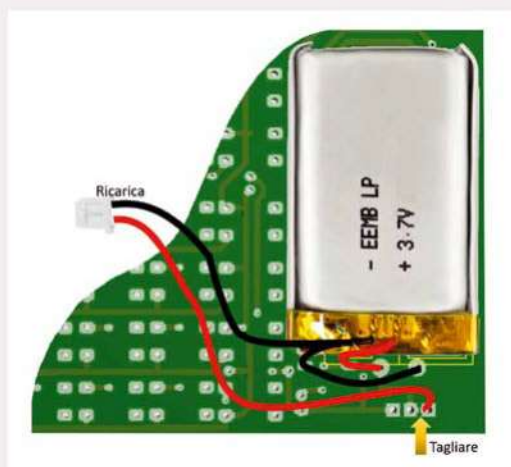


Fig. 4
Retro del circuito con batterie alcaline.



Fig. 5
Il circuito alimentato da una batteria a polimeri di litio.

Fig. 6
Collegamento
di una batteria
ricaricabile.



delle parentesi, delle operazioni unarie e della rappresentazione dei numeri nel formato più conveniente. È facilmente espandibile e permette numerose evoluzioni. La parte di codice relativa alla gestione del dato analogico proveniente dalla lettura della tastiera è quella vista inizialmente, così com'è generata dal programma Matrix. Qui di seguito andremo ad esporre le parti salienti del codice sorgente (interamente scaricabile e consultabile dal nostro sito www.elettronicaain.it) per approfondimenti.

Fig. 7
Calcolatrice
completa e
funzionante.



Il main

Dopo il setup iniziale, dove si impostano le direzioni e la tipologia delle linee di I/O, inizia il ciclo infinito di gestione come mostrato nel **Listato 1**. Ogni LOOPDELAY (1) millisecondi, nel primo dei due blocchi funzionali del ciclo il Display viene aggiornato mostrando la cifra di indice fig. Tale indice si incrementa ad ogni ciclo tornando a zero una volta raggiunto il valore NDIG (8). Ciò implementa il MultiPlexing, una modalità capace di ingannare l'occhio umano in maniera pressoché perfetta, con l'aggiornamento di una cifra ogni millisecondo. Nel secondo blocco, invece, si rileva se è stato premuto un tasto, chiamando la funzione GetKey. In caso positivo verrà chiamata la funzione corrispondente, prelevandone l'indirizzo dalla tabella fun assieme ad un parametro sempre definito nella stessa tabella, che viene mostrata nel **Listato 2**. La tabella indica la funzione da chiamare sia nel caso nominale che per la "seconda funzione" attivabile col tasto 2ND. Definisce anche una stringa utile solo in debugging attivando il Serial Monitor. In alcuni casi è definito un parametro, come per esempio per la funzione *f dig* comune a tutti i tasti numerici, dove il parametro indica il valore della cifra premuta. Si nota com'è facile in questo codice espandere la tabella aggiungendo nuove funzioni. Il corpo delle funzioni è molto semplice ed è mostrato nel **Listato 3**.

Si distinguono le funzioni unarie, che operano in modalità istantanea sull'ultimo numero immesso nella pila; esse eseguono una chiamata a DoCalc alla fine del calcolo, visibile nel **Listato 4**. Questa funzione completa i calcoli pendenti semplificando la pila. A tal proposito, alla fine di questo articolo è riportato un esempio concreto di gestione della pila di calcolo.

La funzione DoCalc mostrata nel **Listato 4** ha lo scopo di smaltire parti della pila LIFO (Last In First Out) che immagazzina operatori e operandi dei calcoli in sospeso. Un ciclo while valuta se sono presenti almeno tre elementi nella pila:

- un operando;
- un operatore;
- un operando.

La parte di codice in rosso mostra l'operazione eseguita, il cui risultato viene rimpiazzato dalla terna appena estratta, impostando il flag inloop a True per poter proseguire con eventuali ulteriori calcoli; osservate che tali azioni tengono conto della priorità degli operatori e delle parentesi aperte e chiuse.

↓ Listato 1 - Il ciclo principale di gestione

```
while (1) {                                     // Main loop

//----- Display management -----
    PORTC = ALLDIG_OFF;

    // Determine the figure to show
    if ((CurValStr[fig]>='0')&&(CurValStr[fig]<='9')) figure[fig]=CurValStr[fig]-'0';
    else switch (CurValStr[fig]) {
    case '-': figure[fig]=NEG; break;
    case 'R': figure[fig]=SEG_R; break;
    case 'A': figure[fig]=SEG_A; break;
    case 'D': figure[fig]=SEG_D; break;
    case 'G': figure[fig]=SEG_G; break;
    case 'E': figure[fig]=SEG_E; break;
    default: figure[fig]=NUL; break;
    }

    // Set the decimal point on the other cases when requested
    if ((DotPos>=0)&&(DotPos==fig)) dp=DOT; else dp=0;

#ifdef COMMON_ANODE
    PORTB = ~(segm[figure[fig]]+dp); // Set segments voltage adding the decimal point if any
    PORTC = (digit[fig]);           // Set current digit voltage
#endif
#ifdef COMMON_CATHODE
    PORTB = (segm[figure[fig]]+dp); // Set segments voltage adding the decimal point if any
    PORTC = ~(digit[fig]);          // Set current digit voltage
#endif

    fig++;                                     // Go to next figure
    if(fig >= NDIG) fig=0;                   // Roll at first figure once the last one has reached

//----- User entry management -----
    // Determine pressed key code
    if (!DEBUG_SW)
        PressedKey = GetKey();

    // Manage pressed key code
    if (PressedKey != KEY_NULL) {
        pfun=fun[PressedKey][f2Pressed].fmgr; // Pointer to target function
        pfun(fun[PressedKey][f2Pressed].par);
    }

    Delay_ms(LOOPDELAY);                      // Loop delay

}
} //~!
```

↓ Listato 2 - Tabella delle funzioni associate ai tasti

```
// Function vs Key id table
struct T_Function {
    char *name;
    void (*fmgr)(FUNVARTYPE);
    unsigned char par;
} fun[][2] = {
// KeyId | ROW | COL | Direct function | 2nd function
// -----|-----|-----|-----|-----
/* 0      1      1 */ {{"CLR", f_clr, 0}, {"MC", f_mc, 0}},
/* 1      2      1 */ {{"[7]", f_dig, 7}, {"", f_nop, 0}},
/* 2      3      1 */ {{"[4]", f_dig, 4}, {"", f_nop, 0}},
/* 3      4      1 */ {{"[1]", f_dig, 1}, {"", f_nop, 0}},
/* 4      5      1 */ {{"DOT", f_dot, 0}, {"", f_nop, 0}},

/* 5      1      2 */ {{"BCK", f_bck, 0}, {"", f_nop, 0}},
/* 6      2      2 */ {{"[8]", f_dig, 8}, {"", f_nop, 0}},
```

```

/* 7      3      2 */ {{ "[5]", f_dig, 5}, {"", f_nop, 0}},
/* 8      4      2 */ {{ "[2]", f_dig, 2}, {"", f_nop, 0}},
/* 9      5      2 */ {{ "[0]", f_dig, 0}, {"", f_nop, 0}},

/* 10     1      3 */ {{ "EXP", f_exp, 0}, {"", f_nop, 0}},
/* 11     2      3 */ {{ "[9]", f_dig, 9}, {"", f_nop, 0}},
/* 12     3      3 */ {{ "[6]", f_dig, 6}, {"", f_nop, 0}},
/* 13     4      3 */ {{ "[3]", f_dig, 3}, {"", f_nop, 0}},
/* 14     5      3 */ {{ "EQU", f_equ, 0}, {"MR", f_mr, 0}},

/* 15     1      4 */ {{ "2ND", f_2nd, 0}, {"", f_nop, 0}},
/* 16     2      4 */ {{ "ADD", f_add, OP_ADD}, {"M+", f_mpl, 0}},
/* 17     3      4 */ {{ "SUB", f_sub, OP_SUB}, {"M-", f_msb, 0}},
/* 18     4      4 */ {{ "MUL", f_mul, OP_MUL}, {"M*", f_mmu, 0}},
/* 19     5      4 */ {{ "DIV", f_div, OP_DIV}, {"M/", f_mdi, 0}},

/* 20     1      5 */ {{ " ( ", f_ope, 0}, {"", f_nop, 0}},
/* 21     2      5 */ {{ "SIN", f_sin, 0}, {"ASN", f_asn, 0}},
/* 22     3      5 */ {{ "QUA", f_qua, 0}, {"", f_nop, 0}},
/* 23     4      5 */ {{ "INV", f_inv, 0}, {"", f_nop, 0}},
/* 24     5      5 */ {{ "LOG", f_log, 0}, {"10x", f_10x, 0}},

/* 25     1      6 */ {{ " ) ", f_clo, 0}, {"", f_nop, 0}},
/* 26     2      6 */ {{ "COS", f_cos, 0}, {"ACS", f_acs, 0}},
/* 27     3      6 */ {{ "SQR", f_sqr, 0}, {"", f_nop, 0}},
/* 28     4      6 */ {{ "REC", f_rec, 0}, {"PI", f_pi, 0}},
/* 29     5      6 */ {{ "LN", f_ln, 0}, {"ex", f_ex, 0}},

/* 30     1      7 */ {{ "DEG", f_deg, 0}, {"RAD", f_rad, 0}},
/* 31     2      7 */ {{ "TAN", f_tan, 0}, {"ATN", f_atn, 0}},
/* 32     3      7 */ {{ "POW", f_pow, OP_POW}, {"", f_nop, 0}},
/* 33     4      7 */ {{ "FAC", f_fac, 0}, {"", f_nop, 0}},
/* 34     5      7 */ {{ "", f_nop, 0}, {"", f_nop, 0}},
);

