

Elettronica In

www.elettronica.in.it

€ 6,00 - Anno XXVI - n. 244 - APRILE 2020



Fotografare oggetti a 360°

Intelligenza artificiale con Jetson Nano



Recensione alimentatori step down



■ Energy meter sul Cloud

■ STM32 Nucleo

■ OBD questo conosciuto

■ Da una USB
a quattro
seriali



Tutto sotto controllo ovunque e in ogni momento!



Manuale di istruzioni
passo-passo per il
collegamento e l'utilizzo.
Tutto in italiano!

TELECAMERA IP WiFi MOTORIZZATA con funzione Smart Tracking Motion 2MPX ONVIF

Vuoi avere sempre tutto sotto controllo anche quando non sei a casa? Niente di più facile! Con la mini telecamera IP a colori motorizzata dotata di ottica fissa da 3,6 mm, 6 LED IR, con funzione Smart Tracking Motion (monitoraggio del movimento) puoi fare questo facilmente! Il modulo Wi-Fi integrato consente di collegare la telecamera in remoto permettendoti di visualizzare su PC e Smartphone (iOS, Android), quanto ripreso dal suo obiettivo che può ruotare sia orizzontalmente sia verticalmente. Grazie al microfono e all'altoparlante integrato potrai comunicare con chi è vicino alla telecamera. Puoi inoltre registrare video su PC oppure sulla micro SD card (max 128 GB, acquistabile separatamente). È possibile visualizzare dall'APP fino a 9 telecamere contemporaneamente.

€ 56,00
cod. CP729



€ 54,00
cod. CP728

Istruzioni
in italiano!

TELECAMERA IP WiFi MOTORIZZATA da 2MPX ONVIF

Permette la visione da remoto tramite smartphone con sistema Android e iOS. Consente di registrare video sulla memoria dello smartphone tramite apposita APP, oppure su micro SD Card (max. 128GB, acquistabile separatamente). Dispone di microfono e di altoparlante che permettono di parlare e ascoltare da remoto.





La fine del Mondo? No, la fine di un Mondo

Mentre ci apprestavamo a scrivere dei vantaggi di una tecnologia rispetto ad un'altra o di come quel nuovo micro avrebbe migliorato le prestazioni di quel sistema, è successo qualcosa di imprevedibile.

La natura, ancora una volta, ha fatto sentire forte la sua voce, infischiosene delle nostre (presunte) capacità di dominare tutto e tutti: è bastato un piccolo, invisibile, microorganismo per mettere in ginocchio interi Paesi. E questa volta, sfortunatamente, al 10 marzo, nel momento in cui scriviamo queste note, uno dei paesi più colpiti è proprio l'Italia, dove la maggior parte delle attività sono bloccate, le scuole sono chiuse e dove le persone sono costrette a rinchiusersi

Sfruttiamo queste settimane per imparare da casa qualcosa di nuovo, un linguaggio di programmazione o una nuova tecnologia.

in casa. E dove si contano già centinaia di decessi.

Se da quasi un secolo non abbiamo più assistito a catastrofiche guerre globali, grazie a quel poco di ragionevolezza che ci è

rimasta, negli ultimi tempi è la natura che ci ricorda quanto siamo vulnerabili rispetto ai grandi e ricorrenti fenomeni naturali, come terremoti ed eruzioni vulcaniche, ma anche – e sempre più spesso – a quei catastrofici fenomeni dovuti alla manipolazione dell'ambiente da parte dell'uomo, alla deforestazione indiscriminata e alla trasformazione dell'atmosfera in una camera a gas. Con questo non vogliamo dire che l'attuale epidemia da Coronavirus sia dovuta all'attività umana (anche se qualcuno lo sospetta), semplicemente che la natura non sempre è benigna e aggiungerci del nostro non è una cosa molto intelligente.

Qualcuno ha detto che ci troviamo di fronte alla fine del Mondo. No, molto più semplicemente ci troviamo di fronte alla fine di un Mondo. E se il nuovo Mondo sarà meglio o peggio di quello che ci stiamo lasciando alle spalle dipenderà da ognuno di noi, da quali insegnamenti sapremo trarre da quanto accaduto. Ciò detto, ci preme un'altra considerazione. Sicuramente, prima o poi tutto questo finirà, le nostre lunghe giornate tra le mura di casa termineranno e la vita, in qualche modo, riprenderà.

Il consiglio che vogliamo dare è proprio quello di non sprecare questo periodo di isolamento nell'ozio e nell'inerzia. Sfruttiamo queste settimane per imparare qualcosa di nuovo, qualcosa che magari non eravamo riusciti a fare per mancanza di tempo: un corso di lingue, un linguaggio di programmazione, una nuova tecnologia. Fortunatamente da questo punto di vista Internet è una miniera senza fine, così come il nostro sito www.elettronica.in e il nuovo www.elettronica.inpro.it dai quali potrete scaricare, gratuitamente, articoli, corsi, progetti, monografie, tutorial e informazioni di ogni genere.

In attesa dell'arrivo del Nuovo Mondo.

Arsenio Spadoni
@arsenio.spadoni@elettronica.in

In copertina:
Una piattaforma rotante controllata da Raspberry Pi per fotografare a tutto tondo qualsiasi oggetto per poi dividerlo in formato foto, GIF e video.



ON LINE

elettronica.in

SI RINGRAZIA

Futura Elettronica
Microkit
RM Elettronica
Skytel

Elettronica In

Rivista mensile, anno XXVI n. 244

APRILE 2020

EDITORIALE

Direttore Responsabile

Arsenio Spadoni

@arsenio.spadoni@elettronica.in

Redazione

Stefano Garavaglia, Paolo Gaspari,
Boris Landoni, Davide Scullino,
Gabriele Daghetta

@redazione@elettronica.in

Grafica

Alessia Sfulcini

@alessia.sfulcini@elettronica.in

Ufficio Pubblicità

Monica Premoli (0331-752668)

@monica.premoli@elettronica.in

Ufficio Abbonamenti

☎ Tel. 0331-752668

@abbonati@elettronica.in

Direzione, Redazione, Pubblicità

FUTURA GROUP srl - Divisione Editoriale
via Adige 11 - 21013 Gallarate (VA)

☎ Tel. 0331-752668

Abbonamenti

Annuo 10 numeri Euro 45,00

Esteri 10 numeri Euro 45,00 (digitale)

Le richieste di abbonamento vanno inviate a:

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

☎ Tel. 0331-752668

oppure tramite il sito

<https://www.elettronica.in/abbonamenti/>

Distribuzione per l'Italia

SO.DI.P. Angelo Patuzzi S.p.A.
via Bettola 18 - 20092 Cinisello Balsamo (MI)

☎ Tel. 02-660301 ☎ Fax 02-66030320

Stampa

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

Tiratura
18.000 copie



Mensile
associato all'USPI,
Unione Stampa
Periodica Italiana

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

Prezzo di copertina Euro 6,00.

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

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

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

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

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

© 2020 Futura Group srl



Contenuti

01

RUBRICHE

- 01 Editoriale
- 05 Q&A Questions & Answers
- 09 Embedded IoT
Digital Transformation of Things
- 13 Componenti & Sistemi
- 17 Appuntamenti & Eventi
- 122 Scienza & Tecnologia
- 126 Fonti Rinnovabili

Progetto di Copertina

CAPTURE 360



70

35



Periferiche

Una USB per
quattro seriali

23

ARTICOLI

DIDATTICA

23

OBD questo conosciuto

Tutto quello che c'è da sapere sulla porta di diagnosi dei moderni autoveicoli e sui dispositivi che vi si possono interfacciare. Prima puntata.

AUDIO

35

Una USB per quattro seriali

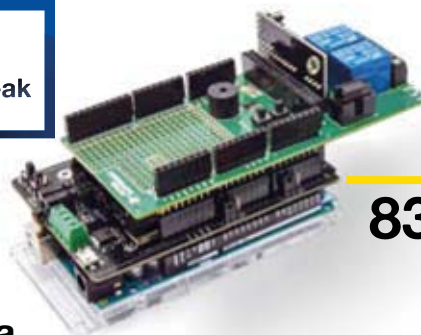
Grazie all'integrato FTDI FT4232, consente di assegnare a una sola porta USB del computer fino a quattro porte seriali.

Didattica

OBD questo conosciuto



23

In vetrina**Alimentatori commerciali
step down****55****Didattica
STM Nucleo****45****Periferiche****Energy Meter IoT****83****Didattica
Corso Kivy****113****Didattica****AI con hardware
specializzato:
Jetson Nano****101****45 DIDATTICA
STM-32 Nucleo**

Conosciamo e impariamo a utilizzare le schede di prototipazione di ST, anche nell'ottica della programmazione tramite l'IDE di Arduino.

**55 IN VETRINA
Alimentatori commerciali
step down**

Abbiamo selezionato e provato per voi alcuni alimentatori DC/DC commerciali di tipo "Buck / Step down"; eccone le conclusioni.

**70 VISIONE
Capture 360**

Realizziamo una piattaforma rotante controllata da Raspberry Pi per fotografare a tutto tondo qualsiasi oggetto per poi condividerlo in formato foto, GIF e video.

**83 MISURE
Energy Meter IoT**

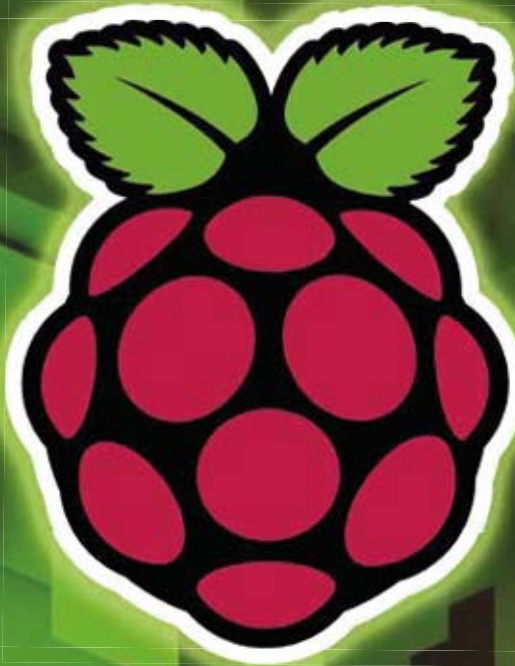
Interfacciamo il nostro Energy Meter al mondo del web per portare sulla piattaforma ThingSpeak le misure e vederne i risultati in comodi grafici.

**101 DIDATTICA
AI con hardware specializzato:
Jetson Nano**

Sviluppiamo applicazioni di Machine Learning utilizzando una piccola ma potente piattaforma realizzata da Nvidia.

**113 DIDATTICA
Conoscere e usare Kivy**

Sperimentiamo con un'interfaccia per la visualizzazione di dati provenienti da sensori e attuatori. Quarta puntata.



RASPBERRY Pi 4

Utilizzala subito con i nuovi set!



CON
RASPBERRY Pi 4
DA **4 GB**

€ 79,00

Cod. RPI4BULK4GB

Il set comprende:

- Board Raspberry Pi 4 da 4 GB di memoria
- Alimentatore 5V-3A con connettore micro USB
- Adattatore Micro USB a Type C

€ 89,90

Cod. RASPKITV10



Il set comprende:

- Board Raspberry Pi 4 con 2 GB di memoria
- Box trasparente per Raspberry Pi 4 con ventola
- Alimentatore 5V-3A
- Micro sdcard hc da 32GBb + adattatore
- Adattatore da micro hdmi a hdmi
- Minuteria varia



Differenze tra LoRa e LoRaWAN

So a cosa serve e dove si utilizza la tecnologia Long Range LoRa ma non riesco a capire la differenza tra LoRa e LoRaWAN. A cosa si riferiscono, esattamente, questi due termini?

Mario Rimoldi > Varese

R: Semplicemente LoRa è una tecnica di modulazione per uno spettro wireless specifico, mentre LoRaWAN è un protocollo aperto che consente ai dispositivi IoT di utilizzare LoRa per la comunicazione.

LoRa, o Long Range, è una tecnologia wireless proprietaria, a bassa potenza e a lungo raggio che utilizza uno spettro wireless senza licenza: in Europa è la banda degli 868 MHz mentre negli Stati Uniti è la banda dei 915 MHz.

Grazie al particolare sistema di modulazione utilizzato, questa tecnologia consente una portata compresa tra 3 e 15 km, utilizzando una potenza RF di pochi milliwatt; il tutto va a scapito della velocità di trasmissione che comunque può

Queste pagine sono dedicate alle richieste, ai suggerimenti ed alle segnalazioni dei lettori.

Raccomandiamo, per quanto possibile, di proporre argomenti di interesse generale.

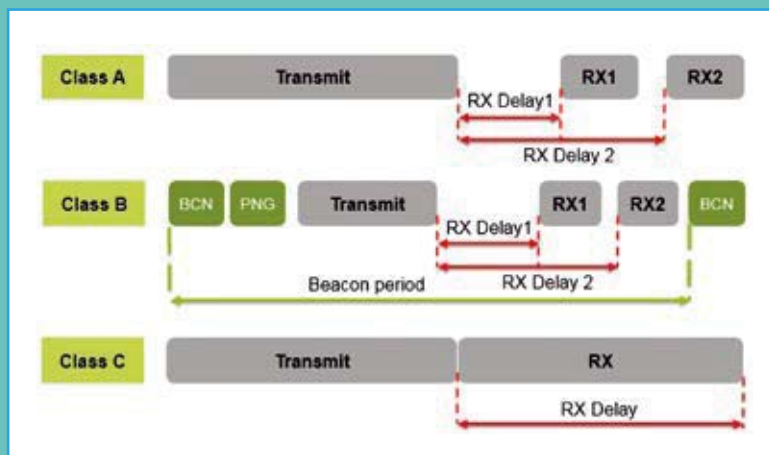
Contattateci all'indirizzo: redazione@elettronica.it

raggiungere i 50 kbps. La tecnologia LoRa appartiene a Semtech, che fornisce anche i circuiti integrati necessari. Il bassissimo consumo consente di alimentare i dispositivi IoT LoRa con comuni batterie che sono in grado di garantire un'autonomia di funzionamento superiore ai 10 anni. Dal punto di vista della rete,

LoRa crea solo un metodo a livello fisico di trasporto wireless, come un qualsiasi chip ricetrasmittitore. Ciò significa che manca il protocollo di rete per gestire il traffico per la raccolta dei dati e la gestione dei dispositivi endpoint.

È qui che entra in scena LoRaWAN, un protocollo aperto basato su cloud - progettato e gestito da LoRa Alliance - che consente ai dispositivi di comunicare in modalità wireless con LoRa. In sostanza, LoRaWAN utilizza la tecnologia wireless LoRa

e aggiunge un componente di rete, incorporando al contempo l'autenticazione del nodo e la crittografia dei dati. Dal punto di vista della distribuzione IT aziendale, le reti LoRaWAN sono ideali per i dispositivi IoT che monitorano continuamente lo stato di qualcosa e quindi attivano gli avvisi ai gateway quando i dati monitorati superano una soglia specificata. Questo tipo di dispositivi IoT richiedono poca larghezza di banda e possono funzionare a batteria per mesi o addirittura anni. Mentre la specifica definisce l'implementazione tecnica, non definisce alcun modello commerciale o tipo di implementazione (pubblica, condivisa, privata, aziendale) e offre quindi al settore la libertà di innovare e differenziare il modo in cui viene utilizzata.



↑ Le tre classi previste dal protocollo LoRaWAN.

La programmazione dei LED driver

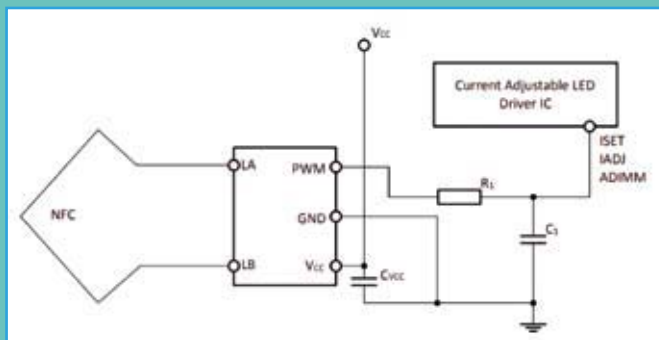
Ho letto che esistono dei LED che possono essere programmati in modalità wireless. Quale è lo scopo di questa procedura?

Roberto Milani > Como

R: In realtà non sono i LED che



↑ Schema di programmazione dei driver LED tramite NFC.



↑ Schema applicativo del chip Infineon NLM0010 da utilizzare all'interno del driver LED per rendere possibile la programmazione NFC.

vengono programmati, bensì i driver LED. Attualmente i produttori di apparecchi a LED per illuminazione utilizzano una varietà di metodi di programmazione per impostare le caratteristiche operative dei driver LED all'interno degli apparecchi. Per semplificare ed armonizzare questa procedura, alcuni produttori hanno dato vita al consorzio MD-SIG (Module-Driver Interface Special Interest Group) che ha standardizzato con successo un approccio di programmazione wireless digitale utilizzando la comunicazione near-field (NFC). Grazie alla standardizzazione MD-SIG, i produttori di apparecchi di illuminazione che impiegano driver LED di produttori diversi possono usare le stesse interfacce di programmazione NFC per configurare i driver, razionalizzando la produzione. NFC consente al produttore di impostare in modalità wireless i parametri operativi del driver LED, come la corrente di funzionamento, l'emissione di lumen costanti e i livelli di regolazione. Questo può essere fatto su una linea di produzione senza la necessità di applicare la tensione al driver. I membri MD-SIG includono i principali

fornitori globali di driver LED e alcune di queste aziende hanno già implementato approcci diversi alla programmazione NFC. Ora, queste aziende hanno deciso di armonizzare i loro metodi di programmazione e useranno lo standard ISO / IEC 15963 per NFC. Il sistema NFC utilizza un dispositivo di programmazione, comunemente noto come lettore NFC, che trasferisce in modalità wireless i dati di programmazione a un IC NFC all'interno del driver LED. I parametri sono programmati utilizzando un lettore NFC standardizzato in combinazione con un software specifico per il fornitore del driver LED. Un elenco di lettori NFC qualificati è disponibile sul sito Web MD-SIG (vedere le specifiche di programmazione NFC). Le attrezzature e il software conformi a MD-SIG consentono ai produttori di apparecchi di illuminazione di ottimizzare la loro produzione e di utilizzare driver LED di diversi fornitori utilizzando lo stesso hardware di programmazione.

La tecnologia UWB per il tracking

Ho visto che la tecnologia UWB viene sempre più spesso utilizzata per determinare la posizione di oggetti, merci o persone all'interno di edifici o spazi all'aperto. Come funziona esattamente questo sistema?

Dario Brescia > Venezia

R: La tecnologia UWB (Ultra Wide Band) consente di trasmettere dati digitali in modalità wireless sfruttando un'ampia banda di frequenze per la propagazione del segnale sulle brevi distanze.

Nel caso dell'UWB vengono utilizzati impulsi di energia a radiofrequenza di durata temporale estremamente

ridotta (da poche decine di picosecondi a qualche nanosecondo) e quindi con occupazione spettrale molto ampia. Questi impulsi sono infatti rappresentati da pochi cicli d'onda di una portante in radiofrequenza e quindi lo spettro in frequenza associato a questa forma d'onda è estremamente ampio (da cui il nome).

Grazie all'ampio spettro di frequenze elettromagnetiche utilizzato, l'ultra wide band è in grado di trasportare grandi quantità di dati ad una distanza massima di circa 100-200 metri impiegando una quantità minima di energia elettrica (circa 1 mW). Inoltre, questa tecnologia è in grado di trasmettere il segnale anche attraverso porte e muri, così da propagarsi senza problemi anche all'interno di spazi chiusi come abitazioni o locali pubblici (centri commerciali, ad esempio).

Pensata e progettata per lo scambio di dati su breve distanza, l'ultra wide band basa il proprio funzionamento sullo scambio di impulsi digitali tra sorgente e ricevitori a intervalli di tempo estremamente regolari. A far la differenza tra l'ultra-wide band e altre tecnologie di scambio dati wireless, infatti, è sostanzialmente la tempistica: la sincronizzazione tra il nodo che invia il segnale e i nodi che lo ricevono deve avere un'accuratezza di pochi trilionesimi di secondo (un trilionesimo equivale a uno diviso un miliardo di miliardi ovvero 1 diviso 10 elevato alla 18esima potenza).

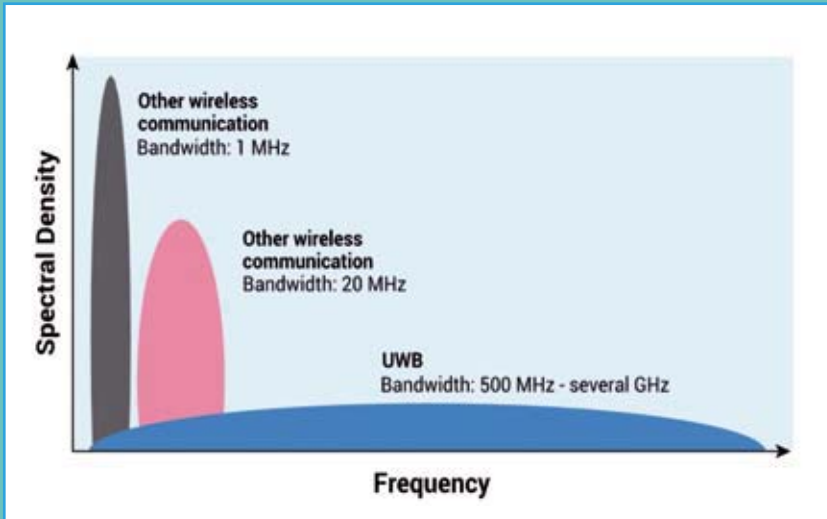
Questa necessità di coordinamento tra antenna sorgente e ricevente ha due conseguenze pratiche degne di nota. Da un lato, lo scambio di impulsi digitali tra sorgente e ricevente è molto frequente e ciò permette la condivisione praticamente istantanea anche di grosse moli di informazioni (le applicazioni più recenti dell'UWB riescono a trasferire dati a una velocità di oltre

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



↑ Densità spettrale per UWB e banda stretta.

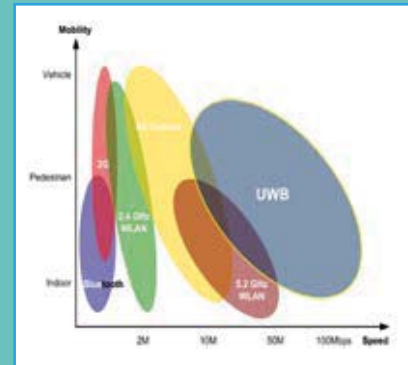
1 gigabit al secondo). Dall'altro lato, i sistemi UWB sono in grado di misurare con estrema precisione il "tempo di transito" delle informazioni: ciò permette di calcolare con uno scarto di centimetri (se non millimetri) quale sia la distanza che separa l'antenna che trasmette il segnale dall'antenna che lo riceve.

L'UWB può operare su tutte le frequenze che vanno dai 3,1 ai 10 gigahertz; una banda, quindi, di circa 7 gigahertz che può essere divisa in canali di almeno 500 megahertz ciascuno, utili per lo scambio veloce e contemporaneo di dati e informazioni. Lavorando anche su frequenze elevate, l'UWB è in grado di passare attraverso muri e altri ostacoli senza grossi problemi. La tecnologia ultra wide band, inoltre, è virtualmente indistinguibile dal cosiddetto "rumore di fondo", caratteristica che la rende particolarmente sicura. Su ogni frequenza utilizzata dall'UWB, in-

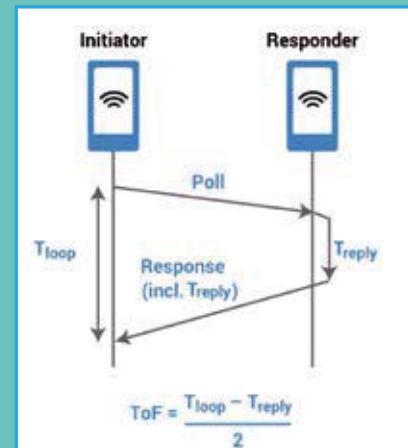
fatti, la potenza del segnale è sempre molto bassa tanto da risultare molto difficile da distinguere da emissioni elettromagnetiche di altro tipo (ad esempio quelle generate dai vari elettrodomestici di casa) e pertanto quasi impossibile da intercettare.

Per quanto riguarda la localizzazione, il concetto operativo è semplice. Quando un dispositivo dotato di una radio UWB come uno smartphone, un braccialetto o una chiave intelligente entra nel raggio di un altro dispositivo UWB, i dispositivi iniziano ad interagire. La distanza viene calcolata eseguendo misurazioni del tempo di volo (ToF) tra i dispositivi. Il TOF viene calcolato misurando il tempo di andata e ritorno dei pacchetti challenge / response. Per conoscere la sua posizione relativa è necessario utilizzare degli ancoraggi UWB fissi e calcolare la sua posizione sulla mappa dell'area.

L'UWB utilizza una larghezza di banda del canale molto grande (500 MHz)



↑ Confronto tra le varie tecnologie radio.



↑ Principio di funzionamento del sistema di localizzazione UWB

con impulsi brevi di circa 2 ns ciascuno; questo aiuta a raggiungere la precisione del centimetro. Il processo di posizionamento UWB avviene in un istante, quindi i movimenti del dispositivo mobile possono essere monitorati in modo molto preciso in tempo reale.



RM ELETTRONICA

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

Distributore autorizzato:

**FUTURA
ELETTRONICA**

Il tuo punto
di riferimento
per l'elettronica
a ROMA

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



Sfoggia on-line i nostri cataloghi

**FUTURA
ELETTRONICA®**



› AI e IIoT secondo Xilinx

Xilinx ha recentemente presentato le sue soluzioni ACAP (Adaptive Compute Acceleration Platform), Industrial IoT (IIoT) e Vitis, la piattaforma software unificata rivolta agli sviluppatori, ingegneri software e scienziati che si occupano di intelligenza artificiale. Vitis è stata utilizzata nella trasmissione video in streaming con larghezza di banda di trasmissione limitata, dove è necessaria un'elaborazione che consenta di codificare la sola regione di interesse con una qualità visiva più elevata rispetto al resto del fotogramma. Utilizzando Vitis AI, la Xilinx Deep Learning Processor Unit (DPU) viene integrata nella pipeline e identifica la maschera del ROI all'interno del frame. Attraverso l'utilizzo delle informazioni della maschera del ROI, la VCU assegna più bit per il ROI rispetto al resto della regione ad un dato bit-rate per mi-



gliorare l'efficienza di codifica. Quanto alla soluzione di inferenza Versal ACAP Machine Learning (ML), utilizza l'Intelligent Engine (AIE) per le applicazioni IIoT edge. ACAP è una piattaforma di calcolo eterogenea completamente programmabile via software che combina motori scalari, motori adattabili e intelligenti per ottenere notevoli miglioramenti delle prestazioni. Xilinx ha anche proposto la demo di una Rete neurale Cloud-trained su dispositivo Xilinx IIoT Edge, ovvero su chip (SoC) integrata da funzioni deterministiche. Dopo che il servizio cloud AWS Sagemaker Neo esegue la formazione per un modello ML e deposita il risultato su cloud storage sicuro, il dispositivo edge basato su Xilinx-SoC lo recupera senza interessare la funzionalità e la capacità di elaborazione real-time.

www.xilinx.com

› L'Internet dei POS

La recente entrata in vigore dell'obbligo di memorizzazione elettronica e trasmissione telematica dei corrispettivi, ha imposto agli esercenti di attività commerciali la necessità di dotarsi di una soluzione di connettività, sicura e dedicata per trasmettere i dati delle vendite all'Agenzia delle Entrate.

Per soddisfare tale esigenza, Vodafone Business, leader mondiale dell'IoT con 94 milioni di oggetti connessi nel mondo, ha ideato per AP.esse (distributrice di terminali POS Axon Mirelec) la connettività wireless cellulare su piattaforma globale IoT di Vodafone (GDSP - Global Data Service Platform), tra le più importanti piattaforme di interconnessione di dispositivi IoT al mondo. Il kit si adatta a qualsiasi registratore di cassa dotato di porta LAN e permette la riconfigurazione da remoto dell'URL dell'Agenzia delle Entrate, in caso di modifiche. Infine, grazie alla



piattaforma GDSP, la soluzione garantisce la massima copertura, sicurezza e usabilità, prevenendo eventuali frodi o utilizzi scorretti della SIM e offrendo

all'esercente la flessibilità necessaria per gestire le sim in autonomia, in risposta alle proprie esigenze.

www.axonmirelec.com

› Il 5G secondo Intel

Intel ha lanciato una serie dispositivi per coprire le esigenze di tutti i nodi della rete del futuro; il primo di questi è l'Intel Atom P5900, un SoC realizzato con processo a 10 nanometri che porterà la potenza dei processori Intel in tutti i punti periferici dell'infrastruttura 5G (edge). Con questi SoC, Intel fornirà la potenza di calcolo necessaria alla gestione della banda offerta dal 5G, oltre che a garantire la bassa latenza promessa dalle reti di nuova generazione.

Oltre che all'IoT edge, Intel ha pensato anche ai core di calcolo dell'infrastruttura di rete, per i quali ha progettato i nuovi processori scalabili Intel Xeon di seconda generazione; questi

processori sono ottimizzati per equipaggiare le reti virtualizzate del futuro, che richiedono capacità computazionale dal lato server. I nuovi Xeon rispondono a tale esigenza con un 36% di prestazioni in più rispetto alla precedente generazione. La terza novità tecnologica di Intel è Diamond Mesa, la soluzione ASIC (Application Specific Integrated Circuit) per il fiorente 5G. Diamond Mesa servirà a integrare i processori e gli FPGA di Intel; sostanzialmente sono dei processori creati per usi specifici, capaci di assicurare più efficienza prestazionale rispetto alle soluzioni convenzionali.

Per quanto riguarda le interfacce di rete, Intel ha messo la

› Antibiotici potentissimi dall'AI

In questo periodo in cui siamo particolarmente sensibili alle infezioni e non solo fisicamente, non possiamo non apprezzare la notizia che arriva dal MIT, dove un algoritmo di machine learning ha aiutato un gruppo di ricercatori a identificare un potentissimo antibiotico chiamato Halicin, in grado di uccidere molti ceppi di batteri, compresi quelli oggi resistenti agli antibiotici tradizionali. Il nome del nuovo antibiotico del Massachusetts Institute of Technology richiama HAL 9000,

l'Intelligenza Artificiale teorizzata nel celebre film di Stanley Kubrick 2001 Odissea nello Spazio. Halicin inibisce la capacità dei batteri di mantenere il gradiente elettrochimico necessario per produrre molecole che immagazzinano energia, portandoli così alla morte: questo meccanismo d'azione è difficilmente contrastabile dai batteri, tant'è che ad esempio il pericolosissimo batterio E. Coli non è riuscito a sviluppare resistenza neppure dopo una sperimentazione di 30 giorni,

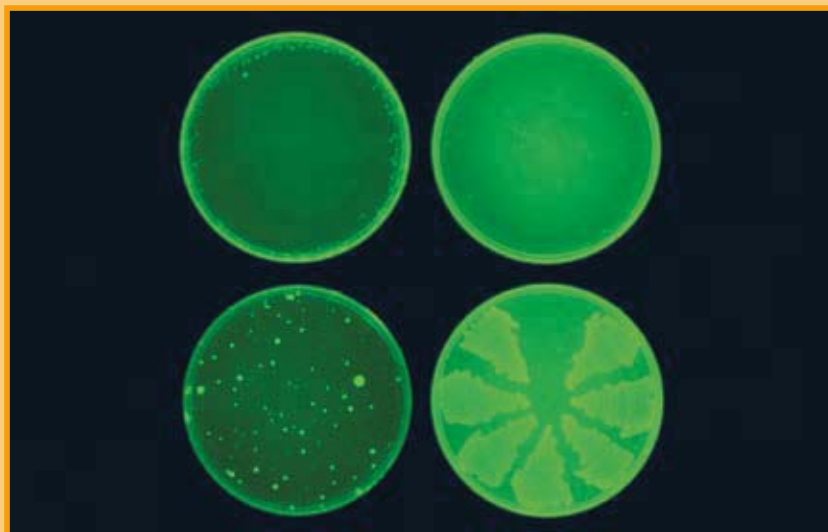
mentre ne bastano appena 3 per contrastare la ciprofloxacina, che è un chinolone e come tale la seconda più recente famiglia di antibiotici.

L'idea di utilizzare modelli computerizzati predittivi non è nuova, ma fino ad ora questi modelli non erano sufficientemente accurati per creare farmaci. In precedenza, le molecole erano rappresentate come vettori che riflettevano la presenza o l'assenza di determinati gruppi chimici, tuttavia le nuove reti neurali possono apprendere automaticamente queste rappresentazioni, mappando le molecole in vettori continui che vengono successivamente utilizzati per prevederne le proprietà.

I ricercatori hanno addestrato il modello di machine learning con circa 2.500 molecole comprendenti sia 1.700 farmaci approvati dalla FDA che 800 prodotti naturali e poi gli hanno fatto analizzare un gruppo di 6.000 composti, al fine di isolare quelli più promettenti.

Ora l'Halicin verrà avviato alla sperimentazione che prima o poi approderà all'uomo. La stessa tecnica con cui è nato potrà in futuro essere applicata per rendere più efficaci antibiotici esistenti o ridurre gli effetti indesiderati.

news.mit.edu



propria esperienza in fatto di networking nella nuovissima Ethernet 700, la prima scheda di rete pensata per il 5G, la prima ad offrire la sincronizzazione del servizio cross-network basata sul sistema GPS con implementazione del Precision Time Protocol (PTP) Hardware-Enhanced. La relativa tecnologia garantirà una precisa sincronizzazione temporale e un allineamento della latenza sui server edge, ovvero su quelli posti nei nodi periferici della rete. Un problema, quello dei requisiti di latenza delle reti 5G, che ha messo in discussione completamente la tecnologia Ethernet esistente e che Intel punta a risolvere con la nuova scheda di rete Ethernet 700.

www.intel.it



› Smart Metering su LoraWAN

Soluzioni innovative di sub-metering basate sulla tecnologia LoRaWAN per un nuovo modello di business nell'ambito del monitoraggio energetico. È questo l'obiettivo del Memorandum of Understanding (MoU) sottoscritto fra Unidata e Zenner, a conferma della forte volontà di collaborazione strategica tra le due società, per lo sviluppo di nuove soluzioni IoT dedicate al mercato italiano ed in particolare applicate al comparto energetico.

Facendo leva sulle rispettive aree di competenza e su oltre 30 anni di esperienza nei settori delle telecomunicazioni e della produzione di contatori per la misura dei consumi di acqua ed energia, Unidata e Zenner avviano quindi una collaborazione che prevede lo sviluppo di soluzioni innovative di sub-metering basate sulla tecnologia LoRaWAN da implementare inizialmente nella città di Roma, con l'obiettivo di applicare un nuovo modello di business nell'ambito dello smart-metering in real-time.

Unidata, operatore di telecomunicazioni con una propria rete e propri data-center, offre soluzioni IoT innovative basate sulla propria rete IoT UniWAN e sulla propria piattaforma di



connettività UniOrchestra. Il segmento IoT (Internet of Things) per la fornitura, tra gli altri, di servizi di Domotica e Smart City, è attualmente in fase di consolidamento e si fonda sulla trasmissione di dati a lunga distanza con bassi consumi tramite la propria rete IoT UniWAN e la propria piattaforma di connettività UniOrchestra.

Zenner sviluppa e produce strumenti di misura, soluzioni e servizi di smart-metering rivolti principalmente alle

utility; i tradizionali contatori di acqua e calore meccanici a lettura diretta si sono trasformati nel tempo utilizzando le più moderne tecnologie di misurazione e trasmissione dati, quali ultrasuoni e LoRaWAN.

I nuovi prodotti di ZENNER, infatti, sono parte integrante delle soluzioni IoT utilizzate nelle Smart City e negli Smart Building e si integreranno perfettamente nell'ecosistema Unidata.

www.unidata.it

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



Interruttore Wi-Fi

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

€7,90

Cod. FR775

Installabile in scatole
di derivazione
o porta frutto



€19,90

Cod. FR776



Micro interruttore Wi-Fi a 1 canale

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

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



Prezzi IVA inclusa.

€24,90

Cod. FR778



Telecomando Wi-Fi, IR

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

Presa smart Wi-Fi

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



€21,90
Cod. FR779

MAX32520, MCU con tecnologia ChipDNA che non consente la clonabilità fisica (PUF)

Maxim Integrated ha introdotto il microcontrollore Arm Cortex-M4 ChipDNA MAX32520, il primo microcontrollore sicuro, con la tecnologia integrata che non permette la clonabilità fisica (PUF), dedicato alla sicurezza finanziaria e di livello governativo. La tecnologia PUF di Maxim consente molteplici livelli di protezione per fornire la più avanzata tecnologia di protezione delle chiavi ad un costo ac-



cessibile per l'uso in sistemi IoT, sanitari, industriali e informatici. Il MAX32520 con

ChipDNA offre molteplici livelli di protezione attraverso la sua tecnologia PUF, la tecnologia di protezione delle chiavi più avanzata del settore, con lo scopo di salvaguardare i segreti utilizzati nelle operazioni crittografiche. Utilizza una chiave PUF a prova di manomissione per la crittografia flash, l'avvio sicuro per il "root-of-trust" e l'emulazione flash seriale.

www.maximintegrated.com

Marvell e Analog Devices annunciano una collaborazione per soluzioni radio 5G

Marvell e Analog Devices hanno annunciato una collaborazione tecnologica che sfrutta la piattaforma digitale 5G di Marvell, all'avanguardia nel settore, e la tecnologia dei transceiver RF a banda larga di ADI per fornire soluzioni completamente ottimizzate per le stazioni base 5G. Nell'ambito della collaborazione, le aziende offriranno soluzioni ASIC 5G digital front-end (DFE) completamente integrate con transceiver RF strettamente accoppiati e collaboreranno per sviluppare soluzioni Radio Unit (RU) di nuova generazione che includono la

tecnologia in banda base e RF ottimizzata per un'ampia serie di split funzionali e di architetture.

La combinazione delle tecnologie di Marvell e ADI, le migliori della categoria, consente progetti di RU altamente ottimizzati che consentono implementazioni sia standard che personalizzate.

www.analog.com



Arm lancia il Machine Learning on-device

Arm ha annunciato importanti aggiunte alla sua piattaforma di intelligenza artificiale (AI), tra cui il nuovo IP per machine learning (ML), il processore Arm Cortex-M55 e il dispositivo Arm Ethos-U55 NPU, la prima microNPU (Neural Processing Unit) del settore per Cortex-M, progettata per offrire prestazioni superiori di 480 volte in ambito Machine Learning per microcontrollori.

Il nuovo IP e il supporto della toolchain consentono agli sviluppatori hardware e software di Intelligenza Artificiale di avere più modi di innovare grazie a livelli senza precedenti di elaborazione ML su dispositivo, per miliardi di piccoli dispositivi em-

bedded IoT con potenza limitata. Mentre l'IoT si interseca con i progressi dell'IA e il lancio del 5G, una maggiore intelligenza su chip significa che i dispositivi più piccoli e sensibili ai costi possono essere più intelligenti e più capaci, beneficiando di una maggiore privacy e affidabilità grazie alla minore dipendenza dal cloud o da Internet.

www.arm.com



Da NXP SoC Bluetooth 5 con periferica NFC

NXP Semiconductors ha annunciato la disponibilità dei suoi sistemi QN9090 e QN9030 Bluetooth 5 System on Chip (SoC) con opzioni hardware compatibili con IEEE 802.15.4, ovvero RF multiprotocollo e tecnologia NFC. Le ultime aggiunte alla sua serie QN di dispositivi Bluetooth Low Energy (BLE) consentono di realizzare dispositivi connessi intelligenti di prossima

generazione con un consumo energetico bassissimo e con in più una CPU ad elevata capacità con un ampio intervallo di



temperature di funzionamento, un mix completo di periferiche analogiche e digitali e supporto per mesh BLE.

I dispositivi QN9090 e QN9030 sono alimentati da un Arm Cortex-M4 a 48 MHz e includono fino a 640 KB di Flash e 152 KB SRAM, offrendo spazio di archiviazione e flessibilità per applicazioni complesse e aggiornamenti over-the-air (OTA) sicuri. I dispositivi della

serie QN aiutano ad accelerare lo sviluppo e il time-to-market per gli sviluppatori che creano prodotti con funzionalità avanzate per diverse applicazioni IoT come dispositivi sanitari personali, tracker sportivi e di fitness, elettrodomestici collegati, automazione di edifici e casa, giocattoli e periferiche di gioco, nonché come beacon e reti mesh.

www.nxp.com



Densità di potenza da record per la nuova serie di CoolMOS

Basandosi sulle innovazioni tecnologiche nel campo della supergiunzione, oltre che su più di 20 anni di esperienza, Infineon Technologies amplia il suo portafoglio CoolMOS con la famiglia di prodotti PFD7 che unisce le prestazioni migliori della categoria con la più grande facilità

d'uso. I dispositivi sono adatti per progetti ad alta densità di potenza come cariche batterie e adattatori, nonché per unità a bassa potenza e applicazioni di illuminazione.

L'aumento della robustezza, dell'affidabilità e dell'efficienza di commutazione, insieme al mi-

glioramento del comportamento termico, rendono questa famiglia di prodotti la scelta definitiva per gli attuali progetti. La serie CoolMOS PFD7 supporta le tendenze chiave dei prodotti mobili, piccoli e leggeri, e il risparmio energetico nei principali elettrodomestici.

www.infineon.com

PIC18-Q43 di Microchip con ancora più periferiche indipendenti (CIP)

Nella progettazione di sistemi basati su microcontrollori (MCU), il software rappresenta spesso il collo di bottiglia sia per il tempo necessario allo sviluppo, sia per le prestazioni che si riescono ad ottenere. Scaricando molte attività software sull'hardware, la famiglia PIC18-Q43 di nuova generazione di Microchip aiuta gli sviluppatori a introdurre più rapidamente sul mercato soluzioni con prestazioni più elevate.

La combinazione di periferiche della famiglia offre agli utenti una maggiore versatilità e semplicità nella creazione di funzioni personalizzate basate su hardware, con strumenti di sviluppo facili da usare. Le periferiche configurabili sono opportunamente interconnesse per consentire una condivisione dei dati a latenza quasi nulla, ingressi logici o segnali analogici senza codice aggiunti-



vo, per una migliore risposta del sistema. Ideale per una varietà di controlli in tempo reale e applicazioni connesse, inclusi elettrodomestici, sistemi di sicurezza, controllo motore e industriale, illuminazione e Internet of Things (IoT), la famiglia PIC18-Q43 aiuta a ridurre lo spazio su scheda, il Bill of Materials (BoM) ed accelera il time to market.

La famiglia funziona fino a 5 V, il che aumenta l'immunità al rumore e consente ai clienti di interfacciarsi con una vasta gamma di sensori.

www.microchip.com

G3VM-21MT, relè MOSFET di Omron riduce la dispersione di corrente

In un mondo in cui i semiconduttori e altri sistemi di collaudo la fanno da padrone, Omron Electronics Components Europe ha creato un modulo basato su relè MOSFET in grado di minimizzare la dispersione di corrente.

Con una innovativa struttura del circuito a T, il G3VM-21MT Omron offre un livello di dispersione di corrente eccezionalmente basso, pari a 1pA (pico-Amp) o meno, assicurando così misurazioni accurate in tutti i tipi di apparecchiature di test. In precedenza i relè meccanici erano preferiti nelle apparecchiature di collaudo, sia a semiconduttore sia di altro tipo, in parte proprio per il loro basso livello di dispersione di corrente. Per contro, i relè meccanici hanno una durata di vita di gran lunga inferiore

re a causa dell'abrasione dei contatti che nel tempo riduce l'accuratezza delle misurazioni. Negli utilizzi più intensivi, potevano essere necessarie sostituzioni frequenti, alzando i costi di manutenzione.

Con una struttura di circuito a T che trasmette la maggior parte della dispersione a terra, il G3VM-21MT combina efficacemente i vantaggi dei relè meccanici e MOSFET per fornire una soluzione di commutazione accurata, compatta e duratura, senza contatti meccanici.

components.omron.eu



Nuova serie di alimentatori NGP800 quadcore da Rohde & Schwarz

Le serie di alimentatori R&S NGP800 aumenta l'efficienza in qualsiasi laboratorio o linea di produzione che esegue test e mi-



sure che richiedono fino a quattro alimentatori DC indipendenti in grado di garantire piena flessibilità, piena funzionalità, sicurezza e connettività. Gli utenti di questi prodotti beneficiano dell'ampio touchscreen da 5" ad alta risoluzione, che mostra anche statistiche dettagliate. I cinque modelli R&S NGP800

soddisfano le esigenze degli utenti che cercano dimensioni ridotte per un alimentatore a due o quattro canali indipendenti, con la massima flessibilità per tensioni fino a 250 V, correnti fino a 80 A e potenza fino a 800 W.

Ogni canale fornisce fino a 200 W con un massimo di 20 A o 64 V, coprendo anche i 48 V utilizzati

in applicazioni automobilistiche e industriali. I canali possono essere gestiti in modo completamente indipendente o sincronizzato; le uscite elettricamente equivalenti e galvanicamente isolate possono essere cablate in serie per uscite fino a 250 V o in parallelo fino a 80 A.

www.rohde-schwarz.com

Nuova piattaforma LoRa Edge di Semtech con singolo chip LR1110

LoRa Edge è una nuova piattaforma basata su LoRa definita da software altamente versatile e a basso consumo che abiliterà numerose applicazioni in ambito sicurezza, industria, edilizia, casa, agricoltura, dei trasporti e della logistica. La piattaforma LoRa Edge viene proposta con una soluzione di geolocalizzazione mirata alle applicazioni IoT per la gestione delle risorse. È dotato di un ricetrasmittitore LoRa a bassissima potenza, funzionalità di sniffing GNSS e tecnologie di scansione Wi-Fi: il tutto nel piccolissimo dispositivo LR1110. Il chip Semtech LR1110 è simile ai normali dispositivi ricetrasmittitore LoRa (completa compatibilità con SX1261/2/8) con supporto di bande di frequenza in tutto il mondo nell'intervallo 150 – 960 MHz, ma con



l'aggiunta delle funzionalità di geolocalizzazione Wi-Fi e GNSS. È dotato di scansione GNSS (GPS / BeiDou) a bassa potenza e scansione passiva a bassissima potenza Wi-Fi 802.11b / g / n. Supporta le modulazioni LoRa e (G) FSK ed è pienamente compatibile con le generazioni precedenti di radio LoRa. Può trasmettere fino a +22dBm in alta potenza o fino a +15dBm in bassa potenza con funzionamento nelle bande ISM 150 MHz – 960 MHz. Il dispositivo viene fornito in un package QFN32 da 5 x 5 mm. www.semtech.com

STSAFE-A110 di STMicroelectronics, a protezione dell'autenticità

STSAFE-A110 offre protezione certificata all'avanguardia e accesso al caricamento sicuro delle credenziali cloud per fornire la registrazione di massa dei dispositivi IoT e garantire che solo i dispositivi autorizzati possano accedere ai servizi online. Questa personalizzazione critica e sicura può essere eseguita in fase di costruzione da ST, senza costi aggiuntivi per quantità annuali superiori a 5.000 unità, semplificando e proteggendo drasticamente la manipolazione di dati segreti durante la produzione di dispositivi IoT. Il dispositivo presenta un sistema operativo embedded ed è basato su hardware certificato secondo Common Criteria



Assessment Assurance Level 5+ (EAL5 +). Ogni unità è dotata di un identificativo univoco e certificati X.509 che aiutano la connessione sicura del dispositivo. Inoltre, questo dispositivo è integrato con l'ecosistema di sviluppo STM32Cube per essere rapidamente incorporato nei nuovi progetti STM32 che richiedono una funzionalità di autenticazione e connessione sicura. www.st.com

Toshiba annuncia tre nuovi fotorelè ad elevata corrente

Toshiba Electronics Europe ha annunciato la disponibilità di tre nuovi fotorelè destinati ad applicazioni nel campo dell'automazione industriale e nelle installazioni per l'automazione di fabbrica, in sostituzione dei relè meccanici di tipo 1-Form-A.

I dispositivi sono basati sull'ultimo processo su semiconduttore U-MOS IX, il quale riduce la resistenza di on, consentendo ai fotorelè TLP3106A, TLP3107A e TLP3109A di offrire valori di RON(max) rispettivamente di 30mΩ, 40mΩ e 65mΩ. I nuovi fotorelè offrono una gamma di opzioni, che includono una tensione terminale di uscita nello stato di OFF (VOFF) compresa tra 30V e

100V e una corrente stazionaria nello stato di ON (ION) tra 3A e 4,5A. I valori di corrente impulsata nello stato di ON sono compresi fra 9A e 13,5A.

Tutti e tre i dispositivi offrono una tensione di isolamento (BVS min) di 1500 Vrms e operano su un intervallo di temperature compreso fra -40°C e +110°C.

www.toshiba.semicon-storage.com



ISO7741E-Q1, primo isolatore digitale di Grado 0 del settore automotive

Texas Instruments ha presentato il primo isolatore digitale del settore con qualifica per la specifica di temperatura ambiente d'esercizio di Grado 0 dello standard AEC-Q100. L'ISO7741E-Q1 presenta una tensione d'esercizio di 1,5 kVRMS, leader del settore, e supporta temperature fino al massimo del Grado 0 pari a 150 °C. Il nuovo isolatore permette agli ingegneri di proteggere meglio i circuiti a bassa tensione dagli eventi di alta tensione nei sistemi dei veicoli elettrici ibridi (HEV) e dei veicoli elettrici (EV), evitando la necessità di progettare sistemi di raffreddamento per ridurre le temperature al di sotto dei 125 °C, ossia il massimo sostenibile dai circuiti integrati (IC) con qualifica di Grado 1. Inoltre, implementando le comunicazioni CAN FD (Controller Area Network Flexible Data Rate) nelle



progettazione dei sistemi, gli ingegneri possono aumentare la protezione e la copertura del segnale nella rete di bordo del veicolo (IVN, in-vehicle network) utilizzando l'ISO7741E-Q1 con il nuovo transceiver TCAN1044EV-Q1 CAN FD di Grado 0.

I circuiti integrati con qualifica di Grado 0 soddisfano i requisiti del più alto grado di temperatura dell'AEC-Q100 (da -40 °C a 150 °C) e sono progettati per aiutare gli ingegneri a semplificare la progettazione di sistemi HEV/EV in ambienti gravosi.

www.ti.com

Sensori per tutte le esigenze!



Kit di 37 moduli con sensori per Arduino, Raspberry, robotica ecc.

€ 29,90

Cod. **SENSORKIT37**

Modulo con igrometro per terreno



€ 11,50

Cod. **U20SMS**

**VASTA SCELTA DI SENSORI
DI ULTIMA GENERAZIONE**

**PER REALIZZARE I TUOI PROGETTI,
ANCHE QUELLI PIÙ COMPLESSI.**

Interruttore fotoelettrico con LASER a luce visibile



€ 28,00

Cod. **VLPSS**

Sensore di movimento a microonde amplificato



€ 14,00

Cod. **HB100AQ**

Sensore qualità dell'aria per particolato PM2.5, PM10



€ 45,00

Cod. **PMSENSOR**

Trasduttore di pressione 0-1,2 MPa



€ 22,00

Cod. **PTS0161**

Modulo con sensore per rilevazione PH e temperatura



€ 29,50

Cod. **YB427**

Sonda per la misurazione del PH



€ 13,50

Cod. **PHSENSOR**

Sensore di Flessione 2,2"



€ 15,50

Cod. **FLEXSENSOR2**

Sensore analogico per misurare la torbidità dell'acqua



€ 14,50

Cod. **TORBISENS**

Sensore fotoelettrico



€ 7,50

Cod. **IRFOTOSENS**

Sensore di forza



€ 9,90

Cod. **SENSOREFORZA**

Prezzi IVA inclusa.

EXPOMOVE2020

La seconda edizione di EXPOMOVE – Mobilità Elettrica e Sostenibile sembra aver imboccato la via giusta per un'altra edizione di successo.

Oltre alle conferme di partner e patrocinii dello scorso anno, si registrano nuove adesioni da parte di molti e, il forte interesse a partecipare, è ribadito dalle tante manifestazioni di interesse di nuovi potenziali espositori. Sono più di trenta le aziende che hanno deciso di assicurarsi uno spazio in quella che sarà la fiera italiana di riferimento per la mobilità elettrica. Un evento ad ampio raggio che non vuole limitarsi ad una banale esposizione di mezzi a trazione elettrica: la mission di EXPOMOVE2020 è di trasmettere e far capire, ai propri interlocutori, la necessità di questo cambiamento epocale, promuovendo la nuova era della mobilità e analizzando un insieme di fattori che risultano essere necessari, focalizzati su un'idea di sviluppo più estesa che raccolga diversi concetti, a partire dal fondamentale cambiamento infrastrutturale delle città.

A questo proposito, saranno 5 le macroaree che caratterizzeranno EXPOMOVE2020: Sharing Mobility, Mobilità Pubblica e Merci, Mobilità Privata, Mobilità Leggera ed Ultimo Miglio e Componentistica e Tecnologia.

Questa differenziazione permetterà ad ogni espositore di mostrare i propri mezzi, le proprie ricerche, le proprie infrastrutture e i propri componenti innovativi in un contesto più congruo: ciò che sarà esposto ad EXPOMOVE2020, sarà necessario ed essenziale per permetterci di entrare con il piede giusto in un futuro che è sempre più prossimo. EXPOMOVE2020 vuole dare la giusta rilevanza ad ognuno di questi settori che sono strettamente dipendenti l'uno dall'altro.

Dal 6 maggio, con l'evento d'apertura che si terrà nel Salone dei Cinquecento di Palazzo Vecchio. La fase espositiva, con oltre 9.000 mq, proseguirà nella Fortezza da Basso dal 7 al 9 maggio 2020.

www.expomove.it



LE SOLUZIONI SAVE L'8 APRILE 2020 IN MOSTRA ALLA FIERA DI BERGAMO

SAVE Milano, l'evento verticale di riferimento per le soluzioni e applicazioni di Strumentazione, Automazione e Sensoristica prosegue il proprio percorso di crescita e si sposta quest'anno alla Fiera di Bergamo, così da beneficiare dell'indotto industriale del territorio e ampliare gli spazi a disposizione dei partecipanti.

La presenza dell'aeroporto adiacente alla fiera, l'Innovation District Kilometro Rosso a pochi passi, e la posizione di Bergamo naturalmente contigua a Milano e a Brescia, oltre che strategica

per il Nord Italia, sono tra i fattori che influiranno positivamente sulla scelta quale migliore spazio per poter supportare lo sviluppo dell'evento. SAVE Milano rimane la vetrina primaverile per le aziende e gli operatori qualificati che desiderano incontrarsi, aggiornarsi (grazie ai tanti momenti formativi), trovare nuovi spunti e fare business insieme, e in questa edizione amplia i propri orizzonti all'intera area metropolitana milanese.

La nuova giornata sarà dedicata all'innovazione e come sempre caratterizzata da un ricco programma, a cominciare dall'ampio ventaglio di convegni con il coinvolgimento delle più importanti associazioni del settore, dei maggiori esperti e influencer.

Strumentazione Smart, Big Data, sensoristica, test & measurement, condition monitoring, asset management, sistemi di visione, efficienza energetica, raccolta dati e analisi e molto altro ancora, sono le tematiche attorno a cui si svilupperà l'appuntamento.

www.exposave.com

FORLÌ

EXPO ELETTRONICA

Quartiere Fieristico – Forlì
Organizzazione: Blu Nautilus srl
Telefono: 054153295
www.expoelettronica.it
info@expoelettronica.it
02 e 03 Maggio 2020

MARIANO COMENSE (CO)

FIERAMERCATO DELL'ELETTRONICA

Palatenda - Via Don Sturzo
Organizzazione: Eventi e Fiere
Telefono: 0362999186
www.eventiefiere.com
info@eventiefiere.com
02 e 03 Maggio 2020

BUSTO ARSIZIO (VA)

EXPO ELETTRONICA

Malpensafiere - Busto Arsizio VA
Organizzazione: Blu Nautilus srl
Telefono: 0541439573
www.expoelettronica.it
info@expoelettronica.it
09 e 10 Maggio 2020

GONZAGA (MN)

FIERA DELL'ELETTRONICA

Fiera Millenaria - Gonzaga (MN)
Organizzazione: Fiera Millenaria di Gonzaga Srl
Telefono: 037658098
www.fieramillenaria.it
info@fieramillenaria.it
23 e 24 Maggio 2020

REGGIO EMILIA

FIERA ELETTRONICA + FUMETTI

Fiera di Reggio Emilia - Via Filangeri
Organizzazione: Expo Fiere Srl
Telefono: 054527548
www.mondoelettronica.net
info@expofiere.net
23 e 24 Maggio 2020

CODEVILLA (PV)

FIERA DELL'ELETTRONICA

Expo Tucano SP1 N° 38 Codevilla
Organizzazione: Eventi e Fiere
www.eventiefiere.com
info@eventiefiere.com
23 e 24 Maggio 2020

AMELIA (TR)

MOSTRA MERCATO NAZIONALE AMELIA EXPO

Loc. Molino Silla - Strada Orte - Amelia
Organizzazione: Consulting Service Srl
Telefono: 3384755526
www.mostremercatoumbria.it
venturag@alice.it
30 e 31 Maggio 2020

L'elenco aggiornato di tutte le Mostre Mercato del 2020 è disponibile sul sito www.elettronica.in.it sul quale è anche possibile scrivere un commento sulle fiere visitate.

L'elenco completo dei più importanti eventi nazionali e internazionali di elettronica, sicurezza e fonti rinnovabili è disponibile sul nostro sito: http://www.elettronica.in.it/myMain/Fiere/p_fiere.asp

FOCUS SULL'ELETTRONICA DI POTENZA A PCIM EUROPE 2020

Negli ultimi anni il settore dell'elettronica di potenza è stato tra quelli col numero più significativo di innovazioni tecnologiche, specie dopo l'introduzione dei semiconduttori di tipo wide band gap. Con la corsa all'elettrificazione, poi, questo settore ha visto crescere in maniera esponenziale il numero delle applicazioni. A tutti questi temi, tecnologie dell'elettronica di potenza e relative applicazioni, è dedicato PCIM Europe 2020 la più importante mostra convegno al mondo dedicata a questa materia, in programma dal 5 al 7 maggio 2020 a Norimberga. Si prevede che saranno circa 500 le aziende espositrici provenienti da 30 paesi per presentare un'ampia gamma di prodotti e soluzioni, dai componenti ai sistemi intelligenti lungo l'intera catena del valore.



