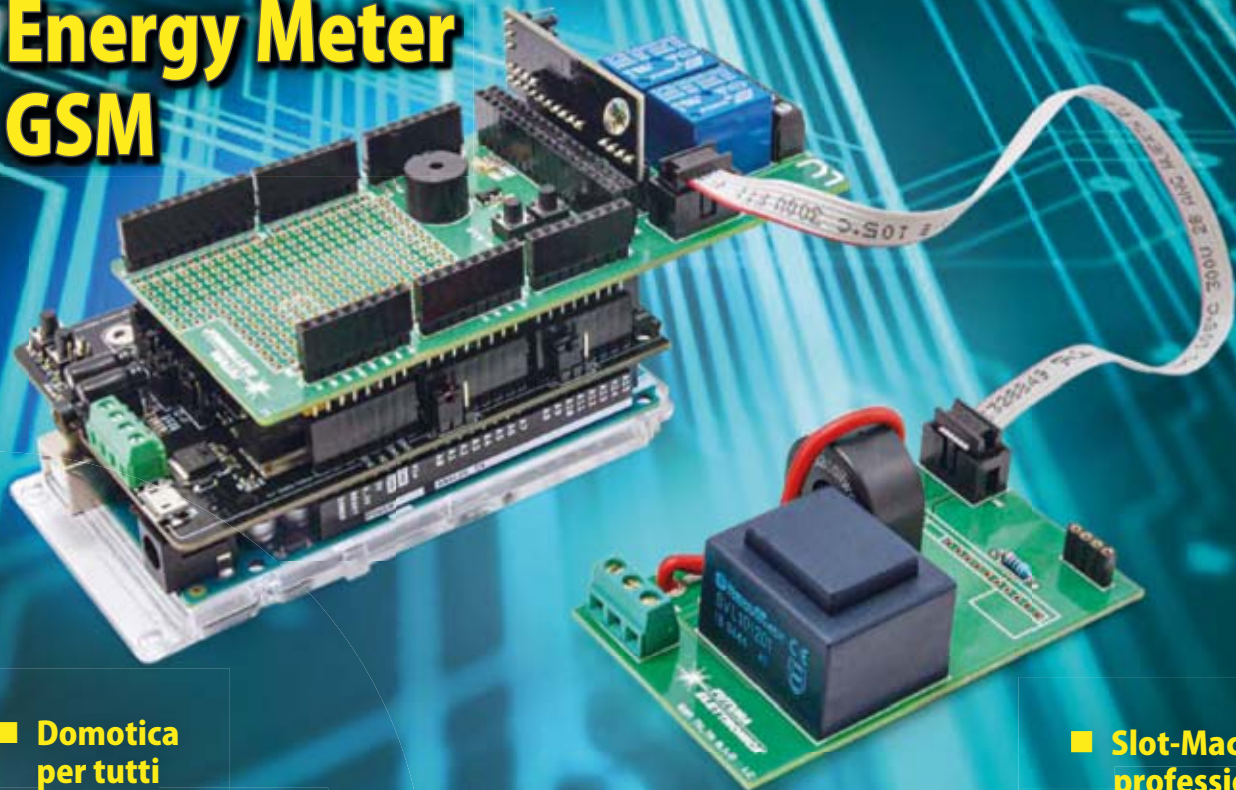


Elettronica In

www.elettronica.in.it

€ 6,00 - Anno XXV - n. 240 - NOVEMBRE 2019

Energy Meter GSM



■ Domotica
per tutti

■ Raspberry Pi
e Intel Neural
Compute Stick 2

■ My IoT:
il Raspberry Pi
per l'industria

■ Slot-Machine
professionale

■ HUB USB
a 4 porte

■ La USB
dalla A alla C

Carico elettronico



IoT water level

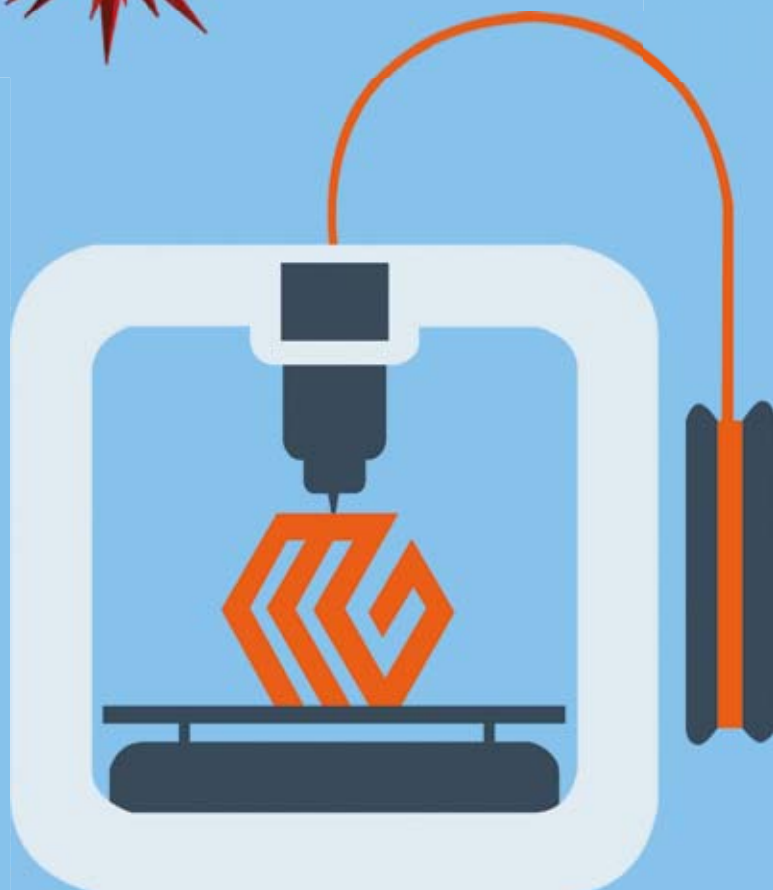


IL MONDO DELL'
IoT
INTERNET OF THINGS

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



Game Over

Quale che sia il giudizio su Greta Thunberg, sicuramente l'attività della ragazzina svedese ha riportato in primo piano il problema dei cambiamenti climatici che sempre di più sta minacciando le nostre vite e quella del nostro pianeta. Ammettiamolo, negli ultimi anni l'interesse verso i problemi ambientali ed i possibili rimedi è stato messo in secondo piano, e la nostra attenzione è stata catturata da fatti più contingenti, la contesa politica, le grandi sfide sportive, il lavoro quotidiano. Nel frattempo, inesorabile, l'inquinamento atmosferico da gas serra, causa prima dei cambiamenti climatici, è cresciuto costantemente raggiungendo la concentrazione in atmosfera di 400 ppm di CO₂ del 45% superiore al periodo preindustriale, e ormai molto vicino alla soglia di 500 ppm considerato da molti il punto di non ritorno.

Immettiamo in atmosfera oltre 37 Gt (giga tonnellate, miliardi di tonnellate) di CO₂: erano circa 10 nel 1960, 20 nel 1980 e 30 nel 2010.

Nel frattempo l'inquinamento da gas serra è cresciuto raggiungendo le 400 ppm di CO₂.

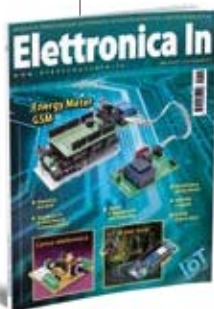
A fronte di tutto ciò, foreste ed oceani riescono ad assorbire non più di 15-20 Gt e il pianeta Terra è sempre più simile ad una serra con la temperatura media che continua a salire e i ghiacci polari (e quelli alpini) che si sciolgono

aumentando il livello dei mari. Nonostante gli accordi internazionali, le emissioni di CO₂ continuano ad aumentare, con un incremento percentuale maggiore, come era ovvio aspettarsi, da parte dei paesi in via di sviluppo. Un problema reso ancora più complesso dal fatto che sono coinvolte tutte le nazioni della Terra: se, ad esempio, le emissioni italiane di CO₂ si riducessero a zero, l'inquinamento globale calerebbe di una percentuale insignificante. Che fare dunque?

In primis c'è sicuramente la necessità di una forte presa di coscienza basata su dati oggettivi, alimentata da programmi per la scuola e sostenuta dalla TV pubblica.

In secondo luogo vengono i comportamenti personali in grado di ridurre l'impatto sull'ambiente: ridurre l'uso dell'auto, migliorare l'isolamento termico di abitazioni e uffici, mangiare alimenti di stagione, utilizzare lampadine ed elettrodomestici a basso consumo e così via.

Terzo, ma non ultimo, investire pesantemente nelle tecnologie in grado di mitigare l'inquinamento globale. Tra queste, le più importanti riguardano la produzione di energia elettrica da fonti rinnovabili (sole, vento, idroelettrico, ecc.). La produzione di energia elettrica è causa oggi del 25% delle emissioni di gas serra; se, con una sorta di Piano Marshall, riuscissimo in pochi anni ad eliminare le centrali alimentate a combustibili fossili, potremo ridurre di un quarto l'inquinamento atmosferico globale, con un successivo ulteriore miglioramento del 15% dovuto all'elettrificazione dei trasporti. Come dice Greta, il tempo è finito, non possiamo più aspettare.



In copertina:
Controlliamo da remoto il consumo di elettricità utilizzando un apposito sensore, un Arduino Mega ed una connessione cellulare GSM che invia i dati al cloud.

Arsenio Spadoni

✉ arsenio.spadoni@elettronica.in



Elettronica In

Rivista mensile, anno XXV n. 240
NOVEMBRE 2019

Direttore Responsabile

Arsenio Spadoni

✉ arsenio.spadoni@elettronica.in

Redazione

Stefano Garavaglia, Paolo Gaspari,
Boris Landoni, Davide Scullini,
Gabriele Daghetta, Alessandro Sottocornola

✉ redazione@elettronica.in

Grafica

Alessia Sfulcini

✉ alessia.sfulcini@elettronica.in

Ufficio Pubblicità

Monica Premoli (0331-752668)

✉ monica.premoli@elettronica.in

Ufficio Abbonamenti

Elisa Guarniero (0331-752668)

✉ elisa.guarniero@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

Estero 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.

© 2019 Futura Group srl

Futura
Group
Edizioni

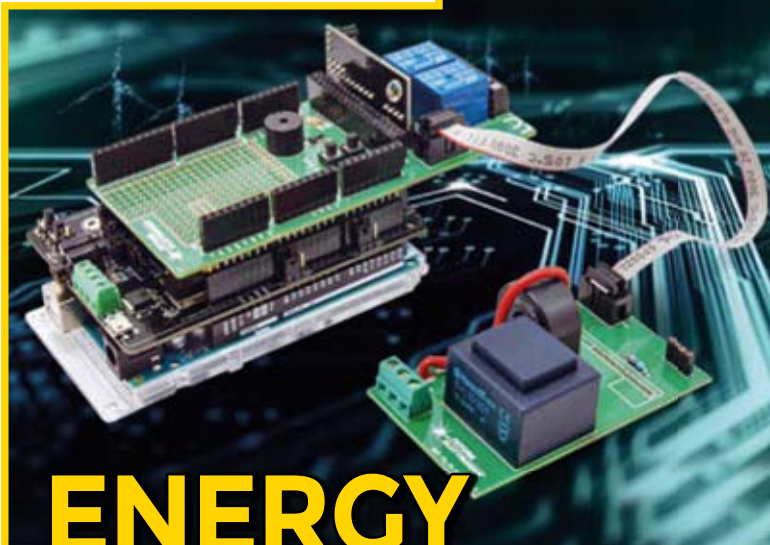
Contenuti

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



ENERGY METER GSM

66

Didattica

Domotica per tutti

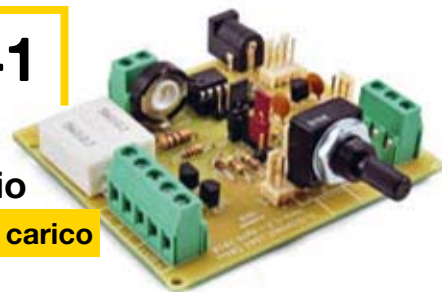


23

41

Laboratorio

Un semplice carico elettronico



23

ARTICOLI

DIDATTICA

23 Domotica per tutti

Proseguiamo l'analisi delle periferiche a bordo della scheda CM3-home conoscendo i bus 1-wire e I²C, le periferiche di input e quelle di output.

RASPBERRY PI

33 Raspberry Pi e Intel Neural Compute Stick 2

Creare applicazioni basate su reti neurali in modo efficiente con un minuscolo kit di sviluppo di Deep Learning, progettato per dar vita a una nuova generazione di dispositivi intelligenti.

LABORATORIO

41 Un semplice carico elettronico

Costruiamo un circuito in grado di simulare, pilotando un transistor di potenza, una resistenza variabile che funziona da carico in corrente continua e che nel caso specifico è destinato al test dei regolatori di corrente per illuminazione a LED.

Raspberry Pi

Raspberry Pi e Intel Neural Compute Stick 2



33

Periferiche**Hub USB2 a 4 porte****49****Gadget****Slot-machine professionale****81****Networking****IoT Water Level****91****Tecnologia****La USB dalla A alla C****57****Raspberry Pi****MY IoT****myIoT**
it makes things easy**99****Didattica****Corso IoT****113****PERIFERICHE****49****Hub USB2 a 4 porte**

Basato su un single-chip della Terminus, consente di connettere quattro dispositivi USB partendo da una porta microUSB; il form-factor è stato studiato per l'abbinamento a Raspberry Pi Zero.

TECNOLOGIA**57****La USB dalla A alla C**

Scopriamo l'evoluzione di uno dei bus di comunicazione più affermati che ha saputo tenersi al passo coi tempi.

STRUMENTAZIONE**66****Energy meter GSM**

Monitora il consumo di elettricità e invia i dati rilevati nel cloud attraverso la rete cellulare GSM; è basato su una Arduino Mega equipaggiata di GSM Shield e su una scheda contenente il rilevatore di assorbimento. Prima puntata.

GADGET**81****Slot-machine professionale**

Completiamo la descrizione del nostro "gioco da bar" domestico e prepariamoci a utilizzarlo dopo averne caricato il firmware. Seconda e ultima puntata.

NETWORKING**91****IoT Water Level**

Dispositivo IOT con tecnologia LoRaWAN per il monitoraggio del livello dell'acqua di un laghetto o di un corso d'acqua.

RASPBERRY PI**99****My IoT**

Un "dispositivo" in grado di rendere Raspberry Pi adatto ad essere utilizzato in ambienti "professionali", basato sul "Compute Module 3".

DIDATTICA**113****Il mondo dell'Internet of Things**

Sperimentiamo l'IOT con l'impiego della rete a basso consumo SigFox, alternativa alle reti wifi e GSM. Quarta e ultima puntata.

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.

Q & A

Gesture control con APDS-9960

Devo integrare un semplice gesture control in un'apparecchiatura audio. Quale prodotto mi consigliate?

Giuseppe Manera • Torino

R: Una soluzione molto semplice è rappresentata dal gesture control APDS-9960 di Broadcom, un sensore di controllo dei gesti ad infrarossi facilmente integrabile in qualsiasi sistema embedded.

La teoria alla base dei sensori di gesti basati su IR è relativamente semplice. Esistono diversi gesti della mano che gli sviluppatori potrebbero voler rilevare:

Movimento verso l'alto/il basso

Movimento verso sinistra/destra

Movimento in avanti/indietro

In ognuno di questi casi il sensore deve poter rilevare la direzione del movimento. Per farlo vengono usati due suoi componenti principali: un

diodo luminescente (LED) e fotodiodi direzionali. Questi ultimi sono costituiti semplicemente da un gruppo di almeno quattro fotodiodi posizionati a una distanza predefinita dal LED IR.

Ad esempio, i fotodiodi del sensore

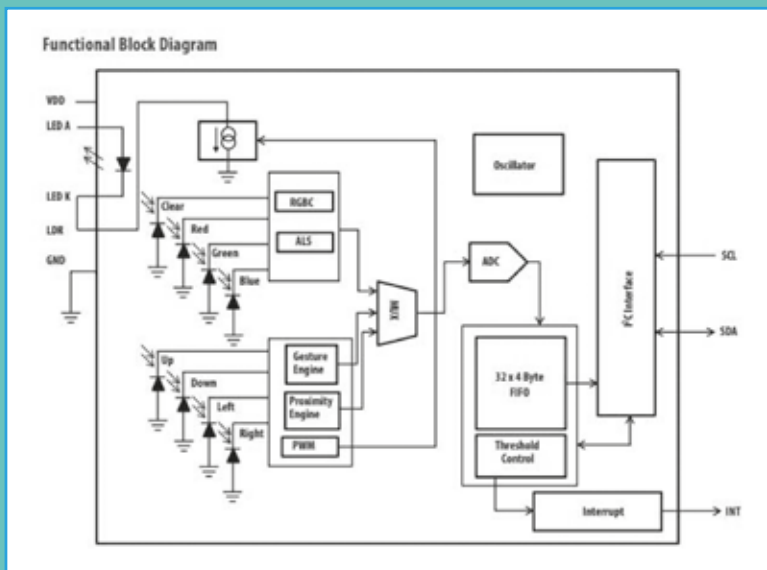
Queste pagine sono dedicate alle richieste, ai suggerimenti ed alle segnalazioni dei lettori. Raccomandiamo, per quanto possibile, di proporre argomenti di interesse generale. Contattateci all'indirizzo: redazione@elettronica.it



di gesti, luce ambiente e prossimità APDS-9960 di Broadcom sono disposti a rombo e ognuno di essi ha un indirizzamento direzionale: su, giù, a sinistra e a destra.

Quando il LED trasmette energia a infrarossi, l'energia verrà emessa in uno spazio vuoto, a meno che in tale spazio non vi sia un oggetto, ad esempio una mano, che la riflette. L'energia riflessa verrà rilevata dai fotodiodi a diverse intensità, a seconda della posizione occupata dall'oggetto. Ad esempio, un fotodiodo posto all'inizio del gesto riceverà inizialmente meno energia riflessa di un fotodiodo posto alla fine, per cui un fotodiodo avrà un conteggio più alto dell'altro. Effettuando misurazioni durante il gesto, i fotodiodi rilevano diverse intensità riflesse nei punti in cui si trovano i vari diodi. Quindi, per determinare il gesto si può analizzare questo flusso di informazioni direzionali.

Una volta compreso come vengono generati i dati per un gesto, il passo successivo consiste nel vedere come interfacciarsi con l'APDS-9960. Quest'ultimo viene fornito in un contenitore a montaggio superficiale a 8 pin (SMD-8) che occupa pochissimo spazio sulla scheda (Figura 3). Il sensore misura solo 3,94x2,36x1,35 mm. Il contenitore è provvisto dei normali pin di alimentazione e messa a terra, oltre che di un'interfaccia I₂C per una connessione digitale con un microcontroller e di pin per personalizzare i circuiti di pilotaggio dei LED. Prevede



inoltre un pin di interrupt per comunicare al microcontroller quando sono disponibili i dati dei gesti da elaborare. Per realizzare un prototipo e interfacciarsi con APDS-9960 esistono numerose opzioni. Ad esempio, la scheda di valutazione APDS-9960 di SparkFun (disponibile presso Futura Elettronica) è una scheda di breakout relativamente piccola che comprende i circuiti di pilotaggio dei LED, per cui è pronta per essere utilizzata da subito. Lo sviluppatore non deve far altro che saldare una basetta per ponticellare alimentazione e terra, collegare il bus I₂C e il pin di interrupt opzionale a un microcontroller, dopo di che può iniziare a sviluppare il proprio software embedded.

Massima sicurezza col Secure Element EdgeLock SE050

Dobbiamo rendere sicura una rete IoT che stiamo progettando. La nostra attenzione è caduta sul dispositivo SE050 di NXP. Voi cosa ne pensate?

Mario Riotta > Roma

R: Il Secure Element EdgeLock SE050 di NXP è stato progettato specificamente per offrire ai dispositivi IoT la serie di meccanismi di protezione e i protocolli necessari per la sicurezza edge-to-cloud. Progettato come soluzione chiavi in mano per la protezione dei dispositivi IoT, SE050 di NXP, che si avvale della grande esperienza dell'azienda nel settore della sicurezza delle smart card, rappresenta una root-of-trust hardware per i progetti IoT. Il dispositivo abbina un microcontroller dedicato e la memorizzazione sicura delle credenziali con uno stack software costruito sul sistema operativo (OS) per le smart card Java Card OpenPlatform (JCOP). La piattaforma operativa di SE050, che ha ottenuto la certificazione indipendente Common Criteria (CC) Evaluation Assurance Level (EAL) 6 augmented (EAL 6+) sino al sistema operativo garantisce un grado di protezione elevatissimo. Un'applet NXP dedicata per la sicurezza dell'IoT, ottimizzata per questa solida piattaforma di protezione, si avvale di una libreria crittografica incorporata che supporta un vasto insieme di algoritmi di crittografia tra cui: Crittografia simmetrica con algoritmi Advanced Encryption Standard (AES) e Data Encryption Standard (DES)

Crittografia asimmetrica con algoritmi Rivest-Shamir-Adleman (RSA) e di crittografia ellittica (ECC) con il supporto per numerose curve ECC, tra cui le curve standard specificate dal National Institute of Standards and Technology (NIST). Numerose funzioni di hash per la costruzione di codici di autenticazione dei messaggi (MAC). Numerosi algoritmi di derivazione della chiave (KDF) per l'hash della password, la corrispondenza chiavi e il rafforzamento della chiave tra gli altri metodi. Per velocizzare le prestazioni, il dispositivo SE050 incorpora acceleratori hardware per l'Advanced Encryption Standard (AES), il Data Encryption Standard (DES) e la crittografia Fast Attribute-based Message Encryption (FAME). Per proteggere le credenziali utilizzate nelle operazioni di crittografia, il dispositivo fornisce sino a 50 KB di memoria utente flash sicura e viene fornito dotato di chiavi radice per il supporto di transazioni IoT tipiche. Gli sviluppatori che creano progetti per volumi elevati possono anche disporre di chiavi personalizzate e credenziali sicure assegnate da NXP o da terzi autorizzati. Il dispositivo, oltre alla memoria flash embedded, supporta l'accesso alla memoria ad accesso casuale (RAM) per la memorizzazione di certificati pubblici, ad esempio, mentre le chiavi private associate rimangono nella memoria flash sicura. Oltre a ciò, l'applet SE050 per l'IoT dispone di capacità di importazione ed esportazione sicure che consentono gli sviluppatori di memorizzare chiavi o dati arbitrari in una memoria esterna senza correre rischi. Nonostante ciò, il dispositivo si avvale di una unità di gestione della memoria che limita l'accesso a diverse parti della memoria alla sola applet per l'IoT, impedendo l'accesso non autorizzato attraverso il microcontroller. Per garantire ulteriore protezione contro le manomissioni e sofisticati attacchi al canale laterale, il dispositivo incorpora anche numerose barriere logiche e fisiche contro gli accessi

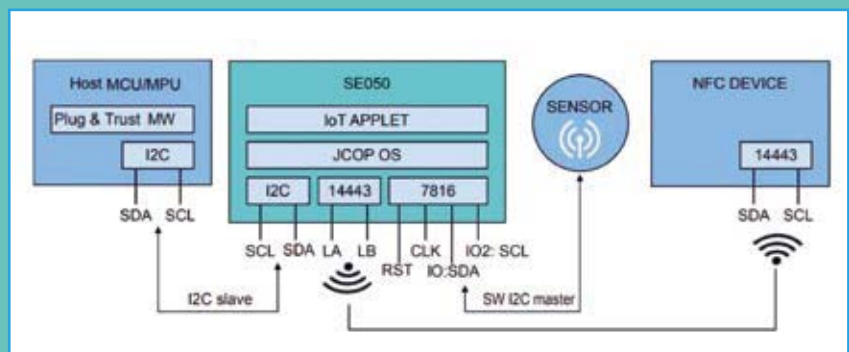
non autorizzati. Alle funzionalità avanzate di SE050 si aggiunge la facilità di design-in. Gli sviluppatori possono facilmente integrare il dispositivo nei loro progetti hardware per l'IoT con un impatto minimo sulla complessità della progettazione o sull'ingombro. Progettato per fungere da dispositivo add-on, con i suoi 3 millimetri per 3 offre un insieme semplice di interfacce seriali. Il dispositivo funge da slave I₂C per comunicare con il microcontroller host. SE050 può anche fungere da master I₂C per collegamenti basati su ISO 7816 con un altro dispositivo seriale come un sensore digitale. Il dispositivo supporta anche una terza interfaccia per la connettività ISO 14443 NFC senza contatto. L'applet SE050 per l'IoT presiede al coordinamento delle chiavi sicure e delle regole di controllo degli accessi per la configurazione e la gestione delle sessioni di ciascuna connessione per tutti i suoi diversi collegamenti. Nelle comunicazioni con l'host, SE050 fornisce un'autenticazione basata su sessione, al fine di impedire l'intercezione di comandi e dati I₂C e gli attacchi MIMT (Man-in-the-Middle). Per dare avvio a una sessione seriale di connessione, gli sviluppatori possono realizzare l'autenticazione con un ID utente che funziona come una password o un numero identificativo personale (PIN).

La tecnologia SiC di Tesla

È vero che la Tesla Model 3 è stata la prima vettura elettrica ad adottare i moduli di potenza realizzati con la tecnologia SiC?

Roberto Milano > Como

R: Nonostante se ne parli da tempo, solo negli ultimi due anni la tecnologia SiC (Silicon on Carbide) ha trovato applicazioni su larga scala, specie nel settore automobilistico. Ed è stata proprio la Tesla la prima





Casa automobilistica di auto elettriche a integrare un completo modulo di potenza SiC nell'inverter principale della Tesla Model 3. I modelli precedenti, le Model S e X, utilizzavano IGBT in contenitori TO-247. L'inverter di Tesla, frutto di una collaborazione con STMicroelectronics, è composto da moduli di potenza SiC montati su un dissipatore di calore. I MOSFET, come l'SCT10N120 di ST, hanno una tensione nominale di 650 volt e per la loro dissipazione termica vengono usate piastre base in rame. In generale, per soddisfare i requisiti di efficienza e densità di potenza delle moderne vetture elettriche, i progettisti devono utilizzare materiali avanzati per i semiconduttori di potenza.

Questi materiali - principalmente il nitruro di gallio (GaN) e il carburo di silicio (SiC) - rappresentano un miglioramento rispetto alle tecnologie dei semiconduttori esistenti, come i transistor a effetto di campo metallo-ossido-semiconduttore (MOSFET) e i transistor bipolari a gate isolato (IGBT). Garantiscono infatti perdite più basse, frequenze di commutazione più elevate, temperature di funzionamento più alte, robustezza in ambienti difficili e tensioni di rottura superiori. Dato che il settore si sta orientando verso batterie

di maggiore capacità che funzionano ad alte tensioni con tempi di ricarica più brevi e perdite complessivamente ridotte, la grande utilità di questi materiali è evidente. Nei semiconduttori WBG (Wide Band Gap) come SiC e GaN, la "banda proibita" è molto superiore rispetto a quella del silicio (1,1 eV): quella del SiC è di 3,3 eV mentre quella del GaN è di 3,4 eV. Per banda proibita si intende l'energia necessaria per eccitare gli elettroni dalla banda di valenza di un materiale a quella di conduzione. I semiconduttori WBG consentono ai dispositivi di funzionare con tensioni, frequenze e temperature molto più elevate rispetto al silicio convenzionale. Inoltre, le perdite di commutazione e di conduzione sono inferiori. I materiali WBG hanno proprietà di conduzione e di commutazione circa dieci volte migliori del silicio. Di conseguenza, la tecnologia WBG rappresenta un'applicazione naturale per l'elettronica di potenza, in particolare per le auto elettriche, in quanto i componenti in SiC e GaN possono avere dimensioni più piccole, velocità più elevate ed essere più efficienti. I vantaggi dei dispositivi WBG vanno però valutati a fronte della complessità della produzione e dei maggiori costi

di produzione in serie. Anche se inizialmente i componenti WBG potrebbero essere più costosi, va detto che il loro costo continua a scendere e, in linea di massima, consentiranno di risparmiare sui costi complessivi del sistema. Ad esempio, l'uso di dispositivi SiC in un'auto elettrica potrebbe far salire i costi iniziali di diverse centinaia di euro, ma il risultato è un risparmio complessivo perché le batterie costano meno, richiedono meno spazio e le soluzioni di raffreddamento sono più semplici. L'inverter che controlla il motore elettrico di trazione nei veicoli elettrici è un esempio di componente fondamentale in un sistema EV che può trarre vantaggio dai dispositivi WBG. La funzione principale dell'inverter è quella di convertire una tensione c.c. in una forma d'onda trifase c.a. per azionare il motore dell'auto e quindi convertire la tensione c.a. sviluppata dalla frenata rigenerativa in una tensione c.c. per ricaricare la batteria. Dato che, per azionare il motore elettrico, l'inverter converte in corrente alternata l'energia immagazzinata nel pacco batterie, più basse sono le perdite durante questa conversione, più efficiente sarà il sistema.

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.elettronicaain.it/webforum



RM ELETTRONICA

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

Distributore autorizzato:

**FUTURA
ELETTRONICA**

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

Il tuo punto
di riferimento
per l'elettronica
a ROMA

Date il benvenuto al regolatore switching con l' I_Q piu' basso al mondo.



Estendete la durata della batteria del vostro sistema grazie alla piu' alta efficienza possibile a carico minimo.

<http://www.ti.com/product/TPS62840>

Il governo australiano ha iniziato a installare, nella Nuova Scozia, telecamere abbinate a un sistema di intelligenza artificiale per scovare e multare chi usa il cellulare alla guida senza utilizzare vivavoce o auricolare. Entro dicembre l'autorità punta a portare a 45 il numero di telecamere installate e operative. Le immagini riprese dalle telecamere, ad alta risoluzione, vengono esaminate da un algoritmo di Intelligenza Artificiale che effettuerà il riconoscimento e attiverà il sistema sanzionatorio automatico che produrrà e recapiterà la multa, di ben 344 dollari australiani.

Questo sistema è stato adottato dopo la constatazione che molti automobilisti si distraggono utilizzando social media, SMS ed anche facendo le comuni telefonate.

Ciascuna unità del sistema di visione contiene, in realtà, due telecamere: una fotografa la targa dell'autoveicolo, e una seconda

Al nella sicurezza stradale



con obiettivo puntato dall'alto, "guarda" in basso attraverso il parabrezza e può vedere cosa fanno i conducenti con le mani. L'unità utilizza l'Intelligenza Artificiale per escludere i conducenti che non toccano il telefono; le foto che mostrano sospetti comportamenti illeciti sono sottoposte a verifica da parte di agenti prima di inviare una notifica di violazione al proprietario del veicolo.

Alcune telecamere verranno fissate permanentemente ai bordi della strada e altre saranno posizionate su rimorchi e spostate nelle strade dello Stato.

Una prova di sei mesi di due telecamere fisse quest'anno ha verificato 8,5 milioni di veicoli e rilevato più di 100.000 conducenti con le mani sui telefoni, incluso un conducente che utilizzava contemporaneamente un telefono e un iPad.

www.nsw.gov.au/news-and-events

5G sul campo nell'agricoltura

L'operatore di telefonia mobile svizzero Sunrise, il colosso cinese Huawei e Agroscope, centro di competenza della Confederazione per la ricerca agricola, hanno firmato una dichiarazione di intenti per cooperare nel campo dell'integrazione della tecnologia wireless 5G nell'agricoltura. Il 5G può ritagliarsi un ruolo chiave nella gestione delle risorse naturali e nella protezione dell'ambiente e il progetto nasce in quest'ottica.

Le applicazioni della nuova generazione di telefonia saranno testate nell'ambito della cosiddetta Swiss Future Farm a Tänikon (TG), una piattaforma di promozione delle nuove tecnologie nel settore agroalimentare.

La cosiddetta Swiss Future Farm è una fattoria intelligente che rappresenta una piattaforma unica in Europa per l'applicazione delle nuove tecnologie nel settore agroalimentare. Questo, unito all'uso delle tecnologie informa-



tiche e della comunicazione in agricoltura, può portare ad alzare la resa, riducendo i costi e i consumi energetici per le lavorazioni; il 5G diverrà il link per interconnettere sensori e sistemi, co-

diuvando tecnologie collaudate come il GPS applicato alla guida automatica delle macchine agricole nella semina.

www.swissfuturefarm.ch

› Gateway IIoT basato su Raspberry Pi

Farnell, Development Distributor, rende finalmente disponibili l'arrivo di Avnet SmartEdge Industrial IoT Gateway, il gateway industriale a basso costo basato sulla popolare board Raspberry Pi. La presentazione del dispositivo è avvenuta alla scorsa edizione di Embedded World di Norimberga e subito dopo è stata registrata una fortissima richiesta di tale gateway per l'utilizzo in varie applicazioni, tra cui gestione degli edifici, monitoraggio dei veicoli, automazione industriale e manutenzione predittiva.

Il gateway, insieme alla connettività continua e sicura resa possibile dalla piattaforma IoTConnect di Avnet, offre una

serie di funzionalità potenti che consentono la raccolta, l'analisi e la visualizzazione in tempo reale dei dati utili, il tutto a un prezzo imbattibile. La piattaforma IoTConnect di Avnet si basa su Microsoft Azure, un set completo di servizi edge-to-cloud che sono pronti per la costruzione, implementazione e gestione da parte delle aziende degli asset connessi a IoT.

Inoltre l'utilizzo dell'Avnet SmartEdge Industrial IoT Gateway offre una connettività versatile, supportata da un'analisi di data intelligence adatta e un'ampia gamma di applicazioni industriali 4.0, tra cui porte aggiuntive per gestire i protocolli industriali, una gamma più elevata di input di potenza, un mo-

› Sicurezza informatica per dispositivi IoT

Sternum, azienda israeliana operante nella sicurezza informatica e nota per l'offerta di soluzioni nate con l'obiettivo di migliorare la sicurezza dei dispositivi medici connessi, propone la propria soluzione "Verifica di integrità integrata" per proteggere i prodotti connessi all'IoT. Installando la soluzione EIV (Embedded Integrity Verification) in un dispositivo IoT, vengono attivati diversi punti di verifica nel codice in grado di verificare se l'integrità del dispositivo è stata violata o meno. Questo tema è quanto mai attuale perché la diffusione dell'Internet of Things sta ponendo seriamente il tema

della sicurezza informatica e della protezione dalle intrusioni, attraverso dispositivi periferici ed edge-computing, nella rete globale. La strategia adottata sinora dai provider, tuttavia, è mirata a irrobustire dispositivi e software nei confronti degli attacchi informatici e ad istruire gli utenti finali sull'impiego delle buone pratiche di sicurezza per ridurre i rischi connessi alle vulnerabilità.

Tuttavia, anziché tentare di bloccare i cyber attacchi ai dispositivi IoT mediante l'uso delle best practices, sarebbe meglio che essi potessero essere protetti dall'interno. Questo è esattamente l'obiettivo

che si è posto Sternum, la cui soluzione EIV fornisce sicurezza end-to-end a un'ampia varietà di dispositivi IoT.

La maggior parte delle soluzioni di sicurezza oggi si concentra sulla sicurezza della rete, nel senso che accede alla rete, analizza i dispositivi IoT collegati e verifica l'esistenza di anomalie e vulnerabilità. Altre soluzioni utilizzano analisi statiche per prevedere le vulnerabilità all'interno del codice.

Quello che l'EIV Sternum fa, invece, è cercare di proteggere il dispositivo stesso indipendentemente dal sistema operativo o dalla vulnerabilità, proprio andando a verificare all'interno del codice.

Quindi l'EIV è fondamentalmente un firewall software integrato nel dispositivo. Il prodotto della Sternum nasce dall'esperienza maturata nel campo dei dispositivi medici IoT, che si basano su una varietà di sistemi operativi real-time che non fanno parte di alcuna rete organizzata e la loro protezione è vitale, perché significa che è possibile fornire assistenza remota e monitoraggio per i pazienti senza preoccuparsi degli attacchi informatici, specie quando si tratta di situazioni dove è a rischio la vita del paziente.

<https://www.sternumiot.com>



dulo computerizzato affidabile che garantisce la sicurezza dei dati hardware end-to-end, nonché una robusta piastra o un contenitore di metallo montabile su guida DIN. Oltre a queste caratteristiche, il gateway IoT presenta livelli di semplicità, capacità e accessibilità senza precedenti in questo settore, grazie anche alla diffusione e al supporto di Raspberry Pi.

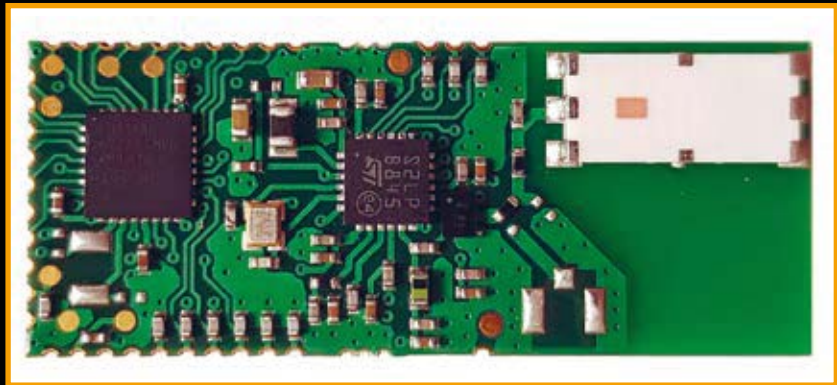
Per gli sviluppatori che hanno familiarità con la prototipazione e lo sviluppo su un dispositivo Raspberry Pi, sarà molto semplice realizzare i propri progetti IoT e portarli in un ambiente di produzione, senza dover cambiare la piattaforma hardware.

www.huawei.com/en



› SigFox: nuovi moduli certificati

Arrow Electronics rende disponibili i moduli SigFox-MOD1 certificati SigFox, creati da XoverIoT, che aiutano ad accelerare lo sviluppo delle soluzioni Internet of Things, sfruttando la connettività wireless SigFox a livello globale e a bassa energia. Tali moduli sono disponibili in due versioni: SIGFOX-MOD1-C con antenna a chip integrata e SIGFOX-MOD1-E con una presa UFL per un'antenna esterna. La flessibilità di riconfigurazione dinamica per il funzionamento FSK a breve distanza o per la comunicazione a lungo raggio consente agli sviluppatori di creare soluzioni connesse per affrontare le sfide del monitoraggio industriale, del tracciamento degli asset, della supply chain e dell'infrastruttura intelligente. Il SIGFOX-MOD1 di XoverIoT sono basati sul chipset SigFox di STMicroelectronics, che comprende il microcontrollore STM32L051K8, il ricetrasmittitore S2-LPQTR 868MHz e il balun integrato BALF-SPI2-01D3, che consente di risparmiare spazio e facilita la progettazione del circuito RF. Il modulo è compatto, misura solo mm 37,6 x 15,0 x 3,8, e consente una lunga durata della batteria, sfruttando le tecnologie a bassissimo consumo del



microcontrollore STM32L0 e del radio transceiver S2-LP. La corrente operativa è di soli 29 mA in modalità di trasmissione con alimentazione di 3,3 V. Il chipset ST su cui sono basati questi moduli è certificato per tutte le zone geografiche SigFox (ZC1+ZC6). I moduli sono dotati di firmware incorporato, che comprende lo stack software SigFox, facilmente controllabile dall'applicazione mediante i comandi AT standard. In alternativa, sfruttando la capiente Flash da 64 kbyte del microcontrollore, si può scegliere di eseguire l'applicazione sul modulo, allo scopo di ottenere la massima efficien-

za e risparmiare sui costi di sistema. È inoltre disponibile una serie completa di tool, tra cui il kit di sviluppo del modulo STM32Nucleo, compatibile con SIGFOX-DEV1 Arduino, il kit mikroBUS SIGFOX-CLK1 e il dongle USB SIGFOX-USB2, per effettuare la comunicazione fra il modulo di test e il modulo base. In collaborazione con Arrow, XoverIoT offre anche altri servizi tecnici per supportare il design-in, la progettazione dell'antenna, l'approvazione del prodotto e la personalizzazione del firmware.

www.arrow.com

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:

500 Kg	cod. LCST500KG	€44,00
300 Kg	cod. LCST300KG	€44,00
200 Kg	cod. LCST200KG	€44,00
100 Kg	cod. LCST100KG	€44,00

cod. LCST1TONS

€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



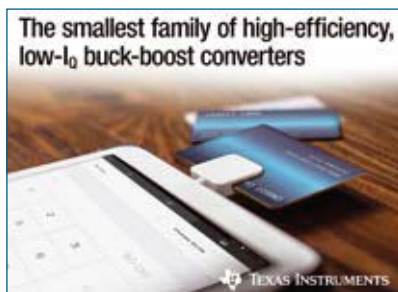
Set composto da una scheda elettronica di interfaccia e da una cella di carico da 5 kg montata all'interno di una struttura meccanica. La scheda elettronica è basata sul chip HX711, un convertitore analogico-digitale a 24 bit, specifico per interfacciare la cella di carico a un microcontrollore come Arduino, Fishino o compatibili.

cod. YB723

€14,50

Nuova famiglia di convertitori buck-boost di TI da 2,5 A

Texas Instruments ha presentato una famiglia di quattro convertitori buck-boost ad alta efficienza e bassa corrente di quiescenza (QI) che presentano un packaging minuscolo e consentono soluzioni di piccole dimensioni grazie anche all'impiego di pochissimi componenti esterni. I convertitori integrati non inverting buck / boost TPS63802, TPS63805, TPS63806 e TPS63810 DC / DC offrono ampi intervalli di tensione di ingresso e uscita che supportano numerose appli-



cazioni a batteria, aiutando gli ingegneri a semplificare e accelerare i loro progetti.