```

Listato 3 - Funzioni di calcolo

```

//----- Calculus functions -----
void f_nop(FUNVARTYPE par) {}
void f_mc (FUNVARTYPE par) {MemVal = 0; Store();}
void f_clr(FUNVARTYPE par) {Clr(); ClrStk();}
void f_bck(FUNVARTYPE par) {Back();}
void f_exp(FUNVARTYPE par) {EnterExp();}
void f_2nd(FUNVARTYPE par) {toggle(&f2Pressed);}
void f_ope(FUNVARTYPE par) {manage_bkop();}
void f_clo(FUNVARTYPE par) {manage_bkcl();}
void f_deg(FUNVARTYPE par) {manage_deg();}
void f_rad(FUNVARTYPE par) {manage_deg();}
void f_dig(FUNVARTYPE par) {PushDigit(par);}
void f_add(FUNVARTYPE par) {manage_oper(par);}
void f_mpl(FUNVARTYPE par) {MemVal += CurVal; Store();}
void f_sin(FUNVARTYPE par) {CurVal = sin(CurVal*d2r); DoCalc();}
void f_asn(FUNVARTYPE par) {CurVal = asin(CurVal)*r2d; DoCalc();}
void f_cos(FUNVARTYPE par) {CurVal = cos(CurVal*d2r); DoCalc();}
void f_acs(FUNVARTYPE par) {CurVal = acos(CurVal)*r2d; DoCalc();}
void f_tan(FUNVARTYPE par) {CurVal = tan(CurVal*d2r); DoCalc();}
void f_atn(FUNVARTYPE par) {CurVal = atan(CurVal)*r2d; DoCalc();}
void f_sub(FUNVARTYPE par) {manage_oper(par);}
void f_msb(FUNVARTYPE par) {MemVal -= CurVal; Store();}
void f_qua(FUNVARTYPE par) {CurVal = pow(CurVal,2.0); DoCalc();}
void f_sqr(FUNVARTYPE par) {CurVal = sqrt(CurVal); DoCalc();}
void f_pow(FUNVARTYPE par) {manage_oper(par);}
void f_mul(FUNVARTYPE par) {manage_oper(par);}
void f_mmu(FUNVARTYPE par) {MemVal *= CurVal; Store();}
void f_inv(FUNVARTYPE par) {ChangeSign();}
void f_rec(FUNVARTYPE par) {if (CurVal!=0) CurVal=1/CurVal; else ShowMessage("ERR"); DoCalc();}
void f_pi (FUNVARTYPE par) {CurVal = _pi; DoCalc();}
void f_fac(FUNVARTYPE par) {Factorial();}
void f_dot(FUNVARTYPE par) {EnterFrac();}
void f_equ(FUNVARTYPE par) {manage_equ();}
void f_mr (FUNVARTYPE par) {CurVal = MemVal; DoCalc();}
void f_div(FUNVARTYPE par) {manage_oper(par);}
void f_mdi(FUNVARTYPE par) {if (CurVal!=0) {MemVal /= CurVal; Store();} else ShowMessage("ERR");}
void f_log(FUNVARTYPE par) {CurVal = log10(CurVal); DoCalc();}
void f_10x(FUNVARTYPE par) {CurVal = pow(10.0,CurVal); DoCalc();}
void f_ln (FUNVARTYPE par) {CurVal = log(CurVal); DoCalc();}
void f_ex (FUNVARTYPE par) {CurVal = pow(_e,CurVal); DoCalc();}

```


▼ Listato 4 - Funzione DoCalc

```
void DoCalc() {
// Execute all pending calculus
double v1, v2;
unsigned char f;
struct T_Stack lastop={0,nop}, prevop={0,nop}, elem;
unsigned short inloop=TRUE;

while (inloop) {

    inloop = FALSE;

    if (nstk >= 2) { // At least three elements are in the stack
        if (stk[nstk].f != NULL) lastop=Pop(); // Extract last function element if any
        prevop = stk[nstk-1]; // Previous expected element shapp be an operator

        // Check priority
        if (Oprio(lastop.f) <= Oprio(prevop.f)) { // Last op has less or same priority
            // Check stack contents validity. A Value-Function-Value sequence is expected
            if ((stk[nstk].f == NULL) && (stk[nstk-1].f != NULL) && (stk[nstk-2].f == NULL)) {

                // Extract the three elements
                elem = Pop(); v2 = elem.v;
                elem = Pop(); f = elem.f;
                elem = Pop(); v1 = elem.v;

                // Perform the calculus
                switch ( f ) {
                case OP_ADD : CurVal = v1 + v2; break;
                case OP_SUB : CurVal = v1 - v2; break;
                case OP_MUL : CurVal = v1 * v2; break;
                case OP_DIV : CurVal = v1 / v2; if (v2!=0) CurVal = v1 / v2; else ShowMessage("ERR"); break;
                case OP_POW : CurVal = pow(v1,v2); break;
                }

                // Insert the result in the stack
                PushVal();

                inloop = TRUE; // Continue with next calculus

            }
        }

        // Re-insert last function if defined
        if (lastop.f != NULL) {
            PushFun(lastop.f);
        }

        // Eliminazione parentesi attorno a singolo valore
        if ((stk[nstk].f == OP_CLO)
            && (stk[nstk-1].f == NULL)
            && (stk[nstk-2].f == OP_OPE)) {

            elem = Pop();
            elem = Pop(); CurVal = elem.v;
            elem = Pop();
            PushVal();

        }
        lastop.f = nop;

    }
}

if (ShowRes==0) { // Write Current string, when no messages have to be shown
    UserLen = dbl2str(CurVal,CurValStr); // Make the output string for display
}
NewVal = TRUE; // Set "New Value" flag
f2Pressed = FALSE; // Reset 2nd function mode
}
```

↓ Listato 5 - Funzione manage_oper

```
void manage_oper(unsigned char f) {
// Manage the entered function
    PushVal();                // Push the current value in the stack
    PushFun(f);               // Push the entered function in the stack
    DoCalc();                 // Perform the queued calculus
}
```

↓ Listato 6 - Funzioni GetKey

```
unsigned GetKey() {
// Return the code of the released key
    unsigned an;
    unsigned short i, KChan;
    unsigned res;

    res = KEY_NULL;           // Return No key pressed per default

    pk0 = k0;                 // Remember previous key under validation
    k0 = KEY_NULL;
    // Analog reading and pressed Key determination
    for (KChan=0; KChan<NUM_KEYBOARDS; KChan++) {
        if (!DEBUG_SW1)       // Debug - Inserimento manuale an
            an = Adc_Read(KChan);
        if (an < ar[NUM_KEYS]) {
            for (i=0; i<=(NUM_KEYS-1); i++) {
                if ((an>ar[i]-VALIDITY_WINDOW) && (an<ar[i]+VALIDITY_WINDOW)) {
                    k0 = ((KChan==0)?0:NUM_KEYS)+i; // Set key code (1...)
                    //k0 = (KChan+1)*100+(i+1);      // Set key code
                    i=NUM_KEYS; KChan=NUM_KEYBOARDS; // Code found. Close the two loops
                }
            }
        }
    }

    pk = k;
    if ((k0!=KEY_NULL)&&(k0==pk0)) { // Save last pressed key in "Previous valid Key" storage
        KeyConfirmations++;        // Same key as previous one
        if (KeyConfirmations >= NUM_CONFIRMATIONS) { // The entered Key has been validated
            KeyConfirmations = 0;   // Reset counter for next entry
            k = k0;                 // Promote the validated key code
        }
    } else {
        k = KEY_NULL;              // No valid key
        KeyConfirmations = 0;      // Reset counter due to failed verification
    }

    if ((k == KEY_NULL) && (pk != KEY_NULL)) { // Key released
        res = pk; // Return the previous valid key code after release
    }
    return res;
}
```

Il flag NewVal induce l'aggiornamento delle cifre da mostrare sul display. Gli operatori aritmetici invocano, invece, la funzione manage_oper, che ha come parametro l'enumerale del codice operazione; questa funzione è dettagliata nel **Listato 5**. Questa funzione, che viene chiamata alla pressio-

ne di un tasto relativo ad una operazione aritmetica, accoda il valore e l'operazione stessa nella pila, ed esegue DoCalc per eseguire i calcoli pendenti, se ve ne sono.