Nuove tendenze e sviluppi saranno presentati al pubblico per la prima volta nel punto d'incontro per esperti, dove i visitatori professionali potranno approfondire la conoscenza delle innovazioni di prodotto, anche con uno scambio personale di informazioni con i produttori, a prescindere dal fatto che provengano dal settore dei sistemi di conversione elettronica di potenza, sensori, semiconduttori, componenti passivi o produzione e accumulo di energia. Per coloro che sono interessati all'elettronica di potenza per la mobilità elettrica, proprio l'area destinata alla mobilità elettrica è il primo posto da visitare. Qui, numerosi produttori forniranno informazioni sui prodotti e componenti più recenti per questa area di applicazione. È in programma anche uno speciale forum sulla mobilità

INTELLIGENZA ARTIFICIALE, APPRENDIMENTO AUTOMATICO E MANUTENZIONE PREDITTIVA: I TEMI CALDI DI HANNOVER MESSE

La manutenzione predittiva è uno dei temi principali della prossima HANNOVER MESSE 2020 in programma ad Hannover dal 20 al 24 Aprile. Molti degli espositori della fiera sono fornitori di soluzioni software basate sull'intelligenza artificiale che prevedono i guasti e prevengono costosi arresti non pianificati negli impianti e nei sistemi di produzione connessi. "Ogni minuto di ogni ora in cui una linea di produzione o una struttura viene chiusa a causa di problemi tecnici è un onere di costo maggiore. È qui che entra in gioco la manutenzione predittiva basata sull'intelligenza artificiale", ha spiegato **Hubertus von Monschaw**, direttore globale Digi-

tal Ecosystems di Deutsche Messe per HANNOVER MESSE. "Se le macchine in questione sono in grado di comunicare tra loro e scambiare dati tra loro, i potenziali problemi possono essere rilevati in modo relativamente rapido eseguendo un software di base basato sull'intelligenza artificiale in grado di identificare anomalie e altri guasti normalmente imprevedibili. La chiave è che i problemi e le anomalie siano rilevati il più presto possibile per evitare interruzioni". Ma non è tutto. La manutenzione predittiva è anche un modo molto utile per interconnettere le catene del valore digitali. Una volta che il sistema di manutenzione predittiva conosce il punto esatto in futuro in

elettrica con interessanti presentazioni tecniche da scoprire in tutti e tre i giorni della mostra. Ulteriore attenzione sarà posta al tema riguardante lo stoccaggio di energia. Questo argomento è incluso sia nel programma di lezioni del Forum dell'Industria che in quello della conferenza orientata all'applicazione. Quale che sia il sistema di accumulo dell'energia stazionaria, l'elettronica di potenza gioca un ruolo centrale nella soluzione, sia in fase di accumulo che durante la generazione di energia elettrica.

Grazie ai tre Forum, i visitatori potranno approfondire le proprie conoscenze tecniche. Oltre all'E-Mobility Forum, il Forum degli Espositori offre approfondimenti esclusivi sulle ultime innovazioni di prodotto. All'ordine del giorno del Forum Industriale presentazioni e panel su argomenti quali "EV/HEV Cost vs. Performance Trade off: A Battle on Multiple Fronts for Power Electronics" (Yole Développement), "SiC- and GaN-Devices: The Game Changers" (Bodo's Power Systems) e "Next Step for the Greener Tomorrow. Second Generation of SiC Power Modules" (Mitsubishi Electric Corporation). Tutti i relatori del

forum Industry saranno disponibili a rispondere alle domande al termine delle presentazioni durante le sessioni "Meet the Speaker".

Parallelamente alla mostra, si svolgerà la conferenza internazionale, in cui i

partecipanti potranno apprendere i più recenti risultati nel campo della ricerca dell'elettronica di potenza in circa 300 lezioni e presentazioni tecniche.

www.pcim-europe.com

cui una determinata macchina dovrà essere spenta per manutenzione, ad esempio, potrà avviare automaticamente i processi logistici associati. Ciò garantisce che tutti i processi di ordinazione delle parti e del lavoro pertinenti siano adeguatamente coordinati. Un buon esempio di ciò è Datawatch, che Altair Engineering presenterà nel padiglione 17. Datawatch può essere integrato nell'infrastruttura esistente e si occupa della manutenzione predittiva in tre fasi: preparazione dei dati, generazione di modelli di dati e previsione in tempo reale. La piattaforma riunisce utenti con diverse competenze

e background in differenti discipline. Un altro esempio è Data Lighthouse, una società di sviluppo software con sede ad Amburgo che presenterà una serie di innovazioni nella Hall 16. Data Lighthouse mostrerà come le aziende industriali possono sfruttare la sua soluzione Data Grid basata su cloud per monitorare i loro impianti di produzione in tempo reale e prevedere il loro comportamento. La soluzione utilizza il gemellaggio digitale e dispositivi mobili per consentire agli utenti di monitorare continuamente le condizioni dei loro impianti di produzione. Ultimo ma non meno importante, la società di consulenza e software Scheer con sede a Saarbrücken presenterà una serie di soluzioni basate sull'intelligenza artificiale nel padiglione 17. Tra queste ci sarà Predictive Intelligence, un'applicazione software basata su algoritmi di autoapprendimento. L'applicazione consente agli utenti di ridurre gli scarti di produzione, pianificare in anticipo i cicli di manutenzione e ridurre il consumo di energia dell'impianto.

In un senso molto reale, la manuten-

zione predittiva va di pari passo con l'apprendimento automatico perché, al fine di poter rilevare in modo affidabile potenziali guasti, un sistema di manutenzione predittiva deve avere una conoscenza preliminare di tutte le possibili situazioni di guasto. "Insegnando" al sistema di manutenzione tutte le varie possibili situazioni di guasto, il software di machine learning basato su AI consente di rilevare guasti specifici in tempo reale e avviare azioni correttive. HANNOVER MESSE 2020 si concentrerà quindi molto sull'apprendimento automatico.

A1 Digital Deutschland, ad esempio, utilizzerà il suo stand nel padiglione 15 per definire la sua offerta come partner di scelta per i progetti di intelligenza artificiale. I visitatori dello stand saranno in grado di discutere le loro applicazioni pianificate con gli specialisti di intelligenza artificiale di A1 Digital e ottenere una migliore comprensione delle loro esigenze di apprendimento automatico e intelligenza artificiale.

www.hannovermesse.de/en





ABBONAMENTO
ANNUALE



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



Visita: www.elettronica.in/abbonamenti



ABBONAMENTO ANNUALE PER RIVISTA DIGITALE

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

COME ABBONARSI

→ **on-line**

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

→ **e-mail**

scrivendo a
abbonati@elettronica.in

→ **telefono**

telefonando a
+39-0331-752668

ABBONAMENTO ANNUALE PER RIVISTA CARTACEA

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

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

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

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



PROGRAMMA E PROGETTA CON ARDUINO

Per chi vuole avvicinarsi al mondo della programmazione e della progettazione con Arduino, tante board studiate per imparare facilmente ad utilizzare il microcomputer più diffuso al mondo.

ESP32 - SCHEDA DI SVILUPPO WiFi E BLUETOOTH

€ 13,⁵⁰
cod. YB555



Board di sviluppo e prototipazione compatta basata sull'ESP32 (ESP-WROOM-32); integra Wi-Fi 802.11 b/g/n, Bluetooth dual-mode (classico e BLE) e 34 GPIO. Programmabile con IDE di Arduino.

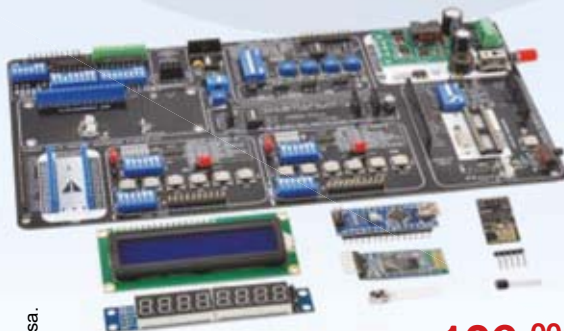
BOARD ATTINY85 USB

€ 5,⁵⁰
cod. ATTINY85USB



Scheda basata sul microcontrollore ATtiny85. Fornita con bootloader, è programmabile tramite IDE di Arduino, dispone di 6 ingressi/uscite (3 con funzione PWM e 4 come ingressi analogici), clock rate massimo 20 MHz (default 16 MHz).

DEMOBOARD PER ARDUINO CON MODULO ATMEGA328PU NANO COMPATIBILE

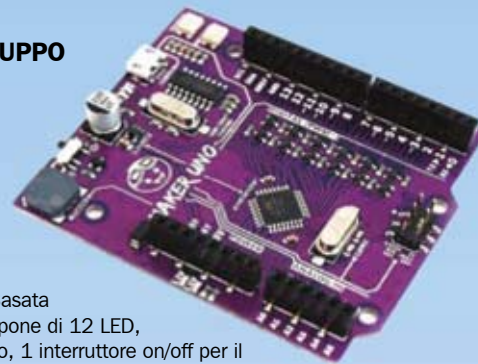


€ 106,⁰⁰
cod. ARDUINODEMO

SCHEDA DI SVILUPPO MAKER UNO

€ 12,⁵⁰
cod. MAKERUNO

Scheda di sviluppo compatibile con Arduino UNO Rev3. Basata sull'ATmega328P, dispone di 12 LED, 1 buzzer piezoelettrico, 1 interruttore on/off per il buzzer e 1 pulsante programmabile. Fornisce ai dispositivi funzionanti a 3,3 V una corrente di ben 500 mA rispetto ai 50 mA di Arduino Uno. Nella confezione sono compresi anche degli adesivi che permettono di identificare immediatamente i pin.



Disponibile anche
nella confezione con
cavo USB-micro USB.

€ 19,⁹⁰
cod. MAKERUNOPLUS



ROUND BOARD ATMEGA328P



€ 5,⁵⁰
cod. ROUNDPAD328

Scheda basata sul microcontrollore ATmega328. Compatibile con Arduino Uno, dispone di 14 ingressi/uscite digitali, 8 ingressi analogici, oscillatore a 16 MHz e connettore a 6 pin da collegare ad un convertitore USB-seriale per la programmazione. Fornita con un bootloader, che permette di caricare qualsiasi nuovo codice applicabile ad Arduino.

ANTENNINO - SCHEDA MULTIRUOLO ARDUINO-RF

Board basata sul chip Atmel, quindi compatibile al 100% con l'IDE Arduino. La board integra sensori e bus di comunicazione per acquisire informazioni esterne, nonché un link wireless per comunicare con l'esterno. A seconda di come viene equipaggiata, può funzionare sia da modulo remoto che da collettore di informazioni (gateway).

€ 39,⁰⁰
cod. ANTENNINO



Prezzi IVA inclusa.

SCOPRI TUTTE LE BOARD SU WWW.FUTURASHOP.IT

OBD QUESTO CONOSCIUTO

di **ALESSANDRO SOTTOCORNOLA**



Tutto quello che c'è da sapere sulla porta di diagnosi dei moderni autoveicoli e sui dispositivi che vi si possono interfacciare. Prima puntata.

La tecnologia elettronica applicata ai veicoli che circolano su strada e in generale ai veicoli a motore endotermico è ormai arrivata a livelli molto elevati e la presenza di sistemi elettronici che interagiscono, specialmente col motore, è ormai massiva. Questo ha portato i tecnici che operano nel settore, ma anche i semplici appassionati, a imbattersi in un fiume di termini spesso sconosciuti e difficilmente decifrabili, oltre che in sistemi che inaspettatamente si trovano aprendo il cofano dei veicoli. Per fare un po' di chiarezza sulla terminologia in uso e specialmente sulla porta di diagnosi che oggi consente di affacciarsi sul mondo dell'elettronica di bordo, abbiamo deciso di pubblicare questo articolo, dove, partendo proprio dall'in-

terfaccia di diagnosi (OBD e sue evoluzioni) descriveremo lo stato dell'arte, per poi fare una rassegna sui principali software e app che permettono di interagire con l'elettronica di bordo attraverso un cavo o un'interfaccia wireless connessa alla presa di diagnosi.

OBD QUANDO NASCE E PERCHÉ

L'introduzione dell'interfaccia diagnostica (OBD=On Board Diagnostic) è stata determinata sostanzialmente dalla necessità di verificare lo stato dei sistemi di abbattimento delle emissioni inquinanti delle auto equipaggiate con motore a combustione interna. Nel 1966, al fine di combattere il crescente smog in aumento nelle aree dell'America, lo stato della California cominciò a richiedere l'installazione a bordo dei veicoli di sistemi che permettessero il controllo delle emissioni; due anni dopo il governo degli Stati Uniti estese questi controlli su tutto il territorio nazionale. Nel 1970 il congresso statunitense approvò il **"Clean Air Act"**, fondando contestualmente l'organo di controllo **EPA** (Environmental Protection Agency). Negli anni seguenti l'EPA definì una serie di standard con l'obiettivo di ridurre gradualmente le emissioni dei gas di scarico. Per uniformarsi a questi standard, le case produttrici cominciarono a introdurre sui veicoli di nuova fabbricazione dei sistemi in grado di controllare l'alimentazione, governati dall'elettronica; appositi sensori misuravano i parametri di funzionamento e le prestazioni del motore, regolandoli, se necessario, al fine di contenere al minimo le emissioni. Questi sensori erano inoltre in grado di fornire informazioni diagnostiche di base alla centralina di gestione motore (ECU) e da essa alla porta di diagnosi, tuttavia qui erano disponibili tramite protocolli non standardizzati.

Anche se i primi tipi di interfaccia diagnostica avevano fatto la loro comparsa già sul finire degli anni '60 del secolo scorso, si è dovuto attendere fino alla seconda metà degli anni '90 affinché questi fossero installati su una serie di veicoli (autovetture, autobus, e mezzi pesanti) circolanti su strada. Successivamente, una **legge federale** del governo statunitense impose l'obbligo, per i veicoli di nuova fabbricazione, di dotarsi di un'interfaccia diagnostica. Nel giro di pochi anni, anche in Europa i veicoli di nuova immatricolazione furono equipaggiati con questo tipo di interfaccia.

Fondamentalmente si tratta di un mezzo con cui, dall'esterno, è possibile monitorare alcuni parametri di funzionamento del veicolo, quali:

- livello delle emissioni dei gas di scarico;

- temperatura liquido di raffreddamento;
- posizione dell'acceleratore;
- livello carburante;
- pressione carburante;
- velocità istantanea e media;
- tensione batteria;
- tempo trascorso dall'ultimo avviamento;
- consumo di carburante medio e istantaneo;
- autonomia residua;

oltre a tanti altri parametri caratteristici relativi a organi meccanici e elettronici installati sul veicolo; guasti e malfunzionamenti registrati dall'elettronica di bordo.

Siccome l'obiettivo principale era consentire una diagnosi il più possibile agevole ed accurata, era ed è ovviamente possibile accedere agli **errori memorizzati in centralina**.

Nel 1988, per uniformare i sistemi di controllo degli impianti a iniezione, nasce il sistema OBD, che inizia ad essere presente su tutte le vetture. Adottato dal 1990 in tutti gli Stati Uniti d'America, nel 1996 viene uniformato con una convenzione Internazionale e nasce così OBD 2 (OBD-II) che inizia nello stesso anno ad essere adottato dalle maggiori case automobilistiche degli Stati Uniti, sbarcando in Europa nel 2001, prendendo così il nome di EOBD 2 (EOBD-II) ossia Enhanced OBD. I veicoli di recente fabbricazione sono, per legge, dotati di un'interfaccia diagnostica OBD-II/EOBD-II attraverso la quale i meccanici, ma anche gli utilizzatori stessi, possono acquisire importanti informazioni sullo stato del veicolo: guasti o malfunzionamenti, stato delle emissioni, parametri di funzionamento del motore e di altre parti del veicolo.

OBD ED EOBD

La normativa EOBD (European On Board Diagnostics) è l'equivalente europeo dell'OBD-II ed è stata introdotta per i motori a benzina nel 2001 assieme al livello di emissioni Euro 3 con la direttiva 98/69/CE. Si applica a tutte le autovetture di categoria M1 (con non più di 8 posti a sedere e un peso lordo del veicolo fino a 2.500 kg) immatricolate per la prima volta all'interno degli Stati membri dell'UE a partire dal 1 gennaio 2001 per le autovetture con motore a benzina e dal 1 gennaio 2004 per le autovetture con motore a gasolio. Per i modelli di nuova introduzione, le date di applicazione della normativa sono state anticipate di un anno: 1 gennaio 2000 per le autovetture a benzina e 1 gennaio 2003 per le autovetture a gasolio.



Attualmente quando si parla di interfaccia di diagnosi, ci si riferisce in modo molto generico al sistema diagnostico OBD che nella realtà ora si trova alla versione OBD-II mentre prima degli anni 2000 era possibile trovarlo in versione OBD-I. Le differenze tra OBD-I e OBD-II sono realmente legate al fatto che nella prima versione non vi era un vero e proprio standard, quindi anche le connessioni elettriche al connettore potevano essere differenti a seconda del costruttore del veicolo, mentre ora con lo standard OBD-II è stata posta chiarezza, quindi tutti i veicoli fabbricati adottano il medesimo standard (protocollo). Il connettore OBD-II è identico per tutti i modelli ma esistono cinque differenti protocolli di comunicazione:

- aVPW (Variable pulse width): Specifico per General Motors, Opel;
- aPWM (Pulse Width Modulation): Specifico per Ford;
- aISO 9141: Specifico per Chrysler in Asia ed Europa;
- aISO 14230 KWP2000 (Keyword Protocol 2000 su K-line);
- aISO 15765 (tramite linea CAN, come SAE-J2284).

DOVE SI TROVA LA PORTA

Ogni autoveicolo di ultima generazione è dotato di connettore di diagnostica OBD-II con il quale è possibile dialogare con la centralina di gestione motore (la ECU o ECM che dir si voglia) allo scopo di acquisire lo stato di eventuali codici errori (TC=Trouble Code) riscontrati durante il funzionamento. Il vantaggio di questo sistema è che gli errori vengono salvati ed è possibile, proprio tramite il protocollo OBD, leggerli per l'autoriparatore, che a quel punto può procedere con la diagnosi vera e propria.

Il protocollo OBD-II, in aggiunta al precedente OBD prevede che a comunicare e salvare gli errori non siano solo le centraline del motore e della trasmissione (cioè quelle che impattano sull'ambiente, motivo per cui è nata la necessità di questo tipo di "autocontrollo"), ma anche le centraline accessorie e quelle legate alla sicurezza.

L'interfaccia OBD è utile anche a chiunque voglia di mano propria conoscere meglio il proprio veicolo, indipendentemente dalle proprie capacità nel ripararlo con il fai-da-te. Giusto per completezza, precisiamo che è possibile trovare anche sistemi di localizzazione o altri sistemi che si collegano all'interfaccia OBD, ma in tal caso si può trattare solo di

sistemi che sfruttano il connettore per prelevare semplicemente alimentazione dal veicolo in quanto proprio su questo connettore, come si vedrà più avanti, è presente una tensione 12V.

Se viene trovato un parametro fuori norma che può causare un problema al motore, attivano la luce del cruscotto (MIL); il codice viene quindi memorizzato.

PINOUT CONNETTORE OBD II

Il sistema di diagnostica di bordo, OBD-II, è ormai su tutti i veicoli a motore dal 2004, senza particolari distinzioni; pertanto sui veicoli c'è sempre un connettore di diagnostica, che si interfaccia di norma ad uno strumento esterno che permette la diagnosi.

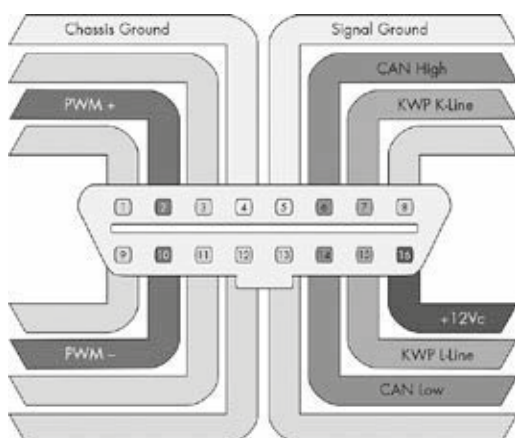
Il connettore in questione, è posizionato in punti differenti del veicolo, a seconda della scelta del costruttore, pertanto se pur ogni casa costruttrice tenda a nascondere per non renderlo visibile, questo può essere facilmente raggiunto. Ad esempio si potrà notare il connettore OBD-II sopra la pedaliera, oppure in un vano nascosto nel tunnel centrale, oppure sotto il bracciolo o sotto il cofano; insomma non c'è una posizione standard, ma basta un semplice sguardo per trovare un connettore come quello evidenziato nella **Fig. 1**.

Qualora vogliate sapere, anche solo per curiosità, dove si trova il connettore OBD-II nel vostro veicolo, sappiate che esistono alcune App che vi indicheranno dove è localizzato: nello Store del vostro Sistema Operativo troverete diverse App, quali ad esempio "Dov'è la mia presa OBD2? Find it!", oppure "Where is OBD", entrambe molto valide se pur in lingua inglese.



Fig. 1
Il connettore
di diagnosi.

Fig. 2
Connessioni
dell'OBD-II.



Anche se il CAN può funzionare fino a 1 Mbps, la versione specificata per OBD-II funziona a 250 o 500 kbps. L'indirizzamento CAN supporta due modalità: una modalità a 11 bit e una modalità estesa a 29 bit. Date le due velocità e due schemi di indirizzamento, ci sono quattro diversi tipi di CAN che si applicano al settore automobilistico. Come il SAE J1850-PWM, questo utilizza segnali differenziali per trasmettere dati su CAN a 2 fili alto (CANH) e CAN basso (CANL). Il pin 6 del connettore OBD-II è la linea CANH e il pin 14 è CANL. La linea CANH utilizza uno stato attivo di 3,5 V - 5 V e la linea CANL utilizza uno stato attivo di 0 V - 2 V. Lo stato passivo di entrambe le linee è 2,5 V.

Tralasciando le versioni precedenti dell'attuale connettore, che non avevano uno standard, spiegheremo quello attuale che a tutti gli effetti potrà essere visionato sui veicoli attualmente in commercio. Se invece possedete un veicolo datato, sicuramente troverete maggiori informazioni sul vostro connettore effettuando ricerche online.

Di seguito riporteremo la pinout del connettore standard, in cui si potrà notare che i pin 4 e 5 sono connessi allo Chassis del veicolo, mentre il pin 16 al 12 volt della batteria. Queste connessioni saranno identiche su tutti i connettori OBD-II come lo saranno anche quelli dei protocolli CAN, ISO, VPW, PWM.

Come si può vedere dalla pin-out standard del connettore, alcune linee non sono utilizzate, ma sono libere per essere utilizzate dalla casa produttrice del veicolo per altre diagnostiche o esigenze.

Nella **Tabella 1** sono dettagliati i pin utilizzati a seconda del protocollo in uso.

ISO15765

Essendo questo lo standard attuale CAN, molte aziende tra cui la più conosciuta ELM Electronics forniscono tutta la documentazione necessaria per comunicare con il veicolo.

ISO9141

Questo è il protocollo usato da Chrysler per veicoli europei e molti di quelli asiatici. Utilizza i pin 7 e 15 sul connettore, si basa su un formato seriale asincrono a 10.400 bps, come un UART.

Il livello di tensione varia da 0 V a 12 V con 12 V come stato passivo e 0 V come stato attivo.

Ogni byte inizia con un bit di start, seguito da 8 bit di dati seguito da un bit di stop.

La comunicazione avviene principalmente sul pin 7 (linea "K"), mentre il pin 15 (linea "L") non è sempre utilizzata. Il pin 15, se utilizzato, fornisce un segnale unidirezionale all'unità di controllo del motore (ECU). I dati dei messaggi possono contenere fino a 12 byte per comando.

Il termine ECU può riferirsi a un singolo modulo o una raccolta di moduli. Sono le centraline del veicolo che monitorano e controllano molte funzioni dell'auto, ad esempio l'ECM (modulo di controllo del motore), il TCM (modulo di controllo della trasmissione) ma ce ne sono tante altre.

Un protocollo alternativo ma simile a ISO 9141-2 è ISO 14230-4 KWP2000: fondamentalmente il formato dati è lo stesso, solo che i messaggi possono contenere fino a 255 byte. Questo è comune nei veicoli statunitensi prodotti fra il 2003 e il 2008.

Tabella 1
Pin del
connettore
OBDII
assegnati al
protocollo.

	GND	VCC	DATI	Note
CAN (ISO15765)	4, 5	16	6, 14	
ISO o KWP (ISO9141)	4, 5	16	7, 15	Pin 15 opzionale in alcuni veicoli
VPW (SAE J1850-VPW)	4, 5	16	2	
PWM (SAE J1850-PWM)	4, 5	16	2, 10	

SAE J1850-VPW

Questo standard si trova spesso nei veicoli General Motors fino al 2008. Questo protocollo a filo singolo utilizza il pin 2 e funziona a circa 10 Kb/s. Il bus può avere una lunghezza fino a 35 metri con 32 nodi. La modulazione a larghezza di impulso variabile (VPW) utilizza simboli a 2 bit, un impulso largo 64 μ s e un impulso largo 128 μ s. Il livello del segnale varia tra 0 V (passivo) e 7 o 8 V (attivo). Ogni pacchetto che inizia con un impulso di inizio del frame (SOF) di 200 μ s, quindi arriva l'intestazione. Seguendo l'intestazione sono i bit di dati. Il numero massimo di byte di dati consentiti dal protocollo in un messaggio normale è 12, che trasmette per primo il bit più significativo (MSB). Una volta che tutti i dati sono stati trasmessi, viene calcolato un byte di controllo di ridondanza ciclico (CRC) e aggiunto al messaggio. Come l'ISO 9141-2, anche il J1850-VPW funziona a 10.400 bps.

SAE J1850-PWM

Questa versione chiamata SAE J1850-PWM è utilizzata nei veicoli Ford antecedenti al 2008. La velocità dei dati è fissa a 10.400 bps come ISO 9141-2. La versione PWM a differenza di quella VPW ha anche una modalità 4x in cui la velocità dei dati aumenta fino a 41.600 bps. PWM utilizza un riconoscimento (ACK) per ciascun pacchetto che trasmette, la lunghezza del messaggio è limitata a 12 byte, incluso CRC e l'intestazione è di 3 byte. La struttura del pacchetto di dati è lo stesso del SAE J1850-VPW.

VARIANTI DEL CONNETTORE PER CASA PRODUTTRICE

Il connettore OBD-II oltre alle linee standard basate sui protocolli CAN/ISO/VPW/PWM, vi sono altre linee che sono riservate alla casa produttrice dei veicoli, consentendo quindi di poter sfruttare queste linee per altri scopi, ad esempio la diagnosi di altre centraline dedicate. Con appositi strumenti viene anche fornita la possibilità di dialogare direttamente con il display di bordo, l'autoradio, come pure accedere al controllo dei servizi secondari, quali luci, finestrini, tergicristalli ecc, ma si ricorda che queste manovre sono altamente sconsigliate senza le dovute conoscenze.

Data questa possibilità da parte di ogni casa produttrice di personalizzare le linee libere del connettore OBD-II, potete trovare dettagli del connettore su https://www.dashlogic.com/docs/technical/obdii_connector_pinout.

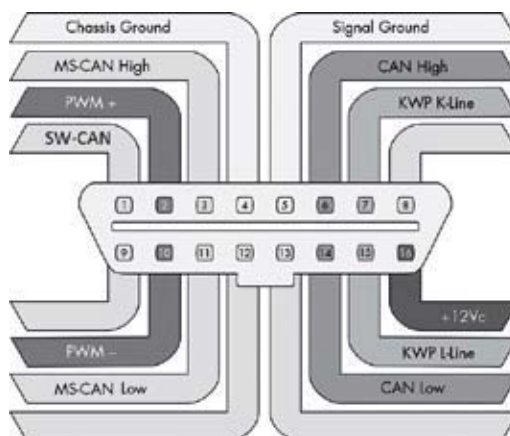


Fig. 3
Connessioni
OBD tipo
Opel.

Controllando il link, potrete senz'altro notare che la GM (General Motors) anche conosciuta in Italia con il marchio OPEL, utilizza altre linee libere.

La **Fig. 3** dettaglia il connettore OBD con pin-out dedicata dove è possibile vedere che sono stati occupati i pin 1, 3, 11.

Nel caso di Opel sono presenti un totale di 3 bus differenti, ovvero il bus principale chiamato HS-CAN sui pin 6 e 14, il bus secondario chiamato MS-CAN sui pin 3 e 11, oltre al terzo bus SW-CAN sul pin 1. Proprio a proposito di quest'ultimo bus (Single Wire CAN) permette di dialogare direttamente con il monitor dell'autoradio, inviando messaggi e intercettando tutto ciò che è di passaggio sul bus. Riguardo gli altri bus, quello principale ovvero facente capo ai pin 6 e 14 permetterà di accedere alla centralina motore, alla diagnostica principale come per tutti i veicoli, mentre sui pin 3 e 11, ovvero il bus a media velocità, permetterà di accedere ai servizi secondari, quali finestrini, luci abitacolo, tergicristalli e molto altro, permettendo il controllo degli stessi.

Non entreremo volutamente nel merito dei protocolli per questioni di sicurezza e non solo, ma per chi fosse intenzionato ad approfondire il discorso, vi informiamo che in commercio potrete trovare l'integrato STN1170 che è proprio predisposto per il controllo di tutti i bus utilizzati da GM.

Se andremo ad analizzare Ford, Fiat, BMW, Volvo o altre case costruttrici verificheremo che la situazione cambia.

Proprio per tutte queste varianti, se si vuole fare una diagnosi del proprio veicolo, va tenuto in considerazione che alcuni parametri saranno accessibili liberamente con sistemi di diagnosi standard, ma ogni casa produttrice avrà bisogno di un sistema di diagnostica dedicato per poter gestire e monitorare a pieno il veicolo.



Fig. 4
Connettore di
diagnosi BMW.

DIFFERENZE TRA CONNETTORE BMW E QUELLO STANDARD

Vi chiederete, visti i cambiamenti avvenuti nel corso degli anni, se i connettori sono stati sempre gli stessi ed è stato uniformato solo il protocollo; ebbene, per la maggior parte dei casi, il connettore di diagnosi utilizzato prima dell'introduzione dello standard, utilizza proprio il medesimo connettore se pur con leggere varianti la cui pin-out è da ritenersi del tutto differente dall'attuale. Oltre a questa tipologia, sarà possibile ritrovare sugli autoveicoli anche connettori di forma differente,

come per le auto BMW, dove il connettore è di tipo circolare (**Fig. 4**).

Nonostante la forma, si tratta sempre e comunque di un connettore di diagnosi, che se pur di tipologia differente, usa comunque delle linee che sono presenti anche nel connettore OBD-II; pertanto sarebbe sufficiente crearsi un adattatore o acquistarne uno in commercio, per poter fare una diagnosi con i moderni strumenti. Per realizzare l'adattatore tra i due connettori potete verificare in **Fig. 5** la corrispondenza tra i contatti.

QUALI DATI LEGGERE DAL BUS

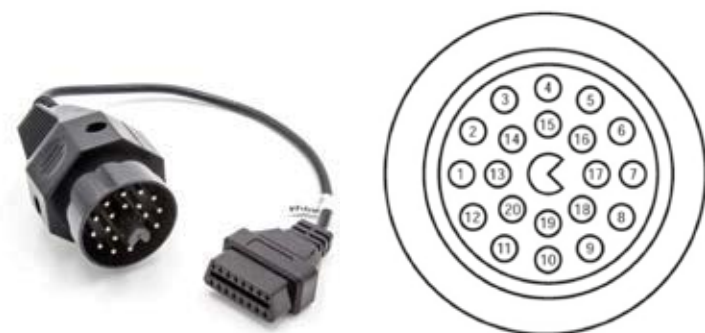
I veicoli di oggi sono "pieni" di elettronica, pertanto ci si può aspettare di leggere dalla OBD anche dati non riguardanti il motore e magari modificare qualche parametro; in realtà le modifiche sono normalmente limitate al personale dell'assistenza tecnica. Inoltre molti dati, se pur disponibili sul bus, non sono accessibili liberamente. Proprio per questo si potranno leggere sensori di vario genere, ma non si potrà ad esempio modificare eventuali configurazioni ad esso abbinati, come soglie massime, calibrizioni, ecc. Come anticipato, se parliamo di sistemi di diagnostica di aziende come Bosch, Wurth e altre famose, si può pensare che queste possano leggere qualsiasi dato del veicolo e su qualsiasi veicolo, ma così non è, motivo per il quale spesso vi sarà capitato di vedere non uno, non due, ma anche una decina di sistemi diagnostica dal vostro meccanico/elettrauto di fiducia. Ogni sistema di diagnosi ha le proprie caratteristiche: ad esempio uno può accedere ad alcune centraline piuttosto che altre, oppure consente di effettuare la lettura di alcuni parametri invece che di altri. Se un sistema da migliaia di euro non permette la diagnosi completa, figuriamoci quelli per uso "dilettantistico". Esistono in commercio molte interfacce dedicate a un pubblico più esigente, potendo svolgere funzioni di controllo simile ad apparecchi professionali. Il loro costo varia intorno ai 300€/400€, interessante per chi lavora nel settore, un po' meno per chi è solo interessato al service e poche altre funzioni di base.

In campo automobilistico è disponibile una grande varietà di interfacce OBD-II a costi ormai abbordabili (poche decine di euro) e per le auto sono spesso disponibili software e applicazioni gratuite o quasi da usare con un notebook o lo smartphone. Anche per moto oggi si trovano svariate App. Vendendo ora ai parametri che possono essere letti dai sistemi di diagnosi OBD acquistabili liberamente, sono in grado di mostrare i dati di qualsiasi



OBD-II (Femmina)	BMW 20 pin (Maschio)	
4, 5	19	Chassis Ground
7	17, 20	K line
15	15	L line
16	16	Power 12V

Fig. 5
Connettori
per realizzare
l'adattatore.



sensores collegato alla centralina, in particolare il sensore di carico motore (calcolato), il contagiri, velocità, temperatura sistema di raffreddamento, sonda lambda, posizione farfalla del gas, pressione del turbo, la quantità di ossigeno circolante, la temperatura dei catalizzatori, lo stato dell'EGR con la relativa percentuale di errore e molti altri. Spesso è anche offerta la possibilità, usando software/app di effettuare anche un log (registrazione in tempo reale dei parametri letti) in modo da effettuare meglio la diagnosi nel caso si presentino guasti o problemi. Grazie a questi dispositivi, anche i non addetti ai lavori hanno inoltre la possibilità di visualizzare gli errori in centralina che potrebbero non essere visibili dalla strumentazione di bordo, al punto che molte volte dietro la temuta spia di "manutenzione motore" si celano errori facilmente risolvibili o talvolta già risolti. Da notare che non tutti i dati sopra elencati sono sicuramente visibili su ogni veicolo, anzi, alcuni di questi potrebbero non esserlo a seguito di limitazioni imposte dalla casa automobilistica. Un'importante funzione comune è la possibilità di effettuare il reset della spia di guasto motore, proprio per il motivo appena indicato; dalla porta OBD è anche possibile resettare la spia o segnalazione Service, ovvero quella che indica la scadenza del controllo e della manutenzione periodica: quello che comunemente definiamo "tagliando". Purtroppo alcune case automobilistiche non consentono tale reset a tutti gli strumenti, ma lo riservano alla propria rete di assistenza, scelta, questa, che disturba e non poco chi è abituato a farsi da se il tagliando. Fortunatamente tale situazione non è così diffusa e su molte auto (Ford, Volkswagen recenti, Peugeot-Citroen) è possibile resettare la spia con una semplice procedura che spesso si compie con il bloccaporte attivato e coinvolge il pulsante del reset del contachilometri, la chiave di accensione e via di seguito. In alternativa, solitamente previo acquisto, è possibile acquistare dei software specifici che a differenza di quelli free permettono proprio l'azzeramento del Service. Dato che la vostra curiosità sarà tanta, è il caso di citare ad esempio un paio di software, quali VAG-Com VCDS (per il marchio Volkswagen/Audi/Seat) oppure Rheingold per BMW/Mini; ma ste tranquilli, perché se non volete spendere troppo trovate facilmente delle App di libero utilizzo per tablet e smartphone, fermo restando che dovrete trovare quella specifica per il vostro veicolo e vi assicuro che non è così facile trovarle, ma proseguendo nell'articolo faremo qualche citazione.



Fig. 6
Interfaccia di
diagnostica
wireless.

QUALI TIPI DI INTERFACCE OBD ESISTONO

Qualora decidiate di procurarvi una interfaccia OBD, scoprirete presto che in commercio se ne trovano davvero una infinità, dalle interfacce di diagnostica di cui abbiamo parlato fino ad ora, ai dispositivi ad uso speciale, come ad esempio sistemi HUD (Head Up Display), localizzatori, microspie, ecc. Ma prendiamo in esame tutto ciò che offre internet, cercando di trattare le categorie principali. Riguardo i sistemi di diagnosi, si possono trovare per la maggiore interfacce Bluetooth/Wi-Fi come quelle offerte da www.futurashop.it con codice FR585 e FR586 (**Fig. 6**). Queste tipologie si interfacciano via radio con il proprio smartphone/tablet sfruttando le App di diagnostica come Torque, Scan Master, ODB Doctor o qualsiasi altra app che è in grado di comunicare con interfacce basate su ELM327. Oltre a questo tipo di interfaccia, che è decisamente la più comoda, compatta e facile da collegare, si può trovare anche in commercio qualche unità collegata direttamente via cavo al connettore OBD che si interfaccia a sua volta con il PC mediante la porta USB (**Fig. 7**). In tal caso possiamo consigliare un ottimo prodotto, ovvero UniCarScan UCSI-1001 che viene fornita corredata di software ScanMaster in licenza OEM. In questo caso il software e l'interfaccia sono aggiornabili gratuitamente e offrono interessanti caratteristiche, quali la funzione Log, oltre a monitorare in tempo reale tutti i dati rilevabili sul veicolo. Sempre in merito ai sistemi di diagnostica, si possono trovare interfacce dotate di display, che permettono la comodità di non avere il pc o smartphone, in quanto tutti i dati sono accessibili su display. In commercio si trovano diverse interfacce di questo tipo, ma attenzione perché proprio come i sistemi Wi-Fi e/o bluetooth, vista la grande disponibilità è importante scegliere bene, evitando prodotti troppo a basso costo, perché poi la qualità

→ **Fig. 7**
Interfaccia
OBD USB.



← **Fig. 9**
Interfaccia
di reset del
service.



viene meno. Possiamo consigliare i prodotti Autel oppure della Konnwei che sono davvero ottimi. Qualche buona alternativa la si può trovare anche tra i prodotti Vgate. Un buon rapporto qualità prezzo lo si può avere l'Autel AL519 (**Fig. 8**).

Sempre per la diagnostica, esistono sistemi di reset service, ovvero unità apposite che non richiedono software particolari come Vag-Com: si tratta di interfacce di reset service specifiche per ogni casa automobilistica, ad esempio quella mostrata in **Fig. 9**, in cui ad ogni pulsante corrisponde una soglia chilometrica precisa. In questo caso è sufficiente collegare l'interfaccia al connettore OBD, accendere il quadro dell'auto e premere il pulsante relativo alla soglia desiderata. Quello del Service è un mondo a se, ovvero che se pur strettamente legato alla diagnostica, richiede spesso delle

interfacce dedicate, motivo per il quale ci si deve recare dal proprio meccanico di fiducia. Giusto a titolo informativo, i più esperti sappiano che però esistono anche interfacce OBD di diagnostica che permettono l'azzeramento del Service in tutta autonomia, sfruttando un'interfaccia proprietaria corredata di App dedicata (dai 50 ai 100 euro è il prezzo di mercato per la parte hardware e software), in tal caso si avrà una interfaccia hardware OBD unica, ma si dovrà poi scegliere l'App/software dedicato in base alla marca del veicolo. Uno dei più noti marchi che realizzano questo tipo di App/OBD è Carly www.mycarly.com.

Oltre ai sistemi di diagnostica tradizionale, esistono però anche altri dispositivi che si interfacciano al connettore OBD-II, ad esempio si possono trovare sistemi HUD, ovvero sistemi dotati di due unità, la prima connessa al connettore OBD che mediante connessione radio (bluetooth e/o WiFi) si interfaccia all'unità display. L'unità display riceve alcuni dati, tipicamente velocità di crociera, giri motore, temperatura e li proietta sul parabrezza della propria auto rendendoli facilmente visibili soprattutto quando c'è poca luce ambiente. Non vanno confusi con le interfacce OBD i dispositivi che pur applicandosi al relativo connettore, lo fanno solo per prelevare alimentazione: è il caso dei localizzatori GPS come quello offerto da Vodafone sotto il nome di V-Auto. In questo caso si tratta di una interfaccia simile a quelle Bluetooth/WiFi compatte che abbiamo citato sopra, ma che non si occupa della diagnostica, anzi sfrutta il connettore principalmente per prelevare alimentazione in modo da essere operativa, inviare le coordinate satellitari via SMS. Tali localizzatori potrebbero avere un segnale GPS scarso, proprio dovuto alla posizione in alcuni casi ostica del connettore di diagnosi.

→ **Fig. 8**
L'interfaccia
Autel.



Come avete visto il connettore OBD è elemento comune in tutti i dispositivi sopra riportati, che non sempre viene utilizzato solo come connettore di interfacciamento per la diagnosi, ma si tratta pur sempre di casi particolari che non rappresentano la natura per cui è nato il connettore.

PRINCIPALI INTEGRATI PER LA DIAGNOSI

Fino ad ora abbiamo parlato delle possibili interfacce disponibili in commercio per poter effettuare la diagnosi, abbiamo tra le righe citato "ELM327" senza però entrare nel merito, pertanto è inevitabile che vi sarete chiesti di cosa si tratta. Per dirla in breve, si tratta di un nome con cui è conosciuto un microcontrollore programmato, prodotto dalla Elm Electronics, al cui interno prevede un codice appositamente scritto per poter dialogare con il veicolo ed effettuare la diagnosi. Ora senz'altro vi sarà chiaro di cosa si tratta, quindi se effettuerete ricerche online, vi sarà facile trovare interfacce di diagnosi in cui troverete indicato "compatibile ELM327" oppure "basate su ELM327", proprio per specificare che all'interno dell'hardware è presente un integrato ELM327, oppure, come spesso capita, un clone di questo. Una possibile alternativa a questo integrato decisamente molto diffuso, la si può trovare nel modello STN1110 della casa OBD Solutions, che offre le stesse caratteristiche di base dell'ELM327.

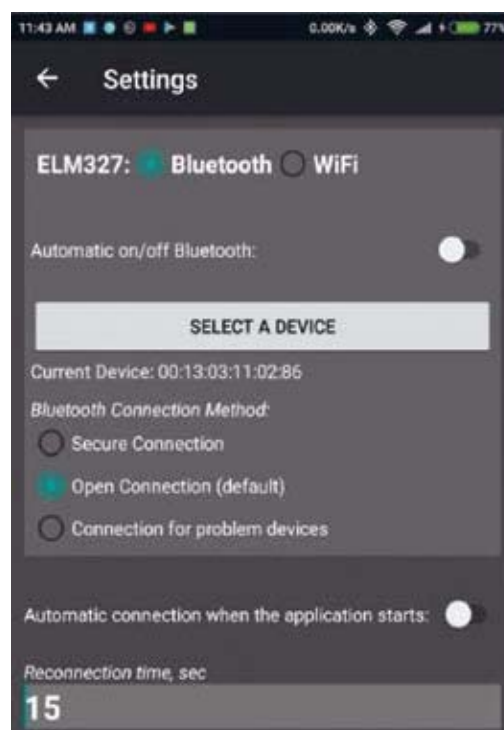
La **Tabella 2** propone un confronto fra i due integrati. Trattandosi di integrati di diagnosi, se vorrete, potrete realizzarvi in tutta autonomia un circuito di diagnostica, in tal caso online troverete davvero tanti schemi, uno dei quali, basato su ELM327 è stato pubblicato proprio su ElettronicaIn 145. Se invece vorrete basarvi sull'integrato della OBD Solutions, ad esempio perché offre aggiornamenti gratuiti, potrete addirittura scaricare il file sorgente del PCB sia per Altium che per Eagle direttamente da <https://www.obdsol.com/solutions/chips/stn1110>. Se però avrete in mente di realizzarvi una interfaccia OBD non compatta come quelle in commercio, vi consigliamo di prevedere sul PCB un connettore DB9-Maschio in modo da poter sfruttare i cavi standard connettore DB9 e OBD maschio per interfacciarsi all'auto come il cod. 2846-CAVOOBDII (<https://www.futurashop.it/cavo-obd-ii-j1962m-db9f-2846-CAVOOBDII>). Il motivo di questa diffusione così massiccia di questi due integrati, riguarda una loro caratteristica, ovvero viene fornito un datasheet completo (per ELM327 e di conseguenza STN1110) in cui viene riportato in modo molto chiaro come dialogare con l'integrato stesso attraverso comandi AT. Questo è un notevole vantaggio per tutti i progettisti un po' esperti che vogliono cimentarsi della diagnosi auto. Ad esempio, se pur l'App che userete avrà limitazioni in quanto effettuerà l'in-

	ELM327 v1.4	STN1110
Base microcontroller	PIC18F2580	PIC24HJ128GP502
Architecture	8-bit	16-bit
Processing speed	4 MIPS	40 MIPS
Flash (ROM)	32 KB	128 KB
RAM	1.5 KB	8 KB
Pin count	28	28
Available packages	PDIP, SOIC	PDIP, SOIC, QFN
Supply voltage range	4.5 to 5.5V	3.0 to 3.6V ¹
Supports all OBD-II protocols	yes	yes
ELM327 command set	yes	yes
Enhanced "ST" command set	no	yes
Firmware upgradeable	no	yes
Large OBD message memory buffer	no	yes
Low power mode	yes	yes
Supported UART baud rates	9600 bps to 500 kbps	38 bps to 10 Mbps
OBD message filtering	basic	advanced
Price each, for 1000 units	\$24	\$10
Price each, high volume	\$19	\$4.95

Tabella 2
ELM327 ed
STN1110 a
confronto.



Fig. 10
Due schermate
dell'app Terminal
ELM327.



terrogazione solo di alcuni PID, con un programma terminale, potrete dialogare via comandi AT ed inviare il comandi di richiesta su un altro PID, se riceverete risposta vi rimarrà solo da capire come decodificare il dato in HEX (Esadecimale). Ovviamente è possibile non solo leggere, ma anche scrivere dei dati sul bus, ma in tal caso è molto rischioso, quindi dovete sapere perfettamente cosa state facendo e cosa state variando. Per gli esperti che sono interessati ad effettuare un test con la propria interfaccia basata su ELM327, vi consigliamo una App Android davvero molto comoda che vi permetterà non solo di inviare comandi AT e verificare la relativa risposta esadecimale, ma anche effettuare un calcolo matematico sfruttando i byte ricevuti ottenendo direttamente il risultato in forma decimale. L'App in questione è Terminal ELM327 e di essa e in Fig. 10 vi riportiamo un paio di schermate, a titolo d'esempio.

CONCLUSIONI

Bene, per ora ci fermiamo qui, dopo avervi introdotto i concetti di diagnostica e relativa porta di comunicazione, esponendone tutte le evoluzioni dalla nascita ad oggi. Abbiamo avuto modo anche di descrivervi le unità di diagnostica collegabili alla porta e funzionanti in stand-alone, ma anche i semplici dispositivi interfacciamento con l'OBD-II

che fanno da tramite tra l'autoveicolo e un computer, tablet o smartphone nei quali andranno in esecuzione software o App in grado di dare un senso ai dati, esporli e memorizzarli, eventualmente creando dei log, grafici e indicatori che consentono una visualizzazione più intuitiva. Software da PC e App per tablet e smartphone saranno l'argomento su cui verterà la seconda e ultima puntata di questo tutorial, nonché lo strumento che molti di voi utilizzeranno per imparare qualcosa di più sul funzionamento dei moderni motori a combustione interna ed anche delle auto ibride ed elettriche, esistendo, ormai, anche App dedicate al mondo dell'elettrico plug-in e ibrido. Alla prossima puntata ...dunque!



Cosa occorre?

Alcuni dei tester OBD presentati in questo articolo sono disponibili presso Futura Elettronica. Il tester palmare OBD2 VGATE VS890 MAXISCAN (cod. FR772) costa Euro 45,00, il tester OBD Bluetooth (cod. FR586) è in vendita a Euro 22,00, mentre il tester OBD Wi-Fi (cod. FR585) è disponibile a Euro 39,00. Il cavo OBD-II con connettore J1962 Maschio e connettore DB9 Femmina (cod. CAVOBDII) è in vendita a Euro 7,50. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

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.



Prezzi IVA inclusa.

LOCALIZZATORE GPS/GSM/GPRS CON ASCOLTO AMBIENTALE

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



cod. FR763

€ 74,00



MICRO LOCALIZZATORE SATELLITARE GPRS/GSM

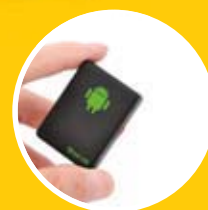
Tutto sotto controllo con il nuovo micro localizzatore GPRS/GSM. Per attivarlo basta inserire una SIM di qualsiasi operatore; inviando un SMS potrai attivare l'ascolto ambientale oppure ricevere un link con le coordinate che indicano la posizione in tempo reale del localizzatore. Dispone di batteria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

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



cod. A8TRACK

€ 14,00



LOCALIZZATORE TASCABILE GPS/GSM/GPRS

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



cod. TK102

€ 54,00

Sostituire il telecomando TV? Nulla di più facile!

Il tuo telecomando TV è rotto o non funziona più bene?
È arrivato il momento di sostituirlo 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!



PRONTI ALL'USO



SAMSUNG

Cod. **TVIR-SAMSUNG**



SONY

Cod. **TVIR-SONY**



PHILIPS

Cod. **TVIR-PHILIPS**



LG

Cod. **TVIR-LG**



Panasonic

Cod. **TVIR-PANASONIC**

€ 5,90
OGNI PEZZO

DA PROGRAMMARE



€ 9,00
cod. **TVFREEDOM2**

2 in 1

DISPONIBILE ANCHE 1 in 1
cod. **TVFREEDOM1**

€ 8,50

Telecomando universale, programmabile da PC, per controllare qualunque dispositivo di qualsiasi marca e modello. Include tutte le funzioni dell'originale, configurazione facile e veloce. Mai obsoleto grazie ad aggiornamenti gratuiti. Oltre 260.000 modelli in continuo aggiornamento, può comandare fino a due dispositivi audio/video: TV, TDT, SAT, DVD, BLURAY, proiettori, soundbar, VCR, media center e impianti Hi-Fi delle migliori marche. Richiede il download su PC-Windows di software gratuito per la selezione e la programmazione del telecomando.



UNA USB PER QUATTRO SERIALI

di DAVIDE SCULLINO



Grazie all'integrato FTDI FT4232, consente di assegnare a una sola porta USB del computer fino a quattro porte seriali.

Più o meno tutti coloro che hanno installato su un Personal Computer un dispositivo interfacciato via USB, sanno che il suo utilizzo passa per la creazione, attraverso i driver caricati nel sistema operativo, di una porta COM virtuale, vale a dire di una seriale RS232 che però fisicamente non è tale, giacché le vere seriali sono gestite da UART che nei PC più recenti erano le 16550 o compatibili. Però l'emulazione è fattibile e realistica in quanto l'USB è una connessione seriale, ovvero un link dove i dati viaggiano serialmente, esattamente come su una RS232, ma con due differenze: il data-rate è nettamente maggiore e inoltre nell'USB si ha a che fare con un bus e non con una connessione punto-punto, perciò con le due linee dati si possono far comunicare fino a 127 dispositivi con un host.

Nato in casa FTDI, rappresenta la quinta generazione di interfacce USB di questo costruttore ormai specializzato in converter RS232/USB per ogni genere di applicazione (li troviamo persino su Arduino) ed è per certi versi innovativo, in quanto consente di affacciare a una sola porta USB del computer (Host) ben quattro porte seriali complete, gestire distintamente e visibili dal computer come altrettante COM virtuali installate dal sistema operativo. L'FT4232H è un converter compatibile USB 2.0 High Speed (480Mb/s) che

dal lato seriale implementa 4 UART, su due dei quali può essere configurato opzionalmente un engine MPSSE; questo permette al componente di operare con due porte UART/Bit-Bang più due motori MPSSE utilizzato per emulare JTAG, SPI, I²C, Bit-bang e altre modalità seriali sincrone. In questo riquadro vedete lo schema a blocchi del componente.

L'FT4232H presenta le seguenti caratteristiche avanzate:

- gestione interna dell'intero protocollo USB senza necessità di firmware da scrivere;
- modalità USB 2.0 High Speed

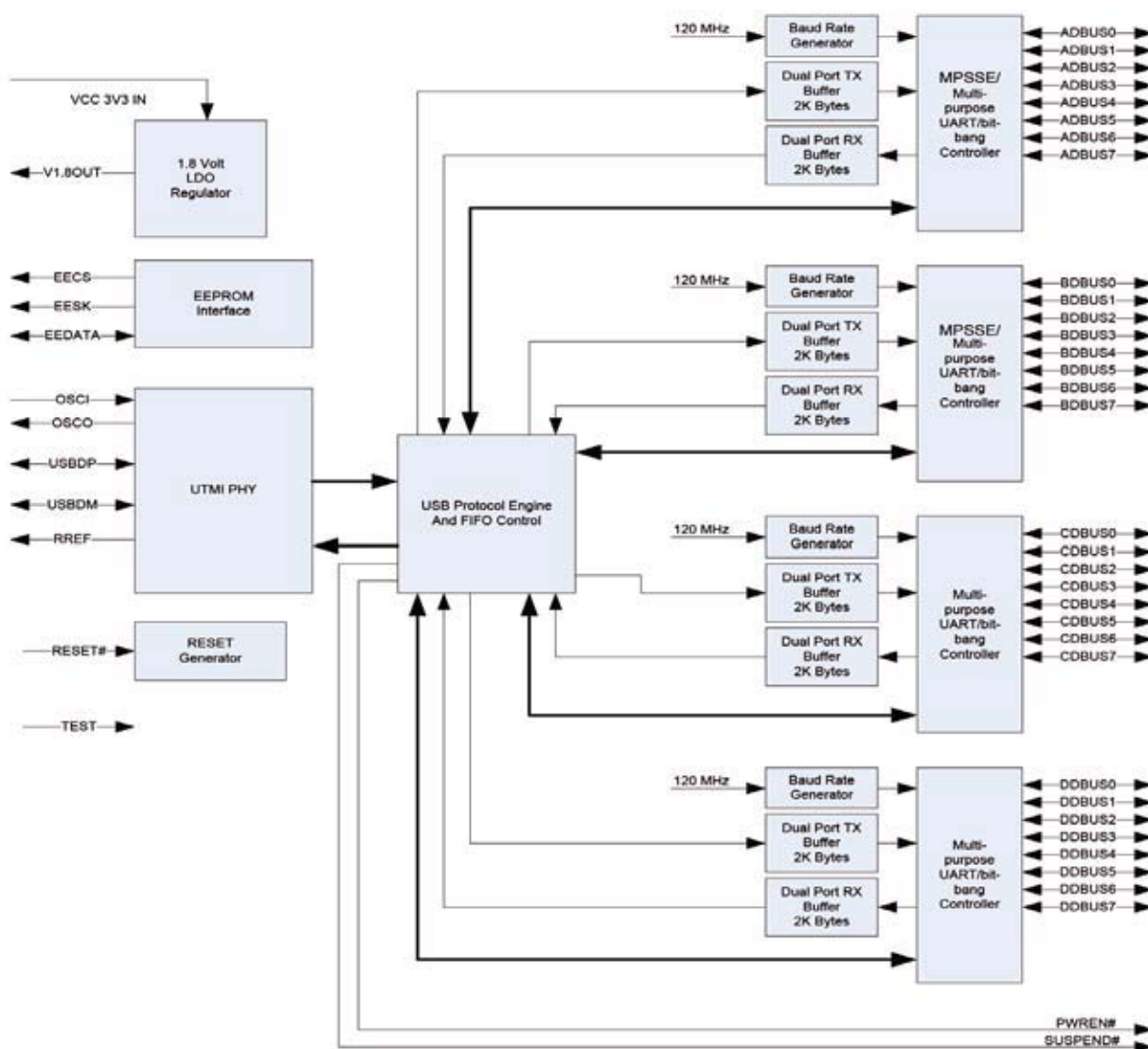
(480Mbit/s) e Full Speed (12Mbit/s);

- due Serial Engine (MPSSE) sincroni multi-protocollo sui canali A e B, per semplificare la progettazione di convertitori USB to JTAG, I²C, SPI o bit-bang;
- possibilità di collegare dei LED TX/RX indicatori del traffico dati mediante uno 74HC595 shift register esterno;
- timeout del buffer di ricezione regolabile;
- regolatore 1,8V LDO per la V_{CORE}, funzione POR integrata e PLL on-chip da 12 a 480 MHz;
- stringhe Operational Configu-

ration Mode e USB Description configurabili da PC nella EEPROM esterna tramite interfaccia USB;

- corrente degli I/O in modalità drive impostabile tra 4, 8, 12 o 16 mA;
- compatibilità con i controller host UHCI/OHCI/EHCI;
- USB Bulk data transfer mode (pacchetti di 512 byte in modalità High Speed);
- grado di temperatura industriale da -40°C a 85°C;

L'FT4232 è disponibile in package 64 LQFP o QFN, ed anche 56 pin VQFN.



Data la limitatezza delle porte USB in un computer, per collegare più dispositivi si sfrutta tale prerogativa dell'hardware come l'hub; esistono, però, esigenze applicative in cui in un dispositivo debba esserci una sola connessione USB (di tipo Device) ma più connessioni seriali indipendenti: è il caso in cui si voglia impiegare una sola USB.

In una situazione del genere diventa utile il circuito descritto in queste pagine, che grazie a uno specifico circuito integrato FTDI, consente di "tirare fuori" ben quattro porte seriali complete da una sola connessione USB device: porte seriali con tanto di segnali di controllo della comunicazione e quindi DTR, RTS, CTR e via di seguito, nonché di generatori di baud-rate indipendenti. I livelli logici delle seriali sono del tipo TTL 0/5V o 0/3,3V in base all'impostazione di un ponticello che descriveremo più avanti e grazie al fatto che l'integrato utilizzato prevede la separazione delle tensioni di alimentazione del core e dei registri di I/O; da tali livelli è poi possibile ricavare quelli standard RS232, RS422 o RS485 mediante convertitori specifici come MAX232, MAX485 ecc.

SCHEMA ELETTRICO

Andiamo dunque a vedere lo schema del circuito, nel quale ci appare la grande semplicità permessa dal componente FTDI: questo integrato, siglato FT4232H, sbriga al proprio interno la creazione, lo smistamento e la gestione dei pacchetti di dati da e verso la USB in direzione delle quattro seriali; ciò senza richiedere alcuna programmazione e senza microcontrollori dei quali scrivere codice, a tutto vantaggio della semplicità d'uso e della compattezza di una soluzione hardware pronta all'uso. Sulle porte seriali è supportato un Transfer Data Rate fino a 12 Mbaud; le interfacce UART supportano le modalità di comunicazione 7/8 bit dati, 1/2 stop bit e le condizioni Odd/Even/Mark/Space/No Parity. L'integrato supporta i protocolli di comunicazione (a livello dati proprie dei link RS232, RS422 e RS485), per implementare le quali è necessario (a livello fisico) interfacciare le linee degli UART desiderati con appositi transceiver traslatori di livelli.

Per il protocollo RS485 è supportata la modalità di controllo auto-transmit mediante il pin TXDEN. L'integrato FT4232H è siglato U1 nello schema elettrico e come vedete si interfaccia direttamente al bus USB tramite un connettore che per nostra scelta è l'USB type C, dato che diverrà lo standard a breve: non a caso lo troviamo già in molti apparati mobile come tablet e smartphone.

Il connettore USB è quello siglato USB1 e come vedete riporta i due fili dati standard che nella connessione USB type C sono quelli compatibili con l'USB 2.0 (visto che il chip FTDI è compatibile USB 2.0 Full Speed e High Speed 480 Mbps) e in pratica fanno capo ai contatti DN1, DN2, DN3, DN4 collegati a coppie, perché in questo caso, pur montando un connettore USB-C comunichiamo secondo il protocollo USB 2, che prevede la comunicazione su due soli fili dati. Questi ultimi li abbiamo protetti da disturbi e picchi di tensione mediante un doppio TVS (Transient Voltage Suppressor) siglato CR1, che risulta utile in quanto il cavo USB, specie se lungo, può captare e portare nel circuito sovratensioni che potrebbero danneggiare l'U1.