Ciascuno dei dispositivi della famiglia seleziona automaticamente la modalità buck, buck-boost o boost in base alle condizioni operative. Le dimensioni complete della soluzione da 19,5 mm² a 25 mm² – sono fino al 25% più piccole rispetto a dispositivi simili – sono il risultato di un packaging compatto e avanzato che richiede pochi condensatori ceramici multistrato esterni e piccoli induttori da 0,47 µH.

www.ti.com

Termocamera ad alte prestazioni per ispezioni industriali

FLIR ha annunciato FLIR T860, l'ultima novità della famiglia ad alte prestazioni T-Series. Questa nuova termocamera è la prima ad integrare il software Inspection Route, ideale per ottimizzare le ispezioni di asset critici, tra cui componenti di sottostazioni elettriche, linee di distribuzione, macchinari di produzione o sistemi elettrici e meccanici. Attenendosi al percorso ispettivo pre-programmato nella termocamera, gli ispettori termografici possono dedicare meno tempo sul campo e creare agevolmente i rapporti dei rilievi. La termocamera a risoluzione 640x480 incorpora la

tecnologia FLIR Vision Processing avanzata brevettata MSX ed UltraMax per garantire una maggiore nitidezza dell'immagine, dimezzando il rumore digitale dei modelli precedenti. L'abbinamento della T860 con un teleobiettivo termico opzionale a 6 gradi consente di ispezionare elementi sotto tensione.

www.flir.com



LED Driver per Flash a 2 e 4 canali

Diodes Incorporated ha annunciato l'AL3644, un driver LED flash a doppio canale ad alta velocità progettato per supportare le funzionalità avanzate di flash della fotocamera e torce, come offerto dagli ultimi smartphone e altri dispositivi portatili di consumo. Le caratteristiche principali includono correnti di uscita controllate indipendentemente configurate attraverso un'interfaccia compatibile con I2C e la possibilità di combinare due dispositivi (con indirizzi di identificazione dei dispositivi diversi) per pilotare quattro LED fino a 6A per applicazioni quad-channel. L'AL3644 è offerto in confezioni in chip-scale packaging e combina

frequenza di commutazione flessibili con corrente di uscita costante.

L'AL3644 impiega un convertitore PWM in modalità corrente a frequenza costante e boost sincrono, per fornire fino a 1,5 A per ciascuna delle due fonti di corrente costante nell'intervallo di tensione di ingresso da 2,5 V a 5 V. L'interfaccia, compatibile con I2C, funziona fino a 400kHz..

www.diodes.com



Spectrum Analyzer da 145 e 170 GHz per applicazioni in onde millimetriche

Anritsu Corporation ha ampliato la sua serie di Spectrum Master MS2760A ultraportable spectrum analyzer con una copertura estesa della frequenza a banda larga da 9 kHz a 170 GHz, nonché con una nuova serie di modelli a sensibilità più elevata, lo Spectrum Master MS2762A, con copertura da 6 GHz a 170 GHz per i più esigenti requisiti di sensibilità. Questa famiglia di soluzioni tascabili basate su

tablet combina flessibilità e portabilità con il migliore range dinamico, consentendo agli ingegneri di progettazione e produzione di eseguire misure di spettro di base, come potenza di canale, larghezza della banda occupata, potenza del canale adiacente e misure armoniche, su segnali CW fino a 170 GHz. La famiglia di analizzatori di spettro ultraportatili Spectrum Master MS276xA fornisce ora

una suite completa di strumenti di verifica a basso costo per un numero crescente di applicazioni



emergenti in onde millimetriche (mmWave). Con gli analizzatori di spettro ultraportatili possono essere condotti test OTA estesi, in alternativa affidati ad analizzatori di spettro voluminosi o misuratori di potenza con un intervallo di lettura limitato. Hanno inoltre un costo inferiore del 30% rispetto a quello dei tradizionali analizzatori di spettro da banco per mmWave.

www.anritsu.com

MAX77827, convertitore buck-boost da 1,5A estende la durata della batteria

Con la più bassa corrente di quiescenza (I_q) nella sua classe del valore di $6\mu A$, così come la più elevata efficienza di picco del settore pari al 96 per cento, i progettisti possono ora massimizzare la vita della batteria di un dispositivo portatile con il convertitore buck-boost MAX77827 di Maxim

Integrated. Questo convertitore compatto ad alta efficienza da



1,5 A consente di lavorare con un ingresso da 1,8 V a 5,5 V e uscita da 2,3 V a 5,3 V, fornendo al contempo la stabilità del sistema. Il MAX77827 soddisfa i requisiti di alimentazione delle applicazioni a bassa potenza per reti WAN (LPWAN), dei dispositivi per il tracciamento dei beni e di mol-

te applicazioni per Internet delle Cose (IoT). Con la più bassa corrente di riposo del settore, la più elevata efficienza di guadagno e le dimensioni ridotte, il convertitore buck-boost MAX77827 definisce lo standard per massimizzare la vita della batteria.

www.maximintegrated.com

STM32GO: un micro STM32 a 32 bit in contenitore 8 pin

I quattro nuovi dispositivi STM32GO offrono una combinazione unica ed economia a 8 pin con una CPU Arm Cortex-M0+ a 64 MHz che offre 59 DMIPS, fino a 8 KB di RAM e 32 KB di memoria flash su chip e periferiche ad alte prestazioni tra cui un ADC da 2,5 Mbps, timer ad alta risoluzione e SPI ad alta velocità. Con una mappatura flessibile dei pin I/O e delle funzioni MCU interne, i progettisti possono aggiornare la funzionalità del prodotto finale facilmente e senza costi. La stabilità dell'oscillatore interno, che è precisa al $\pm 1\%$ su ampi intervalli di temperatura e tensione, consente di risparmiare componenti di clock

esterni.

Beneficiando delle comprovate funzionalità di progettazione a basso consumo della famiglia MCU STM32, i dispositivi STM32GO a 8 pin sono pronti a subentrare in applicazioni attente al consumo energetico, alle prestazioni ed ai limiti delle batterie, a tutte le normative del settore ed alle aspettative di mercato per quanto riguarda i consumi degli elettrodomestici. Le nuove MCU facilitano inoltre la futura scalabilità attraverso le funzionalità disponibili nella serie STM32GO, che offre package fino a 100 pin, fino a 512 KB di byte Flash, periferiche analogiche ad alte prestazioni aggiuntive e funzionalità di protezione informatica.

I nuovi MCU STM32GO a 8 pin sono ora disponibili in SO8N da 6 mm x 4,9 mm, a partire da \$ 0,31 per ordini di 1000 pezzi (STM32G030J6 Value Line).

www.st.com

First 8-pin STM32 MCUs with STM32GO



Toshiba lancia l'IC driver per solenoidi ad alta tensione a due canali

Toshiba Electronics Europe GmbH ha lanciato un IC driver per solenoidi a doppio canale che assicura il pilotaggio ad alta tensione con una bassa resistenza di ON. Il nuovo IC driver TB67S112PG è adatto per un'ampia gamma di applicazioni che includono elettrodomestici (condizionatori d'aria e frigoriferi) e apparecchi industriali (terminali bancari come bancomat, apparecchiature per l'automazione degli uffici e impianti per l'automazione di fabbrica).

La linea di IC driver per solenoidi di Toshiba comprende attualmente dispositivi a 4 e a 8 canali (TB67S111PG e TB67S158NG/FTG). Il nuovo TB67S112PG completa la linea di prodotti, fornendo supporto per applicazioni a doppio canale a una fascia

di prezzo più contenuta.

Il nuovo dispositivo incorpora due canali, ciascuno costituito da un MOSFET low-side e da un diodo comune di rigenerazione che è in grado di pilotare in modo indipendente un solenoide o un relè a tensioni fino a 50V. Una bassa resistenza ON, pari a soli $0,3\Omega$, consente di ottenere una tensione di uscita più elevata, con una conseguente maggiore coppia motrice e una minore generazione di calore durante il funzionamento del motore.

<https://toshiba.semicon-storage.com/eu/>



ROHM offre un nuovo driver per retroilluminazione a LED



ROHM ha comunicato oggi la disponibilità del circuito integrato per driver LED di ultima generazione, il modello BD81A76EFV-M ottimizzato per la retroilluminazione LCD del quadro strumenti, display informativi centrali e navigatori satellitari. Diversamente dai driver convenzionali a 4 canali

che supportano gli LCD fino a 8, questo circuito integrato mette a disposizione 6 canali di uscita (da 120 mA ciascuno), in grado di supportare display LCD fino alla categoria 10-12. Al tempo stesso il controllo buck-boost originale assicura la compatibilità con gli LCD sia di piccole che di grandi dimensioni, usando un

singolo driver. È così possibile sviluppare un comune design delle schede di controllo LCD, idoneo sia ai display convenzionali che ai display di grandi dimensioni di ultima generazione. Il modello BD81A76EFV-M integra il controllo buck-boost proprietario.

www.rohm.com/eu

Modulo GNSS a doppia banda ad altissima precisione per l'industria automobilistica

Quectel Wireless Solutions in collaborazione STMicroelectronics, ha annunciato il rilascio del modulo LG69T, un modulo GNSS ad alta precisione dual-band di livello automobilistico che integra tecnologie di dead-reckoning (DR) e kinematic in tempo reale (RTK, Real Time Kinematic). Il nuovo modulo Quectel, annunciato alla conferenza Apsara che si è svolta recentemente ad Hangzhou, è progettato per facilitare le prestazioni di posizionamento a cielo aperto con una precisione fino a 10 centimetri, che è attualmente la tecnologia di posizionamento più avanzata



del settore per il mercato automobilistico. Il modulo LG69T supporterà le capacità di posizionamento di precisione nei veicoli intelligenti di prossima generazione nonché in scenari di guida autonoma.

Il modulo GNSS LG69T Quectel si basa sull'IC STA8100GA di ST, l'ultimo chip di posizionamento a doppia banda di livello automobilistico con 80 canali di tracciamento e 4 canali di acquisizione rapida compatibile con molte costellazioni: GPS, BeiDou, Galileo, NAVIC / IRNSS e QZSS. Si tratta di un modulo GNSS dual-band (L1 + L5) qualificato AEC-Q100 che integra la tecnologia RTK multi-banda per una precisione a livello del centimetro. La capacità di dead-reckoning del modulo LG69T è assicurata da un'unità di misura inerziale integrata (IMU) che fornisce un posizionamento continuo ad alta precisione.

www.quectel.com

Da Microchip soluzioni di sicurezza hardware-based

Microchip ha presentato la prima soluzione pre-provisioned del settore che fornisce archiviazione sicura delle chiavi per distribuzioni di volumi bassi, medi ed elevati di dispositivi utilizzando l'ATECC608A sicuro element.

La Trust Platform di Microchip per la sua famiglia CryptoAuthentication consente alle aziende di ogni dimensione di implementare facilmente l'autenticazione sicura.

Composta da un'offerta su tre livelli, fornisce elementi di sicurezza pronti all'uso, pre-configurati o completamente personalizzabili, consentendo agli sviluppatori di scegliere la piattaforma più adatta al loro



specifico progetto. Come prima soluzione per fornire un'autenticazione sicura pronta all'uso per il mass market, il primo livello - Trust&GO - fornisce elementi sicuri pre-provisioned, zero touch, e con un MOQ, (Minimum Orderable Quantity) di appena 10 unità. Le credenziali del dispositivo sono pre-programmate, inviate e bloccate all'interno dell'ATECC608A.

www.microchip.com

I nuovi SAW miniaturizzati di Murata soddisfano le esigenze del 5G

Murata ha reso noto l'introduzione dei nuovi componenti SAW (Surface Acoustic Wave) progettati per espletare compiti di filtraggio RF e di duplexer. Questi prodotti sono stati ideati per soddisfare la richiesta, in continuo aumento, di ridurre le dimensioni dei componenti usati nei telefoni mobili: si tratta di un'esigenza che l'avvento della telefonia 5G contribuirà ad acuire ulteriormente.

I componenti della serie SAYAV/SAYAR (per il duplexing) sono disponibili in formato 1612 (1,6x1,2mm), mentre quelli della serie SAFFW (per il filtraggio in ricezione) vengono forniti nel formato 0907

(0,9x0,7mm). Si tratta dei dispositivi più piccoli del loro genere al momento disponibili, in grado di fornire prestazioni confrontabili se non addirittura superiori, rispetto ai tradizionali filtri SAW. Questi dispositivi SAW di Murata rappresentano la soluzione ideale per tutti i progettisti impegnati nello sviluppo di prodotti consumer e/o industriali che integrano funzionalità LTE/Wi-Fi.

www.murata.com



Rohde & Schwarz annuncia la prima soluzione di test per IEEE 802.3cg 10BASE-T1S

Rohde & Schwarz introduce il primo test di conformità 10BASE-T1S disponibile, per gli oscilloscopi R&S RTO e R&S RTP, che fornirà test completamente automatizzati conformi alla versione attuale delle interfacce IEEE 802.3cg 10BASE-T1S. Rohde & Schwarz continua a partecipare attivamente all'OPEN Alliance, sfruttando le sue competenze per guidare lo sviluppo di soluzioni di test complete e affidabili.

Come anche per altri standard Ethernet, il nuovo software di test di conformità R&S RTO-K89 e R&S RTP-K89 per 10BASE-T1S include istruzioni grafiche per guidare gli utenti attraverso le misurazioni, passo dopo passo. Entrambi, gli oscilloscopi e l'analizzatore

di rete R&S ZND vengono configurati automaticamente. I risultati del test sono documentati in un rapporto di test in formato pdf.

Con questa nuova aggiunta, Rohde & Schwarz sarà in grado di fornire un portafoglio completo per testare tutti gli standard Ethernet automotive disponibili (10BASE-T1S, 100BASE-T1, 1000BASE-T1) e per tutti gli altri bus automobilistici standard (LIN, CAN, CAN-FD, FlexRay e CXPI).

www.press.rohde-schwarz.com



Rendi semplice IoT e connettività con le board MERCURY!

**Base Board
per sistema
Mercury**



cod. BB110
€ 23,00

Mercury System è un sistema di sviluppo modulare, hardware/software, specificamente progettato per permettere lo sviluppo di applicazioni IoT e in generale orientate alla connettività. Il sistema è composto dalla Base Board (cod. BB110) che è il "cervello" di tutto il sistema e contiene l'unità logica principale (un microcontrollore) oltre ai vari bus di comunicazione ed interfacce e da un set eterogeneo di schede slave (SB) e schede modem (MB) con le quali interagisce.

LE SCHEDE DEL SISTEMA MERCURY

SCHEDA SLAVE PER
SENSORI E ATTUATORI



cod. SB810
€ 13,00

SCHEDA A ULTRASUONI
INTERFACCIABILE



cod. SB310
€ 19,00

SCHEDA
CON 2 RELÈ



cod. SB110
€ 17,00

SCHEDA MODEM
BLUETOOTH



cod. MB310
€ 14,00

SCHEDA DHT22
CON SENSORE DI
TEMPERATURA E UMIDITÀ



cod. SB330
€ 20,00

SCHEDA HSD (HIGH SIDE
DRIVER) A 4 CANALI



cod. SB140
€ 25,00

SCHEDA A INFRAROSSI
2 CANALI



cod. SB320
€ 12,00

SCHEDA PER STRIP A LED
O RING NEOPIXEL



cod. SB120
€ 14,00

SCHEDA EXPANSION
DISPLAY 16X2



cod. EB210
€ 23,00

SCHEDA PER
CONNESSIONE
PLANARE DI 2 SCHEDE



cod. EB110
€ 7,00

SCHEDA PER CONNESSIONE
DI 4 SCHEDE



cod. EB111
€ 12,00

SCHEDA CON
CONNETTIVITÀ WiFi



cod. MB210
€ 12,00

SCHEDA CON SERVO
A 6 CANALI



cod. SB130
€ 16,00

SCHEDA PER
GESTIONE ALIMENTAZIONE



cod. PB110
€ 17,00

SCHEDA CON RELÈ PER
CARICHI FINO A 10A



cod. SB111
€ 17,00

Prezzi IVA inclusa.

**FUTURA
ELETTRONICA®**

www.futurashop.it

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

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



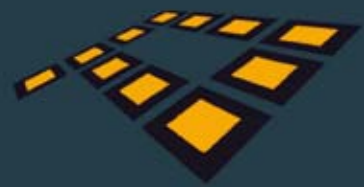
► 12 – 15 NOVEMBRE 2019 PRODUCTRONICA MONACO DI BAVIERA • GERMANIA

È in programma a Monaco di Baviera dal 12 al 15 novembre prossimi l'edizione 2019 di PRODUCTRONICA, una delle più importanti manifestazioni al mondo dedicata ai sistemi di progettazione e produzione in ambito elettronico. La fiera, che si svolge ogni due anni alternandosi con Electronica, nel 2017 ha registrato la presenza di 1.560 espositori provenienti da 43 diversi Paesi ed è stata visitata da 44.000 professionisti del settore, metà dei quali provenienti dall'estero. Tra le novità di quest'anno il padiglione Smart Maintenance dedicato alla manutenzione, sempre più importante nella digitalizzazione dei processi produttivi.

La manutenzione intelligente è necessaria per ottimizzare la produzione: il padiglione Smart Maintenance offre una panoramica completa delle applicazioni e soluzioni attuali e di quelle del futuro. Gli argomenti includono manutenzione predittiva, condition monitoring, mobile maintenance e realtà aumentata. I visitatori avranno

l'opportunità di parlare con esperti e sperimentare in pratica specifici processi di manutenzione.

La Smart Maintenance è una delle componenti del processo di "Accelerating Innovation", che è anche il motto della manifestazione, e che comprende l'E-Mobility e la Smart Factory. Anche le startup avranno uno spazio completamente dedicato alle loro attività grazie al progetto "Fast Forward" realizzato in collaborazione con la rivista Elektor. Le nuove imprese troveranno spazio nel padiglione B2, dove quelle più innovative verranno premiate con il "Fast Forward Award".



productronica

Per venire incontro alle esigenze delle aziende che fanno sempre più fatica a trovare figure professionali adeguate, PRODUCTRONICA ha deciso di puntare sull'incontro tra giovani ed aziende mettendo a loro disposizione l'area espositiva "Accelerating Talents" nella quale si svolgeranno numerose attività tra i quali Hackaton di 48 ore e dove sarà presente anche uno "Speakers Corner" nel quale le aziende presenteranno le proprie attività ed esporranno le loro offerte lavorative.

A PRODUCTRONICA non mancheranno i sistemi di progettazione e produzione dei circuiti stampati e dei prodotti connessi.

La manifestazione si svolgerà in concomitanza con SEMICON Europa 2019, la prima vetrina europea per il business e l'approfondimento in settori chiave quali flexible hybrid electronics, MedTech, automotive, imaging, design and fabless, SMART Manufacturing, power electronics e altro ancora.

www.productronica.com

FORLÌ

EXPO ELETTRONICA

Quartiere Fieristico - Forlì
Organizzazione: Blu Nautilus

Telefono: 054153295
www.expoelettronica.it
info@blunautilus.it
07 e 08 Dicembre 2019

REGGIO EMILIA

**MONDO ELETTRONICA
REGGIO EMILIA**

Quartiere Fieristico Reggio Emilia
Via Filangeri

Organizzazione: Expo Fiere Srl
Telefono: 054527548

www.mondoelettronica.net
info@mondoelettronica.net
07 e 08 Dicembre 2019

PIACENZA

MONDO ELETTRONICA

Piacenzaexpo - Località Le Mose S.S.
per Cremona

Organizzazione: Expo Fiere Srl
Telefono: 054527548

www.mondoelettronica.net
info@expofiere.net
14 e 15 Dicembre 2019

NOVEGRO (MI)

**RADIANT ELECTRONIC
CHRISTMAS DAY**

Parco Esposizioni di Novegro (MI)
Organizzazione: Comis Srl

Telefono: 027562711

www.parcospozizioninovegro.it
radiant@parcospozizioninovegro.it
14 e 15 Dicembre 2019

GENOVA

GIZMARK

Quartiere Fieristico - Genova
Organizzazione: Studio Fulcro

Telefono: 010561111

www.studiofulcro.it
info@studio-fulcro.it
14 e 15 Dicembre 2019

L'elenco aggiornato di tutte le Mostre
Mercato del 2019 è disponibile sul sito

[www.elettronicaonline.it/
calendario-mostre-mercato](http://www.elettronicaonline.it/calendario-mostre-mercato)

L'elenco completo dei più importanti eventi
nazionali e internazionali di elettronica,
sicurezza e fonti rinnovabili è disponibile sul sito
[www.elettronicaonline.it/blog/category/eventi/
eventi_e_manifestazioni](http://www.elettronicaonline.it/blog/category/eventi/eventi_e_manifestazioni)

SEMINARIO IA, L'INTELLIGENZA ARTIFICIALE NEL MONDO REALE

Il mercato dell'IA accelera, e lo stesso vale per la richiesta di soluzioni rapide, dalla definizione all'impiego. Per raccogliere ispirazione e suggerimenti per queste soluzioni, Arrow ha creato l'AI Experience Tour. Questo evento, della durata di un giorno, presenterà le intuizioni degli esperti di IA nei settori di crescita chiave e le ultime novità nel campo delle soluzioni end-to-end per accelerare il percorso dal progetto all'impiego. Partecipando ad AI Experience Tour i visitatori potranno:

- Incontrare le compagnie leader in intelligenza artificiale e apprendimento automatico
- Esplorare presentazioni e zone di innovazione di applicazioni e demo IA
- Ottenere informazioni sul mercato dell'Intelligenza artificiale
- Ascoltare le idee chiave di alto profilo

e le presentazioni sui palchi principali

- Scoprire le offerte end-to-end di Arrow

In Italia è prevista una sola data il 21 Novembre, a Milano presso lo Sheraton Hotel Milan San Siro, Via Caldera, 21, 20153 Milano Italia.

www.arrow.com


NIDAYS EUROPA, IN GERMANIA LA PRIMA EDIZIONE

National Instruments esporta in Europa e Cina il format di NIWeek, l'evento che si svolge annualmente nella sede storica di Austin.

Prendono il via così NIDays Europe che si terrà quest'anno il 20 e 21 novembre 2019 a Monaco, in Germania, mentre Shanghai, Cina, ospiterà NIDays Asia il 14 e 15 novembre. *"Creando eventi unici in Europa e in Asia, offriamo ai partecipanti l'opportunità di interagire con colleghi, partner e leader del settore internazionale"*, afferma la società.

I prossimi NIDays ospiteranno più di 100 sessioni tecniche, 25 corsi di formazione e certificazione, oltre 100 demo proposte dalla stessa NI

e da circa 50 aziende partner.

L'agenda di NIDays prevede due giorni di sessioni tecniche con topic nei settori dei semiconduttori, automobilistico, accademico, IOT, 5G e software. Inoltre, i partecipanti potranno ascoltare stimolanti relatori ed essere in grado di connettersi con colleghi e personale NI.

La keynote di apertura sarà affidata a Mark Gallagher, manager di F1 che in passato ha collaborato con i team Jordan, Red Bull e Ford Cosworth. Più di recente Gallagher ha collaborato con la FIA per sviluppare le nuove normative sui motori ibridi benzina-elettrici introdotte in Formula 1 nel 2014.

www.ni.com


IOTINGS ROME

IOTINGS Rome è l'edizione autunnale di IOTINGS, l'evento internazionale, organizzato da Innovability, totalmente dedicato all'IoT e alla emergente Digital Transformation of Things che si svolge in primavera a Milano.

La seconda edizione romana che si è svolta il 21 e il 22 novembre 2018 ha visto oltre un migliaio di partecipanti che hanno incontrato nei convegni, nell'area expo e nei business meetings 59 aziende tra sponsor, espositori e partner provenienti da tutta Europa. Un pubblico eterogeneo internazionale ha partecipato alle 13 sessioni di conferenza per ascoltare i 92 speaker coinvolti e apprezzare la qualità e l'aggiornamento degli argomenti proposti nelle varie sessioni della manifestazione.

In un mercato come quello dell'IoT che offre continuamente novità e nuove opportunità di business era necessario affiancare ad una edizione primaverile anche una edizione autunnale, dando a ciascuna una specializzazione in ambiti applicativi e target di utenti. In particolare IOTINGS Rome ha una forte focalizzazione sulle applicazioni IoT per le grandi reti di energia, mobilità,

ROME IOTINGS | 27-28 NOV 2019

telecomunicazioni, le infrastrutture critiche, il monitoraggio, la protezione e la sicurezza del territorio. Inoltre IOTINGS Rome è il primo evento italiano su IoT che presenta come i Geographic information system (GIS) sono sempre più importanti e strategici nello sviluppo e l'erogazione

di applicazioni e dei servizi IoT.

A Roma EUR presso lo Spazio Novecento, Piazza Guglielmo Marconi 26/b – 26/d il 27 e 28 Novembre 2019.

www.iotingsrome.com



SEMINARIO SULLE TECNOLOGIE SiC E GaN

Alla scoperta, insieme ad Arrow, delle proposte dei maggiori produttori di tecnologie SiC e GaN che rivoluzioneranno tutto il mondo dell'elettronica, dall'Industria all'Automotive.

SiC è una tecnologia già presente da anni nei Diodi negli Alimentatori (PFC) e nei sistemi fotovoltaici (PV)/ Inverter ma il processo di migrazione tecnologica con la sostituzione dei classici Transistor e Mosfet – soprattutto nel range di tensione 1200V – cambierà il modo di gestire la potenza nei progetti elettronici. GaN offre, invece, un reale guadagno di velocità – soprattutto

nella media tensione dove sono presenti i MOSFET da anni (650V) – ma anche un notevole vantaggio nelle applicazioni RF per future applicazioni, come LIDAR o tutte quelle in cui sarà richiesto di aumentare la velocità.

L'evento si articola in due sessioni parallele tenute dai maggiori produttori al mondo, oltre a una vetrina espositiva con prototipi realizzati con le tecnologie SiC e GaN. L'evento si terrà il 14 Novembre 2019 a Lanzise (VR) presso l'Hotel Parchi del Garda, Via Brusà 16. La partecipazione è gratuita previa registrazione al seguente link: <https://bit.ly/2SYJnWG>

SiC – High Power Applications

- 08:00 – 08:30 **Welcome, Registration & Showcase**
- 08:30 – 08:40 **Introduction**
- 08:40 – 09:40 **Cree Wolfspeed**
- 09:40 – 10:40 **Littelfuse**
- 10:40 – 11:00 **Coffee Break & Showcase**

- 11:00 – 11:30 **ON Semiconductor**
- 11:30 – 12:00 **Infineon**
- 12:00 – 13:30 **Lunch & Showcase**
- 13:30 – 14:30 **Kemet**
- 14:30 – 15:30 **Microsemi**
- 15:30 – 16:30 **Coffee Break & Showcase**

GaN – Low Power Applications

- 08:00 – 08:30 **Welcome, Registration & Showcase**
- 08:30 – 08:40 **Introduction**
- 08:40 – 09:10 **Infineon**
- 09:10 – 10:10 **GaN Systems**
- 10:10 – 10:40 **Showcase**
- 10:40 – 11:00 **Coffee Break & Showcase**
- 11:00 – 12:00 **Silabs**
- 12:00 – 13:30 **Lunch & Showcase**
- 13:30 – 14:30 **TDK**
- 14:30 – 15:30 **Showcase**
- 15:30 – 16:30 **Coffee Break & Showcase**

www.arrow.com





ABBONAMENTO
ANNUALE



Mese dopo mese realizza i **Progetti** descritti sulla rivista, rimani aggiornato con i nostri **Corsi**, approfondisci la conoscenza delle tecniche e dei componenti più avanzati con i nostri **Tutorial** e le nostre **News**.

10
NUMERI

A SOLI

€45

ANZICHÉ € 60,00

COME ABBONARSI

→ on-line

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

→ mail

scrivendo a
abbonati@elettronica.in.it
con i dati riportati nel coupon a lato.

→ posta

compilando il modulo di abbonamento riportato a lato e inviandolo al seguente indirizzo:

FUTURA GROUP srl
Via Adige 11
21013 Gallarate (VA)

→ telefono

telefonando a
+39-0331-752668

Visita: www.elettronica.in.it/abbonamenti

VANTAGGI PER GLI ABBONATI

**RISPARMIA FINO AL 25%
SUL PREZZO DI COPERTINA**

**10% DI SCONTO SUI TUOI ACQUISTI CON
DISCOUNT CARD FUTURA ELETTRONICA**

**1 VOLUME A SCELTA DELLA COLLANA
"L'ELETTRONICA PER TUTTI"**

**RICEVI LA RIVISTA A CASA TUA
COMODAMENTE E VELOCEMENTE**

La possibilità
di **leggere**
gratuitamente
la versione digitale
di Elettronica In



Si desidero abbonarmi per un
anno alla rivista *Elettronica In*.
Riceverò i seguenti omaggi:



DISCOUNT CARD Futura Elettronica

**UNO, A SCELTA, TRA QUESTI VOLUMI
della collana "L'Elettronica per tutti".**



- ☐ **Entra nel mondo di Raspberry Pi 3+**
- ☐ **Fishino - Arduino e l'Internet delle Cose in un'unica innovativa scheda**
- ☐ **Randa - Raspberry Pi + Arduino**
- ☐ **ARDUINO UNO - Programmazione avanzata e Librerie di sistema**
- ☐ **Impara l'elettronica sperimentando**
- ☐ **Primi passi in elettronica**
- ☐ **Arduino e le tecniche di programmazione dei microcontrollori ATMEL**
- ☐ **Raspberry Pi - il mio primo Linux Embedded**
- ☐ **Alla scoperta di Arduino YÚN**
- ☐ **Programmiamo con i PIC**
- ☐ **L'ABC di Arduino**
- ☐ **Stampiamo in 3D**

Resto in attesa del primo numero.

Cognome:

Nome:

Via:

N°:

CAP:

Città:

Provincia:

Telefono:

e-mail:

Data:

Firma:

Resto in attesa di vostre disposizioni per il pagamento.
Ai sensi dell'art. 13 del D.Lgs. 196/2003, in tema di protezione dei dati personali, FUTURA GROUP srl informa che i dati liberamente forniti attraverso la compilazione del presente modulo saranno trattati e conservati in conformità a tutte le normative vigenti. I dati suddetti saranno trattati su supporto cartaceo e informatico, mediante sistemi di protezione atti alla tutela della riservatezza, per finalità connesse all'esecuzione dei rapporti contrattuali instaurati. Inviare in busta chiusa a:



FUTURA GROUP srl
Via Adige 11 • 21013 Gallarate (VA)
Tel. +39-0331-752668

KIT PER RASPBERRY Pi 3 A+

Il kit contiene: board Raspberry Pi 3 modello A+ e relativo contenitore in plexiglass trasparente, SD card da 16GB con applicazione NOOBS, cavo HDMI 2.0 high speed da 0,7 metri, alimentatore switching ultra compatto da 5 V - 2,5 A con connettore micro USB.



€ 59,⁹⁰

Cod. RASPKITV8

STARTER KIT PER RASPBERRY PI ZERO W

Il kit contiene: board Raspberry Pi Zero W e relativo contenitore realizzato in materiale plastico, cavo USB/microUSB con interruttore, cavo HDMI, connettore passo 2 X 2.54 e adattatore HDMI/miniHDMI.



€ 39,⁹⁰

Cod. RASPKITV9

I MIGLIORI KIT PER RASPBERRY Pi LI TROVI SOLO DA NOI!



KIT PER RASPBERRY Pi 3 B+

Il kit contiene: board Raspberry pi 3 modello B+, alimentatore switching ultra compatto con uscita 5 V - 2,5 A con connettore micro USB, cavo HDMI 2.0 high speed, contenitore per Raspberry Pi 3 B+, set di dissipatori.



€ 59,⁹⁰

Cod. RASPKITV7

KIT PER RASPBERRY Pi 3 B

Il kit contiene: board Raspberry Pi 3 modello B, alimentatore switching ultra compatto da 5 V - 2,5 A con connettore micro USB, cavo HDMI 2.0 high speed, cavo FTP CAT5E 2xRJ45, SD card HC da 8GB con applicazione NOOBS, contenitore trasparente per Raspberry Pi 3 tipo B.



€ 69,⁹⁰

Cod. RASPKITV6

DOMOTICA PER TUTTI



di GUIDO OTTAVIANI

Proseguiamo l'analisi delle periferiche a bordo della scheda CM3-home conoscendo i bus 1-wire e I²C, le periferiche di input e quelle di output.

Torniamo ad occuparci della CM3-home, la scheda con core Raspberry Pi che rende semplice lo sviluppo delle applicazioni domotiche: dopo i bus pensati per reti complesse e di notevole estensione, vediamo quelli più semplici, partendo dal 1-Wire; questo nasce per gestire sensori piccoli ed economici, dove ci siano da misurare parametri ambientali come temperatura, umidità, pressione, ecc. 1-Wire è un bus di comunicazione tra dispositivi progettato dalla Dallas Semiconductor ma funziona similmente all'I²C ma con velocità di comunicazione minore e su distanze maggiori. Il nome deriva dal fatto che è possibile far passare alimentazione e dati su un singolo filo usando la cosiddetta "alimentazione parassita", anche se nella pratica il sistema più affidabile è

il classico a tre fili: alimentazione, massa e dati (canale bidirezionale), soprattutto se il bus prevede diversi device collegati. Usando l'alimentazione parassita, questa viene prelevata dal bus dati e un condensatore da 800 pF deve essere presente nel dispositivo per accumulare l'energia necessaria al funzionamento durante la trasmissione dei dati. Il protocollo è master-slave. L'applicazione tipica è nei sensori di temperatura digitali e negli strumenti per meteorologia. Nella CM3-Home è presente un bus 1-Wire, la cui linea dati è interfacciata ai GPIO tramite un buffer dedicato che assicura una comunicazione più robusta senza bisogno di resistori di pull-up.

Per testare il 1-Wire utilizzeremo il binding Exec di OpenHAB (<https://www.openhab.org/addons/bindings/exec/>) e un DS18B20 (Fig. 1). Se non è stato già fatto durante l'installazione, aggiungete la configurazione necessaria nel file **/boot/config.txt** per indirizzare il driver Linux alla lettura sul corretto GPIO:

```
# Enable the 1-wire bus
dtoverlay=w1-gpio,gpiopin=16
```

Al riavvio i sensori connessi saranno automaticamente rilevati dal kernel e ci sarà una directory per ognuno di essi in **/sys/bus/w1/devices**. Ad esempio:

```
$ ls /sys/bus/w1/devices/
28-000006c423d3
```

indica che è stato rilevato un sensore con id=28-000006c423d3. Il file contiene il valore di temperatura letto:

```
$ cat /sys/bus/w1/devices/28-000006c423d3/w1_slave
a4 01 4b 46 7f ff 0c 10 da : crc=da YES
a4 01 4b 46 7f ff 0c 10 da t=26250
```

Per leggere tale valore creiamo lo script **scripts/onewiretemp.sh** così:

```
#!/bin/bash
wert=`cat /sys/bus/w1/devices/$1/w1_slave | tail -n1 | cut -d '=' -f2`
wert2=`echo "scale=3; $wert/1000" | bc`
echo $wert2
```

e gli assegniamo i diritti di esecuzione:

```
sudo chmod +x onewiretemp.sh
```

Quindi, per verificare il corretto funzionamento del tutto andiamo a impartire il comando:

```
./onewiretemp.sh 28-000006c423d3
21.8
```

Questo script può ora essere inserito in un *thing* del binding Exec tramite la *Paper UI*:

```
Configuration > Things > Edit > Command >
/etc/openhab2/scripts/onewiretemp.sh 28-000006c423d3
```

con i parametri predefiniti sarà eseguito ogni 30s aggiornando il valore di temperatura con questa cadenza. Il valore di output dell'oggetto *thing* è usato in **items/onewire.items** per gestire nell'ambiente OpenHAB la variabile di temperatura misurata:

```
String Temp1Wire "Temperature [%s °C]" {channel="exec :command:onewire:output"}
```

Per visualizzarla nell'App mobile aggiungere a **sitemaps/cm3home.sitemaps**:

```
Frame label="1-wire"
{
```

Fig. 1
Esempio di collegamento di un sensore di temperatura 1-Wire DS18B20.

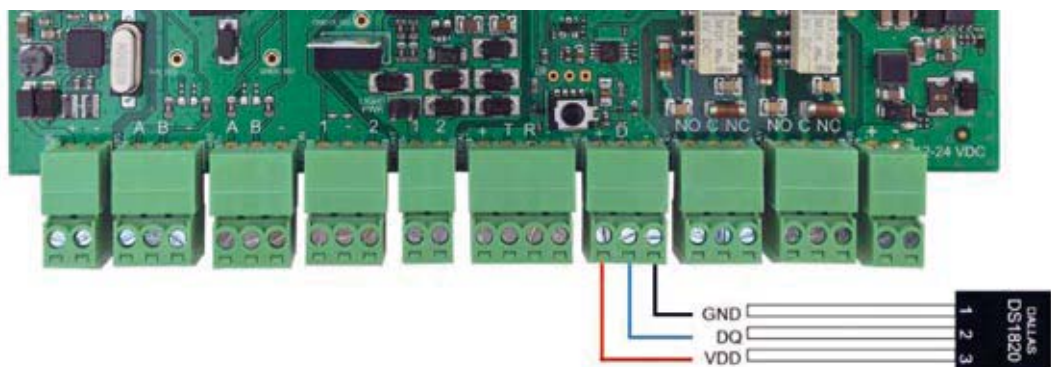




Fig. 2
Connettori
Grove.