Lasciamo al lettore le altre funzioni di calcolo e concludiamo questi stralci di codice con la funzio-

↓ Listato 7 - Tabella con le tensioni associate ai tasti

```
// 21 analog values windows for 20 keys per each keyboard. The last value is a No-Key-Pressed boundary
unsigned ar[21] = {491,387,264,172,53,517,424,317,237,137,541,457,362,293,208,695,658,621,597,569, 800};
```

								)	)											
							4	4	4											
						+	+	+	+	)										
					3	3	3	3	3	7										
				(	(	(	(	(	(	(	7	7								
			X	X	X	X	X	X	X	X	X	X								
		2	2	2	2	2	2	2	2	2	2	2	14	14						
	+	+	+	+	+	+	+	+	+	+	+	+	+	+						
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	15	

Fig. 7
Esempio
di gestione
della pila di
calcolo.

ne GetKey, mostrata nel **Listato 6**. Nella funzione GetKey viene letta la tensione misurata sui piedini definiti come entrate analogiche (ossia associati all'A/D converter), ed attestati alle NUM_KEYBOARDS (2) tastiere.

Come si può vedere, rispetto al codice generato da Matrix, che pur avremmo potuto usare senza alcun problema, si è preferito ridurre la lunghezza eseguendo un ciclo di confronti basato sulla tabella mostrata nel **Listato 7**. La lettura ed il confronto vengono eseguiti per un numero di volte pari al valore di NUM_CONFIRMATIONS (il valore predefinito è 2) in modo da scartare le conseguenze di fenomeni di rimbalzo.

La tabella contiene il valore dell'ADC prodotto dalla chiusura del partitore associato a un tasto; a questo sarà applicata una quantità VALIDITY_WINDOW, sia negativa che positiva, per definire una finestra di validità all'interno della quale viene confermata la pressione di quel tasto.

GESTIRE LA PILA DI CALCOLO

Con la **Fig. 7** vogliamo concludere questo articolo, mostrando un esempio di gestione della pila di calcolo, che si evolve man mano che vengono immessi operandi ed operatori.

Nell'esempio, l'immissione dei tasti "1 + 2 x (3 + 4)" lascia il calcolo in sospeso fino alla parentesi chiusa, dove possiamo eseguire l'ultima addizione, rimpiazzando la terna "4+3" con il suo risultato "7". La coppia di parentesi attorno al singolo operando viene eliminata, e si può in seguito eseguire l'operazione prioritaria "7X2", rimpiazzando questi

tre elementi col singolo elemento risultante, ossia "14", e così via. In ogni calcolo intermedio, notate che il risultato del calcolo stesso va a rimpiazzare operandi e operatore con quelli ottenuti.

CONCLUSIONI

Bene, con questo esempio di gestione della pila di calcolo possiamo ritenere concluso questo lungo articolo, nel quale vi abbiamo presentato la nostra calcolatrice a 360°: descrivendola sommariamente, spiegando perché abbiamo scelto una tastiera analogica e descrivendo come l'abbiamo implementata con l'aiuto di un comune microcontrollore Microchip ad 8 bit con ADC integrato, ma anche snocciolando ogni aspetto della programmazione, fino ad arrivare a note costruttive che comprendono anche spunti per consentire di personalizzare, a ciascuno di voi, la vostra calcolatrice "vintage". □



Cosa occorre?

Il display utilizzato in questo progetto è disponibile presso Futura Elettronica (cod. DISPLAY4DR) a Euro 1,90, mentre il microcontrollore (cod. PIC18F2620-I/SP) è disponibile a Euro 20,00. Il circuito stampato può essere realizzato con il servizio PCBPRODUCTION. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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



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
€ 78,00



cod. TK102
€ 54,00

MICRO LOCALIZZATORE SATELLITARE GPRS/GSM

Tutto sotto controllo con il nuovo micro localizzatore GPRS/GSM. Per attivarlo basta inserire una SIM di qualsiasi operatore; inviando un SMS potrai attivare l'ascolto ambientale oppure ricevere un link con le coordinate che indicano la posizione in tempo reale del localizzatore. Dispone di batteria agli ioni di litio ricaricabile. Ideale per tenere sotto controllo la posizione di autoveicoli, persone, cose, animali domestici, bagagli, ecc. La confezione comprende: il micro localizzatore, il cavetto USB-micro USB per la ricarica della batteria e le istruzioni in italiano. Dimensioni: 43,2x32x14 mm. Peso: 20 grammi.



cod. A8TRACK
€ 14,00

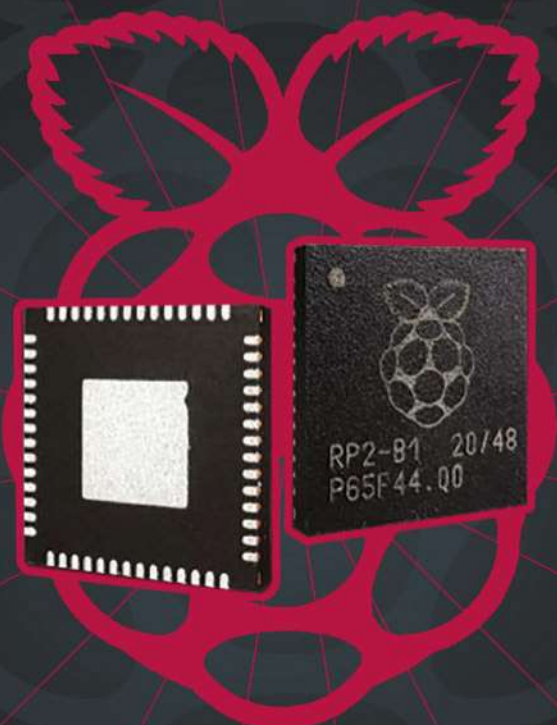


LOCALIZZATORE TASCABILE GPS/GSM/GPRS

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

Programmare con RP2040

di PIER CALDERAN



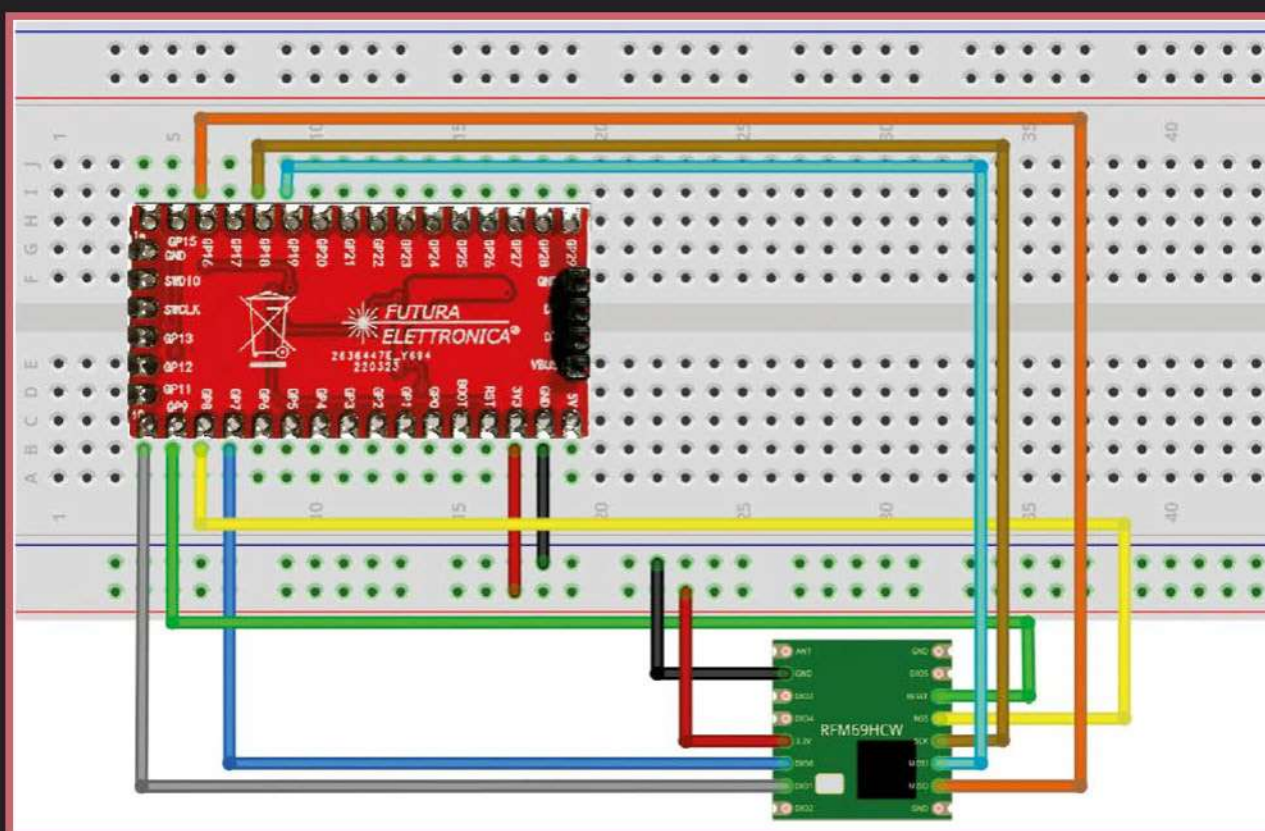
PUNTATA N. 4

In questa puntata continuiamo a parlare di LoRa/LoRaWAN. Imparerete a sfruttare il cloud TTN per creare un dispositivo IoT con visualizzazione dei dati su Node-RED.