Il connettore trasporta anche, tramite l'induttanza di filtro FB1, l'alimentazione per il circuito, la quale può essere esclusa aprendo il ponticello JUSB, laddove si desiderasse alimentare il tutto con una fonte di alimentazione distinta da quella dell'host USB.

L'interfaccia USB viene gestita dall'UTMI (Universal Transceiver Macrocell Interface) PHY che è il blocco deputato alla Full-speed/High-Speed SERDES (serialise – deserialise) ossia alla traslazione da seriale a USB e viceversa; esso fornisce i clock ai rimanenti blocchi del chip.

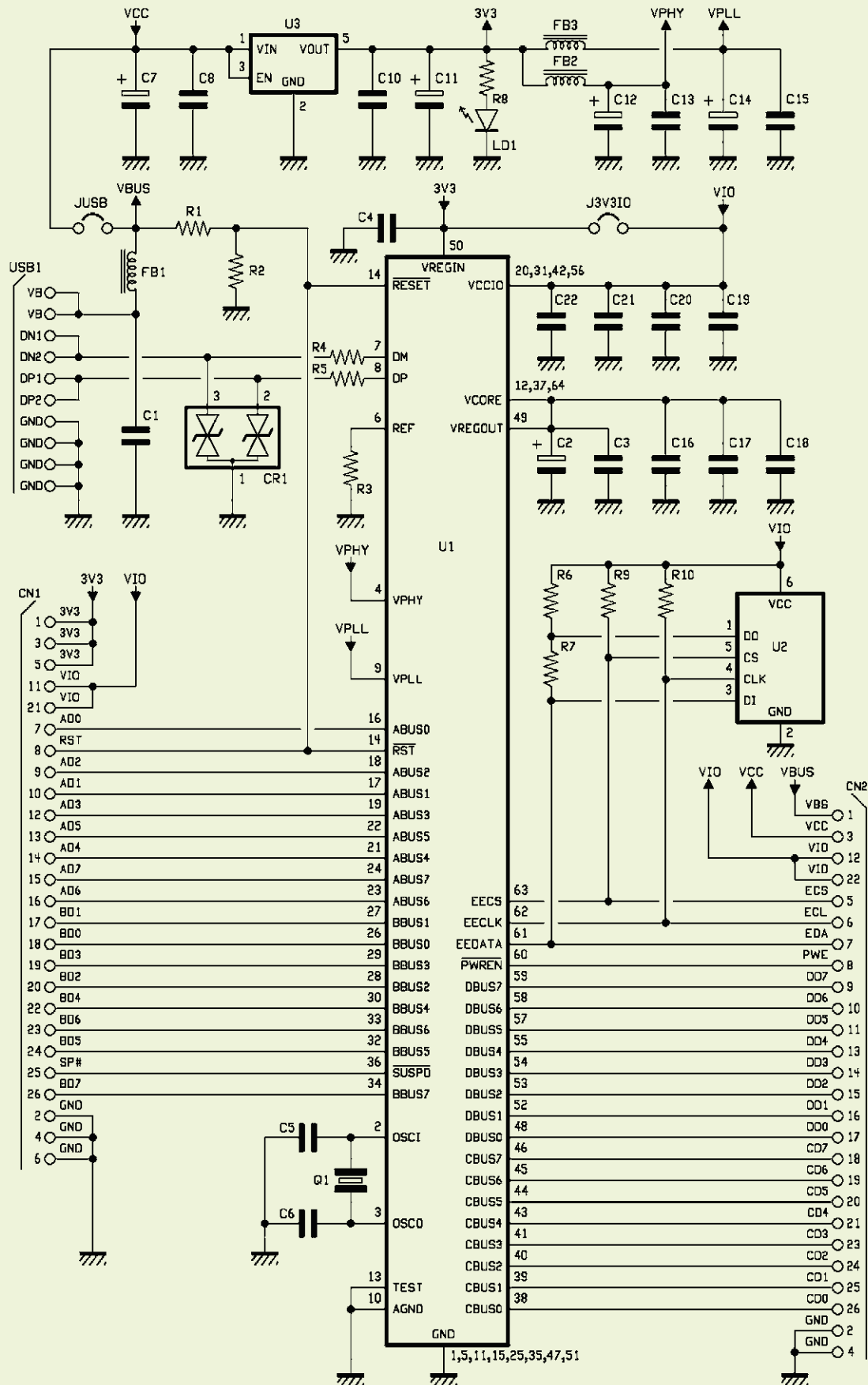
Inoltre la UTMI implementa la funzionalità Bit-stuffing/unstuffing (bit stuff error detection), gestisce l'USB Resume e le funzioni di Wake Up e Suspend (associata al pin omonimo dell'integrato, che descriveremo tra breve).

Le quattro porte seriali sono rese disponibili sui blocchi di linee ABUS 0÷7, BBUS 0÷7, CBUS 0÷7 e DBUS 0÷7, che nel nostro circuito terminano sulle file di contatti CN1 (seriali 1 e 2) e CN2 (seriali 3 e 4); al connettore CN1 arrivano anche la linea di reset, utile a resettare il controller dall'esterno (con un impulso a zero logico) e la SUSPD (suspend), mentre sul CN2 troviamo EECS, EECLK, EEDATA del bus SPI implementato dall'FT4232 e la PWREN, oltre alle alimentazioni interne. Quanto al reset, in realtà è previsto un power-on reset implementato con i resistori R1 e R2: se manca l'alimentazione sulla connessione USB l'integrato viene mantenuto resettato e il reset si porta a livello alto non appena l'alimentazione VBUS è presente. Naturalmente se si applica dall'esterno un livello logico alto sulla linea RTS del CN1 questo prevale e l'integrato esce dalla condizione di reset. Per quanto riguarda gli ABUS, BBUS, CBUS e DBUS, la corrispondenza tra le relative linee e i se-

CARATTERISTICHE TECNICHE

-  **Tensione di alimentazione:**
5÷12 Vcc
-  **Corrente assorbita:**
50 mA
-  **Connessione d'ingresso:**
USB-C
-  **Connessioni seriali:**
4 TTL in modalità RS232, RS422 o RS485
-  **Versione USB supportata:**
2.0 High-Speed a 480Mbps





gnali della seriale è quella proposta nella **Tabella 1**, riferita ad una singola porta, per le altre vale lo stesso, fermo restando che per ciascuna i piedini hanno nomi specifici (ABUS0÷7 per la porta A, DBUS0÷7 per la porta B e via di seguito). Quindi la linea 0 è il TXD, la 4 il DTR e via di seguito; nel caso di utilizzo come parallela, la corrispondenza indicata è quella degli 8 bit, mentre per chi desidera impiegare la porta in configurazione SPI più GPIO, abbiamo riportato la corrispondenza con i fili della SPI completa, a quattro linee. Del canale dati di ciascuna UART, la linea TXD è l'uscita e la RXD l'ingresso, quindi l'integrato si comporta come un DTE. SUSPD è una linea che viene posta a livello logico basso quando il bus USB è in modalità Suspend; invece la PWREN è un'uscita attiva a zero logico quando l'USB funziona regolarmente, mentre commuta a livello alto quando l'USB va in modalità suspend mode o il device non è configurato. Questo può servire, ad esempio, a gestire la modalità low-power in circuiti collegati all'integrato FT4232 quando quest'ultimo si porta in suspend o

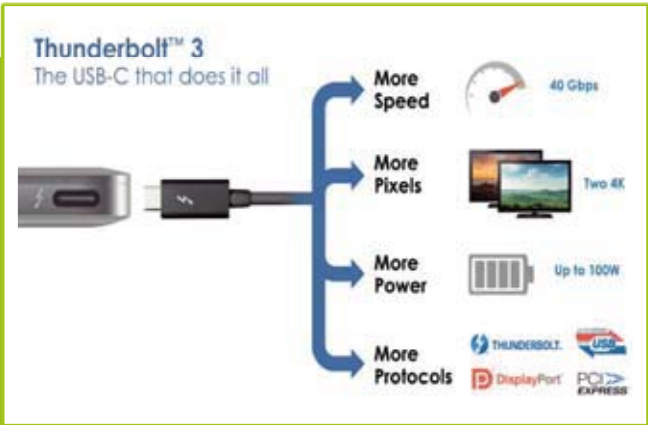
BUS0	TXD	D0	TCK/SK
BUS1	RXD	D1	TDI/DO
BUS2	RTS#	D2	TDO/DI
BUS3	CTS#	D3	TMS/CS
BUS4	DTR#	D4	GPIOL0
BUS5	DSR#	D5	GPIOL1
BUS6	DCD#	D6	GPIOL2
BUS7	RI#/TXDEN*	D7	GPIOL3

non è ancora stato configurato come periferica, in modo da risparmiare energia. Nello schema elettrico, come vedete, c'è una EEPROM seriale a interfaccia SPI (siglata U2) da 2 kbit (128x16) dove viene scritta la configurazione del sistema; ebbene, l'FT4232 la gestisce tramite le linee EECS (Chip Select), EECLK (clock) ed EEDA-TA. Quest'ultima è bidirezionale, quindi è connessa direttamente al Data-In della EEPROM (DI) e me-

Tabella 1
Corrispondenza delle linee delle porte seriali.

THUNDERBOLT CONTRO USB: LA GARA CONTINUA...

La competizione tra l'innovativa interfaccia di Intel (che non è una semplice porta di comunicazione ma un bus universale multiperiferica) e l'Universal Serial Bus non conosce battute d'arresto e recentemente il vicepresidente di Intel, Gregory Bryant, ha preannunciato l'arrivo della nuova versione 4 della sua Thunderbolt. Questa interfaccia promette di surclassare la pur prestante USB 3.2 su cui si basa, peraltro, l'ultima nata USB-C (almeno per quel che riguarda i dati...) offrendo velocità di trasferimento incredibili, con una banda passante totale che raggiunge gli 80 Gbps: quattro volte il data-rate dello standard USB 3.2 Gen 2x2. L'annuncio è stato dato all'ultimo CES di Las Vegas in concomitanza con la presentazione dei nuovi processori Tiger Lake e della scheda grafica discreta DG1; questo perché Thunderbolt 4 sarà parte integrante dell'undicesima generazione di processori Intel. Da un punto di vista fisico, lato processore la Thunderbolt 4 dovrebbe utilizzare quattro canali PCI-E 4.0 per ottenere la velocità record di 80 Gbps; già la Thunderbolt 3 utilizza quattro canali PCI-E 3.0,



limitandosi, tuttavia, a una velocità di comunicazione di 40 Gbps. Per fare un confronto, la più veloce USB attualmente reperibile consente una velocità massima di 20 Gbps (SuperSpeed USB 20 Gbps); la superiorità della Thunderbolt 4, nata per competere con la nascente USB 4, si può comprendere considerando che la USB 4.0, appunto, sarà basata sulla tecnologia Thunderbolt 3. Questa USB, fisicamente interfacciata mediante il connettore USB Type-C e retrocompatibile con tutte le versioni precedenti, parte già indietro rispetto alla Thunderbolt 4 proprio perché si "appoggia" sulla precedente generazione Thunderbolt (quella attualmente in uso) e avrà, quindi, una velocità massima di 40 Gbps: l'esatta metà delle prestazioni della Thunderbolt 4.

Visto che nel circuito abbiamo utilizzato un connettore USB-C, utilizzato in modalità USB 2, ne approfittiamo per spendere qualche riga sul relativo standard, che sul piano della comunicazione è capace di offrire un transfer-rate 10 Gbit/s (1.250 MB/s) nominali. Per quanto riguarda l'erogazione di corrente, la USB tipo C supporta 5 profili di potenza in base alla connessione fisica scelta e all'impostazione all'interno dell'host:

- Profilo 1: 5 V a 2,0 A (10 W);
- Profilo 2: 5 V a 2,0 A (10 W), 12 V a 1,5 A (18 W);
- Profilo 3: 5 V a 2,0 A (10 W), 12 V a 3 A (36 W);

- Profilo 4: 5 V a 2,0 A (10 W), 12 V, 20 V a 3 A (36 W, 60 W);
- Profilo 5: 5 V a 2,0 A (10 W), 12 V, 20 V a 5 A (60 W, 100 W).

Nel nostro caso utilizziamo il Profilo 1, ossia quello standard.

Il nuovo connettore Tipo C, oltre ad essere reversibile (come quelli della Apple, per intendersi....) è il primo a supportare i profili a 12 V e 20 V, ma nonostante ciò rimane polarizzato, tramite due serie di pin (A e B); quindi sul lato dell'alimentazione il connettore tipo C introduce due nuovi pin VBUS a 12 V e 20 V. I connettori di tipo A, B e mini B non hanno i pin VBUS 12 e 20 V,

che comunque dovrebbero essere alimentati tramite un alimentatore esterno perché gli altri connettori USB non prevedono questi PIN.

Il connettore tipo C prevede due gruppi di 12 pin (12 sopra, 12 sotto) per permettere la reversibilità ma rimanere polarizzato. Oltre a questo prevede i pin dell'USB 2.0 e il supporto all'OTG (On The Go USB) ed MHL; la connettività MHL consente l'interfacciamento alle porte HDMI mediante i due fili MHL+ e MHL- che si attestano ai contatti 2 (D-) e 3 (D+), utilizzando il pin 4 come CBUS collegato all'omonimo della presa HDMI. Come mostra l'immagine in questo

riquadro, oltre ai classici D+ (Dp) e D- (Dn) sdoppiati e disposti in modo che qualunque sia il verso d'inserzione del connettore combacino con la presa sul dispositivo, troviamo due linee ad altissima velocità (SuperSpeed Lane 1 e 2), due contatti per l'utilizzo di altri protocolli (SBU 1 e 2), due linee di alimentazione a 5 volt, che servono una per l'alimentazione di piccoli device o "cavi attivi" contenenti, cioè, dell'elettronica, e l'altra per erogare corrente a dispositivi che richiedono potenze più elevate di quanto le USB standard non consentano. Vi sono poi due contatti, chiamati CC1 e

dante il resistore R7 al Data-Out (DO) della stessa U2, che a sua volta è provvisto di resistore di pull-up. Come indicato dal costruttore dell'FT4232, la EEPROM non è necessaria se si vuole utilizzare l'integrato nella modalità predefinita, vale a dire da semplice convertitore USB/4 seriali standard; serve invece se si desidera configurare il componente per ottenere le modalità speciali come la combinazione i seriali standard e JTAG, BitBang e via di seguito, perché in questo caso bisogna configurare l'unità da computer (spiegheremo più avanti come farlo...) e memorizzare nella EEPROM la relativa configurazione, affinché l'FT4232 all'avvio operi come voluto.

Inoltre nella EEPROM viene scritto l'ID USB e il nome che si desidera eventualmente assegnare alla periferica e che verrà mostrato dal Personal Computer una volta installato il circuito; tale nome è, per impostazione predefinita, generico, ma è possibile personalizzarlo, assegnandogli per esempio una sigla, un brand o un nome di periferica a piacimento: ad esempio quello del dispositivo in cui eventualmente si va a integrare il nostro circuito. L'integrato genera il clock necessario alle temporizzazioni interne mediante un oscillatore incorporato, il cui funzionamento è scandito dal quarzo Q1, ausiliato dai condensatori C5 e C6; tramite un PLL viene poi stabilita l'esatta frequenza di clock.

L'alimentazione del circuito prevede varie possibilità: chiudendo JUSB si preleva direttamente dalla porta USB dell'host, mentre volendo un'alimentazione separata è possibile -previa apertura

di detto ponticello- fornirla attraverso CN2; più esattamente, si applica fra VCC e GND. In tutti i casi le tensioni necessarie al funzionamento dell'FT4232 vengono ricavate da un regolatore integrato LDO a basso rumore siglato U3: si tratta di un NCP161ASN330T1G della ON Semiconductor, che fornisce in uscita 3,3V ben stabilizzati partendo dalla tensione VIN, ovvero dai 5V della porta USB o dall'alimentazione VCC, che dev'essere anch'essa a 5 volt. Il regolatore lavora nella configurazione canonica, con il pin di enable fisso a VIN (quindi è sempre attivo) e le linee di ingresso e uscita ben filtrate da condensatori elettrolitici e non polarizzati; all'uscita, quindi sulla linea 3V3, preleviamo la tensione che accende il LED LD1 (utilizzato come spia della presenza dell'alimentazione) e che, tramite due filtri L/C (quindi del secondo ordine) raggiunge le linee VPHY e VPLL che alimentano, rispettivamente, il pin 4 (interfaccia fisica USB, che va alimentata separatamente) e il pin 9 (alimentazione del PLL che ricava il clock) dell'FT4232. L'utilizzo di due filtri distinti è stata voluta per separare le alimentazioni pur partendo da un unico alimentatore, allo scopo di evitare interferenze fra le due.

I 3,3V all'uscita del regolatore U3 alimentano direttamente il pin 50 (VREGIN) dell'FT4232 e se, il jumper J3V3IO risulta chiuso, anche la sezione I/O lato seriali; laddove si desideri far lavorare il lato delle seriali con una tensione differente, bisogna aprire detto jumper e fornire la tensione per gli I/O tramite il contatto VIO. Restando in tema, le linee di I/O dell'integrato sono 5V tolerant, quindi utiliz-

CC2, che rappresentano il canale di configurazione del sistema, con il quale è possibile settare alcune modalità di trasporto dei dati.

Le linee dati ad alta velocità, che sono poi quelle che consentono di raggiungere il data-rate atteso, fanno capo ai due doppi, in particolare per la Lane 1:

- RX1-/RX1+ = doppi differenziale
- TX1-/TX1+ = doppi differenziale

mentre per la Lane 2 i doppi sono:

- RX2-/RX2+ = doppi differenziale
 - TX2-/TX2+ = doppi differenziale
- Queste due linee consentono

la comunicazione full-duplex, aumentando la velocità di scambio dei dati raggiungibile.

Come accennato l'USB C permette di erogare potenze consistenti, tuttavia fino a un certo limite la cosa avviene a livello hardware, mentre se si desidera eseguire il source/sink di oltre 15W o avere un ruolo di alimentazione e dati differente, occorre negoziarlo tra DFP e UFP. Ciò si ottiene con un protocollo seriale "Power Delivery" 1-wire (non su USB).

USB PD è ciò che "sblocca" il cavo USB Type-CTM e può consentire di ottenere fino a 100W di potenza per l'alimentazione.



Rx1n/Rx1p = Differential Pair 1
Tx1n/Tx1p = Differential Pair 1 } SuperSpeed Lane 1
Rx2n/Rx2p = Differential Pair 1
Tx2n/Tx2p = Differential Pair 1 } SuperSpeed Lane 2
SBU1,2 = Side Band Use (Alternate Modes)
Dp, Dn = USB 2.0 High Speed Data Lines
VBUS = 5.0V to 20V (USB Power Delivery)
VCONN = 5.0V on CC line for Active Cables
CC1, CC2 = Configuration Channel

zate come input possono accettare livelli TTL 0/5V anche se la VIO desiderata è impostata a 3,3 volt. Il pin VREGIN alimenta il regolatore interno dell'FT4232, che il costruttore ha previsto per soddisfare il fabbisogno della sezione di uscita seriale, ossia del core: la relativa uscita, facente capo al pin 49 (VREGOUT) fornisce 3,3 volt che ben filtrati dai condensatori C2, C3, C16, C17 e C18 raggiungono i pin 12, 37 e 64 che corrispondono all'ingresso di alimentazione Vcore.

Quindi, riepilogando, l'FT4232 prevede linee di alimentazione completamente distinte per:

- transceiver USB/seriale facente capo a VPHY, da alimentare con regolatore separato e filtro a prescindere dal fatto che questo (U3) riceva la tensione di ingresso dal connettore USB o da VCC;
- PLL, da alimentare a 3,3V con filtro di disaccoppiamento dal resto delle alimentazioni;
- core (cioè l'**USB Protocol Engine**) da alimentare esclusivamente a 3,3 volt utilizzando il regolatore interno;
- I/O, alimentabile sia con i 3,3V ricavati dall'U3, sia da una tensione in arrivo dall'esterno mediante il contatto VIO del connettore CN2;
- regolatore interno, alimentato tramite il pin VREGIN, sempre a partire da ciò che fornisce il regolatore U3.

Naturalmente tutte le alimentazioni dell'integrato, pur distinte tra loro, sono riferite a massa, ovvero ai pin 1, 5, 11, 15, 25, 35, 47, 51, che sono la GND di alimentazione; l'integrato FT4232 ha poi un pin

di massa AGND (analogica) relativa al transceiver USB, che nel circuito stampato viene mantenuta separata per ragioni di immunità ai disturbi e per garantire una comunicazione più precisa anche a elevati data-rate.

Nel caso si opti per l'alimentazione esterna, l'alimentatore del caso dovrà fornire 5Vcc e una corrente di circa 60 milliampere.

Bene, con questo abbiamo concluso l'analisi dello schema elettrico e possiamo passare al lato pratico del progetto.

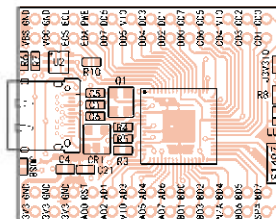
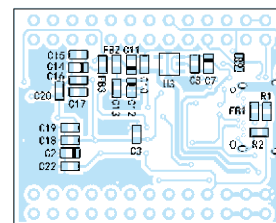
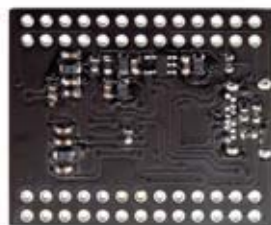
REALIZZAZIONE PRATICA

La costruzione del nostro convertitore a quattro seriali inizia con la preparazione del circuito stampato, che è a doppia faccia e si ottiene per fotoincisione a partire dalle tracce lato rame scaricabili dal nostro sito www.elettronica.in.it e da stampare su carta da lucido o acetato per ricavare le pellicole del caso.

I componenti utilizzati sono tutti per montaggio superficiale, quindi è giocoforza disporre di quel minimo di attrezzatura richiesto per lavorazioni del genere, ossia un saldatore da una trentina di watt a punta molto fine, del filo d lega saldante da non più di 0,5 mm di diametro, della pasta fluxante e una treccia dissaldante (quest'ultima utile per rimuovere lo stagno in eccesso laddove abbiate ecceduto nella saldatura unendo due contatti attigui) oltre che una pinzetta per facilitare il posizionamento dei componenti e una lente d'ingrandimento che aiuterà la vostra vista nelle verifiche finali.

Elenco Componenti:

- R1: 15 kohm (0603)
- R2, R6, R9, R10: 10 kohm (0603)
- R3: 12 kohm 1% (0603)
- R4, R5: 10 ohm (0603)
- R7: 2 kohm 1% (0603)
- R8: 220 ohm (0603)
- CR1: PGB102ST23
- C1: 10µF 6.3 VL ceramico (0603)
- C3, C4, C8, C10, C13, C15, C16, C17, C18, C19, C20, C21, C22: 100 nF ceramico (0603)
- C9: 10 nF ceramico (0603)
- C2, C7, C11, C12, C14: 4,7 µF 6.3 VL tantalio (2012)
- C5, C6: 22 pF ceramico (0603)
- FB1, FB2, FB3: Induttore multistrato 600mA (0603)
- U1: FT4232HQ
- U2: 93LC56BT-E/OT
- U3: NCP161ASN330T1G
- Q1: Quarzo 12 MHz (3.2x2.5mm)
- LD1: LED verde (0603)
- USB1: Connettore USB TypeC da CS
 - Strip maschio 2x13 vie (2 pz.)
 - Circuito stampato S1497 (36x29 mm)



Per la saldatura dei componenti potete anche utilizzare, in luogo del saldatore, una stazione ad aria calda, regolandola su una temperatura di circa 300 °C; in questo caso dovreste dirigere il getto sui terminali fino a quando la lega saldante non brilla, allorché lo rimuoverete. Una volta in possesso del circuito stampato iniziate il montaggio posizionando, sulle piazzole spalmate di pasta flussante, l'integrato FT4232, quindi stagnandone due piedini agli angoli per fermarlo; procedete poi saldando un piedino alla volta, prestando attenzione a non eccedere con la lega saldante per non unire piazzole e pin attigui. Il flussante dovrebbe aiutarvi in questo, perché aiuta lo stagno a posizionarsi sulle piazzole e sui pin corrispondenti. Sistemato l'U1 passate al regolatore di tensione e poi alla memoria U2, quindi saldate i componenti passivi, prestando attenzione alla polarità dei condensatori elettrolitici e a quella del LED; quest'ultimo dev'essere scaldato il meno possibile, onde evitare che si deformi la resina da cui passa la luce. Il TVS non ha polarità, ossia è reversibile, quindi lo potete orientare come meglio credete. Terminate il montaggio con il connettore USB-C, che va posizionato come indicato nel piano di montaggio, utilizzando il solito flussante e facendo attenzione a non cortocircuitare contatti vicini con la lega saldante, dato che i terminali sono a passo molto fitto. Saldare bene anche le alette di ancoraggio, che conferiranno robustezza al con-

nettore evitando che si strappi al primo sforzo. Completate le saldature verificate il tutto con l'aiuto della lente d'ingrandimento e se tutto è a posto potete utilizzare il circuito, dato che non richiede programmazione né tarature preliminari; l'unica accortezza richiesta è impostare correttamente i ponticelli previsti, che si devono realizzare con delle gocce di lega saldante. Più esattamente dovreste chiudere JUSB se volete che il circuito sia alimentato dalla connessione USB (altrimenti lo lascerete aperto e alimenterete il tutto dal contatto VCC) e J3V3IO se interfacerete il circuito con dispositivi funzionanti a 3,3V e comunque che lavorano con livello 0/3,3V.

Fate attenzione che alcuni componenti vanno saldati da un lato del circuito stampato e altri dal lato opposto, quindi riferitevi alle relative tracce lato rame per sapere dove e come posizionarli.

UTILIZZO DEL CONVERTITORE

Per poter utilizzare il circuito occorre, innanzitutto, trovare e installare i driver per il proprio sistema operativo; questi si scaricano dal sito web della FTDI, www.ftdichip.com. I driver vengono concessi in uso dalla FTDI con la formula royalty-free e consentono la creazione di Virtual Com Port (VCP) e connessione Direct (D2XX); vanno bene nella gran parte delle applicazioni. Dal sito, accedendo alla sezione Download, dovreste quindi scaricare i driver VCP, ovvero D2XX se volete un tipo di connessione

diverso da quello canonico consistente nell'emulazione delle COM. Sul sito della FTDI vengono inoltre rese disponibili le DLL dedicate Windows per la realizzazione di applicazioni con convertitori USB to JTAG, USB to SPI e USB to I²C.

Una volta completata l'installazione dei driver, nelle risorse del computer (in Windows, sotto Gestione Risorse o Gestione Dispositivi) troverete le quattro nuove COM installate.

Da questo momento, collegando il circuito mediante un classico cavo USB-A/USB-C il circuito si accenderà e verrà configurato, allorché l'FT4232 creerà la configurazione e la scriverà nella EEPROM on-board.

Vista la molteplicità di driver e configurazioni disponibili è opportuno chiarire il discorso sui driver: quelli siglati VCP sono per l'emulazione di COM virtuali e quindi per l'applicazione predefinita e basilare dell'FT4232, nella quale si potrebbe fare a meno della EEPROM, a meno di non voler cambiare l'ID USB predefinito e il nome della scheda che si vedrà dal computer. Quindi caricando il driver VCP (Virtual Communication Port) si installano nel sistema operativo quattro porte COM virtuali, come è consueto.

Per l'utilizzo nelle modalità avanzate servono, invece, i driver D2XX, il cui funzionamento andrà abbinato alle DLL fornite dalla FTDI, da richiamare nel software applicativo che utilizzerà il converter FT4232; per rendere più semplice l'impostazione e la configurazione delle modalità speciali, la FTDI rende disponibile un programma chiamato **FT_PROG**, scaricabile in rete dal link https://www.ftdichip.com/Support/Utilities.htm#FT_PROG. L'interfaccia utente di questo programma, visibile nella **Fig. 1**, compare lanciando l'eseguibile e consente di impostare la modalità desiderata con estrema semplicità, senza scrivere comandi ma soltanto selezionando (nel riquadro di destra Property) la modalità. Una volta scelta, la relativa configurazione sarà inviata dal computer alla EEPROM della scheda tramite la connessione USB. Notare che una volta avviato l'eseguibile (FTDI - FT Prog) bisogna, per poter operare sul circuito, cliccare sulla tab EEPROM e qui, sul menu DEVICE, andare a impartire il comando **Scan and Parse** (scorciatoia da tastiera F5) che eseguirà una scansione delle porte USB e dopo qualche istante visualizzerà nella finestra di dialogo le informazioni presenti in memoria. Più esattamente, nella sezione **Device Tree** apparirà la lista dei dispositivi trovati con le porte configurate; nel riquadro a destra si potranno modificare le impostazioni, che

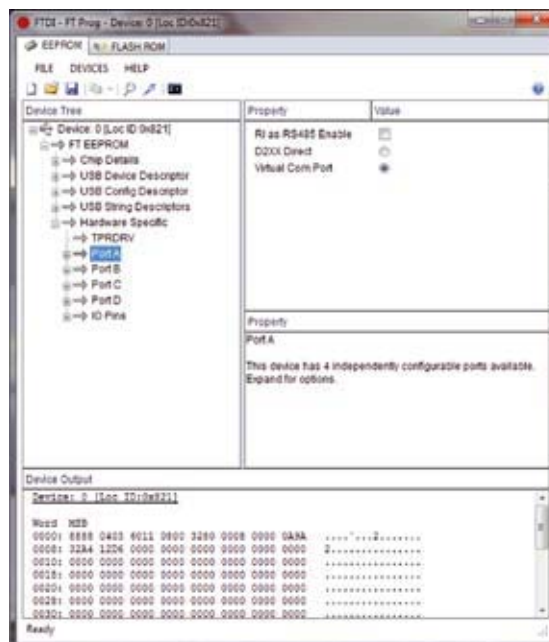


Fig. 1
Console di
configurazione su
PC.

diverranno poi operative cliccando sul pulsante della Toolbar contraddistinto dall'icona a floppy disk.

Tra l'altro, notate che tra le icone della toolbar c'è quella che consente di aprire una finestra

CONCLUSIONI

Il convertitore da USB a quattro seriali che vi abbiamo proposto è una soluzione modulare e molto versatile per gestire da un PC mediante una sola porta USB fino a quattro connessioni seriali UART, implementando anche particolari modalità e supportando a livello fisico le comunicazioni RS232, RS422 ed RS485.

Alimentato via USB consente di realizzare convertitori autoalimentati, mentre incorporato in altri circuiti o sistemi esistenti può prelevare da questi ultimi l'alimentazione, mantenendo in comune con l'host la sola porta lato USB. ■



Cosa occorre?

La breakout board presentata in questo articolo (cod. FT1497) è disponibile a Euro 35,00. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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



Costruisci e programma un ROBOT!



cod. KSR21
€ 42,00

Robot RICCIO - in kit

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



cod. KSR19
€ 35,00

Robot 5 in 1 con codifica meccanica - in kit

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

Robot TOBBIE II con MICRO:BIT in kit

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



cod. KSR20
€ 89,00



Robot TOBBIE II in kit

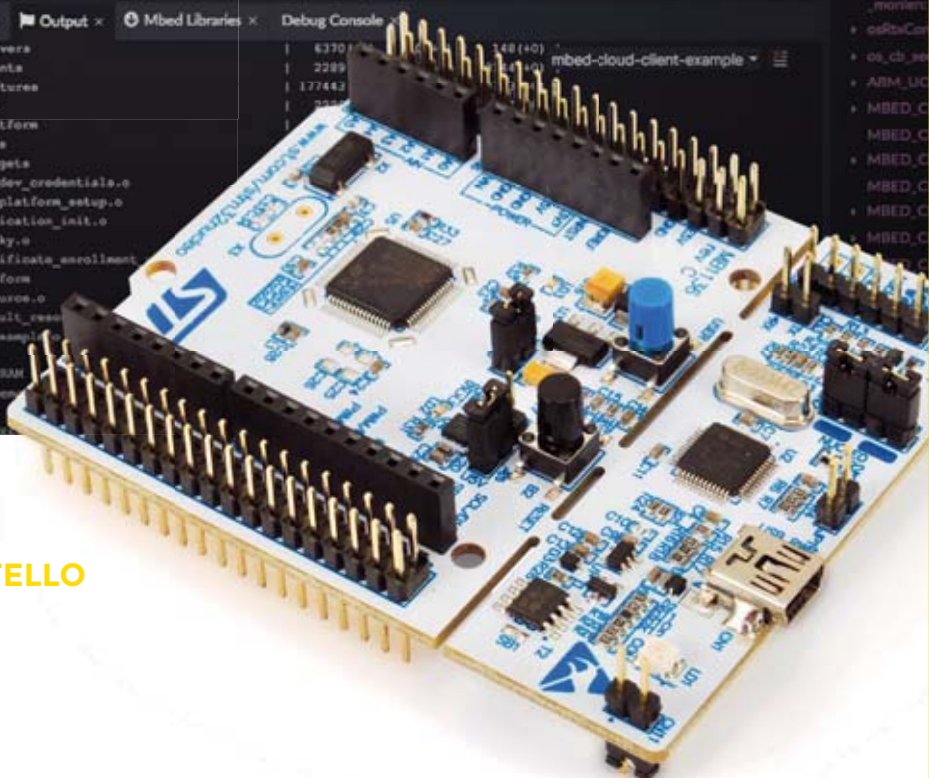
Robot in kit a 6 gambe. Può funzionare in due differenti modalità: "Seguimi" e "Esplora". Nella modalità "Seguimi" Tobbie è in grado di seguire qualsiasi cosa gli si avvicini. Quando funziona in modalità "Esplora", grazie al rilevamento automatico tramite sensori infrarossi, è in grado di evitare gli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

cod. KSR18
€ 37,00

STM32 NUCLEO

dell' Ing.
MIRCO SEGATELLO

Conosciamo e impariamo a utilizzare le schede di prototipazione di ST, anche nell'ottica della programmazione tramite l'IDE di Arduino.



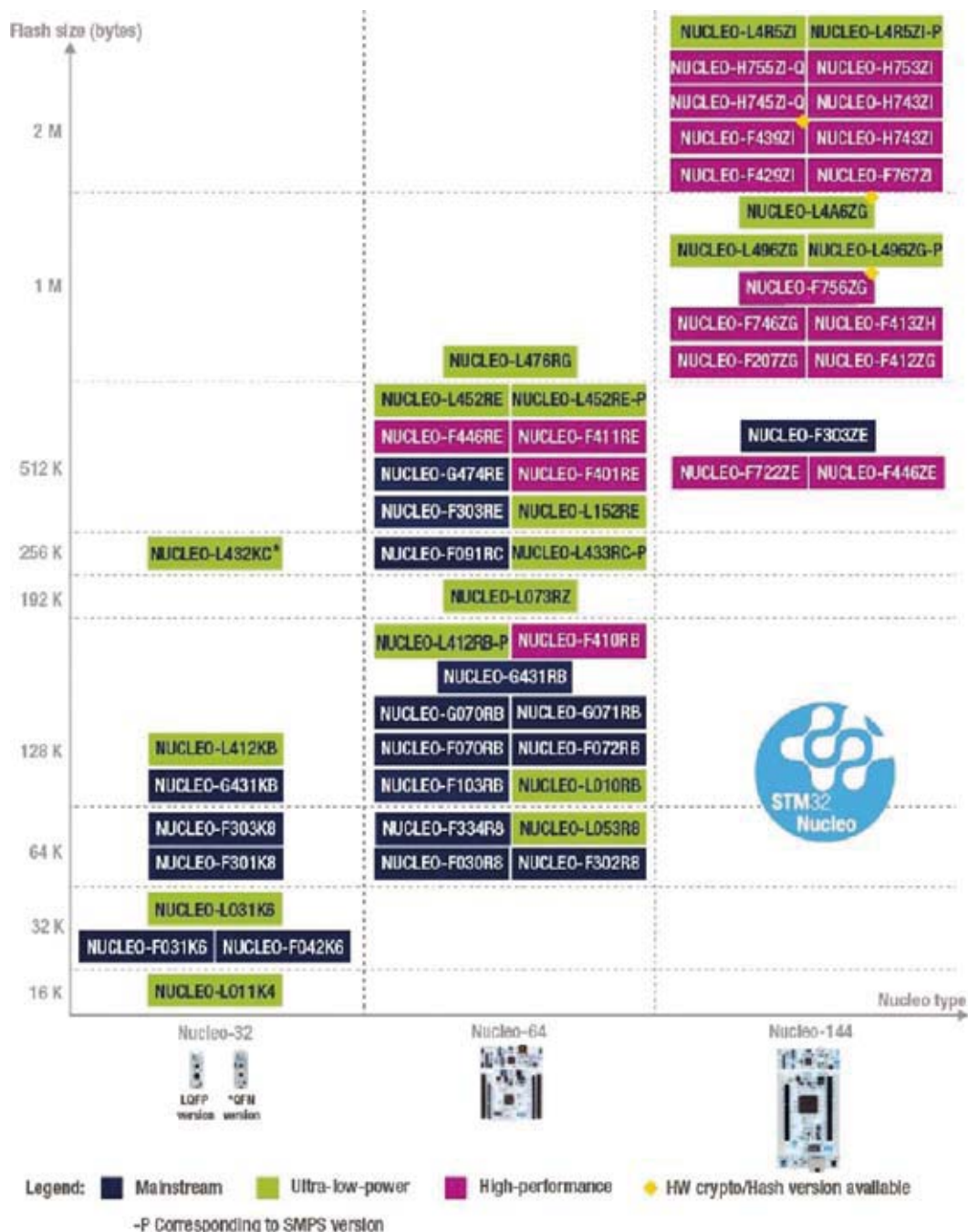
S

TMicroelectronics è un'azienda italo-francese che si occupa della produzione di componenti elettronici, nata nel 1987 dalla fusione di SGS (Società Generale Semiconduttori) Microelettronica e della Thomson Semiconducteurs; molti di voi la conosceranno.

Attualmente è uno dei più grandi produttori di componenti elettronici destinati ai settori della telefonia cellulare, automotive, computer ed elettronica di consumo, per quanto riguarda sia dispositivi MEMS che microcontrollori e microprocessori, tra i quali ricordiamo il famoso TS68000, versione del Motorola 68000 che equipaggiava i computer Macintosh, Amiga e Atari negli anni '90 del secolo scorso.

Negli ultimi periodi un'importante campagna promozionale ha visto la ST consegnare un milione di kit di sviluppo equipaggiati con i propri microcontrollori STM32 e 100.000 kit con

Fig. 1
La famiglia di
schede STM32
Nucleo.



MCU della famiglia STM8, dando la possibilità ai progettisti di tutto il mondo di conoscere meglio i prodotti ST. Sempre in quest'ottica la ST ha messo sul mercato una serie di schede elettroniche basate su microcontrollori a 32 bit (STM32) di fascia media ed alta: si tratta di una famiglia di microcontrollori Flash a 32 bit basati sul processore ARM Cortex-M che combinano elevate prestazioni con bassi consumi per funzionalità in tempo reale. Per pochi euro, le schede di sviluppo STM32

Nucleo rendono accessibile a chiunque, dai maker a sviluppatori indipendenti o piccole società di programmazione, tutto ciò di cui hanno bisogno per iniziare a sviluppare prototipi con la famiglia STM32 a 32 bit. I produttori di microcontrollori hanno capito l'importanza e il successo riscosso dalla piattaforma Arduino nell'avvicinare al mondo dell'elettronica migliaia di persone, le quali associano il nome Arduino alla semplicità e alla rapidità di passaggio da un'idea a una soluzione funzionante.

Per abbracciare anche il folto numero dei maker e sempre nell'ottica della semplicità, la ST ha fatto l'occhiolino ad Arduino copiando alcune soluzioni come ad esempio il form factor e la facilità di utilizzo delle board, non a caso le schede ST-Nucleo dispongono degli stessi connettori che troviamo in una scheda Arduino UNO, oltre al connettore proprietario ST, che è stato chiamato "morpho"; inoltre integrano un programmatore/debugger ST-Link che fa da bridge tra il microcontrollore ed il PC rendendo immediato l'utilizzo di queste schede. Tutte le schede della famiglia ST-Nucleo dispongono di alcune caratteristiche base:

- header compatibili con Arduino Uno Rev3;
- header STMicroelectronics Morpho (per l'accesso a tutti i pin I/O);
- on-board ST-LINK/V2-1 debugger/programmer;
- alimentazione da USB o sorgente esterna (3,3V÷12V);
- due LED di sistema;
- pulsanti utente e reset;
- USB re-enumeration con tre interfacce USB supportate;
- porta COM virtuale
- mass storage (USB Disk drive) per la programmazione drag'n'drop;
- porta di debug.

La famiglia STM-32 Nucleo annovera oltre cinquanta board suddivise in tre grandi categorie in base alla dimensione della scheda: Nucleo-32,

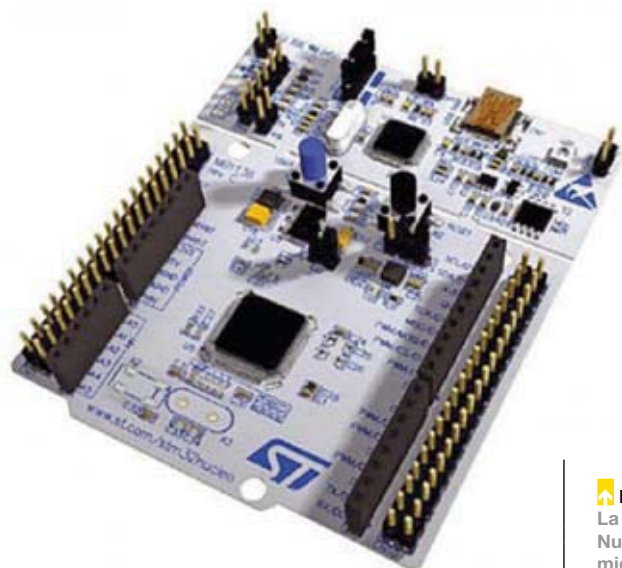


Fig. 2
La scheda STM32-Nucleo con microcontrollore L152RF.

nucleo-64, Nucleo-144; sono illustrate nella **Fig. 1**. L'elenco di tutte le board nucleo può essere trovato sul web, all'indirizzo <https://www.st.com/en/evaluation-tools/stm32-nucleo-boards.html>.

Tra le numerose schede disponibili, in questo articolo ci occuperemo di una versione in particolare, quella equipaggiata con il microcontrollore STM32L152RET6 venduta dalla Futura Elettronica con codice 7309-NUCLEOL152RE a poco meno di 12 euro, che si presenta come visibile in **Fig. 2**. Le caratteristiche tecniche di questa scheda sono di tutto rispetto:

- CPU: ARM Cortex-M3 STM32L152RET6 a 32 MHz;
- Memoria: 512 kbyte di Flash;

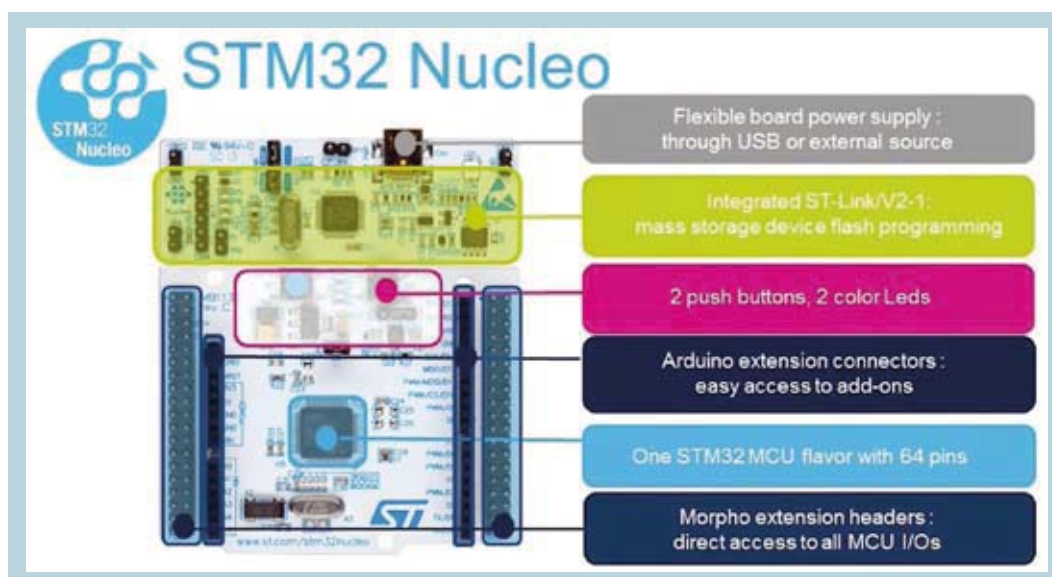
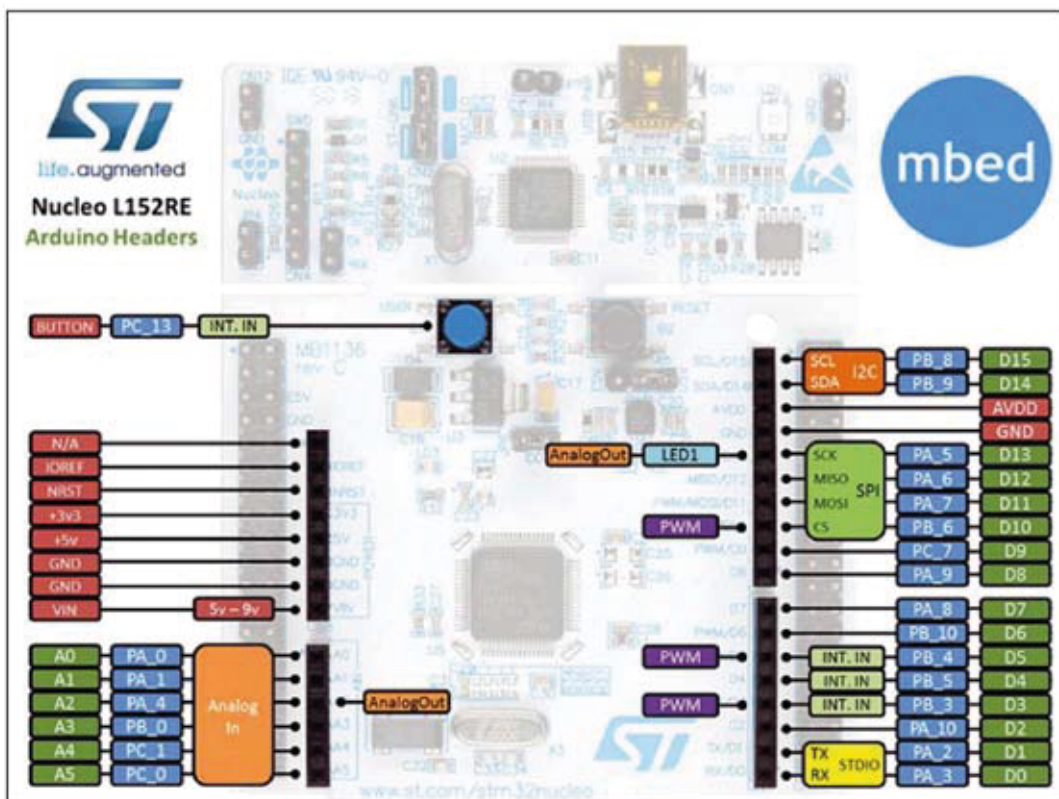


Fig. 3
Componenti della scheda Nucleo.

➔ **Fig. 4**
Connettore Arduino
compatibile.



- SRAM: 80 kB;
- EEPROM 16 kB;
- ADC: 12 bit ADC 1Msps fino a 10 canali;
- DAC: 12bit - 2 canali;
- RTC, Timers (11), I²C (2), USART (5), SPI (8), DAC (2), USB 2.0 Full-Speed;
- connettori per Arduino Uno Rev3 e Morpho STMicroelectronics
- tutti i pin (51 GPIO) del microcontrollore STM32L152RET6 accessibili
- mbed-enabled
- on-board ST-LINK/V2-1 debugger/programmer with SWD connector
- alimentazione: da connessione USB oppure

- esterna, con tensione di 3,3 - 5 - 7 o 12 Vcc;
- tre LED di stato: comunicazione USB (LD1), utente (LD2), alimentazione (LD3);
- due pulsanti: USER e RESET;
- USB re-enumeration capability: Porta COM virtuale, memoria di massa, Porta Debug

➔ **Fig. 5**
Installazione driver
dopo aver inserito
la scheda.



➔ **Fig. 6**
Verifica delle
periferiche
installate.



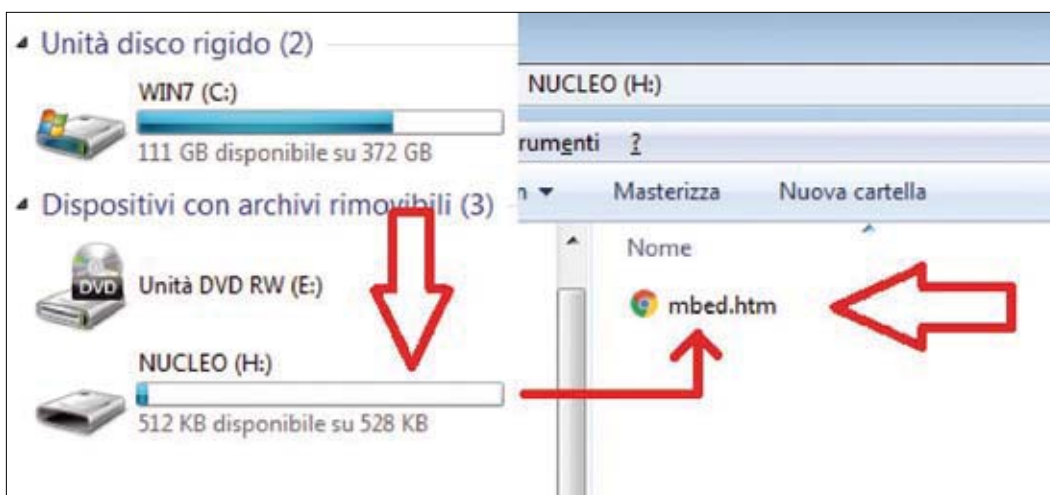


Fig. 7
STM32-
Nucleo
come mass
storage.

- supporto fornito da un'ampia scelta di ambienti di sviluppo integrati (IDE): tra cui IAR, Keil, IDE GCC-based.

La scheda è suddivisa in due zone di cui una con il microcontrollore ed una seconda parte chiamata ST-Link necessaria alla comunicazione con il PC. Questa seconda parte permette al microcontrollore di comunicare con il PC per essere programmato ed eseguire il debug.

Questa interfaccia viene vista dal PC sia come mass-storage (la classica chiavetta), sia (contemporaneamente) come seriale virtuale.

La configurazione predefinita dei vari jumper va già bene per iniziare, pertanto il connettore CN2 e JP6 hanno entrambi i jumper inseriti, JP5 (USB) ha il jumper posizionato verso U5V (**Fig. 3** e **Fig. 4**).

Verificato lo stato dei jumper potete provvedere ad alimentare la scheda tramite la porta USB tramite un cavetto con connettore USB lato PC e connettore MiniUSB lato scheda. Non appena inserita, Windows installerà i driver, normalmente se state già usando l'ambiente Arduino verranno installati sia i driver della seriale virtuale che i driver per il

mass-storage, più che sufficiente per il nostro utilizzo, in Windows dovreste ottenere le schermate visibili nel **Fig. 5** e **6**. Nella scheda è pre-caricato un programma che fa lampeggiare il LED di sistema, azionando il pulsante B1 si potrà modificare la frequenza di lampeggio.

PROGRAMMAZIONE

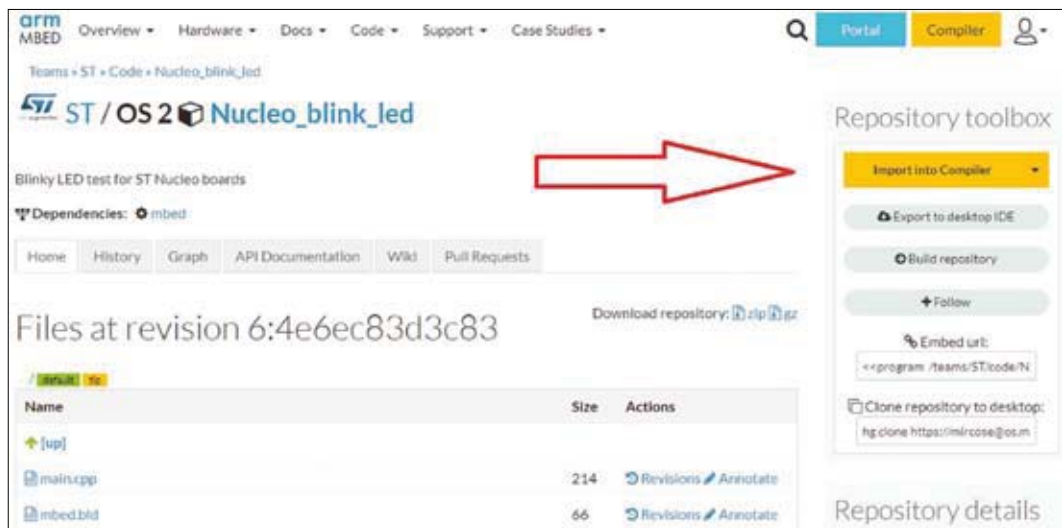
A questo punto viene da chiedersi, come si programmano? E' sufficiente aprire l'HD virtuale di nome NUCLEO da risorse del computer ed aprire il link contenuto all'interno *mbed.htm* (**Fig. 7**), accederete alla pagina di riferimento di quella particolare scheda, in cui troverete oltre a tutte le caratteristiche tecniche, anche tutto il supporto necessario alla programmazione, davvero comodissimo.

Gli ambienti di sviluppo che supportano gli STM32 sono diversi ma l'ambiente di sviluppo STM32Cube mette a disposizione un set di software tools (STM32CubeMX, STM32CubeProgrammer or STM32CubeMonitor) ed embedded software libraries specificatamente pensato per lo sviluppo veloce e semplificato di applicazioni con STM32. STM32cube dispone di una parte di sviluppo con

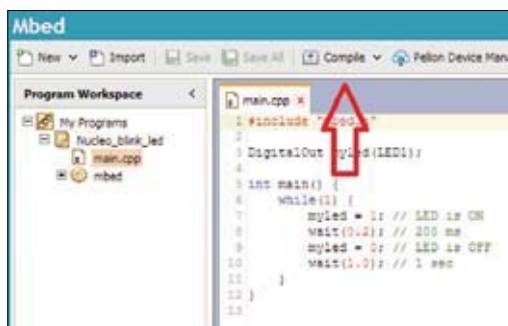


Fig. 8
Il portale
mbed.

➔ **Fig. 9**
Apertura
dell'esempio
Nucleo_blink_led.



➔ **Fig. 10**
Mbed con l'esempio
Nucleo_blink_led.



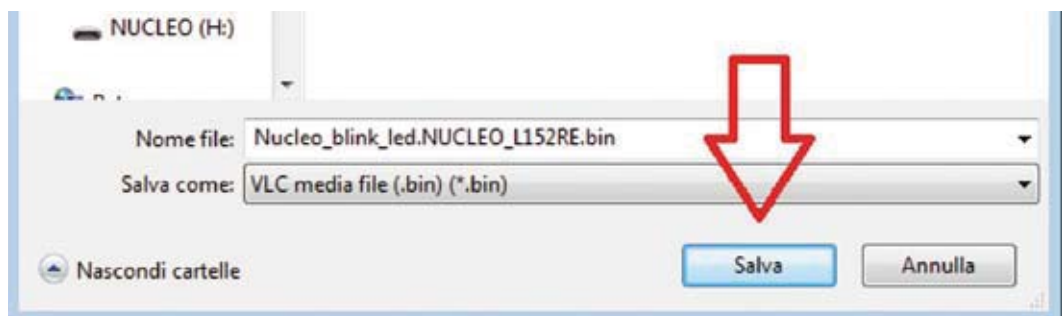
interfaccia grafica che semplifica notevolmente la generazione del codice permettendo di configurare rapidamente pin, multiplexing, timer ed altro. Questo ambiente di sviluppo è però di livello professionale e visto che tutte le board nucleo sono mbed enabled ci focalizzeremo su questo sistema. La società inglese ARM ha creato un portale di nome mbed che mette a disposizione tutti gli strumenti necessari alla programmazione di MCU basate su architettura 32-bit ARM Cortex-M, quindi non solo le schede ST ma moltissime altre sono supportate. Attorno a Mbed si è sviluppata una

vasta comunità di sviluppatori e una rete di partner hardware e prodotti che servono il mercato professionale e makers.

Per poter usare il portale mbed (os.mbed.com) è necessario iscriversi accedendo al menu in alto a destra come visibile in **Fig. 8**.

Terminata l'iscrizione potete accedere al compilatore, che essendo on-line non richiede alcuna installazione; sempre allo stesso compilatore potete accedere dalla pagina html aperta in precedenza tramite il link [mbed.htm](#); in questo caso il compilatore sarà già configurato con la board corretta. Come primo esempio è scontato caricare un programma che faccia lampeggiare il LED di sistema, la via più semplice è accedere alla solita pagina aperta con il link ([mbed.htm](#)), posizionarsi nella sezione Example programs e cliccare sull'esempio **Nucleo_blink_led** e successivamente su *Import into Compiler* come visibile nelle **Fig. 9** e **Fig. 10**. Come potete vedere dalla **Fig. 10** la struttura del listato non è tanto dissimile da quella che normalmente si trova nel mondo Arduino, cambiano invece le istruzioni utilizzate.

➔ **Fig. 11**
Salvataggio
del file .bin



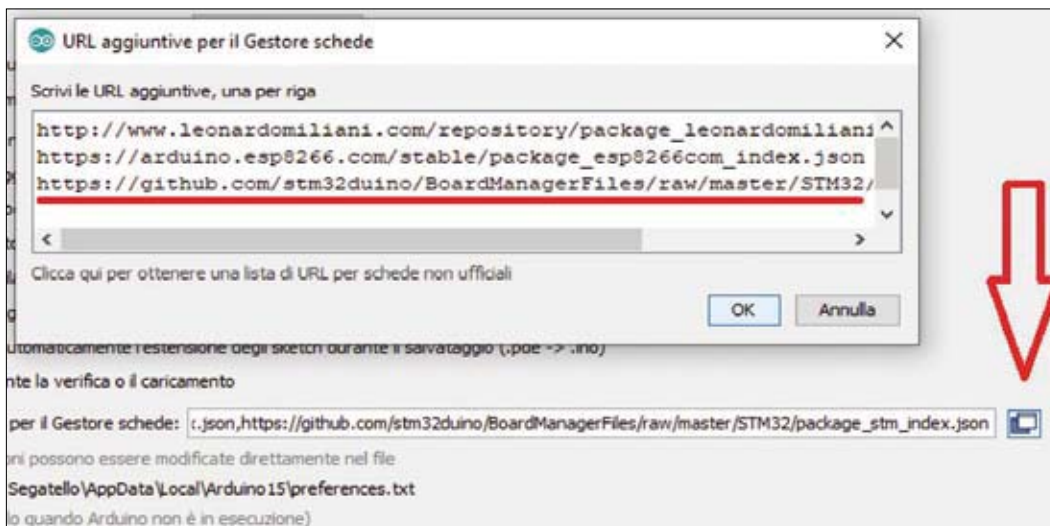


Fig. 12
Aggiunta
delle boards
Nucleo.