↓ Listato 1

```
cat remote.lircd.conf
```

```
# Please take the time to finish this file as described in
# https://sourceforge.net/p/lirc-remotes/wiki/Checklist/
# and make it available to others by sending it to
# lirc@bartelmus.de
#
# This config file was automatically generated
# using lirc-0.9.4c(default) on Sat Mar 24 23:05:34 2018
# Command line used: -d /dev/lirc0 /home/openhabian/lircd.conf
# Kernel version (uname -r): 4.9.80-v7+
#
# Remote name (as of config file): remote
# Brand of remote device, the thing you hold in your hand:
# Remote device model nr:
# Remote device info url:
# Does remote device has a bundled capture device e. g., a
#   usb dongle? :
# For bundled USB devices: usb vendor id, product id
#   and device string (use dmesg or lsusb):
# Type of device controlled
#   (TV, VCR, Audio, DVD, Satellite, Cable, HTPC, ...) :
# Device(s) controlled by this remote:
begin remote
    name IR
    bits          32
    flags SPACE_ENC|CONST_LENGTH
    eps           30
    aeps          100

    header        9098 4447
    one           624 1636
    zero          624 508
    ptrail        625
    repeat        9102 2189
    gap           107732
    toggle_bit_mask 0x0
    frequency     38000

    begin codes
    KEY_1         0x00FFA25D
    KEY_2         0x00FF629D
    KEY_3         0x00FFE21D
    KEY_4         0x00FF22DD
    KEY_5         0x00FF02FD
    KEY_6         0x00FFC23D
    KEY_7         0x00FFE01F
    KEY_8         0x00FFA857
    KEY_9         0x00FF906F
    KEY_0         0x00FF6897
    end codes
end remote
```

```
Text label="Temperature" icon=temperature
{
    Text item=Temp1Wire
}
}
```

Il valore ritornato dallo script è di tipo stringa. Per utilizzarlo, ad esempio, in una **rule** occorre convertirlo in numero. Creiamo quindi una nuova variabile in **items/newwire.items**:

```
String Temp1Wire "1-Wire Temp [%s °C]" {channel="exec:command:1wire:output"}
Number Temp1WireValue "1-Wire Temp [%1f °C]"
```

E applichiamo una regola in **rules/temp.rules**:

```
rule "StringToInt"
when
    Item Temp1Wire changed
then
    var Temp = Temp1Wire.state.toString
    val TempFloat = Float::parseFloat(Temp)
    Temp1WireValue.postUpdate(TempFloat)
end
```

Quando la stringa cambia, la regola viene eseguita e il valore convertito in una variabile numerica che può essere usata in un widget, in un'altra regola per eseguire determinate azioni al passaggio di una determinata soglia o altro.

INTERFACCIA I²C

Dopo il bus 1-Wire, vediamo un altro molto diffuso nei collegamenti a breve distanza: l'I²C. Dovendo scegliere un tipo di connettore nella CM3-Home, non esistendo uno standard abbiamo pensato di usare il connettore del sistema Grove (**Fig. 2**) che è quello che ha il maggior numero di dispositivi e sistemi di collegamento disponibili. Grove è un sistema modulare di connessione standard

per la prototipazione realizzato dalla Seeed studio (http://wiki.seeedstudio.com/Grove_System).

È molto usato nell'ambiente Raspberry Pi e permette di avere numerosi dispositivi collegabili direttamente.

Sulla CM3-Home sono disponibili due porte Grove, una gestita direttamente dal modulo CM3 e l'altra dal Cortex M0+ montato sul modulo YARM, progettate per essere usate con lo standard I²C o come GPIO.

In **Fig. 3** si può vedere lo schema relativo a uno di questi ingressi: si nota che, come tutti gli altri, siano protetti contro le ESD (ElectroStatic Discharge) che potrebbero danneggiare le porte dei microcontrollori.

Le porte del modulo Raspberry collegate sono:

- GPIO 44 – SDA;
- GPIO 45 – SCL.

Per usare questi GPIO come bus I²C occorre aggiungere le configurazioni necessarie al file **/boot/config.txt**:

```
#Set the I2C1 interface on GPIO44 and 45
dtparam=i2c_arm=on
dtparam=i2c_arm_baudrate=400000
dtoverlay=i2c1-bcm2708,sda1_pin=44,scl1_pin=45,pin_
func=6
```

Per verificarne il funzionamento si può installare il package **i2c-tools**:

```
apt-get update
apt-get install i2c-tools
i2cdetect -y 1
```

INTERFACCE UNIDIREZIONALI: PORTA IR

Finora abbiamo visto i bus dati, ma la CM3-Home è equipaggiata anche con interfacce di solo input e solo output. Analizziamole partendo da quelle di ingresso, prima fra tutte quella IR. La CM3-Home è dotata di un ricevitore IR a 38 kHz collegato al GPIO 20 (**Fig. 4**); in alternativa è possibile saldare un sensore differente sui pad preposti. Il sensore è utilizzato tramite il driver standard LIRC di Linux. Per abilitarlo devono essere presenti le seguenti istruzioni nel file **/boot/config.txt**:

```
# Enable the IR Linux driver
dtoverlay=lirc-rpi,gpio_in_pin=20
```

poi occorre installare il driver

```
sudo apt-get install lirc
```

il file **/etc/lirc/lirc_options.conf** deve essere editato come segue:

```
driver = default
device = /dev/lirc0
```

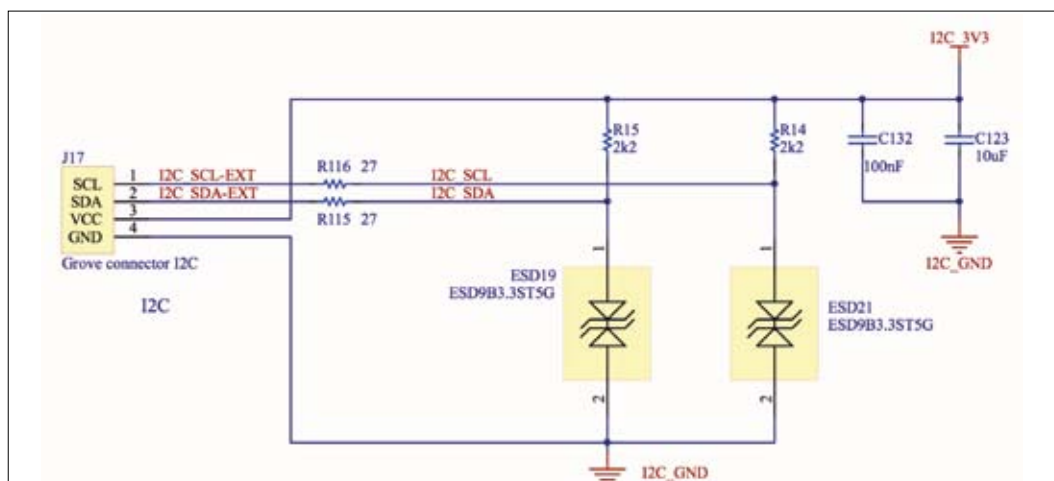
Dopo il reboot si può controllare il corretto funzionamento puntando un telecomando a 38 kHz:

```
sudo systemctl stop lircd.service
sudo systemctl stop lircd.socket
mode2 -d /dev/lirc0
```

Se tutto è a posto, alla pressione di un tasto qualsiasi sarà visualizzato qualcosa del genere:

```
Using driver default on device /dev/lirc0
```

Fig. 3
Schema della parte di protezione degli ingressi Grove.



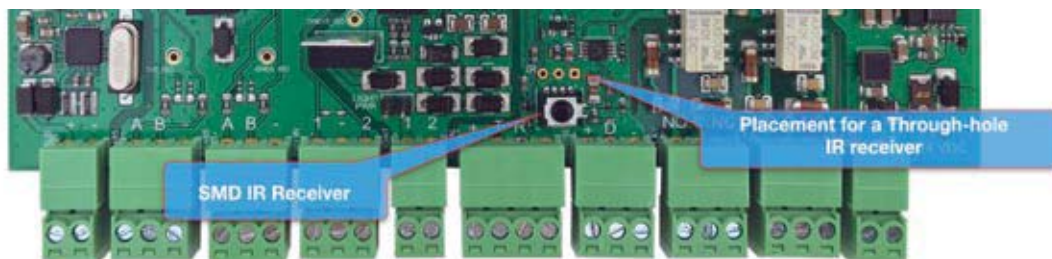


Fig. 4
Sensori IR
sulla
CM3-Home.

```
Trying device: /dev/lirc0
Using device: /dev/lirc0
space 10743268
pulse 243
space 19204
pulse 100
space 2521370
pulse 9126
space 4441
pulse 625
space 506
pulse 623
space 513
pulse 622
space 510
```

Per usare il telecomando occorre memorizzare i tasti all'interno dell'applicazione; allo scopo si usa il programma **irrecord**. La sequenza di operazioni necessarie e l'output di **irrecord** variano in funzione della versione corrente, perciò non possiamo darvi istruzioni dettagliate. È consigliabile seguire attentamente le istruzioni proposte durante il processo di apprendimento dei diversi tasti del telecomando.

```
irrecord -d /dev/lirc0 ~/lircd.conf
```

A conclusione dell'operazione sarà disponibile un file con il nome che abbiamo assegnato in fase di apprendimento: ad esempio quello nel **Listato 1**. Il file generato va spostato nel path corretto:

```
sudo cp remote.lircd.conf /etc/lirc/lircd.conf.d/
```

Dopo aver riavviato il servizio LIRC:

```
sudo systemctl restart lircd
```

si può verificare se il file config è stato riconosciuto:

```
irsend LIST "" ""
```

se tutto è a posto sarà visualizzato il nome del file di configurazione. Per provare i tasti configurati:

```
openhbian@cm3home:~$ irw
00000000ffa25d 00 KEY_1 IR
00000000ff629d 00 KEY_2 IR
00000000ffe21d 00 KEY_3 IR
00000000ff22dd 00 KEY_4 IR
00000000ff02fd 00 KEY_5 IR
00000000ffc23d 00 KEY_6 IR
00000000ffe01f 00 KEY_7 IR
00000000ffa857 00 KEY_8 IR
00000000ff906f 00 KEY_9 IR
00000000ff6897 00 KEY_0 IR
```

A questo punto il sistema hardware e software, è pronto per essere usato nelle nostre applicazioni. A scopo dimostrativo useremo LIRC collegandolo a dei **topic** MQTT, così da esplorare le altre possibilità di OpenHAB di collegarsi ai protocolli più diffusi ed interfacciarsi quindi anche a dispositivi non disponibili direttamente come binding.

Dopo aver installato il binding MQTT (Binding MQTT <https://www.openhab.org/addons/bindings/mqtt1/>), occorre configurare il servizio in **service/mqtt.cfg** per attivare il broker locale:

```
broker1.url=tcp://localhost:1883
broker1.clientId=openhab2
```

Occorre definire anche il **binding event bus**, se si desidera pubblicare e sottoscrivere i topic in modo che siano gestiti direttamente dall'**event bus** di OpenHAB, attivando le notifiche in tempo reale (**service/mqtt-eventbus**):

```
broker=broker1
statePublishTopic=openhab/out/${item}/state
commandPublishTopic=openhab/out/${item}/command
stateSubscribeTopic=openhab/in/${item}/state
commandSubscribeTopic=openhab/in/${item}/command
```

In tal modo avremo due topic principali da usare

Listato 2

```
begin
  button = KEY_1
  prog = irexec
  config = /usr/bin/mosquitto_pub -h localhost -t openhab/out/ir/command -m key_1
  repeat = 0
end
begin
  button = KEY_2
  prog = irexec
  config = /usr/bin/mosquitto_pub -h localhost -t openhab/out/ir/command -m key_2
  repeat = 0
end
begin
  button = KEY_3
  prog = irexec
  config = /usr/bin/mosquitto_pub -h localhost -t openhab/out/ir/command -m key_3
  repeat = 0
end
begin
  button = KEY_4
  prog = irexec
  config = /usr/bin/mosquitto_pub -h localhost -t openhab/out/ir/command -m key_4 repeat = 0
end
begin
  button = KEY_5
  prog = irexec
  config = /usr/bin/mosquitto_pub -h localhost -t openhab/out/ir/command -m key_5 repeat = 0
end
begin
  button = KEY_6
  prog = irexec
  config = /usr/bin/mosquitto_pub -h localhost -t openhab/out/ir/command -m key_6 repeat = 0
end
```

come radice di partenza per tutti i messaggi:

```
openhab/in/#
openhab/out/#
```

Il corretto funzionamento si può verificare tramite programmi di monitor MQTT come mqtt-spy o con MQTTlens. Sottoscrivendo i due topic principali a livello radice si vedranno tutti gli eventi di OpenHAB pubblicati anche come messaggi MQTT. Istruiamo LIRC per eseguire un'azione alla pressione di un tasto del telecomando modificando **/etc/**

lirc/lircrc (Listato 2). In questo modo alla pressione di un tasto sarà pubblicato un messaggio nel topic *openhab/out/ir/command* che indicherà il tasto premuto. Deve essere avviato *irexec* come demone allo startup del sistema e questo si effettua aggiungendo la riga seguente:

```
/usr/bin/irexec -d &
```

al file **/etc/rc.local** oppure con gli equivalenti sistemi di init tipici di Linux.

Configurando gli *item* su **items/ir.items** potremo

Listato 3

```
Switch Ir_0 "IR0" {mqtt= "<[broker1:openhab/out/ir/command:command:ON:key_1], <[broker1:openhab/out/ir/command:command:OFF:key_4]", autoupdate="false"}

Switch Ir_1 "IR1" {mqtt= "<[broker1:openhab/out/ir/command:command:ON:key_2], <[broker1:openhab/out/ir/command:command:OFF:key_5]", autoupdate="false"}

Switch Ir_2 "IR2" {mqtt= "<[broker1:openhab/out/ir/command:command:ON:key_3], <[broker1:openhab/out/ir/command:command:OFF:key_6]", autoupdate="false"}
```



sottoscrivere il topic pubblicato da LIRC associando i messaggi alle azioni (**Listato 3**); come esempio utilizziamo questi item per accendere il LED RGB alla pressione dei tasti del telecomando (**Listato 4**).

INTERFACCIA MIPI PER CAMERA MODULE

Proseguiamo con l'analisi delle interfacce di solo input con la videocamera, che si interfaccia tramite connessione MIPI (lo standard per l'interfacciamento di dispositivi nelle apparecchiature mobile). Sulla CM3-Home è presente un'interfaccia MIPI identica a quella della Raspberry Pi3, posta vicino al connettore del modulo CM3 (**Fig. 5**).

Questa posizione è stata scelta per fare in modo che la camera sia posizionabile facilmente sul pannello frontale del contenitore da barra DIN. Inserito il cavo flat nel connettore con la banda di colore blu orientata verso i relé, dovete lanciare:

```
sudo raspi-config
```

e abilitare la camera usando le opzioni:

```
5 Interfacing Options  Configure connections to
peripherals
P1 Camera  Enable/Disable connection to the Ra-
spberry Pi Camera
```

Fatto ciò, riavviate la CM3-Home. Per ottenere un video streaming dalla CM3-Home in modo semplice tramite Python, si può usare FLASK (troverete un tutorial sull'uso per il video streaming su <https://blog.miguelgrinberg.com/post/flask-video-streaming-revisited>). Per un uso rapido è sufficiente installare Flask con il comando:

```
sudo apt-get update sudo apt-get install python-
flask
```

quindi installare il supporto per la PI Camera:

```
sudo apt-get install python-picamera
```

clonare il repository flask-video-streaming con Git:

```
git clone https://github.com/miguelgrinberg/flask-
video-streaming.git
```

e, infine, lanciare il programma di streaming:

```
cd flask-video-streaming
CAMERA=pi python app.py
```

Listato 4

```
rule "IR_0-ON"
when
    Item Ir_0 received command ON
then
    sendCommand(LedR, OFF)
end
rule "IR_0-OFF"
when
    Item Ir_0 received command OFF
then
    sendCommand(LedR, ON)
end
rule "IR_1-ON"
when
    Item Ir_1 received command ON
then
    sendCommand(LedG, OFF)
end
rule "IR_1-OFF"
when
    Item Ir_1 received command OFF
then
    sendCommand(LedG, ON)
end
rule "IR_2-ON"
when
    Item Ir_2 received command ON
then
    sendCommand(LedB, OFF)
end
rule "IR_2-OFF"
when
    Item Ir_2 received command OFF
then
    sendCommand(LedB, ON)
end
```

Lo streaming della camera è visualizzabile all'indirizzo <http://cm3home.local:5000>.

Usando questa URL all'interno di un *widget* di tipo *frame* in un *HABpanel* si può integrare il video streaming nell'ambiente OpenHAB.

PERIFERICHE DI OUTPUT: USCITA AUDIO

Passiamo adesso a un dispositivo di solo output, che è l'uscita audio che è indispensabile in un sistema di domotica completo.

Per abilitarlo occorre configurare opportunamente il file **/boot/config.txt**:

```
# Set the CM3-Home audio lines
dtoverlay=pwm-2chan,pin=40,func=4,pin2=41,func2=4
```

È utilizzata la stessa soluzione circuitale della Raspberry Pi3 (**Fig. 6**). Tramite le periferiche PWM sui GPIO 40 e 41 si ottiene un segnale digitale



Si tratta di un protocollo standard di tipo publish-subscribe definito dall'ISO che si appoggia sul TCP/IP. È molto leggero e per questo si sta diffondendo nel mondo dell'IoT, dove i requisiti di potenza di calcolo e larghezza di banda possono fare la differenza. Tutti i messaggi passano da un broker che è responsabile della distribuzione dei messaggi ai client destinatari. Il broker, privato o pubblico, se necessario, può essere dotato di risorse anche notevoli e gestire moltissimi device remoti con connessioni lente, sui quali invece gira un codice con basso impatto.

Esistono applicazioni che agiscono da broker, client e supervisione per tutte le piattaforme e librerie di software, per implementare client in tutti i linguaggi di programmazione. L'installazione di OpenHABian prevede la possibilità di installare da menu mosquitto, uno dei più diffusi broker, che comprende anche mosquitto_pub e mosquitto_sub come client a linea di comando.

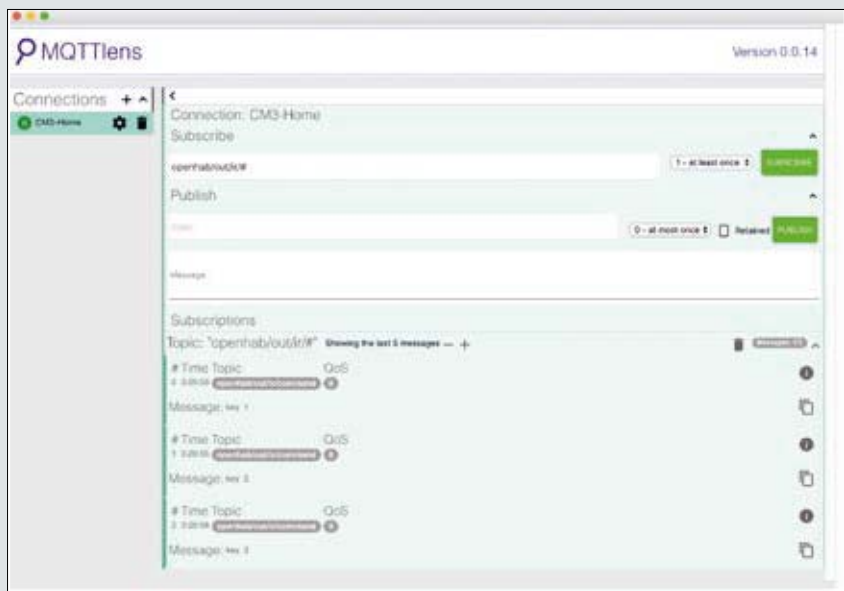


Fig. A - Monitor di topic MQTT tramite MQTTLens.

Per debuggare la propria configurazione MQTT si può usare, tra i tanti monitor disponibili, MQTTLens, l'applicazione che

si installa come estensione di Chrome che ha anche capacità di publish e subscribe di messaggi (Fig. A)

modulato e a valle si trova un buffer per abbassare l'impedenza delle uscite. Passando per un filtro passa-basso vengono soppressate le componenti spurie di alta frequenza provocate dalla digitalizzazione e rimane un segnale analogico con il contenuto del suono desiderato.

In pratica si ottiene un DAC molto economico: non si può certo considerare un audio hi-fi, perché la modulazione PWM non consente di ottenere tale qualità, però è utile per ottenere segnali di allarme, text-to-speech o altre applicazioni nelle quali la qualità audio non è determinante. Sfruttando le caratteristiche multimediali dell'ambiente java sul quale è basato OpenHAB si possono riprodurre dei file MP3 contenuti nella cartella **sounds** di

OpenHAB.

Definiamo gli *item* collegati ai suoni in **items/sounds.item**:

```
Switch AirHorn "AirHorn" (audio)
Switch ALARMBUS "ALARMBUS" (audio)
Switch alarmClock "alarmClock" (audio)
Switch barking "barking" (audio)
...
```

Aggiungiamo le regole che utilizzano questi item:

```
rule "AirHorn"
when
    Item AirHorn changed
then
    playSound("AirHorn.mp3")
end
rule "ALARMBUS"
when
    Item ALARMBUS changed
then
    playSound("ALARMBUS.mp3")
end
```

Gli item possono essere usati in una GUI o attivati

Fig. 5
Connettore MIPI
per Raspberry Pi
cam.



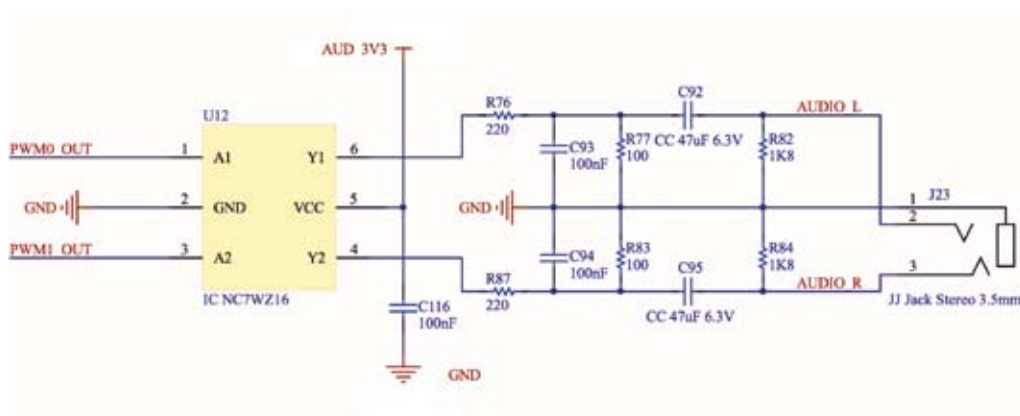


Fig. 6
Schema
elettrico
della sezione
audio.

in base a qualsiasi altro evento nell'ambiente OpenHAB. Volendo usare un player diverso da quello interno di java, magari per ottenere funzioni non disponibili su questo, si può sempre usare il solito `executeCommandLine` per eseguire un programma esterno, ad esempio il classico omxplayer:

```
sudo apt-get install omxplayer
```

Associamo ora gli item definiti a delle regole in **rules/audio.rules** (Listato 5) per riprodurre il suono voluto tramite il jack audio (Fig. 7).



Fig. 7
Uscita audio
su connettore
jack 3,5 mm.

LED RGB

Siamo arrivati alla descrizione dell'ultimo dispositivo di solo output presente a bordo della CM3-Home: si tratta del LED RGB (Fig. 8), che genera segnalazioni circa lo stato di alcune variabili relative al funzionamento del sistema operativo o di altri device integrati nell'ambiente OpenHAB. Il LED RGB è pilotato dalle seguenti linee GPIO:

- GPIO 36 Red 0=on
- GPIO 35 Green 0=on
- GPIO 34 Blue 0=on

Ogni LED è acceso quando il GPIO relativo è a livello logico basso. I LED possono essere utilizzati pilotando direttamente i livelli logici o assegnati ad eventi di sistema.

Utilizzando un DTS overlay e le funzioni del Kernel Linux è possibile definire un insieme di comportamenti standard per ciascun colore; allo scopo definiamo le seguenti funzioni;

- Led verde = Heartbeat. La velocità di lampeggio dipende dal carico della CPU;
- Led rosso = MMC0. Il rosso lampeggerà su qualsiasi accesso in scrittura alla MicroSD;

▪ Led blu = Spento.
Creiamo un file chiamato `cm3-home-leds.dts` (Listato 6) e usiamo i seguenti trigger di sistema:

```
[none] kbd-scrolllock kbd-numlock kbd-capslock kbd-kanalock kbd-shiftlock kbd-altgrlock kbd-ctrllock kbd-altlock kbd-shiftrllock kbd-shiftrlock kbd-ctrl1-lock kbd-ctrlrlock timer oneshot heartbeat backlight gpio cpu0 cpu1 cpu2 cpu3 default-on input panic mmc0
```

Compiliamo il DTS overlay:

```
sudo apt-get update sudo apt-get install device-tree-compiler sudo dtc -@ -I dts -O dtb -o /boot/overlays/cm3-home-leds.dtbo cm3-home-leds.dts
```



Fig. 8
LED RGB.

Listato 5

```
rule "AirHorn"
when
    Item AirHorn changed
then
    val results = executeCommandLine("sudo@@usr/bin/omxplayer@@-o@@local@@/
etc/openhab2/sounds/AirHorn.mp3",5000)
    logInfo("Exec",results)
end
rule "ALARMBUS"
when
    Item ALARMBUS changed
then
    val results = executeCommandLine("sudo@@usr/bin/omxplayer@@-o@@local@@/
etc/openhab2/sounds/ALARMBUS.mp3",5000)
    logInfo("Exec",results)
end
rule "alarmClock"
when
    Item alarmClock changed
then
    val results = executeCommandLine("sudo@@usr/bin/omxplayer@@-o@@local@@/
etc/openhab2/sounds/alarmClock.mp3",5000)
    logInfo("Exec",results)
end
.....
```

Listato 6

```
/dts-v1/;
/plugin/;
/ {
    compatible = "brcm,bcm2708";

    fragment@0 {
        target = <&leds>;
        __overlay__ {
            rgb_red: rgb_red {
                label = "rgb_red";
                gpios = <&gpio 36 1>;
                linux,default-trigger = "mmc0";
            };
            rgb_green: rgb_green {
                label = "rgb_green";
                gpios = <&gpio 35 1>;
                linux,default-trigger = "heartbeat";
            };
            rgb_blue: rgb_blue {
                label = "rgb_blue";
                gpios = <&gpio 34 1>;
                linux,default-trigger = "none";
            };
        };
    };
};
```

Aggiungiamo la definizione in */boot/config.txt*:

```
#CM3-Home led definition
dtoverlay=cm3-home-leds
```

Eseguire il reboot e verificare che l'overlay sia stato caricato correttamente:

```
sudo vcdbg log msg
```

Devono essere presenti le directory riportate nel **Listato 7**.

CONCLUSIONI

Chiudiamo qui questo articolo sulla board CM3-home dopo aver analizzato i bus 1-wire e I²C, oltre alle periferiche di solo input e di solo output. Il mese prossimo faremo una carrellata su ulteriori

funzionalità di questa preziosa scheda nata per semplificare la prototipazione di applicazioni nel campo della domotica. 



Cosa occorre?

I componenti utilizzati in questa presentazione sono disponibili presso Futura Elettronica. La board CM3-Home (cod. CM3HOME) è venduta a Euro 169,00 (il modulo Raspberry Pi CM3 non è compreso), la scheda WiFi (cod. RT5370N) è disponibile separatamente a Euro 9,00. I prezzi si intendono IVA compresa.

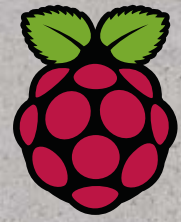
Il materiale va richiesto a:

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

Listato 7

```
$ ls -al /sys/class/leds/rgb*
lrwxrwxrwx 1 root root 0 Oct  5 12:59 /sys/class/leds/rgb_blue -> ../../
devices/platform/leds/leds/rgb_blue
lrwxrwxrwx 1 root root 0 Oct  5 12:59 /sys/class/leds/rgb_green -> ../../
devices/platform/leds/leds/rgb_green
lrwxrwxrwx 1 root root 0 Oct  5 12:59 /sys/class/leds/rgb_red -> ../../
devices/platform/leds/leds/rgb_red
```

RASPBERRY Pi e INTEL NEURAL COMPUTE STICK 2



di PIER CALDERAN

Creare applicazioni basate su reti neurali in modo efficiente con un minuscolo kit di sviluppo di Deep Learning, progettato per dar vita a una nuova generazione di dispositivi intelligenti.

Nel 2016, una startup chiamata Movidius lanciò quello che può considerarsi il primo processore di Deep Learning su chiavetta USB: il Neural Compute Stick. Basato sulla Vision Processing Unit Myriad 2, è stato il primo del suo genere. Poi la società fu acquistata da Intel e nacque così *Movidius Neural Compute Stick*. Dal 2018 è stata resa disponibile la nuova versione di tale hardware, che si chiama **Neural Compute Stick 2** (o più brevemente NCS 2) e include l'ultima unità di elaborazione della visione ad alte prestazioni di Intel: la VPU Myriad X, con più core di elaborazione e un acceleratore hardware dedicato alla "inferenza" della rete neurale profonda (spiegheremo il concetto nell'apposito riquadro). La NCS 2 (**Fig. 1**) offre prestazioni fino a otto volte superiori a quelle della precedente generazione Movidius.

NCS 2 SU RASPBERRY PI

Una volta aperta la confezione del Neural Compute Stick 2, troviamo lo stick, delle dimensioni di 73×27×14 mm, pesante appena 35 g, e un foglietto di istruzioni che indica solamente il link di Get Started: <https://developer.movidius.com/start>. Da questa pagina si possono seguire le istruzioni per installare il toolkit **OpenVINO**, che è il motore di inferenza di Intel, adatto sia per Windows che per Linux. Da notare che oltre al motore di inferenza viene installata la libreria OpenCV. In questo articolo non approfondiremo l'hardware ma ci dedicheremo a spiegare l'installazione e l'uso di NCS 2 su una scheda Raspberry Pi modello 3B+.

CONFIGURARE RASPBERRY PI

Se state utilizzando la scheda Raspberry Pi per altre mansioni, è consigliabile installare una nuova versione "pulita" di Raspbian Stretch con desktop scaricandola dal sito ufficiale: (<https://www.raspberrypi.org/downloads/raspbian>). Se non volete utilizzare schermo, tastiera e mouse, potete comodamente effettuare una connessione VNC via LAN o WLAN e il computer di lavoro. È meglio usare un alimentatore che fornisca almeno 2.000 mA perché, in genere, Raspberry Pi richiede circa 1.000 mA, il modulo della videocamera richiede 250 mA e il Neural Compute Stick 2 richiede almeno 500 mA. Dopo aver avviato e configurato Raspbian per gli usi consueti si può iniziare con l'installazione del software.

INSTALLAZIONE DI OPENVINO

Per installare il software necessario di supporto al Neural Compute Stick 2 si possono tranquillamen-

te ignorare le istruzioni del foglietto di istruzioni, perché sono mirate a computer x86_64. Intel ha fornito istruzioni alternative e ci baseremo su di esse. Per installare il toolkit OpenVINO, digitare sul terminale i seguenti comandi per scaricare il toolkit:

```
wget https://download.01.org/opencv/2019/
openvino_toolkit/1_openvino_toolkit_raspbi_p_2019.1.094.
tgz
```

Per scompattare il file compresso tgz con tar:

```
tar -zxvf 1_openvino_toolkit_raspbi_p_2019.1.094.tgz
```

Infine, per modificare lo script per l'installazione impartiamo questi comandi:

```
sed -i "s|<INSTALLDIR>|$(pwd)/inference_engine_vpu_
arm|" inference_engine_vpu_arm/bin/setupvars.sh
```

Prima di aggiungere lo script di installazione alla fine del file **.bashrc** eseguite la modifica seguente:

```
echo "source inference_engine_vpu_arm/bin/setupvars.
sh" >> .bashrc
source .bashrc
```

e apparirà a video:

```
[setupvars.sh] OpenVINO environment initialized
```

Nell'ultima versione (OpenVINO 2019R1.1 di maggio 2019), c'è un problema con la configurazione di PYTHONPATH ed è necessario aggiungere al percorso in fondo al file **.bashrc** quanto segue:

```
export PYTHONPATH="${PYTHONPATH}:/home/pi/inferen-
ce_engine_vpu_arm/python/python3.5/armv7l"
```

A questo punto, basta eseguire lo script delle regole per installare le nuove regole **udev** in modo che Raspberry Pi possa riconoscere il Neural Compute Stick 2 quando lo si collega a una porta USB:

```
sudo usermod -a -G users "$(whoami)"
sh inference_engine_vpu_arm/install_dependencies/in-
stall_NCS_udev_rules.sh
```

... e al termine apparirà:

```
Update udev rules so that the toolkit can communica-
te with your neural compute stick
```

 **Fig. 1**
Il dispositivo USB
Neural Compute
Stick 2.



INFERENZA: COS'È?

L'inferenza (in inglese *inference*) è il processo che coinvolge il ragionamento che conduce a conseguenze logiche. Etimologicamente, la parola latina *inferre* significa "portare dentro", quindi si potrebbe tradurre come "implicare".

L'inferenza è tradizionalmente divisa in deduzione e induzione, una distinzione che risale ad Aristotele e al sillogismo.

- La deduzione è inferenza che fa derivare conclusioni logiche da premesse conosciute o ritenute vere.
- L'induzione è la deduzione da premesse particolari che portano a conclusioni universali.

Un tipico esempio viene dal famoso sillogismo aristotelico:

- Tutti gli uomini sono mortali (prima premessa).
- Tutti i greci sono umani (seconda premessa).
- Tutti i greci sono mortali (conclusione).

In questo caso, si può facilmente verificare che se le premesse sono vere, allora la conclusione è necessariamente anche vera. Con la stessa logica, si può partire da premesse vere e arrivare a una conclusione falsa.

- Tutte le mele sono frutta (premessa vera)
- Tutte le banane sono frutta (premessa vera)
- Pertanto, tutte le banane sono mele (conclusione falsa)

La cosa che può sorprendere è che se una premessa è falsa e una è vera, si può arrivare a una conclusione vera:

- Tutte le persone alte sono musicisti (premessa falsa)
- John Lennon era una persona alta (premessa vera)
- Quindi, John Lennon era un musicista (conclusione vera)

MOTORE DI INFERENZA

Nel campo dell'Intelligenza Artificiale, il motore di inferenza (o inferenziale) è un componente del sistema che applica le regole logiche per dedurre nuove informazioni. La logica utilizzata da un motore di inferenza è generalmente rappresentata dalle regole IF...THEN. Il formato generale di tali regole è: IF <espressione logica> THEN <espressione logica>. I motori di inferenza precoce si concentravano principalmente sulla concatenazione diretta. Questi sistemi venivano solitamente implementati nel linguaggio di programmazione *Lisp*. *Lisp* era una piattaforma per la ricerca iniziale dell'IA in virtù della sua forte capacità di manipolare simbolicamente. Inoltre, come linguaggio interpretato offriva ambienti di sviluppo produttivo, appropriati per il debug di programmi complessi. Una conseguenza necessaria di questi benefici è che programmi *Lisp* tendevano ad essere più lenti rispetto ai linguaggi compilati del tempo, come C.

Uno dei primi e più popolari motori di *forward chaining* (concatenamento logico in avanti) fu *OPS5 (Official Production System)* creato a fine anni '70. Un'altra tecnologia molto popolare è stata il linguaggio *Prolog*, un linguaggio di programmazione logica associato all'intelligenza artificiale e alla linguistica computazionale. Il linguaggio è stato utilizzato per la dimostrazione di teoremi, sistemi esperti, riscrittura dei termini, pianificazione automatizzata, nonché il suo campo di utilizzo originario, ovvero l'elaborazione del linguaggio naturale (*Natural Language Processing*). Per gli interessati, si consigliano i riferimenti su *Prolog/Inference Engines*:

https://en.wikibooks.org/wiki/Prolog/Inference_Engines
<https://www.amzi.com/ExpertSystemsInProlog/>.

```
[install_NCS_udev_rules.sh] udev rules installed
```

Attenzione! Si consiglia di riavviare Raspberry Pi, in modo che tutte queste modifiche possano avere effetto.

Quindi collegare il Neural Compute Stick 2.

Aperto il terminale dopo il riavvio apparirà la seguente indicazione che ci rassicura che l'ambiente OpenVINO è correttamente installato e funzionante:

```
[setupvars.sh] OpenVINO environment initialized
```

COLLEGARE NCS 2 A RASPBERRY PI

Dopo il riavvio, inserite lo stick in una presa USB di Raspberry Pi. Fate un controllo con il comando **dmesg** sul terminale per vedere se tutto è a posto: dovrebbe apparire qualcosa di simile alla schermata visibile nella **Fig 2**, contenente gli ultimi messaggi relativi al dispositivo NCS 2:

```
usb 1-1.3: Product: Movidius MyriadX  
usb 1-1.3: Manufacturer: Movidius Ltd.  
usb 1-1.3: SerialNumber: 03e72485
```

Se non appaiono questi messaggi significa che lo stick non è stato riconosciuto; provate quindi a riavviare Raspberry Pi e impartire il comando **dmesg | grep Movidius** per visualizzare le indicazioni seguenti:

```
usb 1-1.3: Product: Movidius MyriadX  
usb 1-1.3: Manufacturer: Movidius Ltd.
```

TEST DEL PRIMO MODELLO DI APPRENDIMENTO AUTOMATICO

L'esperienza immediata con Neural Compute Stick 2 è basata su C++, pertanto manca uno strumento necessario per poterla fare. Allo scopo, prima di tutto dobbiamo installare **cmake** con il comando seguente:



 **Fig. 2**
Esito del comando
dmesg.

```

pi@raspberrypi: ~
File Modifica Schede Aiuto
12.594218] Bluetooth: L2CAP socket layer initialized
12.594261] Bluetooth: SCO socket layer initialized
12.605640] Bluetooth: HCI UART driver ver 2.3
12.605654] Bluetooth: HCI UART protocol H4 registered
12.605658] Bluetooth: HCI UART protocol Three-wire (H5) registered
12.605837] Bluetooth: HCI UART protocol Broadcom registered
13.048688] Bluetooth: BNEP (Ethernet Emulation) ver 1.3
13.048699] Bluetooth: BNEP filters: protocol multicast
13.048620] Bluetooth: BNEP socket layer initialized
13.187741] Bluetooth: RFCOMM TTY layer initialized
13.187781] Bluetooth: RFCOMM socket layer initialized
13.187802] Bluetooth: RFCOMM ver 1.11
57.036352] wl_master_driver wl_bus_master1: Attaching one wire slave 00.8000
00000000 crc 8c
57.041336] wl_master_driver wl_bus_master1: Family 0 for 00.800000000000.8c
is not registered.
67.951533] usb 1-1.3: new high-speed USB device number 5 using dwc_otg
68.082161] usb 1-1.3: New USB device found, idVendor=03e7, idProduct=2485
68.082188] usb 1-1.3: New USB device strings: Mfr=1, Product=2, SerialNumber
=3
68.082198] usb 1-1.3: Product: Movidius MyriadX
68.082206] usb 1-1.3: Manufacturer: Movidius Ltd.
68.082215] usb 1-1.3: SerialNumber: 03e72485
pi@raspberrypi: ~ $

```

```
sudo apt-get install cmake
```

Fatto questo, possiamo compilare una delle demo di *Face Detection* pre-addestrate. Per prima cosa, bisogna compilare l'esempio entrando nella directory corrispondente con il comando:

```
cd inference_engine_vpu_arm/deployment_tools/inference_engine/samples.
```

A questo punto occorre creare una cartella build:

```
mkdir build
```

ed entrare nella cartella con:

```
cd build
```

Ora basta compilare l'esempio **object_detection_sample_ssd** con il comando seguente:

```
cmake .. -DCMAKE_BUILD_TYPE=Release -DCMAKE_CXX_FLAGS="-march=armv7-a"
make -j2 object_detection_sample_ssd
```

La compilazione impiega circa una decina di minuti e, se non insorgono problemi, dovrebbe apparire sul terminale:

```
[100%] Built target object_detection_sample_ssd
```

A differenza della distribuzione Intel, la versione del toolkit per Raspberry Pi manca del modello e del file XML associato con il modello.

Prima di poter eseguire il modello, è necessario scaricare entrambi i file dal repository Intel, imparando, allo scopo, i comandi di seguito riportati:

```
wget --no-check-certificate https://download.01.org/
openvinotoolkit/2018_R5/open_model_zoo/face-detection-adas-0001/FP16/face-detection-adas-0001.bin
```

```
wget --no-check-certificate https://download.01.org/
openvinotoolkit/2018_R5/open_model_zoo/face-detection-adas-0001/FP16/face-detection-adas-0001.xml
```

Avremo anche bisogno di un'immagine per eseguire il test di rilevamento del volto, che può essere una qualsiasi, che va posta nella stessa cartella dell'esempio.