Nella scorsa puntata vi abbiamo spiegato come collegare la nostra scheda FTR2040 a un breakout LoRa/LoRaWAN della HopeRF RFM95W e come connetterla a The Things Network (Fig.1). Se siete riusciti a connettere il vostro end device a un gateway di TTN avrete sicuramente visto la stringa "Hello world!", arrivata sotto forma di dati grezzi, ovvero una serie di numeri esadecimali corrispondenti ai caratteri ASCII della stringa. Ovviamente, avete avuto bisogno di ricorrere a un convertitore da *hex* a *string* per verificare il tutto. On-line esistono diversi decoder, ad esempio <https://codebeautify.org/hex-string-converter> ma ricorrere a questi servizi, per quanto molto utili, è anche parecchio scomodo. Vediamo quindi come far compiere questa conversione in modo automatico all'applicazione TTN.

PAYLOAD FORMATTER

Se avete già creato un'applicazione su TTN, con almeno un end device registrato, potete procedere nella lettura. In caso contrario rileggete la scorsa puntata che vi spiega come farlo nei dettagli. Accedendo all'applicazione e al vostro end device, dovrete vedere sul menu a sinistra della pagina del device una serie di voci, fra cui Payload formatters (Fig.2). Qui potete trovare una serie di cosiddetti "formattatori di payload" di diverso tipo. Quello che useremo come Payload formatter è all'interno del menu *Uplink* e si chiama *Custom Javascript formatter*, selezionabile dal menu a tendina. Questo Payload formatter è molto facile da usare e consente di scrivere direttamente nell'editor una funzione Javascript che servirà, appunto, a formattare il Payload.



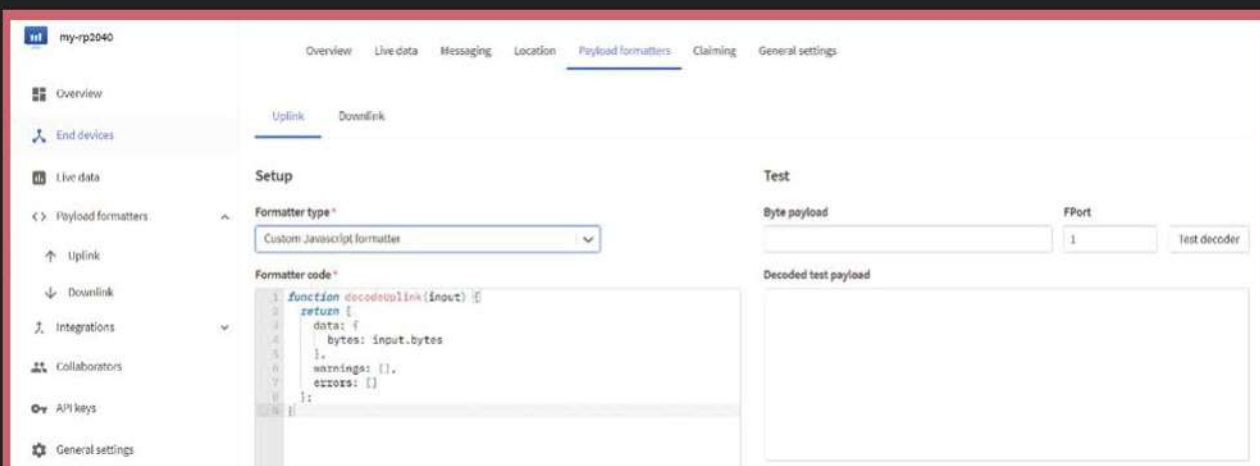
All'interno dell'editor, dovreste vedere una funzione predefinita simile a questa:

```
function decodeUplink(input) {
  return {
    data: {
```

```

        bytes: input.bytes
    },
    warnings: [],
    errors: []
};
}

```



Listato 1

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

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

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

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

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

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

Come potete intuire, la funzione non fa nulla di speciale e bisognerà scrivere del codice che abbia senso. Per esempio, supponiamo di voler formattare il valore di temperatura inviato dal nostro end device. Come già dovreste sapere dalla puntata 2 del corso, il chip RP2040 è dotato di un sensore di temperatura incorporato e più avanti spiegheremo come scrivere il codice per leggere i dati da questo e mandarli a TTN tramite l'end device. La funzione che dovreste scrivere all'interno dell'editor potrebbe essere simile a quella che illustriamo nel **Listato 1**. Una volta scritta la funzione fate clic sul pulsante Save change. Come potete vedere nel listato, abbiamo commentato solo la parte che ci interessa di più all'interno della funzione Decoder:

```
function Decoder(bytes, port) {
  var decoded = {};
  decoded.t1 = bytes[0]; // assegnazione della temp alla var
  return decoded;
}
```

La variabile `decoded` è un oggetto JSON vuoto. Per chi non lo sapesse, i valori letterali degli oggetti JSON sono racchiusi tra parentesi graffe {}.

I valori letterali dell'oggetto JSON contengono coppie chiave/valore separate da due punti. Pertanto, con `decoded.t1 = bytes[0]` assegniamo il valore del byte in ingresso alla chiave `t1` dell'oggetto. Quindi avremo un output tipo `t1:28` dove `t1` è la chiave e 28 il valore dell'oggetto JSON. Ovviamente il valore cambierà in base alla temperatura letta dal sensore.

IL CODICE DI ESEMPIO

Il codice di esempio si trova nella stessa cartella *examples* che avete clonato nella scorsa puntata dal repository GitHub di Sandeep Mistry.

Se non lo avete ancora fatto, eseguite il seguente comando:

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

Fra le diverse sottocartelle, trovate *otaa_temperature_led* che contiene tre file:

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

Aprite il file *config.h* e modificatelo in questo modo:

```
#define LORAWAN_REGION          LORAMAC_REGION_EU868
#define LORAWAN_DEVICE_EUI      "70B3D57ED0054DEF"
#define LORAWAN_APP_EUI        "0000000000000000"
#define LORAWAN_APP_KEY         "9CABA532B2630BE471F0C769B81A977"
```

Ovviamente, questo è solamente un esempio; dovete sostituire le stringhe con quelle dell'end device che avete creato in precedenza su TTN. Quindi, dopo aver modificato il file *config.h*, potete proseguire e compilare il codice per il vostro end device.

Per fare questo, entrate nella cartella *pico-lorawan*, create una cartella *build* (se non l'avete già fatto), entrate nella cartella ed eseguite prima *cmake* e quindi *make*.



Fig. 3 - Il debug del dispositivo sul terminale.

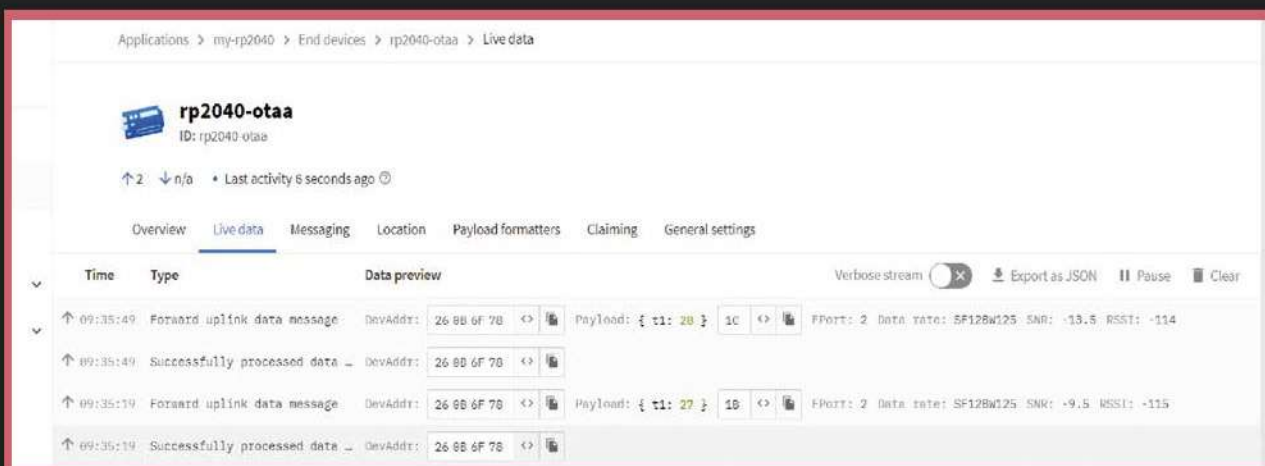


Fig. 4 - I dati di temperatura in arrivo nel device su TTN.