A questo punto possiamo cliccare su **Compile** ed attendere che si apra una finestra con la possibilità di salvare il file .bin appena compilato in una cartella a scelta (**Fig. 11**). Sarà quindi sufficiente eseguire una copia incolla di questo file nella cartella NUCLEO affinché avvenga la programmazione e l'esecuzione del programma. Durante il trasferimento del file vedrete il LED LD1 del programmatore STlink lampeggiare rapidamente. Un metodo decisamente particolare per chi arriva dal mondo Arduino ma, in fin dei conti, semplice e veloce. Adesso potete sbizzarrirvi provando a modificare il listato e caricando altri programmi, ce ne sono di tutti i tipi. Esiste anche l'istruzione per far comunicare la board con il PC sfruttando la seriale virtuale, però mbed non dispone di un sistema per la visualizzazione dello scambio dati, pertanto dovreste utilizzare un programma di terze parti come ad esempio Teraterm o PuTTY, in grado di gestire la comunicazione seriale.

Chi è abituato al mondo Arduino potrà trovarsi inizialmente spaesato, ma è sufficiente realizzare alcuni programmi per apprezzare questa nuova modalità, inoltre i numerosi esempi a disposizione facilitano notevolmente il programmatore. Per chi invece non vuole rinunciare all'immediatezza ed alla semplicità del mondo Arduino nessun problema perché, grazie al lavoro della community ed il supporto della stessa ST, è stato realizzato un porting dell'ambiente di programmazione Wiring su piattaforma STM32 utilizzando l'IDE di Arduino. Il progetto prende il nome di STM32DUINO ed è disponibile nel repository all'indirizzo <https://github.com/stm32duino>. Per installare l'estensione per STM32 board, è necessario recarsi nelle impostazioni dell'IDE, alla voce URL aggiuntive per il Gestore schede ed inserire il link: https://github.com/stm32duino/BoardManagerFiles/raw/master/STM32/package_stm_index.json come in **Fig. 12**.

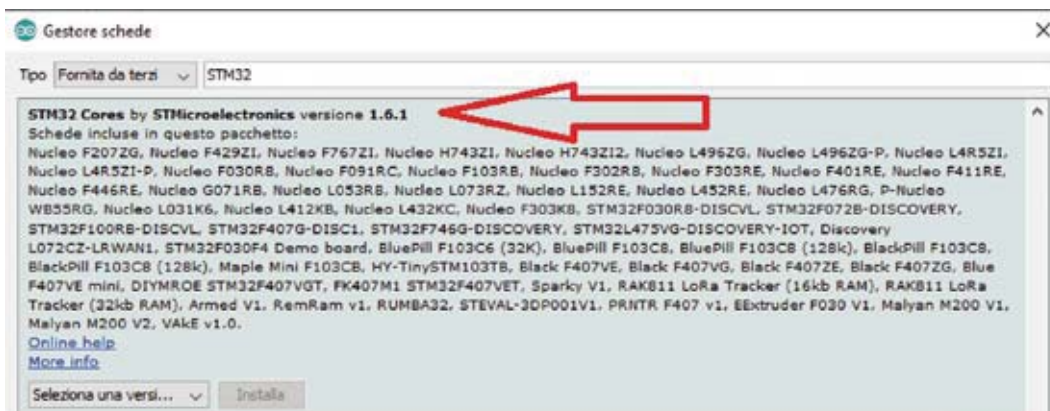


Fig. 13
Installazione
pacchetto
STM32 Cores.

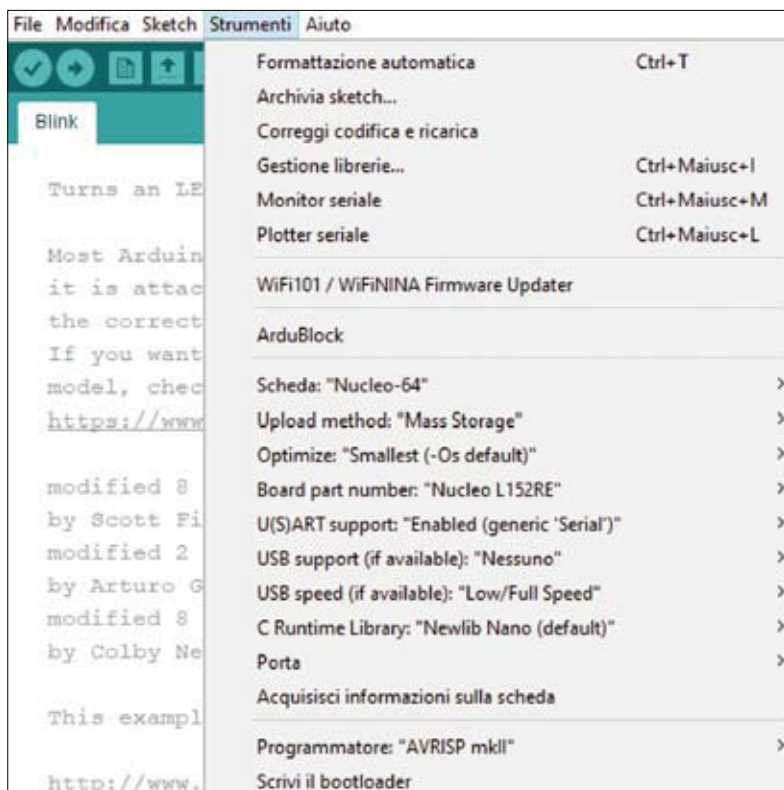


Fig. 14
Selezione della scheda Nucleo L152RE nell'IDE di Arduino.

A questo punto potete recarvi nel gestione schede ed installate il pacchetto STM32 Cores come visibili nella **Fig. 13**. Collegate come al solito la scheda

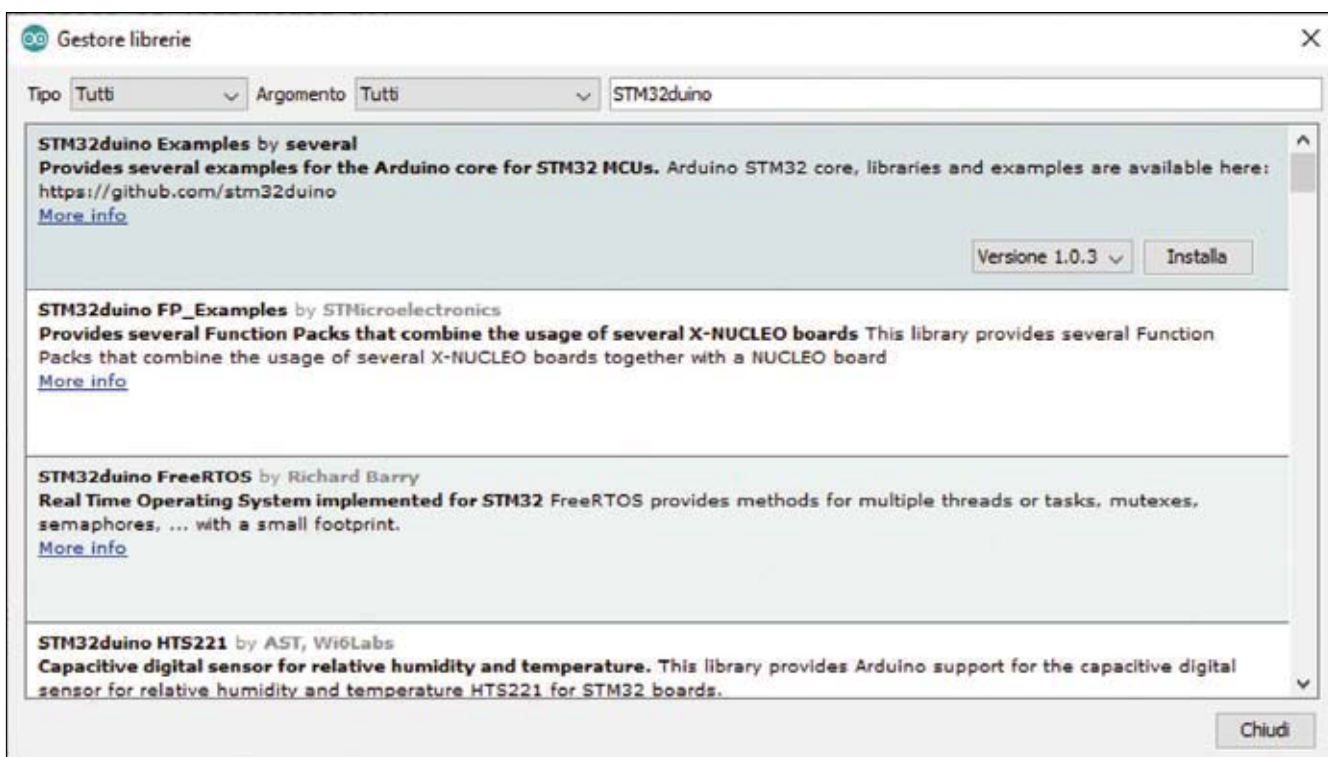
al PC con il cavo USB e dal menu Strumenti settate le impostazioni della scheda come in **Fig. 14**, facendo attenzione a selezionare la corretta porta COM. L'ambiente di programmazione si comporterà come nell'utilizzo di una scheda Arduino UNO, con tutte le classiche funzioni perfettamente integrate, quindi **analogRead()** restituirà un valore a 10bit (anche se l'ADC è a 12 bit) e **Serial.print()** interagirà come al solito con Serial Monitor. Tutte le principali librerie sono state adattate per funzionare con STM32, ulteriori librerie compatibili possono essere scaricate accedendo alla sezione Gestione librerie e facendo una ricerca con il termine STM32duino, verranno elencate tutte le librerie compatibili e vi sarà la possibilità di installare anche degli esempi (**Fig. 15**).

CONFRONTO CON ARDUINO

Viene spontaneo fare un paragone con la scheda Arduino UNO per capire quali siano le analogie e i benefici nel passare ad una scheda STM32-Nucleo; nella **Tabella 1** abbiamo riportato le principali caratteristiche delle due schede. Dal confronto appare evidente un risultato schiacciante per le schede STM32 in quanto decisamente più ricche di periferiche e più veloci nell'eseguire il software, grazie ad un clock a frequenza doppia ed istruzioni a 32bit. Da alcune prove fatte eseguendo calcoli in virgola mobile le schede ST sono risultate

CARATTERISTICA	ARDUINO UNO	STM32-NUCLEO L152RE
Architettura	AVR RISC 8bit	ARM Cortex M3 32bit
Frequenza di Clock	16MHz	32MHz
Memoria Flash	32KB	512KB
Memoria EEPROM	1KB	16KB
Memoria RAM	2KB	80KB
Numero di pin	23	51
Canali ADC	6 (10bit)	10 (12bit)
Canali DAC	0	2 (12bit)
Canali PWM	6 (10bit)	15 (16bit)
RTC interno	NO	SI
Porte	1-UART, 1-SPI, 1-I2C	5-UART, 8-SPI, 2-I2C, 2-I2S
Interfaccia USB	NO	SI
DMA	0	12ch
Timers	3	11

Tabella 1
Confronto tra Arduino UNO e STM32-Nucleo.



sempre le più veloci. Potevamo avere solo alcuni dubbi sulla compatibilità con le shield Arduino ma dobbiamo considerare che quasi tutti i circuito integrati funzionanti a 5V lo sono anche a 3,3V, inoltre la maggior parte dei pin degli STM sono 5V tolerant. Ad esempio, abbiamo testato lo shield LCD (codice 1686-SHIELD-LCD) tramite il portale mbed. In questo caso è possibile utilizzare il sito <https://os.mbed.com/components/> che riporta l'elenco di tutti gli hardware compatibili e relativo esempio di programmazione. Spuntate la voce Text LCD e selezionate Freetronic LCD shield, cliccate su Import program relativamente all'esempio Hello Word e procedete poi con la normale programmazione drag-and-drop.

Anche usando l'IDE di Arduino e l'esempio fornito con la libreria (di sistema) liquid crystal tutto ha funzionato senza problemi, confermando la piena compatibilità hardware e software. Un confronto alla pari si potrebbe fare con le schede MKR basate su MCU Arm Cortex M0 SAM D21, che però visto il formato non utilizzano (se non con appositi adattatori) gli shield di Arduino UNO.

CONCLUSIONI

Viene spontaneo chiedersi perché non usare le schede STM32 al posto di Arduino UNO, esse risul-

tano decisamente superiori in quanto ad hardware ed il costo gioca tutto a loro favore. La piena compatibilità con l'IDE di Arduino permette una facile migrazione verso i microcontrollori ST a 32bit ed il connettore compatibile con le shield Arduino risolve molti dei problemi di cablaggio.

Se dopo aver letto questo articolo avete ancora qualche dubbio sappiate che la Futura Elettronica distribuisce, per qualche euro in più, anche la scheda STM32 Nucleo (codice prodotto 7309-NUCLEOF401RE) equipaggiata con un potentissimo ARM Cortex M4 STM32F401 capace di un clock fino a 84MHz che farà letteralmente "volare" le vostre applicazioni. ■

Fig. 15
Installazione
libreria
aggiuntive
compatibili
STM32.



Cosa occorre?

Le schede Nucleo utilizzate in questo articolo sono disponibili presso Futura Elettronica. La scheda Nucleo F401RE (cod. NUCLEOF401RE) è in vendita a Euro 17,50 mentre la board Nucleo L152RE (cod. NUCLEOL152RE) è disponibile a Euro 11,90. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

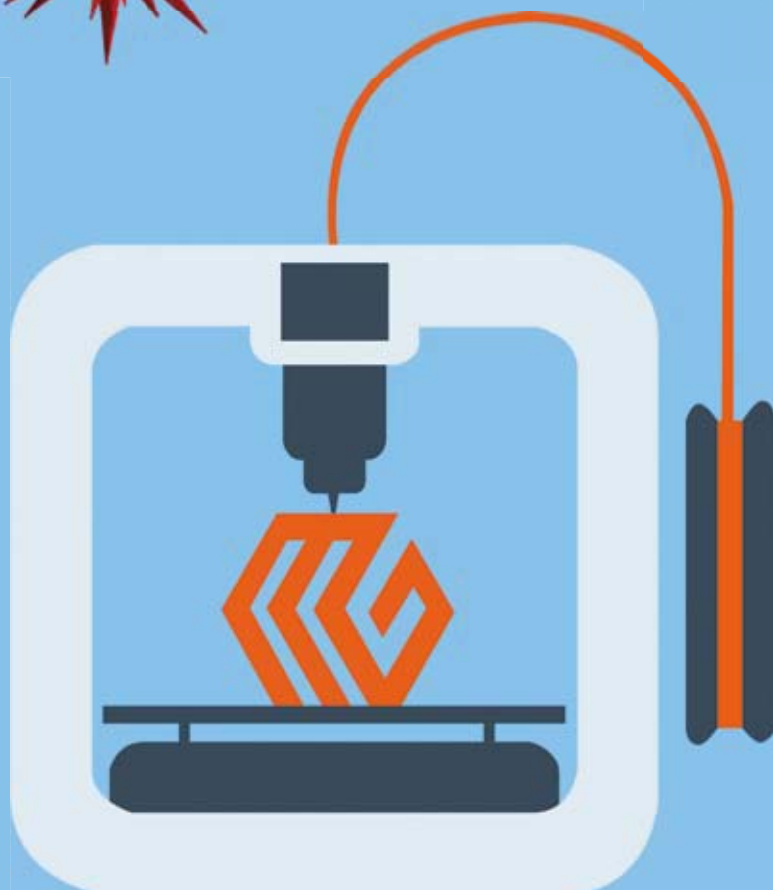
Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 - <http://www.futurashop.it>



Trasformiamo le tue idee in realtà!

Servizio taglio laser e incisione

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



Servizio stampa 3D

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

Informazioni dettagliate su
www.futurashop.it

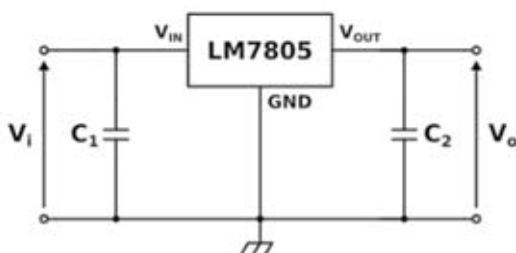
ALIMENTATORI COMMERCIALI STEP DOWN

di **GABRIELE PERETTO**

Abbiamo selezionato e provato per voi alcuni alimentatori DC/DC commerciali di tipo "Buck / Step down"; eccone le conclusioni.

Il nostro impegno e i nostri obiettivi primari sono da sempre la presentazione in anteprima di soluzioni tecnologiche di alto livello, che pur rimangono in una sfera alla portata di tutti gli appassionati di elettronica. Cerchiamo sempre di presentarvi sia nuovi componenti, sia breakout board e schede elettroniche reperibili già pronte in commercio, in grado di risolvere con semplicità e poca spesa molti problemi che comunemente il progettista professionista o l'hobbista si trovano ad affrontare. Tra le schede già pronte oggi disponibili in gran quantità sul mercato, figurano certamente gli alimentatori, specialmente quelli DC/DC in tecnologia switching. Questo perché tutte le schede, o quasi, che continuamente proponiamo oppure che ognuno di voi si appresta a progettare e realizzare, ha bisogno, per poter funzionare, di un'adeguata alimentazione che sia anche in grado di garantire stabilità ed affidabilità. Negli anni anche i sistemi di alimentazione si sono evoluti divenendo via-via più facili da realizzare e di dimensioni più contenute, grazie soprattutto all'elevata integrazione dei componenti e soprattutto, caratteristica fondamentale, sono diventati "più efficienti" e quindi anche più "ecologici" poiché

Fig. 1



utilizzano meglio, l'energia disponibile. Ovviamente, come sempre, bisogna scendere a qualche compromesso; infatti se da un lato i tradizionali alimentatori lineari offrono caratteristiche ottimali in termini di stabilità di uscita e ondulazione residua (ripple) che si riduce praticamente a zero se la fonte di alimentazione primaria è una batteria, il rovescio della medaglia è che tale soluzione porta ad avere un'elevata perdita di efficienza e quindi di potenza dissipata in calore, soprattutto quando vi è un elevato salto di tensione tra ingresso e uscita. Ad esempio se abbiamo a disposizione una tensione di batteria a 12 Vcc e vogliamo prelevare in uscita una tensione di 5 Vcc con un carico da 2A, dovremo mettere in conto che il circuito dovrà dissipare una potenza P_d di:

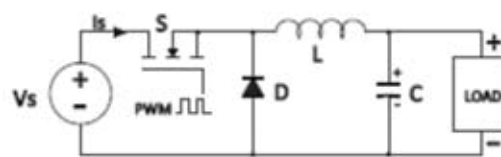
$$P_d = (12-5)V \times 2A = 14W.$$

Questo esempio riguarda il classico regolatore lineare di tensione di tipo serie, proposto nel circuito di **Fig. 1**, che utilizza il canonico stabilizzatore 7805. Se poi il circuito verrà alimentato tramite la rete elettrica di distribuzione domestica (230Vca-50Hz) con un trasformatore più ponte diodi, allora dovremo fare i conti anche con il residuo di alternata (ripple) che può riversarsi ovunque nei circuiti, se non ben dimensionati, soprattutto nelle piste di massa mal distribuite. Per questo, normalmente vengono utilizzati condensatori di livellamento dimensionati in funzione della corrente prelevata che fungono da filtro per ridurre il "rumore" l'ondulazione residua appunto.

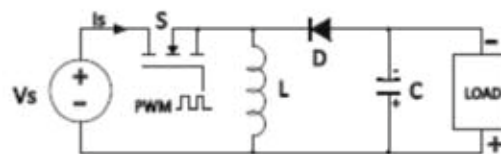
Tutto questo a scapito delle dimensioni del circuito, della necessità di utilizzare dissipatori di notevoli dimensioni e di accettare uno spreco di energia non indifferente.

GLI ALIMENTATORI SWITCHING

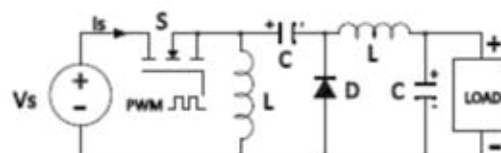
Per superare questi limiti, l'evoluzione tecnologica ha portato allo studio e all'implementazione pratica degli alimentatori cosiddetti "switching" che sono poi quelli a commutazione; a differenza dei regolatori di tensione lineari serie, dove la tensione



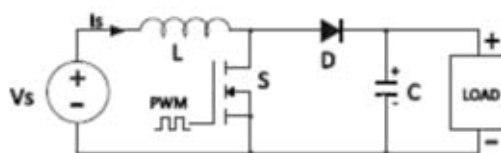
Buck Converter



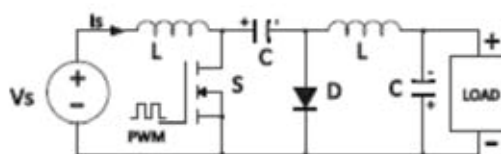
Inverting Buck-Boost Converter



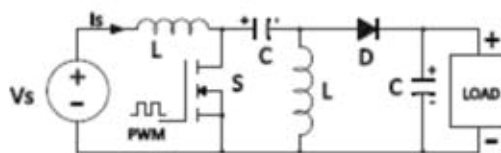
Zeta Converter



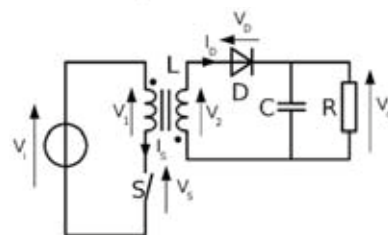
Boost Converter



Cuk Converter



Sepic Converter



Fly back Converter

Fig. 2
Configurazioni dei
converter DC/DC.

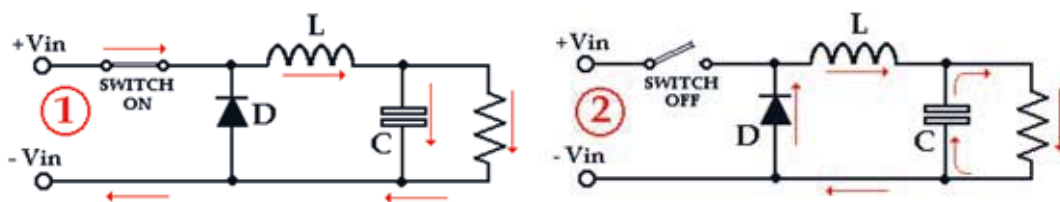


Fig. 3
Principio di
funzionamento
del convertitore
Buck: a sinistra
l'interruttore
è chiuso e a
destra è aperto.

di uscita viene ottenuta abbassando e facendo trattenere all'elemento regolatore la differenza con quella di ingresso, negli switching si lavora sui parametri della potenza e lo si fa convertendo l'energia di ingresso. In pratica il funzionamento consiste nel ricavare impulsi rettangolari periodici dalla componente continua di ingresso e nell'inviarli all'uscita giocando sulla loro larghezza nell'arco del periodo, implementando la tecnica nota come PWM (Pulse Width Modulation) che consiste nella modulazione della larghezza degli impulsi. Questa soluzione consente di variare il valore medio della tensione ottenuta portando in uscita tali impulsi, giacché a parità di ampiezza e frequenza degli stessi, variando la larghezza si ottiene una tensione media e quindi una componente "continua" di valore direttamente proporzionale: impulsi più larghi determinano una tensione maggiore e viceversa.

Tale tecnica può essere implementata in vari modi e infatti esistono varie tipologie di convertitori switching tra le quali, le più conosciute sono le seguenti (**Fig. 2**):

- convertitore "Buck" detto anche "Step-down";
- convertitore "Boost" detto anche "Step-up";
- convertitore "Buck-Boost";
- convertitore Sepic;
- convertitore "Cuk" (Chook);
- convertitore "Zeta";
- convertitore "Fly-back".

Questi alimentatori presentano il vantaggio di avere un elevato rendimento, spesso vicino al 90-95%, ma se non sono ben progettati possono presentare rilevanti problemi di "ripple" e di disturbo elettromagnetico (EMI) irradiato nell'ambiente, dovuto al fatto che la base del loro funzionamento è un oscillatore che normalmente lavora tra i 15 kHz fino a oltre 1 MHz e che può generare un numero infinito di armoniche.

In questo articolo, ci occuperemo unicamente di Alimentatori step-down ovvero Buck Converter. Innanzitutto diciamo che come fa desumere il termine "step-down", questa tipologia di Conver-

tore serve per "abbassare/ridurre" una tensione ad un determinato valore prefissato partendo da una fonte di alimentazione più alta: ad esempio per ottenere 5 volt a partire da 12V in ingresso.

COME FUNZIONA LO SWITCHING

Il principio di funzionamento di questo circuito, che poi è fondamentalmente simile a quello di tutti gli altri convertitori switching, si può vedere in **Fig. 3** e si basa su un induttore, un diodo, un condensatore e un interruttore. Quando si chiude l'interruttore, si caricano l'induttore e il condensatore (schema a sinistra) e quando lo stesso si apre, l'energia accumulata scorre nel circuito ed è recuperata dal diodo che la restituisce all'induttore (schema a destra). Nella pratica, al posto dell'interruttore viene utilizzato un transistor BJT o un MOSFET, pilotato da un segnale rettangolare la cui frequenza è quella di oscillazione, necessario a gestire l'intero ciclo, ovvero ad alternare impulsi (che fanno andare il transistor in conduzione) a pause (nelle quali il transistor risulta interdetto).

In **Fig. 4**, viene illustrato il principio di funzionamento con l'andamento delle correnti nei vari rami del circuito in funzione della tensione rettangolare PWM utilizzata per pilotare il transistor che effettua la commutazione.

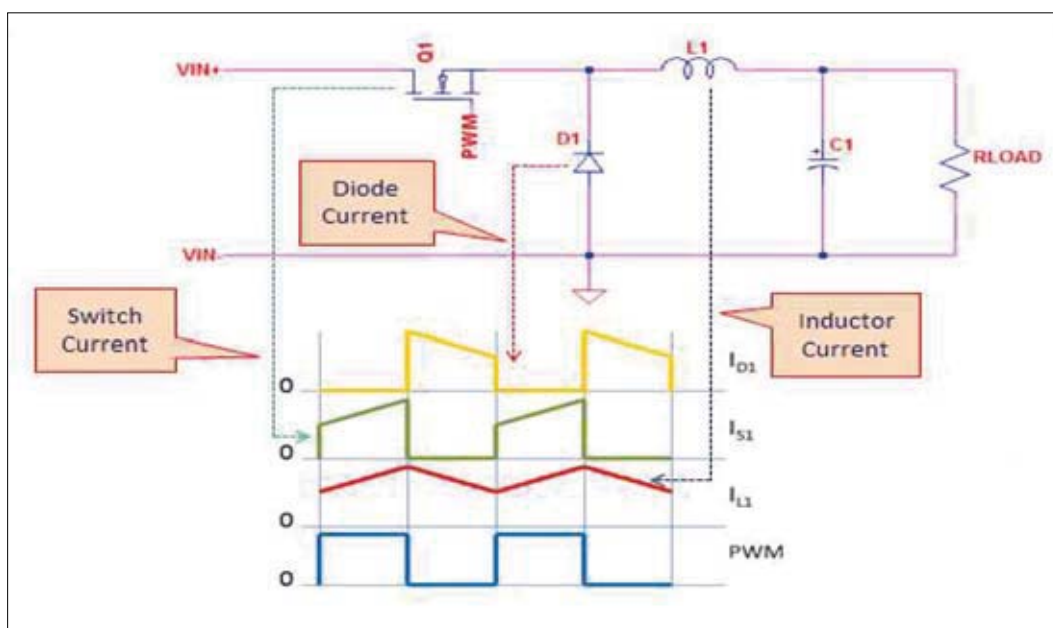
CIRCUITI COMMERCIALI

Ora che ci siamo chiariti un pochino le idee e che certi concetti sono stati esposti quanto basta a comprendere il funzionamento dei convertitori DC/DC a commutazione, passiamo a fare una rassegna dei circuiti che abbiamo voluto provare, fornendo sia le caratteristiche, sia un resoconto delle prove cui li abbiamo sottoposti.

Tra gli svariati convertitori switching presenti sul mercato, abbiamo selezionato sette di essi, di cui due a tensione di uscita fissa a 5V e i restanti a tensione regolabile; di questi ultimi, due permettono anche la regolazione della corrente di uscita, utile per pilotare carichi a corrente costante come ad esempio serie di LED.

Degli alimentatori a tensione d'uscita regolabile che abbiamo testato, due consentono la visua-

Fig. 4
Grafici di
funzionamento del
convertitore Buck.



lizzazione delle tensioni e correnti su display LED 7 segmenti e uno dispone di visualizzazione basata su LCD e regolazioni tramite pulsanti. Infine, uno di essi, in virtù delle sue peculiarità, può essere utilizzato per caricare batterie di ogni tipo. Tutti questi moduli li abbiamo fatto parte dell'assortimento di Futura Elettronica e potrete trovarli on-line sul sito www.futurashop.it.

Fig. 5

DC/DC A USCITA FISSA CON MP1584

Iniziamo l'analisi con il primo modulo da noi testato, che si basa su un chip MP1584 della Monolithic Power, le cui caratteristiche sono:

- tensione di ingresso tra 4,5 e 28 Vcc;
- frequenza di switching programmabile tra 100 kHz e 1,5 MHz;

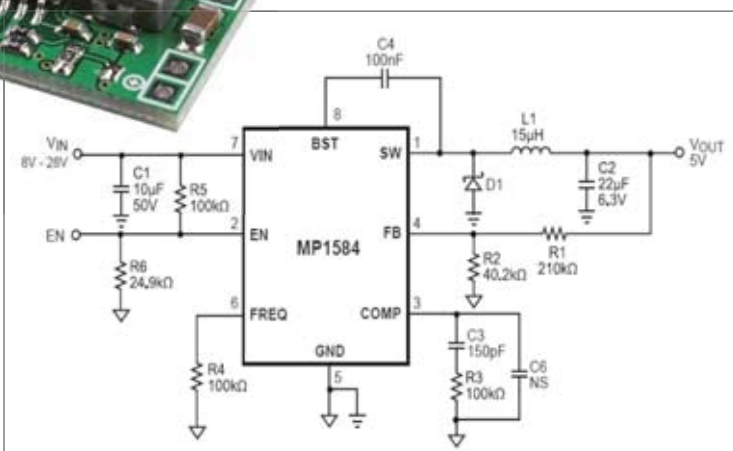
- corrente massima erogabile 1,5A;
- potenza massima dissipabile a $T_A 25^\circ\text{C} = 2,5\text{W}$;
- protezione interna da eccesso di temperatura ($T_J 150^\circ\text{C}$);
- resistenza interna di limitazione della corrente;
- dimensioni: 23x17x4,8 mm.

Il modulo e il relativo schema elettrico, molto simile a quello tratto dal datasheet dell'integrato MP1584, sono visibili nella **Fig. 5**.

Il modulo converter in questione è commercializzato dalla Futura elettronica (www.futurashop.it) con il codice 3085-MP1584ENDCDC.

Il circuito in nostro possesso eroga 5 Vcc ed è stato programmato per oscillare a circa 900 kHz.

Noi lo abbiamo testato con tensioni di ingresso di 12 e 24 Vcc ed abbiamo constatato che regge fino a 2A di Out mentre a 2,5A, va in protezione termica. Abbiamo notato che il maggior rendimento si ha tra 10 e 12Vin e 2A out dove il circuito riesce a stabilizzarsi a 90°C . Bisognerebbe comunque mettere sul piano di massa inferiore un piccolo dissipatore di calore, ovvero prevedere un ampio piano di massa il cui rame possa smaltire il calore. Il condensatore di uscita di soli 10 μF non aiuta molto nella riduzione del ripple, che comunque risulta discreto. Applicando in uscita un semplice filtro LC di cui spiegheremo in seguito, si abbassa di circa 30 dBm il valore di picco delle spurie, a tutto vantaggio della compatibilità elettromagnetica del prodotto. Le caratteristiche in **Tabella 1** e i relativi oscillogrammi in **Fig. 6** permettono di farsi un'idea



Vin [V]	Iin[A]	Vout[V]	Iout [A]	η [%]	Ripple (mV)	F. Peak (KHz)	Peak (dBm)	Ripple (mV) LC	Peak (dBm) LC
12	0	4.95	0	/	17	900	-57	7	-78
12	0,46	4.92	1	87.8	27	920	-39.7	13	-74
24	0,49	4.89	2	79.8	84	919	-32	49	-64

Tabella 1
Parametri operativi con applicato il filtro.

delle caratteristiche del convertitore DC/DC, da noi rilevate nelle prove eseguite in laboratorio.

DC/DC A USCITA FISSA CON TPS4005

Il secondo modulo a tensione d'uscita fissa a 5 volt che abbiamo testato (Fig. 7) si basa su un chip TPS4005 della Texas Instruments, e le sue caratteristiche sono:

- tensione di ingresso tra 9 e 35 Vcc
- tensione di uscita: 5 Vcc
- frequenza di switching circa 270 KHz
- corrente massima erogabile 5A,
- potenza massima in uscita a T_A 25 °C = 25W
- protezione interna dall'eccesso di temperatura T_j che interviene a 150 °C;
- dimensioni (mm): 45x31x16.

Lo schema del circuito è praticamente derivato da quello tratto dal datasheet dell'integrato, che è sostanzialmente quello proposto nella Fig. 8.

Il circuito alimentatore DC/DC in questione eroga 5 Vcc ed è stato programmato per oscillare a una frequenza di circa 270 kHz.

Questo modulo DC/DC (commercializzato on-line su www.futurashop.it con il codice 2846-DCDCSTEPDOWN) basato sull'integrato della Texas Instruments è stato da noi testato con tensioni di ingresso di 9, 12, 24 e 30Vcc, a valori di corrente d'uscita compresi fra 1,2 e 4A e si è comportato egregiamente, senza manifestare surriscaldamenti e/o cedimenti, nè altre anomalie degne di nota. L'alimentatore switching in questione presenta un livello di ripple non eccellente, ma discreto e con il solito Filtro LC in uscita va decisa-

Fig. 6
Oscillogrammi del DC/DC con MP1584:
A) ripple a 1A.
B) Ripple a 2A, con Filtro LC in uscita.
C) Armoniche generate a 1A.
D) Armoniche eliminate con Filtro LC.

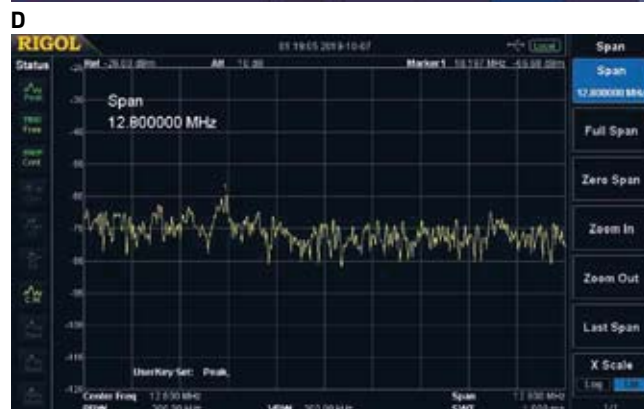
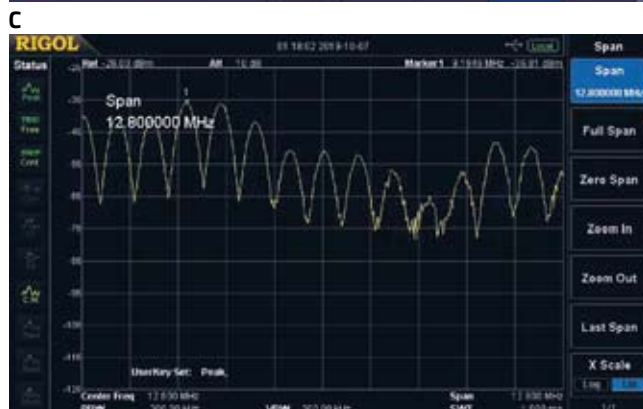
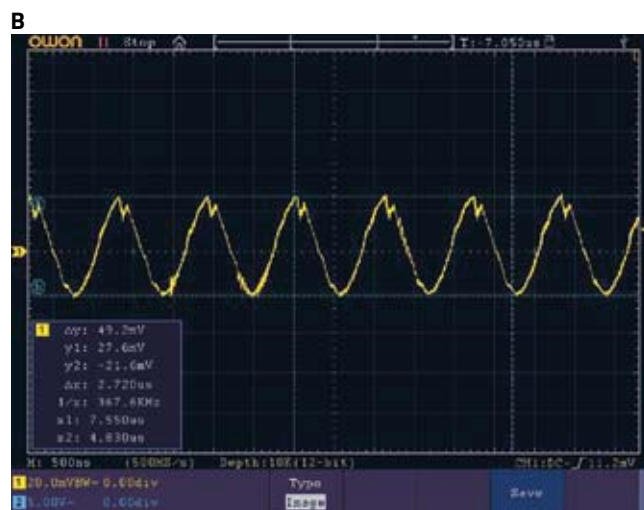
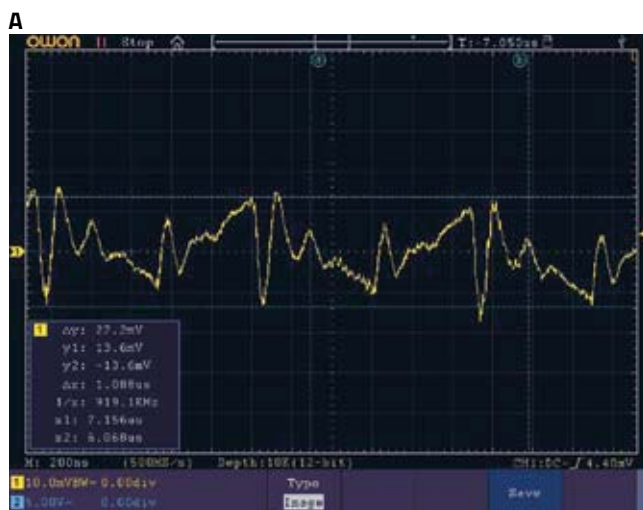


Fig. 7
Il modulo
alimentatore basato
sul TPS4005.



proposti nella **Fig. 9**, potete vedere il comportamento assunto dal circuito DC/DC nelle varie condizioni di funzionamento testate al banco in laboratorio.

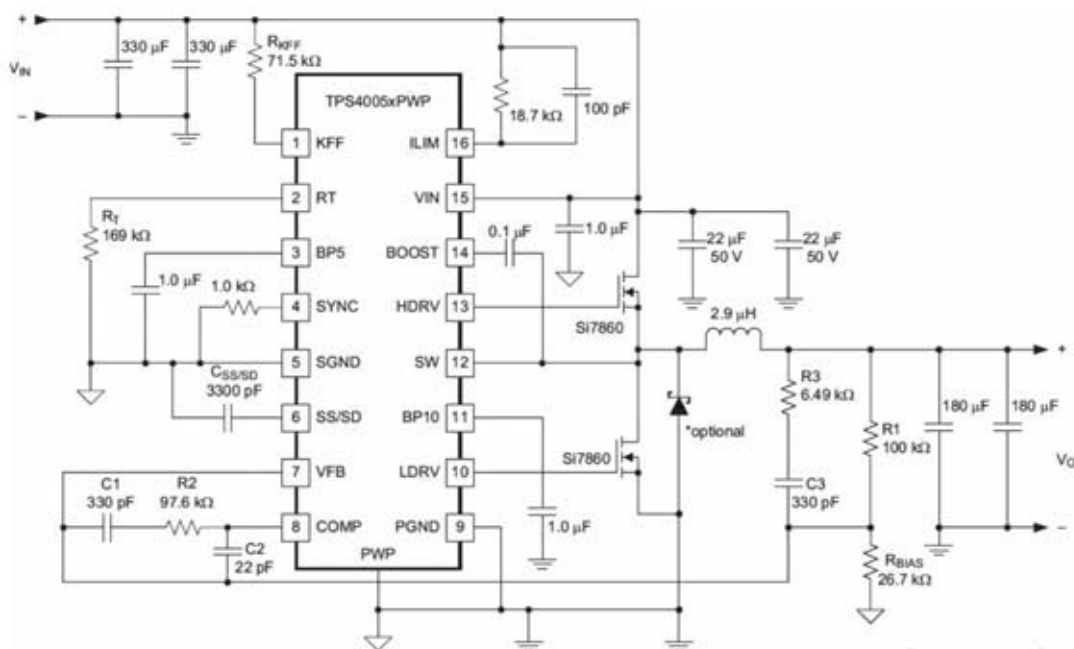
ALIMENTATORE CON LM2596

Il terzo modulo da noi testato (lo vedete nella **Fig. 10**) è a tensione d'uscita stabilizzata ma regolabile tramite un trimmer multigiri: si basa su un chip LM2596, che è un robusto e ampiamente collaudato convertitore buck single-chip. Le caratteristiche principali di questo integrato, configurato come suggerito dal costruttore secondo lo schema di **Fig. 11**, sono le seguenti:

- tensione di ingresso tra 3 e 40 Vcc;
- tensione di uscita regolabile da 1,5 a 35 Vcc;
- frequenza di switching 56,7 kHz;
- corrente massima erogabile 3A;
- potenza massima dissipabile a Ta 25 °C = 2,5W;
- protezione interna dall'eccesso di temperatura

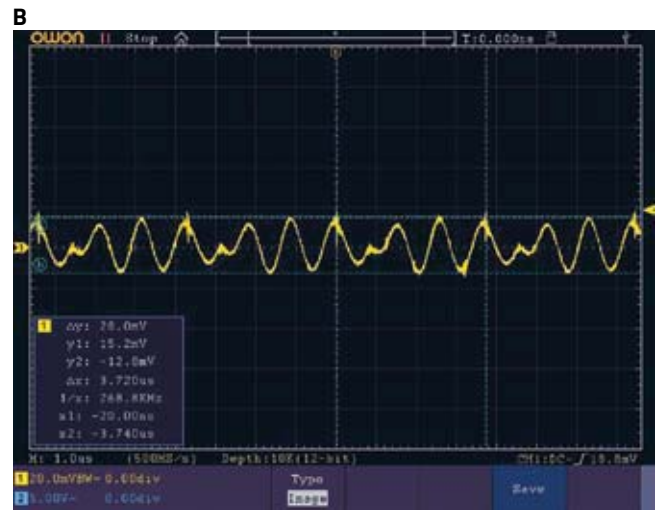
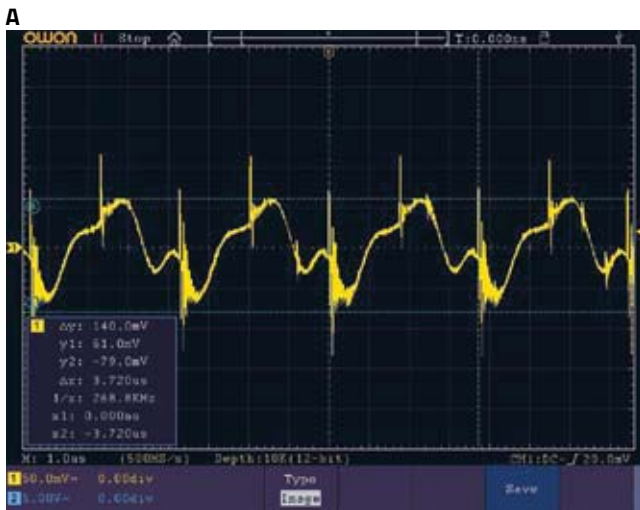
mente meglio. Anche in questo caso sono presenti molte armoniche in uscita, che però in questo caso si vanno via-via attenuando e che spariscono se opportunamente filtrate.
Nella **Tabella 2** e nei relativi oscillogrammi

Fig. 8
Schema applicativo
dell'integrato
TPS4005.



Vin [V]	Iin[A]	Vout[V]	Iout [A]	η [%]	Ripple (mV)	F. Peak (Khz)	Peak (dBm)	Ripple (mV) LC	Peak (dBm) LC
12	0,024	5,02	0	/	92	269	-19	8	-65
9	2,48	4,94	4	87	116	269	-19,6	34	-62,5
12	1,82	4,94	4	89,5	140	269	-18	35	-62
24	0,46	4,97	2	82,5	114	269	-19	28	-63,5
30	0,19	4,99	1	85,8	120	269	-17,8	22	-64,8

Tabella 2



- (Tj 150 °C);
- resistenza interna di limitazione della corrente;
 - dimensioni (mm): 48,5x23,5x14.

Lo schema elettrico del modulo convertitore è sostanzialmente quello proposto dal costruttore nel datasheet del componente, quindi molto simile a quello visibile in **Fig. 11**.

Il modulo è commercializzato dalla Futura Elettronica (on-line su www.futura-shop.it) con il codice prodotto 2846-MODULODCDC.

Abbiamo sottoposto a test questo circuito in diverse condizioni di alimentazione, di tensione di uscita e corrente variabile al fine di valutarne le condizioni ottimali, tralasciando volutamente l'utilizzo alla massima corrente utilizzata (3A), poiché abbiamo notato un certo surriscaldamento.

Il ripple, in questo caso, risulta un po' più elevato del valore ammissibile in buona parte delle applicazioni, quindi per garantire un funzionamento stabile e poco disturbato dei circuiti alimentati con il DC/DC abbiamo ritenuto opportuno utilizzare il filtro LC passa-basso applicato all'uscita, in modo da poter raggiungere valori accettabili (intorno ai 40÷60 mV in base alle condizioni di carico del circuito).

I parametri desunti dalle prove svolte in laboratorio e le relative condizioni di test sono riportati nella **Tabella 3**; gli oscillogrammi più significativi catturati durante i test al banco sono quelli riportati nella **Fig. 12**.

ALIMENTATORE CON XL4015

Il quarto modulo convertitore DC/DC che abbiamo testato lo vedete nella **Fig. 13** e si basa su un circuito integrato siglato XL4015 e prodotto dalla XLSEMI, le cui caratteristiche principali sono:

- tensione di ingresso compresa tra 5 e 32 Vcc;
- tensione di uscita regolabile da 0,8V a 30 Vcc;
- frequenza di switching 180 kHz;
- corrente massima erogabile 5A;
- potenza massima erogabile 75W;
- protezione interna da eccesso di temperatura;
- protezione interna da cortocircuito;

Fig. 9
 A) Ripple con 12 Vcc In e 4A Out.
 B) Ripple con 24 Vcc In e 2A Out + filtro LC.

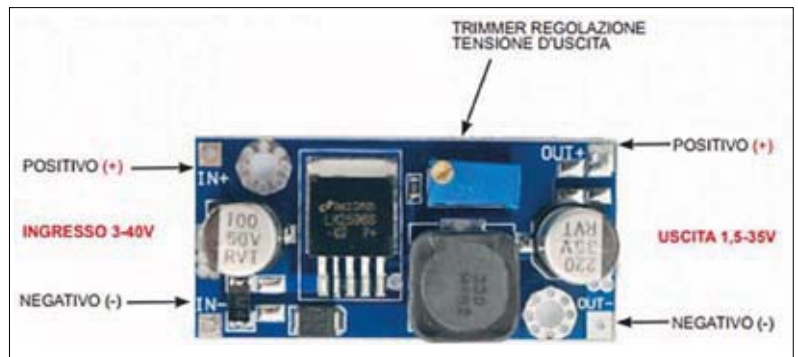


Fig. 10
 Il DC/DC basato sull'integrato LM2596.

Fig. 11
 Schema applicativo dell'LM2596.

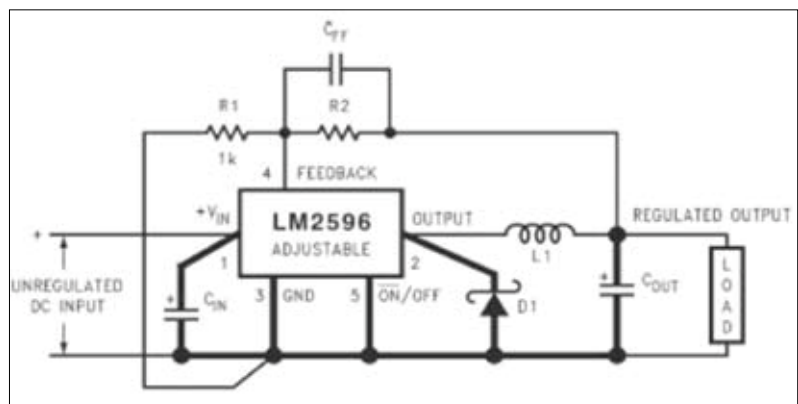


Tabella 3

Vin [V]	Iin[A]	Vout[V]	Iout [A]	η [%]	Ripple (mV)	F. Peak (Khz)	Peak (dBm)	Ripple (mV) LC	Peak (dBm) LC
12	0,52	5	1	75	196	56,67	-17,92	27,6	-50,48
12	1,1	5	2	68	245	56,67	-17,97	39,2	-50,46
24	0,64	12	1,1	83,6	454	56,67	-10,49	45,4	-43,14
28	1	12	2	83,3	364	56,81	-11,66	61,6	-44,17

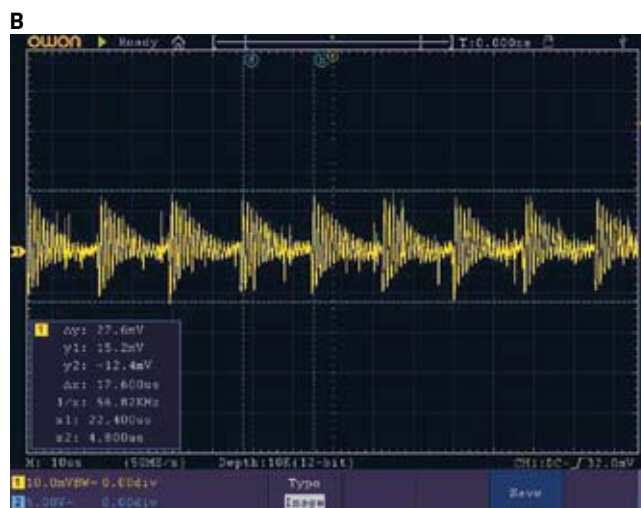
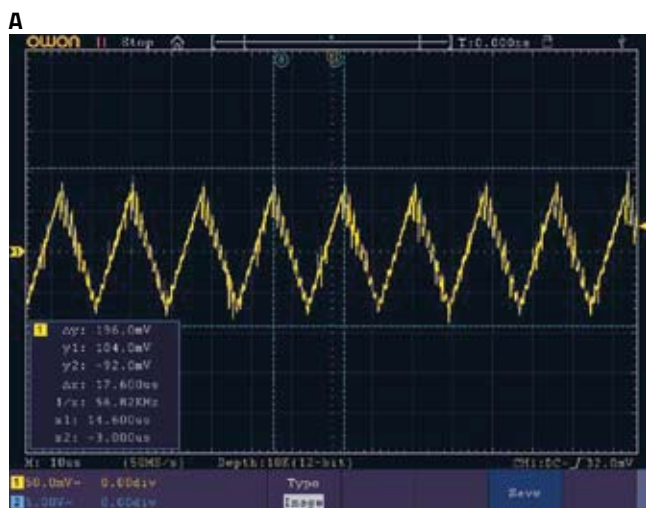


Fig. 12
A) Ripple a 12 Vin e 5 Vout 1A.
B) Ripple a 12 Vin e 5 Vout 1A, + filtro LC.

- limitatore di corrente con riduzione f a 48 kHz;
- dimensioni (mm): 51x26,3x14.

Questo modulo è ancora del tipo a tensione d'uscita stabilizzata ma regolabile tramite trimmer multigiri ed è anch'esso reperibile da Futura Elettronica con il codice 2846-STEPD5V32.

Il suo schema elettrico è derivato da quello applicativo suggerito dal costruttore nel datasheet dell'integrato, che vedete nella **Fig. 14**.

Nel nostro caso, però, è stato aggiunto anche un

trimmer multigiri per la regolazione precisa della corrente di uscita.

Questo risulta utile soprattutto quando si devono pilotare carichi a corrente costante come ad esempio i LED.

Anche questo regolatore DC/DC lo abbiamo testato in laboratorio in diverse condizioni intermedie che abbiamo inserito nella **Tabella 4** insieme alle grandezze caratteristiche rilevate; le forme d'onda corrispondenti desunte dai test sono quelle visibili negli screenshot degli oscillogrammi proposti nella **Fig. 15**.

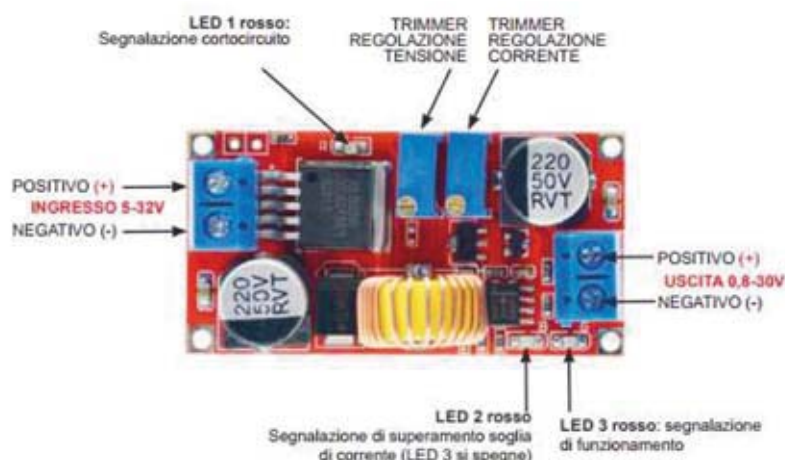
I valori registrati per il ripple sono da ritenere buoni, compresi tra 50 e 125 mV e diventano ottimi con il solito filtro LC passa-basso applicato all'uscita; tale filtro riesce ad abbattere il ripple contenendolo, in base alle condizioni di carico all'uscita, fra appena 30 e 66 mV.

Inoltre, le maggiori dimensioni del circuito stampato garantiscono una maggiore dissipazione del calore, sia per la maggior distanza fra i componenti montati, sia per la superficie più elevata delle piste di rame.

Queste caratteristiche ci consentono tranquillamente di prelevare dall'uscita del convertitore una potenza di 50W.

Nelle prove effettuate al banco, a 24 V di tensione d'ingresso e con impostata all'uscita una Vout di 12, erogando una corrente di 4A, la temperatura

Fig. 13
il DC/DC converter basato sull'integrato XL4015.



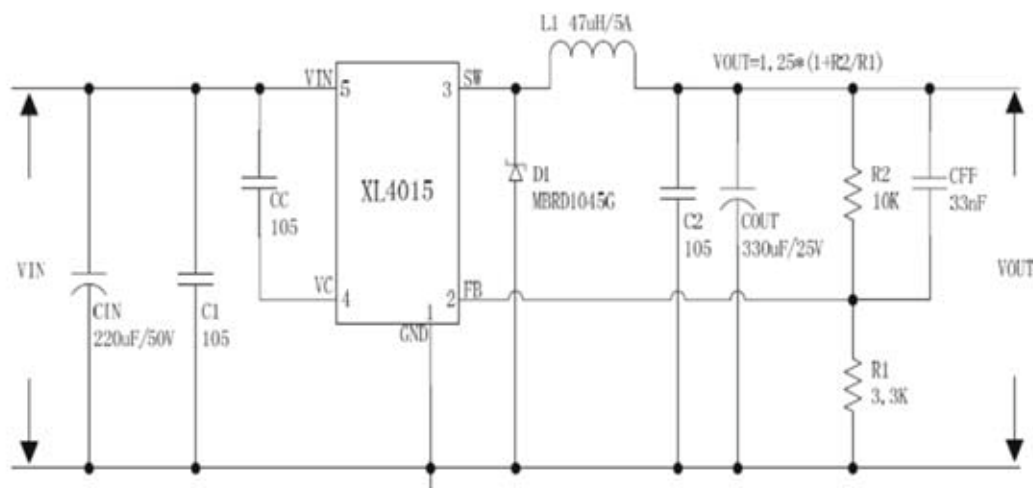


Fig. 14
Schema
applicativo
dell'integrato
XL4015.

Vin [V]	Iin[A]	Vout[V]	Iout [A]	η [%]	Ripple (mV)	F. Peak (KHz)	Peak (dBm)	Ripple (mV) LC	Peak (dBm) LC
9	1,97	5	3	81,8	52	179,5	-31,3	30,4	-72,6
12	2,13	6	3,6	81,7	50,8	180,8	-25,7	29,4	-66,4
24	1,23	12	2,2	88,2	72	185,7	-23,9	28,6	-64,9
28	2,3	24	2,5	92,7	125	198	-27,2	66	-69,9

Tabella 4

raggiunta dalla basetta è di 80 °C, con 25 °C di temperatura ambiente.

ALIMENTATORE CON LM2596 E DISPLAY

Passiamo al quinto dei moduli convertitori DC/DC provati, che si basa nuovamente sul chip LM2596 come il terzo e come quest'ultimo permette la regolazione del valore di tensione all'uscita; in più è dotato di un display e le caratteristiche principali sono:

- tensione di ingresso tra 3 e 40 Vcc;
- tensione di uscita regolabile da 1,5 a 35 Vcc;
- frequenza di switching 48,5 kHz;
- corrente massima erogabile 2A;
- potenza massima dissipabile a una temperatura ambiente di 25 °C, pari a 2,5W;
- protezione interna dall'eccesso di temperatura che interviene quando T_j supera 150 °C;
- resistenza interna di limitazione della corrente;
- dimensioni (mm): 56x35x13.

Questo modulo DC/DC converter è commercializ-

Fig. 15

- A) Ripple a 24Vin – 12VOut – 2.2A.
B) Ripple a 24Vin – 12VOut – 2.2A e filtro LC.

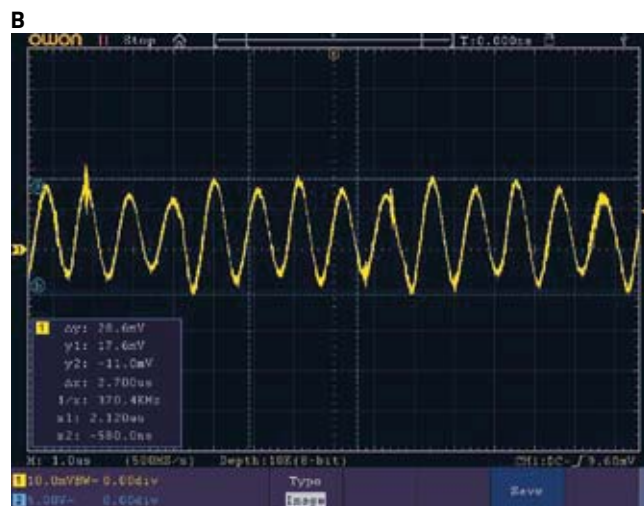
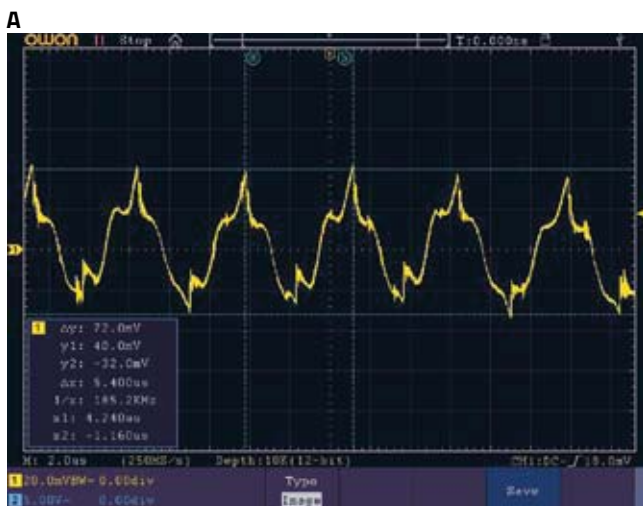




Fig. 16
Il secondo
DC/DC con
LM2596.

zato dalla Futura Elettronica (on-line, sempre sul sito Internet www.futurashop.it) con il codice prodotto STEPDOWNDISP e lo vedete nella **Fig. 16**. Lo schema elettrico del modulo deriva da quello applicativo proposto dal costruttore nel datasheet dell'integrato, che riportiamo nella **Fig. 17**. Rispetto a questo, come variante, troviamo un display LED a sette segmenti e tre digit, che serve per visualizzare sia la tensione d'ingresso che quella di uscita regolata tramite il trimmer a bordo. I due pulsanti servono per visualizzare alternativamente una o l'altra tensione e per accendere o spegnere il display quando non serve, allo scopo di limitare le perdite ed elevare il rendimento del convertitore DC/DC, giacché il display consuma una parte dell'energia prelevata dall'ingresso. L'andamento tensione/corrente/ripple, rispecchia fondamentalmente quello già visto per il **terzo modulo** testato, ossia quello con LM2596 a tensione d'uscita regolabile, anche se in questo caso le prove da noi effettuate al banco ci dicono che i valori di ripple sono leggermente più alti; infatti si può arrivare fino a 700 mV nel caso di funzionamento con all'ingresso una tensione di 30Vcc e 24V all'uscita,

con 1A di corrente erogata. L'ampiezza del ripple, come sempre, può essere ridotta con l'aiuto del filtro applicato in serie all'uscita.