Per comodità, si può scaricare dal sito della rivista un'immagine di test (*pier.jpg*) dell'autore.

Ora possiamo avviare l'esecuzione del test con il comando seguente:

```
./armv7l/Release/object_detection_sample_ssd -m face-
detection-adas-0001.xml -d MYRIAD -i pier.jpg
```

Sul terminale di Raspberry Pi appariranno parecchie informazioni.

All'avvio si potrà notare come il motore di infe-

renza venga inizializzato e come viene aggiunta l'immagine da elaborare:

```
[ INFO ] InferenceEngine:
      API version ..... 1.6
      Build ..... 22443
Parsing input parameters
[ INFO ] Files were added: 1
[ INFO ]   pier.jpg
[ INFO ] Loading plugin

      API version ..... 1.6
      Build ..... 22443
      Description ..... myriadPlugin
```

Quindi vengono caricati i due file di face-detection che abbiamo scaricato in precedenza:

```
[ INFO ] Loading network files:
      face-detection-adas-0001.xml
      face-detection-adas-0001.bin
```

Seguono la preparazione del blob di ingresso e del blob di uscita e l'avvio del motore di inferenza:

```
[ INFO ] Preparing input blobs
[ INFO ] Batch size is 1
[ INFO ] Preparing output blobs
[ INFO ] Loading model to the plugin
[ WARNING ] Image is resized from (640, 480) to (672, 384)
```

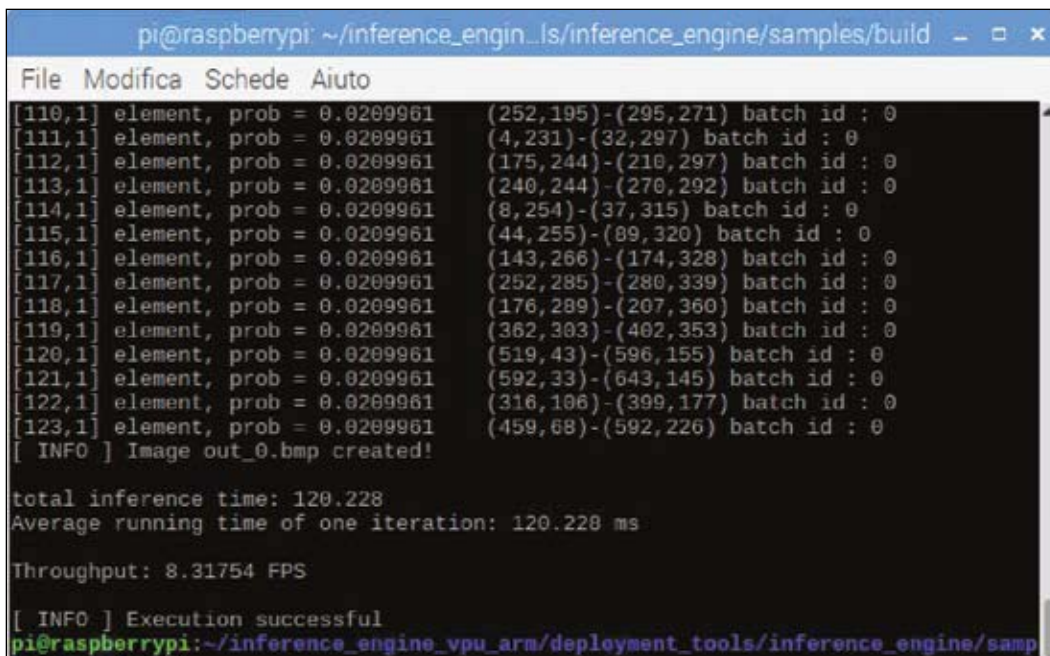
```
[ INFO ] Batch size is 1
[ INFO ] Start inference (1 iterations)
[ INFO ] Processing output blobs
```

Man mano che l'immagine viene elaborata, si vedranno una serie di numeri corrispondenti ai rilevamenti probabilistici di un volto. Un rilevamento è molto preciso quando appare una probabilità vicina a "1" mentre più il valore si discosta da 1, meno probabilità di rilevamento ci sono.

Quando viene trovato l'elemento con la probabilità più alta, appare la scritta **WILL BE PRINTED!**. Questo significa che il motore di inferenza ha trovato, nel nostro caso, un volto e che verrà creato un file di output.

Ecco un estratto dell'output sul terminale:

```
[0,1] element, prob = 0.98291   (333,42)-(581,388)
batch id : 0 WILL BE PRINTED!
...
...
[80,1] element, prob = 0.146484   (302,383)-
(342,423) batch id : 0
[81,1] element, prob = 0.146484   (324,474)-
(363,510) batch id : 0
[82,1] element, prob = 0.146484   (1061,564)-
(1093,601) batch id : 0
[83,1] element, prob = 0.146484   (936,600)-
(978,644) batch id : 0
```



```
pi@raspberrypi: ~/inference_engine_ls/inference_engine/samples/build
File Modifica Schede Aiuto
[110,1] element, prob = 0.0209961   (252,195)-(295,271) batch id : 0
[111,1] element, prob = 0.0209961   (4,231)-(32,297) batch id : 0
[112,1] element, prob = 0.0209961   (175,244)-(210,297) batch id : 0
[113,1] element, prob = 0.0209961   (240,244)-(270,292) batch id : 0
[114,1] element, prob = 0.0209961   (8,254)-(37,315) batch id : 0
[115,1] element, prob = 0.0209961   (44,255)-(89,320) batch id : 0
[116,1] element, prob = 0.0209961   (143,266)-(174,328) batch id : 0
[117,1] element, prob = 0.0209961   (252,285)-(280,339) batch id : 0
[118,1] element, prob = 0.0209961   (176,289)-(207,360) batch id : 0
[119,1] element, prob = 0.0209961   (362,303)-(402,353) batch id : 0
[120,1] element, prob = 0.0209961   (519,43)-(596,155) batch id : 0
[121,1] element, prob = 0.0209961   (592,33)-(643,145) batch id : 0
[122,1] element, prob = 0.0209961   (316,106)-(399,177) batch id : 0
[123,1] element, prob = 0.0209961   (459,68)-(592,226) batch id : 0
[ INFO ] Image out_0.bmp created!

total inference time: 120.228
Average running time of one iteration: 120.228 ms

Throughput: 8.31754 FPS

[ INFO ] Execution successful
pi@raspberrypi:~/inference_engine_vpu_arm/deployment_tools/inference_engine/samp
```

Fig. 3
Termine del processo di elaborazione.



Fig. 4
Immagine per il test.



Fig. 5
Immagine di uscita
del test.

```
...
[110,1] element, prob = 0.0209961 (252,195)-
(295,271) batch id : 0
[111,1] element, prob = 0.0209961 (4,231)-
(32,297) batch id : 0
[112,1] element, prob = 0.0209961 (175,244)-
(210,297) batch id : 0
[113,1] element, prob = 0.0209961 (240,244)-
(270,292) batch id : 0
```

Dopo circa un centinaio di iterazioni, il processo di

elaborazione termina e viene creata un'immagine chiamata **out_0.bmp** nella stessa cartella (**Fig. 3**).

```
[ INFO ] Image out_0.bmp created!
total inference time: 120.228
Average running time of one iteration: 120.228 ms
Throughput: 8.31754 FPS
[ INFO ] Execution successful
```

Se nell'immagine di partenza (**Fig. 4**) viene rilevato un volto, apparirà un riquadro colorato attorno al volto nell'immagine di uscita (**Fig. 5**).

AGGIUNGERE UNA VIDEOCAMERA

Per il nostro prossimo esperimento faremo uso del modulo Picamera di Raspberry Pi, quindi prima di entrare in contatto con i wrapper Python che ci consentiranno il rilevamento in tempo reale, assicuriamoci di avere la nostra videocamera installata e funzionante.

Per collegare la videocamera a Raspberry Pi, si deve usare il connettore CSI al centro della scheda (**Fig. 6**). Si raccomanda di controllare bene l'inserimento fisico del cavo flat nel connettore della scheda. Effettuare questa operazione con la scheda spenta. Se non è già stato fatto, è necessario attivare il supporto per la videocamera dal pannello *Preferenze > Raspberry Pi Configuration > Interfaces* di Raspbian.

Per verificare che la videocamera funzioni, digitare sul terminale il comando

```
raspistill -o testshot.jpg
```

Questo farà vedere per qualche secondo l'imma-

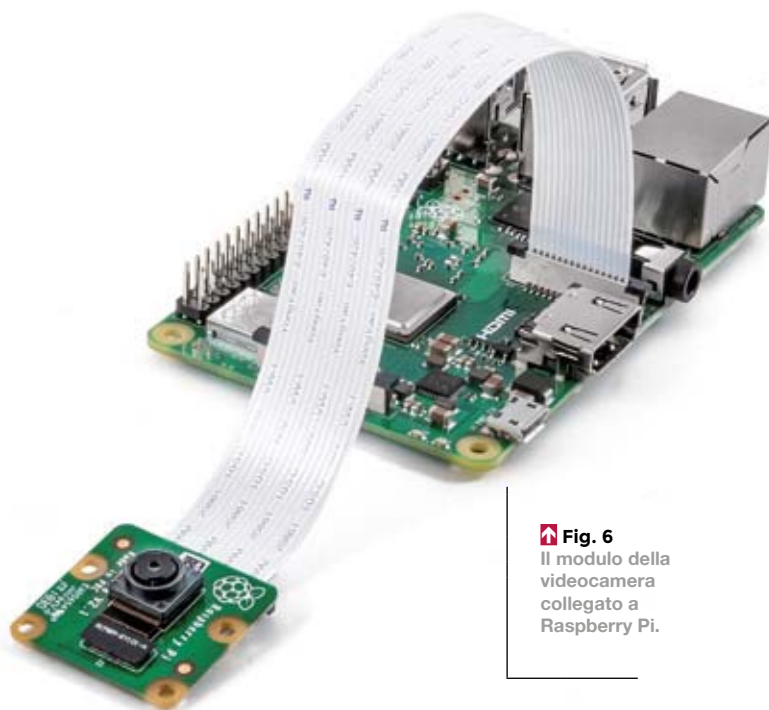


Fig. 6
Il modulo della
videocamera
collegato a
Raspberry Pi.

gine sullo schermo e scatterà una foto salvandola come immagine jpg nella directory home/pi.

STREAMING VIDEO IN TEMPO REALE

Siccome è disponibile del codice incluso nel kit di sviluppo, lo sfrutteremo per creare un'applicazione in Python e dimostrare l'efficienza dell'inferenza su uno streaming video dalla videocamera. Lo script in Python dovrà accedere alla videocamera, per cui, se non è già stato fatto, bisogna installare alcuni tool dal terminale:

```
sudo apt-get install python3-picamera
```

Per la compatibilità con Video4Linux (una API per Linux per la cattura di immagini), è necessario caricare il modulo del kernel BCM2835 appropriato con il comando seguente:

```
sudo modprobe bcm2835-v4l2
```

In questo modo si creerà un dispositivo `/dev/video0`, compatibile con V4L. Quindi bisognerà aggiungere il modulo BCM2835 a `/etc/modules`, in questo modo:

```
sudo -i
# echo 'bcm2835-v4l2' >> / etc / modules
# exit
```

Il modulo verrà caricato al riavvio.

WRAPPER PYTHON PER OPENVINO

Il wrapper Python per il toolkit OpenVINO richiede il supporto NumPy e OpenCV, quindi prima di fare qualsiasi altra cosa, bisogna installarli.

```
sudo apt-get install python3-numpy
pip3 install opencv-python
```

Sono necessarie anche alcune librerie aggiuntive:

```
sudo apt-get install libgtk-3-dev
sudo apt-get install libavcodec-dev
sudo apt-get install libavformat-dev
sudo apt-get install libswscale-dev
```

Per il nostro esperimento useremo una versione modificata dello script di esempio che si chiama **object_detection_demo_ssd_async.py**. Questo script in Python effettua il riconoscimento di oggetti in tempo reale.



Fig. 7
Il rilevamento di un volto in tempo reale.

Per modificare lo script, bisogna aprirlo con un qualsiasi altro editor di testo o, meglio, con IDLE di Python in questo modo:

```
cd /home/pi/inference_engine_vpu_arm/deployment_tools/inference_engine/samples/python_samples/object_detection_demo_ssd_async
```

```
idle3 object_detection_demo_ssd_async.py
```

Una volta aperto lo script, ne dovete modificare la riga 84, ovvero la seguente:

```
input_stream = 0
```

sostituendone il contenuto con questo codice:

```
input_stream = '/dev/video0'
```

In questo modo si imposta il percorso completo del nostro dispositivo compatibile V4L.

A questo punto, possiamo finalmente provare lo script in Python lanciando questo comando da terminale, che attiva il file **face-detection-adas-0001.xml**, messo in precedenza nella cartella build, e il motore MYRIAD:

```
python3 ./object_detection_demo_ssd_async.py -m
../../../../build/face-detection-adas-0001.xml -i cam -d
MYRIAD -pt 0.5
```

Se tutto va bene, dovrebbero apparire sul terminale una serie di informazioni e quindi una finestra con uno streaming video della videocamera:

```
[ INFO ] Initializing plugin for MYRIAD device...
[ INFO ] Reading IR...
[ INFO ] Loading IR to the plugin...
[ INFO ] Starting inference in async mode...
```

CORSI FUTURA ACADEMY

Puoi scegliere tra numerosi argomenti:

- Arduino
- Arduino e i robot
- Centralino telefonico con Raspberry Pi
- C/C++
- CNC
- Droni
- Eagle
- Elettronica
- Fotovoltaico
- FPGA
- FreeCAD
- FreeRTOS
- Hardware PC
- Java
- KiCad
- Labview
- Microchip
- Modellazione 3D
- Mit Appinventor 2
- MPLabX/PIC32
- OrCAD
- Raspberry Pi
- Robotica
- Stampa 3D

... e molti altri ancora!



NON ASPETTARE

Se ti iscrivi in anticipo avrai diritto ad uno sconto!

INFORMAZIONI DETTAGLIATE E ISCRIZIONI SU
www.futura.academy



[INFO] To switch between sync and async modes
press Tab button

[INFO] To stop the demo execution press Esc button

To close the application, press 'CTRL+C' or any key
with focus on the output window

Vi facciamo notare che tutto funzionerà bene sul display principale di Raspberry Pi, quindi con un monitor collegato, la finestra dovrebbe aprirsi senza problemi. Per l'uso senza monitor, la cosa più semplice è abilitare VNC e collegare Raspberry Pi al Personal Computer sfruttando proprio il desktop remoto.

La **Fig. 7** mostra il rilevamento di un volto in tempo reale, con un riquadro in cui viene evidenziata la percentuale di successo.

Per altri tipi di oggetto bisognerà scaricare i modelli dalla directory [https://download.01.org/opencv-notoolkit/2018_R5/open_model_zoo/](https://download.01.org/opencv/notoolkit/2018_R5/open_model_zoo/) e seguire le procedure descritte per la compilazione e l'esecuzione dei modelli.

CONCLUSIONI

Bene, per il momento possiamo fermarci qui. In queste pagine vi abbiamo spiegato come realizzare, partendo da una "semplice" Raspberry Pi dotata di telecamera dedicata, un riconoscimento di volti e delle loro caratteristiche mediante software di libero utilizzo e la stick NCS2. Appena possibile continueremo le nostre sperimentazioni con NCS 2 (Neural Compute Stick 2) della Movidius, sfruttando le sue capacità neurali per il riconoscimento dell'età e del genere del soggetto ripreso. Alla prossima!



Cosa occorre?

Il materiale utilizzato in questo progetto è disponibile presso Futura Elettronica. La board Raspberry Pi 3 Tipo B+ (cod. RPI3BPLUS) è in vendita a Euro 44,00, il modulo con telecamera da 8 megapixel per Raspberry Pi 3 (cod. 9132664RS) è disponibile a Euro 39,90. La telecamera è disponibile anche nella versione senza filtro IR (cod. 9132673RS) al prezzo di Euro 39,90.

Il materiale va richiesto a:

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

UN SEMPLICE CARICO ELETTRONICO

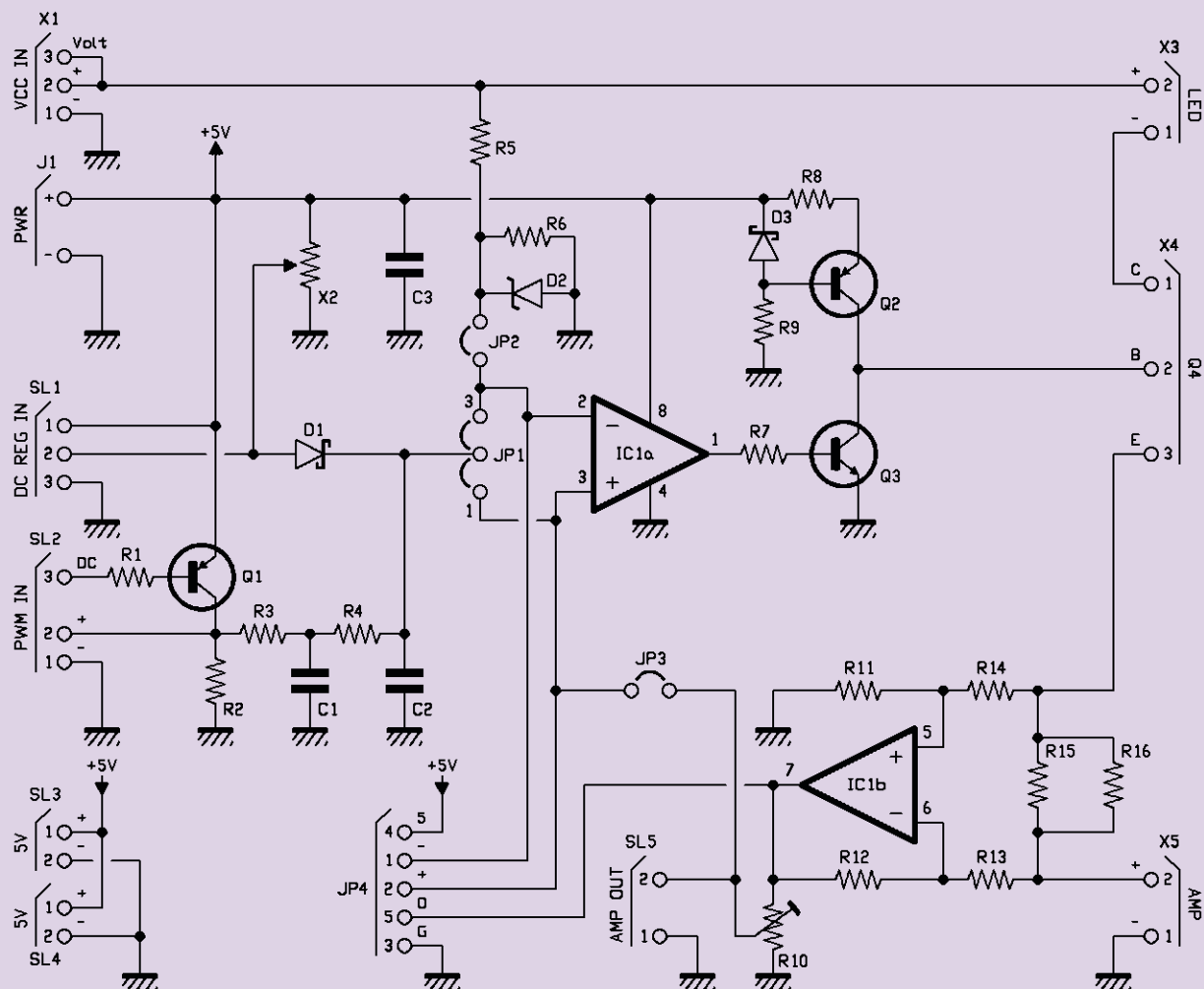


Costruiamo un circuito in grado di simulare, pilotando un transistor di potenza, una resistenza variabile che funziona da carico in corrente continua e che nel caso specifico è destinato al test dei regolatori di corrente per illuminazione a LED.

di **DIEGO FRASSON**

Tante volte capita di dover installare degli alimentatori, specie a corrente costante per i LED, e domandarsi quali saranno i loro limiti, ovvero se le specifiche dichiarate dal costruttore saranno reali. In casi del genere occorrerebbe disporre di un carico campione in grado di farsi attraversare dalla corrente prescritta e sotto la tensione di lavoro prevista; qualcosa del genere può essere ottenuto con un reostato o con combinazioni di resistori di potenza, ma meglio ancora è utilizzare un carico elettronico, il quale sostanzialmente è un circuito capace di presentare un assorbimento costante sottoposto a determinate differenze di potenziale. Il circuito descritto in queste pagine è proprio un carico elet-

tronico (anche noto come Carico Attivo) a corrente regolabile, nato dall'esigenza effettiva dell'autore di provare degli alimentatori a corrente costante per LED, dei quali non si riusciva a capire dalle specifiche tecniche (ormai molto vaghe, specie in dispositivi di produzione cinese...) quali fossero i limiti e se la corrente che erogavano era esattamente quella che ci si aspettava, quali fossero le tolleranze e l'effettivo intervento delle protezioni previste ai valori di corrente dichiarati. Tenuto conto che un'apparecchiatura di tipo commerciale costava svariate centinaia di euro, abbiamo pensato di realizzare un carico elettronico utilizzando, come elemento di potenza, il popolare e collaudatissimo 2N3055, che è un pezzo di storia importante per gli appassionati di elettronica. Abbiamo pensato quindi di realizzare un circuito adatto sia a carichi a



corrente costante che a carichi a tensione costante, ottenendo un ottimo carico elettronico versatile e poco costoso, senza ricorrere ad alcun sistema digitale e, se vogliamo, un po' retrò, visto che fa uso

di componentistica tradizionale e del 2N3055, che qualcuno potrebbe considerare "storico". Il circuito è quello che vedete nello schema elettrico in questa pagina e che andiamo a descrivere.

SCHEMA ELETTRICO

Il circuito prende le sue origini dal circuito di potenza ad amplificatore lineare in base comune utilizzato nei vecchi schemi di stabilizzazione lineari di tensione; infatti stabilizziamo la tensione, cambiando la configurazione del circuito, su un resistore di potenza. Per funzionare, il circuito abbisogna di un'alimentazione a tensione di 5Vcc facilmente reperibile tramite comuni alimentatori a spina che si usano anche per la ricarica dei cellulari e che potete trovare sul sito della Futura Elettronica, www.futurashop.it.

Vediamo in dettaglio lo schema elettrico: il cuore del circuito è rappresentato dai transistor Q2, Q3 e Q4 (2N3055) montato all'esterno e collegato

Fig. 1
La scheda del
carico elettronico.



tramite il connettore siglato X4. Il Q2 funziona da generatore di corrente costante che alimenta la giunzione base-emettitore del transistor di potenza Q4, il quale viene inserito con il circuito collettore-emettitore in serie all'alimentazione applicata ai punti Vccin; è proprio per questo che viene collegato esternamente al circuito stampato e montato su dissipatore, in quanto è esso stesso che fa da carico e dissipa -sotto forma di calore- tutta la potenza fornita dall'alimentatore in prova. La corrente generata da Q2 è tarata ad un valore molto alto e tale da permettere di far scorrere oltre 2A di corrente di prova.

Il calcolo che permette di determinare la corrente di lavoro di tale generatore di corrente costante è il seguente:

$$I_{c2} = (V_{d3} - V_{be2}) / R_8$$

dove la corrente I_{c2} è quella di collettore di Q2, V_{d3} è la tensione dello Zener D3 e V_{be2} è la tensione V_{be} del Q2.

La regolazione avviene come in tutti gli schemi di questo tipo, ossia drenando verso massa, tramite un transistor (nel nostro caso è Q3...), la corrente di pilotaggio della base e quindi facendo diminuire la corrente di polarizzazione del transistor di potenza. L'effetto è, nel caso di prova a corrente costante, quello di alzare la tensione ai capi del carico al fine di ristabilire la corrente, mentre nel caso di operazione a tensione costante, quello di diminuire la corrente di test.

In altre parole, la corrente nel carico determina la tensione d'ingresso dell'amplificatore differenziale basato su IC1b, la quale cresce in proporzione la corrente di emettitore del transistor e quindi

dell'assorbimento da parte del carico elettronico: se la corrente sale, a parità di valore richiesto, la tensione fra i piedini 5 e 6 dell'IC1b aumenta e con essa sale il potenziale sul piedino 7 dell'operazionale, quindi cresce anche il potenziale sul cursore del trimmer R10. Aumenta, di conseguenza, il potenziale sul piedino 3 dell'IC1, il che comporta un aumento della tensione d'uscita dell'IC1a e quindi la V_{be} del transistor Q3, cosicché sale la corrente di collettore di quest'ultimo e, supponendo costante la corrente di collettore del Q2, viene sottratta corrente alla base del Q4, il quale quindi diminuisce la propria corrente di collettore e, di riflesso, quella di emettitore.

Viceversa, se la corrente di emettitore del 2N3055 scende sotto il valore atteso, la caduta di tensione sul parallelo R15-R16 cala e con essa si riduce la tensione d'ingresso dell'amplificatore differenziale, quindi scende il potenziale di uscita dell'IC1b e quello sull'ingresso non-invertente dell'IC1a e quest'ultimo riduce la propria tensione d'uscita, abbassando così la V_{be} del Q3 e consentendo un aumento della corrente nella base del Q4. Nello schema elettrico del nostro circuito abbiamo messo in serie due collegamenti: uno per un amperometro (connettore X3) ed uno per un eventuale carico a corrente costante (connettore X5, cui possiamo collegare, ad esempio, dei LED di potenza da regolare) che possono essere cortocircuitati nel caso non occorressero.

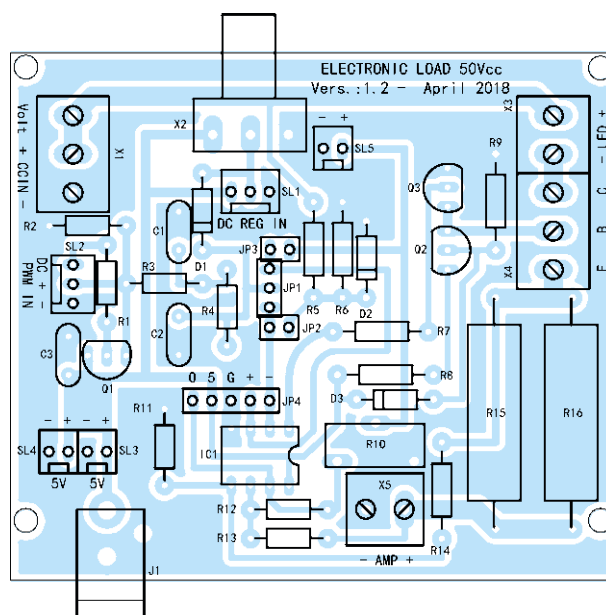
In pratica l'alimentatore o regolatore di corrente da provare si deve connettere alla morsettiera Vccin e alla morsettiera siglata LED potete eventualmente connettere la lampada o strip a LED che l'alimentatore dovrebbe alimentare; altrimenti X3 andrà cortocircuitata.

CARATTERISTICHE TECNICHE

- ↓ **Tensione di alimentazione:**
5Vcc (PWR)
- ↓ **Tensione d'ingresso:**
12÷55Vcc (VccIN)
- ↓ **Corrente assorbita:**
100 mA
- ↓ **Corrente del carico:**
0÷5 A
- ↓ **Controllo PWM:**
da ingresso PWM IN 0/5V
- ↓ **Uscita per amperometro:**
- ↓ **Controllo in continua:**
tramite potenziometro

Fig. 2
Il sistema completo di voltmetro e amperometro.





Elenco Componenti:

R1, R7, R10, R13, R14: 10 kohm
 R2: 1 kohm
 R3, R4, R5: 47 kohm
 R6: 4,7 kohm
 R8: 100 ohm 1/2W
 R9: 220 ohm
 R11, R12: 22 kohm
 R15, R16: 1 ohm 5W
 C1, C2, C3: 100 nF ceramico
 D1: BAT42
 D2: Zener 5,6 V 400 mW
 D3: Zener 2,7 V 400 mW

IC1: MCP6002P
 Q1, Q2: BC556
 Q3: BC546
 X2: potenziometro 10 kohm

Varie
 - 2N3055
 - Vite 12 mm M4 (2 pz.)
 - Dado M4 (2 pz.)
 - Dissipatore
 - Plug alimentazione
 - Zoccolo 4+4

- Morsetto 2 vie passo 5,08mm (2 pz.)
 - Morsetto 3 vie passo 5,08mm (2 pz.)
 - Jumper (2 pz.)
 - Strip maschio 2 vie (2 pz.)
 - Strip maschio 3 vie
 - Strip maschio 5 vie
 - Connettore filo-scheda da CS 2 vie (3 pz.)
 - Connettore filo-scheda da CS 3 vie (2 pz.)
 - Modulo Voltmetro/Amperometro
 - Circuito stampato S1480 (70x80 mm)

L'una o l'altra opzione è, per il circuito, poco rilevante, perché l'elettronica si adatta per far fluire comunque la corrente di test.

Spiegato ciò, a questo punto vediamo il circuito di retroazione, per ottenere il quale abbiamo utilizzato un moderno circuito integrato della Microchip, che contiene due amplificatori operazionali ad alte prestazioni.

REGOLAZIONE A CORRENTE COSTANTE

In questa modalità dovremo chiudere il jumper J1 nella posizione 2-3, chiudere il jumper JP2 e lasciare aperto il jumper JP3. Quello che dobbiamo regolare e stabilizzare, visto che l'alimentatore per LED in prova deve stabilizzare la corrente, è la tensione a capi del carico costituito dal carico ausiliario (LED di potenza), dal transistor Q4, dal resistore di

potenza e dall'amperometro. Con un partitore di tensione abbassiamo la tensione da stabilizzare a meno di 5 volt e mettiamo un diodo Zener di protezione all'ingresso invertente dell'operazionale (piedino 2).

La relazione è la solita di un qualunque amplificatore non invertente e in questo caso il funzionamento viene descritto dalla formula seguente:

$$V_o = (1 + R_5/R_6) * V_i$$

Qui è il caso di spendere qualche riga per spiegare il modo con il quale il circuito provvede a regolare la corrente. Di solito si fa in modo di variare il rapporto delle resistenze (nel nostro caso R5 e R6) mettendo una tensione di riferimento su V_i tuttavia nessuno ci dice che non possiamo cam-

biare, invece, la V_i ; anzi, tale metodica porta con sè l'indiscutibile vantaggio che potendo arrivare a 0 volt, possiamo ridurre la tensione di uscita teoricamente fino a zero. Infatti quello che faremo sarà regolare la tensione V_i di ingresso dell'amplificatore operazionale, regolando in questa maniera il circuito fino ad una tensione, ai capi dell'alimentatore in prova, di 55V, condizione, questa, che si raggiunge portando la V_i al positivo dei 5V di alimentazione.

Se vogliamo sapere a quanto ammonta la corrente nel circuito, ci basterà procurarci un amperometro (analogico o digitale, va bene lo stesso) e collegarlo all'uscita "Misura I" il cui funzionamento spiegheremo nelle prossime righe.

REGOLAZIONE A TENSIONE COSTANTE

In questa configurazione dovremo chiudere il jumper J1 nella posizione 1-2, lasciare aperto il jumper JP2 e chiudere il jumper JP3. Questa volta dobbiamo regolare la tensione ai capi della resistenza di potenza rappresentata dal parallelo di R15 e R16 in modo che la tensione ai suoi capi rappresenti la corrente che scorre nel circuito. Qui usiamo il secondo amplificatore operazionale IC1B che è collegato in modalità differenziale e quindi raddoppia (in base ai valori usati sulle 4 resistenze R11, R12, R13 e R14) la tensione ai capi della resistenza "campione" (il parallelo R15 e R16). Questa tensione di errore viene condotta all'ingresso dello stadio di regolazione rappresentato dal primo operazionale IC1A, il quale provvede al confronto con la tensione di riferimento e quindi regola la corrente di test che scorre nel collettore (o meglio, tra collettore ed emettitore) del transistor Q4.

All'uscita dell'amplificatore IC2 è collegato un trimmer che serve per poter fare una taratura sulla corrente di misura in quanto i componenti che usiamo hanno una tolleranza abbastanza significativa che abbisogna di una compensazione.

LA TENSIONE DI RIFERIMENTO

Spendiamo ora due parole su come impostare la tensione di riferimento. Il primo modo, il più classico è quello di usare una tensione molto ben stabilizzata e regolare con un potenziometro la tensione che giunge all'ingresso dello stadio di confronto costituito dall'operazionale IC1A. Questo primo sistema è rappresentato dal circuito che sta a monte del diodo D1, che è un diodo Schottky per minimizzare la caduta di tensione e renderla minore di 0,2V, costituito o da un potenziometro

o da una sorgente in tensione continua stabile applicata sul connettore SL1.

Il secondo sistema è invece pensato per coloro che vogliono collegare una sorgente di tipo PWM la cui tensione media è pari al duty-cycle, moltiplicata la tensione massima del segnale. Il segnale può essere applicato così com'è applicandolo sul pin 2 del connettore SL2 oppure, se l'uscita da cui si preleva il PWM è di tipo open-collector, sul pin 3 dello stesso connettore (in questo caso il transistor Q1 provvederà ad amplificare correttamente il segnale, invertendolo di livello).

Lo stadio successivo non è nient'altro che un integratore passivo realizzato con due celle di filtro R-C passa-basso del prim'ordine in cascata, dimensionato per avere una frequenza di taglio superiore fatta per funzionare perfettamente con il segnale creato dall'istruzione **analogWrite** di Arduino, la quale sulle uscite predisposte genera un PWM a una frequenza di 440 Hz; con i valori inseriti in sede progetto potete comunque giocare intorno a quel valore di frequenza e, per l'esattezza, non dovrete avere problemi se decideste scegliere, per il segnale PWM, una frequenza inferiore, anche fino ai 100 Hz.

Una frequenza più alta, invece, non creerebbe alcun problema, qualunque ne sia il valore, giacché mentre abbassando troppo la frequenza si arriva a un punto in cui ai capi dei condensatori la tensione pulsa e rende quindi discontinua la resistenza dinamica introdotta dal nostro carico elettronico e, nello specifico, simulata dalla conducibilità del circuito di collettore del Q4.

Il diodo Schottky D1, presente nello schema, serve ad evitare che eventuali potenziometri saldati o sorgenti in continua collegate al connettore SL1 influenzino il funzionamento in PWM modificandone la resistenza di uscita dello stadio di ingresso responsabile del filtraggio.

Attenzione: è importante che la tensione massima che arriva sugli ingressi dell'operazionale non superi i 5V, altrimenti l'operazionale stesso può guastarsi irreparabilmente.

COSTRUZIONE E PROVE PRATICHE

Spiegato il funzionamento del circuito, passiamo adesso alle note costruttive: il tutto richiede un circuito stampato facilmente realizzabile per fotoincisione a partire dalla traccia lato rame scaricabile dal nostro sito Internet www.elettro-nicain.it; trattandosi di un PCB di tipo monofaccia, la realizzazione è praticamente alla portata di tutti. Inoltre non presenta particolari problemi,



→ Fig. 3

La scheda si alimenta tramite il plug, mentre il circuito su cui agisce il carico si connette alla morsettiera Vccin.



anche perché i componenti utilizzati sono tutti per montaggio tradizionale, quindi niente attrezzature speciali, paste flussanti e lenti d'ingrandimento. Dunque, una volta inciso e forato lo stampato, disponetevi i componenti a partire dalle resistenze di piccola potenza e dai dodi al silicio (per i quali dovrete rispettare il verso d'inserimento indicato nel piano di montaggio pubblicato in queste pagine) e proseguendo con lo zoccolo dell'integrato, i transistor in TO-92 (anche per questi seguite il verso indicato nel piano di montaggio) e i condensatori, quindi i pin-strip e i connettori a passo 2,54 mm.

Proseguite inserendo e saldando i resistori di potenza a filo, il jack di alimentazione, le morsettiere, il trimmer verticale e il potenziometro.

Il componente cui prestare una certa attenzione è il transistor di potenza Q4, giacché abbiamo pensato di racchiudere il circuito in un mobile di tipo metallico e il transistor Q4 andrà applicato (fissato mediante viti M4 provviste di dado), spalmato di grasso al silicone sulla parte posteriore, su un dissipatore, a sua volta montato su una delle superfici metalliche del mobile come vedete dalla foto del prototipo in queste pagine.

Non dovete mai dimenticare che tutta la potenza che fornite con l'alimentatore in prova viene dissipata in calore. Funziona quindi ottimamente, anche come stufetta elettrica, arrivando fino ad un centinaio di watt di dissipazione...

Nel nostro prototipo, come vedete nelle foto in queste pagine abbiamo inserito uno strumento combinato che misura sia la tensione ai capi del carico che la corrente in modo da avere uno strumento di test semplice ma efficace.

Il carico elettronico l'abbiamo testato con due alimentatori della Velleman per LED, riuscendo finalmente a capire tensione minima e massima fornita dalle apparecchiature e verificando la "tenuta" di tali alimentatori al carico descritto.

OVERBOOST

Una delle cose che di solito si chiedono ai progettisti è come aumentare i limiti di tensione e corrente stabiliti nel progetto. Ebbene, il nostro carico elettronico è stato provato in laboratorio facendo due semplici modifiche, abbiamo portato il resistore R5 raddoppiandone il valore e portandolo a 100 kΩ e mettendo al posto del 2N3055 un MJ11016G (è un transistor Darlington di potenza con $V_{ce_{max}} = 120V$ e $I_{c_{max}} = 30A$) e lo abbiamo provato con un alimentatore a corrente costante fino a 100V e con alimentatore a tensione costante con corrente fino a 5A e il circuito non ha fatto una piega.

Si deve assolutamente usare un Darlington di potenza, altrimenti lo stadio di pilotaggio della corrente del transistor di potenza del nostro circuito è insufficiente, visto il basso h_{FE} del 2N3055.

Una cosa importante cui stare attenti è il non eccedere con corrente e soprattutto tensione, perché superati i 120 V in corrente continua si entra nelle apparecchiature di classe I, il che, tradotto in pratica, significa che non si è più in bassa tensione, quindi c'è il rischio concreto di folgorazione.

Tenetelo ben presente se intendete operare con tensioni superiori a quella massima prevista dal progetto!

CONCLUSIONI

Bene, con questo abbiamo terminato la descrizione del nostro carico elettronico. Il circuito, seppure sia nato per il test dei regolatori di corrente per illuminotecnica a LED, può essere impiegato come generico carico attivo per provare alimentatori in continua anche privi di regolazione della corrente, per esempio per verificarne sperimentalmente la retta di carico. □



Cosa occorre?

Il materiale utilizzato in questo progetto può essere acquistato presso Futura Elettronica, in particolare l'operazionale MCP6002 in versione DIP (cod. MCP6002P) è disponibile al prezzo di Euro 0,90 e il Voltmetro/Amperometro da pannello (cod. PAN100V10A) è in vendita a Euro 9,00. Il prezzo si intende IVA compresa.

Il materiale va richiesto a:

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

La tua casa sotto una nuova luce!

Rinnova e migliora l'illuminazione della tua abitazione.

**Set 3 lampade LED bianco naturale
220 VAC - attacco E27 - 11W**

**SET 3
LAMPADINE**



cod. HT6011
€ 6,90

Confezione contenente 3 lampade LED con emissione luminosa bianco naturale, angolo di emissione >270°, temperatura colore 4000 K, attacco E27, basso consumo energetico e accensione istantanea. Alimentazione: 175-250 VAC, dimensioni: Ø60x109 mm.