La compilazione potrebbe richiedere qualche minuto:

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

Se tutto va bene dovreste avere il file `pico_lorawan_otaa_temperature_led.uf2` nella cartella `build/examples/otaa_temperature_led`. A questo punto, potete caricare questo file UF2 nella scheda nel modo consueto, ovvero mettendo a massa il pin BOOT e inserendo il breakout nella presa USB. In questo modo, si aprirà la cartella del dispositivo. Trascinate al suo interno il file UF2 e riavviate la scheda togliendo il corto a massa e ricollegando il dispositivo alla presa USB.

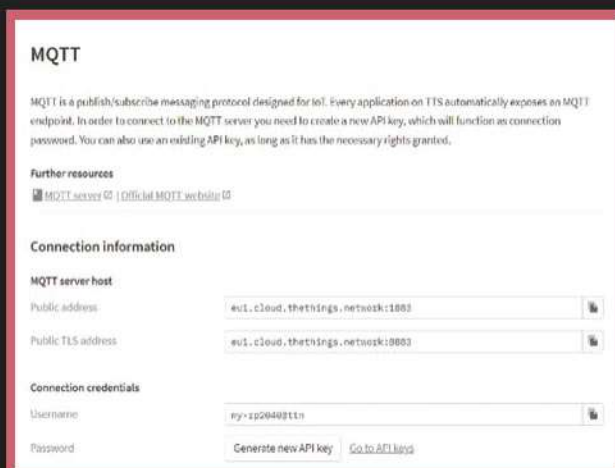


Fig. 5 - Pagina Integrations per l'integrazione MQTT di TTN

Dal terminale potete lanciare il comando seguente per vedere il debug della porta seriale USB:

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

Se il collegamento ha successo, avrete (Fig. 3):

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

success! significa che l'inizializzazione del dispositivo ha avuto successo.

Joining LoRaWAN network ... joined successfully! significa che il collegamento (join) con l'applicazione su TTN ha avuto successo.

sending internal temperature: 28 °C (0x1c)... success! significa che il messaggio è stato inviato con successo a TTN sotto forma di numero esadecimale.

Essendo l'esempio `otaa_temperature_led` basato su una connessione di tipo OTAA (Over-The-Air-Activation) bisognerà aspettare TTN per il JOIN, cioè l'unione fisica all'applicazione: i dispositivi eseguono una procedura di unione con la rete (join), durante la quale viene assegnato un indirizzo di dispositivo dinamico e le chiavi di sicurezza vengono negoziate con il dispositivo (vedi il BOX OTAA e ABP).

Quando apparirà "success", significa che il dispositivo è unito (joined) all'applicazione e dovreste vedere arrivare i dati nella pagina dell'applicazione e del device (Fig. 4).

Il dato di temperatura viene formattato esattamente come previsto dal Payload formatter. Infatti invece di un solo numero esadecimale, dovreste vedere la stringa `t1` seguita dal valore di temperatura già in gradi Celsius.

Per esempio `t1:28`.

OTAA e ABP

Ogni end device deve essere registrato con una rete prima di inviare e ricevere messaggi. Questa procedura è nota come attivazione. Sono disponibili due metodi di attivazione.

OTAA

Over-The-Air-Activation (OTAA) è il metodo di attivazione più sicuro e consigliato (soprattutto da TTN) per gli end device. I dispositivi eseguono una procedura di unione con la rete (join), durante la quale viene assegnato un indirizzo di dispositivo dinamico e le chiavi di sicurezza vengono negoziate con il dispositivo.

ABP

Activation By Personalization (ABP) richiede l'hardcoding dell'indirizzo del dispositivo e delle chiavi di sicurezza nel dispositivo. ABP è meno sicuro di OTAA e presenta anche lo svantaggio che i dispositivi non possono cambiare provider di rete senza cambiare manualmente le chiavi nel dispositivo.

Potete trovare ulteriori informazioni sui metodi attivazione a questo link:

<https://www.thethingsnetwork.org/docs/lorawan/end-device-activation>

ATTENZIONE! L'esempio *otaa_temperature_led* prevede di mandare i dati la stringa del sensore ogni 30 secondi. Dato che TTN pone delle limitazioni nella Community Edition, ovvero alla versione gratuita, vi consigliamo di portare l'invio ad almeno 60 secondi.

Per fare questo, andate nella cartella dell'esempio e aprite il file *main.c*. Andate alla riga seguente e cambiate il valore dei millisecondi fra un messaggio e l'altro portandolo a 60000:

```
if (lorawan_process_timeout_ms (60000) == 0)
```

INTEGRAZIONE MQTT E NODE-RED

MQTT (Message Queue Telemetry Transport) è un protocollo di messaggistica di pubblicazione/sottoscrizione progettato appositamente per l'IoT e per questo motivo scegliere questo tipo di integrazione per poter visualizzare i dati

su un nostro server Node-RED.

TTN mette a disposizione diversi tipi di integrazione.

Per scoprire quelle previste da TTN è sufficiente aprire il menu a sinistra alla voce *Integrations* e troverete queste possibilità:

- MQTT
- Webhooks
- Storage integrations
- AWS IoT
- Azure IoT
- LoRa Cloud

Facendo clic su MQTT, si aprirà una pagina simile alla Fig. 5. Qui troverete tutte le informazioni da inserire in seguito in Node-RED per l'integrazione MQTT di TTN.

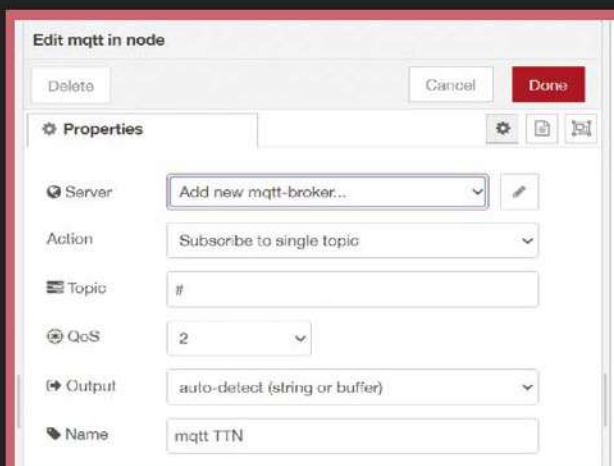


Fig. 6 - Il nodo MQTT.

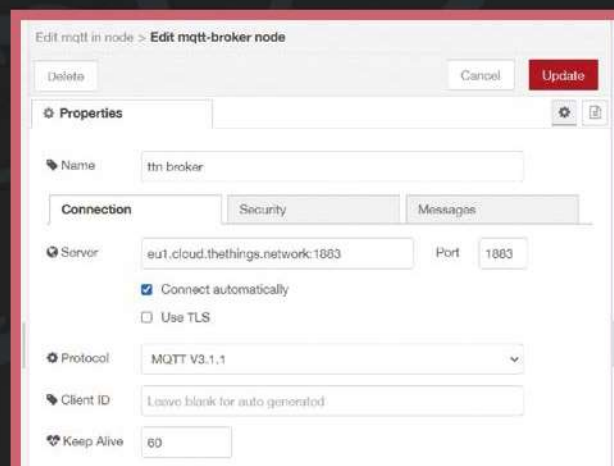


Fig. 7 - Il tab Connection del nodo MQTT



Fig. 8 - Il tab Security del nodo MQTT.



Fig. 9 - Il nodo MQTT In connesso a TTN.

MQTT server host
Public address eu1.cloud.thethings.network:1883
Public TLS address eu1.cloud.thethings.network:8883

Connection credentials

Username: my-rp2040@ttn
Password: Generate new API key

Oltre all'indirizzo del server MQTT e la porta di connessione, vengono date le credenziali di accesso, che sono relative al nome utente, ovvero il nome dell'applicazione che state utilizzando seguito da @ttn, e la password che viene generata