Nella **Tabella 5** si possono rilevare le caratteristiche del convertitore DC/DC nelle varie condizioni operative. Per le forme d'onda, poiché sono molto simili a quelle proposte negli oscillogrammi del terzo modulo, non stiamo a riproporle; al loro posto vi proponiamo (nella **Fig. 18**) quelle rilevate delle condizioni peggiori, ovvero con una tensione d'ingresso di 30Vcc e una d'uscita a 24Vcc con erogazione di corrente di 1A; in tali condizioni di test si evidenzia che risulta più marcata la frequenza dell'armonica di 1/3 inferiore e cioè a 16 kHz circa. Abbiamo anche effettuato una prova con tensione d'uscita di 5V e assenza di carico, dalla quale è emerso un nutrito spettro di armoniche generate.

ALIMENTATORE DC/DC CON MP2307

Il sesto modulo che abbiamo testato si basa su un chip MP2307 della Monolithic Power, le cui caratteristiche sono:

- Tensione di ingresso tra 5 e 23 Vcc
- Tensione di uscita tra 1 e 16,5 Vcc
- Frequenza di switching fissa a 340 kHz
- Corrente massima erogabile 2A (3A peak)
- Display LCD 37,5x17 per visualizzazione
- Regolazione tensione di uscita tramite pulsanti
- Visualizzazione tensione di uscita e corrente
- Dimensioni (mm): 50x30x12.

Il modulo viene fornito con un elegante e funzionale contenitore in plexiglass da montare. Le regolazioni della tensione di uscita vengono effettuate mediante due comodi pulsanti e le prestazioni sono complessivamente buone.

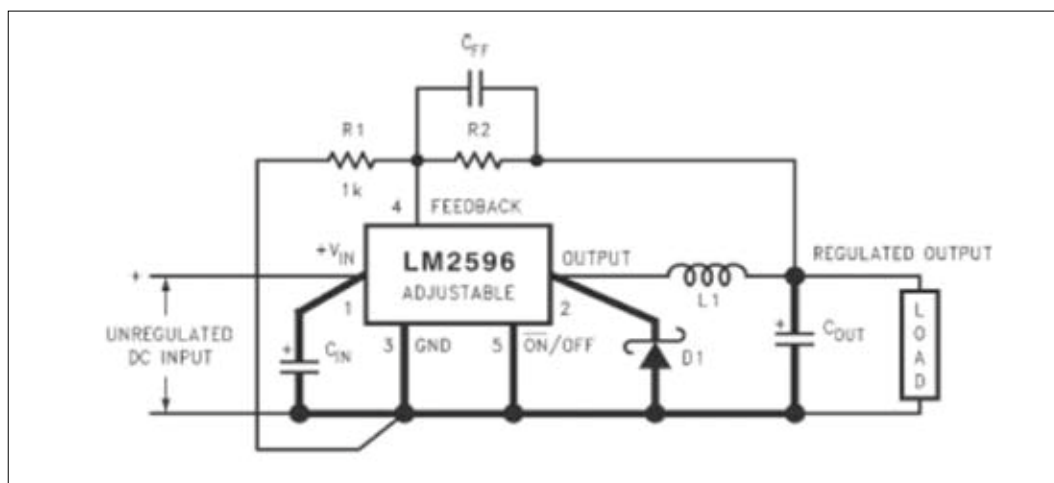


Fig. 17
Schema
applicativo
dell'LM2596.

Vin [V]	Iin[A]	Vout[V]	Iout [A]	η [%]	Ripple (mV)	F. Peak (Khz)	Peak (dBm)	Ripple (mV) LC	Peak (dBm) LC
12	0	5	0	/	35	48,67	-42,9	8	-76,6
8	0,86	5	1	62,4	120	17	-23,7	31	-39,2
12	0,92	5	1,6	62	312	48,5	-13,9	17	-44,9
24	0,88	12	1,5	82,7	486	48,2	-8,6	18	-39,6
30	0,86	12	1,8	80,6	560	19	-9,9	87	-36,3
30	0,9	24	1	87,5	700	49	-12	72	-43

Tabella 5

Viene commercializzato anche questo dalla Futura Elettronica con il codice 2846-DCDISPAVDOWN e lo vedete in fotografia nella **Fig. 19**.

Lo schema elettrico di questo circuito ricalca quello applicativo dell'integrato MP2307 suggerito dal costruttore nel datasheet, quindi è sostanzialmente quello visibile nella **Fig. 20** con aggiunto il modulo voltmetro basato sul display a cristalli liquidi. Testato in varie condizioni, come da **Tabella 6** si evidenziano discreti livelli di ripple (max. 180 mV) senza filtro di uscita. Alle correnti più elevate sono presenti alcuni picchi di HF che non siamo riusciti ad eliminare anche schermando le sonde e delle quali comunque abbiamo ritenuto di non valutare.

Anche in questo caso si evidenziano molte armoniche alle quali ormai abbiamo fatto abitudine ed imparato a convivere quando si utilizzano alimentatori switching. Gli oscillogrammi mostrati nella **Fig. 21** descrivono il comportamento del DC/DC converter, riepilogato anche dai parametri riportati nella **Tabella 6**, relativi ai test condotti in laboratorio.

ALIMENTATORE CON XL4015 E DISPLAY

Il settimo modulo testato, si basa sullo stesso chip del "quarto modulo" ovvero XL 4015 della XLSEMI codice Futurashop.it 8300-YB728 le cui caratteri-

stiche peculiari in questo caso sono:

- tensione di ingresso: da 7 a 36 Vcc;
- tensione di uscita: da 1,25 a 32 Vcc;
- corrente massima erogabile: 5A;
- potenza massima 75W;
- frequenza di lavoro da 186 a 189 kHz;
- efficienza di conversione da 62 a 90%;
- protezione termica interna;
- limitazione di corrente in uscita;
- protezione da inversione di polarità: NO;
- protezione dal cortocircuito in uscita;
- display 7 segmenti 4 cifre LED rossi;
- temperatura operativa, da -45 °C a +85 °C;
- dimensioni scheda (mm): 68x39x15;
- dimensioni con il case (mm): 78x49x25;
- peso: 40 grammi.

Anche questo modulo viene dotato di un bel contenitore trasparente realizzato in plexiglass e fornito in KIT di montaggio. Il modulo dispone di due trimmer multigiri che servono per impostare la tensione di uscita e la corrente erogabile al carico. Inoltre sono presenti due tasti che servono sia per il settaggio/taratura dello strumento, sia per la visualizzazione delle grandezze in gioco, vale a dire:

- tensione di ingresso in volt;

Fig. 18
A) Ripple a 30Vin – 24VOut – 1A.
B) Ripple a 30Vin – 24VOut – 1A e filtro LC.

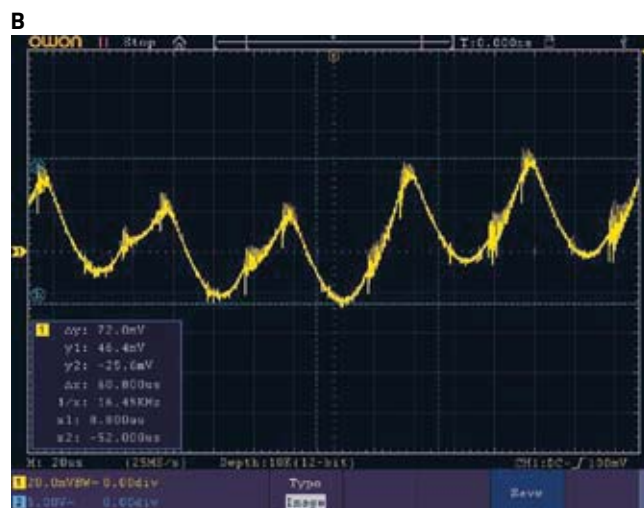
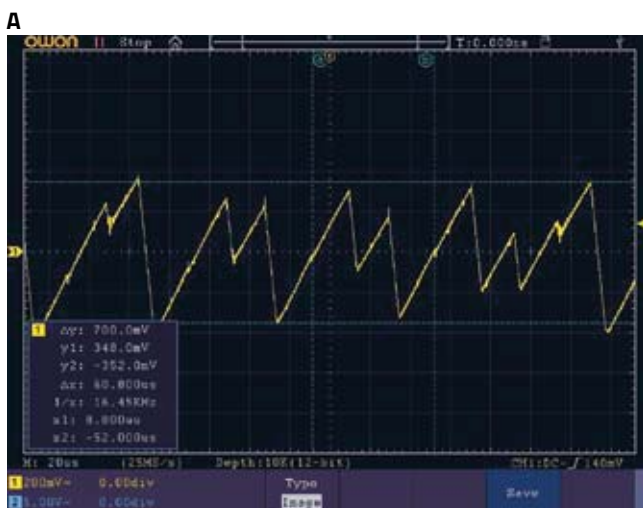
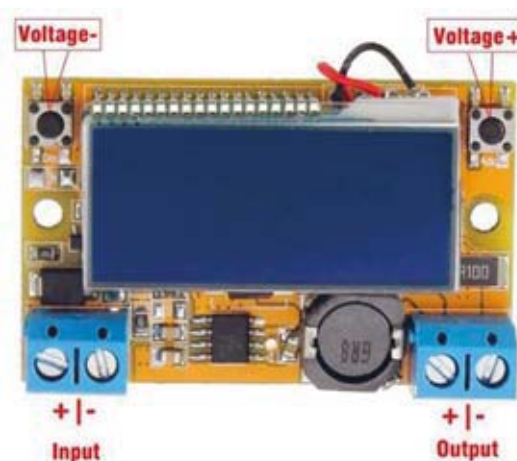


Fig. 19
Convertitore
DC/DC con
display.



- tensione di uscita in volt;
- corrente erogata espressa in ampere;
- potenza sul carico espressa in W.

Inoltre la presenza di cinque LED permette l'identificazione di varie funzioni. Poiché questo circuito si presta molto bene ad essere utilizzato come carica batterie di tutti i tipi a corrente costante (litio, NiMH, piombo), ci soffermeremo un po' di più nelle varie analisi e test.

La **Tabella 7** ne riassume le caratteristiche salienti relative all'utilizzo come caricabatterie alle tensioni standard di 4,2 V (piena carica delle batterie al litio da 3,7V) e 14V (batterie al piombo da 6 elementi, ossia 12V).

Inoltre abbiamo eseguito un test simulando l'applicazione del mantenimento della carica per una batteria al piombo (tipo quelle da auto e moto) a tensione di 13,8 con corrente di mantenimento stabilita in 190 mA.

Gli oscillogrammi relativi a questa applicazione di prova sono riportati nella **Fig. 23**.

A proposito del test di mantenimento, abbiamo potuto notare che a 24V di ingresso e impostando in uscita una differenza di potenziale di 13,8V, la minima corrente settabile in cortocircuito è di circa 800 mA, ma il circuito risulta alquanto instabile e siamo quindi saliti fino al minimo valore di cortocircuito al quale sembrava stabilizzarsi e siamo giunti a 930 mA.

Con questi valori di settaggio e cioè 13,8V a vuoto in uscita e una corrente di cortocircuito di 930 mA, abbiamo poi simulato un carico di mantenimento a 12,6V di tensione d'uscita, dove la corrente assorbita risulta di 190 mA.

Fig. 20
Schema
applicativo
dell'integrato
MP2307.

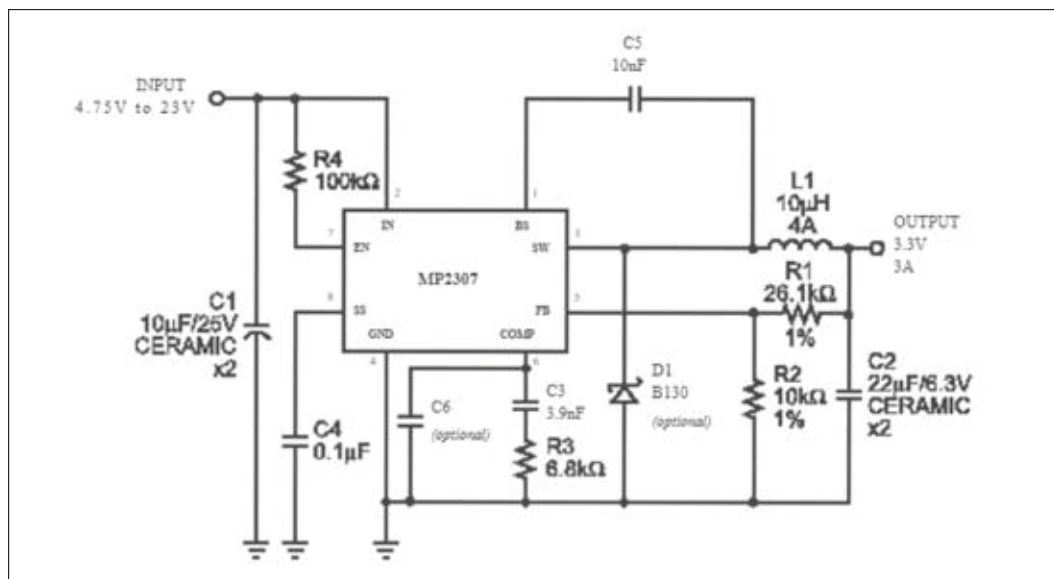
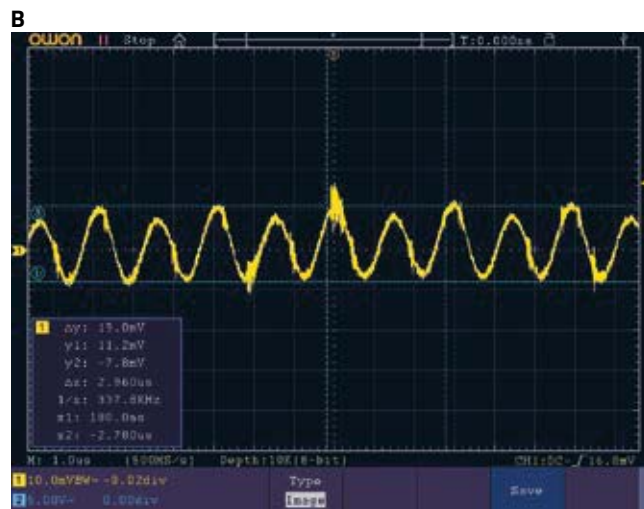
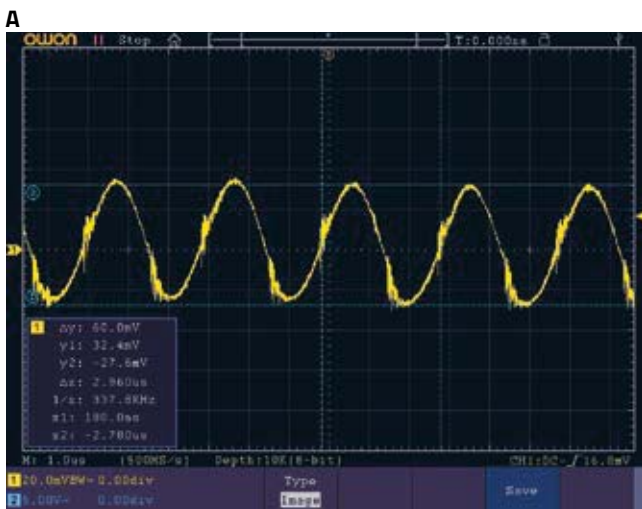


Tabella 6
Risultati della
prova del DC/
DC con display.

Vin [V]	Iin[A]	Vout[V]	Iout [A]	η [%]	Ripple (mV)	F. Peak (Khz)	Peak (dBm)	Ripple (mV) LC	Peak (dBm) LC
12	0	5	0	/	60	338,7	-21,9	19	-76
5	0,83	3,6	1	84,7	25	325,3	-38,7	10	-78
12	0,7	5	1,5	88	170	336	-22	26	-60,6
20	0,97	12	1,5	92,2	182	330,7	-12,2	65	-55,7



In queste condizioni abbiamo rilevato che il sistema funziona, anche se con un comportamento alquanto singolare. Si rileva la frequenza di oscillazione a circa 189 kHz, ma c'è un treno d'onda con f di circa 6 kHz e con 12 picchi da 5,3 μ s (189 kHz) per un periodo totale di:

$$t = 5,3 \times 12 = 63,6 \mu s$$

corrispondenti a una frequenza di 15,7 kHz, con ampiezza di circa 420 mV.

La forma d'onda presenta l'involuppo tipico di una modulazione di ampiezza AM con portante 180 kHz e modulante a 15,7 kHz. Potrebbe derivare, ad esempio, da un'interferenza causata dall'oscillatore del multiplexer utilizzato per il pilotaggio del display. Con filtro di uscita si smorza tutto, fino a limitare il picco a un massimo di 29 mV.

Alla fine spendiamo due parole circa il filtro che abbia-

mo realizzato. Trattasi di un semplice filtro LC passa basso di II° ordine come da schema canonico (induttore in serie e condensatore in parallelo all'uscita). Il calcolo della frequenza di taglio si effettua con la seguente formula:

$$\frac{1}{2\pi\sqrt{LC}}$$

da cui:

$$Fc = \frac{1}{2\pi\sqrt{27 \times 100}} = 0.0030629 \text{ MHz (3.06 KHz)}$$

Notate che il valore L (27) è espresso in μ H e il C (100) in μ F.

Il valore tipico dell'impedenza di uscita risulta:

$$Z = \sqrt{\frac{L}{C}} = \sqrt{\frac{27}{100}} = 0.52 \Omega$$

Fig. 21

- A) Ripple e frequenza a 12Vin – 5Vout – 0A.
B) Ripple a 12Vin – 5Vout – 0A con filtro LC.



Cosa occorre?

Il materiale presentato in questo articolo è disponibile presso Futura Elettronica. Il regolatore con MP1584 (cod. MP1584ENDCDC) è disponibile a Euro 2,50, il regolatore con TPS4005 (cod. DCDCSTEPDOWN) è in vendita a Euro 6,50, il regolatore con LM2596 (cod. MODULODCDC) costa Euro 4,30, il regolatore con XL4015 (cod. STEP5V32) è disponibile a Euro 5,90, il regolatore con LM2596 e display (cod. STEPDOWNDISP) è in vendita a Euro 6,90, il regolatore con MP2307 (cod. DCDISPAVDOWN) costa Euro 11,00, mentre il regolatore con XL4015 (cod. YB728) è disponibile 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 - <http://www.futurashop.it>

Fig. 22

Il modulo DC/DC con XL4015 e display a LED.



Più sicurezza con le nuove telecamere

Sicurezza subito, installazione facile e veloce!



cod. FR769

€ 73,00

Telecamera WiFi motorizzata

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

Prezzi IVA inclusa.

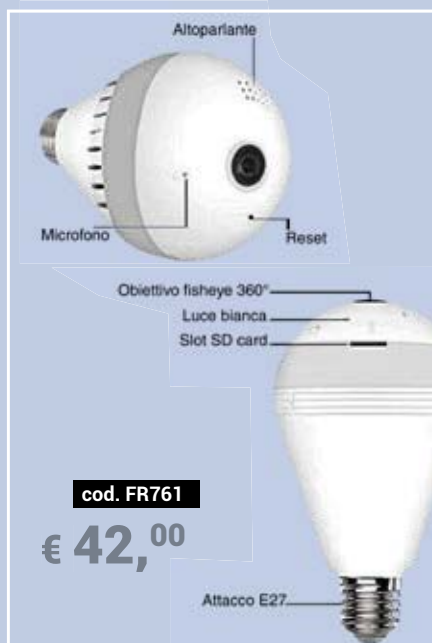


cod. FR724N

€ 98,00

Telecamera WiFi da 1.3 Mpx ad ottica fissa

- › Elemento sensibile 1/3 CMOS
- › ISP Hisilicon Hi3518E
- › Risoluzione 1.3 Mpx - 1280x1080p
- › Frame rate 1280x1080@25fps
- › Apertura angolare 110°
- › 6 LED IR da 850nm
- › Sensore PIR motion detection
- › Portata LED IR 6 metri
- › Alimentazione con 2 batterie 18650 (incluse)
- › Temperatura di lavoro da -20°C a 55°C
- › Dimensioni 95x53x80 mm
- › Peso 550g



cod. FR761

€ 42,00

Telecamera WiFi a 360° nascosta in una lampadina

- › Potenza lampadina 6W
- › Sensore di immagine da 2Mpx CMOS
- › Visuale di ripresa 360°
- › Definizione immagine 1080PH
- › Microfono bidirezionale per ascoltare e parlare direttamente attraverso la telecamera
- › Modulo Wi-Fi integrato
- › Night & Day con tecnologia IR-Cut
- › Registrazione su SD-Card (max 128GB)
- › Compressione video H.264
- › Funzione motion detection
- › Push di allarme e invio email
- › Peso: 200 g

Telecamera WiFi da 2Mpx con pannello solare e batteria

cod. FR767

€ 125,00



- › Sensore d'immagine FHD 1080P CMOS 1/2.7 PS5230
- › Risoluzione Video 1920x1080/15 fps 640x360/30 fps
- › Modalità Day/Night automatica
- › Illuminatore IR con LED IR da 850 nm e portata di circa 10 metri
- › Obiettivo 2G2P HD, F=2,8, con angolo di ripresa di 120°
- › Alimentazione con 2 batterie al litio tipo 18650 ricaricate tramite pannello solare integrato
- › Formato di compressione video H264
- › Protocolli: Network TCP/IP, HTTP, TCP, UDP, SMTP; Wi-Fi 802.11b/g/n
- › Registrazione su SD-Card (da 4GB a 64GB)
- › Grado di protezione IP67
- › S.O. supportati: iOS10.0, Android 5.0 o superiore
- › Temperatura di funzionamento da -10°C a +50°C

cod. FR766

€ 42,00

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

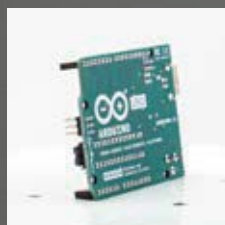
- › Elemento sensibile CMOS colori 1/3", 2 Mpx
- › Risoluzione massima: 1920x1080
- › Obiettivo: 360° fisheye
- › Illuminatore IR: 3 LED
- › Portata: 10 m
- › Night&Day automatico
- › Motion detection con push di allarme
- › Porta Ethernet RJ45: 10/100Mbps
- › Protocolli: ONVIF 2.0, TCP/IP, HTTP, TCP, UDP, SMTP, FTP, DHCP, DNS, DDNS, NTP, UPnP, PPPoE, P2P
- › Sicurezza Wi-Fi: WPS encryption
- › Visione remota tramite Smartphone (Android)
- › Uscita video: IP
- › APP: IP PRO per Android scaricabile gratuitamente
- › Alimentazione 12 Vdc < 1A (con adattatore di rete non incluso)
- › Temperatura di lavoro: da -10°C a +50°C
- › Dimensione 15,2x16,5x 4 cm
- › Peso 250 g



Realizziamo una piattaforma
rotante controllata da
Raspberry Pi per fotografare
a tutto tondo qualsiasi
oggetto per poi condividerlo
in formato foto,
GIF e video.

di **LUCA BELLAN**

CAPTURE360



S

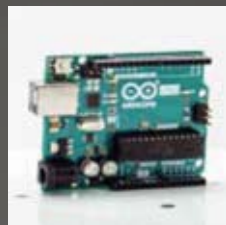
cattare fotografie a 360° e visualizzarle in un file GIF animato è una tecnica suggestiva e già utilizzata in pratica, per mantenere o comunicare l'aspetto complessivo di un oggetto; sostanzialmente somiglia molto a quello che fa uno scanner 3D a sola telecamera, con la sola differenza che qui viene creato un video o, se si

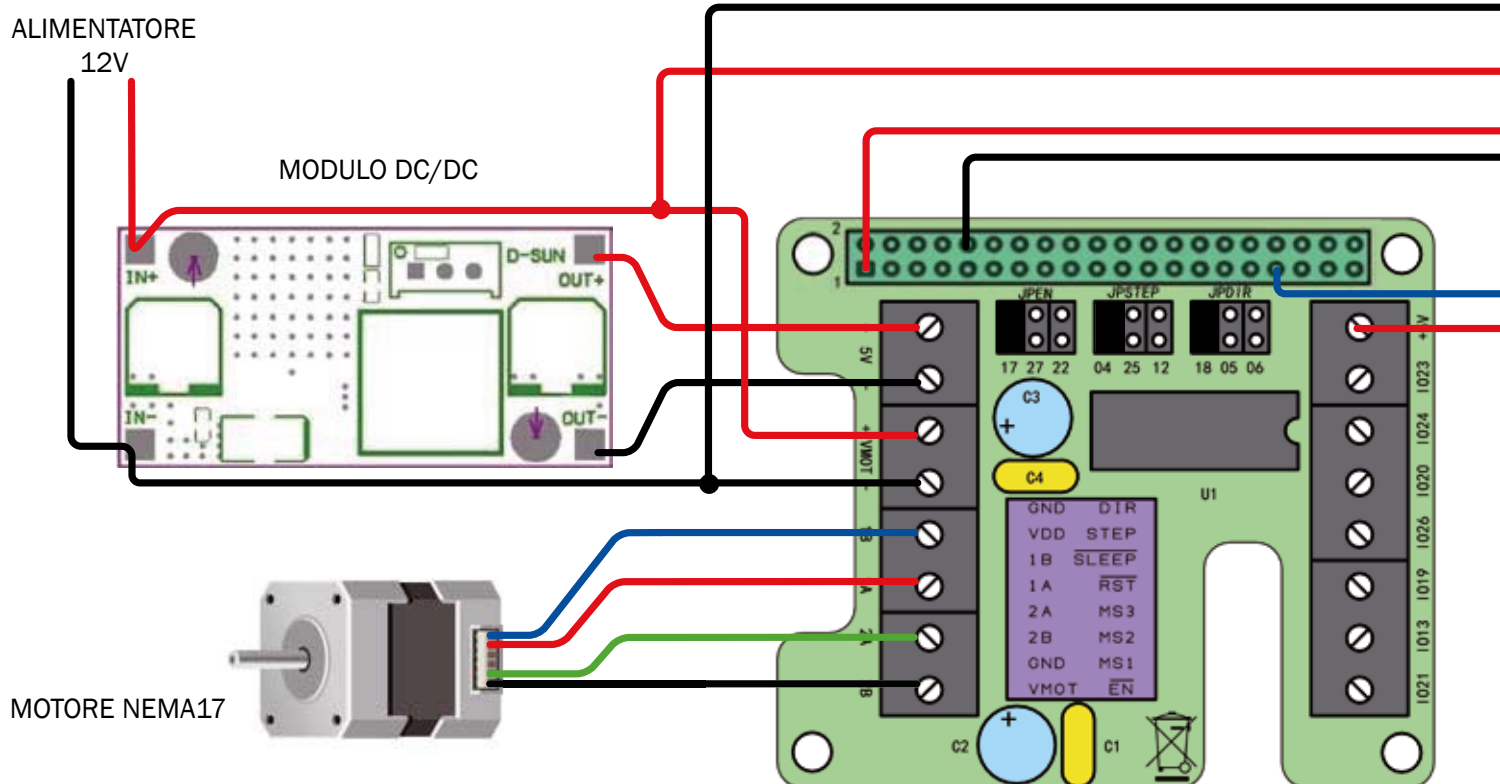
preferisce, una GIF animata dell'intera ripresa. Come nello scanner 3D, l'oggetto viene fatto ruotare sul piano orizzontale mediante un piatto motorizzato e l'avvio della rotazione fa da trigger all'inizio della ripresa o allo scatto di una sequenza di fotogrammi da parte della telecamera.

In questo articolo vi presentiamo una piattaforma estremamente semplice da utilizzare e veloce da costruire che, tramite una web app, consente di scattare delle foto o registrare dei video a un oggetto posto su una piattaforma messa in rotazione da un motore passo-passo.

Nel dettaglio, sono state previste tre modalità di ripresa, per ciascuna delle quali è necessario impostare specifici parametri di configurazione:

- Photo: viene eseguita una sequenza di fotogrammi dei quali si può impostare la risoluzione e la quantità delle fotografie scattate per ciascun giro di 360 gradi;





- GIF: creazione di una Animated GIF con la possibilità di impostare la risoluzione delle foto che comporranno la GIF stessa, la quantità di scatti per giro completo, il frame-rate (FPS) con cui sarà generata la GIF e il numero di giri che la piattaforma dovrà compiere;
- Video: corrisponde a creare un filmato, per il quale è possibile impostare la risoluzione video, il frame-rate corrispondente e il numero di giri che la piattaforma girevole dovrà compiere.

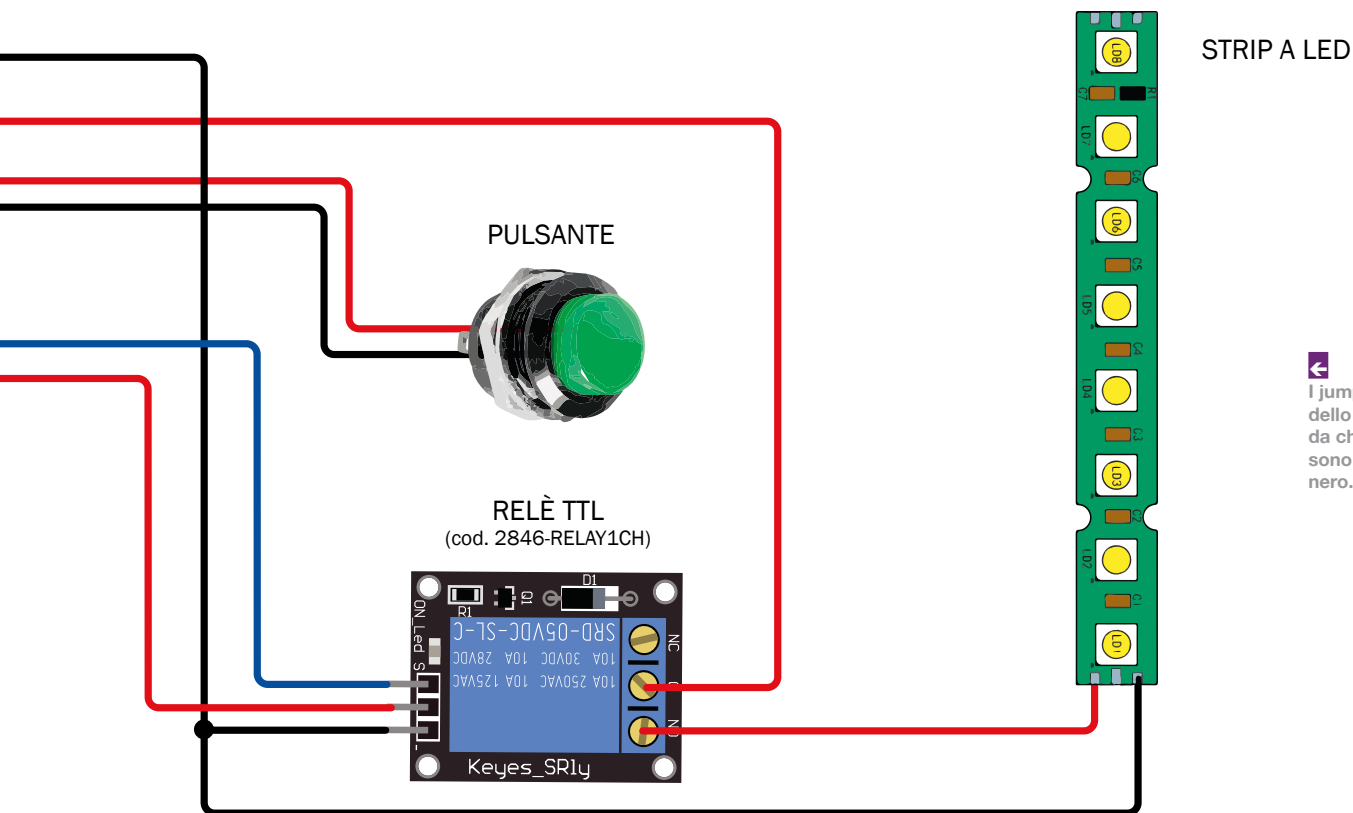
Inoltre è prevista un'impostazione di trigger che consente di salvare una delle tre modalità (con i relativi parametri impostati) in modo da poterla richiamare a piacimento tramite la pressione del pulsante trigger sull'interfaccia grafica o tramite un pulsante di trigger fisico che sarà collegato a un preciso pin dell'header di espansione GPIO della Raspberry Pi. Sarà altresì possibile distanziare la Raspberry Pi e la Pi Camera dalla piattaforma rotante per poter inquadrare gli oggetti più voluminosi; il pulsante di trigger fisico potrebbe risultare comodo proprio in questo caso, per non doversi recare ogni volta alla postazione PC per avviare la cattura. Nel caso in cui vengano scaricate le foto, nell'archivio troveremo anche un utile widget HTML che

potrà essere inserito in una pagina web e, tenendo cliccato il puntatore sulla foto mentre lo si trascina verso destra o verso sinistra, l'oggetto fotografato ruoterà su se stesso come un oggetto 3D; naturalmente in una cartella dell'archivio scaricato troveremo tutte le foto utilizzate dal widget e che potremo anche usare come meglio vogliamo.

L'interfaccia della web app sarà fruibile su un browser di qualsiasi dispositivo fisso o mobile della nostra rete (dopo aver impostato la connessione WiFi sulla board Raspberry Pi), oppure direttamente sull'ambiente desktop di Raspberry Pi collegando un monitor, un mouse e una tastiera (o, in alternativa, un display touchscreen come il 2850-10INCHHDMI reperibile sul sito e-commerce www.futurashop.it).

L'HARDWARE DEL SISTEMA

Detto ciò, passiamo a spiegare come è fatto il nostro sistema, che consta di un telaio sul quale è montata l'elettronica, formata dalla scheda Raspberry Pi 3 Tipo A+ con Wi-Fi e Bluetooth e dalla sua telecamera, che è del tipo con ottica regolabile, oltre che da uno shield per la gestione del motore passo-passo (NEMA 17) che fa ruotare il piatto su cui l'oggetto verrà appoggiato.



←
I jumper
dello shield
da chiudere
sono quelli in
nero.

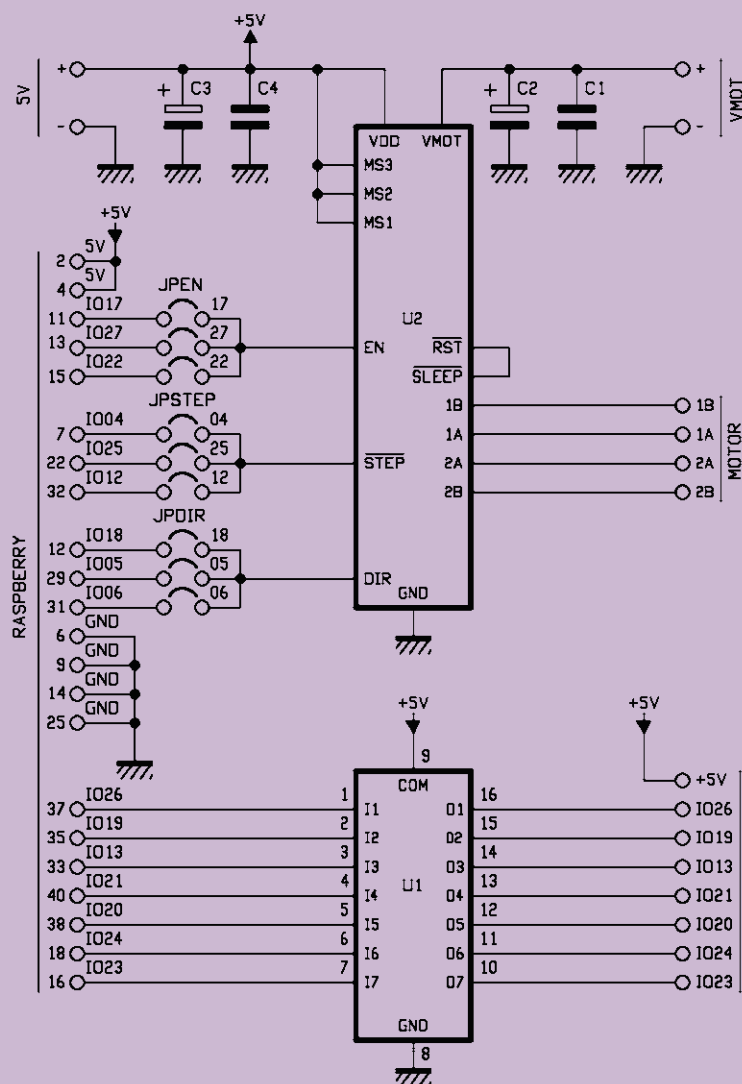
Abbiamo preferito una telecamera con obiettivo regolabile per ottimizzare la messa a fuoco dell'oggetto, in quanto ci siamo accorti che quella originale per la Raspberry Pi ha una distanza di messa a fuoco maggiore di quella esistente tra la base e il piatto girevole che ospita l'oggetto da riprendere. Potendo regolare con precisione la messa a fuoco, siamo in grado di ottenere la miglior definizione in fase di taratura, prima di avviare la scansione. La telecamera ha una risoluzione di 5MP per Raspberry Pi, con una lunghezza focale da 6 mm regolabile (è a catalogo Futura Elettronica - www.futurashop.it con il codice RPICAMERA-FOC).

Il sistema si connette in rete: la gestione di tutte le funzioni, così come le impostazioni dei parametri, avviene mediante rete Wi-Fi, quindi è effettuabile anche da remoto, tramite un qualsiasi computer.

SCHEMA ELETTRICO

Diamo adesso uno sguardo all'elettronica di governo del sistema, che si basa sulla Raspberry Pi e sullo shield che abbiamo appositamente progettato e che va innestato sul connettore header di espansione della Raspberry Pi stessa. Lo shield, come mostra lo schema elettrico corrispondente pubblicato in queste pagine, contiene un driver per

stepper-motor basato sull'integrato DRV8825 della Texas Instruments (cod. DRV8825), oltre a un line-driver ULN2003 (U1), utilizzato per gestire eventuali dispositivi esterni comandati delle linee di GPIO della Raspberry Pi, per lo scopo configurate come uscite. Il driver dello stepper motor (U2) permette di impostare la direzione di rotazione dell'albero del motore e il numero di gradi che il rotore deve compiere alla ricezione di ogni comando; in altre parole, possiamo decidere se, quando gli forniamo un impulso di comando, il modulo deve far ruotare l'albero di uno step alla volta, oppure di 1/2, 1/4, 1/8, 1/16 o 1/32, in base all'accuratezza desiderata. In realtà nel nostro progetto abbiamo impostato il funzionamento a 1/32 di step ponendo a 1 logico i pin MS1, MS2 e MS3, così da ottenere la massima precisione di movimento. La Raspberry Pi comanda il driver U2 impostando lo stato logico delle linee /STEP, EN e DIR; alla prima (gestita con uno a scelta tra IO04, IO12 o IO25) invia gli impulsi che determinano l'avanzamento di un passo alla volta nella rotazione dell'albero del motore, mentre DIR (IO05, IO06 o IO18) viene definito di volta in volta a seconda che si voglia ottenere la rotazione in senso orario o antiorario. EN (IO17, IO22 o IO27) serve per abilitare o disabilitare il modulo. Per comandare



la rotazione di un passo dell'albero del motore, il microcontrollore pone EN a livello logico alto e lo mantiene in questa condizione, poi imposta DIR ad uno logico se la rotazione deve avvenire in senso antiorario o zero per quello orario (mantenendo anche questa condizione logica) e infine fornisce alla linea /STEP un impulso a livello basso; fatto ciò, riporta a riposo sia EN che DIR. Se l'avanzamento dev'essere di più step consecutivi, mentre EN e DIR sono attivi il micro invia a /STEP la sequenza di impulsi richiesta.

Ricordate che il motore fa ruotare il proprio albero, ad ogni impulso che la Raspberry Pi manda sul pin /STEP, in base all'impostazione di MS1, MS2, MS3; nel nostro caso, per compiere un passo ne occorrono

32. Notate che per ogni linea di comando del driver abbiamo previsto la possibilità di scegliere, tramite ponticelli, fra tre linee di GPIO; questo per darvi la possibilità di montare più shield sulla Raspberry Pi e quindi di gestire altre applicazioni, oltre allo scanner, ma anche di utilizzare lo shield come controller generico per altri impieghi e comandare magari due o tre stepper-motor (in tal caso è gioco forza utilizzare GPIO differenti per le varie schede).

Una volta montato lo shield posizionate i jumper come preferite e annotatelo, perché dovrete poi impostare le linee GPIO nel software, secondo quanto spiegato più avanti. L'impostazione dei GPIO corrisponde al posizionamento dei jumper dello shield come mostrato nel disegno di cablaggio visibile in queste pagine.

Ciascun driver consta di un doppio ponte ad H governato da un'elettronica che permette di impostare il verso di rotazione del campo elettromagnetico e quindi dell'albero dello stepper-motor. Ogni volta che sul piedino STEP giunge un impulso (la durata minima ammissibile è 1 μ s) a livello alto, a meno di diversa impostazione, le uscite 1A, 1B, 2A e 2B forniscono gli impulsi per comandare lo spostamento del rotore del motore di uno step o frazione. Concludiamo la panoramica sui driver con il piedino /SLEEP, il quale attiva la modalità di riposo (logica attiva e driver spenti) e con RST, che azzerà il controllore che governa i driver e pone le uscite di comando dei motori (1A, 1B, 2A, 2B) a zero logico anche se STEP continua a ricevere impulsi. Nella nostra applicazione RST e SLEEP sono disattivate perché SLEEP dispone di un resistore interno di pull-up e unendole sono entrambe a 1.

Lo shield è predisposto per ricevere due alimentazioni distinte: VMOT si applica all'apposita morsettiera ed è quella a 12 volt cc per il motore, mentre 5V (riportata sul connettore di espansione della Raspberry Pi) alimenta la logica del driver e il line-driver ULN2003; quest'ultima tensione va regolata sul converter DC/DC utilizzato allo scopo.

REALIZZAZIONE PRATICA

Lo shield si ottiene montando sul PCB (è a doppia faccia e ne trovate le tracce rame sul nostro sito www.elettronica.in.it) inciso e forato i pochi componenti occorrenti, vale a dire l'integrato ULN2003 (preferibilmente su zoccolo) i quattro condensatori e il modulo di controllo dello stepper-motor (da zoccolare utilizzando allo scopo due strip femmina a passo 2,54 mm); per l'orientamento degli elementi polarizzati riferitevi al piano di montaggio visibile in queste pagine.

Completate la scheda con i pin-strip per realizzare i jumper e le morsettiere, nonché con il doppio strip femmina a pin lunghi che salderete come mostrato nelle foto del prototipo, in modo che risulti passante, nelle piazzole corrispondenti.

Per quanto riguarda il cablaggio, la Raspberry Pi dev'essere alimentata tramite lo shield; alla morsettiera 12V di quest'ultimo dovete applicare 12 volt in continua prelevati da un alimentatore stabilizzato che fornisca tale tensione e almeno 1,5 ampere di corrente, mentre a quella siglata 5V dovete fornire 5 Vcc stabilizzati, ottenuti da un regolatore, che nel nostro caso è un DC/DC converter che accetta in ingresso gli stessi 12 V forniti allo shield ed eroga 5V.

Il regolatore (si acquista da www.futurashop.it con il codice MP1584ENDCDC) è in realtà un convertitore switching da 5V da 3A.

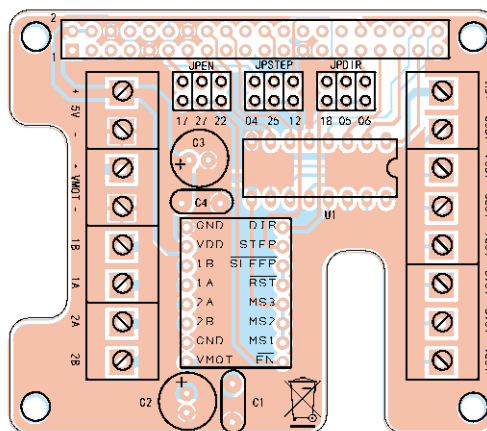
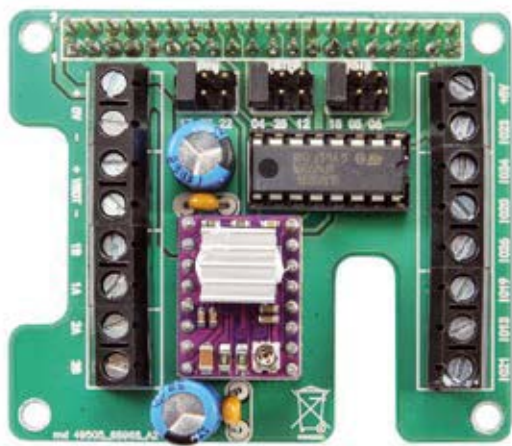
LA MECCANICA

Passiamo adesso alla struttura meccanica, formata da parti in plexiglass; i blocchi principali sono la base, che contiene al centro l'elettronica e un braccio in cima al quale è snodata la telecamera, ed un piatto girevole che sorregge l'oggetto.

Il piatto può essere ottenuto da un disco di plexiglass o forex spesso 4 mm; individuate il centro e riportatelo sotto forma di punto con un pennarello indelebile, perché vi sarà utile a centrare l'oggetto e a ottenere una ripresa della rotazione che sia perfetta, circolare e non eccentrica.

Il piatto rotante è sostenuto da quattro colonnine che lo vincolano al cerchio interno di una ralla metallica; una traversa avvitata a due di queste colonnine lo mette in rotazione, a sua volta vincolata tramite un mozzo cilindrico, avvitato mediante un grano all'albero di un motore passo-passo NEMA17,

piano di MONTAGGIO



a sua volta pilotato dallo shield montato sulla Raspberry Pi.

La parte esterna della ralla è avvitata a una base di sostegno composta da due dischi in plexiglass uniti e spazati da quattro colonnine distanziali.

Il motore passo-passo è avvitato con la flangia anteriore al disco superiore in plexiglass.

COME FUNZIONA

Per acquisire le foto dell'oggetto, il nostro sistema lo fa ruotare su un piatto mentre la fotocamera scatta ad intervalli regolati fotogrammi o, se impostato, riprende un filmato.

Per fotografare l'intera superficie l'oggetto viene fatto ruotare di un angolo pari al giro completo (360 gradi) ma dalla pagina web è anche possibile impostare un numero superiori di giri.

IL SOFTWARE

Passiamo ora alla parte relativa ai programmi che ci servono a far funzionare il sistema: scarichiamo Raspbian con l'ambiente desktop (al momento della scrittura di questo articolo l'ultima versione è "Raspbian Buster with desktop" del 13/02/2020) dal sito www.raspberrypi.org/downloads/raspbian e scriviamo l'immagine sulla scheda MicroSD con il programma Win32DiskImager.

Alla fine della scrittura, prima di estrarre la scheda MicroSD dal PC, creiamo un file vuoto nella cartella principale della scheda e rinominiamolo in "ssh" (senza estensione): questo ci servirà per abilitare la connessione SSH utile per l'installazione del software.

Creiamo anche il file "wpa_supplicant.conf" che permetterà al Raspberry Pi di connettersi alla nostra rete WiFi e all'interno inseriamo le seguenti righe modificandole opportunamente:

```
country=IT
ctrl_interface=DIR=/var/run/wpa_supplicant
GROUP=netdev

network=
{
    ssid="INSERIRE_NOME_RETE_WIFI"
    psk="INSERIRE_PASSWORD_RETE_WIFI"
    key_mgmt=WPA-PSK
}
```

Inseriamo ora la scheda MicroSD nella Raspberry Pi, alimentiamo il tutto e, dopo aver individuato l'IP della nostra board tramite l'interfaccia del router o con programmi come Advanced IP Scanner, connet-

tiamoci in SSH con Putty o MobaXTerm inserendo l'utente "pi" e la password predefinita "raspberrypi" quando richiesto.

Lanciamo il comando:

```
sudo raspi-config
```

e, navigando con la tastiera tra i vari parametri, effettuiamo la seguente configurazione di sistema:

- Network Options -> Hostname: capture360;
- Boot Options -> Desktop/CLI: Desktop autologin;
- Localisation Options -> Change Locale: it_IT.UTF-8 UTF-8, selezionare it_IT.UTF-8;
- Localisation Options -> Change Timezone: Europe, selezionare Rome;
- Localisation Options -> Change Keyboard Layout;
- Localisation Options -> Change WiFi Country: IT;
- Interfacing Options -> Camera: Yes;
- Advanced Options -> Expand Filesystem.

Abbiamo così configurato la Raspberry Pi in lingua italiana, abilitato la Pi Camera (essenziale per questo progetto), impostato l'avvio in modalità desktop ed espanso il file system su tutto lo spazio disponibile nella scheda MicroSD; quindi riavviamo il sistema per rendere effettive le modifiche con il comando `sudo reboot`.

Connettiamoci ancora in SSH all'IP precedentemente trovato oppure inserendo l'hostname "capture360", spostiamoci nella home dell'utente Pi:

```
cd /home/pi
```

Scarichiamo il software da GitHub:

```
git clone https://github.com/open-electronics/capture360.git
```

Spostiamoci nella cartella del software, impostiamo i permessi di esecuzione e lanciamo l'installazione con questi tre comandi:

```
cd capture360/
sudo chmod a+x install.sh
sudo bash install.sh
```

Quindi attendiamo qualche minuto per il termine dell'installazione, dopo la quale Raspberry Pi si riavvierà automaticamente.

Il file "*install.sh*", dopo aver verificato la disponibilità della connessione ad Internet, aggiorna la lista dei pacchetti software nei vari repository e installa *gpac*



CARATTERISTICHE TECNICHE

Telecamera: 5 megapixel con ottica regolabile	Risoluzione video 1920x1080 a 16:9
Rotazione oggetto 360° tramite stepper-motor	Connessione dati WiFi
Tensione di alimentazione: 12 Vcc	Interfaccia utente WebApp
Corrente assorbita: 1,5 ampere	

(in questo progetto viene utilizzato per convertire i video registrati dalla Pi Camera in mp4), *imageio* (libreria Python per la gestione delle immagini) e *flask* (framework Python su cui girerà tutta la nostra web application); alla fine di queste operazioni crea un file systemd che permette l'avvio automatico del server Python-Flask ad ogni accensione. L'interfaccia grafica della nostra web app è contenuta nel file "templates/index.html" ed è stata realizzata con Materialize (*materializecss.com*): un framework CSS che consente di dare uno stile moderno e responsivo alle pagine web. All'interno della cartella "static" troviamo il maggior numero di elementi di questa applicazione:

- cartella "config": contiene il file "config.json" dove vengono salvati i parametri che dovranno essere richiamati alla pressione dei pulsanti trigger e il GPIO a cui è connesso il pulsante di trigger fisico;
- cartella "css": contiene i file di stile per la web app;
- cartella "img": contiene le immagini di servizio della web app (ad esempio la favicon della pagina web);
- cartella "js": contiene i file Javascript del framework JQuery e Materialize, inoltre è presente il file "foto360.js" che consente la comunicazione bidirezionale tra la pagina web e il server in Python (ad esempio premendo un pulsante di avvio cattura sull'interfaccia web, viene richiamata una funzione Javascript che a sua volta esegue una chiamata POST al server in Python per fare partire effettivamente la cattura delle immagini);
- cartella "preview": contiene il file "preview.jpg" che

viene continuamente sovrascritto dalle nuove foto generate dalla Pi Camera, questo consente di visualizzare un'anteprima praticamente in tempo reale di quello che stiamo inquadrando;

- cartella "shoot": contiene l'elenco di tutte le catture effettuate (foto, GIF o video) pronte per essere scaricate;
- cartella "widget": contiene i file HTML e Javascript utile per creare il widget HTML che verrà scaricato insieme alle foto per consentire l'effetto di rotazione al trascinamento del puntatore.

Ultimo, ma assolutamente indispensabile, è il file "server.py" che rappresenta il vero e proprio motore di questa applicazione e viene lanciato all'avvio del sistema: non solo si occupa di creare la web app con cui noi utilizziamo Capture360 ma gestisce anche l'acquisizione di foto e video e la successiva rielaborazione per trasformarle in widget HTML, GIF o video in formato mp4.

Analizzando questo file nel dettaglio possiamo notare che è composto da diverse funzioni: queste funzioni vengono principalmente richiamate dal file Javascript "foto360.js" che abbiamo analizzato in precedenza tramite delle chiamate POST.

Le funzioni Start, Trigger e TriggerButton fanno partire la cattura da diversi entry point: la prima viene chiamata se premiamo il pulsante Start

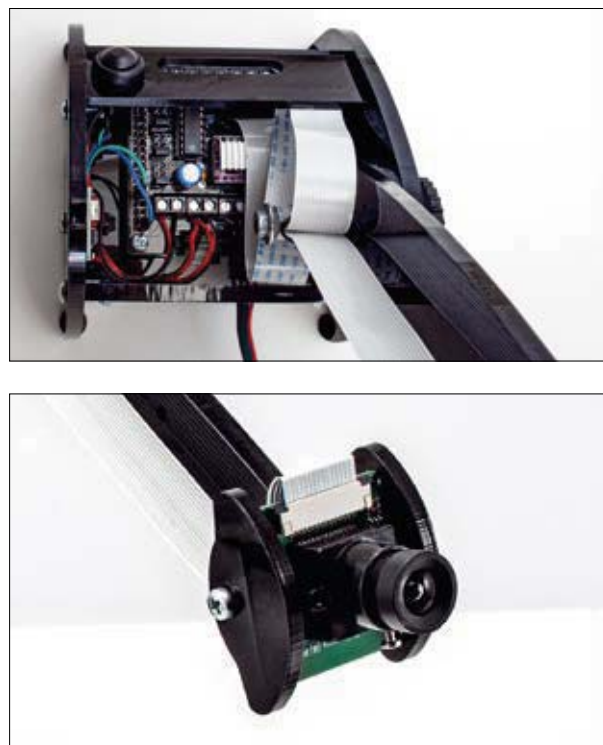


Fig. 1
L'elettronica
nella base e
la telecamera
in cima al
braccio.



Fig. 2
Schermata iniziale
della Web App.

nelle sezioni Photo, GIF o Video, la seconda viene chiamata quando premiamo il pulsante di trigger con il fulmine nell'interfaccia web mentre la terza viene chiamata quando premiamo il pulsante trigger fisico connesso al Raspberry Pi; queste tre funzioni richiamano la funzione principale CreateShooting passando dei parametri come la risoluzione, il numero delle foto per giro, gli FPS e il numero di giri della piattaforma.

Viene così creata una nuova cartella rinominata con data, ora e modalità di cattura nella cartella "shoot"; successivamente, viene richiamata sempre dal Javascript la funzione Shoot che si occupa di catturare le immagini con i parametri che sono stati impostati. Nel caso della foto o della GIF, la funzione divide i 360 gradi con il numero di foto per giro richieste poi fa ruotare di qualche grado la piattaforma, cattura una foto e ripete questa operazione fino a giro completo (per la GIF compie tanti giri quanti ne sono stati impostati); nel caso del video viene fatta partire la cattura con la Pi Camera e mentre la piattaforma ruota di 360 gradi (sempre per il numero di giri impostati).

Finita la cattura, la stessa funzione Shoot, in modalità Photo crea uno ZIP per le foto inserendo anche il widget HTML, in modalità GIF crea la GIF unendo tutte le foto con la libreria *imageio* mentre in modalità Video converte il video dal formato h264 al formato mp4 con il programma *MP4Box* contenuto in *gpac*. La funzione Status viene richiamata ogni due secondi da *"foto360.js"* e ritorna lo stato attuale del sistema (se è in corso una cattura, la percentuale di completamento e la foto di anteprima), mentre la funzione Refresh viene chiamata ogni tre secondi dallo stesso file Javascript e ritorna l'elenco completo delle catture pronte per essere scaricate.

La funzione Stop consente, tramite i pulsanti Stop della web app, di fermare una cattura già avviata.

UTILIZZO

Per utilizzare Capture360 alimentiamo il tutto e attendiamo qualche secondo per la connessione alla rete WiFi; tramite browser (in locale su Raspberry Pi o su un dispositivo connesso alla stessa rete) raggiungiamo l'indirizzo <http://capture360:5000> oppure http://IP_RASPBERRY:5000.

L'interfaccia della web app è composta principalmente da due sezioni (**Fig. 2**): quella superiore ci mostra l'elenco delle catture completate (pronte per essere scaricate o eliminate) e l'anteprima della Pi Camera; la sezione inferiore contiene la riga di impostazioni del pulsante trigger e le tre schede **Photo**, **GIF** e **Video** da cui è possibile impostare, iniziare o arrestare una cattura e il relativo pulsante di trigger. In fondo troviamo il pulsante per spegnere completamente il sistema.

Posizioniamo l'oggetto sulla piattaforma rotante e, tramite l'anteprima della Pi Camera, regoliamo l'inquadratura e le luci dell'ambiente fino a quando non siamo soddisfatti.

Apriamo ad esempio la scheda GIF (**Fig. 3**) settiamo la risoluzione preferita e il numero di scatti per giro (shoots), premiamo poi sul pulsante Start per avviare la cattura: l'immagine di anteprima della Pi Camera verrà oscurata da un'altra immagine che ci avverte della cattura in corso e la barra orizzontale nella sezione GIF mostrerà il progresso della cattura (**Fig. 4**) che si fermerà quando saranno state scattate il numero di foto scelte e la piattaforma avrà compiuto le rotazioni impostate.

A questo punto, l'archivio della nostra cattura comparirà nell'elenco in alto a sinistra pronto per essere scaricato.

Se avete scelto di scattare le foto (utilizzando la sezione Photo) dopo aver effettuato il download troveremo il widget HTML, la cartella con i file Javascript per fare funzionare il widget e le foto che ci servono nella cartella *"img"*.

Per provare il widget occorre aprire il file *"index.html"* con qualsiasi browser, posizionare il puntatore sulla foto e trascinarlo a destra e sinistra tenendo premuto il tasto sinistro del mouse: avremo così l'effetto rotazione che potrà essere utilizzato nel nostro sito per aggiungere dinamicità alla pagina web e meglio mostrare l'oggetto che intendiamo condividere o vendere sul nostro sito.

La schermata per acquisire le foto è visibile in **Fig. 5** e, mentre quella per i Video la potete vedere in **Fig. 6**.

Tenete presente che al crescere della risoluzione e delle foto/giro impostate, occorrerà più tempo alla Raspberry Pi per processare tutto il materiale al



Fig. 3
Scheda GIF
e relative
impostazioni.

termine della cattura (creazione ZIP, creazione GIF o conversione video) arrivando anche a elaborare per diversi minuti.