**Set 3 lampade LED bianco naturale
220 VAC - attacco E14 - 5W**

**SET 3
LAMPADINE**



cod. HT4505
€ 4,90

Confezione contenente 3 lampade LED con emissione luminosa bianco naturale, angolo di emissione >270°, temperatura colore 4000 K, attacco E14, basso consumo energetico e accensione istantanea. Alimentazione: 175-250 VAC, dimensioni: Ø45x81 mm.

**Set 3 lampade LED candela
bianco naturale 220 VAC
attacco E14 - 5W**

**SET 3
LAMPADINE**



cod. HT3705
€ 4,90

Confezione contenente 3 lampade LED a candela con emissione luminosa bianco naturale, angolo di emissione >270°, temperatura colore 4000 K, attacco E14, basso consumo energetico e accensione istantanea. Alimentazione: 175-250 VAC, dimensioni: Ø45x81 mm.

**Lampada tubolare con LED bianco
a luce neutra - 230VAC - E27**



cod. APT37N
€ 4,90

Lampada tubolare a LED con luce neutra con potenza di 9W, temperatura di colore 4000K, flusso luminoso 800lm.

**Lampada LED con crepuscolare
attacco E27 - 10W**



**SENSORE
CREPUSCOLARE
INTERNO**

**800
LUMEN**

Lampada a basso consumo energetico. Accensione istantanea, alimentazione: 230 Vac, luce bianca neutra, dimensioni: Ø60x115 mm, attacco E27.

cod. LAMPWSENSOR
€ 7,50

**Lampada a LED solare con batteria interna,
sensore crepuscolare e di movimento**



Lampada a LED 4W da esterno con grado di protezione IP44. Non necessita di collegamenti alla rete elettrica per funzionare in quanto dotata di batteria interna e pannello solare per la ricarica. Dispone di sensore crepuscolare e sensore PIR di movimento. Portata del sensore di movimento (PIR) 3-5 metri. Dimensioni 135x100x50 mm, peso 180 grammi.

cod. SOL01
€ 14,90

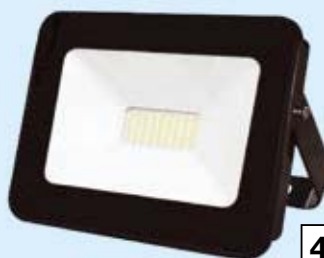
**Faro a LED da esterno (IP65) con sensore a microonde
e telecomando**



Composto da 30 LED SMD ad alta luminosità con luce bianca naturale. Dispone di sensore di movimento a microonde e di telecomando per le impostazioni e il controllo a distanza. Può essere acceso/spento manualmente tramite il tasto ON/OFF, può essere impostato per accendersi automaticamente al passaggio di qualcuno. Flusso luminoso: 1400 lumen, temperatura colore 4000 K, angolo di emissione 120°, alimentazione 230 VAC.

cod. LEDA7002NW-BM
€ 29,90

Faro a LED da esterno (IP65) Bianco neutro 50W - 4500 Lumen



**4500
LUMEN**

Faretto da esterno leggero e elegante, rifinito con vernice epossidica di colore nero. Lampada led a luce bianca neutra (4000K) con una potenza totale di 50 watt e un flusso luminoso di 4500 LUMEN ad elevata efficienza energetica. Accensione istantanea e senza sfarfallamento, resistente all'acqua e alle alte temperature. Completo di staffa di montaggio regolabile.

cod. FB50N
€ 34,00

UNI-T® STRUMENTI DI MISURA PER OGNI ESIGENZA!

PINZA AMPEROMETRICA PROFESSIONALE

Misura correnti AC e DC fino a 100 A, tensioni AC e DC fino a 600 volt, resistenze fino a 20 Mohm, capacità fino a 20 mF, continuità e diodi. Dispone di display LCD retroilluminato a 2000 conteggi, individuazione di tensione senza contatto (NCV), autospegnimento, indicatore di batteria scarica e data hold. Alimentazione con 2 batterie tipo AAA 1,5V (includere).



€68,00

Cod. TP187

OSCILLOSCOPIO PALMARE 16 MHZ CON MULTIMETRO INTEGRATO

Oscilloscopio palmare con multimetro integrato dalle caratteristiche professionali con larghezza di banda di 16MHz. Dispone di ampio display monocromatico (60x60mm) con una risoluzione di 160x160 pixel molto facile da leggere grazie alla funzione di retroilluminazione di cui è dotato. Integra una memoria interna che consente di salvare e visualizzare sul display fino a 10 segnali; pulsante "Auto-Set" per lavorare in modo semplice e veloce. È dotato di porta USB che consente di trasferire i dati dei segnali dall'oscilloscopio al computer. Completo di software di gestione per PC. Alimentazione tramite batterie oppure adattatore di rete.

€289,00

Cod. TP174



MULTIMETRO DIGITALE CON MISURA DI TEMPERATURA, CAPACITÀ, FREQUENZA, E TRANSISTOR

€31,90

Cod. TP0060

Misura correnti continue e alternate fino a 20 A, tensioni continue fino a 1000 V e alternate fino a 750 V, resistenze fino a 200 Mohm, capacità da 2nF a 20μF, frequenze fino a 20 kHz, temperatura da -40°C a +1000°C con termocoppia inclusa, diodi, transistor e continuità elettrica. Alimentazione con batteria a 9 V (inclusa). La confezione comprende: manuale, puntali, batteria, guscio di protezione e sonda di temperatura (termocoppia tipo K).



MINI FONOMETRO DIGITALE

Fonometro con risoluzione di 0,1 dB dotato di display LCD retroilluminato con indicazione digitale della misura. Rileva intensità sonore comprese tra 30 e 130 dB. Memorizza i valori minimi e massimi rilevati. Leggero, compatto, semplice da usare, ideale per monitorare i livelli di rumore negli hotel, nelle sale conferenze, nell'home theater, negli ospedali e in qualsiasi altro ambiente sensibile al rumore. Funziona con 3 batterie tipo AAA (includere).



€ 25,00

Cod. TP0070

TACHIMETRO DIGITALE CON LASER E INTERFACCIA USB PER PC

Tachimetro digitale con puntatore laser; dispone di porta USB per la gestione e il salvataggio dei dati su PC. Indicatore per la misurazione, senza contatto, della velocità angolare di alberi motore, ruote dentate e pulegge, può anche essere utilizzato come contapezzi. Funziona con 4 batterie tipo AA (includere).



€78,00

Cod. TP0030

TERMOMETRO DIGITALE PORTATILE PER TERMOCOPPIE TIPO K E J

€ 19,90

Cod. TP0062

Rileva temperature fino a 1300 °C. Dispone di ampio display LCD retroilluminato a 4 cifre che permette di leggere facilmente i valori. Visualizzazione in °C e in °F selezionabile. Mostra la temperatura massima, minima e media rilevata. La termocoppia tipo K inclusa permette di rilevare una temperatura massima fino a 260°C. Funziona con 3 batterie tipo AAA (includere).



LUXMETRO PORTATILE 0 - 199900 LUX

€ 34,00

Cod. TP0063



Misura l'illuminazione prodotta da LED (luce visibile), lampade fluorescenti, lampade ad alogenuri metallici, lampada al sodio ad alta tensione o lampada ad incandescenza. Dispone di display LCD retroilluminato a 4 digit, unità di misura selezionabili LUX / FC, indicazione di min e max, indicazione di batteria scarica, spegnimento automatico, data hold. Funziona con 3 batterie tipo AAA (includere).

ANEMOMETRO DIGITALE

€ 37,00

Cod. TP0064



Misura la velocità e la temperatura dell'aria. Lettura della velocità dell'aria in m/s, km/h, ft/min, knots, mph. Lettura della temperatura in gradi Centigradi e gradi Fahrenheit, display LCD retroilluminato a 3 cifre, memorizzazione dei valori minimi, massimi e medi rilevati, funzione di autospegnimento, indicatore di batteria scarica. Funziona con 3 batterie tipo AAA (includere).

Prezzi IVA inclusa

HUB USB2 A 4 PORTE

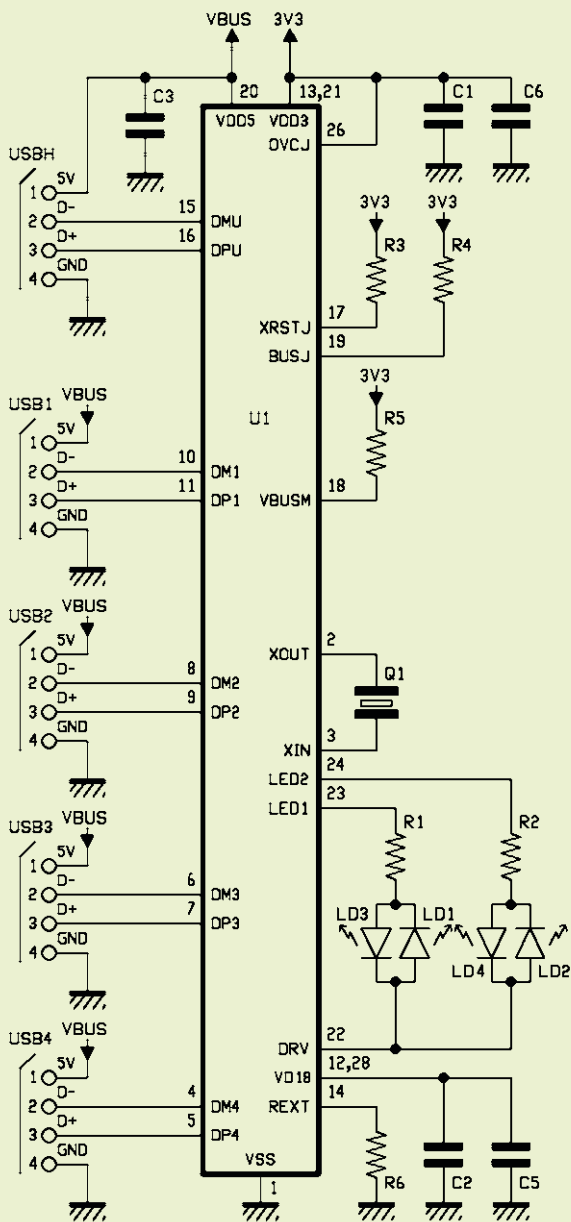


di ALESSANDRO SOTTOCORNOLA

Basato su un single-chip della Terminus, consente di connettere quattro dispositivi USB partendo da una porta microUSB; il form-factor è stato studiato per l'abbinamento a Raspberry Pi Zero.

Più o meno tutti sappiamo che cos'è un hub USB, dal momento che almeno una volta ci abbiamo avuto a che fare a casa o al lavoro per collegare a una porta del Personal Computer tanti dispositivi, ossia per connettere più dispositivi di quante erano le porte USB disponibili nel PC.

Ebbene, l'hub è sostanzialmente un moltiplicatore di porte USB che permette, partendo da un canale connesso a un Host USB (quale la connessione USB interna al computer, per esempio...) di sdoppiarlo, quadruplicarlo e via di seguito, distribuendo, oltre al bus, anche l'alimentazione. Lo smistamento in realtà consiste in una ripetizione del canale dati, che è fattibile perché secondo il protocollo Universal Serial Bus, a un bus USB, composto dai canonici due fili di dati (D+ e D-) riferiti alla massa comune con l'alimenta-



zione (GND), si possono affacciare 127 dispositivi (il 128° è l'Host, ossia l'unità Master del bus). Sebbene in commercio si trovino hub di tutte le forme e dimensioni, alimentati e non, ci è sembrato interessante sviluppare un nostro progetto e proporvelo in queste pagine, per almeno due ragioni: si tratta di una piccola scheda facilmente integrabile in altri apparati elettronici e per di più il suo form-factor è stato studiato per sovrapporla alla popolare Raspberry Pi Zero, alla quale si connette dal connettore microUSB posto sul lato per espanderne le possibilità (la Raspberry Pi Zero dispone infatti di una sola USB).

Andiamo dunque ad analizzare lo schema elettrico del nostro piccolo hub universale, che supporta la versione 2.0 High Speed del protocollo e che consente quindi di collegare dispositivi consentendo una comunicazione tra essi alla velocità (data-rate) di ben 480 Mbit/s.

IL CIRCUITO

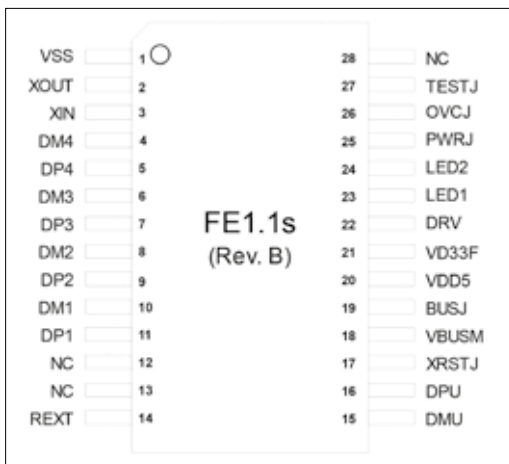
L'intero hub è stato realizzato praticamente da un solo componente, che è il circuito integrato FE1.1 della Terminus, il quale internamente ingloba un completo hub USB 2.0 High Speed a quattro porte (chiamate "di downstream") più una d'ingresso (o di partenza, se preferite, che è quella di "upstream"), il tutto incapsulato in un package SMD da 14+14 pin SSOP, come vedete nella **Fig. 1** in cui è mostrata la piedinatura dell'integrato.

Il nostro componente supporta funzionalità speciali come la GSMA Universal Charging Solution (UCS), è pienamente conforme alla specifica USB-IF "Universal Serial Bus rev. 2.0 e alla Battery Charging rev. 1.1; il dispositivo permette di implementare un hub USB 2.0 a 4 porte a basso consumo.

L'alta qualità dell'FE1.1 è garantita dalle funzionalità integrate di scansione complessiva di sistema e modalità di autotest integrate, che possono essere applicate sul front-end analogico (AFE) High, Full e Low Speed in fase di costruzione e collaudo.

L'FE1.1 potrebbe essere configurato opzionalmente per supportare la modalità Charging Downstream Ports, secondo quanto definito dalla specifica USB-IF; abilitando questa funzionalità, l'hub USB può essere facilmente trasformato in una soluzione di ricarica delle batterie di dispositivi portatili come smartphone e tablet. Ciò stabilito, andiamo a esaminare lo schema elettrico, nel quale vediamo l'integrato U1 collegato nella classica configurazione che prevede un

Fig. 1
Il package
dell'FE1.1 utilizzato
nel nostro hub.



connettore USB di collegamento all'Host (USB H) e quattro per le periferiche device (USB1, USB2, USB3, USB4) che sono equivalenti tra loro; dall'USBH si preleva anche l'alimentazione per il chip e per le quattro USB device.

L'integrato ha un regolatore di tensione interno a 3,3V la cui tensione d'uscita, disponibile tra il piedino 21 e massa, viene filtrata dai condensatori C1 e C6 e va a portare a livello logico alto il pin OVCJ, che riguarda lo stadio di protezione dall'eccessivo assorbimento di corrente. Tutta la logica interna all'integrato funziona a 3,3V mentre a 5V opera invece il lato dati USB.

Il funzionamento del tutto ed in particolare il data-rate delle porte USB, è sincronizzato attraverso l'oscillatore interno che fa capo al quarzo Q1; nello schema elettrico vedete il solo quarzo perché i condensatori di "carico" che con esso concorrono a determinare la frequenza di oscillazione, sono integrati nell'FE1.1.

I resistori di pull-up che vedete nel circuito servono per impostare o disattivare alcune funzionalità facenti capo ai relativi piedini: per esempio quello collegato sul piedino XRSTJ pone a livello alto la linea di reset esterno, che in questo nostro progetto non utilizziamo e che perciò manteniamo a 1 logico (3,3V); la linea consente di resettare, per esempio con un pulsante, il funzionamento dell'hub, laddove si bloccasse, quindi se volete disporre di un hard reset collegate un pulsante unipolare normalmente aperto tra il pin 17 e GND e lo otterrete.

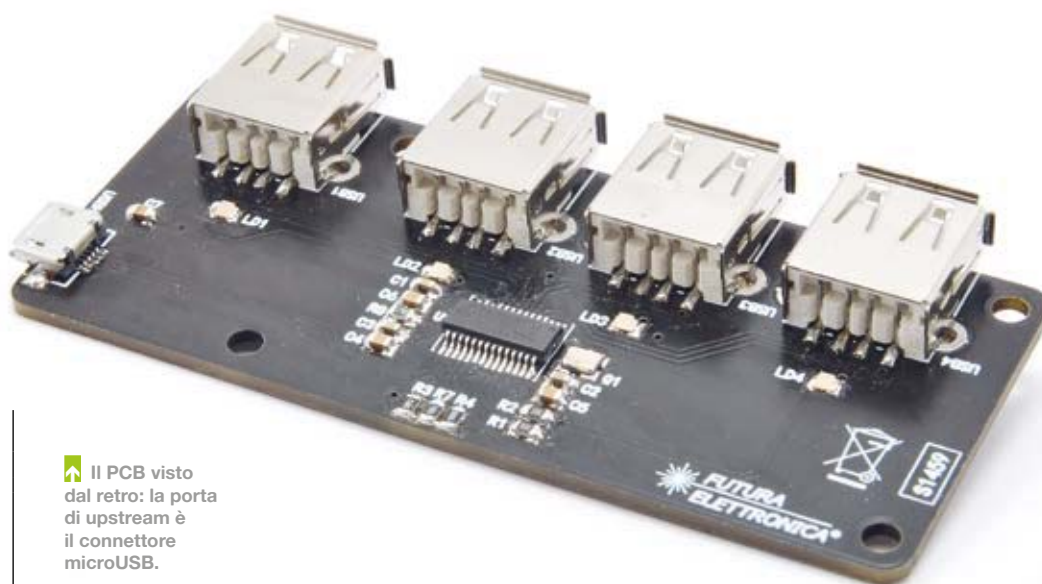
R4 imposta il funzionamento del Bus Power Sense attraverso il piedino VBUSJ, il cui stato

identifica la fonte di alimentazione principale power dell'hub; nel nostro caso, il livello alto indica che l'hub è autolimentato e che la fonte ha sufficiente potenza da alimentare dispositivi High-Power sul lato delle porte downstream (è il caso dell'utilizzo di un alimentatore che fornisca corrente -attraverso un plug coassiale- ai piedini Vdd5 e GND).

Altrimenti, ossia se il pin 19 viene tenuto a livello basso, si presume che l'alimentazione provenga attraverso la presa USB Host e pertanto l'integrato limita la corrente di ogni porta USB downstream a 100 mA.

R5, applicata al pin 18, gestisce la funzione Upstream Port Power (VBUS) Monitor, nel senso che l'integrato monitora la presenza dell'alimentazione sul bus USB lato Host: livello alto significa che l'Host è alimentato e quindi l'hub può funzionare normalmente, mentre se il livello è basso significa che sulla presa di ingresso USBH manca alimentazione e quindi l'hub viene messo nella condizione di power-down state.

Nel circuito troviamo anche quattro LED montati a due a due in antiparallelo e pilotati tramite due pin specifici che sono LED1 (piedino 23) e LED2 (piedino 24); i quattro LED hanno a loro volta un pin comune che è DRV, il quale viene commutato dalla logica interna in modo da far accendere i diodi in modalità multiplex. In pratica il controllo dei LED avviene così: quando occorre accendere LD1 e LD2 (o uno solo dei due) si pone la linea DRV a livello alto e LED1 e/o LED2 a zero logico, mentre invece quando si deve illuminare LD3 e/o LD4, si manda la predetta DRV a zero logico e si mettono LED1 e/o LED2 a livello alto.



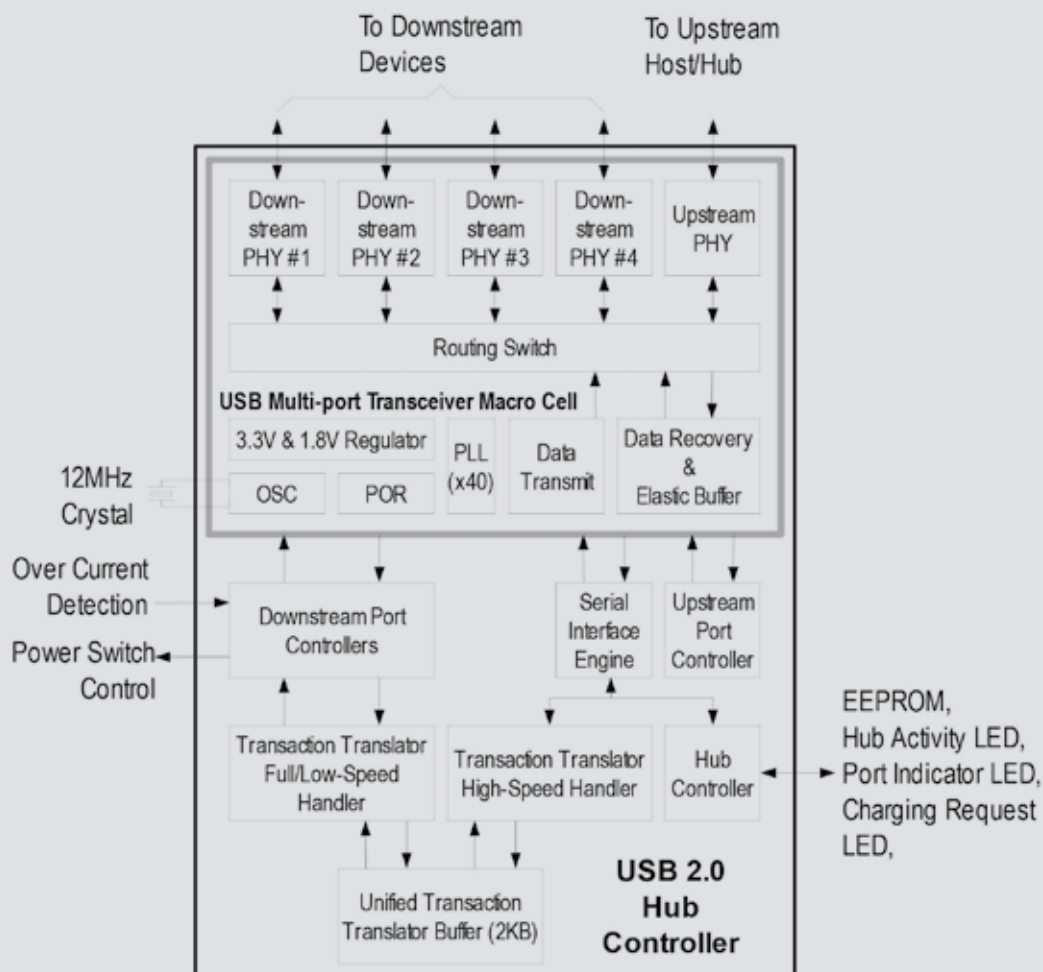
Il PCB visto dal retro: la porta di upstream è il connettore microUSB.

Il nostro hub è praticamente tutto contenuto in un integrato contenente le sezioni di interfaccia Universal Serial Bus lato ingresso e uscita; esternamente, il nostro chip richiede davvero un minimo di componenti, che sono essenzialmente un quarzo per il clock che definisce la velocità delle porte in downstream (uscite per i device). L'integrato è pienamente compatibile con la specifica Universal Serial Bus Revision 2.0 (USB 2.0) e supporta, lato Host (Upstream Facing Port) la velocità massima dell'USB 2.0 (480MHz) e la modalità Full-Speed, mentre dal lato device (4 porte di downstream) è compatibile High-Speed, Full-Speed e Low-Speed (12MHz). È inoltre compatibile con il profilo Universal Charging Solution e con l'USB Battery Charging Specification 1.1.

Il componente integra quattro transceiver USB 2.0 e i resistori di pull-up sulle linee dati (sull'upstream sono 1,5 kΩ e così pure downstream). Il relativo schema a blocchi è riportato in questo riquadro e in esso possiamo vedere che all'interno del

chip c'è un blocco di Power-On-Reset ed uno di rilevamento del power failure. Il funzionamento dell'insieme è scandito da un oscillatore quarzato interno con clock principale a 12MHz con registro di feedback e controllo da quarzo esterno (i condensatori sono all'interno); per generare il clock per la massima velocità USB 2.0 è integrato un PLL che partendo dai 12MHz arriva a 480MHz. Integrata è anche la funzione di rilevamento automatico della connessione di dispositivi portatili per fornire il supporto UCS. Il blocco single Transaction Translator (Single TT) è unico per tutte le porte in uscita (downstream ports) e l'alimentazione per tutte le sezioni viene garantita da due regolatori di tensione LDO da 3,3V e 1,8V che funzionano a partire dai 5 volt presenti sul piedino VDD5.

L'FE1.1 è disponibile in commercio in due package, uno dei quali è l'SSOP da 14 pin per lato utilizzato nel nostro progetto mentre l'altro è un 24 pin WQFN delle dimensioni di appena 4x4 mm con pad metallico centrale.



Con la prima uscita si gestiscono i LED indicati che un device risulta connesso alle porte downstream 1 e 3, ovvero LD1 riguarda l'attività sul connettore USB1 e LD3 quella sul connettore USB3; quando a LD2, mostra la presenza di device collegati a USB2 (LD2) e al connettore USB4 (LD4).

Ciascuna coppia di LED assorbe una corrente limitata dal resistore che si trova in serie (R1 per D1/D3 ed R2 per D2/D4) e che chiaramente interviene su un solo diodo per volta, giacché, trovandosi, i LED, in antiparallelo, quando conduce uno l'altro è interdetto e viceversa, pertanto nel resistore fluisce sempre la corrente di un diodo per volta.

Procediamo con l'analisi del circuito andando al resistore R6, che imposta la corrente di bias degli stadi analogici interni all'integrato.

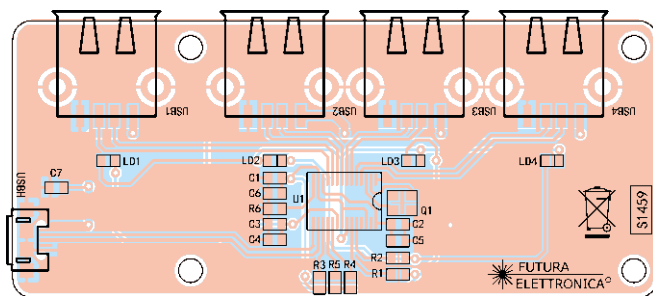
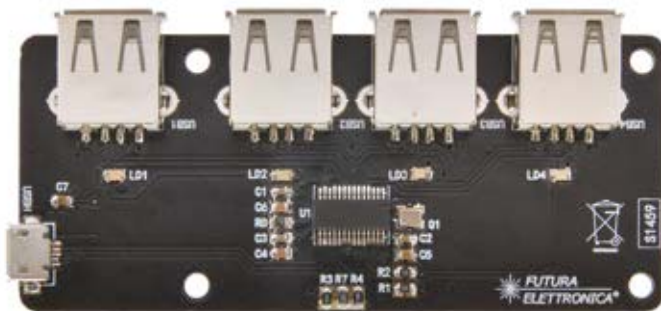
Concludiamo la descrizione dello schema elettrico con i condensatori C2 e C5, posti tra il piedino VD18 e massa, i quali, al pari di quelli applicati ai piedini VDD3 (tra essi e la massa) servono a filtrare la tensione generata dal regolatore a 1,8V interno all'integrato.

REALIZZAZIONE PRATICA

Bene, spiegato lo schema elettrico, passiamo ora alle note costruttive, iniziando dal circuito stampato sul quale prenderanno posto tutti i componenti occorrenti; questo è stato disegnato in modo da consentire il montaggio a sandwich su varie schede, comprese le Raspberry Pi e la piccola e semplice Raspberry Pi Zero: allo scopo, i quattro fori di fissaggio della basetta sono disposti analogamente a quelli delle predette schede. Per rispettare il passo di foratura delle Raspberry Pi, come vedete dalle fotografie del prototipo pubblicate in queste pagine, la USB1 non è stata posta equidistante dalla vicina USB2, come sono invece le altre tre prese (USB2 dista dalla 3 quanto la USB3 dista dalla USB4...) e questo perché nel mezzo dovrà starci la colonnina di montaggio utile per sovrapporre i due stampati e fissarli l'uno all'altro.

Il PCB da noi progettato è del tipo a doppia faccia e va realizzato per fotoincisione dopo aver ottenuto le necessarie pellicole dalle stampe delle rispettive tracce rame su carta da lucido o acetato. I componenti richiesti per la realizzazione del

piano di montaggio



Elenco Componenti:

- R1, R2: 470 ohm (0805)
- R3, R4, R5: 100 kohm (0805)
- R6: 2,7 kohm 1% (0805)
- C1, C2, C3: 10 μ F 6,3V ceramico (0805)
- C4, C5, C6, C7: 100 nF ceramico (0805)
- LD1, LD2, LD3, LD4: LED verde (0805)
- Q1: Quarzo 12 MHz (3,2 x 2,5 mm)
- U1: FE1.1s
- USB1, USB2, USB3, USB4: Connettore USB-A femmina da CS
- USBH: Connettore micro-USB

Varie

- Circuito stampato S1459 (86x37 mm)

Fig. 2
Il contenitore da noi stampato in 3D con la scheda posizionata all'interno.

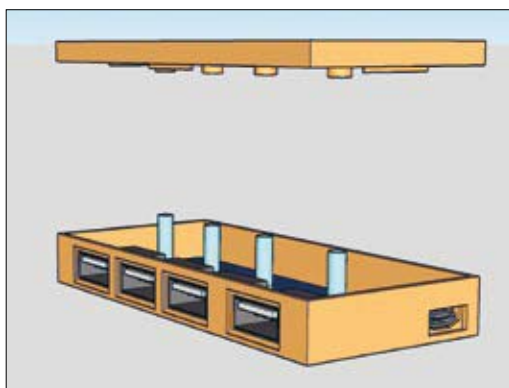


Fig. 3
Vista dal basso dell'hub nel suo contenitore stampato in 3D.

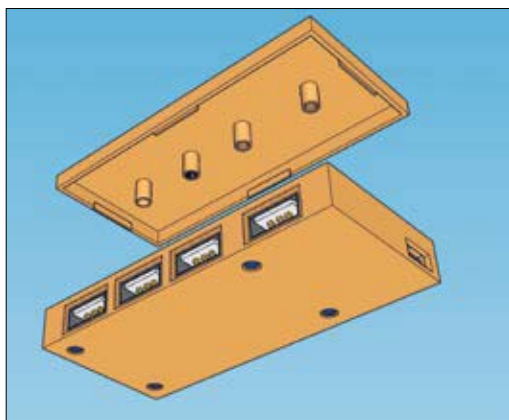
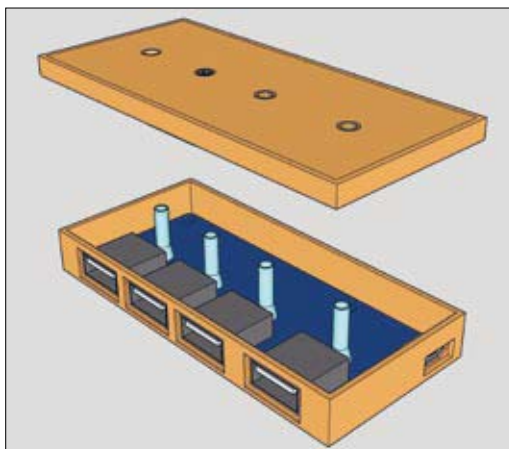


Fig. 4
Vista interna del contenitore per l'hub con la sua scheda.



nostro hub USB sono tutti per montaggio superficiale ad eccezione delle quattro prese USB-A, le quali sono ibride in quanto i pin dei contatti si montano in superficie e le alette di ancoraggio sono a foro passante.

Per il montaggio della scheda dell'hub avete bisogno di un saldatore da 20÷30W a punta fine per integrati, filo di lega saldante da 0,5 mm max di diametro, della pasta flussante di bassa densità (va bene anche quella reperibile in siringa, tipo

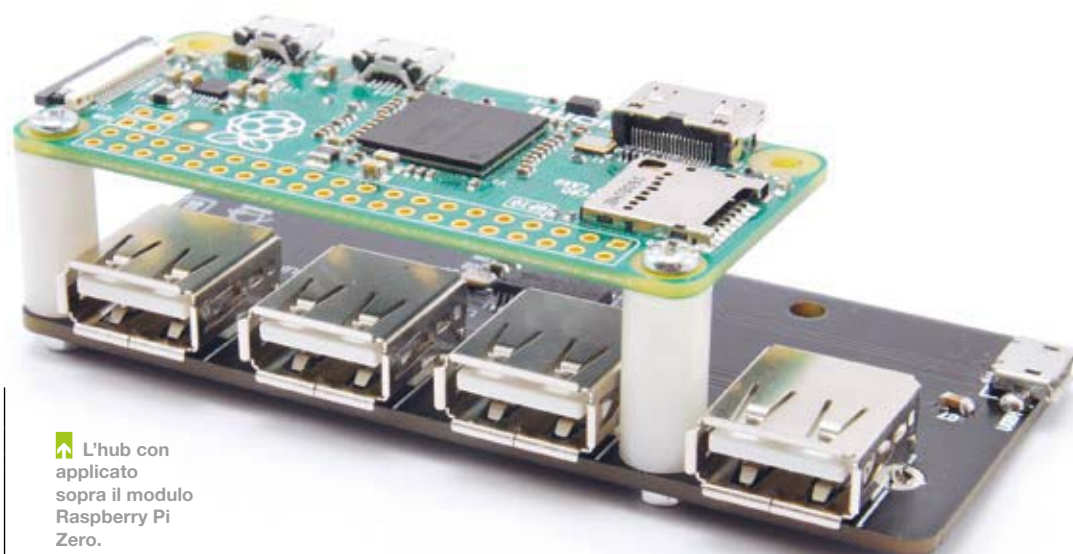
Kester o Amtech) una pinzetta per posizionare i componenti e una lente di ingrandimento per verificare, a fine delle saldature, che tutto sia al proprio posto e che non vi siano "baffi di stagno" a unire componenti o piazzole estranei tra loro. Una volta in possesso di tutti i componenti, spalmate le piazzole dell'integrato con il flussante e, posizionato il componente con il riferimento orientato come mostra il piano di montaggio che trovate in queste pagine, iniziate a stagnarne i piedini agli angoli, quindi procedete con gli altri pin uno alla volta alternando la saldatura dai due lati. Volendo, in alternativa potete procedere con una stazione ad aria calda, staginando prima le piazzole (fatelo delicatamente, altrimenti il calore e l'azione meccanica della punta del saldatore potrebbero farle saltare via) dell'integrato, quindi spalmando della pasta flussante e deponendo il componente sulle piazzole così trattate; fatto ciò, centrate l'integrato, stagnatene un pin di un angolo e poi scaldate i terminali con il getto d'aria calda, facendolo girare intorno ai piedini finché non vedrete lo stagno diventare brillante e i pin affondare in esso. A questo punto rimuovete l'aria calda e lasciate raffreddare.

Avete così sistemato il componente più critico e potete passare agli altri, che sono tutti di facile montaggio con saldatore e filo di lega saldante; al termine montate il connettore microUSB, spalmando il flussante sulle sue piazzole e sui suoi terminali per agevolare la deposizione della lega saldante, quindi staginate i pin uno per volta badando di non unirli con lo stagno. Curate la stagatura delle alette di ancoraggio, altrimenti nell'uso dell'hub, dopo un paio di inserzioni del connettore maschio vi ritroverete la presa spostata o, peggio, con le piazzole strappate.

Gli USB tipo A si montano inserendone per prima cosa a fondo nei rispettivi fori le linguette di ancoraggio finché sia il corpo che i piedini dei contatti (posti sul retro) non siano bene appoggiati alla superficie del PCB, quindi staginando le alette abbondantemente e saldando solo dopo i quattro contatti.

Fatto ciò il montaggio del circuito è completato: non resta che fare una verifica con la lente di ingrandimento per accertare che non vi siano cortocircuiti tra piazzole o piste attigue, facilmente rimovibili con l'aiuto del saldatore e di un pezzo di treccia dissaldante.

A questo punto potete pensare a un contenitore per l'hub o all'abbinamento e montaggio a sandwich su Raspberry Pi Zero; per il contenitore



 L'hub con applicato sopra il modulo Raspberry Pi Zero.

potete orientarvi verso quello da noi ideato, che potrete stampare da voi in PLA o ABS se possedete una stampante 3D come la 3Drag, a partire dai modelli che potete scaricare dal nostro sito www.elettronica.in.it, insieme agli altri file del progetto, sotto forma di file **.STL**. Nella **Fig. 2** e nella **Fig. 3** vedete due rendering di tale contenitore completo di coperchio da due angolazioni differenti e in quella dal basso sono evidenti le viti di fissaggio dei distanziali.

Per stabilizzare la scheda e mantenerla ben orizzontale anche quando si inseriscono e si sfilano i connettori USB per i device, il contenitore da noi disegnato prevede delle colonnine da incastrare nei blocchi a tubo del coperchio (**Fig. 4**).

Se pensate all'integrazione con la Raspberry Pi Zero o con altri apparati, considerate che per l'interconnessione occorre un cavetto con connettori microUSB maschio alle due estremità, per connettere la porta di ingresso (upstream) al dispositivo da abbinare; la presa microUSB Host dell'hub è laterale e va collegata a una delle due microUSB della Raspberry Pi Zero.

Invece volendo abbinare l'hub a una scheda Raspberry Pi 2, 3 ecc, il cavo da utilizzare è da un lato microUSB e dall'altro USB-A.

CONCLUSIONI

Il piccolo hub USB che vi abbiamo descritto assolve agli stessi compiti cui assolvono gli hub USB 2.0 che trovate in commercio e permette quindi di connettere fino a quattro dispositivi USB (per esempio stampanti, mouse e tastiere, scanner...) in modalità device a un'unica porta USB, quale può essere quella di un Personal Computer.

Provvede anche all'alimentazione dei device, a patto che assorbano, tutti insieme, la corrente massima prelevabile dalla USB del computer e supportabile dalla presa microUSB che costituisce la porta d'ingresso.

Per poter contare su una corrente più elevata alle porte di uscita bisognerebbe interrompere la pista che trasporta il +5V dalla presa microUSB alle quattro USB-A e alimentare la linea +5V di queste ultime con un alimentatore esterno, capace di erogare tutta la corrente che lo standard Universal Serial Bus 2.0 contempla per le porte USB; la cosa è perfettamente fattibile, anche se non è stata fatta nel nostro hub perché il circuito non nasce con tale scopo.

Il circuito potrà comunque essere utilizzato come hub stand-alone o integrato in apparati esistenti, magari abbinato a una board Raspberry Pi. 



Cosa occorre?

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

Il gerber del PCB è scaricabile gratuitamente dal sito della rivista, mentre il chip FE1.1S (cod. FE11S) è in vendita al prezzo di Euro 1,50 su www.futurashop.it. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

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

LA USB DALLA A ALLA C

di DAVIDE SCULLINO

Scopriamo l'evoluzione di uno dei bus di comunicazione più affermati che ha saputo tenersi al passo coi tempi.

Universal Serial Bus, meglio noto come USB, soprattutto agli utenti finali, è un bus di comunicazione seriale molto semplice e oramai tanto diffuso da essere presente praticamente in qualsiasi apparato elettronico che necessita di una connettività cablata ed anche di veicolare l'alimentazione necessaria al funzionamento e alla ricarica delle eventuali batterie. L'affermazione di questo link si deve sia alla velocità di comunicazione che permette, sia alla modalità plug & play (che permette ai computer e in generale agli host di identificare i dispositivi connessi alle porte) e alla possibilità di affacciare su un bus fino a 127 dispositivi slave. Proprio la diffusione, dovuta ai punti di forza, ha spinto a mantenere, dopo anni, e ad evolvere adattandolo alle crescenti esigenze della tecnologia, il bus USB, che è passato

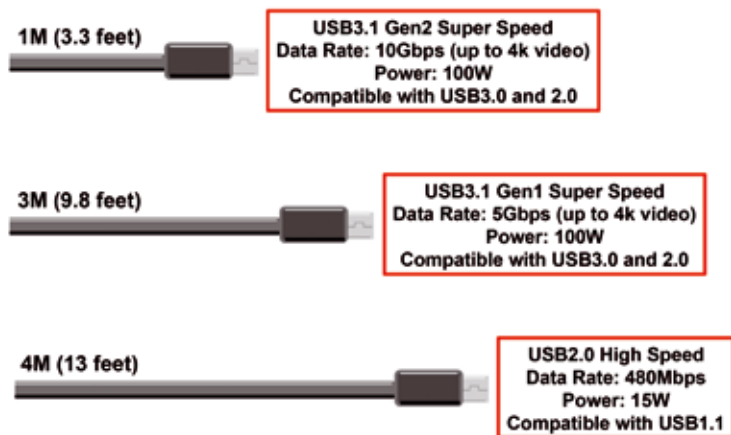


Fig. 1
Caratteristiche e
distanze coperte
dalle varie versioni
di USB.

dalla sua prima implementazione fino all'attuale USB Type-C, che possiamo considerare lo stato dell'arte e che in realtà è divenuta qualcosa di più di una USB, perché è una connessione fisica ospitante più protocolli.