Listato 2

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

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

return [msg1]
```

tramite il pulsante Generate new API key. Dovrete copiare queste informazioni dopo aver creato un flow su Node-RED, come spieghiamo nel prossimo paragrafo.

MQTT E NODE-RED

Fra i framework più utilizzati per la visualizzazione di dati IoT su un'interfaccia grafica web-based, crediamo che Node-RED sia sicuramente ai primi posti.

Per approfondire l'uso di Node-RED vi consigliamo il corso Node-RED pubblicato su Elettronica In dal numero 258 al numero 265. Per la connessione con TTN, dovete innanzitutto creare un nodo MQTT In, cioè un nodo che rimarrà in ascolto dei dati del broker MQTT di TTN.

In un flow di Node-RED trascinate il nodo MQTT In dalla barra laterale sinistra ed editate il contenuto in questo modo (Fig. 6):

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

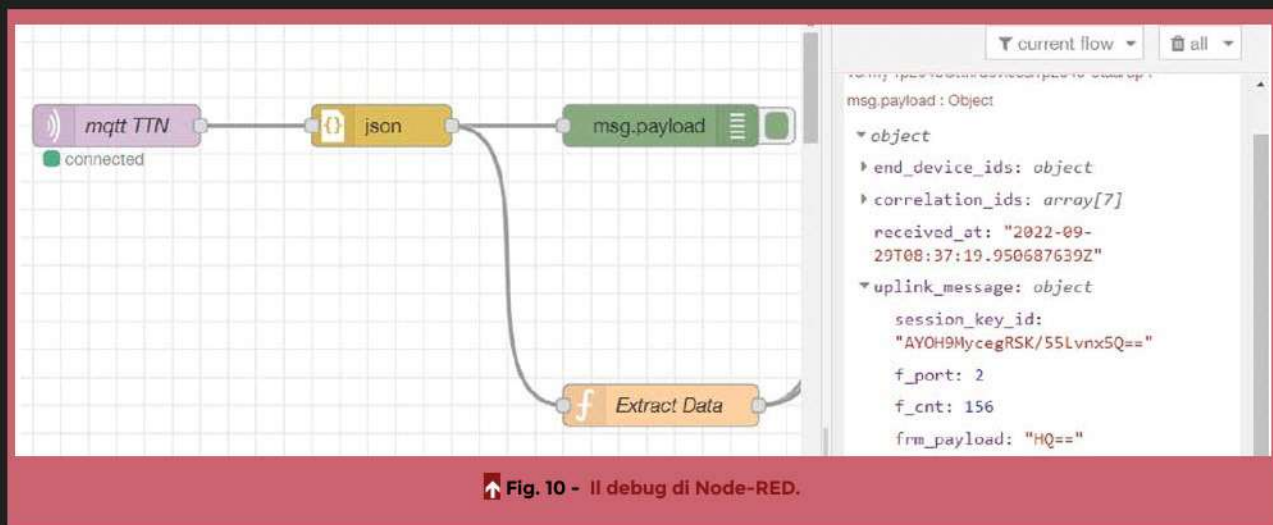


Fig. 10 - Il debug di Node-RED.

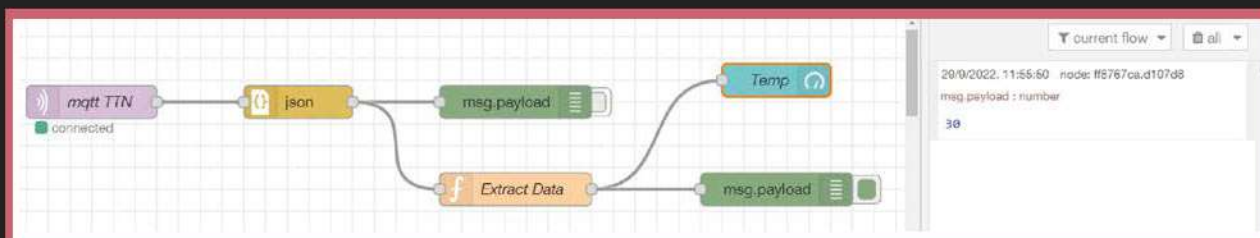


Fig. 11 - Il debug di Node-RED con solo il valore di temperatura.

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

Nel tab *Connection* inserite l'indirizzo del broker TTN e un nome qualsiasi (Fig. 7):

Name: ttn broker

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

Nel tab *Security* inserite Username e password generati da TTN (Fig. 8).

Username: my-rp2040@ttn

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

Fate clic su *Update* e *Done* per chiudere l'editor e quindi su *Deploy* per distribuire il nodo. Dovreste vedere la connessione attivarsi, ovvero visualizzare *connected* in corrispondenza del nodo MQTT In (Fig. 9). Senza entrare nei dettagli, possiamo solo dirvi che i dati del nodo vanno convertiti in un oggetto JSON da un nodo *json* collegato all'uscita del nodo MQTT In. Collegando un nodo *debug* all'uscita del nodo *json* dovreste vedere nella finestra di debug qualcosa di simile alla Fig. 10. Appariranno tutti i dati provenienti dall'end device, compreso la coppia chiave:valore che abbiamo decodificato con il Payload formatter: `{"t1":28}`

A questo punto possiamo collegare un nodo *function* e scrivere una funzione che "isoli" solo il dato che ci interessa, ovvero il valore di temperatura. Per fare questo, all'interno del nodo *function* scriviamo una funzione che trovate nel Listato 2. Nella variabile *obj* creiamo una coppia chiave:valore composta dalla chiave "temp" e dal valore *msg.payload.uplink_message.decoded_payload.t1*, ovvero dal valore di *t1*, che corrisponde alla chiave di temperatura. Con la variabile *var res_temp = obj["temp"]* assegniamo il valore decodificato della temperatura e con *var msg1 = { payload:res_temp }* creiamo l'oggetto da restituire con *return [msg1]*. Collegando un nodo *debug* all'uscita del nodo *function* dovreste vedere solo il valore di temperatura (Fig. 11). A questo punto, per visualizzare il valore di temperatura nella dashboard di Node-RED, basta aggiungere un nodo *gauge* e collegarlo all'uscita del nodo *function*. Ovviamente dovreste creare anche un gruppo e tab per visualizzare il nodo *gauge* nella dashboard. Andando all'in-

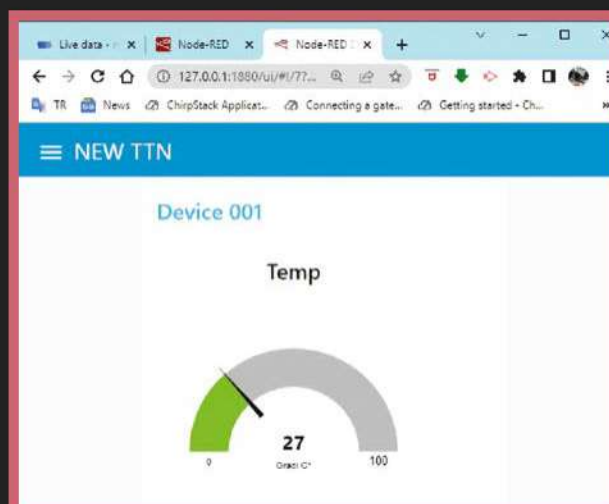


Fig. 12 - Il debug di Node-RED con solo il valore di temperatura.

dirizzo della dashboard (solitamente 127.0.0.1:1880/ui) dovreste vedere qualcosa di simile alla Fig. 12.

Ogni qualvolta il valore di temperatura del vostro end device cambierà, vedrete l'ago del gauge muoversi e indicare la temperatura esattamente come è stata rilevata.

CONCLUSIONI

Con la speranza di essere stati sufficientemente chiari, vi invitiamo a sperimentare il progetto presentato in queste pagine e vi aspettiamo il mese prossimo per continuare la sperimentazione con la nostra scheda FTR2040. □

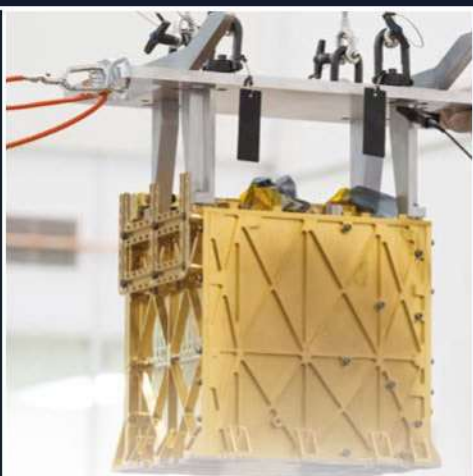


Cosa occorre?

La board di Futura Elettronica con RP2040 (cod. FTR2040) è disponibile a Euro 13,90 e include i pin strip da saldare, il modulo LoRa (cod RFM95-868) è in vendita a Euro 13,50. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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



Ossigeno per gli astronauti su Marte

La pianificazione del viaggio di ritorno è parte integrante dei preparativi per una missione su Marte con equipaggio.

Gli astronauti richiederanno 50 tonnellate di propellente per razzi per il decollo e circa 31 tonnellate di ossigeno. Questo peso va trasportato sin dalla partenza dalla Terra, ma l'ideale sarebbe produrre l'ossigeno dalla stessa atmosfera marziana ricca di anidride carbonica, utilizzando un sistema chiamato MOXIE, ossia Mars Oxygen ISRU (In-Situ Resource Utilization), un'unità di 18 chilogrammi ospitata all'interno del rover Perseverance su Marte, il cui compito è scomporre elettrochimicamente l'anidride carbonica raccolta dall'atmosfera marziana in ossigeno puro e monossido di carbonio. Gli esperimenti condotti su Marte hanno dimostrato la capacità di produrre ossigeno sia di notte che di giorno, quando le temperature possono variare di oltre 100 °C. I tassi di generazione e purezza dell'ossigeno soddisfano anche i requisiti per la produzione di propellente per razzi e per la respirazione. MOXIE è un primo passo verso un sistema molto più ampio e complesso per supportare l'esplorazione umana di Marte. I ricercatori stimano un tasso di generazione richiesto da 2 a 3 chilogrammi all'ora; tenendo conto degli 8 mesi per arrivare su Marte, in 14 mesi sarebbe possibile avere l'ossigeno che occorre al viaggio di rientro. Il sistema è energivoro e potrebbe essere alimentato da reattori a fissione Kilopower della NASA, che vanno trasportati su Marte e, non potendo atterrare tutti nello stesso luogo, andranno interconnessi tramite cavi.

mars.nasa.gov

La metasuperficie: potente come una stanza di computer