L'utilizzo del pulsante "Set trigger button" risulta comodo nel caso in cui dovessimo effettuare tante catture in sequenza e lontani dall'interfaccia grafica: in questo caso, dopo aver settato i vari parametri come resolution e shoots per turn nella modalità Photo, clicchiamo su questo pulsante: immediatamente i valori visibili nella riga "Trigger button setup:" vengono aggiornati e siamo pronti ad avviare le catture sia con il tasto con il fulmine dall'interfaccia grafica, sia con il pulsante fisico che potremo posizionare dove ci è più comodo.

I valori del pulsante trigger vengono mantenuti anche dopo lo spegnimento del sistema.

UN PO' DI HACKING

Proviamo ora a modificare il progetto, ad esempio cambiamo il GPIO dove normalmente è connesso il pulsante fisico di trigger; per fare questo connettiamoci in SSH al Raspberry Pi, spostiamoci sotto alla cartella del file di configurazione "*config.json*" e apriamolo con nano:

```
cd /home/pi/capture360/static/config/
sudo nano config.json
```

Individuiamo il parametro "TriggerGPIO" e al posto di 10 inseriamo 3; successivamente chiudiamo e salviamo con CTRL+X, Y e invio.

Spegniamo il sistema, spostiamo il cavo del pulsante dal GPIO 10 al GPIO 3 (mantenendo inalterata la posizione del cavo connesso al 3,3 volt) e riavviamo il sistema per verificare il funzionamento della modifica.

In alcune occasioni potremmo avere bisogno di catturare foto o video con risoluzioni che non sono

presenti di default; per aggiungere una risoluzione occorre prima verificare l'elenco delle risoluzioni disponibili presente a questo indirizzo:

<https://www.raspberrypi.org/documentation/raspbian/applications/camera.md>.

Dopo aver verificato la presenza della nuova risoluzione nell'elenco, connettiamoci in SSH ed editiamo il file "index.html" con questi due comandi in sequenza:

```
cd /home/pi/capture360/templates/
sudo nano index.html
```

Fig. 4
Schermata
dell'app durante
la cattura.

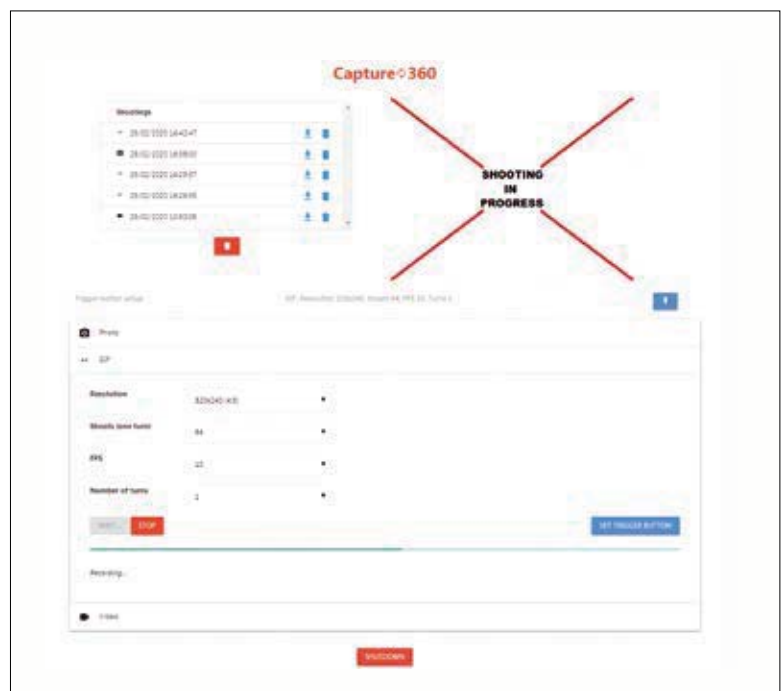


Fig. 5
Scheda Photo
e relative
impostazioni.

Nel caso volessimo aggiungere una nuova risoluzione alla modalità Video, individuiamo l'elemento select con id = "VideoResolution" e aggiungiamo una opzione di questo tipo:

```
<option value="1640x922">1640x922 (16:9)</option>
```

Chiudiamo e salviamo il file con CTRL+X, Y e invio, successivamente riavviamo il sistema con il comando *sudo reboot*.

A questo punto, nel parametro resolution della scheda Video, potremo settare la risoluzione di 1640x922.

Per inserire nuove risoluzioni nella modalità Photo, l'elemento select ha l'id = "PhotoResolution", mentre per la modalità GIF la select ha l'id = "GIFResolution"; possiamo aggiungere ulteriori opzioni anche agli altri parametri dell'interfaccia come il numero di foto per ogni giro della piattaforma (PhotoShoots e GIFShoots) o il numero di FPS (GIFFPS e VideoFPS).

Passiamo ora a una modifica più completa che ci

consentirà di controllare l'illuminazione del nostro set fotografico: con questa implementazione saremo in grado di accendere e spegnere delle luci direttamente dalla web app.

Per prima cosa colleghiamo un modulo relè ai GPIO del Raspberry Pi e facciamo passare attraverso a un relè l'alimentazione di una striscia a LED bianchi o di alcuni faretti che posizioneremo in maniera ottimale per ottenere un'illuminazione omogenea dell'area della piattaforma rotante, così da annullare le varie ombre possibili.

Come uscita per il comando della strip a LED utilizzate il pin 33 (GPIO13) della Raspberry Pi. Connettiamoci ora in SSH ed editiamo il file "index.html":

```
cd /home/pi/capture360/templates/  
sudo nano index.html
```

Posizioniamoci in fondo al file, prima del blocco "<div class = "row">" che contiene il pulsante Shutdown e inseriamo questo nuovo blocco HTML:

Fig. 6
Scheda Video
e relative
impostazioni.

```

<div class = "row">
  <div class="col s6" align="center">
    <a onclick="SetLight(1);"
class="waves-effect waves-light btn">Light ON</a>
  </div>
  <div class="col s6" align="center">
    <a onclick="SetLight(0);"
class="waves-effect waves-light btn">Light OFF</a>
  </div>
</div>

```

Chiudiamo e salviamo il file con CTRL+X, Y e invio: abbiamo inserito nell'interfaccia della web app i due pulsanti che utilizzeremo per accendere e spegnere le luci.

Spostiamoci ora sotto alla cartella dei file Javascript e modifichiamo il file "foto360.js":

```

cd /home/pi/capture360/static/js/
sudo nano foto360.js

```

Posizioniamoci in fondo al file e inseriamo questa funzione:

```

function SetLight(value){
  $.ajax({
    type: "POST",
    url: "/light",
    data : {
      status: value
    },
    cache: false,
  }).done(function( data ) {
    if(value == 1) {
      M.toast({html: "Light ON."});
    } else {
      M.toast({html: "Light OFF."});
    }
  });
}

```

Chiudiamo e salviamo il file con CTRL+X, Y e invio: questa funzione, richiamata alla pressione di uno dei due pulsanti, si occuperà di ricevere un valore 0/1 (value) e lo passerà al server Flask (il file server.py) mediante una chiamata POST alla url "light"; successivamente mostrerà un messaggio di conferma della luce spenta o accesa. Infine editiamo il file "server.py" per creare la funzione che verrà richiamata dal file Javascript precedentemente modificato:

```

cd /home/pi/capture360/

```

```

sudo nano server.py

```

Spostiamoci quasi alla fine del file e, sotto alla funzione Trigger(), inseriamo quest'altra funzione:

```

@app.route("/light", methods = ['POST'])
def Light():
    Status = request.form["status"]
    if(Status == 1):
        GPIO.output(5, GPIO.HIGH)
    else:
        GPIO.output(5, GPIO.LOW)
    return ""

```

Aggiungiamo anche la seguente riga prima della penultima "app.run(...):GPIO.setup(5, GPIO.OUT) Ricordiamoci di sostituire il numero 5 (sia nella funzione Light che nella riga appena inserita) con il numero del GPIO a cui abbiamo collegato il relè che intendiamo controllare, chiudiamo e salviamo il file con CTRL+X, Y e invio, infine riavviamo il sistema con il comando sudo reboot.

Ora che abbiamo realizzato il nostro piccolo set fotografico rotante potremo catturare gli oggetti di una collezione, dei manufatti o i prodotti della nostra attività commerciale (nel caso in cui avessimo un negozio online) e mostrarli da diverse angolazioni anche in maniera interattiva utilizzando il widget HTML.

Bene, con questo abbiamo terminato; vi lasciamo ricordandovi che il progetto qui proposto si presta a varie personalizzazioni anche sotto l'aspetto meccanico, secondo la vostra fantasia e la voglia di hackerarlo. ■



Cosa occorre?

Tutto il materiale per realizzare il sistema Capture360 è disponibile presso Futura Elettronica. Il kit (cod. CAPTURE360) comprendente tutta la parte elettronica (Raspberry Pi 3 A+, telecamera, piattaforma rotante con stepper motor e scheda driver) è disponibile a Euro 185,00.

I componenti che compongono il progetto sono disponibili anche separatamente. In particolare Raspberry Pi 3 A+ (cod. RPI3MODELAPLUS) è venduto a Euro 34,90, la telecamera da 5Mpx (cod. RPICAMERA-FOC) è disponibile a Euro 29,00, la scheda driver (cod. FT1255K) costa Euro 21,00 e la piattaforma rotante per stepper motor (cod. ROTATINGBASE-ST) è venduta a Euro 29,50. Tutti i prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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



La tensione giusta per ogni progetto!

Scegli il convertitore di tensione DC-DC che meglio si adatta alle tue esigenze.

CONVERTITORI STEPDOWN DC-DC

Da 7-28V a 5V - 3A



€ 2,50
cod. MP1584ENDCDC

Da 5-36V a 1,25-32V - 5A



€ 14,00
cod. YB728

Da 5-23V a 0-16,5V
da 2A a 3A



€ 11,00
cod. DCDISPAVDOWN

Da 4-40V
a 1,25-37V - 2A



€ 6,90
cod. STEPDOWNDISP

Da 9-35V
a 5V - 5A



€ 6,50
cod. DCDCSTEPDOWN

Da 5-32V
a 0,8-30V - 5A



€ 5,90
cod. STEPD5V32

Da 3-40V
a 1,5-35V - 3A



€ 4,30
cod. MODULODCDC

CONVERTITORI STEPUP DC-DC

Da 3,5-30V
a 4-30V - 2A



€ 6,50
cod. STEPUP30V

Da 0,9-5V
a 5V - 0,6A



€ 3,00
cod. STEPUPUSB

Da 3-4,5V a 5V - 1A



€ 4,90
cod. DCDC5V

DISPONIBILE ANCHE
NELLE VERSIONI:
da 3-6V a 9V - 1A
cod. DCDC9V € 4,90
Da 3-6V a 12V - 0,4A
cod. DCDC12V € 4,90

CONVERTITORE STEPUP e STEPDOWN DC-DC

Da 4-35V a 1,25-25V - 2A



€ 16,00
cod. STEPUPDOWN



**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

Prezzi IVA inclusa.

ThingSpeak



di MATTEO DESTRO

ENERGY METER IOT



Interfacciamo il nostro Energy Meter al mondo del web per portare sulla piattaforma ThingSpeak le misure e vederne i risultati in comodi grafici.

Q

ualche numero addietro abbiamo introdotto l'integrato MCP39F511, capace di eseguire misure elettriche in impianti elettrici monofase attraverso la scheda FT1346M. Il tutto l'abbiamo integrato con la board GSM (fascicolo n° 240) e relativa libreria di gestione per affacciare il sistema al mondo esterno. Ora vogliamo mostrarvi come abbinare tale hardware a una piattaforma web per la gestione dei dispositivi IoT che si chiama "ThingSpeak" (<https://thingspeak.com>). Per l'occasione, abbiamo quindi scritto un apposito sketch che sfrutta la libreria GSM con integrati i comandi AT per la gestione delle connessioni HTTP e la libreria di gestione dell'integrato MCP39F511; in queste pagine vi spiegheremo come integrare le due librerie e creare su ThingSpeak una pagina di visualizzazione dei dati acquisiti. Dato che ricorreremo all'HTTP per la connessione alla piattaforma ThingSpeak, potremo utilizzare solo i moduli GSM della SIMCOM: nello sketch abbiamo usato il SIM800C.

➔ **Fig. 1**
Creare un nuovo canale.



INTRODUZIONE THINGSPEAK

ThingSpeak è una piattaforma che offre un servizio per l'aggregazione, la visualizzazione e l'analisi di un flusso di dati proveniente da dispositivi elettronici identificati come IOT (Internet of Things). E' quindi possibile inviare dei dati dal proprio dispositivo alla piattaforma e visualizzare i dati ricevuti

 A screenshot of the 'New Channel' form in the ThingSpeak web interface. The form contains several input fields and checkboxes. At the top, there is a 'Name' field and a 'Description' field. Below these are eight 'Field' inputs, each with a label (Field 1 to Field 8) and a dropdown menu to select the data type. Below the fields is a 'Metadata' field and a 'Tags' field. Further down are fields for 'Link to External Site', 'Link to GitHub', and 'Elevation'. There are also checkboxes for 'Show Channel Location' and 'Show Video'. Below these are fields for 'Latitude', 'Longitude', and 'Video URL'. At the bottom, there is a 'Show Status' checkbox and a green 'Save Channel' button.

➔ **Fig. 2**
Il form di creazione del canale.

in tempo reale, eventualmente inviando allarmi mediante i servizi Twitter e Twilio. Inoltre la piattaforma mette a disposizione MATLAB Analytics, con il quale è possibile scrivere ed eseguire codice MATLAB per processare, analizzare e visualizzare i dati ricevuti. Questo permette ai progettisti di realizzare sistemi IoT senza impostare dei servizi server o scrivere software per il web.

L'utilizzo di ThingSpeak è gratuito per lo sviluppo di piccoli progetti per uso non commerciale, ad esempio con invio di un numero di messaggi verso la piattaforma inferiore a 3 milioni l'anno (una media di 8.200 al giorno).

Quando si attivano delle licenze a pagamento si comprano delle Units corrispondenti a 33 milioni di messaggi che possono essere processati e memorizzati in un anno o in altre parole circa 90.000 messaggi al giorno.

Inoltre le Units permettono di creare un numero finito di Channels.

Per la nostra applicazione abbiamo usato la versione Free, più che sufficiente per creare una demo.

Fatta questa breve introduzione alla piattaforma possiamo passare alla descrizione di come fare a creare un Channels con i suoi relativi campi.

Il primo passo, per chi non è ancora un utente abilitato, è registrarsi e per fare ciò si deve cliccare sulla voce "Sign Up" e seguire i passi indicati.

Una volta eseguita la registrazione si può procedere alla creazione di un nuovo Channel e per fare questo si deve cliccare sul pulsante "New Channel" come evidenziato dalla **Fig. 1**, sezione cerchiata in rosso. Comparirà un form da compilare con svariati campi, come indicato nella **Fig. 2**, i passi da seguire sono i seguenti:

- 1) dare un nome al nuovo Channel (campo Name); abbiamo deciso di assegnare il nome EnergyMeter;
- 2) inserire una descrizione del nuovo Channel da creare (campo Description); in questo caso abbiamo inserito come descrizione "MCP39F511 Energy Meter";
- 3) per ogni Channel si possono attivare otto campi differenti, ognuno di questi rappresenta una misura da salvare e processare; abbiamo attivato sei campi su otto; in particolare abbiamo i seguenti:
 - Field 1 → Voltage
 - Field 2 → Current
 - Field 3 → Power Factor
 - Field 4 → Active Power
 - Field 5 → Reactive Power
 - Field 6 → Apparent Power

- 4) se volete, potete inserire i dati di geo-localizzazione ovvero Elevation, Latitude e Longitude; con tali parametri verrà visualizzata una cartina geografica indicante la posizione del sistema di misura realizzato.

Le voci Metadata, Tags, Link to External Site e Link to GitHub non sono usati per questa applicazione. Concluse tutte queste inizializzazioni si deve cliccare sul pulsante "Save Channel".

Ora, sotto la voce "My Channels" comparirà il canale appena creato (**Fig. 3**): alla sinistra del suo nome compare un lucchetto chiuso, indicante che la visualizzazione è privata; è possibile configurare il Channel come pubblico (il lucchetto si apre) e quindi visibile dall'esterno. Quando si pubblica il Channel, questo può essere visibile da tutti o soltanto da utenti abilitati.

Cliccando sul Channel appena creato si apre una sezione in cui è possibile:

- modificare i parametri di configurazione del Channel appena creato cliccando sulla voce "Channel Settings";
- rendere pubblico/privato il Channel creato cliccando sulla voce "Sharing";
- configurare la visualizzazione dei dati ricevuti per ognuno dei "Field" creati nella sezione "Private View" (i dati visualizzati sotto questa voce saranno visibili solo nella sezione privata);
- configurare la visualizzazione dei dati ricevuti per ognuno dei "Field" creati nella sezione "Public View" (i dati visualizzati sotto questa voce saranno visibili a chiunque o a una cerchia ristretta di utenti selezionati);
- cliccando sulla voce "API Keys" si ha accesso alle chiavi di lettura/scrittura associate al Channel creato.

La chiave di scrittura è ovviamente diversa dalla chiave di lettura; la prima, associata all'indirizzo URL di connessione a ThingSpeak, permette di inviare i dati in formato stringa ai campi precedentemente creati. Ad esempio:

```
https://api.thingspeak.com/update?api_
key=89QC6FHH0AU5536Z&field1=233
```

Come si può osservare, nella stringa compare la chiave di scrittura assegnata al nostro Channel e in coda alla chiave ci sono i campi a cui si devono inviare i dati; nell'esempio c'è soltanto il "field1" a cui viene assegnato il valore 233.

Sotto la voce "Private View" è possibile configurare

quali dati visualizzare tra quelli a disposizione e come visualizzarli. Vediamone i comandi.

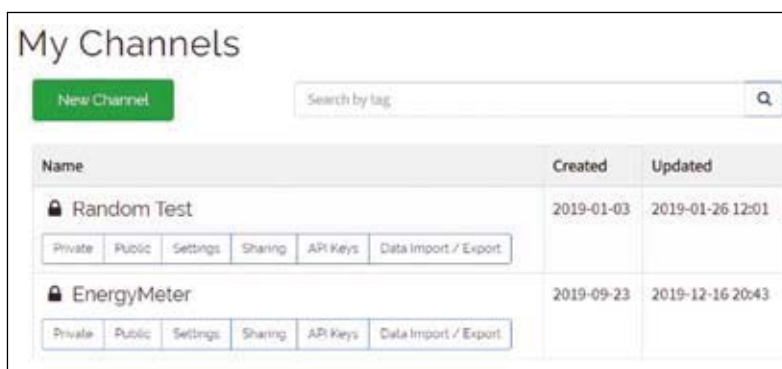
- "Add Visualization" → permette di associare un grafico per ogni "Field" configurato tra gli otto disponibili. Se si clicca sul comando viene proposta una finestra di selezione da cui scegliere il "Field" di cui si vogliono visualizzare i dati (**Fig. 4**). In questo caso è disponibile solo il "Field 3", se si clicca sul grafico e successivamente su "Save" verrà creato il grafico corrispondente. La **Fig. 5** mostra il grafico appena creato corrispondente al "Field 3".

Il grafico ovviamente può essere personalizzato e per accedere alla sezione di configurazione si deve cliccare sull'icona della matita in alto a destra del grafico. Così facendo si richiama l'apposita finestra di configurazione evidenziata dalla **Fig. 6**.

In questa finestra è possibile configurare:

- **Title** serve per assegnare un'etichetta al grafico corrispondente al "Field 3": "Power Factor";
- **X-Axis** serve per assegnare un'etichetta all'asse X: "Time";
- **Y-Axis** serve per assegnare un'etichetta all'asse Y: "PF";
- **Color** permette di impostare valore esadecimale e il colore del grafico;
- **Background** serve per impostare valore esadecimale e colore dello sfondo del grafico;
 - **Type** imposta la tipologia di grafico: Line, Bar, Column, Spline e Step;
 - **Dynamic?** serve per impostare se il grafico debba essere dinamico oppure no (dinamico significa che viene automaticamente aggiornato ogni 15 secondi);
 - **Days** imposta quanti periodi da 24 ore bisogna aspettare prima di includere il valore ricevuto nei feed;

Fig. 3
Il canale creato.



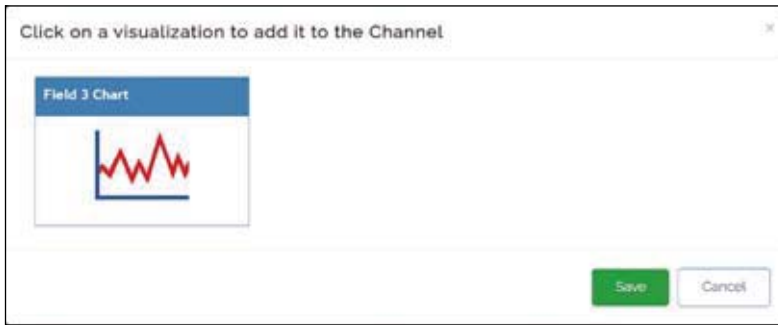


Fig. 4
Scelta del
field di cui
visualizzare i
dati.



Fig. 5
Grafico
relativo al
Field 3.

- **Results** imposta il numero di campioni da visualizzare nel grafico;
- **Timescale** serve per impostare ogni quanti minuti si deve visualizzare un dato tra quelli ricevuti (facoltativo);
- **Average** imposta ogni quanti minuti si deve calcolare e visualizzare il valore medio dei dati ricevuti;
- **Median** imposta ogni quanti minuti si deve calcolare e visualizzare la mediana dei dati ricevuti;
- **Sum** serve per impostare ogni quanti minuti si deve calcolare e visualizzare la somma dei dati ricevuti;
- **Data Min** serve per impostare il valore minimo atteso;
- **Data Max** serve a impostare il valore massimo atteso;
- **Y-Axis-Min** imposta il valore minimo da visualizzare sull'asse Y;
- **Y-Axis Max** serve a impostare il valore massimo da visualizzare sull'asse Y;
- **Add Widgets**; permette di configurare altri metodi di visualizzazione dei dati ricevuti; cliccando su questo comando appare una finestra di

selezione dalla quale è possibile scegliere:

- **Gauge** serve per impostare una visualizzazione grafica tipo tachimetro (la **Fig. 7** mostra la configurazione usata per la visualizzazione della tensione di rete); ci sono diversi campi che si devono compilare al fine di ottenere il risultato voluto tra cui:
 - **Name** imposta il nome da assegnare al visualizzatore (in questo caso "Voltage");
 - **Field** serve a impostare il "Field" associato da cui prelevare i dati da visualizzare; qui facciamo riferimento al "Field 1";
 - **Min** serve per impostare il valore minimo; il valore minimo atteso è 0 V;
 - **Max** serve per impostare il valore massimo; ci si aspetta come valore massimo 350 V;
 - **Units** serve per impostare l'unità di misura (in questo caso volt "V");
 - **Tick Interval** serve per impostare il delta di visualizzazione dei numeri sul tachimetro. In questo caso abbiamo impostato step da 25 unità in modo da rendere leggibili i valori senza che questi si possano sovrapporre;
 - **Update Interval** imposta ogni quanto tempo si deve eseguire un refresh del valore da visualizzare (impostato a 15 secondi);
 - **Range** serve per impostare sezioni distinte di visualizzazione sul tachimetro; qui la sezione centrale è tra 180 V e 250 V (valore normale) e le due esterne tra 0 V e 180 V (tensione di rete troppo bassa) e tra 250 V e 350 V (tensione di rete troppo alta) hanno un diverso colore;
- **Numeric Display** serve per impostare una visualizzazione di tipo numerico (la **Fig. 8** mostra la configurazione usata per la visualizzazione della potenza attiva); anche qui ci sono diversi campi da impostare:
 - **Name** serve per impostare il nome da assegnare al visualizzatore (in questo caso, "Active Power");

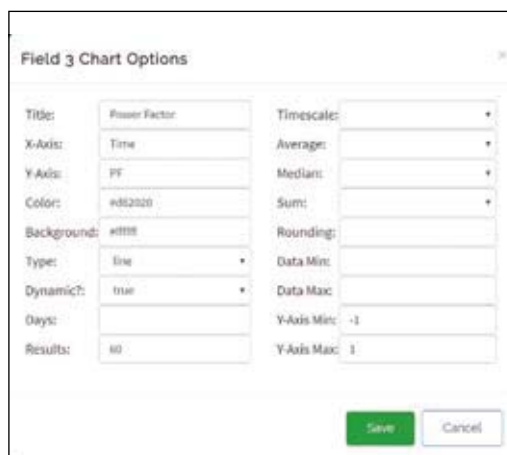


Fig. 6
Finestra di
configurazione.

- **Field** serve per impostare il "Field" associato da cui prelevare i dati da visualizzare (qui facciamo riferimento al "Field 4");
- **Update Interval** serve per impostare ogni quanto si deve eseguire un refresh del valore da visualizzare (impostati 15 secondi);
- **Units** unità di misura (watt "W");
- **Data Type** serve per impostare se il valore da visualizzare è di tipo intero o con decimali; in quest'ultimo caso va scritto quanti decimali si desiderano (in questo caso tre);
- **Lamp Indicator** serve per impostare un allarme ed è come un LED normalmente spento che si accende se il dato ricevuto è sopra o sotto soglia (dipende dalle impostazioni);
- **Export recent data**; permette di esportare i dati ricevuti in JSON, XML e CSV. È quindi possibile esportare tutti i "Field" in un unico file usando le opzioni della riga uno "EnergyMeter Channel Feed" oppure ogni canale singolarmente usando le righe successive;
- **MATLAB Analysis**; ci permette di creare del codice MATLAB per l'analisi dei dati ricevuti;
- **MATLAB Visualization** consente di creare del codice MATLAB per l'analisi e la visualizzazione dei dati ricevuti.

Sotto la voce "Public View" possiamo configurare, come già fatto per la sezione privata, la visualizzazione dei dati ricevuti dai vari "Field" con la differenza che questi sono visibili all'esterno. Per rendere visibili i dati si deve richiamare la sezione "Sharing" e seguire le istruzioni. La configurazione della sezione pubblica è identica a quella privata e quindi non ci soffermeremo oltre.

La **Fig. 9**, la **Fig. 10** e la **Fig. 11** mostrano come abbiamo configurato il nostro "Channel" per la visualizzazione dei dati ricevuti dall'integrato MCP39F511. Nell'esempio abbiamo voluto visualizzare le misure di Tensione, Corrente e fattore di potenza sia tramite grafico che mediante indicatore a lancetta. Stesso discorso per le misure di potenza Attiva, Reattiva e Apparente con la differenza che per queste abbiamo anche usato la visualizzazione numerica. In fondo alle misure compare la localizzazione geografica. Come si nota dalla **Fig. 11** abbiamo introdotto un grafico generato tramite MATLAB (3-Day Active Power Comparison) che correla le acquisizioni della potenza attiva relative a tre giorni, così da poter fare un confronto. I giorni possono anche essere di più o di meno, in base al codice MATLAB. Per dare un'idea, nel **Listato 1** riportiamo il codice

Fig. 7
Opzioni
per il grafico.

Fig. 8
Configurazione
per la
visualizzazione
della potenza
attiva.

MATLAB usato per la generazione del grafico: le righe di codice che iniziano con "%" e sono colorate di verde identificano i commenti. Sono state dichiarate tre variabili che identificano ID del canale usato (readChannelID), il campo di cui si vuole fare il grafico (myFieldID) e la chiave di lettura dei dati (readAPIKey). La funzione "thingSpeakRead" si occupa di leggere i dati del canale selezionato nel giorno richiesto e la "plot" disegna il grafico. In fondo troviamo le stringhe associate al titolo del grafico, alle etichette assi X - Y e una breve legenda.

INTEGRARE LIBRERIE GSM E MCP39F511

Ciò detto, andiamo oltre: per realizzare lo sketch che invia le misure elettriche alla piattaforma "ThingSpeak" è necessario prima di tutto integrare e armonizzare le due librerie GSM e MCP39F511 affinché possano funzionare assieme senza generare conflitti con i vettori di interrupt inerenti alla gestione del TIMER 1, usato da entrambe per generare la base dei tempi utilizzata per la gestione





Fig. 9
Visualizzazione
di tensione,
corrente e
fattore di
potenza.

Fig. 10
Indicatori
di potenza
attiva,
reattiva e
apparente.

di tutte le variabili temporali usate nelle librerie. Infatti se prendiamo le librerie senza apportare nessuna modifica e le includiamo in uno sketch otterremo il seguente errore dal compilatore C:

```
(.text+0x0): multiple definition of `__vector_20'
```

Quindi è necessario implementare un'unica gestione del vettore di interrupt del TIMER 1. Cominciamo dalla libreria MCP39F511 e in particolare sui file "Isr_MCP39F511.h" e "Isr_MCP39F511.cpp". Nel file "Isr_MCP39F511.h" creiamo una direttiva al compilatore denominata:

```
#define ENABLE_ISR_TIMER1
```

che consente di abilitare/disabilitare le sezioni di codice inerenti alla gestione dell'interrupt TIMER 1. In altre parole, se la direttiva è commentata il codice di gestione dell'interrupt TIMER 1 sarà ignorato, viceversa sarà incluso nella libreria. Quindi quando si dovrà usare la libreria MCP39F511 da sola in uno sketch dovrà essere decommentata, mentre se va usata assieme a quella del GSM deve essere commentata. All'interno del file "Isr_MCP39F511.cpp" le funzioni interessate da tale direttiva sono:

```
void Isr_MCP39F511::EnableLibInterrupt(void)
void Isr_MCP39F511::EnableTimerInterrupt(void)
ISR(TIMER1_OVF_vect)
```

Oltre alla direttiva appena introdotta dobbiamo creare una nuova funzione dentro la quale dobbiamo mettere il codice usato nella gestione del vettore di interrupt del TIMER 1:

```
static volatile void Isr_MCP39F511::Timer1_
Interrupt(void)
```

Dentro la funzione abbiamo il codice di gestione delle variabili temporali necessarie alla libreria MCP39F511 che vedete nel **Listato 2**. A seconda dell'UART selezionata verrà utilizzata la giusta variabile temporale. Passiamo ora alla libreria di gestione dei moduli GSM e in particolare i file "Isr_GSM.h" e "Isr_GSM.cpp". Nel primo dobbiamo introdurre una direttiva al compilatore:

```
#define ENABLE_EXT_TIMER1
```

la quale ci permette di abilitare/disabilitare il codice per la gestione dell'interrupt TIMER 1 esterno, ovvero la parte di codice creata ex novo nella libreria MCP39F511. Quindi creiamo un puntatore a funzione diretto all'indirizzo in Flash dove è memorizzata la funzione di gestione dell'interrupt TIMER 1 della libreria MCP39F511:

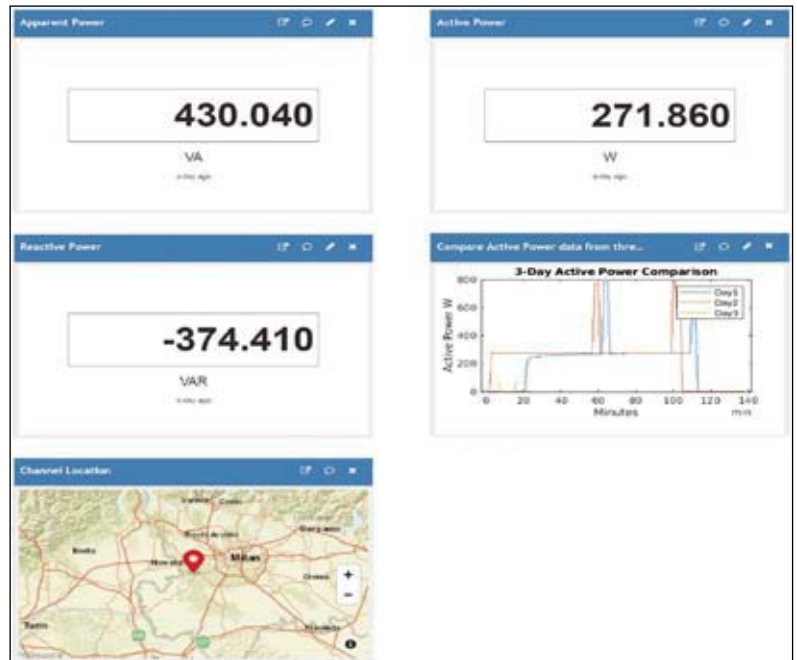
```
#ifdef ENABLE_EXT_TIMER1
    static void (* volatile __Ext_Timer1_Vector)
(void);
#endif
```

Il puntatore a funzione viene considerato solo se la direttiva al compilatore `ENABLE_EXT_TIMER1` non è commentata e quindi attiva. Il puntatore a funzione è attivo nel momento in cui la direttiva al compilatore `ENABLE_EXT_TIMER1` è attiva (ovvero non è commentata). Detto ciò dobbiamo creare una funzione, da richiamare durante l'inizializzazione dello sketch, la quale assegna al puntatore descritto prima l'indirizzo di partenza in FLASH dove è memorizzata la funzione di gestione dell'interrupt `TIMER 1` presente nella libreria di gestione dell'integrato `MCP39F511`:

```
#ifndef ENABLE_EXT_TIMER1
    void Ext_Timer1_Vector(void (*function)
(void));
#endif
```

Anche questa è attiva se la direttiva lo permette. Il codice presente in questa funzione è:

```
#ifndef ENABLE_EXT_TIMER1
    void Isr_GSM::Ext_Timer1_Vector(void (* volatile
function)(void))
    {
        __Ext_Timer1_Vector = function;
    }
#endif
```



e serve per caricare nel puntatore a funzione l'indirizzo in Flash della funzione da richiamare. Ora abbiamo l'indirizzo cui puntare e non ci rimane che aggiungere all'interno della gestione dell'interrupt `TIMER 1` presente nella libreria `GSM` il codice

Fig. 11
Visualizzazione
delle potenze con
geolocalizzazione.

Listato 1

```
% Channel ID to read data from
readChannelID = 870302;
% Active Power Field ID
myFieldID = 4;
% One day date range
oneDay = [datetime('yesterday') datetime('today')];
% Channel Read API Key
% If your channel is private, then enter the read API key between the ' ' below:
readAPIKey = 'LU2P9AT2E0TD79Q8';
% Read Active Power Data
ActivePower_Day_1 = thingSpeakRead(readChannelID,'Fields',myFieldID, ...
'dateRange', oneDay, 'ReadKey',readAPIKey);
ActivePower_Day_2 = thingSpeakRead(readChannelID,'Fields',myFieldID, ...
'dateRange',oneDay-days(1),'ReadKey',readAPIKey);
ActivePower_Day_3 = thingSpeakRead(readChannelID,'Fields',myFieldID, ...
'dateRange', oneDay-days(2),'ReadKey',readAPIKey);
% Create array of durations
myTimes1 = minutes(1:length(ActivePower_Day_1));
myTimes2 = minutes(1:length(ActivePower_Day_2));
myTimes3 = minutes(1:length(ActivePower_Day_3));
% Visualize the data
plot(myTimes1,ActivePower_Day_1, myTimes2,ActivePower_Day_2, myTimes3, ActivePower_Day_3);
legend({'Day1', 'Day2', 'Day3'});
xlabel('Minutes');
ylabel('Active Power W');
title('3-Day Active Power Comparison');
```

necessario a richiamare la funzione desiderata:

```
ISR(TIMER1_OVF_vect) {
    TCNT1 = SLOWBASETIME;
    .....
    .....
#ifdef ENABLE_EXT_TIMER1
    if (__Ext_Timer1_Vector != NULL) {
        __Ext_Timer1_Vector();
    }
#endif
}
```

Con questa abbiamo concluso le modifiche alle due librerie. Ora all'interno dello sketch, durante la fase di inizializzazione (void setup()), dobbiamo inserire nel punto giusto la seguente riga di codice:

```
Isr.Ext_Timer1_Vector(Isr_39F511.Timer1_Interrupt);
```

Richiamando questa funzione ci siamo assicurati che durante la gestione del TIMER 1 da parte della libreria GSM vengano anche gestite le variabili temporali della libreria MCP39F511.

La libreria di gestione dell'integrato MCP39F511 oltre alle modifiche sopra citate è stata interessata da una ulteriore miglioramento dovuto all'introduzione di una serie di funzioni per il calcolo del valore medio su alcune misure elettriche. Tramite la direttiva al compilatore:

```
#define ENABLE_AVERAGE_CALC
```

si abilitano/disabilitano le funzioni per il calcolo del valore medio e relativi array di gestione. Oltre alla direttiva di cui sopra ne abbiamo introdotta un'altra per la definizione delle dimensioni che devono avere gli array necessari al calcolo del valore medio:

```
#define AVERAGE_ARRAY_SIZE 4
```

I valori ammessi per questa direttiva al compilatore sono 4, 8, 16 e 32. Più le dimensioni degli array aumentano e più memoria SRAM sarà necessaria. Quindi la dimensione da usare va scelta con accuratezza a seconda dell'applicazione che si deve realizzare (faremo un esempio concreto quando parleremo dello sketch per l'invio dei dati alla piattaforma "ThingSpeak"). Le funzioni per il calcolo del valore medio sono quelle descritte di seguito.

- **uint16_t MCP39F511_RmsVoltageRawAverage(void)** → calcola il valore medio per quanto riguarda la misura della tensione. La funzione lavora sui valori RAW letti dall'integrato MCP39F511 e ritorna un valore intero senza segno a 16 bit.
- **uint32_t MCP39F511_RmsCurrentRawAverage(void)** → Calcola il valore medio per quanto riguarda la misura delle correnti. La funzione lavora sui valori RAW letti dall'integrato MCP39F511 e ritorna un valore intero senza segno a 32 bit.
- **uint16_t MCP39F511_PowerFactorRawAverage(void)** → Calcola il valore medio per quanto riguarda la misura del fattore di potenza. La funzione lavora sui valori RAW letti dall'integrato

↓ Listato 2

```
//=====
// Timeout 8 bit
if (Isr_39F511.ZcdTimeOut > 0) { Isr_39F511.ZcdTimeOut--; }
//=====

//=====
// Timeout 16 bit
#if defined(HARDWARE_UART1_39F511) || defined(SOFTWARE_UART1_39F511)
    if (Isr_39F511.Uart1_TimeOut > 0) { Isr_39F511.Uart1_TimeOut--; } // TimeOut for hardware/software
serial COM 1
#endif
#if defined(HARDWARE_UART2_39F511) || defined(SOFTWARE_UART2_39F511)
    if (Isr_39F511.Uart2_TimeOut > 0) { Isr_39F511.Uart2_TimeOut--; } // TimeOut for hardware/software
serial COM 2
#endif
#if defined(HARDWARE_UART3_39F511) || defined(SOFTWARE_UART3_39F511)
    if (Isr_39F511.Uart3_TimeOut > 0) { Isr_39F511.Uart3_TimeOut--; } // TimeOut for hardware/software
serial COM 3
#endif
//=====
```


MCP39F511 e ritorna un valore intero senza segno a 16bit. Il valore calcolato deve poi essere convertito tra i valori consentiti -1 e +1 e necessita della funzione ausiliaria uint16_t CalcAveragePowerFactor(void).

- **uint16_t MCP39F511_LineFrequencyRawAverage(void)** → Calcola il valore medio per quanto riguarda la misura della frequenza di rete. La funzione lavora sui valori RAW letti dall'integrato MCP39F511 e ritorna un valore intero senza segno a 16 bit.
- **uint32_t MCP39F511_ActivePowerRawAverage(void)** → Calcola il valore medio per quanto riguarda la misura della potenza attiva. La funzione lavora sui valori RAW letti dall'integrato MCP39F511 e ritorna un valore intero senza segno a 32 bit.
- **uint32_t MCP39F511_ReactivePowerRawAverage(void)** → Calcola il valore medio per quanto riguarda la misura della potenza reattiva. La funzione lavora sui valori RAW letti dall'integrato MCP39F511 e ritorna un valore intero senza segno a 32 bit.
- **uint32_t MCP39F511_ApparentPowerRawAverage(void)** → Calcola il valore medio per quanto riguarda la misura della potenza apparente. La funzione lavora sui valori RAW letti dall'integrato MCP39F511 e ritorna un valore intero senza segno a 32 bit.

Le funzioni per il calcolo del valore medio sono simili, cambia tuttavia l'array su cui devono lavorare. Analizziamo la funzione per il calcolo del valore medio della tensione di rete:

```
uint16_t Cmd_MCP39F511::MCP39F511_
RmsVoltageRawAverage(void) {
    uint8_t Index = 0;
    uint32_t AverageTemp = 0;
    do {
        AverageTemp += VoltageRmsRaw_Array[Index++];
    } while (Index < AVERAGE_ARRAY_SIZE);
    return((uint16_t)(AverageTemp / AVERAGE_ARRAY_SIZE));
}
```

Come si può osservare, a parte le variabili locali, abbiamo il costrutto **do – while** che si occupa di eseguire la media sui valori contenuti nell'array corrispondente.

Più le dimensioni dell'array sono grandi e più tempo ci vorrà per calcolare il valore medio. La funzione, in questo caso, ritorna un valore intero senza segno a 16 bit.

Per caricare i valori nei rispettivi array si usano le

seguenti funzioni:

- **void MCP39F511_UpdateRmsVoltageRawArray(void)** → Carica nell'array "VoltageRmsRaw_Array" l'ultimo valore letto della tensione RMS;
- **void MCP39F511_UpdateLineFrequencyRawArray(void)** → Carica nell'array "LineFrequencyRaw_Array" l'ultimo valore letto della frequenza di rete;
- **void MCP39F511_UpdatePowerFactorRawArray(void)** → Carica nell'array "PowerFactorRaw_Array" l'ultimo valore letto del fattore di potenza;
- **void MCP39F511_UpdateRmsCurrentRawArray(void)** → Carica nell'array "CurrentRmsRaw_Array" l'ultimo valore letto della corrente RMS;
- **void MCP39F511_UpdateActivePowerRawArray(void)** → Carica nell'array "ActivePowerRaw_Array" l'ultimo valore letto della potenza attiva;
- **void MCP39F511_UpdateReactivePowerRawArray(void)** → Carica nell'array "ReactivePowerRaw_Array" l'ultimo valore letto della potenza reattiva;
- **void MCP39F511_UpdateApparentPowerRawArray(void)** → Carica nell'array "ApparentPowerRaw_Array" l'ultimo valore letto della potenza apparente.

L'esatta sequenza di utilizzo delle funzioni esposte, supponendo di lavorare sulla tensione RMS, è:

```
Cmd_39F511.MCP39F511_ReadRmsVoltageRaw();
Cmd_39F511.MCP39F511_UpdateRmsVoltageRawArray();
Cmd_39F511.MCP39F511_RmsVoltageRawAverage();
```

FUNZIONI DI LIBRERIA GPRS, TCP/IP E HTTP

Nella libreria GSM, di cui abbiamo parlato nei precedenti articoli, abbiamo aggiunto le funzioni necessarie alla gestione delle connessioni dati. Sono stati quindi creati tre nuovi file chiamati rispettivamente "GprsCmd_GSM.cpp", "TcpIpCmd_GSM.cpp", "HttpCmd_GSM.cpp" e relativi file .h. I moduli GSM SimCom sono supportati pienamente, il Quectel M95 parzialmente, mentre il Fibocom G510 non è supportato.

Quando è necessario usare i comandi AT per la gestione delle connessioni dati, si deve eseguire una sezione preliminare di inizializzazione del modulo GSM simile a quella già usata e ampiamente spiegata per le connessioni GSM (SMS e Voce). Quindi l'ultima revisione disponibile della nostra libreria può eseguire:

- l'inizializzazione base del modulo GSM per gestione SMS e voce;
- l'inizializzazione al servizio GPRS (General Packet Radio Service);
- l'inizializzazione connessione dati per applicazioni che usano il servizio HTTP con GET/POST.



Quindi la sequenza di inizializzazione aggiuntiva per le connessioni dati è la seguente:

- **AT+CGATT?** → inizia con un comando che verifica se il dispositivo è collegato o no al servizio GPRS e, in base al risultato ottenuto eseguirà oppure no il comando successivo;
- **AT+CGATT=1** → aggancia il modulo GSM al GPRS;
- **AT+CIPMUX=0** → abilita una connessione di tipo "Single IP Connection";
- **AT+CGEREP=1** → abilita l'invio da parte del GSM, tramite interfaccia seriale, degli "unsolicited result code" riguardanti la connessione al servizio GPRS;
- **AT+CSTT="OWN OPERATOR"** → imposta l'APN (Access Point Name) e se necessario fa inviare anche User Name e Password;
- **AT+CIICR** → attiva la connessione dati al servizio GPRS;
- **AT+CIFSR** → comando per visualizzare l'indirizzo IP locale assegnato dal servizio GPRS;
- **AT+CGDCONT=1,"IP","OWN OPERATOR"** → usato per definire il contesto PDP ("Packet Data Protocol"); il contesto PDP utilizzato è "IP";
- **AT+CGATC=1,1** → attiva il contesto PDP;
- **AT+CGPADDR=1** → comando usato per visualizzare l'indirizzo IP assegnato al contesto PDP che è diverso dall'indirizzo IP assegnato dal servizio GPRS;
- **AT+CIPHEAD=1** → usato per configurare il modulo GSM a ritornare in testa al pacchetto dati ricevuto l'indirizzo IP del mittente;
- **AT++CIPSRIP=1** → imposta il modulo GSM per ritornare, tramite "unsolicited result code", l'indirizzo IP remoto e relativa porta, quando si riceve un pacchetto dati;
- **AT+SAPBR=3,1,"CONTYPE","GPRS"** → configura il tipo di connessione dati per applicazioni con IP;
- **AT+SAPBR=3,1,"APN","OWN OPERATOR"** → usato per configurare APN proprio operatore telefonico per applicazioni basate su IP;
- **AT+SAPBR=1,1** → attiva la portante per applicazioni basate su IP;
- **AT+SAPBR=2,1** → richiesta indirizzo IP assegnato (l'IP restituito è uguale a quello assegnato dal servizio GPRS).

Per attivare l'inizializzazione della sezione GPRS e HTTP si devono settare a uno i seguenti flag:

```
Gprs.GprsFlag.Bit.EnableGprsFunctions = 1;
// Enable GPRS functions
Http.HttpFlag.Bit.EnableHttpFunctions = 1;
// Enable HTTP functions
```

Invece il codice di gestione nello sketch per assicurare la giusta sequenza di inizializzazione è:

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

Come si può notare, contiene tre sezioni indipendenti di inizializzazione: la prima è quella classica per applicazione voce e SMS, la seconda si occupa dell'inizializzazione del servizio GPRS e infine l'inizializzazione per applicazioni HTTP (usata per inviare i dati a ThingSpeak).

Di seguito, la sequenza completa di comandi AT inviati per l'inizializzazione della connessione dati:

```
AT+CGATT?<CR><LF>
<CR><LF>
+CGATT: 1<CR><LF>
<CR><LF>
OK<CR><LF>
AT+CIPMUX=0<CR><LF>
<CR><LF>
OK<CR><LF>
AT+CGEREP=1<CR><LF>
<CR><LF>
OK<CR><LF>
AT+CSTT="iliad"<CR><LF>
<CR><LF>
OK<CR><LF>
AT+CIICR<CR><LF>
<CR><LF>
OK<CR><LF>
AT+CIFSR<CR><LF>
<CR><LF>
10.145.242.241<CR><LF>
AT+CGDCONT=1,"IP","iliad"<CR><LF>
<CR><LF>
OK<CR><LF>
```

```

AT+CGACT=1,1<CR><LF>
<CR><LF>
+CGEV: ME PDN ACT 1<CR><LF>
<CR><LF>
OK<CR><LF>
AT+CGPADDR=1<CR><LF>
<CR><LF>
+CGPADDR: 1,"10.65.240.242"<CR><LF>
<CR><LF>
OK<CR><LF>
AT+CIPHEAD=1<CR><LF>
<CR><LF>
OK<CR><LF>
AT+CIPSRIP=1<CR><LF>
<CR><LF>
OK<CR><LF>
AT+SAPBR=3,1,"CONTYPE","GPRS"<CR><LF>
<CR><LF>
OK<CR><LF>
AT+SAPBR=3,1,"APN","iliad"<CR><LF>
<CR><LF>
OK<CR><LF>
AT+SAPBR=1,1<CR><LF>
<CR><LF>
OK<CR><LF>
AT+SAPBR=2,1<CR><LF>
<CR><LF>
+SAPBR: 1,1,"10.145.242.241"<CR><LF>
<CR><LF>
OK<CR><LF>

```

Oltre ai comandi AT appena descritti la libreria mette a disposizione una serie di altri comandi per la gestione della connessione dati tramite servizio HTTP (usati nel nostro sketch di esempio).

- **AT+HTTINIT** → usato per l'inizializzazione del servizio HTTP, deve essere eseguito per primo.
- **AT+HTTPSTATUS?** → legge lo stato del servizio HTTP; riceve come risposta quattro parametri: <mode>, <status>, <finish> e <remain>. Il primo parametro indica se è in corso un comando "GET", "POST" o "HEAD". Il secondo parametro indica lo stato in cui si trova il servizio ovvero "0" in attesa, "1" ricezione in corso o "2" trasmissione dati in corso. Il terzo parametro indica quanti dati sono stati trasmessi e l'ultimo parametro invece indica quanti dati devono essere ancora inviati o trasmessi.
- **AT+HTTTPARA** → usato per impostare i parametri del servizio HTTP. Il comando è diviso in due parti distinte, la prima parte imposta il parametro per il servizio HTTP (AT+HTTTPARA="CID",1) mentre la seconda parte imposta lo URL di

connessione con in coda i dati da inviare a ThingSpeak" (AT+HTTTPARA="URL","http://api.thingSpeak.com/update.json?.....").

- **AT+HTTPACTION=0** → usato per impostare se si deve eseguire un "GET" (0), un "POST" (1), un "HEAD" (2) o un "DELETE" (3).
- **AT+HTTPTERM** → termina il servizio HTTP e deve essere eseguito come ultimo comando.

Quanto segue è la sequenza dei comandi AT usati per l'invio dei dati alla piattaforma ThingSpeak tramite il servizio HTTP: in testa e in coda abbiamo rispettivamente il comando per l'inizializzazione del servizio HTTP e quello per la terminazione del servizio. Dopodiché si sfrutta il comando HTTPSTATUS per verificare che il servizio non sia occupato e se libero si inviano i dati alla piattaforma tramite il comando HTTTPARA (due comandi eseguiti in sequenza). Segue il comando HTTPACTION per l'esecuzione del comando "GET" verso l'URL selezionato (**Listato 3**).

Un'ultima considerazione per quanto riguarda la gestione della connessione dati: ci sono delle costanti stringa che devono essere salvate in EEPROM come già fatto per codici PIN, PUK e altro. Quindi ci sono delle direttive al compilatore dove sono definite queste costanti e lo spazio massimo che possono occupare in EEPROM. Tuttavia la scrittura in EEPROM di tali costanti stringa può solo avvenire tramite apposito sketch di cui abbiamo già avuto modo di parlare in altre occasioni. Le costanti stringa che devono essere salvate in EEPROM sono:

- APN operatore telefonico;
- User Name e Password per APN (non sono sempre utilizzati ma sono stati comunque definiti in caso di necessità);
- contesto PDP;
- indirizzo IP server remoto per connessione dati TCP/IP; sono consentiti fino a un massimo di tre indirizzi IP nel caso si facesse uso di connessioni multiple (AT+CIPMUX=1);
- dominio server remoto per connessione dati TCP/IP (sono consentiti fino a un massimo di tre domini in caso di connessioni multiple);
- porta di connessione al server remoto (sono consentite fino a un massimo di tre porte in caso di connessioni multiple);
- URL per il servizio HTTP;
- IP Proxy;
- porta Proxy.

Gli indirizzi di partenza in EEPROM per ognuna di



↓ Listato 3

```
AT+HTTPINIT<CR><LF>
<CR><LF>
OK<CR><LF>
AT+HTTPSTATUS?<CR><LF>
<CR><LF>
+HTTPSTATUS: GET,0,0,0<CR><LF>
<CR><LF>
OK<CR><LF>
AT+HTTTPARA="CID",1<CR><LF>
<CR><LF>
OK<CR><LF>
AT+HTTTPARA="URL","http://api.thingspeak.com/update.json? api_key=89QC6FHH0AU9636Z&field1=230.44&field2=
00.000&field3=0.998&field4=-00000.00&field5=-00000.00&field6=00000.00"<CR><LF>
<CR><LF>
OK<CR><LF>
AT+HTTTPACTION=0<CR><LF>
<CR><LF>
OK<CR><LF>
<CR><LF>
+HTTTPACTION: 0,200,278<CR><LF>
AT+HTTPSTATUS?<CR><LF>
<CR><LF>
+HTTPSTATUS: GET,0,0,0<CR><LF>
<CR><LF>
OK<CR><LF>
AT+HTTPTERM<CR><LF>
<CR><LF>
OK<CR><LF>
```

queste stringhe vengono recuperati tramite codice durante la fase di inizializzazione dello sketch e salvate nelle apposite strutture dati, affinché siano sempre accessibili.

LO SKETCH

Prima di iniziare la descrizione del firmware Arduino, spendiamo due parole sulla programmazione della memoria EEPROM con tutte le costanti stringa. La definizione della scheda Arduino utilizzata è contenuta nella libreria GSM, in particolare nel file "lo_GSM.h". Nel nostro caso stiamo usando Arduino Mega 2560 e quindi è attiva la direttiva:

```
#define ARDUINO_MEGA2560_REV3
```

mentre l'altra è ovviamente commentata. Sempre nel file "lo_GSM.h" sono state abilitate le direttive per la gestione delle macchine a stati per la gestione dei comandi AT per GPRS, TCP/IP e HTTP.

```
#define ENABLE_ANSWER_GPRS_AT_CMD_STATE
#define ENABLE_ANSWER_TCP_IP_AT_CMD_STATE
#define ENABLE_ANSWER_HTTP_AT_CMD_STATE
```

e relativi stati:

```
#define ANSWER_GPRS_AT_CMD_STATE 5
```

```
#define ANSWER_TCP_IP_AT_CMD_STATE 6
#define ANSWER_HTTP_AT_CMD_STATE 7
```

Nello sketch di scrittura in EEPROM devono essere definite le costanti stringa necessarie alla connessione dati in particolare, per la nostra applicazione, sono strettamente necessarie solo le seguenti:

```
const char FLASH_APN[] PROGMEM = "\iliad\";
const char FLASH_HTTP_URL[] PROGMEM =
"\http://api.thingspeak.com/update.json?api_
key=89QC6FHH0AU9636Z\";
```

La prima indica l'APN fornito dal proprio operatore telefonico per la connessione dati, quindi dovrete inserire lo APN fornito dal vostro operatore, mentre la seconda è l'URL necessario per la connessione dati alla piattaforma "ThingSpeak" con riferimento al proprio canale. Se utilizzati, dovrete configurare anche la stringa del codice PIN e PUK. Tutto il resto può essere lasciato invariato. Fatta questa premessa, si dovrà provvedere a programmare Arduino Mega con il codice presente nello sketch.

Analizziamo il nostro sketch, il quale è suddiviso in cinque file distinti.

- **GSM_EnergyMeter_Thingspeak.ino** → File principale con tutte le dichiarazioni di variabili, dichiara-



zione degli I/O potenzialmente utilizzabili dallo sketch, le direttive al compilatore delle costanti stringa usate dal monitor seriale, la definizione delle macchine a stati usate nonché le due funzioni "void setup()" e "void loop()".

- **DigitalInput.ino** → contiene la gestione degli I/O digitali di ingresso. Attualmente in questa versione dello sketch non sono utilizzati gli ingressi digitali dei due pulsanti P1 e P2 della board GSM_EnergyMete R.1.0 e tantomeno i due pulsanti della board GSM.
- **DigitalOutput.ino** → contiene la gestione degli I/O digitali di uscita. Gli I/O di uscita della board GSM_EnergyMete R.1.0 riguardano il buzzer, i due relé e i quattro LED. Usiamo solo i LED 1 e 4.
- **Measures.ino** → contiene il codice necessario alla gestione della macchina a stati che si occupa di eseguire la lettura dei dati sulle misure elettriche effettuate dall'integrato MCP39F511 e relativo codice per la stampa sul monitor seriale.
- **Timersint.ino** → contiene la gestione del vettore di interrupt per il TIMER 5. Base dei tempi 2 ms.

Analizziamo per primo il file GSM_EnergyMeter_Thingspeak.ino dove in testa troviamo la direttiva al compilatore che indica ogni quanto si vuole inviare i dati inerenti le misure elettriche alla piattaforma "ThingSpeak". La direttiva:

```
#define TIMEOUT_THINGSPEAK_SEND_DATA
```

accetta come valori 20, 30 o 60, a indicare che i dati devono essere inviati rispettivamente ogni 20, 30, 60 secondi. Questa direttiva va accompagnata con le direttive al compilatore presenti nella libreria di gestione dell'integrato MCP39F511 ovvero:

```
#define ENABLE_AVERAGE_CALC
#define AVERAGE_ARRAY_SIZE 4
```

La prima abilita il calcolo del valore medio sulle misure elettriche effettuate, mentre la seconda definisce la dimensione dell'array da usare per eseguire la media dei valori. I valori consentiti sono 4, 8, 16 o 32.

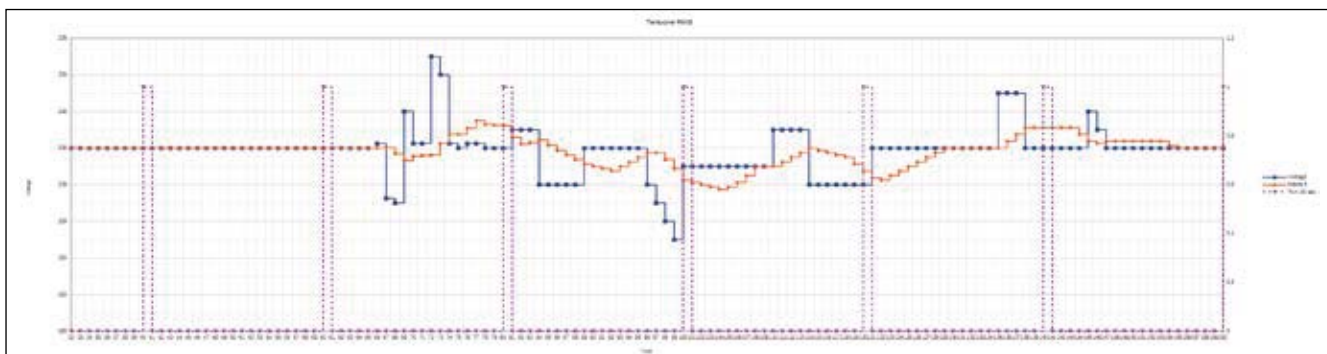
Dato che la lettura delle misure elettriche avviene ogni due secondi si consiglia di avere un buffer di lunghezza 8 per l'invio dei dati ogni 20 secondi, un buffer di lunghezza 16 per invio dei dati ogni 30 secondi e un buffer da 32 se si inviano i dati ogni 60 secondi. Non è obbligatorio calcolare il valore medio da inviare alla piattaforma; volendo inviare il valore istantaneo senza fare la media bisogna commentare la direttiva di cui sopra e la media dei valori letti verrà automaticamente disabilitata. La **Fig. 12** mostra un ipotetico scenario di misura dove la traccia in blu è l'andamento nel tempo della tensione di rete letta ogni 2 secondi (un campione ogni due secondi), il segnale rosso è il valore medio calcolato con buffer di lunghezza 8 e in viola abbiamo il momento in cui i dati vengono inviati a ThingSpeak (ogni 20 s). Ora si aprono due possibili scenari: il primo vede il calcolo del valore medio disabilitato, quindi ogni 20 secondi si invierà alla piattaforma l'ultimo valore istantaneo misurato. Così facendo si perdono tutti i campioni compresi tra un invio e il successivo.