Nel frattempo la connessione USB, nata a bordo dei Personal Computer, è sbarcata sui dispositivi mobili ed è approdata all'automobile, dove ormai è indispensabile strumento di connettività per applicazioni di car entertainment e non solo.

In questo articolo riepiloghiamo le caratteristiche dell'USB, percorrendone le tappe dell'evoluzione e concentrandoci poi sull'USB Type-C.

TECNOLOGIA DELL'USB

Prima di tracciare la cronistoria dell'USB è opportuno un rapido richiamo all'aspetto puramente elettronico del bus, il quale è composto da due

linee, chiamate D+ e D-, che costituiscono il canale dati bidirezionale; la connessione riporta anche l'alimentazione a 5Vcc, che serve per alimentare i Device con il dispositivo Host, secondo le specifiche del protocollo. Proprio in virtù della presenza dell'alimentazione, l'USB è diventato anche uno standard di alimentazione, tanto che il relativo connettore viene adottato persino negli alimentatori e nei caricabatterie, talvolta per veicolare i soli 5 volt.

Il connettore di base ha dunque 4 poli (questo vale per l'USB A e il B) ma nell'evoluzione USB 3 ne conta 8, giacché trasporta due canali dati invece di uno (allo scopo di accrescere la velocità di comunicazione) e il doppio della corrente. Esistono poi le connessioni microUSB, che oltre ad essere miniaturizzate contano due piedini in più uno dei quali permette di comunicare all'Host se deve utilizzare una periferica come Device o meno, ovvero che attiva la porta USB in modalità Host.

Per inciso, Host e Device sono le due modalità previste dall'USB, cioè i due modi in cui un dispositivo USB può funzionare: Host è l'unità master, ossia quella che governa la comunicazione, mentre Device è il dispositivo puro e semplice, ossia una periferica, che viene gestita dall'Host, il quale avvia la comunicazione con essa. Inoltre a un Host possono essere collegati più Device, perché può indirizzarli, mentre un Device si collega a un solo Host.

La distanza tra gli apparati connessi in USB dipende dalla qualità del cavo, ma tipicamente è contenuta in 5 metri, almeno se si vuole sfruttare il

Fig. 2
Prestazioni e
connettori
delle varie USB.

	Version	Speed	Bits/sec	HD movie 25GB
	USB 1.1	Low speed (LS)	1.5Mbps	~ 9.25 hours
		Full speed (FS)	12 Mbps	
	USB 2.0	High speed	480 Mbps	~14 mins
	USB 3.0	SuperSpeed (SS)	5 Gbps	~70 sec
	USB 3.1 Gen1			
	USB 3.1 Gen2	SuperSpeedPlus (SSP)	10 Gbps	~35 sec



massimo data-rate, specialmente nelle USB 2.x. La **Fig. 1** schematizza il concetto, proponendo un confronto tra versioni di USB e rispettive connessioni elettriche.

USB: ORIGINI E SVILUPPI

Quando nacque, l'USB aveva la connessione standard a quattro linee rimasta fino alla revisione 2 e forniva una velocità di comunicazione relativamente bassa, che non superava i 40 Mbps: ci riferiamo alla versione 1.0, che peraltro non fu mai utilizzata realmente, perché rimase in laboratorio fino allo sviluppo della USB 1.1, prima implementazione su apparati reali, capace di un data-rate di 60 Mbps, corrispondenti a circa 8 MB/s, che possono spingersi fino a 12 Mbps. L'interfaccia Host erogava fino a 500 milliampere disponibili per alimentare Device come periferiche HID, memorie di massa ecc.

La più rilevante evoluzione fu l'USB 2.0, che portò il data-rate a 480 Mbps (60 MB/s) mantenendo la stessa connessione e la corrente di uscita, che però nelle release successive crebbe fino ad 1 ampere, arrivando anche ai 2.

Altra caratteristica dell'USB 2.x è che nelle connessioni con connettore microUSB e relativo cablaggio supporta la modalità OTG, ossia USB On The Go: questo significa che portando a massa un pin del connettore microUSB, che ha 5 poli invece dei classici quattro, il dispositivo mobile che dispone della porta (tipicamente un tablet o uno smartphone) può comportarsi come Host invece che da Device, quindi gestire dei dispositivi di memoria di massa come le Pen Drive USB. Infatti mentre nei Personal Computer e nei TV dotati di supporto alle memorie di massa la porta USB è un Host, nei dispositivi mobile come tablet e smartphone (o anche cellulari e strumenti dotati di microUSB) la USB è di tipo Device e viene utilizzata per la ricarica della batteria e l'accesso alla memoria interna da parte di un Host quale un Personal Computer.

Mediante un opportuno cavo che connette il pin 4 (Mode Detect) del connettore a massa è quindi possibile convertire il profilo della porta USB del dispositivo da Device a Host.

Venne poi l'USB 3, capace, anche grazie al raddoppio dei canali dati, di raggiungere velocità di comunicazione decuplicate, ossia 4,8 Gbps (corrispondenti a 600 MB/s) utilizzando due coppie intrecciate ad alta velocità; con questa tecnica e il data-rate teorico, la specifica ritiene ragionevole ottenere almeno 3,2 Gbit/s al netto dell'overhead di protocollo (corrispondenti a 400 MB/s). Siamo

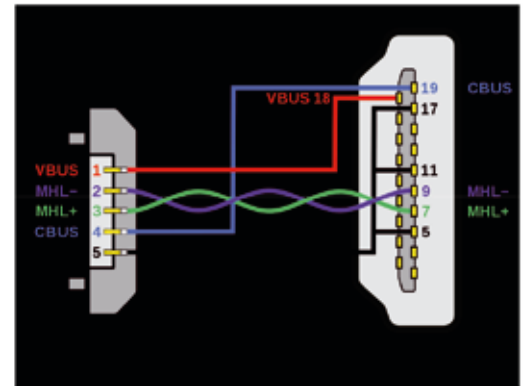
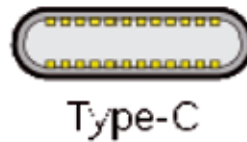


Fig. 3
A sinistra, il connettore tipo C visto dal lato di inserimento. A destra la connessione MHL di una microUSB con una HDMI.

quindi arrivati all'USB Type-C, che rappresenta lo stato dell'arte.

La **Fig. 2** riepiloga le principali caratteristiche e le connessioni delle varie versioni di USB: nella tabella, l'ultima colonna propone, per il confronto tra i data-rate delle varie versioni, il tempo richiesto dal trasferimento di un filmato HD da 25 GB.

USB TYPE-C: LO STATO DELL'ARTE

Entriamo adesso nel dettaglio dell'USB di tipo C, che riveste molto interesse sia per la velocità di comunicazione che rende possibile, sia per la capacità di alimentare i device ad esso collegati, con potenze anche consistenti.

Sul piano della velocità, questo standard è capace di offrire un transfer-rate 10 Gbit/s (1.250 MB/s) nominali.

La USB tipo C supporta 5 profili di potenza in base alla connessione fisica:

- 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).

Altra caratteristica rilevante e innovativa dell'USB Type-C è che non solo consente di scambiare dati a velocità molto elevata e di farlo con una linea full-duplex dedicata e differente da quella degli standard USB precedenti, ma offre anche compatibilità e link di trasporto fisico configurabili per protocolli dati di vario genere; il tutto in un'unica connessione elettrica.

Con lo standard USB C è stato introdotto il nuovo connettore Tipo C che, 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

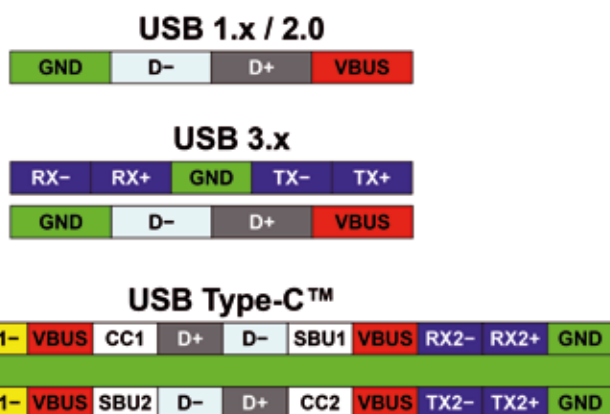


Fig. 4
Contatti delle varie versioni di USB.

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 saranno aggiornati con i pin VBUS 12 e 20 V, ma il connettore tipo A supporterà i 10 Gbit/s.

Nei nuovi dispositivi i connettori di tipo A e C saranno di colore Blu/Acqua se utilizzabili sia per l'alimentazione che per la trasmissione dati, mentre il connettore C sarà di colore nero se utilizzabile soltanto per alimentazione.

Il nuovo connettore tipo C (**Fig. 3 a sinistra**) 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 (**Fig. 3 a destra**).

I due nuovi pin aggiunti per il VBUS a 12V e 20V, devono essere alimentati tramite un alimentatore esterno perché gli altri connettori USB non prevedono questi PIN.

Quindi, attualmente non è possibile realizzare alimentatori da incasso come con l'USB 2.0, sempre

se non si realizza un'appendice per i nuovi pin, in stile PoweredUSB.

Il connettore Type C supporta modalità alternative come DP, HDMI, Thunderbolt ecc.

La **Fig. 4** permette di valutare l'evoluzione della connessione USB relativamente al numero di contatti: come vedete si è passati dai 4 delle USB 1 e 2 agli 8+1 della USB 3.x, fino ad arrivare alla Type-C, che conta più contatti disposti su due file affiancate l'una sull'altra; per l'esattezza sono 12 per fila.

La **Fig. 5** mostra i tipi di connettore più recenti, ossia da quelli dell'USB 2.0 in avanti.

Il connettore maschio Type-C è reversibile, cioè può essere inserito nella presa corrispondente in qualsiasi verso, senza che ciò pregiudichi né l'integrità, né il corretto funzionamento degli apparati. L'interfaccia USB 3 nelle sue versioni è tuttora molto affermata e si sta diffondendo, come standard di comunicazione, non solo sui computer e dispositivi mobile, ma anche nell'automotive, dove si candida a diventare la connettività d'elezione per l'infotainment (USB 3.1) e il collegamento di unità di memoria di massa e di telefoni, tablet ecc.

Per questo sono in sviluppo evoluzioni della USB 3, vale a dire la USB3.2, capace di un data-rate di 20 Gbps e la USB4.0, che salirà a 40 Gbps (2x20 Gbps); tutte finalizzate al rapido trasferimento di file di grandi dimensioni come quelli video in alta risoluzione.

Le applicazioni presenti e future spaziano inoltre fino a VR Console, Mobile HotSpot, Facial Recognition, Infotainment, VOIP, Electric Whiteboard, STB, Smart TV, iPC, Gateway, ATM Machines, Routers, IOT, POS, Karaoke, Gaming, Kiosks, Charging Stations, Surveillance, Robotics, Imaging.

Tra l'altro la USB Type-C può ospitare, sulla stessa connessione fisica anche lo scambio di dati secondo protocolli differenti da quello nativo, ed è proprio per questo che è molto appetibile e rappresenta una potenzialità per le applicazioni future. Come mostra la **Fig. 6**, oltre ai classici D+ (Dp) e D- (Dn) sdoppiati e disposti in modo che qualunque sia il verso d'inserzione del connettore combacino



Fig. 5
Connettori USB 2 e 3; il Type-C è anche detto USB-CTM.

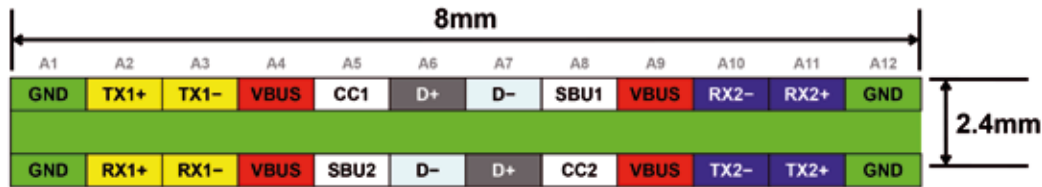


Fig. 6
Significato dei
contatti del
connettore USB
tipo C.

Rx1n/Rx1p = Differential Pair 1
Tx1n/Tx1p = Differential Pair 1
Rx2n/Rx2p = Differential Pair 1
Tx2n/Tx2p = Differential Pair 1
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

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 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

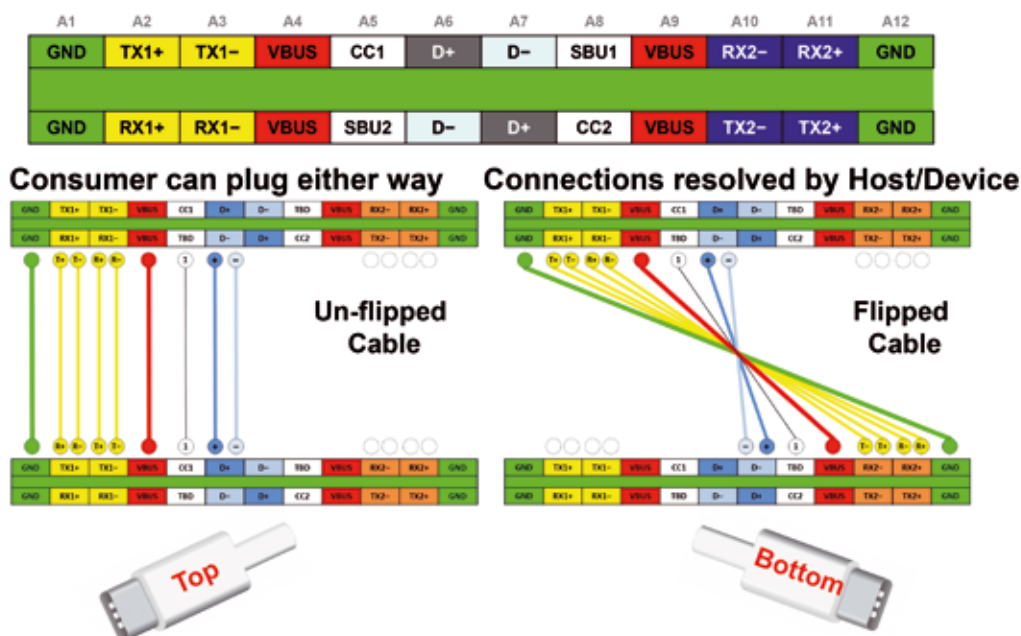


Fig. 7
Modalità di
connessione
supportate
dall'interfaccia USB
tipo C.

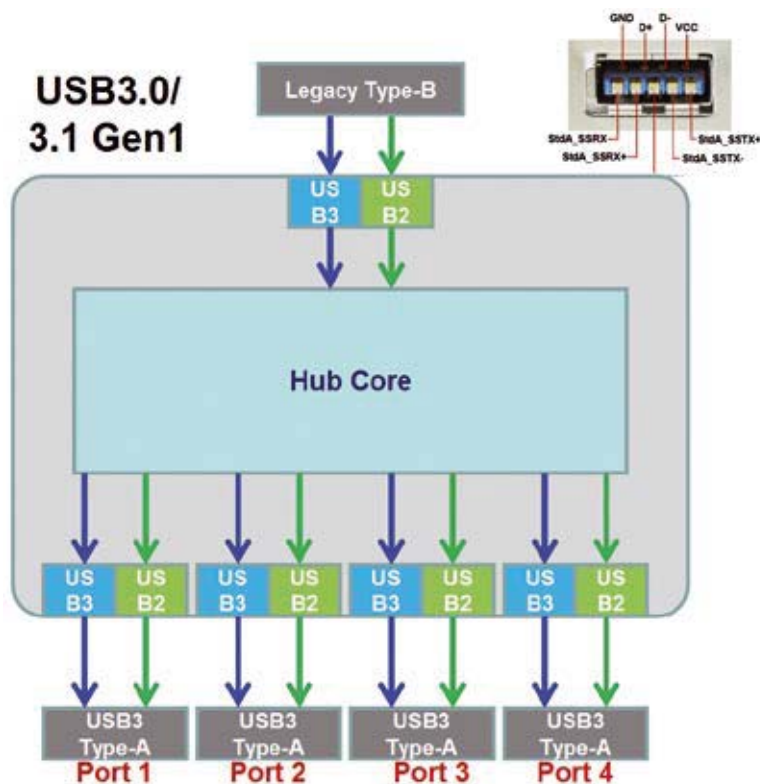


Fig. 8
Schematizzazione
dell'hub 3.1 e
connessioni del
connettore USB 3.x.

Queste due linee consentono la comunicazione full-duplex, aumentando la velocità di scambio dei dati raggiungibile.

La **Fig. 7** schematizza alcune modalità di connessione ottenibili sfruttando i molteplici canali dati di cui l'USB Type-C dispone.

La connessione USB D+/D- tradizionale serve a garantire la retrocompatibilità, ovvero la possibilità di comunicare anche con dispositivi come quelli USB 2.x.

Il supporto ad altri protocolli si ottiene con quella che nello standard USB Type-C viene chiamato "Alternate Modes" e permette di gestire la comunicazione secondo protocolli di terze parti (Thunderbolt, DisplayPort, HDMI ecc.).

Negli "Alternate Modes" la comunicazione è negoziata utilizzando l'USB PD one-wire serial protocol ed è abilitata mediante l'handshake di protocollo tramite messaggi strutturati definiti dal fornitore (VDM). Le modalità alternative sono associate ai fornitori (ad esempio Tramite VID) o ai gruppi/entità Standard (per esempio Tramite SID).

Chiamiamo la pluralità di VID e SID che possono esistere all'interno di un dispositivo PD: "SVID".

ALIMENTAZIONE DA USB/USB PD/PD

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.

Per fare un confronto basato sulla potenza erogabile tra le varie versioni di USB, proviamo a immaginare di dover ricaricare un accumulatore da 3.500 mAh con la corrente erogabile; nel caso della USB2.0, a 500mA si impiegano 7 ore, che diventano 2,3 potendo erogare 1,5 A se si dispone di una USB2.0 BC1.2.

Impiegando una USB Type-C nelle due precedenti modalità di erogazione di potenza valgono le stesse tempistiche, ma se si negozia la modalità USB2.0 PD @ 12V, dove quindi l'host eroga al device 12 volt invece che 5, la solita batteria da 3.500mAh si può caricare con 3 ampere in soli 36 minuti.

Negoziando la modalità a 20 volt, in modalità USB2.0 PD @ 20V, la batteria si può caricare in appena 13 minuti, erogando 5.000 mAh.

Chiaramente il device dev'essere un USB C e deve poter essere in grado di accettare tali tensioni e convertirle internamente mediante un DC/DC; ciò viene stabilito, appunto, durante la negoziazione. I tempi indicati nei due casi di ricarica con tensione erogata di 12 e 20 volt sono calcolati ipotizzando di non avere perdite nella conversione e si basano sulla potenza trasmessa, che a 12V con 3.000 mA sono 36 watt e a 20V con 5.000 mA, ben 100 W.

HUB USB 3.1

Vediamo adesso come è composto tipicamente un hub USB ed in particolare quello conforme USB 3-1, premettendo che al pari di tutti gli hub USB dall'1.1 in avanti serve a espandere una singola porta Universal Serial Bus in più connessioni così da collegare più device a un host e da alimentarle in parallelo. Notare che l'hub non moltiplica le porte (e nemmeno gli host) ma le connessioni, cioè consente il collegamento di più periferiche ad un unico host; non permette, invece, il collegamento tra host.

La tipica struttura a blocchi dell'hub USB 3.1 è mostrata nella **Fig. 8**.

Nel contesto si collocano concetti come DFP (Downstream Facing Port) ossia la porta dell'Host



il cui ruolo (cioè la funzione che dovrà assolvere secondo il protocollo) è Source, come può essere nel caso di PC laptop e desktop, console automotive; in pratica il connettore USB eroga corrente all'apparato che vi viene connesso.

Sul piano dell'alimentazione, a questo si contrappone l'UFP (Upstream Facing Port) che è una porta Device (come ad esempio quella di tastiere e mouse, Pen Drive ecc.) il cui ruolo preferenziale è Sink, nel senso che sotto l'aspetto del trasferimento di potenza sono predisposti per assorbire corrente dai pin di alimentazione del connettore USB.

Ebbene, un tipico hub USB contiene sia una UFP che una DFP e chiaramente la prima funziona in modo Sink, mentre la seconda opera in modalità Source.

Esistono tre tipi di hub: Stand Alone, Dock w/ Embedded Hub e Monitor w/ Embedded Hub.

Esiste anche una particolare porta chiamata DRP, capace di operare, sul piano dell'alimentazione, in modalità sia Source/DFP che Sink/UFP.

Quando non è connessa, la DRP alterna le modalità Source (DFP) e Sink (UFP) nel tentativo di stabili-

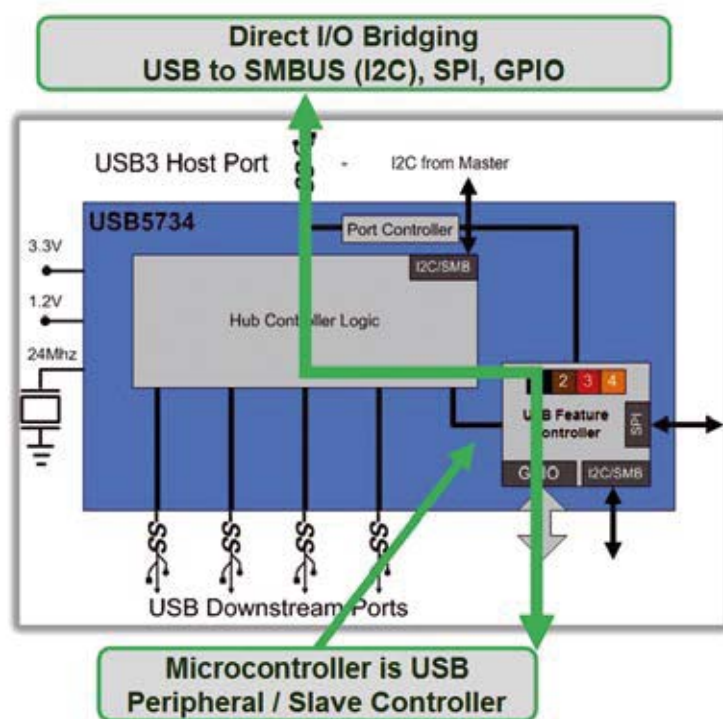


Fig. 9
Schema a blocchi dell'attività del feature controller USB.

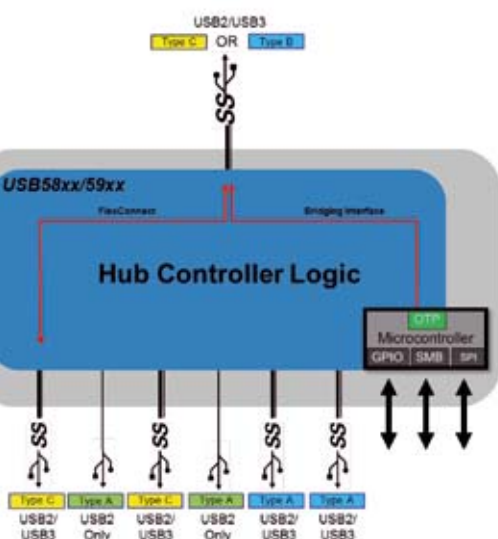
- re una a connessione; questo viene ottenuto da:
- connessione di una Rp (pull-up) a CC1/CC2 quando funziona da Source;
 - connessione di una Rd (pull-down) a CC1/CC2 quando è in modo Sink.

Il progettista può controllare il "duty-cycle" della commutazione per ottenere la risoluzione del ruolo preferito, ossia può decidere di impostare la durata, all'interno del ciclo di alternanza delle due modalità, per essere certo che venga agganciata più facilmente la modalità desiderata.

La possibilità di controllare le caratteristiche dell'hub mediante un apposito controller consente di implementare i cosiddetti Smart Hub.

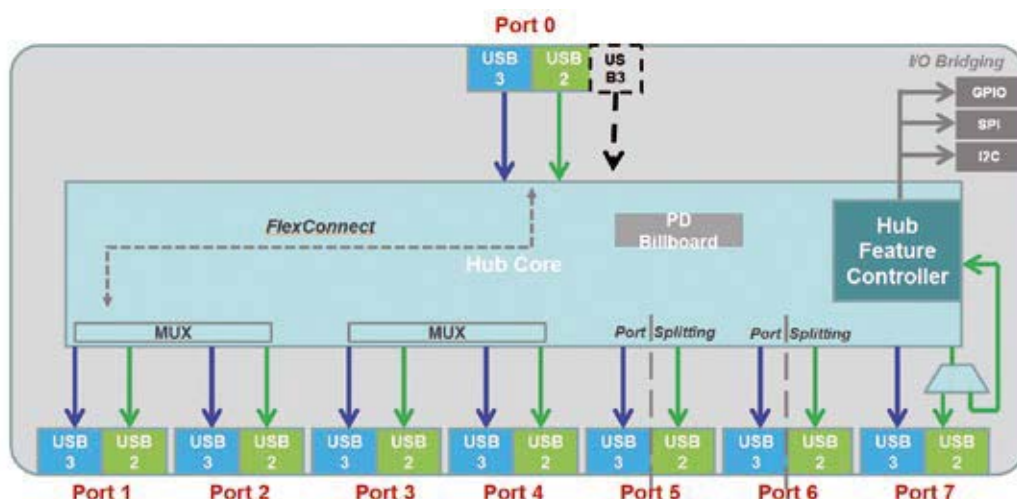
Fig. 10
Schema a blocchi dei controller USB58xx e 59xx Microchip.

Tabella 1
Caratteristiche dei controller della famiglia Microchip USB 58xx e della 59xx.



P/N	Upstream Configuration	Downstream Configuration	Battery Charging	I/O Bridging / FlexConnect	Package	Industrial Grade
USB5807	Type B	7 Type A	Yes	-	100-pin SQFN	Yes
USB5806	Type B	6 Type A	Yes	Yes	100-pin SQFN	Yes
USB5816	Type B	4 Type A / 1 USB-C	Yes	Yes	100-pin SQFN	Yes
USB5826	Type B	2 Type A / 2 USB-C	Yes	Yes	100-pin SQFN	Yes
USB5906	USB-C	6 Type A	Yes	Yes	100-pin SQFN	Yes
USB5916	USB-C	4 Type A / 1 USB-C	Yes	Yes	100-pin SQFN	Yes
USB5926	USB-C	2 Type A / 2 USB-C	Yes	Yes	100-pin SQFN	Yes

➔ **Fig. 11**
Hub USB 5+1
basato su soluzioni
Microchip.



CONTROLLER FUNZIONI E FEATURE DELL'HUB USB

Esegue funzioni di sistema (send and receive data) verso le periferiche esterne senza bisogno di una MCU esterna. L'USB Feature Controller abilita le seguenti funzioni:

- Direct I/O bridging
- SMBus, GPIO, UART, SPI
- Full function UART in USB5734
- FlexConnect – Host/Device function swapping.

Opzioni di configurazione flessibili attraverso:

- SPI FLASH/EEPROM;
- OTP FLASH;
- Strapping Options.

La **Fig. 9** schematizza il funzionamento dell'insieme controller funzioni.

Microchip fornisce due famiglie di controller, ossia la USB port upstream UFP – USB58xx (che include la USB5807) e la USB-C UFP – USB59xx. Le famiglie supportano anche queste funzionalità:

- Integrated 2:1 MUX for USB-C;
- SmartHub: FlexConnect e I/O Bridging;
- PortSplitting e Billboard class.

Nella **Tabella 1** sono riepilogate le principali caratteristiche delle due famiglie. Invece nella **Fig. 10** è proposto lo schema a blocchi delle due famiglie di controller, che operano supportando le funzionalità richieste.

MICROCHIP 7-PORT USB HUB

Una soluzione sia per USB2.0 che USB3.1 è quella proposta da Microchip e consente di realizzare

hub USB a 5+1 porte standard USB3.1 Gen 1. Il relativo schema a blocchi viene proposto nella

Fig. 11 e le caratteristiche principali sono:

- Firmware to Manage Power Delivery incl Billboard
- Native USB Type C Connector Support (CC pin);
- All Port FlexConnect;
- Port ID/Screening (Mini-Host), interfaccia I²S;
- Hub Feature Controller managing FlexConnect, I/O Bridging, Battery Charging, SPI Flash e OTP;

Il tutto in un singolo package QFN da 100 pin (25 per lato) a passo 0,4 mm pin-to-pin compatibile con la serie USB73xx ma con 10 Gbps di data-rate.

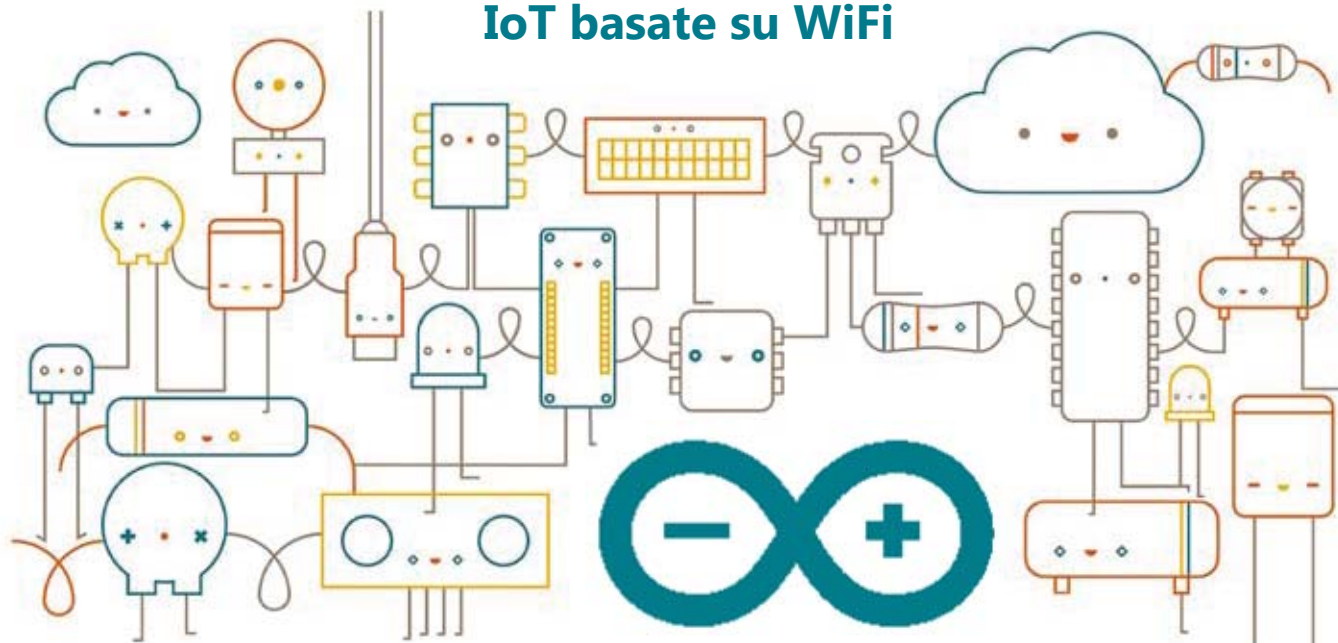
Il controllore UPD 350 è un 32 bit e supporta le centralized uC enable system feature features (PB). Le funzionalità delle porte Type C variano da un prodotto all'altro e la caratteristica FlexConnect per HostFlexing consente un rapido set-up e adattamento.

CONCLUSIONI

In questo articolo abbiamo fatto una carrellata sulla porta USB e come essa si sia evoluta negli anni, divenendo, con la versione C, più di una USB e non soltanto in termini di velocità, ma anche di flessibilità e compatibilità con altri protocolli e connessioni eterogenee, nell'ottica di far diventare la USB qualcosa di simile alla Thunderbolt, vale a dire una porta universale destinata a far transitare dati non solo da e verso memorie di massa o da dispositivi di acquisizione a computer, ma anche da questo a monitor HDMI e altre periferiche. □

ARDUINO PROFESSIONAL

Schede Arduino professionali progettate per
velocizzare e semplificare lo sviluppo di applicazioni
IoT basate su WiFi



Prezzi IVA inclusa.

**Arduino
MKRWAN
1300**

Offre le funzionalità di
Arduino Zero
e la connettività della
tecnologia LoRa™

€ 44,00

Cod. MKRWAN1300

**Arduino
UNO
WiFi
REV2**

Il WiFi integrato
facilita la connessione
a Internet. Dispone
di sensore IMU
(Inertial Measurement Unit)

€ 54,90

Cod. ARDUINOWIFI

**Arduino
MKRFOX
1200**

Ideale per
realizzare progetti
IoT (Internet of Thing)
basati su Sigfox

€ 43,90

Cod. MKRFOX1200

**Arduino
MKR
1000**

€ 45,00

Cod. MKR1000

**Arduino
MKR
VIDOR
4000**

Dotata di modulo Wi-Fi
u-blox, rende la tecnologia
FPGA accessibile
a tutti

€ 78,00

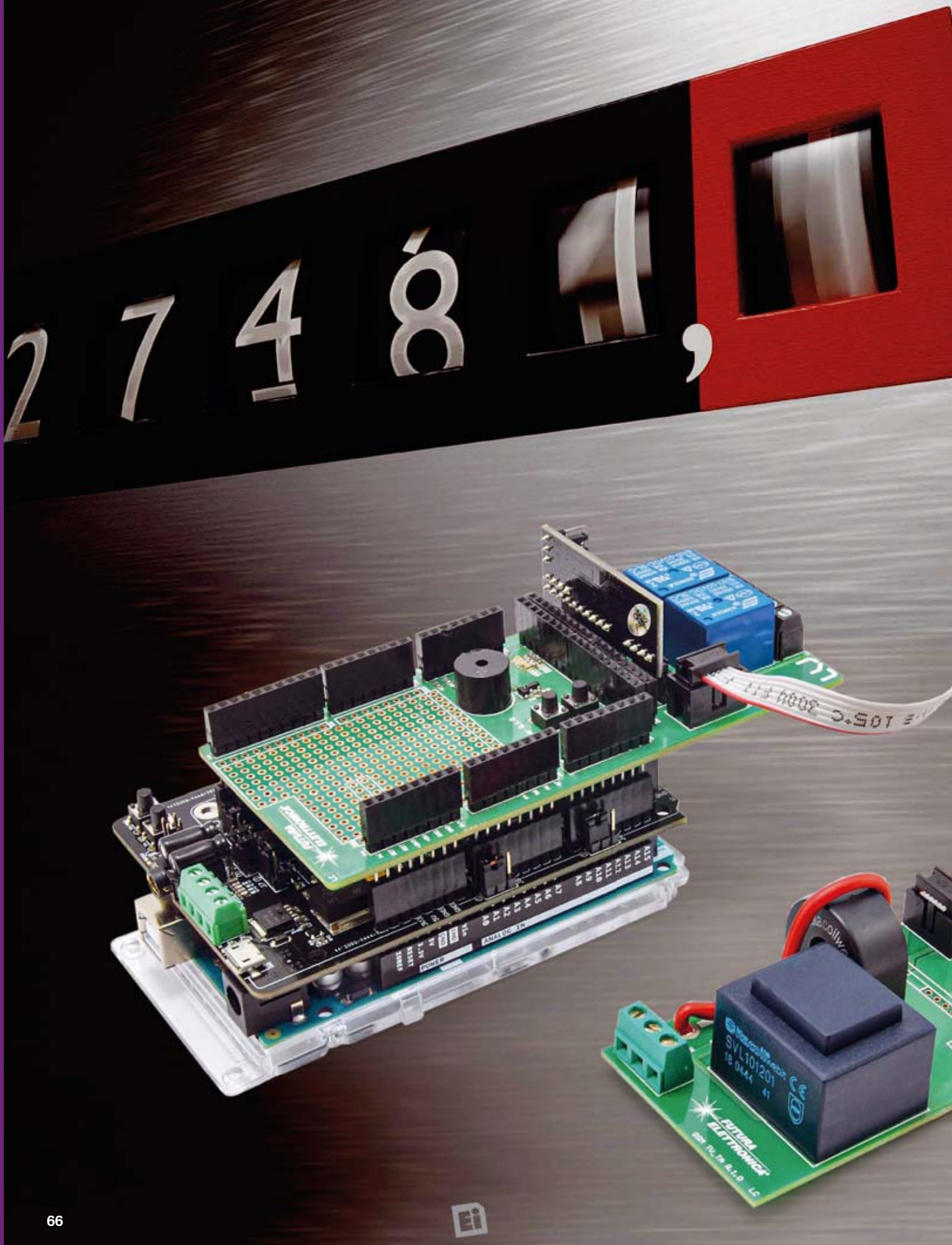
Cod. MKRVIDOR4000

**Arduino
MKR
WiFi
1010**

Progettata per
velocizzare e semplificare
la prototipazione di
applicazioni IoT basate
sulla connettività Wi-Fi

€ 34,00

Cod. MKRWIFI1010



kWh

ENERGY METER GSM

Monitora il consumo di elettricità e invia i dati rilevati nel cloud attraverso la rete cellulare GSM; è basato su una Arduino Mega equipaggiata di GSM Shield e su una scheda contenente il rilevatore di assorbimento. Prima puntata.

dell'Ing. **MATTEO DESTRO**

M

onitorare i consumi elettrici della propria abitazione o azienda è un tema sempre più attuale e le misure elettriche effettuate, raggruppate in una base di dati e quindi mostrate in forma grafica, permettono di valutare l'andamento nel tempo dei consumi. Ecco perché, sfruttando la

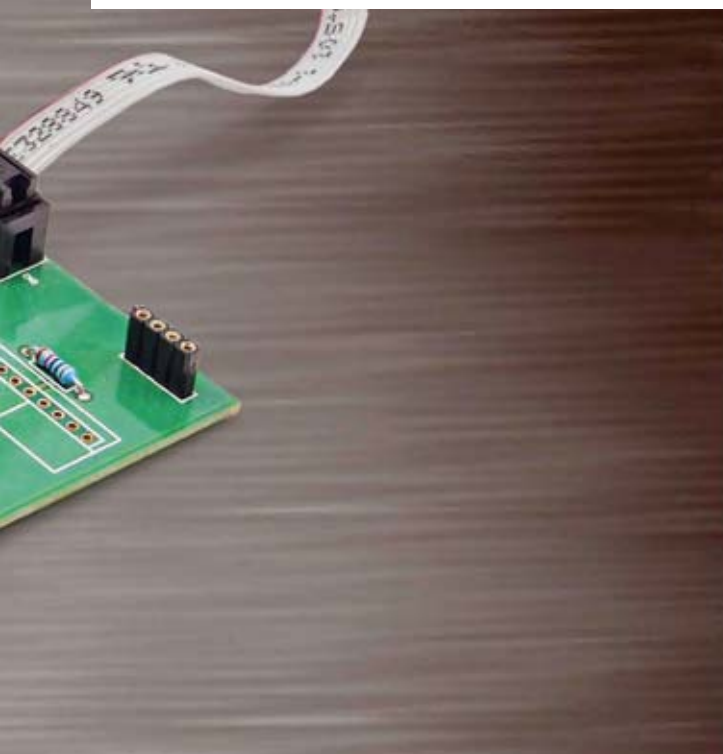
nostra scheda "energy meter" FT1346M (basata sull'integrato MCP39F511 e presentata nel numero 220 di Elettronica In) proporremo una serie di articoli in cui andremo a sviluppare un sistema di misura basato, in prima battuta, sulle schede Arduino e successivamente su altre architetture. Il tutto verrà abbinato al nuovo GSM Shield per aggiungere una connessione dati mobile e inviare le misure acquisite a un server remoto. Nel progetto descritto in queste pagine faremo uso della Arduino Mega 2560 (o della Fishino Mega) nella quale "girerà" uno sketch integrante la libreria di gestione dell'MCP39F511 e di un hardware dedicato per l'interconnessione della scheda FT1346M. L'hardware che abbiamo sviluppato ha le seguenti caratteristiche:

- connessione dati alla scheda FT1346M tramite interfaccia hardware UART 3 (I/O 14 e 15);
- connessione linee di controllo MCLR e RESET alla scheda FT1346M (I/O 46 e 45);
- connessione linea ZCD (Zero Cross Detection) alla scheda FT1346M (I/O A12 o 20);
- connessione TA e TV alla scheda FT1346M (TA e TV sono montati su una scheda ausiliaria);
- jumper di selezione connessione della linea ZCD alla scheda Arduino; connessione a I/O A12 per gestione tramite periferica PCINT20 oppure a I/O 20 tramite periferica INT1;
- due relé indipendenti per la gestione di eventuali carichi;
- buzzer per segnalazione acustiche a seconda delle condizioni operative;
- quattro LED, più due pulsanti per uso generico.