Gli scienziati dei Sandia National Laboratories e del Max Planck Institute for the Science of Light hanno riferito di un dispositivo che potrebbe sostituire una stanza piena di apparecchiature, collegando i fotoni in un effetto quantistico chiamato "entanglement". Questo dispositivo, una specie di materiale nanotecnologico chiamato metasuperficie, apre la strada all'entanglement di fotoni in modi complessi che non sono stati possibili con le tecnologie solide.

Quando gli scienziati dicono che i fotoni sono entangled, significa che sono collegati in modo tale che le azioni su uno influiscano sull'altro, non importa dove o quanto distanti siano i fotoni nell'universo: è un effetto della meccanica quantistica.

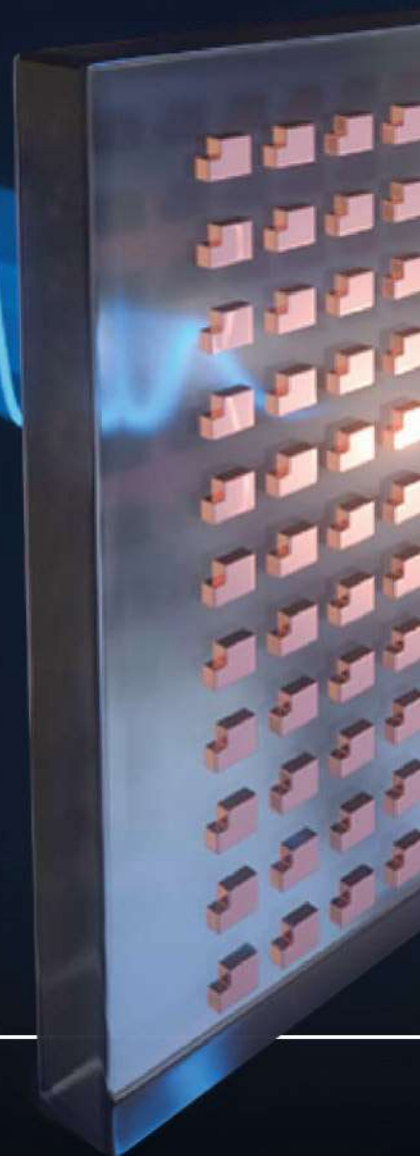
Sebbene il fenomeno possa sembrare strano, gli scienziati lo hanno sfruttato per elaborare le informazioni in modi nuovi. Ad esempio, l'entanglement aiuta a proteggere le informazioni quantistiche delicate e a correggere gli errori nell'informatica quantistica, un campo che un giorno potrebbe avere un impatto enorme sulla sicurezza nazionale, sulla scienza e sulla finanza. L'entanglement consente inoltre nuovi metodi di crittografia avanzati per comunicazioni sicure.

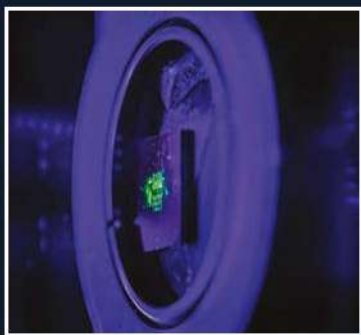
Nell'innovativo dispositivo, che è cento volte più sottile di un foglio di carta, entra luce ed escono i fotoni entangled. La nuova metasuperficie funge da porta a questo insolito fenomeno quantistico.

Invece di spostarsi attraverso il loro nuovo dispositivo, gli scienziati lo attraversano con un laser. Il raggio di luce passa attraverso un campione ultrasottile di vetro ricoperto da strutture su scala nanometrica fatte di un comune materiale semiconduttore chiamato arseniuro di gallio, che confonde tutti i campi ottici.

Occasionalmente, una coppia di fo-

toni entangled a diverse lunghezze d'onda emerge dal campione nella stessa direzione del raggio laser in arrivo. Questo dispositivo è progettato per produrre reti complesse di fotoni entangled, non solo una coppia alla volta, ma diverse coppie tutte intrecciate insieme e alcune che possono essere indistinguibili l'una dall'altra. Alcune tecnologie necessitano di queste complesse varietà di cosiddetto multi-entanglement per





sofisticati schemi di elaborazione delle informazioni. Anche altre tecnologie in miniatura basate sulla fotonica del silicio possono implementare fotoni entanglement, ma senza il tanto necessario livello di multi-entanglement complesso.

Finora, l'unico modo per produrre tali risultati era con più piani di laser, cristalli specializzati e altre apparecchiature ottiche.

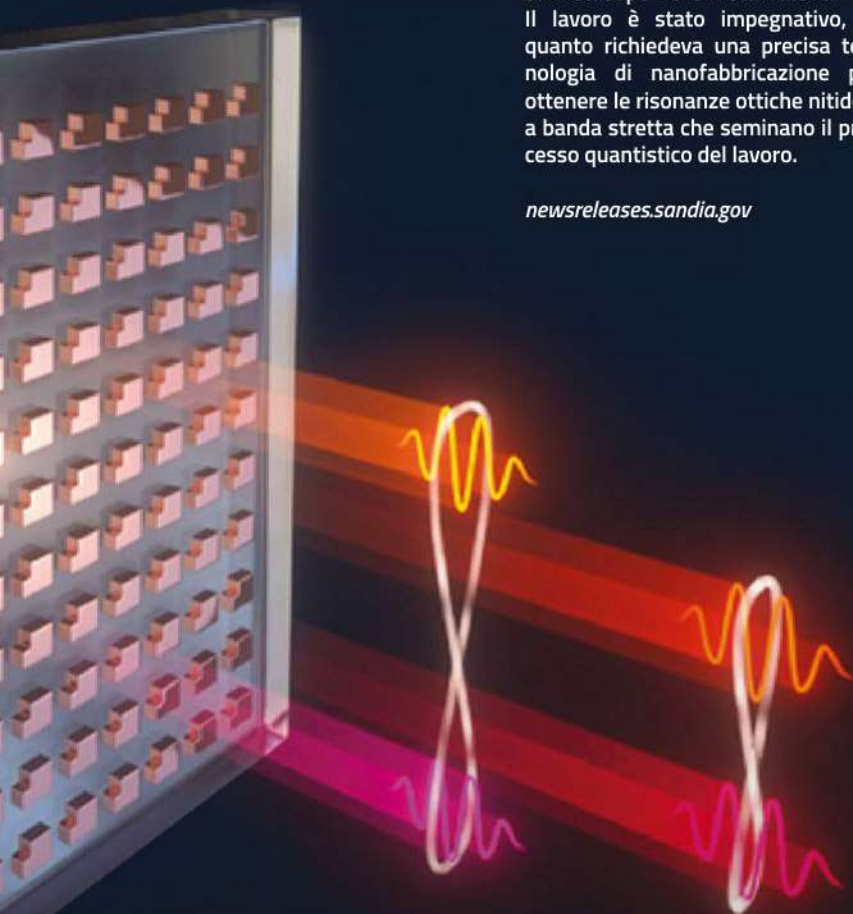
Il team ha sintonizzato con successo la propria metasuperficie per produrre fotoni entangled con lunghezze d'onda variabili, un precursore fondamentale per la generazione simultanea di diverse coppie di fotoni intrinseci. Tuttavia, i ricercatori osservano che l'efficienza del loro dispositivo - la velocità con cui possono generare gruppi di fotoni entangled - è inferiore a quella di altre tecniche e deve essere migliorata.

Una metasuperficie è un materiale sintetico che interagisce con la luce e altre onde elettromagnetiche in modi impossibili nei materiali convenzionali.

Sandia è una delle principali istituzioni al mondo che effettua ricerche su metasuperfici e metamateriali.

Il lavoro è stato impegnativo, in quanto richiedeva una precisa tecnologia di nanofabbricazione per ottenere le risonanze ottiche nitide e a banda stretta che seminano il processo quantistico del lavoro.

newsreleases.sandia.gov



Sintetizzare il metanolo dal gas di altoforno nei mini-impianti

Il metanolo è una sostanza chimica di base dell'industria e uno dei principali vettori energetici nell'economia dell'idrogeno. Tuttavia, la produzione convenzionale di metanolo da carbone e gas naturale genera grandi emissioni di gas serra. Con il progetto Carbon2Chem si sta valutando la possibilità di produrre metanolo dai gas di scarto industriali, ad esempio della siderurgia. A riguardo, in un miniimpianto campione del Fraunhofer ISE è stata dimostrata per la prima volta la stabilità a lungo termine della sintesi del metanolo dal gas d'altoforno purificato, con una capacità di produzione di dieci litri al giorno per un totale di oltre 5.000 ore. La sintesi di metanolo fossile e la produzione di acciaio tramite altoforno a carbone sono responsabili di significative emissioni di CO₂; la sinergia fra i due processi consente di sostituire il metanolo dai combustibili puramente fossili facendo reagire l'idrogeno verde con le emissioni della produzione di acciaio. L'impianto del Fraunhofer ISE si basa su due reattori adiabatici non raffreddati e un riciclo di tipo industriale dei gas non reagiti. I gas di acciaieria vengono trattati in un sistema di purificazione del gas di thyssenkrupp Industrial Solutions che utilizzano catalizzatori e assorbenti Clariant e sono privi di veleni catalizzatori per la successiva sintesi. Un importante prerequisito per l'ottimizzazione del processo è il modello cinetico, ottenuta con un gemello digitale del miniimpianto.

www.fraunhofer.de

ZF e Freudenberg per veicoli pesanti alimentati da celle a combustibile



La divisione Commercial Vehicle Solutions (CVS) di ZF ha annunciato un accordo di sviluppo congiunto con il fornitore di fuel-cell e sistemi di batterie, Freudenberg e-Power Systems, per lo sviluppo di soluzioni "powerpack" e-Drive pulite, costituite da una cella a combustibile e un powertrain. L'obiettivo iniziale dell'accordo è lo sviluppo di una soluzione e-drive a celle a combustibile altamente integrata per veicoli commerciali. Mentre Freudenberg offrirà sistemi di alimentazione elettrica scalabili in kit con varie potenze, ZF offrirà sistemi di trasmissione elettrici completi fino