Il secondo scenario prevede il calcolo del valore medio, quindi ogni 20 secondi viene inviato il valore medio calcolato su 8 campioni (un campione ogni due secondi).

Così facendo non c'è perdita di informazione, perché i campioni intermedi tra un invio e il successivo vengono sempre considerati.

Il concetto diventa più chiaro guardando la **Fig. 13**, dove i dati vengono inviati ogni 30 secondi e ogni 60 secondi rispettivamente. Nel grafico in alto il valore medio è calcolato usando un buffer di lunghezza 16, segnale in giallo, come si può osservare rispetto a quello di **Fig. 12**, il segnale è più lento

Fig. 12
Misura con tensione di rete letta ogni 2 secondi.



nell'inseguire il segnale originale campionato per via del buffer più lungo. Stesso discorso per il grafico in basso, dove il buffer è lungo 32; in questo caso è più evidente che se la media fosse disabilitata, l'invio del valore istantaneo letto ogni 60 secondi sarebbe povero di informazioni in quanto si perderebbe tutta la varietà campionata tra un invio e il successivo.

Il buffer aiuta a rimediare a questo problema, tenendo traccia dei campioni passati.

Il rovescio della medaglia è che più il buffer è lungo e più SRAM è necessaria per contenere i campioni necessari ad eseguire la media; inoltre richiede un tempo di calcolo più lungo. Notate che l'invio di un campione ogni 20 secondi significa che verranno inviati alla piattaforma 1.576.800 campioni all'anno, cioè poco più della metà di quelli disponibili con la versione free. Se invece si inviano i campioni ogni 30 secondi, in un anno saranno 1.051.200 campioni. Infine, con cadenza di 60 secondi la piattaforma riceverà 525.600 campioni all'anno.

Detto ciò procediamo con l'analisi dello sketch, in testa al file troviamo le costanti stringa usate sia durante la stampa sul monitor seriale delle misure elettriche, sia le stringhe che definiscono il campo "field" per i valori da inviare alla piattaforma.

Detto ciò analizziamo la funzione "void setup()" che inizializza lo sketch la quale deve occuparsi di:

- recuperare gli indirizzi di EEPROM dove sono memorizzate tutte le costanti di gestione della libreria GSM comprese le connessioni dati;
- configurare gli I/O di ingresso e uscita della

shield GSM;

- configurare gli I/O di ingresso e uscita della shield Energy Meter;
- configurare la gestione ISR Timer 1 libreria Energy Meter da libreria GSM;
- configurare l'UART per il monitor seriale;
- configurare l'UART per comunicazione col modulo GSM;
- abilitare l'interrupt libreria GSM;
- abilitare l'interrupt libreria Energy Meter;
- avviare l'inizializzazione del modulo GSM;
- stampare sul monitor seriale la revisione dello sketch.

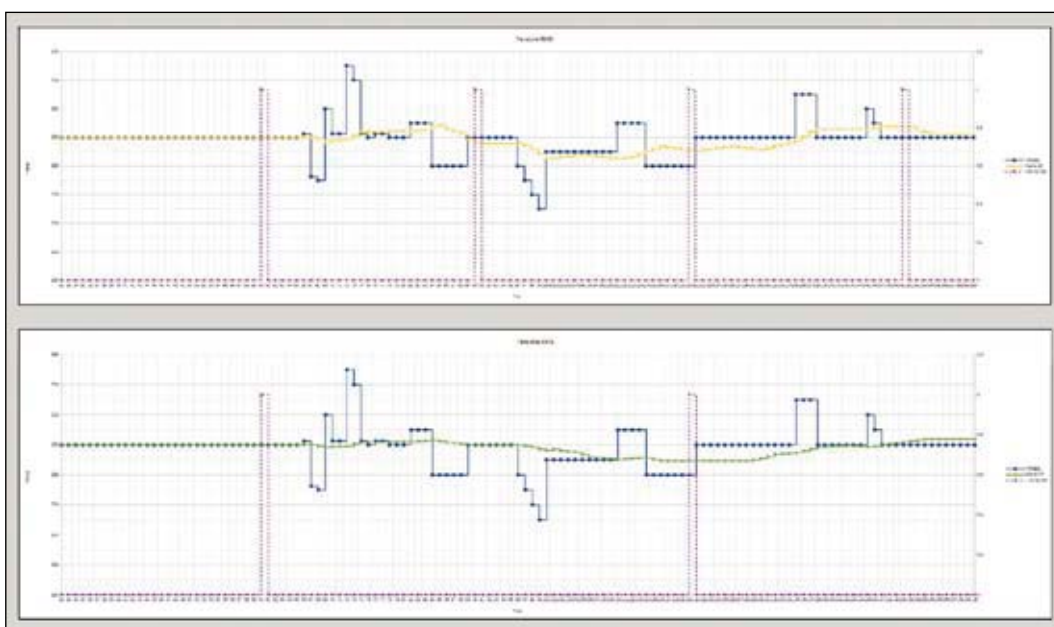
A questo punto l'inizializzazione è conclusa e si passa alla funzione "void loop()" dove troviamo:

```
ProcessStateMachineGsm();
```

```
ProcessHttpCmd();
```

La prima funzione contiene il codice necessario per la gestione del modulo GSM, dei tre step di inizializzazione e le funzioni di gestione delle macchine a stati a regime per l'invio dei comandi AT necessari al servizio HTTP verso la piattaforma ThingSpeak. A regime vengono anche gestite la macchina a stati per la lettura delle misure elettriche, la quale inizia la sequenza di lettura ogni due secondi, e la relativa stampa dei valori letti sul monitor seriale, in questo caso una volta ogni cinque secondi. Durante la lettura delle misure elettriche, se la funzione è abilitata, vengono caricati i rispettivi array per il calcolo del valore medio. Quando necessario,

→ **Fig. 13**
Grafici
con dati
campionati
ogni 30
secondi
(in alto) e 60
secondi
(in basso).



▼ Listato 4

```
void ProcessHttpCmd(void) {
    if ((Gsm.GsmFlag.Bit.GsmSendCmdInProgress == 0) && (Gsm.GsmFlag.Bit.GsmInitInProgress == 0) &&
        (Gsm.GsmFlag.Bit.GprsInitInProgress == 0) && (Gsm.GsmFlag.Bit.HttpInitInProgress == 0)) {
        if (Isr.TimeOutWait == 0) {
            Isr.TimeOutWait = T_1SEC;

            switch (LocalStateSendCmd)
            {
                case CMD_AT_HTTP_INIT:
                    Http.SetCmd_AT_HTTP_INIT();
                    LocalStateSendCmd = CMD_ATQ_HTTP_STATUS_1;
                    break;
                case CMD_ATQ_HTTP_STATUS_1:
                    Http.SetCmd_AT_HTTP_STATUS();
                    LocalStateSendCmd = CMD_AT_HTTP_PARA_STEP_1;
                    break;
                case CMD_AT_HTTP_PARA_STEP_1:
                    if (Http.HttpFlag.Bit.HttpStatus > 0) {
                        Http.SetCmd_AT_HTTP_STATUS();
                        break;
                    }
                    Io.LedOn(PIN_LED6);
                    Http.SetCmd_AT_HTTP_PARA(HTTP_PARAM_CID_CODE);
                    LocalStateSendCmd = CMD_AT_HTTP_PARA_STEP_2;
                    break;
                case CMD_AT_HTTP_PARA_STEP_2:
                    Http.HttpFlag.Bit.AddHttpParameters = 1;
                    SetDataToHttpThingSpeak();
                    Http.SetCmd_AT_HTTP_PARA(HTTP_PARAM_URL_CODE);
                    LocalStateSendCmd = CMD_AT_HTTP_ACTION;
                    break;
                case CMD_AT_HTTP_ACTION:
                    Http.SetCmd_AT_HTTP_ACTION(HTTP_ACTION_METHOD_GET);
                    LocalStateSendCmd = CMD_ATQ_HTTP_STATUS_2;
                    break;
                case CMD_ATQ_HTTP_STATUS_2:
                    TimeOutHttpAction = T_5SEC;
                    Http.SetCmd_AT_HTTP_STATUS();
                    LocalStateSendCmd = CMD_AT_HTTP_TERM;
                    break;
                case CMD_AT_HTTP_TERM:
                    if (Http.HttpFlag.Bit.HttpStatus > 0) {
                        Http.SetCmd_AT_HTTP_STATUS();
                        break;
                    }
                    if (Http.HttpFlag.Bit.HttpStatusCode != 200) {
                        if (TimeOutHttpAction > 0) {
                            break;
                        }
                    }
                    Http.SetCmd_AT_HTTP_TERM();
                    LocalStateSendCmd = CMD_AT_HTTP_INIT;
                #if TIMEOUT_THINGSPEAK_SEND_DATA == 20
                    Isr.TimeOutWait = T_12SEC;
                #elif TIMEOUT_THINGSPEAK_SEND_DATA == 30
                    Isr.TimeOutWait = T_22SEC;
                #elif TIMEOUT_THINGSPEAK_SEND_DATA == 60
                    Isr.TimeOutWait = T_52SEC;
                #else
                    Isr.TimeOutWait = T_22SEC;
                #endif
                Io.LedOff(PIN_LED6);
                break;
                case CMD_AT_NOTHING:
                    Io.LedOff(PIN_LED6);
                    break;
                default:
                    break;
            }
        }
    }
}
```

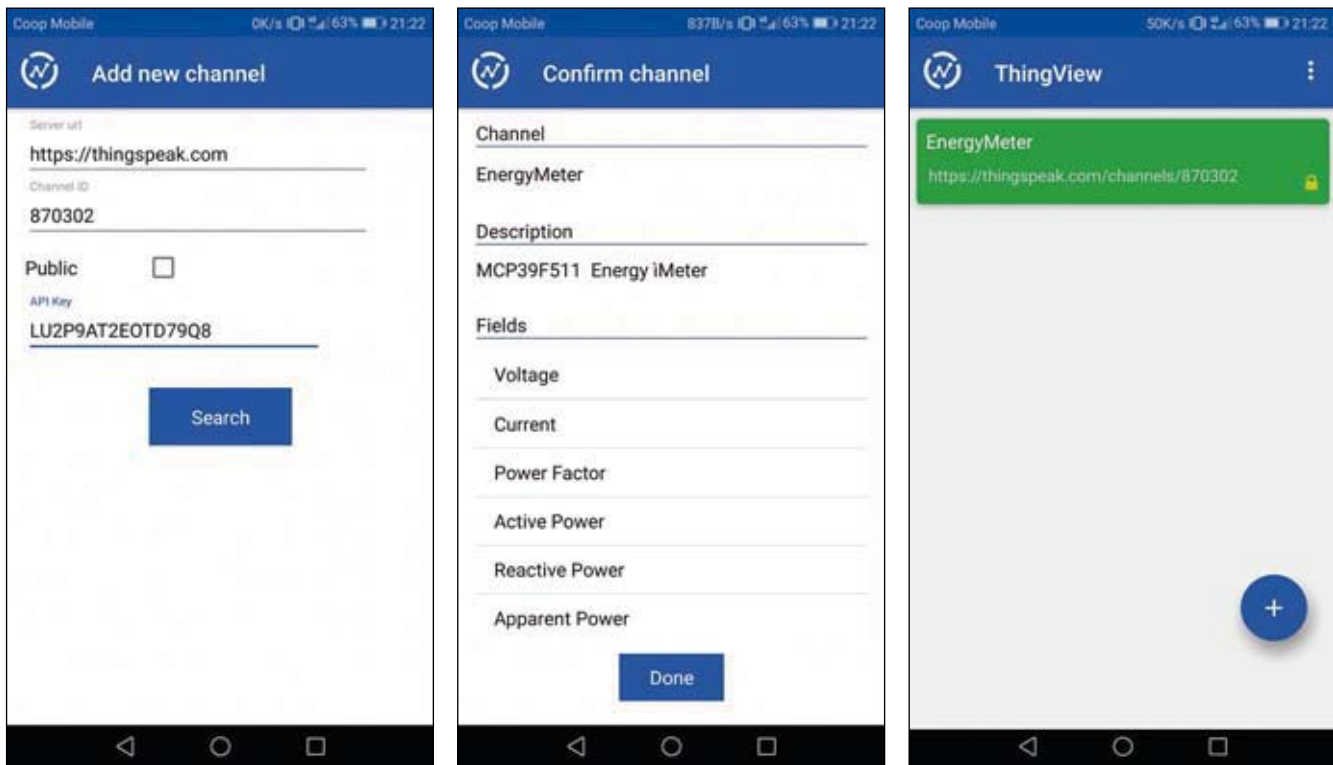


Fig. 14
Pagina di
configurazione.

Fig. 15
Pagina
rieepilogativa
e di conferma
del canale.

Fig. 16
Le misure
relative
al nostro
canale.

viene poi chiamata la funzione corrispondente che esegue il calcolo del valore medio per ognuna delle funzioni per cui è prevista. Ad esempio le funzioni necessarie a calcolare il valore medio vengono chiamate prima di stampare sul monitor seriale il valore medio letto oppure prima di inviare il dato a ThingSpeak. I valori contenuti negli array sono di tipo RAW, quindi devono essere convertiti al giusto valore da visualizzare a seconda di quanti decimali sono previsti. Ad esempio, per stampare il valore medio della tensione RMS si usa la seguente:

```
Serial.println((float)(Cmd_39F511.MCP39F511_RmsVoltageRawAverage())/ (pow(10, VOLTAGE_DEC)));
```

La costante VOLTAGE_DEC è pari a due, in quanto le misure di tensione RMS prevedono solo due decimali; come vedete il calcolo del valore medio è stato svolto durante la stampa su Serial Monitor. Parliamo ora della macchina a stati per la gestione dei comandi AT per l'invio dei dati alla piattaforma tramite il servizio HTTP. La funzione prevede una serie di step gestiti tramite un costrutto "switch - case". Ogni comando AT viene inviato con una cadenza di un secondo l'uno dall'altro. Quindi la funzione per prima cosa si assicura che non ci siano dei processi pendenti verificando lo stato di alcuni flag e se la variabile temporale "Isr.TimeoutWait"

è a zero procede con l'invio del comando AT corrispondente allo stato attuale.

La sequenza prevede l'inizializzazione del servizio HTTP (step 1) seguita da una richiesta di stato del servizio (step 2). Se il servizio è libero si inizia l'invio dei dati usando il comando HTTPPARA diviso in due step (step 3 e 4). Nello step 4 si inviano i dati alla URL selezionata (salvata in EEPROM) e si mettono in coda i dati da inviare alla piattaforma. Lo step successivo (step 5), attraverso il comando HTTPACTION, impone di eseguire un GET verso la URL selezionata. Allo step successivo (step 6) si invia nuovamente il comando per verificare lo stato del servizio. In caso positivo si verifica che il comando ACTION abbia finito con successo e solo dopo si dà inizio alla chiusura del servizio HTTP con apposito comando di terminazione. Nel **Listato 4** trovate il codice appena descritto.

La composizione del comando di invio dati alla URL selezionata sfrutta la funzione "SetDataToHttpThingSpeak()" per caricare apposito array della libreria GSM "Http.UriParameters". Impostando il flag "Http.HttpFlag.Bit.AddHttpParameters" a "1" si dice alla libreria che deve mettere in coda al comando AT la stringa presente nell'array "Http.UriParameters". Durante l'invio dei dati verso la piattaforma viene acceso il LED 6 (giallo) sulla shield GSM mentre il

LED 9 (verde) lampeggia a indicare che il codice sta girando regolarmente.
Invece sullo shield EnergyMeter viene acceso il LED 1 (rosso) durante la fase di lettura delle misure elettriche e il LED 4 (verde) durante la fase di stampa sul monitor seriale delle stesse.

CURIOSITÀ

Esistono APP per smartphone che permettono di visualizzare sul telefonino i grafici presenti su ThingSpeak per l'ID selezionato e relativa chiave API di lettura. Abbiamo provato, su piattaforma Android, "ThingView Free" (esiste anche la versione a pagamento che è più ricca di funzioni chiamata "ThingView Full" che costa circa 2€) e una volta installata ci permette di configurare il canale da visualizzare tra quelli disponibili sul proprio account. La **Fig. 14** mostra la sezione di configurazione dove si deve inserire l'ID del canale e togliere o lasciare la spunta sull'etichetta "Public" (se il canale è privato togliere la spunta).
La **Fig. 15** mostra il nostro canale con le misure elettriche corrispondenti: cliccando su "Done" si confermano i dati e si ottiene il riferimento al canale con le misure (**Fig. 16**); selezionando il canale compariranno i soli grafici impostati nel canale come mostrato nella **Fig. 17** e nella **Fig. 18**. Con la



Cosa occorre?

I componenti utilizzati in questo progetto sono facilmente reperibili. Il modulo Energy Meter (cod. FT1346M) è venduto a Euro 14,50, il trasformatore di corrente (cod. SBT002) costa Euro 14,00, quello di tensione (cod. SVL101201) Euro 9,50, la shield GSM (cod. WWGSM SHIELD) è disponibile a Euro 54,90, i moduli GSM da utilizzare con la shield sono raccolti tra i prodotti correlati della shield nel sito www.futurashop.it. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

versione a pagamento si ottengono funzioni aggiuntive e avanzate per la visualizzazione dei dati, che permettono analisi più approfondite (**Fig. 19**).

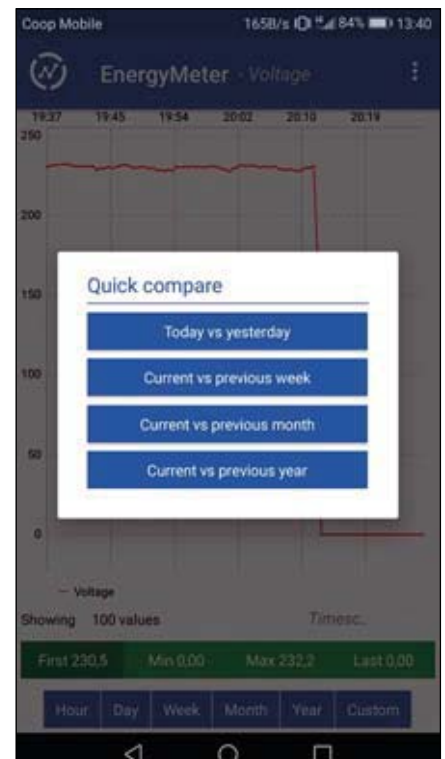
CONCLUSIONI

Con questo possiamo ritenere esaurita la trattazione sia dello sketch per l'invio dei dati alla piattaforma ThingSpeak che delle modifiche apportate alle librerie di gestione dei moduli GSM nonché dell'integrato MCP39F511.

Fig. 17
Grafici impostati nel canale: tensione e corrente.

Fig. 18
Altri grafici.

Fig. 19
Funzionalità avanzata.



STAMPANTE 3D4040



oggi ancora più ricca di accessori!

STAMPANTE 3D 40x40x40 cm in KIT

stampante 3D FDM (Fused deposition modeling)
capace di produrre stampe di **40x40x40 cm** (64.000 cm³)

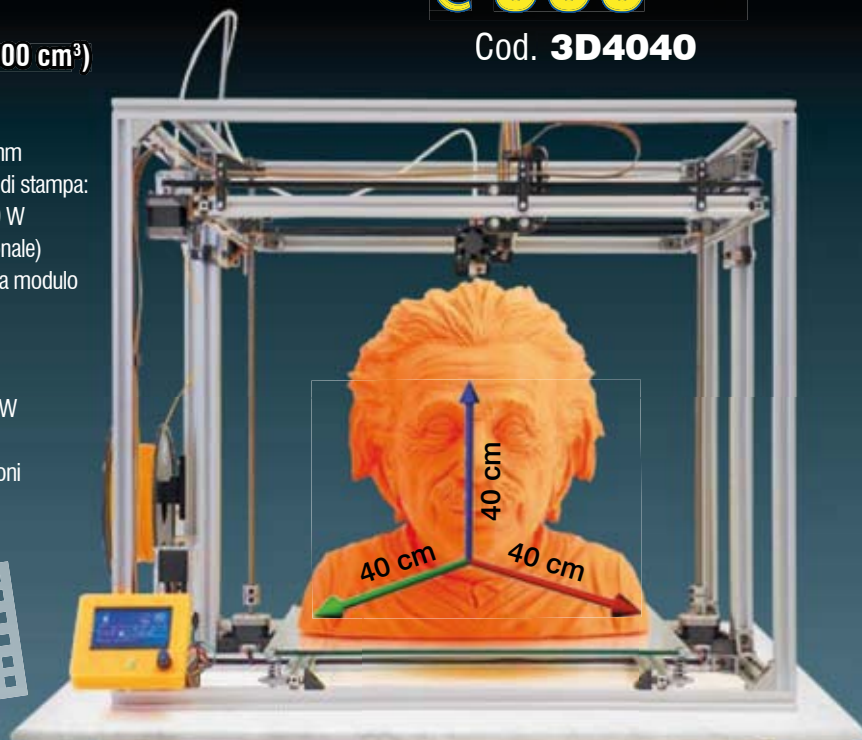
€ 999,00

Cod. **3D4040**

- ✓ Struttura interamente in alluminio
- ✓ Area di stampa: X 40 cm, Y 40 cm, Z 40 cm
- ✓ Diametro filamento: 1,75 mm
- ✓ Tipo di filamento: ABS, PLA, NYLON ed altri ancora
- ✓ Diametro ugello fornito: 0,4 mm
- ✓ Diametro ugelli opzionali: da 0,3 mm a 0,8 mm
- ✓ Velocità di stampa massima: 300 mm/s (in funzione dell'oggetto da stampare)
- ✓ Piatto di stampa fisso: vetro temperato da 6 mm
- ✓ Riscaldatore per piatto di stampa: 40 x 40 cm – 12V/240 W con adesivo 3M (opzionale)
- ✓ Controllabile da PC o da modulo LCD (opzionale)
- ✓ Alimentazione tramite modulo switching 220 VAC/12 VDC 350 W
- ✓ Istruzioni di montaggio in italiano con illustrazioni



Versione montata
cod. 3D4040/M
€ 1.299,00 IVA inclusa.



Box in plexiglass per
stampante 3D4040

VASTA GAMMA DI ACCESSORI!

SCOPRILI TUTTI SU
www.futurashop.it

Controller per stampa
autonoma con display grafico

NOVITÀ!

Cod. FT1147K
€ 34,00

Relè allo stato
solido per piatto riscaldato

Foglio riscaldato
in kapton 40x40cm

Cod. BOX3D4040
€ 198,00

Cod. FT1357K
€ 15,00

Cod. 3D4040HPLATE
€ 65,00

AI CON HARDWARE SPECIALIZZATO: JETSON NANO

dell'ING. DANIELE DENARO



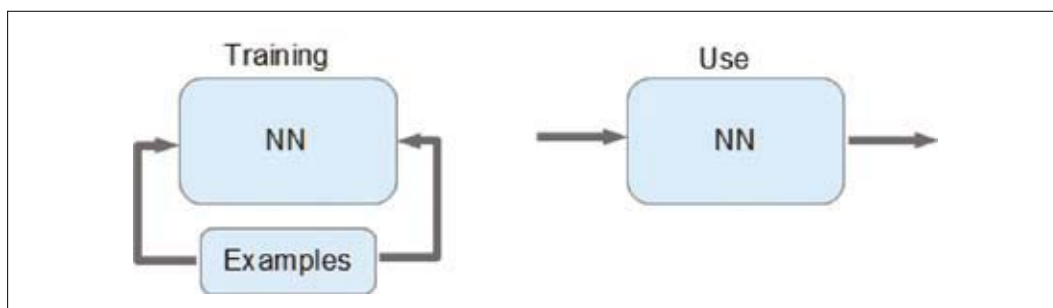
Sviluppiamo applicazioni di Machine Learning utilizzando una piccola ma potente piattaforma realizzata da Nvidia.

N

ei precedenti articoli abbiamo visto alcune caratteristiche basilari del *Machine Learning*, sostanzialmente basato sulle Reti Neurali (*Neural Networks* o NN) e soprattutto abbiamo visto come utilizzare questa tecnologia anche con hardware poco potente come Arduino.

Questi possibili utilizzi su Arduino erano limitati a riprodurre funzionalità input-output tra sensori e attuatori. In ogni caso queste funzionalità, per quanto semplici, possono avere importanti caratteristiche di multi-dimensionalità per l'input come per l'output. Inoltre, in genere, sono funzionalità non lineari. In queste condizioni, una realizzazione con programmi tradizionali basati sulla classica struttura a flusso logico e procedurale, diventa problematica, mentre l'utilizzo di un sistema di apprendimento automatico tramite esempi di coppie input-output corrette può essere molto più generalizzante e, tutto sommato, più semplice.

→ **Fig. 1**
Schematizzazione
del Machine
Learning con NN.



Per questa soluzione abbiamo messo a punto una libreria NNlib, la cui schematizzazione è mostrata **Fig. 1**, che potete trovare su github.com/open-electronics/Artificial_Intelligence. A questo punto ci si è chiesti come si potesse utilizzare l'approccio NN su un hardware più potente come Raspberry Pi. Naturalmente la libreria NNlib per Arduino può girare anche su Raspberry Pi, visto che è sviluppata in C++, ma è semplificata e permette di definire solo reti neurali a due strati completamente connessi. Le straordinarie performance delle applicazioni AI attuali sono invece dovute a strutture NN più complesse: strati convolutivi, strati ricorrenti, ed altre tipologie complesse. In totale alcune decine di strati successivi (*deep-learning*) e grande quantità di parametri (pesi delle connessioni), fino a decine di milioni. Data la complessità delle strutture NN utilizzate nel *deep-learning*, è necessario dotarsi di un ambiente di sviluppo adeguato. Per cui abbiamo preso in considerazione TensorFlow o meglio la sua evoluzione Keras. Questo ambiente trova il suo maggiore utilizzo con il linguaggio di programmazione Python. Nell'articolo pubblicato nel fascicolo n° 243, abbiamo introdotto le caratteristiche fondamentali di Keras, con la costruzione di una banale rete neurale di esempio. Ma poiché le reti profonde risultano molto pesanti per Raspberry Pi abbiamo preso in considerazione anche un particolare ambiente sviluppato in C chiamato Darknet. In questo modo abbiamo cercato di verificare le massime opportunità di Raspberry Pi con una struttura progettata per la massima ottimizzazione dell'hardware. In ogni caso il compito del "*object detection*" è risultato troppo gravoso per un Raspberry Pi 3, a meno di non utilizzare dell'hardware di aiuto come lo "Intel Neural Compute Stick 2" (vedi rivista numero 240 per la descrizione e l'uso della penna Intel). In realtà anche i normali PC, se non particolarmente attrezzati con schede ad alto parallelismo come le NVIDIA CUDA, soffrono le complesse elaborazioni delle reti deep-learning.

Ne consegue una certa difficoltà, per l'interessato sperimentatore, a realizzare applicazioni AI sofisticate su hardware limitato, portatile ed economico. Fortunatamente comincia ad essere disponibile un hardware di piccolo formato e basso costo ma di grande potenzialità di calcolo parallelo dedicato proprio alla elaborazione di reti neurali.

In questo articolo parleremo di un interessante piccolo computer delle dimensioni di Raspberry Pi ma con una potente sezione di 128 nodi CUDA: lo Nvidia Jetson Nano.

JETSON NANO: CARATTERISTICHE

Il Jetson Nano (**Fig. 2**) ha queste caratteristiche:

- SoM (System on Module) serie Nvidia Tegra X1 con Processore Quad-core ARM Cortex-A57 MPCore a 1,5 GHz con potente sottosistema hardware per l'alta definizione audio/video (gestione multi standard video e gestione delle immagini JPEG). Sono disponibili pin per la gestione di porte seriali di vario tipo, USB 2 e 3, HDMI, Display port e pin per 2 telecamere (Raspberry compatibili). La GPU è ad architettura NVIDIA Maxwell con 128 core Nvidia CUDA.
- Ram LPDDR4 4 GB 64-bit (saldato sul modulo).
- Unità eMMC 16 GB (saldato sul modulo per Boot ed altro).
- Socket per scheda di memoria microSD.
- Power Management Controller con sensore di temperatura e gestione della tensione e della frequenza del clock. Comprende anche un RTC.
- 256 pin di collegamento (con chiave di inserimento) per tutte le periferiche.

Per utilizzarlo, Nvidia mette a disposizione una scheda (Carrier Board) completa di connettori per periferiche e pettine di espansione compatibile con quello di Raspberry Pi. La carrier board realizza, insieme al modulo, un potentissimo microcomputer (**Fig. 3**) pronto per essere utilizzato

sia per applicazioni hobbistiche che professionali lasciando comunque al microscopico modulo la possibilità di essere inserito in strutture personalizzate a livello industriale, contenendo esso tutto il cuore del sistema. L'alimentazione (5V/2+4A) può avvenire tramite il connettore microUSB (massimo 2A) oppure tramite la presa Power Jack (che può ricevere anche 4/5A). L'alternativa è decisa dal ponticello J48 (**Fig. 4**) il quale se viene inserito disabilita il μ USB ed abilita il Power Jack. Il connettore doppio J38 riporta l'eventuale tensione presente sul connettore Ethernet nel caso sia presente l'opzione PoE (Power on Ethernet), ma il suo utilizzo è lasciato ad eventuali personalizzazioni che si devono far carico del controllo della tensione. Comunque l'alimentazione (5V) può essere fornita (o prelevata) anche direttamente sul connettore di espansione (pin 2 e 4). Il connettore J40 è dedicato all'implementazione dei bottoni (Power On e Reset), ma se l'ultima coppia 7-8 è lasciata aperta, il computer fa lo startup automaticamente appena data tensione (**Fig. 4**). Sul retro della carrier board sono presenti anche le piazzole per saldarci un contenitore

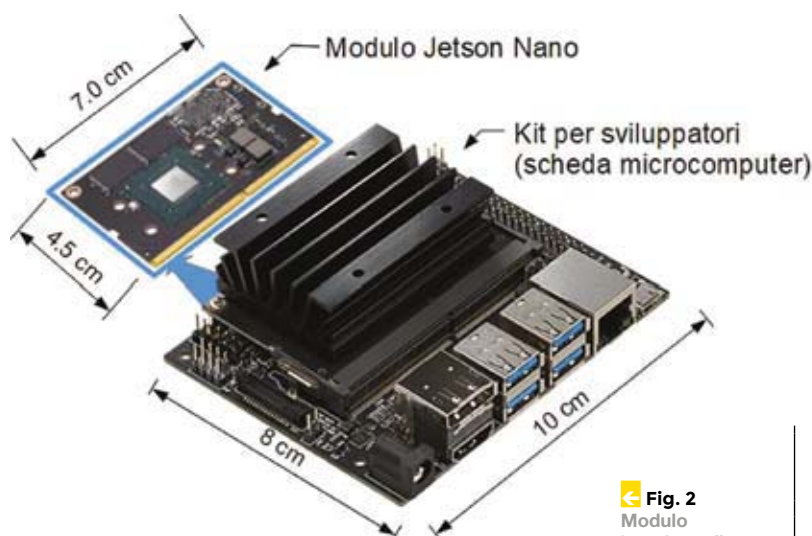


Fig. 2
Modulo inserito nella sua carrier board.

per eventuali batterie tampone per lo RTC. Non è presente un Wi-Fi, ma è predisposto per una scheda aggiuntiva agganciata al connettore J18 dedicato a questo scopo. Sulla scheda è presente un connettore compatibile con la telecamera di Raspberry Pi.

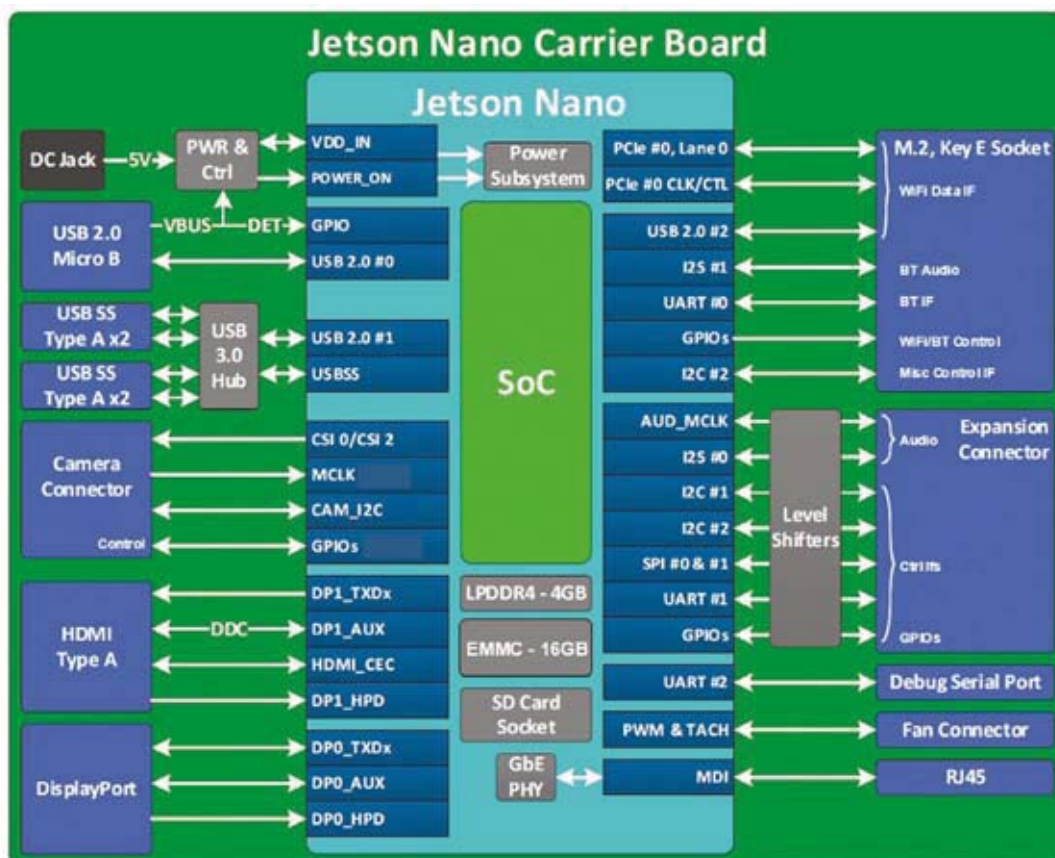


Fig. 3
Schema a blocchi del modulo sulla sua carrier board.

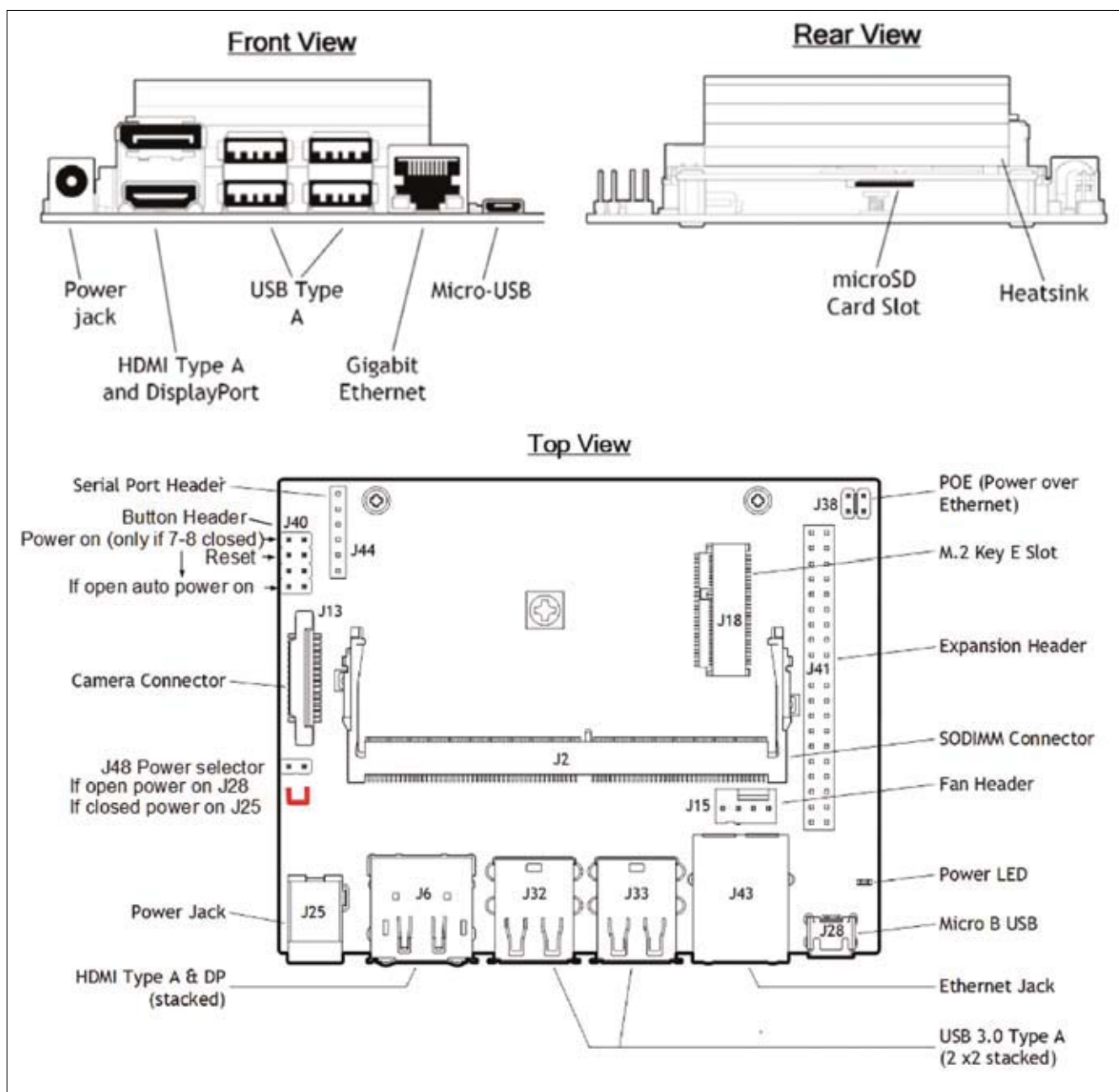


Fig. 4
Schema dei
connettori della
carrier board.

SOFTWARE E INIZIALIZZAZIONE

Come per Raspberry Pi, occorre utilizzare una scheda di memoria micro SD dove caricare il sistema operativo ed il resto del software necessario allo sviluppo. Il pacchetto complessivo è costituito da un file immagine (JetPack) da scaricare tramite il puntamento alla release più recente che potete trovare su <https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit>. La scheda ha una capienza consigliata di almeno 32GB. Va caricata con il file immagine del sistema

tramite i soliti programmi che sono utilizzati anche con Raspberry Pi ovvero per esempio Win32DiskImager oppure BalenaEtcher, che ha il pregio di utilizzare direttamente il file immagine in formato compresso. Una volta inizializzata la scheda la si inserisce in Jetson Nano. Quindi si collegano tastiera, mouse e schermo, e possibilmente la rete. A questo punto se si ha a disposizione un alimentatore con spina microUSB lo si collega, altrimenti, se si ha a disposizione un alimentatore con power-jack, bisogna prima pon-

ticellare il selettore J48. Con la configurazione originaria i ponticelli J40 sono aperti, quindi risulta abilitato lo startup automatico, per cui basta dare tensione per far partire la fase di inizializzazione che presenterà, come al solito le varie selezioni per lingua, tastiera ecc.

Completata l'inizializzazione, è possibile utilizzare un approccio remoto tramite un software di accesso SSH su PC, come per esempio MobaXterm. Ma oltre all'accesso in modalità terminale è possibile utilizzare anche un server di desktop remoto. Per esempio si può installare Xrdp così che si possa pilotare Jetson in modo grafico direttamente da Window. Per fare questo usate i seguenti comandi dopo esservi collegati in modalità terminale:

```
sudo apt-get update
sudo apt-get install xrdp
```

Ora potete utilizzare su Windows l'applicazione Desktop Remoto per collegarvi a Jetson Nano (e noterete che il desktop ha però una grafica leggermente diversa da quella sullo schermo locale). Il software installato dal JetPack è abbastanza complesso e prevede molte librerie e componenti dedicati a interfacciare sia il cuore CUDA, per la elaborazione delle reti neurali, che le applicazioni per la *computer vision* o la grafica del video o la telecamera. In sostanza, oltre al kernel Linux e al file system derivato dalla distribuzione Linux Ubuntu, sono presenti i seguenti componenti (Fig. 6):

- NVIDIA Jetson Driver Package (L4T);
- NVIDIA TensorRT: un motore di inferenza che può operare in modo ottimizzato sugli schemi

Pin in modalità NVIDIA										Pin in modalità Raspberry Pi									
3.3V	1	2	5.0V							3.3V	1	2	5V						
I2C1_SDA	3	4	5.0V							GPI02	3	4	5V						
I2C1_SCL	5	6	GND							GPI03	5	6	GND						
GPI009	7	8	UART1_TXD							GPI04	7	8	GPI014						
GND	9	10	UART1_RXD							GND	9	10	GPI015						
UART1_RTS*	11	12	I2S0_SCLK							GPI017	11	12	GPI018						
SPI1_SCK	13	14	GND							GPI027	13	14	GND						
GPI012	15	16	SPI1_CS1*							GPI022	15	16	GPI023						
3.3V	17	18	SPI1_CS0*							3.3V	17	18	GPI024						
SPI0_MOSI	19	20	GND							GPI010	19	20	GND						
SPI0_MISO	21	22	SPI1_MISO							GPI09	21	22	GPI025						
SPI0_SCK	23	24	SPI0_CS0*							GPI011	23	24	GPI08						
GND	25	26	SPI0_CS1*							GND	25	26	GPI07						
I2C0_SDA	27	28	I2C0_SCL							DNC	27	28	DNC						
GPI001	29	30	GND							GPI05	29	30	GND						
GPI011	31	32	GPI007							GPI06	31	32	GPI012						
GPI013	33	34	GND							GPI013	33	34	GND						
I2S0_FS	35	36	UART1_CTS*							GPI019	35	36	GPI016						
SPI1_MOSI	37	38	I2S0_DIN							GPI026	37	38	GPI020						
GND	39	40	I2S0_DOUT							GND	39	40	GPI021						

TensorFlow per massimizzare le performance;

- cuDNN: per implementare le NN su CUDA;
- CUDA toolkit per Jetson (e cross-compiler per PC);
- VisionWorks OpenCV: librerie per *computer vision* e *image* processing ottimizzate per Jetson;
- NVIDIA driver per lo schermo e librerie OpenGL;
- Multimedia API.

Una descrizione più dettagliata la potete trovare su <https://docs.nvidia.com/jetson/l4t/index.html>.

Fig. 5
Connettore di espansione J41.



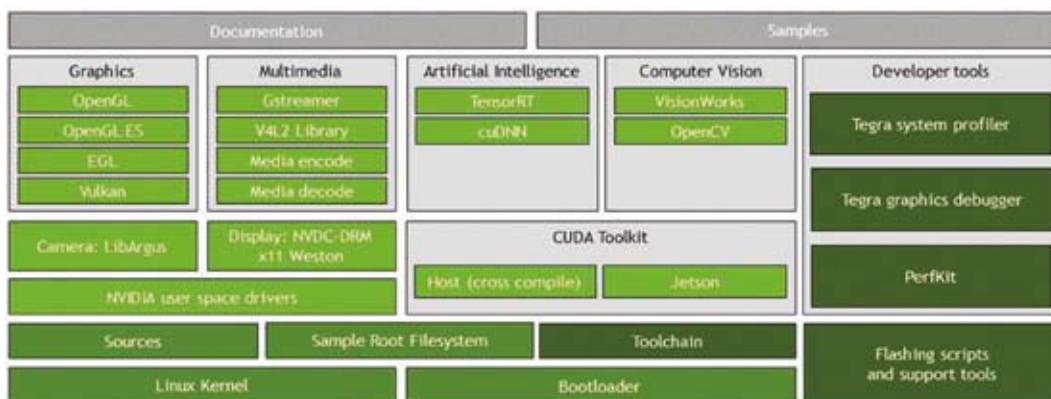
ORARIO: dal lunedì al sabato mattina 9:00 - 13:00
dal lunedì al venerdì pomeriggio 15:30 - 19:30
sabato pomeriggio chiuso

Informatica, Radiocomunicazioni, Action-Cam, Modem & Networking e molto altro!

Distributore autorizzato: **FUTURA ELETTRONICA**

Via Fiorentina 218/A - 52100 Arezzo • Tel: +39.0575984906 • assistenza@skytel.it

➔ **Fig. 6**
Struttura
del software.



SVILUPPIAMO CON JETSON NANO

La distribuzione Linux Ubuntu contiene le solite applicazioni di utilità come un editor grafico ("gedit"), un navigatore del file system ("nautilus"), ecc. Contiene anche la *suite* LibreOffice e il compilatore C++ e l'interprete Python, ma nessun ambiente di sviluppo IDE. Per visualizzare i package installati e quelli disponibili usate il programma "synaptic" equivalente al gestore "apt". Per installarlo utilizzate il solito comando:

```
sudo apt-get install synaptic
```

Una volta installato potete lanciarlo con il comando "sudo synaptic" così da scorrere tutti i package presenti e quelli disponibili ad una eventuale installazione, che può essere attivata direttamente dalla finestra grafica. Sicuramente è preferibile, per lo sviluppo, installare un ambiente IDE per Python.

Suggeriamo "PyCharm", ma sfortunatamente non è direttamente disponibile l'installazione per l'architettura ARM64. Si può ovviare installando Java e poi scaricando la distribuzione PyCharm per Linux. Infatti PyCharm gira su un Java Virtual

➔ **Listato 1**
Script Python di
esempio per l'uso
dei pin GPIO.

↓ Listato 1

```
import RPi.GPIO as GPIO
import time
# Pin Definitions
output_pin = 18 # BOARD pin 12, BCM pin 18
def main():
    # Pin Setup:
    # Board pin-numbering scheme
    GPIO.setmode(GPIO.BCM)
    # set pin as an output pin with optional initial state of HIGH
    GPIO.setup(output_pin, GPIO.OUT, initial=GPIO.HIGH)
    print("Starting demo now! Press CTRL+C to exit")
    curr_value = GPIO.HIGH
    try:
        while True:
            time.sleep(1)
            # Toggle the output every second
            print("Outputting {} to pin {}".format(curr_value, output_pin))
            GPIO.output(output_pin, curr_value)
            curr_value ^= GPIO.HIGH
    finally:
        GPIO.cleanup()
if __name__ == '__main__':
    main()
```

Environment. Quindi procedete come segue:

- usando synaptic o semplicemente "sudo apt-get install ..." installate "openjdk-11-jdk" (o la versione più recente disponibile al momento);
- scaricate l'archivio "pycharm-community-2019.3.3.tar.gz" dal sito: <https://www.jetbrains.com/pycharm/download/#section=linux>;
- scompattatelo e lanciate il file "./pycharm.sh" dalla sottodirectory "/bin" della distribuzione scompattata.

Se avete utilizzato PyCharm in altri ambienti, sapete già come operare, altrimenti guardate la documentazione sul sito PyCharm, ed in particolare la fase di configurazione con il link all'interprete Python presente su Jetson Nano. Se siete abituati all'ambiente Raspberry Pi potete installare anche un paio di software che riconoscerete e che su Ubuntu sono diversi: "pcmanfm" come gestione file e "leafpad" come editor.

PROGRAMMIAMO CON I GPIO

Prima di passare a sfruttare le grandi potenzialità dell'hardware di Jetson Nano facciamo un primo test utilizzando il connettore di espansione della scheda. Come detto, il connettore è compatibile con quello di Raspberry Pi e può essere configurato in vari modi. In ogni caso tutti i pin di I/O usano il livello 3,3V, ma la corrente in/out è molto limitata pari a circa $\pm 1\text{mA}$. Quindi bisogna utilizzare un buffer (transistor) per pilotare qualunque carico, anche un LED. Nella distribuzione installata su Jetson Nano è presente anche l'interfaccia GPIO per Python.

Proviamo ad utilizzarla con un esempio molto semplice. Altri esempi li potete trovare su: <https://github.com/NVIDIA/jetson-gpio>. Per prima cosa dobbiamo importare la libreria GPIO, con il comando **import Rpi.GPIO as GPIO**. Poi dobbiamo decidere quale schema per i pin intendiamo utilizzare,

per esempio con **GPIO.setmode(GPIO.BCM)**.

Infatti i pin del connettore di espansione possono essere identificati attraverso diversi schemi:

- schema BCM; praticamente la stessa denominazione GPIO di Raspberry Pi (vedi **Fig. 5**);
- schema BOARD: la numerazione dei pin sul connettore J41;
- schema CVM: nomi dei segnali sul pettine del modulo Jetson Nano invece di numeri;
- schema TEGRA_SOC : nomi dei segnali sul chip TEGRA.

A questo punto l'utilizzo è simile a quello con Raspberry Pi. Nel **Listato 1** potete vedere lo script Python che attiva e disattiva il pin 18 (in formato BCM o 12 in formato board), con intervallo di un secondo.

Per rilevarlo potete utilizzare semplicemente un tester, altrimenti potete usare un transistor come buffer per pilotare un LED.

Gli altri esempi che potete trovare sul sito sopra indicato, e che potete scaricare come "clone", coprono le varie possibilità di input/output ma solo digitale (non è prevista la modalità PWM).

AI CON JETSON NANO

Passiamo ora a scoprire le potenzialità di questo potente hardware nel campo del *Deep Learning*. Proviamo a caricare una struttura per il riconoscimento di oggetti. In realtà sono stati previsti da NVIDIA una serie di esempi con reti già addestrate (**Fig. 7**) potenti e conosciute per:

- riconoscimento di immagini (*Image Recognition*); per esempio riconoscere che una figura rappresenta una mela piuttosto che un'arancia;
- rilevazione di oggetti in un filmato (*Object Detection*); per esempio inquadrare e riconoscere esseri umani, auto, cani ecc. ripresi da una telecamera;
- suddivisione di un'immagine in parti ricono-

Fig. 7
Categorie delle reti pre-addestrate fornite come esempio.



Image Recognition
Classification



Object Detection
Localization



Segmentation
Free Space

↓ Listato 2

```
import jetson.inference
import jetson.utils
import argparse
import sys

# parse the command line
parser = argparse.ArgumentParser(description="Locate objects in a live camera stream using an object detection
DNN.",
                                formatter_class=argparse.RawTextHelpFormatter, epilog=jetson.inference.
detectNet.Usage())
parser.add_argument("--network", type=str, default="ssd-mobilenet-v2", help="pre-trained model to load (see below
for options)")
parser.add_argument("--overlay", type=str, default="box,labels,conf", help="detection overlay flags
(e.g. --overlay=box,labels,conf)\ninvalid combinations are: 'box', 'labels', 'conf', 'none'")
parser.add_argument("--threshold", type=float, default=0.5, help="minimum detection threshold to use")
#parser.add_argument("--camera", type=str, default="0", help="index of the MIPI CSI camera to use
(e.g. CSI camera 0)\nor for VL42 cameras, the /dev/video device to use.\nby default,
MIPI CSI camera 0 will be used.")
parser.add_argument("--camera", type=str, default="/dev/video0", help="index of the MIPI CSI camera to use
(e.g. CSI camera 0)\nor for VL42 cameras, the /dev/video device to use.\nby default,
MIPI CSI camera 0 will be used.")
parser.add_argument("--width", type=int, default=1280, help="desired width of camera stream (default is 1280
pixels)")
parser.add_argument("--height", type=int, default=720, help="desired height of camera stream (default is 720
pixels)")
try:
    opt = parser.parse_known_args()[0]
except:
    print("")
    parser.print_help()
    sys.exit(0)

# load the object detection network
net = jetson.inference.detectNet(opt.network, sys.argv, opt.threshold)

# create the camera and display
camera = jetson.utils.gstCamera(opt.width, opt.height, opt.camera)
display = jetson.utils.gldisplay()
# process frames until user exits
while display.IsOpen():
    # capture the image
    img, width, height = camera.CaptureRGBA()
    # detect objects in the image (with overlay)
    detections = net.Detect(img, width, height, opt.overlay)
    # print the detections
    print("detected {:d} objects in image".format(len(detections)))
    for detection in detections:
        print(detection)

    # render the image
    display.RenderOnce(img, width, height)

    # update the title bar
    display.SetTitle("{:s} | Network {:.0f} FPS".format(opt.network, net.GetNetworkFPS()))
    # print out performance info
    net.PrintProfilerTimes()
```

→ **Listato 2**
Programma di
test per Object
Detection.

sciute (*Segmentation*): per esempio delimitare le case dal cielo, in un filmato.

Per ognuno di questi settori applicativi sono disponibili alcune reti alternative per poter valutare

le differenze di prestazioni, in termini di accuratezza e rapidità, delle diverse implementazioni. Queste serie di esempi non sono contenuti nel JetPack (immagine completa del sistema) che abbiamo caricato sulla schedina di memoria SD.



Vanno quindi installati a parte. Tutto il *package* che li contiene va scaricato da *github*, e le istruzioni di come farlo e di come renderlo operativo li potete trovare in:

<https://github.com/dusty-nv/jetson-inference/blob/master/docs/building-repo-2.md#downloading-models>. Infatti, il complesso pacchetto, va non solo scaricato ma anche compilato. Ma, state tranquilli, è tutto abbastanza automatizzato. Alla fine del processo di attivazione viene lanciato lo *script* linux: **download-models.sh** che mostra un menu con la possibilità di selezionare una o più reti pre-addestrate da scaricare. Questo menu può essere lanciato anche in seguito, per scaricare di volta in volta esempi senza appesantire subito lo spazio sulla SD. Alla fine del processo sarà stata creata la directory "jetson-inference" nel vostro ambiente su Jetson.

Le reti caricate sono in formato TensorRT che è il formato di NVIDIA utilizzato per ottimizzare l'utilizzo della struttura di elaborazione parallela basata sui core CUDA. In seguito vedremo come passare da TensorFlow a TensorRT, ma intanto possiamo utilizzare queste strutture di reti deep-learning già in formato utile (contengono già i pesi ed il file che le contiene ha estensione ".uff"; oppure contengono le strutture in formato "caffe" ed il modo di scaricarle su CUDA).

Queste reti sono contenute nella sottodirectory "release" del repository <https://github.com/dusty-nv/jetson-inference/releases> e in "jetson-inference" su <https://github.com/dusty-nv/jetson-inference>. Il problema dell'eterogeneità dei formati delle reti deep-learning affligge il settore, e soprattutto si basa su un differente approccio tra la ricerca accademica che predilige formati più "strutturati" e l'utilizzo di linguaggi vicini alla sperimentazione come Python, mentre l'utilizzo ("deployment") delle reti, a livello industriale/commerciale, si preoccupa delle performance, dove il linguaggio C++ eccelle. Jetson Nano come espressione della volontà di portare performance utilizzabili nell'utilizzo del deep-learning anche sui mini-hardware portatili ed economici, non può sfuggire a questa contraddizione e lo sforzo software di NVIDIA si focalizza sul trasferimento dei formati delle reti in strutture ottimizzate per il suo hardware parallelo.

Alla fine della procedura **download-models.sh** viene chiesto se si vuole installare anche l'ambiente di programmazione "PyTorch". Questo ambiente, che vedremo in prossimi articoli, semplifica il "deployment" delle reti su CUDA ed è utile

se si vogliono utilizzare le reti pre-addestrate per modificare o migliorare l'addestramento con dei propri esempi.

Come detto proviamo innanzitutto con una rete adatta al riconoscimento di oggetti sullo "streaming" prodotto da una telecamera (*Object Detection*). In questo modo verifichiamo quanto più efficiente sia una struttura hardware dedicata alle reti neurali rispetto ad un hardware generale come Raspberry Pi. Nell'articolo precedente (rivista numero 243) abbiamo visto che anche facendo i cosiddetti "salti mortali" nell'ottimizzazione della rete e nella scelta di quella più efficiente (programma C Darknet che caricava la struttura di rete YOLOv3), si ottenevano circa 0,2 frame per secondo su Raspberry Pi 3; mentre si aveva un risultato migliore, ma ancora insufficiente, su un PC potente ma "generalista" (circa 3 frame per secondo). Per questo test possiamo utilizzare una veloce rete "SSD-MobileNet-v2" che è simile come funzionamento e performance alla rete YOLOv3. La rete "SSD-MobileNet-v2" è stata addestrata a riconoscere 91 tipi di oggetti, il cui elenco lo potete trovare nella sottodirectory di jetson-inference relativa all'esempio considerato. Tenete comunque presente che il programma è stato attivato con i parametri di base, a parte il link alla webcam USB, che sostituisce il link alla telecamera Raspicam. Con i parametri di base, la risoluzione della telecamera risulta di ben 1280x720, nettamente più impegnativa dei 416x416 della risoluzione della rete YOLO. Abbiamo utilizzato una webcam USB per aumentare la genericità del test. Il test ha fornito ottimi risultati in termini di velocità:

20 f/s (ovvero 50 ms per frame). Con questa velocità è possibile rilevare anche oggetti in movimento e lo streaming appare assolutamente fluido.

Nel **Listato 2** potete vedere il programma Python



Cosa occorre?

La board Jetson Nano completa di accessori (cod. JNANO) è venduta a Euro 229,00. Il kit comprende la scheda Jetson Nano, una Micro SD Card da 64GB, un adattatore di rete 5V/3A, un SD card reader e il manuale Nvidia. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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



che esegue il test. Il programma si chiama **"detectnet-camera.py"** e può essere configurato con diversi parametri. In particolare potete lanciare diversi tipi di reti dedicate al **"Object Detection"**. Se non viene indicata alcuna rete, la rete caricata per default è la "SSD-MobileNet-v2". Gli altri possibili nomi di rete sono elencati nel sito <https://github.com/dusty-nv/jetson-inference> (sotto la colonna CLI). Sia "detectnet-camera.py" che le altre procedure Python per le diverse tipologie di reti sono tutti versioni Python di programmi C++. E' quindi possibile utilizzare sia un ambiente C++ che Python. Le versioni Python non fanno altro che richiamare (**"binding"**) moduli C++. Tutti questi programmi fanno poi riferimento alla super-classe "tensorNet.cpp" che gestisce l'utilizzo di CUDA.

A questo punto dobbiamo rilevare che la gestione grafica (librerie GL) ha problemi a funzionare su desktop remoto "xrdp" o su terminali "X11".

Per effettuare il test bisogna quindi operare in locale collegando tastiera, mouse e monitor.

Anche il desktop remoto con VNC non funziona correttamente. Questi problemi sono riconosciuti nelle "talk" NVIDIA. Il fatto è che ci vorrebbe un sistema di streaming dal DISPLAY locale al desktop remoto. In questo modo si sgancerebbe la potente gestione della grafica realizzata da Jetson dalla velocità di comunicazione e di rendering remoto.

Comunque tali problemi esistono solo nell'utilizzo di librerie collegate alla gestione CUDA; le altre applicazioni grafiche funzionano sul desktop remoto o su terminale X11. Se non si vuole comutare tastiera, mouse e schermo quando si usa

il software per NN, si può adottare la modalità invertita: l'interazione principale su Jetson e il collegamento remoto sul PC tramite il software "Remmina" presente nell'installazione JetPack (è il software per desktop remoto rdp in ambiente Linux).

CONCLUSIONI

In questo articolo avete visto le potenzialità della piattaforma Jetson, che sono notevoli, forse troppe per un semplice appassionato e forse un po' complesse per essere sfruttate a fondo. Ma sicuramente, anche utilizzando solo una parte di queste potenzialità, è possibile studiare e/o applicare le moderne strutture deep-learning in modo efficace.

Cosa quasi impossibile con un hardware "generalista". Facendo correre la fantasia, possiamo immaginare applicazioni come il riconoscimento facciale in tempo reale di persone in movimento, rilevazioni di targhe di auto, rilevazione dei bordi di una strada ed eventuali ostacoli, rilevazione di particolari oggetti e loro posizione per l'azione di un braccio meccanico ecc. Insomma tutte applicazioni che hanno bisogno di azioni in tempo reale e che solo un hardware molto veloce può fornire. Ma anche studi più generali su architetture deep-learning possono giovare di questo hardware, per la grande velocità che si può ottenere nella fase di apprendimento. Ma naturalmente lo potete utilizzare anche semplicemente come un Raspberry Pi più potente.