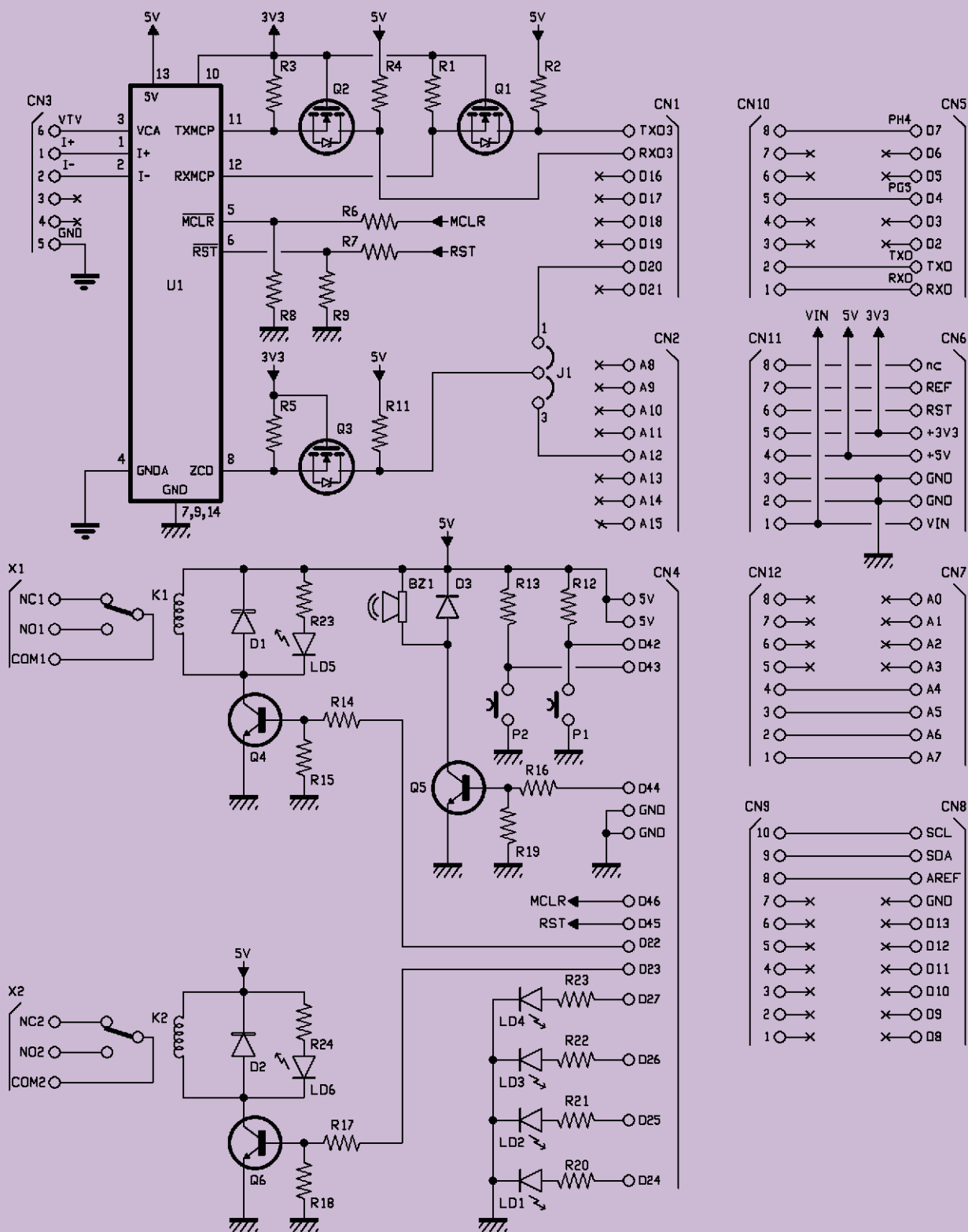
Il PCB prevede una piccola zona di prototipazione dove montare eventuali circuiti di test a componenti THT per testare specifiche applicazioni. Data questa semplice premessa passiamo alla descrizione sia dello schema a blocchi che dello schema elettrico.

SCHEMA A BLOCCHI HARDWARE

La **Fig. 1** mostra la composizione del sistema: la scheda qui presentata nasce per essere abbinata alla nostra scheda di



GSM ENERGY METER SHIELD



sviluppo GSM (cod. WWGSMFIELD) e di conseguenza collegata a una scheda Arduino Mega 2560 o Fishino Mega.

L'alimentazione alla scheda può arrivare direttamente da Arduino Mega, passando dalla GSM shield tramite cavo micro-USB, oppure sfruttando

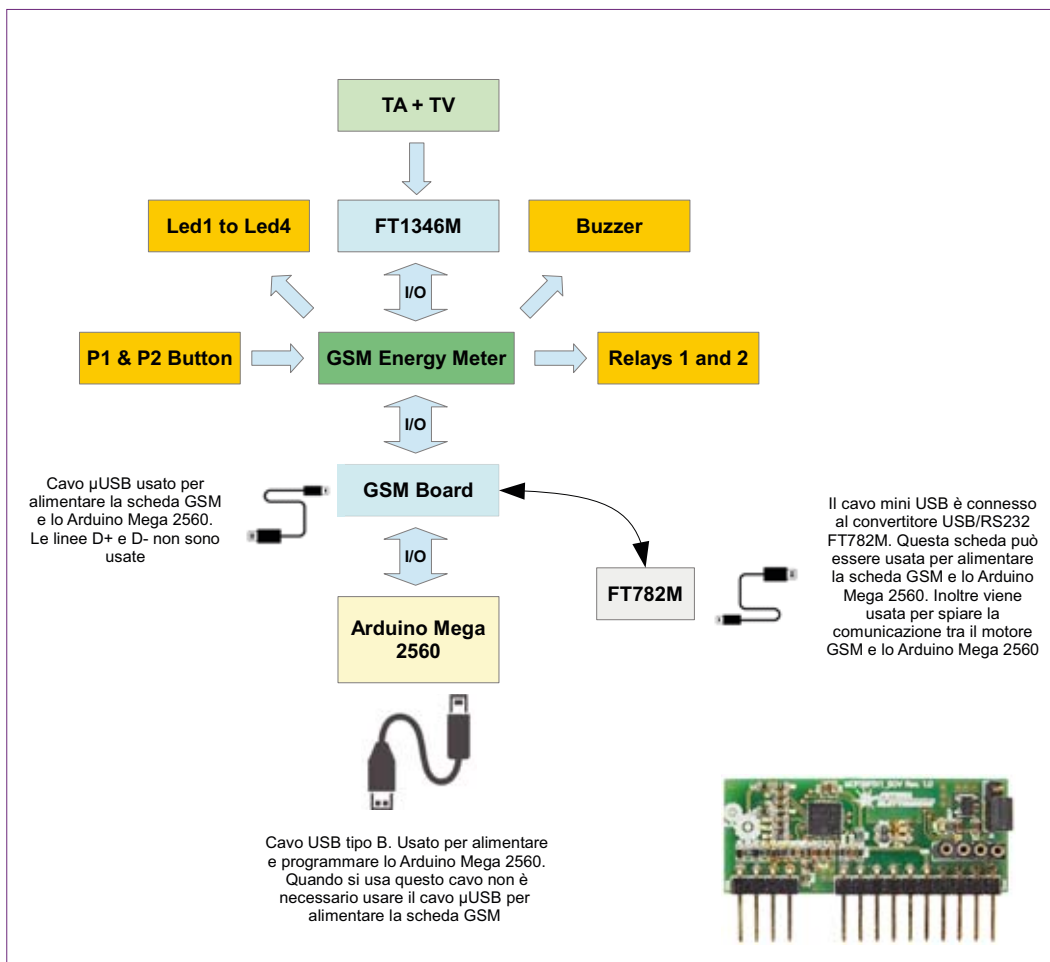
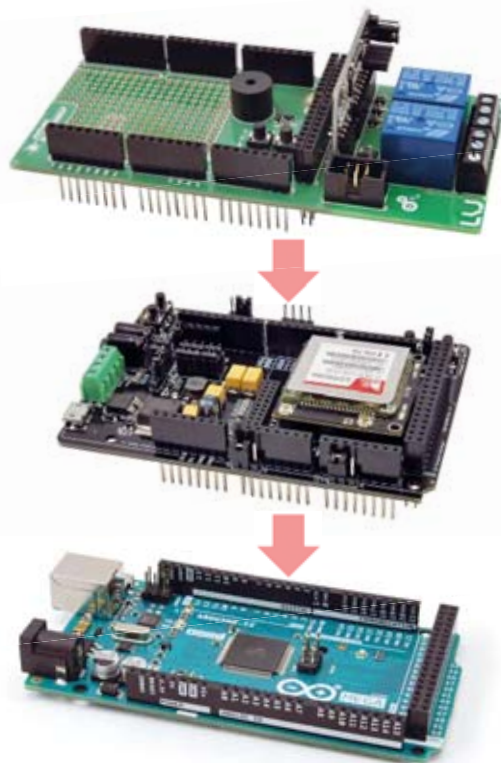


Fig. 1
Schema a blocchi del sistema; in basso a destra la scheda FT1346M provvista di strip.

il converter FT782M usato per la spia dei comandi AT inviati al modulo GSM.
Lo shield EnergyMeter può essere usato anche senza la scheda GSM board.

SCHEMA ELETTRICO SHIELD

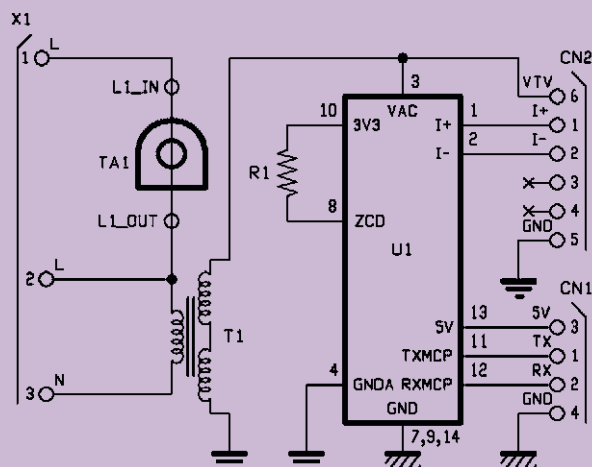
Come si può vedere lo schema elettrico è molto semplice. Cominciamo dalla connessione della scheda di misura FT1346M alla scheda Arduino Mega 2560, per la quale abbiamo previsto un adattatore di livello tra i segnali, che sono 0/5V per Arduino Mega e 0/3,3V per la FT1346M. Il compito è svolto dai MOSFET BSS123. Ripassiamo brevemente il concetto analizzando la linea TXD3, ovvero da Arduino a MCP39F511. Se la linea TXD3 dal lato +5V è a livello logico alto allora lo sarà anche dal lato +3,3V in quanto il diodo interno del MOSFET è interdetto e il canale del MOSFET è aperto. Nel momento in cui si porta la linea bassa dal lato dei +5V succede che il diodo interno entra in conduzione riportando lo stato logico basso anche sul lato dei +3,3V. Questo fa sì che la V_{GS} superi la soglia innescando il MOSFET. Lo stesso concetto si



Esploso dell'hardware: in alto lo shield GSM Energy Meter per FT1346M, in mezzo il GSM Shield e sotto la board Arduino Mega.

Schema elettrico

SCHEMA TA - TV



applica alla linea RX con la differenza che il segnale viaggia dall'integrato MCP39F511 ad Arduino. Detto ciò abbiamo sfruttato tale connessione per le linee **TX** e **RX** della **UART3** (rispettivamente I/O 14 e 15) e per la linea **ZCD** (connesso a I/O A12 se si usa la periferica PCINT20 oppure connesso a I/O 20 se si sfrutta periferica INT1). La selezione avviene tramite jumper J1; se in posizione 1-2 GPIO 20, se in posizione 2-3 GPIO A12. Invece per quanto riguarda le linee **MCLR** e **RST** (collegate rispettivamente agli I/O 46 e 45) abbiamo optato per un più semplice partitore di tensione. La linea **RST** serve per mantenere in reset il convertitore analogico digitale delta-sigma, l'amplificatore **PGA**, il V_{REF} interno e altre sezioni analogiche riguardanti il front-end. Invece la linea **MCLR** serve per mantenere in reset l'intera elettronica del dispositivo. La scheda prevede due pulsanti per uso generico (P1 e P2) connessi rispettivamente agli I/O 42 e 43; la condizione di riposo dei pulsanti, ovvero non premuti, è il livello logico alto (+5V) mantenuto tale da due resistori di pull-up. Quando un pulsante viene premuto, si forza lo zero logico all'ingresso. Tramite apposite routine se ne testa lo stato e si effettua il debouncing: lo stato viene ritenuto valido se permane per almeno 50 ms. È stato predisposto un buzzer per segnalazioni acustiche collegato alla linea di I/O 44 mentre i due relay per gestione carichi sono connessi alle linee di I/O 22 e 23. Infine abbiamo predisposto quattro led per uso generico connessi rispettivamente alle linee di I/O 24, 25, 26 e 27.

SCHEMA ELETTRICO SCHEDA TA - TV

Oltre a questa elettronica abbiamo un secondo circuito dove sono montati il TA e il TV e, tra-

mite apposito connettore a vaschetta, i segnali vengono portati allo shield e quindi alla FT1346M. Osservandone lo schema elettrico vediamo il TA SBT002 e il TV SVL101201 entrambi della Itacoil. I segnali analogici di misura vengono portati al connettore CN2; sono predisposti ma non montati il connettore CN1, per connessione interfaccia USB/Seriale FT782M, e la schedina di misura FT1346M (l'hardware è pensato per essere usato in autonomia, in altre parole montando la schedina di misura e l'interfaccia USB si ha un sistema di misura da collegare direttamente al PC e grazie al software EnergyMeter R.1.0.4.0 o superiore si ha la possibilità di monitorare le misure elettriche). Questa elettronica è pensata per essere inserita nel contenitore modulare per barra DIN Italtronic 05.0302530 (H53).

REALIZZAZIONE PRATICA

Diamo ora qualche suggerimento sulla costruzione delle due schede, partendo dallo shield, il quale è stato progettato più lungo della Arduino Mega 2560 al fine di poter alloggiare i due relé e la scheda di misura FT1346M. La scheda fa ampio uso di componenti SMD, per montare i quali suggeriamo di procurarsi della pasta flussante da deporre sulle piazzole prima della saldatura e di filo di lega saldante molto fine; quindi saldare prima di tutto i transistor, poi le resistenze e i condensatori, quindi i diodi. Fatto ciò si passa ai componenti THT, ossia pin-strip, morsettiere e relé. Per quanto riguarda la scheda con il TA e il TV i componenti sono tutti THT; suggeriamo di montare quindi prima il connettore CN2, la morsettieria X1 il TA e il TV il resto non è da montare salvo l'utilizzo per diversa applicazione.

LIBRERIA ARDUINO MCP39F511

Iniziamo ora a parlare della libreria appositamente sviluppata per le schede Arduino e utilizzata nello sketch di presentazione. Il codice C/C++ della libreria è stato suddiviso in più file al fine di facilitarne la comprensione e per raggruppare le varie funzioni sviluppate in categorie ben distinte come:

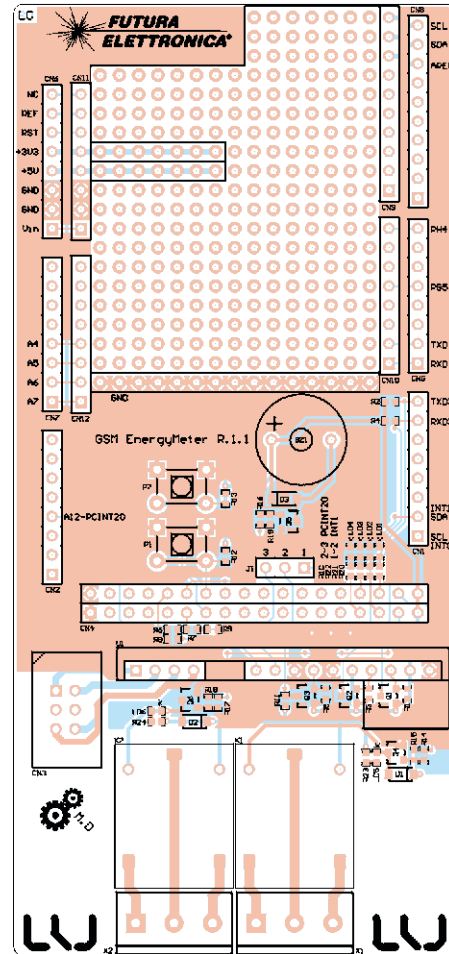
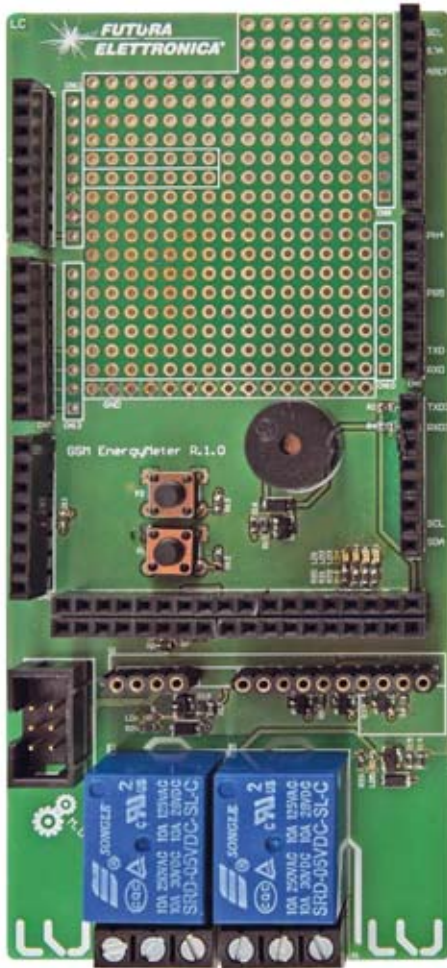
- funzioni di gestione degli I/O;
- funzioni di gestione interrupt;
- funzioni di gestione comunicazione seriale;
- funzioni di gestione lettura/scrittura registri integrato MCP39F511.

Quindi sono stati creati i seguenti file con estensione .cpp e i relativi file .h:

- file "**Io_MCP39F511.cpp**" e relativo file .h;

piano di montaggio

GSM ENERGY METER SHIELD



Elenco Componenti:

R1, R2, R3, R4, R5, R8, R9, R11, R12, R13:
10 kohm 1% (0603)
R10, R20, R21, R22, R23, R24:
1,2 kohm 1% (0603)
R14, R15, R17, R18, R19:
4,7 kohm 1% (0603)
R16: 1,8 kohm 1% (0603)
R6, R7: 5,6 kohm 1% (0603)
Q4, Q5, Q6: BC817
Q1, Q2, Q3: BSS123
LD1, LD5, LD6: LED rosso (0603)

LD2: LED giallo (0603)
LD3, LD4: LED verde (0603)
D1, D2, D3: MMSD4148T1G
P1, P2: Microswitch KPT1105T
K1, K2: Relé SRD-05VDC-SL-C
U1: Modulo Energy Meter FT1346M
BZ1: Buzzer OBO-1205D-A2
X1, X2: Morsetto 3 vie passo
5,08 mm (2 pz.)
CN1, CN2, CN5, CN6, CN7:
Strip Arduino 8 vie (5 pz.)

CN8: Strip Arduino 10 vie
CN4: Strip Arduino 2x18 vie
CN10, CN11, CN12: Strip femmina 8 vie
(4 pz.)
CN9: Strip femmina 10 vie
CN3: Connettore IDC 2x3 vie
J1: Strip maschio 3 vie

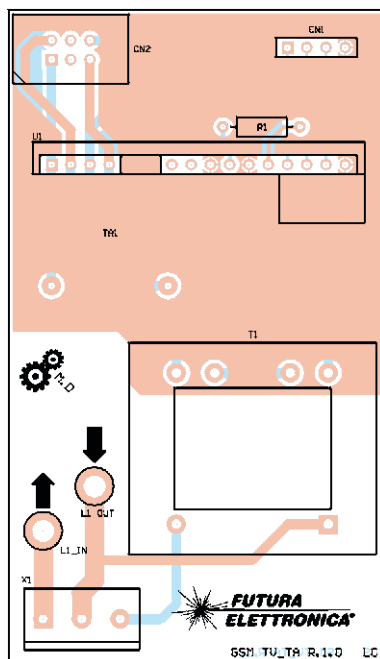
Varie
- Jumper
- Circuito stampato S1476 (56x122 mm)

- file "**Isr_MCP39F511.cpp**" e relativo file .h;
- file "**Uart_MCP39F511.cpp**" e relativo file .h;
- file "**Cmd_MCP39F511.cpp**" e relativo file .h;
- file "**Typedef_MCP39F511.h**" contenente le associazioni *typedef* usate nel codice per i tipi fondamentali come *int8_t*, *uint16_t* ecc. Sono stati definiti dei *typedef* anche nel file "**Cmd_MCP39F511.h**" ma sono riferiti ad alcuni registri di configurazione/stato dell'integrato.

Cominciamo la descrizione della libreria dal file "**Io_MCP39F511.cpp**" e relativo file .h contenente le definizioni dei pin di I/O utilizzati nonché le dichiarazioni delle variabili e delle funzioni per la gestione degli stessi. Cominciamo dal file .h che in testa trova la possibilità di scegliere quale scheda Arduino si vuole utilizzare ovvero Arduino Uno R3 oppure Arduino Mega 2560 (Fishino Mega). Come di consueto si utilizza la direttiva al compilatore

piano di montaggio

SCHEMA TA - TV



Elenco Componenti:

R1: 10 kohm 1%
U1: Modulo MCP39F511 (FT1346)
T1: Trasformatore SVL101201
TA1: Trasformatore corrente SBT002

Varie:

- Morsetto 2 vie estraibile (2 pz.)
- Connettore IDC 2x6 vie
- Strip femmina 4 vie
- Strip femmina 10 vie
- Circuito stampato S1477

#define seguita da una stringa. Per selezionare la scheda desiderata si deve semplicemente togliere i caratteri “//” da inizio riga, de-commentando la stessa; può essere selezionata solo una scheda alla volta.

Segue la sezione di selezione dell'interfaccia UART che si vuole utilizzare; nel nostro caso abbiamo selezionato la UART3 in hardware. A seconda dello hardware che si va ad utilizzare e quindi a seconda di dove si è deciso di collegare la UART dell'integrato MCP39F511 si andrà a selezionare la UART desiderata. Nel caso si decida di usare delle UART di tipo software si deve prestare attenzione in quanto in ricezione, sulle schede Arduino, difficilmente si riuscirà ad andare oltre i 19200 Baud. Quindi dato che MCP39F511 di fabbrica è impostato a 115200 Baud si dovrà prima impostare lo stesso per poter lavorare alla velocità di 19200. Tale configurazione può essere fatta sfruttando il software EnergyMeter che abbiamo già presentato sulle pagine della rivista e di cui parleremo più avanti dato che ne è stata sviluppata una nuova versione con delle funzioni aggiuntive rispetto alla precedente.

Selezionata l'interfaccia di comunicazione si deve decidere quale periferica di interrupt si vuole usare per gestire la linea **ZCD (Zero Cross Detection)**. Come detto in precedenza, sfruttando l'hardware messo a disposizione, possiamo decidere di usare o l'interrupt esterno **INT1** oppure l'interrupt ester-

no **PCINT20** (selezionare collegamento hardware tramite jumper).

Anche in questo caso si può usare/selezionare una sola periferica. Ricordiamo che selezionando l'interrupt di periferica **INT1** si perde la possibilità di usare l'interfaccia **PC** (questo per la scheda Arduino Mega 2560, su Arduino Uno R3 la periferica INT1 è mappata altrove) mentre se si seleziona l'interrupt di periferica **PCINT20** non si può usare la **UART** di tipo software in quanto i vettori di interrupt del port change sarebbero già assegnati alla libreria SoftwareUart.

Seguono le definizioni dei pin di I/O associati alle linee **RST**, **MCLR** e **ZCD**. Queste associazioni sono calibrate sul nostro hardware tuttavia si possono modificare a piacere a seconda dello hardware che state usando/progettando. Come si può notare ci sono due possibili configurazioni: una inerente allo Arduino Mega 2560 R3 e una allo Arduino Uno R3. Per ultimo seguono le definizioni di I/O dell'interfaccia UART differenziate a seconda della scheda Arduino usata. Anche in questo caso la definizione dei pin è centrata sul nostro hardware tuttavia si possono mappare i pin in modo totalmente diverso a seconda dello hardware a disposizione. La scheda Arduino Uno R3 non ha interfacce UART hardware al di fuori di quella usata per la connessione al monitor seriale dello IDE Arduino.

In fondo al file nelle definizioni di classe ritroviamo le dichiarazioni delle variabili usate per la gestione

degli I/O che andremo a descrivere a breve.
Le funzioni disponibili sono:

1. **"MCP39F511_SetControllLine"**
2. **"MCP39F511_ResetLine"**
3. **"MCP39F511_ClearLine"**

La prima funzione si occupa semplicemente di configurare i pin di I/O descritti prima e di settare la loro condizione iniziale di lavoro per quanto riguarda le uscite digitali. Quindi le linee **RST** e **MCLR** vengono poste a livello logico "0" mantenendo in reset l'integrato MCP39F511 fintantoché da codice non ne verrà cambiata la condizione operativa. La seconda funzione serve per gestire la linea RST e quindi il suo livello logico mentre la terza funzione gestisce la linea MCLR. All'interno di uno sketch, le funzioni appena descritte potrebbero essere utilizzate così:

```
Io.MCP39F511_SetControllLine();  
delay(250);  
Io.MCP39F511_MclrLine(1);  
delay(500);  
Io.MCP39F511_ResetLine(1);
```

Per prima cosa si richiama la funzione per la configurazione delle linee e, solo dopo una attesa di 250 ms, si procede a portare a livello logico alto prima la linea MCLR e poi la linea RST interponendo un ritardo di 500ms tra un'azione e la successiva. A questo punto l'integrato MCP39F511 è pronto, se configurato correttamente, ad eseguire le misure elettriche e a renderle disponibili alla nostra libreria tramite interfacciamento seriale.

Passiamo quindi al successivo file **Isr_MCP39F511.cpp** e relativo file .h i quali contengono il codice per la gestione degli interrupt usati. La libreria utilizza i seguenti vettori di interrupt:

- **"TIMER 1"** – Usato per la gestione della base dei tempi da 2mSec. La libreria usa due variabili temporali dette **"TimeOut"**: una necessaria per la gestione della comunicazione seriale e l'altra per la gestione della linea ZCD
- **"INT x"** – Usato per la gestione della linea ZCD (la libreria è impostata su INT 1, vedi hardware scheda di sviluppo)
- **"PCINT x"** – Usato per la gestione della linea ZCD (la libreria è impostata su PCINT 20, vedi hardware scheda di sviluppo)

Ovviamente **"INT x"** e **"PCINT x"** non possono e non devono essere usati assieme. Sarà cura dello

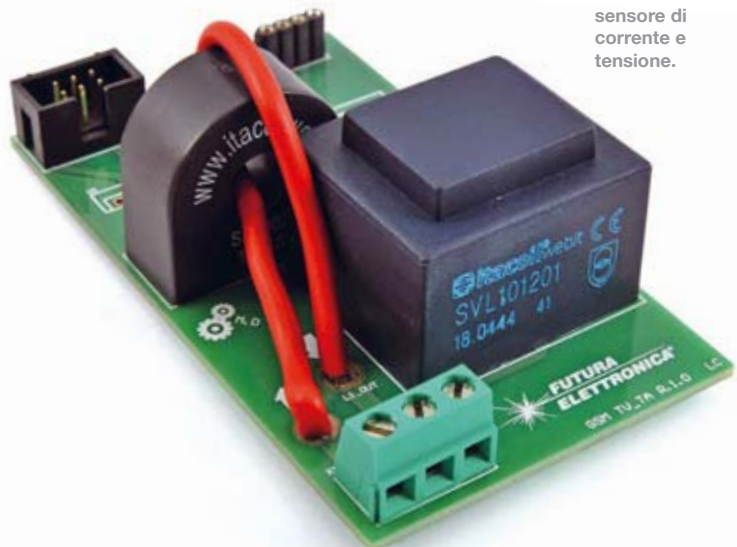
sviluppatore scegliere quale dei due utilizzare a seconda della situazione. Come detto prima la libreria dà la possibilità di sfruttare le sorgenti di interrupt esterne più consone alla propria applicazione, tutto dipende da come viene sviluppato lo hardware. Quindi nel file **"Isr_MCP39F511.cpp"** trovano posto le seguenti funzioni di cui una pubblica e tutte le altre private e quindi non disponibili all'utente:

- **"EnableLibInterrupt"**
- **"EnableTimerInterrupt"**
- **"EnableZcdIntx"**
- **"EnableZcdPortCx"**

La prima funzione è l'unica disponibile all'utente la quale serve per richiamare le funzioni private di configurazione dei vettori di interrupt. All'interno di tale funzione ritroviamo le righe di codice che richiamano le funzioni private e in particolare la configurazione e abilitazione del TIMER 1, la configurazione e abilitazione dell'interrupt esterno INT x (se selezionato) e infine la configurazione e abilitazione dell'interrupt esterno PCINT x (se selezionato).

Le funzioni di configurazione e abilitazione degli interrupt contengono le direttive al compilatore **"#ifdef"** per selezionare le porzioni di codice corrette a seconda delle scelte di configurazione fatte nel file di libreria **"Io_MCP39F511.h"**. Quindi a seconda della scheda Arduino selezionata e del tipo di interrupt esterno scelto il compilatore sarà in grado di processare e compilare il codice corretto. Tutti i vettori di interrupt usati sono stati inseriti nel codice usando la direttiva al compilatore **"ISR()"** dove all'interno delle parentesi tonde c'è il nome del vettore di interrupt.

↓
La scheda con i trasformatori TA e TV: il sensore di corrente e tensione.



Parliamo ora del file **"Uart_MCP39F511.cpp"** che si occupa di gestire la comunicazione seriale tra la scheda Arduino e l'integrato MCP39F511 ed eventualmente di abilitare/disabilitare la comunicazione seriale verso il monitor seriale dello IDE al fine di poter eseguire delle sezioni di debug del codice. Le funzioni messe a disposizione sono:

- **"EnableDisableIdeMonitor"**
- **"SetBaudRateUart"**
- **"UartSendData"**
- **"UartReceivedData"**
- **"SetUartBegin"**
- **"SetUartEnd"**

Le prime quattro funzioni sono pubbliche e disponibili all'utente le ultime due sono private e usate esclusivamente dalla libreria. Per la comunicazione seriale è stato predisposto un array di byte di lunghezza 64 sufficiente a gestire i pacchetti dati trasmessi e ricevuti.

La prima funzione serve per abilitare/disabilitare e configurare la comunicazione seriale verso il monitor seriale dello IDE al fine di poter eseguire il debug del codice e per stampare a video delle stringhe di testo.

La seconda funzione serve per abilitare/disabilitare e configurare la comunicazione seriale tra la scheda Arduino selezionata e l'integrato MCP39F511. A seconda delle impostazioni fatte nel file **"Io_MCP39F511.h"** verrà configurata una delle possibili UART a disposizione sfruttando le funzioni private **"SetUartBegin"** (configura e attiva la UART) oppure **"SetUartEnd"** (disabilita UART selezionata precedentemente).

Le funzioni **"UartSendData"** e **"UartReceivedData"**, come suggeriscono i nomi, vengono usate per inviare un pacchetto dati e per ricevere un pacchetto dati. All'interno dello sketch le funzioni possono essere usate nel seguente modo:

```
Uart.EnableDisableIdeMonitor(true, BAUD_115200);  
Uart.SetBaudRateUart(true, BAUD_115200);
```

La prima riga abilita il monitor seriale dello IDE alla velocità di 115200 baud, la seconda abilita la comunicazione seriale tra la scheda Arduino e MCP39F511 sempre alla velocità di 115200 baud. Per disabilitare le comunicazioni seriali è sufficiente sostituire il primo parametro con **"false"** al posto di **"true"**.

Non ci rimane che affrontare l'ultimo file di libreria ovvero **"Cmd_MCP39F511.cpp"** (e relativo file .h), il

più corposo in termini di codice, che contiene tutte le funzioni necessarie per leggere/scrivere i registri dell'integrato MCP39F511.

Cominciamo con il file .h il quale contiene le definizioni usate per la gestione dei registri e della memoria EEPROM da 512 Byte interna all'integrato MCP39F511.

In testa al file troviamo le direttive al compilatore che definiscono la mappatura della memoria EEPROM con indicata la locazione di memoria in cui sono salvati i dati di fabbrica per i registri di configurazione della nostra scheda FT1346M. Tali dati possono essere recuperati dalla memoria EEPROM e trasferiti ai registri di configurazione tramite apposita funzione di libreria di cui discuteremo più avanti.

Come promemoria nel file .h è indicato in forma tabellare quanto appena discusso rendendo più chiaro il concetto appena esposto.

Dopo le costanti di gestione della memoria EEPROM troviamo le direttive al compilatore con indicati i comandi per la gestione dell'integrato MCP39F511. Ricordiamo che la comunicazione seriale prevede l'invio di pacchetti dati che devono essere costruiti secondo un certo protocollo.

La **Fig. 2** mostra come deve essere composto il pacchetto dati, come si può osservare abbiamo:

- **"Header Byte"** in testa al pacchetto e deve sempre essere 0xA5;
- **"Number of Bytes"** numero di byte da cui è composto l'intero messaggio;
- **"Command Packet"** contiene sia il comando da inviare all'integrato sia eventuali byte di dati necessari al comando stesso. Possono essere inviati più comandi consecutivi e relativi dati
- **"Checksum"** chiude il pacchetto dati il checksum.

La lista di comandi disponibile è visibile in **Fig. 3** e sono tutti implementati tranne i comandi di auto-calibrazione. Quando si invia un comando all'integrato questo risponderà o con un semplice ACK oppure con un ACK seguito dai dati richiesti. Quando il pacchetto comprende anche una serie di dati si avrà in coda il codice di checksum ottenuto facendo la somma dei byte da trasmettere e dividendo il valore ottenuto per 256. Il resto della divisione rappresenta il checksum.

Oltre ai codici comando sono stati parametrizzati i seguenti:

- **HEADER** → 0xA5 (header byte)



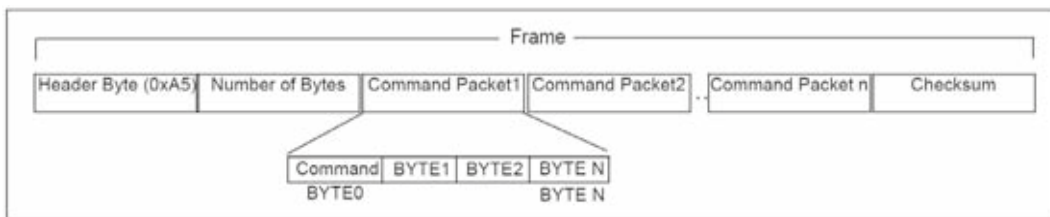


Fig. 2
Struttura del
pacchetto di
dati.

- ACK → 0x06 (acknowledge)
- NOACK → 0x15 (no acknowledge)
- CHK_FAIL → 0x51 (checksum fail)
- RETREIVE FACTORY → 0xA5A5 (codice per ripristino parametri di fabbrica Microchip).

A quanto appena detto seguono una serie di codici di gestione della libreria e in particolare per la decodifica delle risposte. In altre parole abbiamo codificato in modo univoco i comandi da inviare all'integrato, così facendo alla ricezione del pacchetto la funzione di libreria che decodifica la risposta sa che comando abbiamo inviato e quindi sa quali dati deve aspettarsi come risposta in modo da caricarli nella giusta struttura dati sempre che il checksum sia corretto.

Quindi i comandi di lettura sono codificati a partire da 0x00; i comandi per la scrittura sui registri sono codificati a partire da 0x40; i comandi per la lettura delle pagine EEPROM partono da 0x80; i codici di scrittura EEPROM partono da 0xA1 e infine il codice per congelare i dati inviati ai registri ha codice 0xFF.

La funzione di decodifica delle risposte può ritornare i codici 0x00 (decodifica OK), codice 0x01 (ricevuti meno dati di quelli indicati dal pacchetto), 0x02 (checksum error) oppure 0x03 (no acknowledge). Segue una serie di costanti che impostano i decimali assegnati alle varie misure elettriche effettuate dall'integrato; tali costanti sono strettamente legate alla configurazione della scheda di misura FT1346M e quindi sconsigliamo di modificarli. Subito dopo ritroviamo la mappatura completa dei registri presenti nell'integrato MCP39F511 con indicato nel commento allegato la sua dimensione (byte, word, double word ecc) e se sono registri che prevedono il segno oppure no. Ad esempio il fattore di potenza mappato all'indirizzo **0x000C** prevede il segno e quindi è indicato come **s16** ovvero word con segno.

Dopo questa rassegna di costanti, ritroviamo le dichiarazioni delle funzioni sviluppate per gestire l'integrato e le variabili usate. Abbiamo deciso di assegnare una variabile appropriata per ogni registro disponibile al fine di dare una struttura

dati completa all'utilizzatore. Con questo approccio l'utente ha il controllo completo dell'integrato. Analizziamo quindi il file "**Cmd_MCP39F511.cpp**". Le prime funzioni in testa al file sono private e quindi non disponibili all'utente durante la realizzazione dello sketch. Si può dire che sono funzioni di basso livello in quanto servono per preparare il pacchetto dati da inviare a seconda che si debba scrivere o leggere dei registri. In particolare abbiamo:

- "**MCP39F511_FreezeFlash**" → serve per validare e salvare in modo definitivo i dati inviati ai registri; durante una configurazione si inviano i dati ai registri, ma non vengono salvati in modo permanente finché tale comando non viene inviato, quindi nella libreria ci sarà una funzione pubblica (**MCP39F511_FreezeFlashData**) atta ad assolvere tale funzione
- "**MCP39F511_WriteData_Byte**" → serve per scrivere un byte (8 bit) all'indirizzo specificato
- "**MCP39F511_WriteData_Word**" → serve per scrivere una word (16 bit) all'indirizzo specificato;
- "**MCP39F511_WriteData_DoubleWord**" → serve per scrivere una double word (32 bit) all'indirizzo specificato;
- "**MCP39F511_ReadData**" → serve per leggere uno o più byte fino a un massimo di 32 dall'indirizzo specificato;
- "**MCP39F511_WriteEeprom**" → serve per scrivere dati in EEPROM, la quale è gestita in pagine da 16 byte ciascuna (8 word) quindi si deve tenere conto di questo in quanto se si vuole cambiare un solo byte presente nella pagina desiderata, prima si deve leggere l'intera pagina, modificare il byte di interesse e poi riscrivere la pagina stessa;
- "**MCP39F511_ReadEeprom**" → serve per leggere dati dalla EEPROM;
- "**MCP39F511_EraseEeprom**" → serve per cancellare tutta la memoria EEPROM;
- "**CalcChecksum**" → serve per calcolare il checksum del pacchetto dati inviato o ricevuto;
- "**MCP39F511_DecodeDataReceived**" → serve per decodificare il pacchetto dati ricevuto a seconda del comando inviato; come detto prima i co-

→ **Fig. 3**
Lista dei comandi.