a 360 kW di potenza continua. Ciò è in linea con la nuova trasmissione centralizzata elettrica CeTrax 2 di ZF per veicoli pesanti. La cooperazione di ZF con Freudenberg svilupperà soluzioni di e-drive a celle a combustibile altamente integrate per l'industria dei veicoli commerciali. Riducendo i costi di sviluppo e progetto dei produttori, le soluzioni powerpack aiuteranno anche a ridurre il costo totale di proprietà. È attualmente in corso la fase pilota dell'accordo, che vede lo sviluppo di prototipi per veicoli dimostrativi di autobus e camion; il primo di essi è previsto entro il 2023. Ciò corrisponde alla partnership già annunciata di ZF per le celle a combustibile per lo sviluppo di un prototipo di autobus nell'ambito del progetto HyFleet. La tecnologia a fuel-cell potrebbe coprire, entro il 2030, una quota del 20 % dei veicoli commerciali pesanti.

www.zf.com



Dragon Flying Robot, il drone operaio tuttfare

Siamo ormai abituati a veder volare droni e multicotteri, che oltre allo svago sono destinati ad impieghi operativi che si ampliano di anno in anno: dopo i droni militari, con scopi di combattimento e spionaggio o assistenza alle operazioni belliche e pre-belliche, dopo i droni corrieri impiegati per recapitare pacchi, dopo i droni per la manutenzione delle linee elettriche, ora è arrivato anche il manipolatore volante tuttfare Dragon. Si tratta di una combinazione di multicotteri opportunamente dimensionati e connessi, che può sostituire alcuni lavori manuali come ad esempio aprire e chiudere valvole idrauliche. Dragon ha più gradi di libertà di quelli che potremmo immaginare, tanto da modificare la sua forma per adattarsi a piccoli spazi, manipolare e afferrare gli oggetti e, come testato, azionare valvole industriali. La cosa singolare è che la forza necessaria deriva dalla spinta delle eliche, non dagli attuatori del robot. Dragon pesa ben 7,6 chilogrammi e, sebbene il suo carico utile sia di 3,4 kg, il tempo di volo massimo di 3 minuti è un vincolo che dovrà essere risolto prima che il drone approdi alla vita reale. Con la quantità di gradi di libertà che ha, teoricamente è in grado di fare tutto ed il suo creatore, il ricercatore Moju Zhao della University of Tokyo, sta lavorando per dare a Dragon la possibilità di camminare, oltre che di operare in volo.

www.jsk.tu-tokyo.ac.jp/~chou/

Mercedes EQXX: 1.200 km di elettrico!

Non molto tempo fa, un'autonomia di 300 miglia sembrava irraggiungibile dalle auto elettriche. Oggi, la Mercedes-Benz Vision EQXX sfata quel mito, dato che l'elegante prototipo di berlina viaggiando per 1.202 km dalla Germania al circuito di Formula 1 a Silverstone, in Inghilterra, con una batteria di scorta. La EQXX, è una piccola berlina di lusso Mercedes, che dovrebbe essere in commercio entro il 2024, in una versione costruita sulla nuova piattaforma Mercedes Modular Architecture dell'azienda, la stessa di Classe A e la CLA Coupé. Sviluppato in soli 18 mesi, il Vision EQXX a trazione posteriore è alimentato da un unico motore elettrico a flusso radiale, alimentato da un pacco

batteria con poco meno di 100 kWh. I suoi 241 cv permettono di passare da 0 a 100 in 7 secondi, ma anche di percorrere grandi distanze grazie al Cx di appena 0,17 e ad un pannello fotovoltaico sul tetto, con 117 celle, che riduce il carico alimentando un sistema convenzionale a 12 volt per far funzionare gli accessori di bordo come il quadro, l'autoradio, i vetri elettrici ecc. Contrariamente alle comuni auto elettriche, batteria e motore sono raffreddati ad aria, risparmiando il consumo di circuiti di raffreddamento a liquido e pompe.

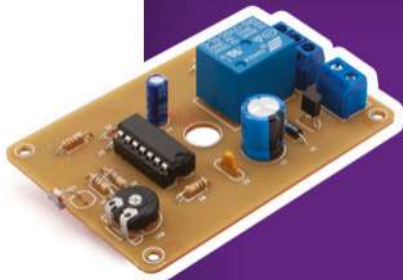
www.mercedes-benz.com



Entra nel mondo dell'elettronica!

Le scatole di montaggio KIT di Futura Elettronica sono semplici da montare e di facile utilizzo, ideali per avvicinarsi al fantastico mondo dell'elettronica.

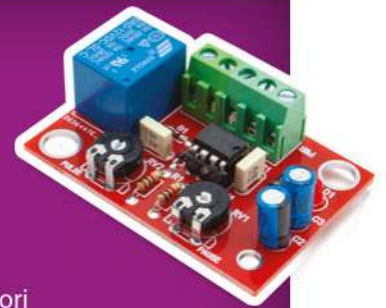
Non utilizzano microcontrollori e sono fornite con schemi elettrici e istruzioni così da capire e personalizzare il funzionamento.



cod. FK001 | € 9,00

Interruttore Crepuscolare

Semplice circuito che permette di accendere e spegnere l'illuminazione elettrica, o altri utilizzatori, quando la luce ambientale è troppo bassa.



cod. FK002 | € 12,00

Modulo Intermittenza

Permette di attivare a intermittenza circuiti e apparati elettrici, come lampadine, proiettori o elementi per modellismo.



cod. FK003 | € 10,50

Alimentatore stabilizzato da 1,5 a 35 V

Piccolo e compatto alimentatore in grado di fornire una tensione di uscita regolabile tra 1,5 e 35 V con una corrente di 1 A.



cod. FK004 | € 14,00

Alimentatore stabilizzato duale e regolabile

Completo alimentatore che consente di ottenere tensioni stabilizzate simmetriche da 3 a 15 Vcc e correnti di 1 A per ogni ramo.



cod. FK005 | € 14,00

Timer Start e Stop da 1 a 240 secondi

Permette di attivare per un predefinito tempo, tramite il relé di bordo, un utilizzatore ed anche di disattivarlo manualmente prima che l'intervallo stabilito sia trascorso.



cod. FK006 | € 14,00

Interruttore Battimani

Riconosce il rumore del battito delle mani (ed anche quelli simili per spettro di frequenza) e attiva in modo astabile (temporizzato) o bistabile un dispositivo alimentato elettricamente.

Nuovo oscilloscopio **WiFi** per PC e tablet 2 canali - 70 MHz

ALIMENTAZIONE
AUTONOMA
CON BATTERIA
RICARICABILE
AL LITIO!



cod. DSOPC1070A

€ 265,00

CARATTERISTICHE TECNICHE PRINCIPALI

- Canali: 2
- Banda passante: 70 MHz
- Frequenza di campionamento (in tempo reale): 250 MS/s usando un solo canale - 125 MS/s usando due canali
- Impostazione del coefficiente di attenuazione della sonda: x1, x10, x100, x1000
- Tensione di ingresso massima: 35 Vpk (CC + picco)
- Sensibilità di ingresso: 10 mV/div ~ 5 V/div
- Funzione FFT: (Fast Fourier Transform) integrata

Può essere collegato contemporaneamente con iPad e iPhone.

ANTENNA WiFi

PRESA USB

CH 1

CH 2

EXT