Siamo lavorando a un'applicazione per disegnare una rete nostra e per attivarla su NVIDIA CUDA; vi terremo aggiornati. ■

MICROKIT



Componenti elettronici professionali, amatoriali e strumentazione

Distributore autorizzato:

**FUTURA
ELETTRONICA®**

Corso Torino 47/R - 16129 Genova • Tel: +39.010 5530071 • microkit@microkit.com



Hai comprato questa rivista e ti è piaciuta?

Abbonati oggi stesso e ti sconteremo il prezzo di questo numero sul tuo nuovo abbonamento!

Hai capito bene!

Per poter avere il prezzo scontato manda una mail a community@elettronica.in.it indicando i tuoi dati e se preferisci ricevere la rivista in formato digitale o cartaceo, riceverai un codice coupon del valore di 6 € per sottoscrivere il tuo nuovo abbonamento a Elettronica In

e ricorda che abbonandoti avrai :

- La **rivista puntuale** ogni mese direttamente a casa tua (il primo giorno del mese per l'abbonamento digitale)
- **10% di sconto** per i tuoi ordini su *futurashop.it*
- **Gratis 1 volume a scelta** della collana "L'elettronica per tutti" oppure **50% di sconto** per l'acquisto di un volume della libreria "Elettronica In" (solo con l'abbonamento cartaceo)
- La possibilità di leggere **Gratis** la versione digitale di Elettronica In (solo con l'abbonamento cartaceo)

PER ABBONARTI COLLEGATI AL SITO

www.elettronica.in.it

Futura
Group
Edizioni

FUTURA GROUP srl
Via Adige 11
21013 Gallarate (VA)
TEL.: 0331 752668

Precisione fino all'ultimo grammo!

Celle di carico ed elettronica necessari per la misura del peso di oggetti in moltissimi ambiti dalle bilance elettroniche alle pese industriali.



Cella di carico 50 kg

Composta da una struttura in metallo e da piccoli elementi chiamati estensimetri (strain gauges) montati in precisi punti sulla struttura. Fornisce in uscita un segnale elettrico che varia in funzione della deformazione che la forza, applicata sul componente, produce.

cod. LC50KG

€9,90



Scheda elettronica per cella di carico HX711

cod. HX711BOARD

€5,00

Scheda elettronica basata sul chip HX711, un convertitore analogico-digitale a 24 bit, specifico per interfacciare la cella di carico a un microcontrollore come Arduino, Fishino o compatibili. Dimensioni: 33,6x20,5x3 mm.



Cella di carico tipo "S" range 1 Tonnellata

Cella di carico di tipo "S" realizzata in acciaio inossidabile, classe di precisione C3, range 1 tonnellata e due fori filettati M12. Fornisce in uscita un segnale elettrico che varia in funzione della deformazione che la forza (trazione o compressione) applicata sul componente produce.

DISPONIBILE ANCHE IN DIFFERENTI RANGE:

cod. LCST1TONS

€44,00

500 Kg cod. LCST500KG **€44,00**
300 Kg cod. LCST300KG **€44,00**
200 Kg cod. LCST200KG **€44,00**
100 Kg cod. LCST100KG **€44,00**



Kit cella di carico con elettronica

Set composto da una scheda elettronica di interfaccia e da una cella di carico da 20 kg montata all'interno di una struttura meccanica. La scheda elettronica è basata sul chip HX711, un convertitore analogico-digitale a 24 bit, specifico per interfacciare la cella di carico a un microcontrollore come Arduino, Fishino o compatibili.

cod. LOADCELLKIT

€28,00

Set cella di carico 20 Kg con board HX711

cod. SET20KGLOADC

€11,00



Set composto da una scheda elettronica di interfaccia e da una cella di carico da 20 Kg. La scheda è basata sul chip HX711, un convertitore analogico-digitale a 24 bit, specifico per interfacciare la cella di carico a un microcontrollore come Arduino, Fishino o compatibili.

Cella di carico 10 kg

Composta da una struttura in metallo e da piccoli elementi chiamati estensimetri montati in precisi punti sulla struttura. Fornisce in uscita un segnale elettrico che varia in funzione della deformazione che la forza applicata sul componente produce.

DISPONIBILE ANCHE IN DIFFERENTI RANGE:

5 Kg cod. 5KGLOADCELL **€8,00**
1 Kg cod. 1KGLOADCELL **€8,00**



cod. 10KGLOADCELL

€8,00

Modulo sensore per pesatura



Set composto da una scheda elettronica di interfaccia e da una cella di carico da 5 kg montata all'interno di una struttura meccanica. La scheda elettronica è basata sul chip HX711, un convertitore analogico-digitale a 24 bit, specifico per interfacciare la cella di carico a un microcontrollore come Arduino, Fishino o compatibili.

cod. YB723

€14,50

Conoscere e usare

kivy

di PIER CALDERAN

Sperimentiamo con un'interfaccia per la visualizzazione di dati provenienti da sensori e attuatori. Quarta puntata.

Nelle puntate precedenti abbiamo visto come costruire interfacce utente per gestire l'output digitale usando i pulsanti virtuali dell'interfaccia, cioè, in pratica abbiamo acceso e spento alcuni LED collegati alla porta GPIO, toccando lo schermo. Oggi spiegheremo l'operazione inversa, ovvero come operare la gestione degli input digitali allo scopo di intercettare eventi esterni prodotti, per esempio, da sensori, pulsanti fisici o componenti che si comportano come tali. In pratica, impareremo a creare, nell'interfaccia grafica, delle aree per visualizzare i dati provenienti dall'esterno, ma anche un orologio con la data e l'ora, oltre ad altri

elementi; lo faremo attraverso lo sviluppo di un progetto pratico comprendente dell'hardware reale.

Per il primo progetto che realizzeremo in questa puntata del corso abbiamo pensato di utilizzare un sensore di temperatura e umidità: si tratta del diffusissimo DHT11, sicuramente noto agli sperimentatori e a chi pratica con le piattaforme di prototipazione (per esempio Arduino e Raspberry Pi).

IL SENSORE DHT11

Esistono in commercio diversi sensori in grado di rilevare la temperatura e l'umidità ambiente, ma il più comune fra tut-



 **Fig. 1** - Il sensore di temperatura e umidità DHT11.

ti è senza dubbio il DHT11 (Fig. 1). Questo sensore, pur non brillando per precisione e stabilità della misura, risulta molto economico e facile da usare, quindi è più che adatto per la sperimentazione. Per chi non conoscesse questo componente, di seguito ne riportiamo le caratteristiche principali.

- Tensione di alimentazione: 3 ÷ 5,5 Vcc.
- Segnale di uscita: digitale single bus (One Wire).
- Elemento sensibile: resistenza in polimero.
- Range di misurazione della temperatura: 0÷50 °C.
- Range di misurazione dell'umidità: 20-90% di umidità relativa (RH).
- Precisione (temperatura): +/-2 °C.
- Precisione (umidità): +/-4% RH.
- Tempo di rilevamento: 2s circa.
- Connettore: 4 pin.
- Dimensioni: 12x15,5x5,5 mm.

L'uscita dei dati del DHT11 è in formato digitale e utilizza un singolo canale per cui basta alimentarlo e collegare il suo unico pin di uscita a un ingresso digitale di Raspberry Pi. Lo schema di collegamento è illustrato in Fig. 2;

- il pin dati del sensore (pin 2) va collegato a un resistore *pull-up* da 10 K Ω ;
- il pin 3 non è collegato;
- ingresso GPIO4 di Raspberry Pi per la lettura dei dati.

Per usare il DHT11 è sempre necessario importare una libreria dedicata; per Python se ne possono trovare diverse, più o meno complesse, ma tutte molto facili da usare. Una di queste è disponibile a questo repository GitHub: https://github.com/szazo/DHT11_Python.

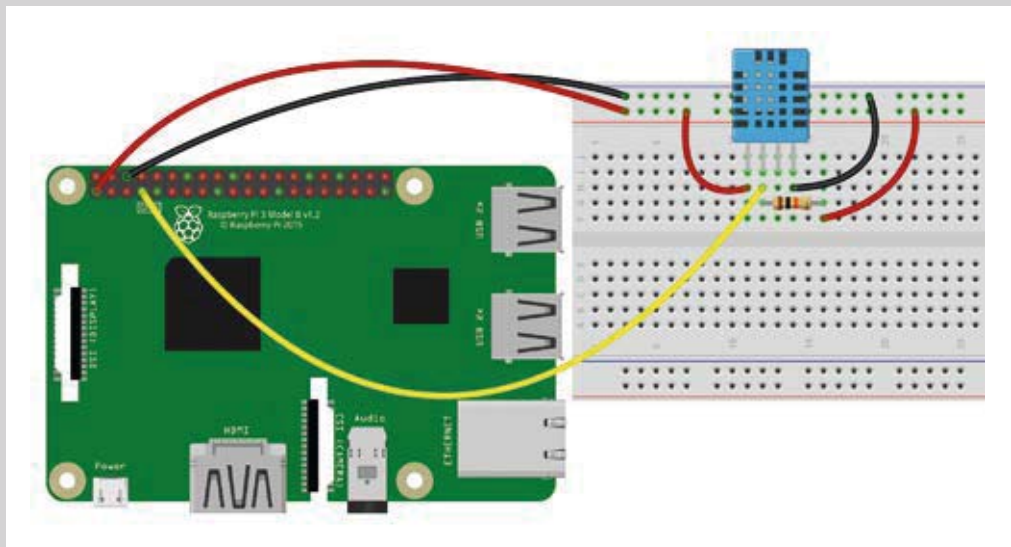
Una volta clonato l'archivio nella home di Raspbian, basta copiare il file *dht11.py* nella stessa cartella di lavoro degli script. In questo si può disporre della libreria importandola come modulo *dht11* in Python.

VISUALIZZAZIONE DELLA TEMPERATURA

Iniziamo a creare la nostra interfaccia partendo dalla parte più semplice. Impostiamo una singola *Label* per visualizzare la temperatura sul touchscreen.

Per fare questo basta importare il widget *Label* e, senza dover ricorrere al linguaggio KV, si può scrivere lo script direttamente in Python aggiungendo il widget al layout vuoto. Il **Listato 1**, oltre a mostrare il codice utilizzato per la creazione del widget *Label*, illustra anche come usare il timer di Kivy per visualizzare la temperatura a intervalli regolari di tempo. Ricordiamo che per poter utilizzare un timer con Kivy non è possibile usare la libreria *time* di Python, perché i processi di Kivy devono essere totalmente asincroni e necessitano di operare in autonomia.

A questo scopo, qui verrà utilizzata la proprietà *Clock.schedule_interval* del modulo *Clock*.



 **Fig. 2** - Schema di collegamento del sensore DHT11 a Raspberry Pi.

IL MODULO CLOCK DI KIVY

Una volta importato il modulo *kivy.clock* all'inizio dello script, è possibile creare direttamente l'oggetto *Clock*, ovvero un timer che interagisce autonomamente con le funzioni e le classi di Kivy.

L'oggetto *Clock* consente di pianificare una chiamata di funzione principalmente in due modi:

- **Clock.schedule_once** una volta
- **Clock.schedule_interval** a intervalli specifici

È possibile ottenere il tempo trascorso tra la programmazione temporale e la chiamata della funzione *callback* tramite l'argomento *dt*, ovvero l'argomento *delta-time*. Il *delta-time* non è altro che la differenza di tempo trascorso da un frame temporale e un altro.

Esempi:

```
def my_callback(dt): # dt sta per delta-time
    # codice da eseguire
    pass
```

Clock.schedule_interval(my_callback, 0.5) # chiamata alla funzione *my_callback* ogni 0.5 secondi (delta-time di 0.5 sec)

Clock.schedule_once(my_callback, 5) # chiamata alla funzione *my_callback* in 5 secondi (delta-time di 5 sec)

Clock.schedule_once(my_callback) # chiamata alla funzione *my_callback* appena possibile (il delta-time è il frame successivo).

Notare che se la funzione *callback* restituisce *False*, la chiamata verrà annullata e non verrà ripetuta.

Se si desidera pianificare una funzione da chiamare con argomenti predefiniti, è possibile utilizzare il modulo Python *functools.partial*:

```
from functools import partial
```

```
def my_callback(value, key, *larges): # i valori passati sono quelli delle variabili my_value e my_key
    pass
```

Clock.schedule_interval(partial(my_callback, 'my value', 'my key'), 0.5) # chiamata alla funzione *callback* passando i valori parziali di *my_value* e *my_key*

Per i dettagli vi rimandiamo al riquadro "Il modulo clock di kivy" che vedete in alto in questa pagina.

NOTA: come spesso evidenziato fin dalla prima puntata di questo corso, bisogna ricordarsi di aggiungere all'inizio di ogni script Python le righe riportate qui di seguito (a seconda del tipo di scheda usato, togliere il commento alla riga che interessa):

```
import os
#os.environ["KIVY_WINDOW"] = 'sdl2'
#uso con Raspberry Pi 4B
#os.environ['KIVY_GL_BACKEND'] = 'gl'
#uso con Raspberry Pi 3B+
```

Facciamo notare che nel **Listato 1** vengono importate le consuete librerie di Kivy ma anche *partial* e *RPi.GPIO*:

- **partial** permette di creare funzioni parziali che consentono di derivare una funzione con parametri *x* in una funzione con meno parametri e valori fissi impostati per la funzione;

- **RPi.GPIO** per la gestione della porta GPIO della Raspberry Pi;
- **DHT11** per lettura dei dati del sensore DHT11 (mettere il file *dht11.py* nella stessa cartella).

Come vedremo fra poco, è molto importante notare che abbiamo chiamato il callback che si trova all'interno di una classe. In questo modo, si potranno gestire più chiamate contemporaneamente. Per i dettagli, si raccomanda di leggere i commenti al codice.

Nel **Listato 1** viene creata una classe *DisplayData* derivata dal modulo *Label*, che visualizzerà i dati di temperatura provenienti dal DHT11 e che verranno aggiornati ogni 6 secondi con l'istruzione:

```
Clock.schedule_interval(partial(duration.update,
str(temperature)), 6)
```

Il posizionamento della *Label* è arbitrario e non dipende da nessun layout preimpostato, per cui verrà usata la proprietà



Fig. 3 - La schermata di visualizzazione della temperatura.

`self.size_hint=(1.0, 1.0)` per la posizionarla al centro dello schermo.
Grazie alla libreria DHT11 si potrà leggere dall'ingresso digi-

tale sia la temperatura che l'umidità con la funzione **result = temp.read()**.

Con la variabile **temperature = result.temperature** si leggerà solo il dato di temperatura.

Con la proprietà **text = "Temperatura: "+str(temperature)** si visualizzerà il testo nella Label con il dato di temperatura preceduto dalla parola "Temperatura: ".

Il risultato di tutto questo è visibile in **Fig. 3**.

VISUALIZZAZIONE DELLA TEMPERATURA E DELL'UMIDITÀ

Abbiamo voluto aggiungere anche una Label per poter visualizzare l'umidità insieme alla temperatura, ma con temporizzazioni diverse, ovvero con due funzioni callback.

Per fare questo, bisogna creare una seconda Label per il valore di umidità e cambiare i parametri di `size_hint` in modo da posizionare la seconda Label accanto a quello di temperatura.

Per brevità, il **Listato 2** mostra solamente il codice aggiuntivo. In pratica, la funzione di callback è identica alla

↓ Listato 1 - Codice di esempio temperatura_dht11.py

```
import os
#os.environ["KIVY_WINDOW"] = 'sdl2' #uso con Raspberry Pi 4B
#os.environ["KIVY_GL_BACKEND"] = 'gl' #uso con Raspberry Pi 3B+

#moduli importati
from kivy.app import App
from kivy.uix.widget import Widget
from kivy.uix.label import Label
from kivy.clock import Clock
from functools import partial
from kivy.uix.screenmanager import Screen
import RPi.GPIO as GPIO
import dht11

temp = dht11.DHT11(pin = 4) #crea l'istanza temp per la lettura dal pin GPIO4
temperature = 0 #variabile di temperatura

GPIO.setwarnings(True)
GPIO.setmode(GPIO.BCM)

class DisplayData(Label): #classe derivata da Label
    def update(self, index, *args): #funzione callback per l'aggiornamento dei dati del sensore
        result = temp.read() #lettura dei dati del sensore DHT11
        temperature = result.temperature #legge il dato di temperatura
        if result.is_valid(): # se il risultato è valido...

            print("Temperatura: %-3.1f C" % temperature) #stampa la temperatura sul terminale
            self.text = "Temperatura: "+str(temperature) #visualizza il testo nella Label
            self.font_size='30sp' #dimensione del font
            self.size_hint=(1.0, 1.0) #posizione al centro dello schermo

class inputGPIO(App): #la classe inputGPIO istanziata come App
    def build(self): #funzione build
        screen = Screen() #istanza Screen
        duration = DisplayData() #istanza per la classe Label

        screen.add_widget(duration) #aggiunta del widget allo Screen
        Clock.schedule_interval(partial(duration.update, str(temperature)), 6) #chiamata al callback ogni 6 secondi
        return screen #ritorno dello Screen con la Label aggiunta

inputGPIO().run() #esecuzione della classe principale inputGPIO
```





Fig. 4 - La schermata di visualizzazione della temperatura e dell'umidità.



Fig. 5 - La schermata di visualizzazione della temperatura, dell'umidità, della data e dell'orario.

precedente con la sola differenza che viene letto il dato di umidità. Il `Clock.schedule_interval` ha un diverso timing, per cui aggiornerà la lettura dell'umidità in modo indipendente. Per i dettagli, si raccomanda di leggere i commenti al codice. Come si può vedere in Fig. 4, le due Label vengono visualizzate a sinistra e a destra dello screen. Si fa notare che i parametri di `size_hint` sono diventati:

0.5 per la posizione x e **1.0** per la posizione y della Label temperatura **1.4** per la posizione x e **1.0** per la posizione y della Label umidità

Ricordiamo che `size_hint` imposta sempre le posizioni di x e y in percentuale da 0 a 1 e non in pixel.

TEMPERATURA, UMIDITÀ, DATA, GIORNO E ORA

Per terminare il progetto pratico in bellezza, riferiamoci al **Listato 3**, dove è mostrato il codice per costruire una terza Label che verrà posizionata sopra le altre due. Qui verranno mostrati la data, il giorno e l'orario aggiornato a intervalli di un secondo.

Importando il modulo `datetime` di Python è possibile leggere data e ora direttamente da un server NTP, per cui è necessario che Raspberry Pi sia connesso a Internet. Quello che vogliamo ottenere è un'interfaccia contenente un orologio preciso al secondo, che mostra altresì la temperatura rilevata ogni sei secondi e l'umidità misurata ogni quattro. Resta inteso che questi intervalli potranno essere modificati a piacimento.

Per brevità, il **Listato 3** mostra solamente il codice aggiunto per la gestione di data e ora. In pratica, la funzione di aggiornamento è identica alle precedenti con la sola differenza che viene letto e visualizzato l'orario da Internet ogni secondo. E' possibile cambiare il colore della font di una Label attraverso l'utilizzo del formato RGBA (*Red, Green, Blue, Alpha*). Nel nostro caso, i valori **100,0,0,1** attribuiscono un colore rossiccio alla data e all'ora.

Facciamo notare che anche il colore di fondo della finestra principale è stato modificato importando all'inizio il modulo `Window`.

Listato 2 - Codice di esempio temperatura_umidita_dht11.py

```
humidity=0 # variabile per l'umidità
class DisplayData2(Label): #classe per la visualizzazione della seconda Label

    def update(self, index, *args): #funzione di aggiornamento della seconda Label
        result = temp.read() #lettura del sensore DHT11
        humidity = result.temperature #legge solo il dato di umidità
        if result.is_valid(): # se il risultato è valido...

            print("Umidità: %-3.1f C" % humidity) #stampa la temperatura sul terminale
            self.text = "Umidità: "+str(humidity) #visualizza il testo della Label
            self.font_size='30sp' #dimensione del font
            self.size_hint=(1.4, 1.0) #posizione della seconda Label a destra dello schermo

#nella classe viene aggiunta l'istanza duration2 e un Clock che aggiorna la seconda Label ogni 4 secondi
duration2 = DisplayData2()
screen.add_widget(duration2)
Clock.schedule_interval(partial(duration2.update, str(humidity)), 4)
```

↓ Listato 3 - Codice di esempio temperatura_umidita_data_dht11.py

```
class DisplayData3(Label): #classe per la visualizzazione della terza Label
    def update(self, index, *args):
        date = datetime.datetime.now().strftime('%d-%m-%y %a %H:%M:%S') #formattazione della data e dell'orario per giorno,
        mese, anno, giorno della settimana, ore, minuti e secondi
        self.text = "Data/Ora: "+ date #il testo della Label con data e orario
        self.font_size='30sp' #dimensione del font
        self.size_hint=(1.0, 1.2) #posizione della terza Label sopra le altre due
        self.color=[100, 0, 0, 1] #colore del font
```

Con l'istruzione `Window.clearcolor` viene impostato il colore in background, sempre in formato RGBA, ma con i valori 0.5, 0.5, 0.5, 1 che creano un grigio chiaro. Ricordiamo che in questo caso bisogna usare i valori in percentuale da 0 a 1 e non valori assoluti da 0 a 255:

```
from kivy.core.window import Window #modulo Window
Window.clearcolor = (0.5, 0.5, 0.5, 1)
#Colore in background
```

Nella classe `inputGPIO(App)` viene aggiunta l'istanza `duration3` e un `Clock` che chiama il callback ogni secondo.

```
Duration3 = DisplayData3()
screen.add_widget(duration3)
Clock.schedule_interval(partial(duration3.update, str(date)), 1)
```

Il risultato è visibile in **Fig. 5**, dove la terza Label viene posizionata sopra le altre due Label a sinistra e a destra dello schermo. I parametri di `size_hint` per la terza Label sono 1.0 per x e 1.2 per y per posizionare la Label appena un po' sopra le altre due.

INPUT BUTTON

Per il secondo progetto hardware abbiamo pensato di as-

semblare tre pulsanti su una breadboard, collegati ad altrettanti ingressi digitali di Raspberry Pi. Lo scopo di questo semplice circuito è quello di simulare l'input proveniente da sensori esterni di tipo On/Off. Invece di pulsanti, si potranno usare, per esempio: moduli PIR, moduli Tilt, moduli Switch o altri moduli in grado di fornire un contatto aperto/chiuso. Osservando il circuito di **Fig. 6**, basta inserire a cavallo delle due sezioni della breadboard tre pulsantini a due vie (quelli che si trovano di solito in commercio e negli starter kit); collegare i terminali di una via a massa e collegare i terminali della seconda via rispettivamente ai pin GPIO6, GPIO19 e GPIO13 di Raspberry Pi.

Quello che si otterrà è l'attivazione o meno di tre pulsanti virtuali sullo schermo, in relazione alla pressione dei tre pulsanti esterni.

Osservando il **Listato 4**, per posizionare i pulsanti virtuali sullo schermo, abbiamo preferito usare un `GridLayout`, ovvero un layout a griglia formato da quattro celle verticali: le prime tre celle ospitano ciascuna un widget di tipo `Button`, mentre nella quarta cella abbiamo voluto caricare un'immagine dalla cartella di lavoro.

Per questo esempio, la cosa più importante da fare è creare tre classi `InputButton`. Come nel progetto precedente, anche in questo caso è necessario creare tre istanze per aggior-

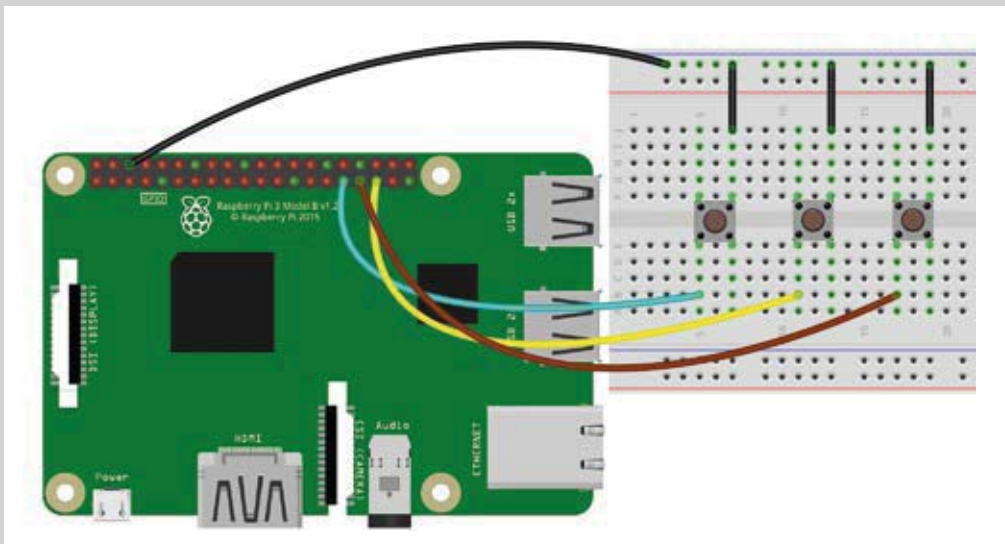


Fig. 6 - Circuito per il collegamento di tre pulsanti alla GPIO.

Listato 4 - Codice di esempio input_3_buttons.py import os

```

import os
#os.environ["KIVY_WINDOW"] = 'SDL2' #uso con Raspberry Pi 4B
#os.environ["KIVY_GL_BACKEND"] = 'gl' #uso con Raspberry Pi 3B+

from kivy.app import App
from kivy.uix.button import Button
from kivy.uix.gridlayout import GridLayout #modulo GridLayout per impostare la griglia
from kivy.uix.image import Image #modulo Image per importare un'immagine
from kivy.clock import Clock
import RPi.GPIO as GPIO

buttonPin1 = 6 #pulsante 1 GPIO6
buttonPin2 = 19 # pulsante 2 GPIO19
buttonPin3 = 13 # pulsante 3 GPIO13

GPIO.setmode(GPIO.BCM)
GPIO.setup(buttonPin1, GPIO.IN, pull_up_down=GPIO.PUD_UP) #modalità pull-up del pulsante 1
GPIO.setup(buttonPin2, GPIO.IN, pull_up_down=GPIO.PUD_UP) #modalità pull-up del pulsante 2
GPIO.setup(buttonPin3, GPIO.IN, pull_up_down=GPIO.PUD_UP) #modalità pull-up del pulsante 3

class InputButton1(Button): #classe per il pulsante 1 istanziata come Button
    def update(self, deltaTime): #funzione di callback chiamata dal Clock
        if GPIO.input(buttonPin1) == True: #se il pulsante 1 non viene premuto...
            self.state = 'normal' #lo stato è normal
        else: #altrimenti...
            self.state = 'down' #lo stato è down
            print (deltaTime) #stampa il delta-time sul terminale

#classe per il pulsante 2 identica alla precedente
class InputButton2(Button):
    def update(self, deltaTime):
        if GPIO.input(buttonPin2) == True:
            self.state = 'normal'
        else:
            self.state = 'down'
            print (deltaTime)

#classe per il pulsante 3 identica alla precedente
class InputButton3(Button):
    def update(self, deltaTime):
        if GPIO.input(buttonPin3) == True:
            self.state = 'normal'
        else:
            self.state = 'down'
            print (deltaTime)

class inputGPIO(App): #classe dell'applicazione principale
    def build(self):
        layout = GridLayout(cols=4, spacing=30, padding=30, row_default_height=150) # griglia con 4 colonne, spaziatura 30, distanza dai
        bordi 30 e altezza delle righe 150 pixel

        inputDisplay1 = InputButton1(text="BUTTON 1") #istanza basata sulla classe InputButton1
        inputDisplay2 = InputButton2(text="BUTTON 2") #istanza basata sulla classe InputButton2
        inputDisplay3 = InputButton3(text="BUTTON 3") #istanza basata sulla classe InputButton3

        Clock.schedule_interval(inputDisplay1.update, 1.0/10.0) #delta-time di 10 ms per il pulsante 1
        Clock.schedule_interval(inputDisplay2.update, 1.0/10.0) # delta-time di 10 ms per il pulsante 2
        Clock.schedule_interval(inputDisplay3.update, 1.0/10.0) # delta-time di 10 ms per il pulsante 3

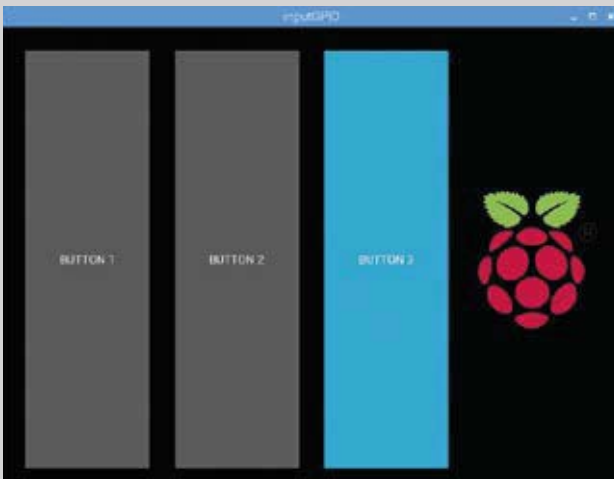
        wimg = Image(source='logo.png') #importazione di un'immagine

        layout.add_widget(wimg) #aggiunta del widget Image al GridLayout
        layout.add_widget(inputDisplay1) #aggiunta del widget Button 1 al GridLayout
        layout.add_widget(inputDisplay2) #aggiunta del widget Button 2 al GridLayout
        layout.add_widget(inputDisplay3) #aggiunta del widget Button 3 al GridLayout

        return layout #ritorno del GridLayout

inputGPIO().run() #esecuzione dell'app inputGPIO

```



↑ **Fig. 7** - La schermata con tre pulsanti (di cui uno premuto) e l'immagine del logo Raspberry Pi.

nare con `Clock.schedule_interval` lo stato del `Button` ad ogni pressione dei pulsanti fisici. Solo così si potrà intercettare la pressione del pulsante fisico e visualizzarne lo stato. Si fa notare che i tre ingressi della porta GPIO collegati ai pulsanti sono impostati in *pull-up*, in modo tale che premendo un pulsante si porta a massa il relativo ingresso. Da notare che, quando il pulsante non viene premuto, il suo stato (*state*) viene impostato come *normal*, mentre quando il pulsante viene premuto il suo stato viene impostato come *down*. Questi due stati permetteranno di cambiare il colore dello sfondo del pulsante sullo schermo, a seconda dello stato del pulsante fisico.

Detto questo, analizziamo il codice leggendo con attenzione anche i commenti.

Il risultato che si ottiene nel touchscreen è visibile in **Fig. 7**: come si può vedere, il `GridLayout` è impostato a 4 colonne (*cols*), che mostrano i primi tre pulsanti `Button` con spazia-

tura (*spacing*) 30 pixel, distanza dai bordi (*padding*) 30 pixel e altezza predefinita delle righe (*row_default_height*) 150 pixel. La quarta colonna mostra un'immagine (*logo.png*) caricata dalla cartella di lavoro tramite l'istruzione **`Image(source='logo.png')`**.

Dopo aver aggiunto tutti i widget al layout, si potrà cambiare lo stato *normal* e *down* dei tre pulsanti `Button`, premendo uno o più pulsantini.

INPUT BUTTON PERSONALIZZATO

Per il terzo progetto hardware si può usare lo stesso circuito visto sopra. Quello che andremo a fare è la personalizzazione dei pulsanti, ovvero il caricamento dinamico di un'immagine di fondo del pulsante nello suo stato normale e abbassato.

Per brevità riporteremo il codice solo per il pulsante collegato a GPIO6. Basterà replicare il codice per il numero di pulsanti utilizzati.

Per personalizzare lo sfondo del pulsante basta creare un'immagine png con un programma di fotoritocco (per esempio, Photoshop). Per il nostro pulsante abbiamo creato una forma circolare di colore rossiccio. La stessa immagine è stata replicata schiarendo lo stesso colore facendolo diventare bronzato (**Fig. 8**).

La risoluzione può essere di 72 DPI con la dimensione dell'immagine a piacere. Per il nostro esempio, le nostre immagini sono 800x800 pixel a 72 DPI.

Il nome dei due file da mettere nella stessa cartella sono **`button_up.png`** e **`button_down.png`**.

Vediamo ora come cambiare le due immagini in base allo stato del pulsante.

In questo esempio abbiamo usato ancora un `GridLayout`, ovvero un layout a griglia formato da quattro celle e quattro colonne. La parte più interessante del codice è relativa al comportamento del pulsante. Siccome abbiamo usato



↑ **Fig. 8** - Le due immagini usate per lo stato normale e premuto del pulsante.



→ **Fig. 9** - I due stati On e Off del pulsante.

Listato 5 - Codice di esempio input_image_button.py

```
buttonPin1 = 6 #variabile per il pin GPIO6
pinStatus = 0 #variabile per lo stato del pin
status = 0 #variabile per lo stato dell'immagine in background

def my_callback(dt): #funzione di callback una volta dopo 0.5 secondi.
    pinStatus = 0 #lo stato del pin è 0
    global status #variabile globale
    status = not status #con "not" si nega il valore dello stato

class InputButton1(Button):
    def update(self, deltaTime):
        global pin #variabile globale
        if GPIO.input(buttonPin1) == False & pinStatus==0: #se viene premuto il pulsante e lo stato del pin è 0...
            pinStatus=1 #allora lo stato del pin diventa 1

            if status == 0: #se lo stato dell'immagine è 0...
                self.state = 'down' #la proprietà "state" diventa down
                self.text="BUTTON ON" #il testo del pulsante diventa BUTTON ON
            if status == 1: #se lo stato dell'immagine è 1...
                self.state = 'normal' #la proprietà "state" diventa normal
                self.text="BUTTON OFF" #il testo del pulsante diventa BUTTON OFF

        Clock.schedule_once(my_callback,0.5) #intervallo di 0.5 secondi per il callback

class inputGPIO(App):
    def build(self):
        layout=GridLayout(cols=10,rows=10,spacing=20,padding=20) #layout a griglia

        #proprietà del Button
        inputDisplay1 = InputButton1(text="BUTTON OFF" #testo del Button
                                     ,font_size="20sp" #dimensione del font
                                     ,size=[256,256] #dimensione dell'immagine in background
                                     ,size_hint=[None,None] #posizione automatica nella prima cella
                                     ,background_normal = "button_up.png" # imagine "normal"
                                     ,background_down=" button_down.png ") # imagine "down"

        Clock.schedule_interval(inputDisplay1.update, 0.5) # intervallo di lettura del pulsante ogni 0.5 sec
        layout.add_widget(inputDisplay1) # aggiunta del widget Button
        return layout # ritorno del layout

inputGPIO().run() #esecuzione dell'app principale inputGPIO
```

gli stessi pulsantini di tipo normalmente aperto e non con funzionalità *toggle*, cioè On/Off, abbiamo creato una routine per fare in modo che con una prima pressione del pulsante si attivi la funzione On e con una seconda pressione si attivi la funzione Off.

Impostando il `Clock.schedule_interval` a 0,5 secondi abbiamo evitato anche il potenziale rimbalzo del pulsante. Si tenga presente che questo intervallo deve essere adattato a seconda del tipo di pulsantino.

Tutto il rimanente codice è simile al precedente con la differenza che abbiamo impostato le due immagini di stato con le due proprietà:

```
background_normal = "button_up.png"
background_down= "button_down.png"
```

All'avvio, il pulsante nella sua posizione normale avrà la prima immagine di sfondo **button_up.png** e nella sua posizione abbassata avrà la seconda immagine **button_down**.

png. Anche il testo del pulsante cambierà rispettivamente in **text="BUTTON OFF"** e **text="BUTTON ON"**.

La Fig. 9 illustra lo stato del pulsante On e Off con le immagini personalizzate.

Riportiamo nel **Listato 5** solo la parte di codice che interessa il comportamento *toggle* del pulsante.

Vi invitiamo a notare l'utilizzo delle variabili *pinStatus* per lo stato fisico del pin e *status* per lo stato dell'immagine in background.

Così facendo, si può utilizzare solo una funzione di callback ogni 0,5 secondi per ottenere dal pulsante il comportamento di tipo On/Off.

CONCLUSIONI

Bene, per il momento ci fermiamo qui. Con la speranza di essere stati sufficientemente chiari nell'esposizione dei concetti, vi diamo appuntamento alla prossima puntata, nella quale esporremo nuove proprietà e funzionalità di Kivy, con il supporto di ulteriori esempi applicativi.





Il Bell Boeing CMV-22B Osprey completa con successo il primo volo

Il primo CMV-22B Osprey, costruito da Boeing e Bell Textron ha completato le prime flight operations presso il Bell's Amarillo Assembly Center. Il CMV-22B è l'ultima variante della tiltrotor fleet che si unisce agli MV-22 e ai CV-22 utilizzati dagli U.S. Marine Corps e dalla U.S. Air Force.

La U.S. Navy utilizzerà il CMV-22B per sostituire il C-2A Greyhound per il trasporto di personale, posta, forniture e high-priority cargo dalle basi costiere alle portaerei in mare. Bell Boeing ha progettato la variante per la Marina specificatamente per le carrier fleet operations, prevedendo una maggior dimensioni dei serbatoi per una autonomia maggiore. La flessibilità di missione dell'Osprey aumenterà

Da Milano a Malpensa in 10 minuti con Hyperloop, parte lo studio di fattibilità

FNM e Hyperloop Italia avvieranno, per la prima volta in Italia, uno studio di ingegneria e di fattibilità tecnico-economico-giu-

ridica per valutare la possibilità di realizzare un sistema di trasporto di passeggeri e merci dalla stazione di Milano Cadorna alla stazione di Milano Malpensa che preveda l'utilizzo di una tecnologia di levitazione magnetica passiva. L'obiettivo è quello di studiare i vantaggi riguardo la diminuzione delle

tempistiche di trasporto ferroviario tradizionali che, secondo una prima ipotesi, potrebbero passare dagli attuali 43 minuti a 10 minuti. Saranno anche approfonditi i consumi energetici legati al nuovo sistema e i costi connessi all'intera operazione. I termini dell'intesa tra FNM e Hyperloop Italia sono contenuti in una lettera d'intenti che le parti sottoscriveranno. *"Le nuove forme di mobilità, soprattutto se improntate alla sostenibilità ambientale e all'impatto zero"* – commenta il presidente di FNM

HYPERLOOP IT

#HyperloopIT



le capacità operative e la prontezza, oltre a trasportare i principali componenti del motore dell'F-35.

"Con la capacità di percorrere fino a 1.150 miglia nautiche, il CMV-22B sarà un'ancora di salvezza per i nostri militari in mare", ha affermato Kristin Houston, vice presidente di Boeing Tiltrotor Programs e direttore del Bell Boeing V-22 Program. "La qualità e la sicurezza integrate in questo velivolo rivoluzioneranno il modo in cui la Marina degli Stati Uniti compie le sue missioni critiche di consegna carrier onboard". Bell Boeing consegnerà il primo CMV-22B all'Air Test and Evaluation Squadron (HX) 21 all'inizio del 2021 per i developmental test.

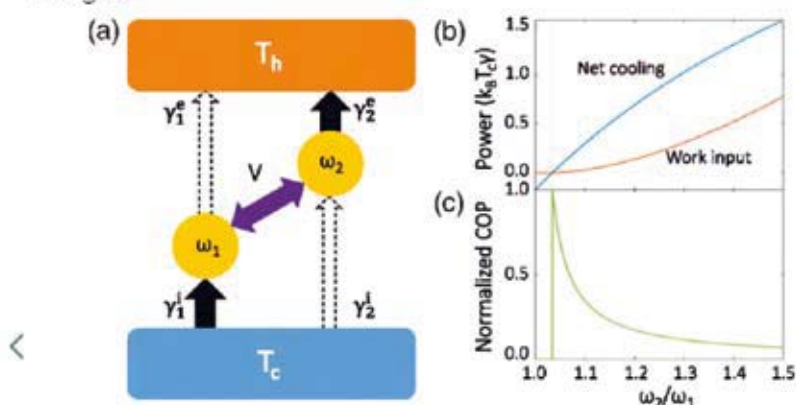
www.boeing.com

Andrea Gibelli - sono una frontiera che vogliamo esplorare e un'opportunità che vogliamo cogliere. In questa iniziativa, le competenze ingegneristiche e infrastrutturali che da sempre caratterizzano FNM possono allearsi con le tecnologie trasportistiche più innovative per creare un nuovo e avveniristico modello di mobilità".

www.fnmgroup.it



Images



Un frigorifero fotonico per abbassare la temperatura dei chip

Da sempre la generazione di calore da parte dei circuiti integrati, specie di quelli più performanti, ha rappresentato un ostacolo nella corsa alla miniaturizzazione. Per ovviare a questi problemi sono stati adottati numerosi accorgimenti, dalla riduzione della tensione di alimentazione, all'utilizzo di nuove geometrie, ma l'aumento dei transistor utilizzati (siamo arrivati a chip con miliardi di transistor) ha sempre riportato a galla il problema dell'eccessivo riscaldamento.

Un recente studio della Stanford University, in California, indica come combattere questo fenomeno con una tecnologia assolutamente innovativa.

Tutti gli oggetti irradiano calore mentre ricevono anche calore radiativo dall'ambiente. Un'analisi teorica suggerisce che un dispositivo potrebbe aumentare la radiazione in uscita abbastanza da raffreddare un oggetto in un ambiente caldo. Basta aumentare la frequenza dei fotoni irradiati in modo da superare la frequenza dei fotoni in arrivo dall'ambiente. Per ottenere questo effetto, i ricercatori californiani guidati da

Shanhui Fan hanno inventato una sorta di "frigorifero fotonico" realizzato con nanomateriali stratificati e installati su componenti elettronici per mantenerli freddi.

Per realizzare questa impresa, il frigorifero fotonico avrebbe un indice di rifrazione che oscilla nel tempo. Questa oscillazione, o "modulazione", potrebbe essere realizzata in diversi modi. In ogni caso il raffreddamento non è gratuito. Il team ha calcolato l'efficienza energetica del proprio sistema e ha scoperto che, nel caso ideale, il frigorifero fotonico poteva funzionare al limite superiore di efficienza fissato dalla termodinamica classica.

<https://news.stanford.edu/>



Maverick, l'aereo di linea del futuro secondo Airbus

Al recente Singapore Air Show, Airbus ha presentato il modello di un nuovo aereo commerciale in grado di ridurre sensibilmente il consumo di carburante grazie alla speciale forma delle ali e del corpo. Il modello dimostrativo esposto è lungo due metri circa.

Nel giugno dello scorso anno in Francia Airbus ha eseguito i primi test segreti del modello e prevede di continuare i test nella seconda parte dell'anno in corso. Il concetto di "ala blended" è stato sviluppato dal 1940 e ha portato alla creazione di aerei da attacco quali il B-2 e anche al progetto congiunto di Boeing e NASA X-48.

La combinazione delle ali con il corpo consentirà di ridurre la resistenza all'avanzamento e di conseguenza ridurrà il consumo di carburante e le emissioni di carbonio. Secondo le stime di Airbus, Maverick potrebbe ridurre il consumo di carburante del 20% rispetto ad altri velivoli simili esistenti.

www.airbus.com



Starlink a quota 300, un altro sogno di Elon Musk sta per diventare realtà

Prosegue senza sosta, con lanci da 60 satelliti alla volta, la realizzazione della costellazione Starlink che garantirà una connessione Internet, economica e ad alta velocità, in ogni angolo della Terra. SpaceX ha affermato che la sua rete Starlink comprenderà inizialmente 1.584 satelliti, tutti operanti in orbita terrestre bassa. Ma la società ha ottenuto l'approvazione della Federal Communications Commission per la gestione di altri 12.000 satelliti e prevede di chiedere l'autorizzazione per altri

30.000, per un totale di 42.000.

Elon Musk, CEO e fondatore di SpaceX, ha affermato che la compagnia avrà bisogno di almeno 400 satelliti in orbita per fornire una copertura minima e almeno 800 per fornire una copertura "moderata". Con i 300 ora in orbita, tale copertura potrebbe iniziare già nel 2020. Musk afferma che operando a un'altitudine più bassa, i satelliti Starlink possono ridurre la latenza, fornendo una copertura affidabile a un prezzo accessibile a più parti del mondo. SpaceX sta inoltre sviluppando terminali utente che consentiranno ai consumatori di collegarsi alla rete Starlink da casa, dal lavoro o anche in viaggio. Inoltre, la compagnia ha un contratto con le forze armate statunitensi.

www.spacex.com



La spesa la porta a casa Nuro R2

È il veicolo autonomo di nuova generazione che ha iniziato di recente un periodo di sperimentazione in un quartiere di Houston dopo aver ricevuto il via libera dal Dipartimento dei trasporti degli Stati Uniti (DOT) e dalla National Highway Traffic Safety Administration (NHTSA).

Questa decisione offre la certezza normativa a Nuro per l'utilizzo di un veicolo a guida autonoma di seconda generazione, costruito per trasportare pacchi anziché persone. R2, infatti, è stato pensato per la consegna, nell'ambito dell'ultimo miglio, di prodotti di consumo, generi alimentari e cibi caldi da negozi e ristoranti locali. Con le sue dimensioni, il peso, il front-end di protezione dei pedoni, la velocità operativa, la propulsione elettrica e le capacità di guida, R2 è pronto per iniziare il servizio come veicolo di quartiere di cui potersi fidare. Un ulteriore piccolo passo per incrementare la fiducia nei confronti dei veicoli a guida autonoma, che proprio taxi autonomi, veicoli per il trasporto di merci e shuttle dedicati dovrebbero contribuire ad alimentare. Nelle prossime settimane R2 inizierà i test su strada per preparare le sue prime consegne alle case dei clienti partner a Houston, in Texas. Questo veicolo potrà essere rapidamente messo in servizio anche in altre città che vorranno beneficiare di consegne on demand sicure, efficienti e convenienti.

www.nuro.ai

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.



Fiat 500 elettrica: buona la prima

Sicuramente chi la comprerà lo farà innanzitutto per il suo design, fedele alle radici e fortemente iconico, nel migliore stile italiano.

Poi, per quanto riguarda le caratteristiche, le prestazioni e il prezzo, non c'è che da sbizzarrirsi nel confronto con le decine di altre utilitarie completamente elettriche finalmente disponibili sul mercato.

La versione lanciata pochi giorni fa non è un adattamento del modello esistente (come invece lo era la versione americana) ma è una vettura tutta nuova realizzata su una piattaforma apposita, con dimensioni leggermente superiori.

La batteria è montata tra gli assi, sul pianale, ed ha una capacità di 42 kWh, sufficienti a garantire un'autonomia WLTP pari a 320 km. La ricarica può avvenire mediante una colonnina fast DC (80% della

carica in 35 minuti), oppure con il caricatore di bordo, in corrente alternata fino a 11 kW.

Il motore eroga 87 kW (118 CV) ed è in grado di garantire una velocità massima di 150 km/h, limitata elettronicamente, e un'accelerazione da 0 a 50 km/h in 3,1 secondi.

La nuova 500 prevede varie modalità di utilizzo, per recuperare maggiore energia durante il viaggio o limitare le prestazioni: il tutto al fine di aumentare l'autonomia che potrà così raggiungere i 400 km.

Il prezzo è stato fissato in 37.900 Euro, forse un po' troppo alto rispetto ad analoghi modelli della concorrenza. Naturalmente il prezzo finale può scendere in funzione degli incentivi statali, regionali o della stessa Casa.

www.fiat.it

Già
ammortizzato
dopo due anni
di attività
il più grande
sistema
al mondo
di accumulo
energetico

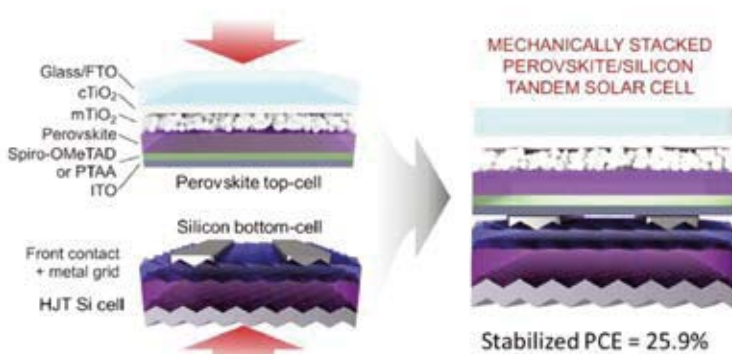


Era stato accolto con grande scetticismo il più grande sistema di accumulo di energia agli ioni di litio al mondo realizzato da Tesla in Australia: un impianto entrato in funzione alla fine del 2017 con una capacità di 100 MW/129MWh, costruito in 100 giorni e costato 90 milioni di dollari. L'Hornsedale Power Reserve (HPR), di proprietà e gestito dal produttore francese di energia rinnovabile Neoen, ha svolto un ruolo significativo nella stabilizzazione della rete sin dalla sua installazione nel 2017, una funzione precedentemente dominata dai generatori a combustibile fossile che facevano impennare i prezzi dell'energia durante i picchi di richiesta o i guasti.

Il sistema, che copre circa un ettaro nel parco eolico di Hornsdale, fornisce stabilità alla rete per i consumatori, immagazzinando energia in eccesso durante i periodi non di punta e fornendo energia quando le fonti rinnovabili non riescono a soddisfare la richiesta. Quando una centrale elettrica "cade" o quando c'è una maggiore domanda di energia, è praticamente una corsa per soddisfare quella domanda. Tesla HPR a batteria batte i fornitori di energia che dipendono dal combustibile fossile e offre ai consumatori uno scenario vantaggioso per tutti. I consumatori beneficiano dei prezzi più bassi di HPR e si evitano blackout che possono interrompere le attività commerciali o le attività domestiche quotidiane. HPR ha recentemente annunciato che amplierà l'impianto aggiungendo una capacità di ulteriori 50 MW / 64,5 MWh.

<https://hornsdalepowerreserve.com.au/>

Fotovoltaico: è made in Italy la tecnologia per celle solari "tandem" più efficienti



Un'innovativa cella solare "tandem" in perovskite e silicio con un'efficienza record superiore al 26% è stata messa a punto da un gruppo tutto italiano composto da ricercatori ENEA del Laboratorio di Tecnologie Fotovoltaiche, Università di Roma "Tor Vergata" (con il centro CHOSE), l'IIT - Istituto Italiano di Tecnologia (con Graphene Labs e il suo spin-off BeDimensional). I risultati di questa ricerca sono stati pubblicati sulla rivista internazionale Joule.

La cella sviluppata è composta da due celle solari accoppiate meccanicamente una sull'altra in modo da lavorare in tandem. La cella frontale, a base di perovskite, opportunamente dimensionata, converte bene la luce blu e verde dello spettro solare, lasciando passare la luce solare rossa ed infrarossa verso la

cella posteriore realizzata in silicio. Due elementi chiave nella realizzazione della cella tandem hanno permesso di ottenere alta efficienza: il grafene ha migliorato le prestazioni nella cella in perovskite, mentre l'eterogiunzione con film amorfi nella cella posteriore in silicio ha consentito di aumentarne la tensione. Finora è stata ottenuta l'efficienza record del 26,3%, ma l'obiettivo è di superare il 30%. Grazie alla tecnica messa a punto dai ricercatori italiani nella struttura tandem delle celle, è possibile conservare i vantaggi delle singole tecniche di fabbricazione, combinando la semplicità di realizzazione di film sottili in perovskite mediante "solution process" con la produzione di celle in silicio ad eterogiunzione.

www.enea.it



Equipmake annuncia il motore elettrico con la più alta densità di potenza al mondo

Grazie anche alla tecnologia di produzione 3D additiva, il nuovo motore denominato Ampere della società inglese Equipmake è in grado di erogare 220 kW (295 CV) a 30.000 giri/min con un peso di appena 10 Kg. Ciò significa che la densità di potenza è di ben 20 kW/Kg, quattro volte circa la densità di potenza di simili motori a magneti permanenti. Anche i motori elettrici più performanti, come quelli utilizzati dalle Tesla presentano valori di gran lunga inferiori. La Model S, ad esempio, utilizza un motore da 270 kW (362 CV) che pesa 32 kg; con una densità di potenza, quindi, di 8,4 kW/kg. L'Ampere, dunque, consentirebbe una riduzione di peso abbastanza significativa all'auto.



Per non parlare dei droni e dei velivoli elettrici che sempre di più si stanno diffondendo grazie al basso costo di esercizio. Equipmake afferma che la tecnologia di stampa 3D che sta utilizzando (in collaborazione con gli specialisti della società HiETA) le ha permesso di utilizzare meno metallo nel suo design, combinando componenti di forme complesse che non potevano essere raggiunte con la fresatura o la fusione, preservando allo stesso tempo l'eccezionale capacità di raffreddamento del magnete a raggi e la riduzione dell'inerzia che consentono una elevatissima velocità di rotazione.

<https://equipmake.co.uk/news/>

VIK, lo yacht a propulsione elettrica del designer russo Max Zhivov

Max Zhivov, il designer che ha al suo attivo il concetto di casa galleggiante Modul GO e lo yacht modificabile MYBO ispirato a LEGO, propone questa volta il design di un'imbarcazione ecologica, silenziosa, senza vibrazioni e senza gas di scarico, che può essere ricaricata da pannelli solari o energia eolica: la barca VIK. L'imbarcazione è stata progettata per navigare con grande efficienza energetica e poter trascorrere in mare un week-end con una sola ricarica delle batterie. L'imbarcazione dispone di una spaziosa cabina di prua con servizi igienici e doccia e c'è anche una zona notte sotto il tetto apribile che può essere utilizzata nei climi caldi e nei mesi più caldi. A poppa c'è un pozzetto aperto con sedili, ideale per socializzare, e una piattaforma da bagno, mentre l'albero è stato dotato di un dispositivo di sol-

levamento che può essere utilizzato per un piccolo tender. L'elica e il timone sono stati progettati con una protezione aggiuntiva da oggetti sommersi, forti alette laterali offrono maggiore stabilità e manovrabilità. Il motore elettrico dell'imbarcazione è ottimizzato per la velocità di crociera, con una potenza sufficiente per affrontare forti venti e onde. In caso di condizioni meteorologiche impreviste o cambi di rotta non programmati, è stato aggiunto un piccolo generatore diesel come fonte di energia di riserva. È inoltre possibile installare pannelli solari opzionali e un generatore eolico per ricaricare le batterie durante il viaggio e quando l'imbarcazione è ormeggiata.

www.vik-boats.com



FlixBus riduce le emissioni di CO₂ con i pannelli solari sul tetto del bus

FlixBus ha installato dei pannelli solari sul suo autobus che collega Dortmund, via Eindhoven, Anversa, Bruges e Calais a Londra. I pannelli solari generano energia per il consumo elettrico del bus. Ciò riduce efficacemente il consumo di carburante e le emissioni di CO₂. I tappetini solari installati sul tetto di FlixBus, ultra-sottili e leggeri per impedire la resistenza dell'aria, sono connessi all'alternatore del bus attraverso un regolatore di carica posizionato a bordo. La nuova tecnologia ricarica la batteria del bus principalmente ad energia solare, alleviando parzialmente l'alternatore dalla necessità di ricaricarla con il carburante. Durante il periodo di prova, sono stati risparmiati in media 1,7 litri di gasolio per 100 km. Con una distanza media di 600 chilometri al giorno, si tratta di un risparmio giornaliero di carburante di circa 10 litri.

La batteria, a sua volta, garantisce il consumo energetico di tutta l'elettronica di bordo. Come al solito con FlixBuses, il bus con pannelli solari è anche dotato di porte USB, prese, aria condizionata, Wi-Fi e un sistema di intrattenimento multimediale. Con la nuova tecnologia, tutte le apparecchiature di bordo sono ora alimentate da energia solare. In questo modo, i viaggiatori di FlixBus possono caricare i loro dispositivi mobili con energia rinnovabile, vale a dire con il sole che splende sul bus verde.

<https://global.flixbus.com/>



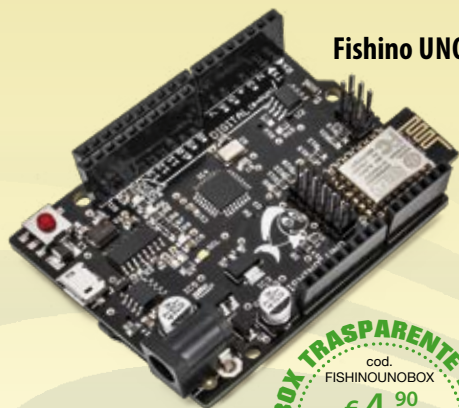
Fishino

Design With Simplicity

I TUOI PROGETTI DIVENTANO WIRELESS

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

Fishino UNO



cod. FISHINO UNO
€ 36,00



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

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



Fishino MEGA



cod. FISHINOMEGA
€ 49,90

Fishino32



cod. FISHINO32
€ 59,00



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

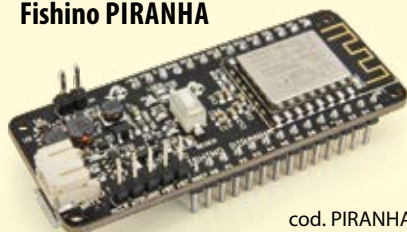
Fishino GUPPY



cod. GUPPY
€ 33,90

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

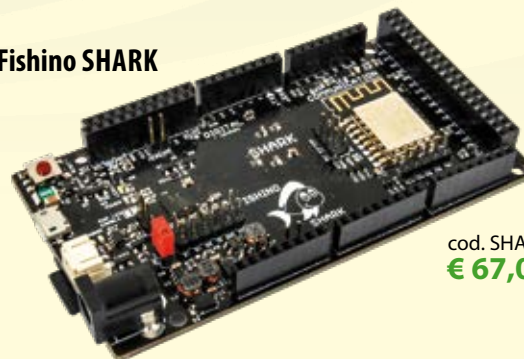
Fishino PIRANHA



cod. PIRANHA
€ 44,90

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

Fishino SHARK



cod. SHARK
€ 67,00

- Formato standard Arduino MEGA
- Processore PIC32MX470F512
- 512 kB di ROM
- 128 kB di RAM
- Frequenza di clock di 105 MHz
- Interfaccia USB nativa, sia device che host
- RTC incorporato
- Lettore per schede microSD
- Modulo WiFi integrato
- Codec audio
- Tensione di alimentazione: 3-20Vdc
- Alimentazione a batteria, plug e/o via connettore USB
- Funzionamento interno a 3,3V
- Pin DIGITALI 5V-tolerant
- Ricarica automatica di una batteria LiPo in standby
- Spegnimento da software con processore in standby
- Wake-up tramite pin esterno o in tempi prefissati

Utilizza subito Fishino UNO!

Set contenente il necessario per realizzare applicazioni con Fishino UNO.

cod. FISHINOSTARTER
€ 65,00



Prezzi IVA inclusa.

DAI FORMA ALLE TUE IDEE!

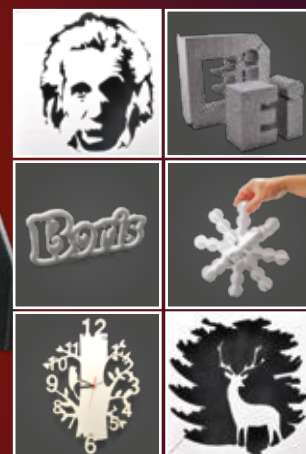


cod. FOAMCUTTER

€ 320,00

PLOTTER TAGLIA POSTIROLO - IN KIT

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



CARATTERISTICHE TECNICHE:

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

Prezzo IVA inclusa.

**FUTURA
ELETTRONICA®**

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

www.futurashop.it