Command #	Command	Command ID	Instruction Parameter	Number of bytes	Successful Response UART_TX
1	Register Read, N bytes	0x4E	Number of bytes	2	ACK, Data, Checksum
2	Register Write, N bytes	0x4D	Number of bytes	1+N	ACK
3	Set Address Pointer	0x41	ADDRESS	3	ACK
4	Save Registers To Flash	0x53	None	1	ACK
5	Page Read EEPROM	0x42	PAGE	2	ACK, Data, Checksum
6	Page Write EEPROM	0x50	PAGE	18	ACK
7	Bulk Erase EEPROM	0x4F	None	1	ACK
8	Auto-Calibrate Gain	0x5A	None		Note 1
9	Auto-Calibrate Reactive Gain	0x7A	None		Note 1
10	Auto-Calibrate Frequency	0x76	None		Note 1

mandi per la lettura o scrittura dei registri sono tutti codificati in modo univoco; tale funzione, una volta validato il pacchetto ricevuto, ne estrae il contenuto e salva i dati nelle apposite strutture al fine di renderli disponibili all'utente. Oppure se i dati sono stati inviati all'integrato si assicura di avere ricevuto l'ACK.

Per quanto riguarda le funzioni pubbliche e quindi disponibili per gli sketch abbiamo le seguenti, divise in categorie:

- Ripristino parametri di fabbrica
- Gestione memoria EEPROM
- Lettura misure elettriche
- Scrittura registri

Cominciamo dalla prima categoria ovvero "**Ripristino parametri di fabbrica**", questa è composta da due sole funzioni: la prima serve per ripristinare i parametri di fabbrica Microchip dell'integrato MCP39F511 la seconda per ripristinare i parametri di fabbrica della nostra scheda di misura FT1346M.

Quindi la funzione "**MCP39F511_RetrieveMicrochipFactoryValues**", come suggerisce il nome, ripristina i parametri di fabbrica Microchip inviando il codice 0xA5A5.

Invece la funzione "**MCP39F511_RetrieveFT1346M_FactoryParameters**" ripristina i parametri di fabbrica della nostra scheda di misura. Quindi la funzione esegue una serie di letture in EEPROM al fine di recuperare i dati necessari e inviarli ai corrispondenti registri. Eseguito il ripristino si deve congelare il contenuto e questo viene reso possibile chiamando la funzione "**MCP39F511_FreezeFlash**". È a carico dell'utente chiamare tale funzione. La seconda categoria "**Gestione memoria EEPROM**"

mette a disposizione tre funzioni le quali permettono di leggere/scrivere/cancellare le pagine della memoria EEPROM.

La funzione "**MCP39F511_ReadEepromPage**" permette di leggere una pagina di memoria EEPROM a piacere e l'unico parametro da passare alla funzione è il numero di pagina che si desidera leggere. Quando si eseguono delle operazioni di lettura/scrittura generiche della memoria EEPROM ci si appoggia a un array di lunghezza 16 byte detto "**EepromPageArray**" il quale conterrà i dati letti, in caso di lettura, o dovrà essere caricato con i dati che si vuole scrivere in memoria in caso di scrittura. Quando si esegue una scrittura ci si deve solo preoccupare di caricare i dati nello array e richiamare l'apposita funzione a tutto il resto ci penserà la libreria. La funzione che si occupa di scrivere i dati in memoria è "**MCP39F511_WriteEepromPage**" e come per la precedente si deve passare il parametro che indica su quale pagina lavorare.

Per ultima abbiamo la funzione "**MCP39F511_BulkEraseEeprom**" la quale cancella tutta la memoria EEPROM riportandone il contenuto al suo valore di fabbrica ovvero 0xFF.

La terza categoria "**Lettura misure elettriche**" è composta da una serie di funzioni che leggono i registri contenenti le misure elettriche effettuate dall'integrato. Di seguito la rassegna con una breve descrizione, tutte le funzioni che seguono non richiedono parametri:

- "**MCP39F511_ReadStatusAndVersion**" → legge il valore contenuto nei registri "**System Status Register**" e "**System Version Register**"; i dati letti sono posti nelle rispettive strutture dati "**SystemStatusFlag**" e "**SystemVersionFlag**";
- "**MCP39F511_ReadSystemConfiguration**" → legge il valore contenuto nel registro "**System**

Configuration Register" e carica la struttura dati **"SystemConfigurationFlag"**;

- **"MCP39F511_ReadEventStatus"** → legge il valore contenuto nel registro **"Event Configuration Register"** e carica la struttura dati **"EventConfigurationFlag"**;
- **"MCP39F511_ReadRmsVoltageRaw"** → legge il valore della tensione RMS; il valore letto è un RAW e va manipolato per ottenere il valore con i decimali corretti; nel caso della nostra scheda per la misura della tensione RMS abbiamo 2 decimali quindi l'utente, una volta letto il valore di tensione RAW lo deve dividere per 100 (10^2) se i decimali fossero stati tre si sarebbe dovuto dividere per 1.000 ecc.;
- **"MCP39F511_ReadLineFrequencyRaw"** → legge il valore della frequenza di linea, anche in questo caso il valore letto è un RAW e va diviso per 1000 in quanto sono stati definiti 3 decimali per questa misura;
- **"MCP39F511_ReadSarAdc"** → legge il valore del convertitore ADC ad approssimazioni successive; l'integrato MCP39F511 ha un ingresso analogico indipendente che può essere usato per misure analogiche generiche;
- **"MCP39F511_ReadPowerFactorRaw"** → legge il valore del fattore di potenza; il registro è in complemento a due, quindi una volta letto si deve richiamare una funzione per convertire il valore letto in un numero decimale da -1 a +1; la funzione che si occupa della conversione è **"CalcPowerFactor"** ed è cura dell'utente richiamarla dopo la lettura del registro (anche in questo caso il valore convertito è un RAW e va quindi diviso per 1.000);
- **"MCP39F511_ReadRmsCurrentRaw"** → legge il valore della corrente RMS in formato RAW; il valore va diviso per 1000 in quanto sono stati definiti 3 decimali;
- **"MCP39F511_ReadActivePowerRaw"** → legge il valore della potenza attiva in formato RAW; il valore va diviso per 100 per avere due decimali;
- **"MCP39F511_ReadReactivePowerRaw"** → legge il valore della potenza reattiva in formato RAW; il valore va diviso per 100 per avere due decimali;
- **"MCP39F511_ReadApparentPowerRaw"** → legge il valore della potenza apparente in formato RAW; il valore va diviso per 100 per avere due decimali;
- **"MCP39F511_ReadAllRawData"** → questa funzione legge tutti i valori appena descritti in formato RAW;
- **"MCP39F511_ReadImportedActiveEnergy"** →

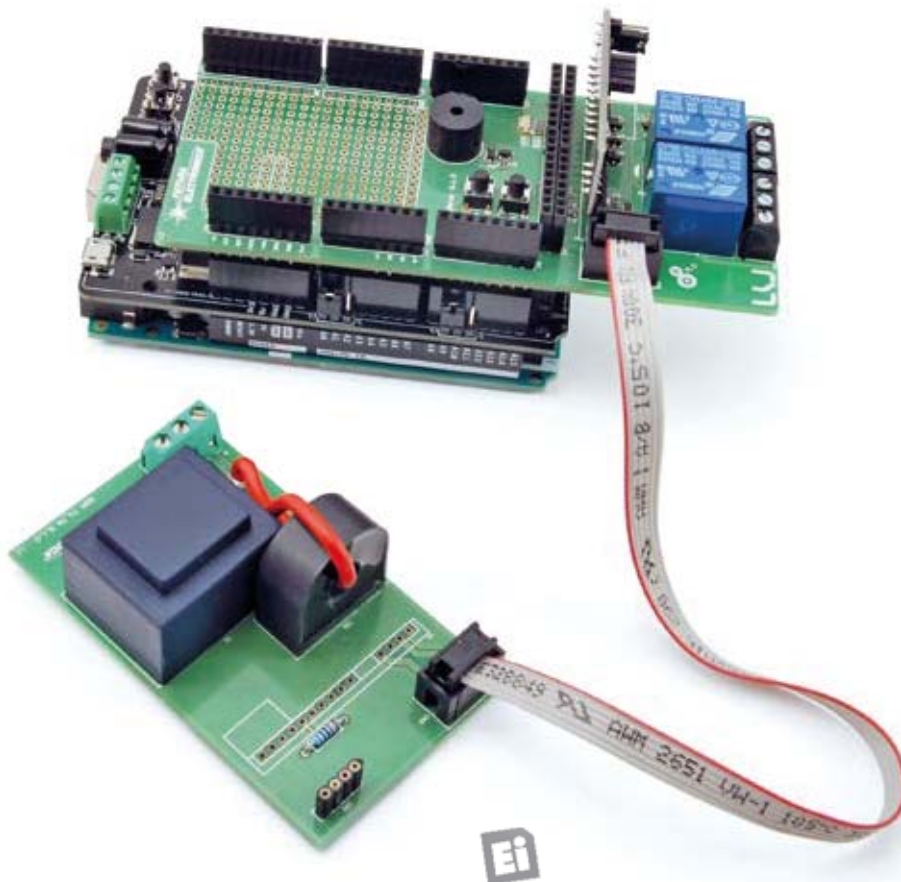


Fig 4
La scheda
TA/TV
collegata allo
shield GSM
Energy Meter
tramite flat-
cable.

- legge il valore di energia attiva importata;
- **"MCP39F511_ReadExportedActiveEnergy"** → legge il valore di energia attiva esportata;
- **"MCP39F511_ReadImportedReactiveEnergy"** → legge il valore di energia reattiva importata;
- **"MCP39F511_ReadExportedReactiveEnergy"** → legge il valore di energia reattiva esportata;
- **"MCP39F511_ReadAllEnergy"** → legge tutti i valori di energia appena descritti;
- **"MCP39F511_ReadMinimumRecord_1"** → legge il registro (record 1) contenente il valore minimo registrato tra le misure supportate tra le quali abbiamo Tensione RMS, Corrente RMS, Frequenza di linea, Fattore di potenza, Potenza attiva, Reattiva e Apparente;
- **"MCP39F511_ReadMinimumRecord_2"** → legge il registro (record 2) contenente il valore minimo registrato tra le misure supportate;
- **"MCP39F511_ReadMaximumRecord_1"** → legge il registro (record 1) contenente il valore massimo registrato tra le misure supportate;
- **"MCP39F511_ReadMaximumRecord_2"** → Legge il registro (record 2) contenente il valore massimo registrato tra le misure supportate.

Passiamo ora alla quarta categoria **"Scrittura registri"** la quale è composta da una serie di funzioni per scrivere nei registri di configurazione; di seguito le descriviamo, premettendo che non richiedono parametri a patto di caricare i dati da scrivere nei registri nelle apposite strutture dati.

- **"MCP39F511_WriteGainCurrentRms"** → Scrive il registro di calibrazione del guadagno per *Gain Current RMS*.
- **"MCP39F511_WriteGainVoltageRms"** → Scrive il registro di calibrazione del guadagno per *Gain voltage RMS*.
- **"MCP39F511_WriteGainActivePower"** → Scrive il registro di calibrazione del guadagno per *Gain Active Power*.
- **"MCP39F511_WriteGainReactivePower"** → Scrive il registro di calibrazione del guadagno per *Gain Reactive Power*.
- **"MCP39F511_WriteOffsetCurrentRms"** → Scrive il registro di calibrazione per *Offset Current RMS*.
- **"MCP39F511_WriteOffsetActivePower"** → Scrive il registro di calibrazione per *Offset Active Power*.
- **"MCP39F511_WriteOffsetReactivePower"** → Scrive il registro di calibrazione per *Offset Reactive Power*.
- **"MCP39F511_WriteDcOffsetCurrent"** → Scrive il registro di calibrazione per *DC Offset Current*.
- **"MCP39F511_WritePhaseCompensation"** →

Scriva il registro di calibrazione per *Phase Compensation Register*.

- **"MCP39F511_WriteApparentPowerDivisor"** → Scrive il registro *Apparent Power Divisor*, che serve per allineare la risoluzione delle misure di corrente RMS con le misure di tensione RMS.
- **"MCP39F511_WriteSystemConfiguration"** → Scrive il registro di sistema per la configurazione delle seguenti:
 - **"PGA setting"** canale 0 e canale 1
 - **"Voltage Reference Trim"**
 - **"Uart Baud Rate"**. Si consiglia di non cambiare la configurazione predefinita (115200 Baud).
 - **"ZCD setting"**
 - **"Single Wire Auto-Transmission"**
 - **"Temperature Compensation"**
 - **"ADC Reset State"**
 - **"ADC Shutdown State"**
 - **"Voltage Reference Shutdown Control"**.
- **"MCP39F511_WriteEventConfiguration"** → Scrive il registro di sistema per la gestione degli eventi a seconda delle misure effettuate.
- **"MCP39F511_WriteRange"** → Scrive il registro di configurazione del fattore di scala. È un registro a 32 bit e contiene il fattore di scala per le misure di corrente RMS, tensione RMS e potenza (attiva, reattiva e apparente).
- **"MCP39F511_WriteCalibrationCurrent"** → Scrive il registro per la calibrazione della corrente. Corrente target usata durante la calibrazione
- **"MCP39F511_WriteCalibrationVoltage"** → Scrive il registro per la calibrazione della tensione. Tensione target usata durante la calibrazione.
- **"MCP39F511_WriteCalibrationActivePower"** → Scrive il registro per la calibrazione della potenza attiva. Potenza attiva target usata durante la calibrazione.
- **"MCP39F511_WriteCalibrationReactivePower"** → Scrive il registro per la calibrazione della potenza reattiva. Potenza reattiva target usata durante la calibrazione.
- **"MCP39F511_WriteLineFrequencyReference"** → Scrive il registro contenente la frequenza di linea di riferimento.
- **"MCP39F511_WriteAccumulationIntervalParameter"** → Scrive il registro contenente il parametro intervallo di accumulazione – *Accumulation Interval Parameter*. In pratica sono il numero di cicli usati per la computazione delle misure elettriche.
- **"MCP39F511_WriteVoltageSagLimit"** → Scrive il registro *Voltage Sag Limit*. Soglia per la tensione RMS superata la quale viene registrato il flag corrispondente.

- **"MCP39F511_WriteVoltageSurgeLimit"** → Scrive il registro *Voltage SurgeLimit*. Soglia per la tensione RMS superata la quale viene registrato il flag corrispondente.
- **"MCP39F511_WriteOverCurrentLimit"** → Scrive il registro *Overcurrent Limit*. Soglia per la corrente RMS superata la quale viene registrato il flag corrispondente.
- **"MCP39F511_WriteOverPowerLimit"** → Scrive il registro *Overpower Limit*. Soglia per la potenza attiva superata la quale viene registrato il flag corrispondente.
- **"MCP39F511_WriteTempCompensationFrequency"** → Scrive il registro *Temperature Compensation for Frequency*.
- **"MCP39F511_WriteTempCompensationCurrent"** → Scrive il registro *Temperature Compensation for Current*.
- **"MCP39F511_WriteTempCompensationPower"** → Scrive il registro *Temperature Compensation for Power*.
- **"MCP39F511_WriteAmbientTempReference"** → Scrive il registro *Ambient Temperature Reference Voltage*.
- **"MCP39F511_WritePwmPeriod"** → Scrive il registro *PWM Period*. Contiene il valore del periodo della forma d'onda PWM generata.
- **"MCP39F511_WritePwmDutyCycle"** → Scrive il registro *PWM Duty Cycle*. Contiene il valore del duty-cycle associato alla forma d'onda PWM generata.
- **"MCP39F511_WriteMinMaxPointer_1"** → Scrive il registro *MinMaxPointer1*. Contiene l'indirizzo di memoria associato alla misura elettrica che si vuole monitorare per la rilevazione del suo valore massimo e minimo.
- **"MCP39F511_WriteMinMaxPointer_2"** → Scrive il registro *MinMaxPointer2*. Contiene l'indirizzo di memoria associato alla misura elettrica che si vuole monitorare per la rilevazione del suo valore massimo e minimo.
- **"MCP39F511_ResetMinMaxPointer_1"** → Resetta il registro *MinMaxPointer1*. Se il valore contenuto nel registro è zero non viene registrato nessun massimo e minimo.
- **"MCP39F511_ResetMinMaxPointer_2"** → Resetta il registro *MinMaxPointer2*. Se il valore contenuto nel registro è zero non viene registrato nessun massimo e minimo.
- **"MCP39F511_WriteOverTemperatureLimit"** → Scrive il registro *Overtemperature Limit*. Nel caso si manifesti l'evento di superamento soglia di temperatura verrà settato apposito flag.
- **"MCP39F511_WriteEnergyControl"** → Scrive il registro *Energy Control*. Serve per resettare/iniziare un ciclo di accumulazione di energia. I registri contenenti l'energia accumulata si resettano in caso di mancanza di alimentazione. È cura dello sviluppatore salvare tali registri in apposite locazioni di memoria non volatile nella propria applicazione.
- **"MCP39F511_WritePwmControl"** → Scrive il registro per il controllo del segnale PWM – *PWM Control*.
- **"MCP39F511_WriteNoLoadThreshold"** → Scrive il registro *No Load Threshold*. Questo registro serve a indicare una soglia di potenza sotto la quale non viene registrata l'energia accumulata. Un valore di 100 corrisponde a 1W di potenza. In pratica i valori hanno una risoluzione di due decimali.
- **"MCP39F511_FreezeFlashData"** → Questa funzione serve per congelare i valori scritti nei registri. In altre parole questa funzione deve essere richiamata tutte le volte che si esegue una sessione di scrittura dei registri di sistema. Non è necessario chiamarla per ogni registro che si va a modificare. Può essere eseguita semplicemente alla fine di un ciclo di scrittura di "n" registri.

CONCLUSIONI

Bene, con questa rassegna abbiamo completato la descrizione delle funzioni messe a disposizione dalla libreria; da ora possiamo concentrarci su uno sketch dimostrativo al fine di mostrare come includere la libreria nei vostri progetti. Avendo esaurito lo spazio a nostra disposizione in questo fascicolo, vedremo questo sketch ed il software di gestione da PC nella prossima, conclusiva puntata. 



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>



Il termostato diventa WiFi!

Termostato touch screen da incasso

**Istruzioni
in italiano!**

Vuoi gestire facilmente il clima domestico anche quando sei lontano da casa e risparmiare sui consumi energetici? Con il nuovo termostato Wi-Fi puoi fare tutto questo tranquillamente. Grazie all'App gratuita per Android e iOS, con un semplice tocco sullo smartphone, puoi attivare da remoto il riscaldamento, modificare la temperatura e impostare la programmazione.

- ✓ Alimentazione: 230 VAC 50 Hz
- ✓ Precisione: circa 0,5° C
- ✓ Range di temperatura: da 5° C a 35° C (impostabile)
- ✓ Limite range di temperatura: 5-99° C
- ✓ Consumo: <0,3 Watt
- ✓ Sensore di temperatura: NTC (10k)1%
- ✓ Montaggio: a incasso
- ✓ Dimensioni: 86 x 86 x 17 mm
- ✓ Peso: 234 grammi

**COMPATIBILE
CON AMAZON
ECHO E GOOGLE
HOME!**



€ 59,00

cod. **FR695**

PLACCA PER TERMOSTATO WiFi FR695

Placca da utilizzare in abbinamento al termostato Wi-Fi FR695 e alla scatola da incasso modello 503E (acquistabile separatamente).
Dimensioni: 116x74,9x7,6 mm



€ 4,90

cod. **PLACCAFR695**



SLOT-MACHINE PROFESSIONALE



del **PROF. MICHELE MENNITI**

Completiamo la descrizione del nostro "gioco da bar" domestico e prepariamoci a utilizzarlo dopo averne caricato il firmware. Seconda e ultima puntata.

R

iprendiamo la descrizione del progetto della nostra slot machine da dove lo avevamo lasciato, ossia dopo aver introdotto l'hardware, averne descritta l'elettronica nelle due schede che lo compongono ed aver affrontato le fasi di montaggio che porteranno ad ottenere il risultato visibile nelle foto

che trovate nelle prossime pagine. Ricordiamo che il progetto riguarda la realizzazione di una slot-machine grande come quelle dei bar e delle sale giochi, dotata di display alfanumerici capaci di visualizzare, oltre a numeri e lettere, simboli grafici di base, per arrivare a simulare il funzionamento del classico visualizzatore a tamburi che riportano i tradizionali simboli come frutta, campane ecc.

Iniziamo questa seconda e conclusiva puntata dell'articolo

dedicato alla slot machine con qualche operazione preliminare da effettuare una volta finito il montaggio. Aver completato la slot machine è indispensabile perché alcune importanti operazioni di taratura preliminare vanno espletate con tutti i collegamenti completi, dopo aver fornito l'alimentazione, anche se prima di caricare il firmware sul microcontrollore.

IMPOSTAZIONI HARDWARE PRELIMINARI

Prima di tutto occorre regolare la corrente del driver per lo stepper motor e allo scopo, sulla relativa scheda è presente un piccolo trimmer SMD, che va ruotato per ottenere il valore di tensione desiderato (in base ai calcoli che consiglia il datasheet); la tensione è rilevabile collegando il puntale negativo alla massa del circuito generale e quello positivo direttamente sul trimmer. Per la regolazione è consigliabile usare un piccolo giravite con la punta in plastica, per evitare che scivolando possa creare un cortocircuito che distruggerebbe inevitabilmente il driver.

Nel caso del nostro driver, il valore della tensione è stato fissato a 0,8V. Con questa impostazione ed il tipo di motore che abbiamo usato nel nostro prototipo, il driver resta nei normali limiti di temperatura operativa e non necessita di raffreddamento, passivo o attivo.

La gettoniera arriva già settata dalla fabbrica, quindi se decidete di non metterci mano, dovrete adeguare il firmware al numero di impulsi che essa genera in base alla moneta inserita. Noi abbiamo preferito modificare questi valori per superare alcuni (rari) errori nella lettura del numero di impulsi, distanziandoli da moneta a moneta.

L'operazione di per sé è molto semplice e richiede

solo una buona dose di pazienza perché la condizione ideale è quella di disporre di almeno 10÷15 monete per ogni valore (il datasheet prescrive che ne servirebbero 20, ma si può sopperire introducendo alcune monete due volte), e quindi occorrerà effettuare un centinaio di inserimenti per completare l'operazione. Nell'apposito riquadro è elencata la sequenza di operazioni da effettuare per adattare la gettoniera alle esigenze del nostro firmware. Con questa operazione abbiamo assegnato alle monete da 10c, 20c, 50c, 1€ e 2€ rispettivamente 1, 2, 4, 7 e 10 impulsi, che poi saranno tramutati in 1, 2, 5, 10 e 20 crediti dal firmware. Una volta inserita la moneta, se la gettoniera è attiva (quindi non inibita) essa uscirà dalla fessura inferiore, finendo nel cassetto e contestualmente il piccolo display mostrerà il numero di impulsi ad essa associati, che saranno inviati al micro.

Se invece la gettoniera si trovasse in stato di inibizione, la moneta verrebbe restituita al giocatore uscendo dalla fessura anteriore, che è sottostante a quella di inserimento. Il pulsante rosso serve per i casi in cui la moneta dovesse bloccarsi, per allargare il meccanismo e permettere alla moneta di uscire agevolmente.

LA PROGRAMMAZIONE DEL MICRO

Dopo questa operazione la SLOT MACHINE non è ancora pronta per funzionare, perché richiede che vi sia caricato il firmware nel microcontrollore U1. Arrivati a questo punto, il firmware, che sia quello previsto da noi e scaricabile dalla sezione download del sito www.elettronica.in.it, oppure una versione modificata secondo le vostre esigenze, è pronto per essere caricato nel microcontrollore della scheda di

 **Tabella 1**
La matrice di costruzione dei simboli.

SIMBOLO	SEGMENTI (0=spento - 1=acceso)															
	A1	A2	B	C	D1	D2	E	F	G1	G2	H	J	K	L	M	N
0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	0	1	1	1	0	1	1	0	0	0	0	0	0
3	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	0
4	0	0	1	1	0	0	0	1	1	1	0	0	0	0	0	0
5	1	1	0	1	1	1	0	1	1	1	0	0	0	0	0	0
6	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0
7	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
9	1	1	1	1	1	1	0	1	1	1	0	0	0	0	0	0
A	1	1	1	1	0	0	1	1	1	1	0	0	0	0	0	0
B	1	1	1	1	1	1	0	0	0	1	1	0	0	0	0	1
C	1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0
D	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	1
E	1	1	0	0	1	1	1	1	1	1	0	0	0	0	0	0
F	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0
GI	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1
GS	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
GT	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

LA PROGRAMMAZIONE DELLA GETTONIERA DG600F

Prima di poter essere utilizzata nella nostra slot machine, la gettoniera che la correda va impostata in modo da generare impulsi che il microcontrollore della scheda possa interpretare come gli esatti valori delle monete introdotte. Di seguito vedete la procedura di programmazione.

- Premere il pulsante A (a sinistra del display) fino a far apparire le lettere CP.
- Premere il pulsante B (a destra del display) fino a far apparire le lettere CC, in tal modo tutti i parametri sono azzerati.
- Premere il pulsante A fino a far apparire le lettere CP.
- Premere il pulsante A una volta per leggere C1.
- Premere il pulsante B una volta e appare 00, premere di nuovo B per leggere 01 (1 impulso associato alla moneta da 10c).
- Passare ora 20 monete da 10c, possibilmente molto differenti nelle condizioni fisiche, maggiore è la varietà delle monete, migliore sarà poi il loro riconoscimento.
- Dopo la ventesima moneta il display mostra la lettera F ad indicazione della corretta memorizzazione del taglio.
- Il display ora mostra automaticamente C2.
- Premere il pulsante B una volta e appare 00, premere di nuovo B fino a leggere 02 (2 impulsi associati alla moneta da 20c).
- Passare ora 20 monete da 20c, possibilmente molto differenti nelle condizioni fisiche, maggiore è la varietà delle monete, migliore sarà poi il loro riconoscimento.
- Dopo la ventesima moneta il display mostra la lettera F ad indicazione della corretta memorizzazione del taglio.
- Il display ora mostra automaticamente C3.
- Premere il pulsante B una volta e appare 00, premere di nuovo B fino a leggere 04 (4 impulsi associati alla moneta da 50c).
- Passare ora 20 monete da 50c, possibilmente molto differenti nelle condizioni fisiche, maggiore è la varietà delle monete, migliore sarà poi il loro riconoscimento.
- Dopo la ventesima moneta il display mostra la lettera F ad indicazione della corretta memorizzazione del taglio.
- Il display ora mostra automaticamente C4.
- Premere il pulsante B una volta e appare 00, premere di nuovo B fino a leggere 07 (7 impulsi associati alla moneta da 1€).
- Passare ora 20 monete da 1€, possibilmente molto differenti nelle condizioni fisiche, maggiore è la varietà delle monete, migliore sarà poi il loro riconoscimento.
- Dopo la ventesima moneta il display mostra la lettera F ad indicazione della corretta memorizzazione del taglio.
- Il display ora mostra automaticamente C5.
- Premere il pulsante B una volta e appare 00, premere di nuovo B fino a leggere 10 (10 impulsi associati alla moneta da 2€).
- Passare ora 20 monete da 2€, possibilmente molto differenti nelle condizioni fisiche, maggiore è la varietà delle monete, migliore sarà poi il loro riconoscimento.
- Dopo la ventesima moneta, il display mostra la lettera F ad indicazione della corretta memorizzazione del taglio.
- Il display ora mostra automaticamente C6.
- Premere il pulsante A per due secondi, apparirà il numero 88 ad indicare la fine della programmazione.

Fatto ciò, la gettoniera è pronta per essere connessa al resto della slot machine e utilizzata.



gestione del sistema.

L'operazione si effettua mediante programmazione ISP, dopo aver settato i fuse del microcontrollore sui valori: low = FF, high = DF, ext = 07.

A chi non fosse ancora pratico di tale metodo di programmazione, consigliamo la lettura dell'articolo "ARDUINO E CIRCUITI STAND-ALONE", pubblica-

to dallo stesso autore di questo progetto, sul numero 159 della nostra rivista, tendendo conto che in quell'articolo veniva usata la versione IDE 0022, ma se usate la versione 1.0.5 (o 1.0.6) la tecnica è praticamente identica. Lo stesso articolo, ma con aggiornamenti per la versione 1.0.x nelle discussioni del blog, può essere visualizzato in lingua inglese

all'indirizzo <http://www.open-electronics.org/arduino-isp-in-system-programming-and-stand-alone-circuits>. Infine vogliamo segnalarvi un eccellente strumento didattico, consigliato e utilissimo alla realizzazione di circuiti stand-alone basati sui principali microcontrollori delle famiglie ATMEL ATmega e ATtiny: si tratta del manuale tecnico-pratico "ARDUINO E LE TECNICHE DI PROGRAMMAZIONE DEI MICROCONTROLLORI ATMEL", disponibile presso Futura Elettronica (www.futurashop.it/libro-arduino-e-le-tecniche-di-programmazione-dei-microcontrollori-atmel-8330-atproma).

IL FIRMWARE

Ciò stabilito, passiamo adesso all'analisi del firmware che gestisce la nostra slot machine, il quale, viste le funzioni in gioco, è relativamente e inevitabilmente complesso, ma comunque ben comprensibile; questo in quanto è stato scritto in modo ordinato e con l'ausilio di diverse "schede" specifiche. Descriviamo qui di seguito la struttura ed il funzionamento di base del firmware scaricabile dal sito della rivista.

SCHEDE PRINCIPALE

In questa abbiamo incluso due librerie: *Tlc5940.h* e *PinChangeInt.h*; la prima serve per pilotare con facilità gli integrati adottati per il pilotaggio dei display, appunto i TLC5940, mentre la seconda serve per poter usare i normali pin dell'ATmega328P con funzione di interrupt; questo microcontrollore dispone solo di due interrupt fisici: INTO (pin 4) e INT1 (pin 5), quindi la libreria ci permette di usare altri pin svolgendo la stessa funzione, anche se con

priorità inferiore.

Le note tecniche della libreria spiegano che bisogna valorizzare la variabile NUM_TLCS del file tlc.

config.h, con il numero di TLC5940 che compongono la catena; nel nostro caso la riga diventa:

```
#define NUM_TLCS 5.
```

Dopo l'inclusione delle librerie abbiamo la dichiarazione delle COSTANTI che servono per preimpostare il valore del PWM del driver e di conseguenza la luminosità dei display (light) il numero di simboli dei display alfanumerici (noi ne abbiamo previsti 19) e dei display numerici (10), il numero di segmenti degli stessi display (16 e 7). Ancora abbiamo il valore di crediti oltre il quale la gettoniera si disattiva (50), il numero di crediti da raggiungere per la vincita (95) e per finire i punteggi assegnati alle combinazioni di simboli previste per le giocate.

Tutti questi valori sono personalizzabili, in base alle esigenze dell'utente.

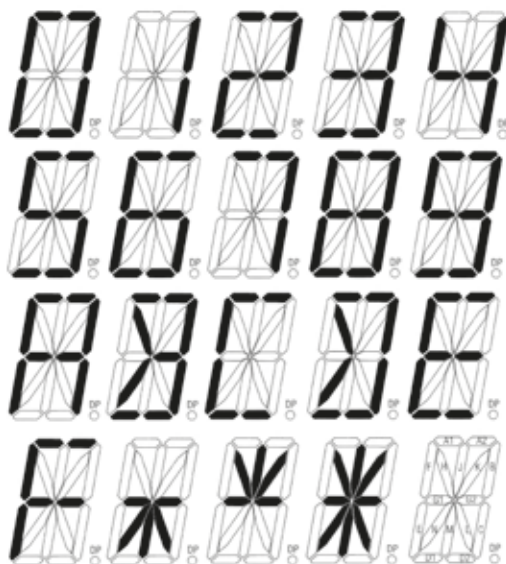
A seguire abbiamo la matrice booleana di creazione dei simboli; in pratica ogni riga rappresenta il simbolo riportato con un commento alla fine della stessa (il primo è lo "0" e l'ultimo lo abbiamo chiamato "GT"), ed è costituita da una serie di 0 e 1 che serviranno per spegnere (0) o accendere (1) il relativo segmento (il cui elenco è riportato sulla riga precedente alla dichiarazione della matrice, e va da F a G1). Come è facilmente intuibile il numero di righe potrebbe essere aumentato notevolmente con tutte le altre lettere dell'alfabeto (attualmente si ferma alla F) e con tutta una serie di simboli "grafici" realizzati con le più disparate combinazioni di uno o più segmenti che non rappresentino un numero o una lettera (noi ne abbiamo realizzati tre, a titolo di esempio). Nella **Tabella 1** abbiamo riportato la nostra matrice in forma tabellare per rendere più semplice la comprensione della generazione simboli, mentre in **Fig. 1** sono visibili le combinazioni grafiche dei segmenti dei display.

Naturalmente è possibile aumentare il numero di simboli, aggiungendo una riga per ognuno di essi, bisognerà poi modificare il valore della costante "n_symgr".

A questo punto vengono impostate le due matrici per i simboli assegnati ai quattro display alfanumerici ed ai due display numerici ed a seguire tutte le numerose variabili utilizzate nel programma.

Questa sezione viene chiusa con la dichiarazione dei pin che abbiamo descritto nella descrizione dello schema elettrico, ai quali sono collegati: i 5 pulsanti arcade, lo switch di fine corsa, il pin di inibizione della gettoniera, quello di attivazione della riproduzione audio, quello di abilitazione del driver

Fig. 1
Sono visibili le combinazioni grafiche dei segmenti dei display.



del motore, quelli per il controllo del movimento del motore.

Da notare la non corrispondenza numerica dei pin elencati nella descrizione dello schema elettrico con quelli usati nel firmware; il motivo risiede nel fatto che, anche se il nostro circuito utilizza il microcontrollore in modalità stand-alone e senza bootloader, operiamo pur sempre in ambiente Arduino. Per rendere chiara la corrispondenza tra i pin fisici dello schema elettrico e quelli logici del firmware, riportiamo in **Fig. 2** un'immagine tratta dal già citato manuale "ARDUINO E LE TECNICHE DI PROGRAMMAZIONE DEI MICROCONTROLLORI ATMEL".

L'immagine mostra all'interno del riquadro la numerazione dei pin fisici del microcontrollore (uguale a quella degli altri modelli della stessa famiglia, denominata xx8), e all'esterno le corrispondenti funzioni. Per esempio il pin fisico 4 corrisponde al pin digitale 2 e anche all'Interrupt 0. Ancora il pin fisico 18, che nel nostro schema è stato utilizzato per abilitare o disabilitare la gettoniera, corrisponde al pin digitale D12 (oltre al segnale MISO per la programmazione ISP); infatti nel nostro firmware la riga "const byte inh = 12;" è appunto quella che assegna il nome **inh** al pin logico 12 (quindi al pin fisico 18) che sarà usato per pilotare il transistor T1, come meglio spiegato nella sezione dedicata allo schema elettrico.

Bene, a questo punto crediamo che i più esperti tra voi avranno notato che i pin fisici 5, 14, 15, 16 e 19, destinati al pilotaggio della catena di integrati TLC5940, non sono stati dichiarati nel firmware; non si tratta, naturalmente, di una dimenticanza, ma di una scelta il cui motivo è semplice: si occupa di tutto la libreria che abbiamo incluso nel firmware.

Sezione SETUP

Passiamo adesso all'analisi del SETUP: in questa sezione vengono attivate la libreria del TLC5940, la gettoniera e l'interrupt fisico relativo agli impulsi provenienti dalla gettoniera, quando in essa vengono inserite una o più monete.

A seguire vengono attivati gli interrupt software relativi allo switch di finecorsa del cassetto delle monete e al pulsante 1, che assume in questo caso la funzione di interruttore on/off per la sezione audio hardware della slot machine.

Sempre nel SETUP, vengono impostate le funzioni di input (con attivazione delle pull-up interne) o output dei vari pin digitali del micro.

Infine sono presenti due righe, una esegue una routine che accende e spegne per tre volte tutti

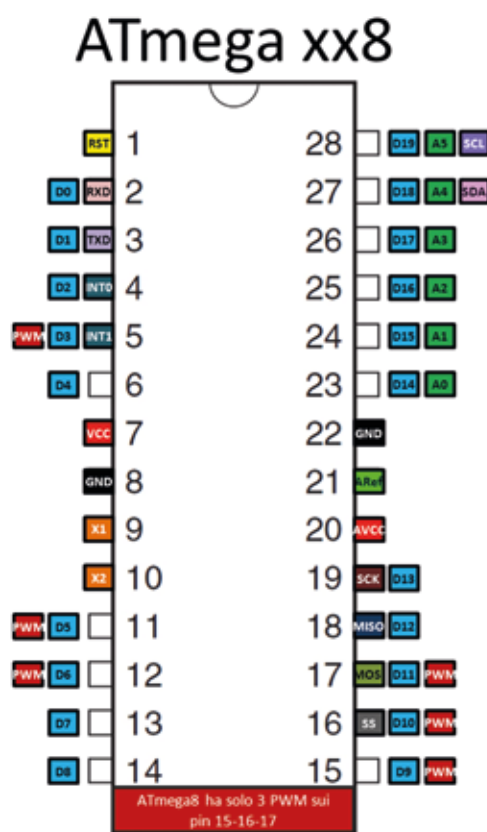


Fig. 2
Corrispondenza
dei pin tra
Arduino e
ATmega328P-PU.

i segmenti dei sei display, per la verifica del buon funzionamento della sezione visiva, ed una esegue una routine per il controllo della posizione del cassetto delle monete (routine "PULL_OPENING"); in pratica se il cassetto è aperto (non ha importanza quanto) lascia aperto anche lo switch di finecorsa, e quindi il motore viene attivato e fatto girare in modo da far rientrare il cassetto; appena lo switch risulterà premuto la routine si fermerà. A tal proposito vediamo i comandi per utilizzare i tre pin del driver dello stepper motor, al fine di gestire il movimento dello stesso e, quindi, del cassetto delle monete. In condizioni di riposo il pin "enable" è impostato su HIGH, quindi il driver è disabilitato e la condizione degli altri due pin ("enable" e "step") è ininfluente. Per far muovere il motore occorre prima decidere la direzione della rotazione, nel nostro caso il pin "direction" settato su HIGH gira in direzione antioraria, e quindi fa chiudere il cassetto, mentre settato su LOW gira in direzione oraria, provocando l'apertura del cassetto. Quindi si abilita il driver, ponendo "enable" su LOW, ed infine si inviano gli impulsi per il movimento, attivando un segnale pwm "analogWrite(stp, 10)" per il pin "step".

Sezione LOOP

In questa porzione del codice c'è il programma principale, che andiamo a descrivere nella stessa sequenza in cui è scritto nello sketch. All'accensione, la macchina, dopo il test dei display e l'eventuale chiusura del cassetto delle monete, entra in una fase ciclica, in attesa dell'inserimento di una o più monete per poter iniziare la partita. Il loop è costituito da dieci cicli durante i quali i display alfanumerici assumono altrettante serie di combinazioni assolutamente casuali, mentre i display numerici mostrano l'assenza di credito (un doppio "0"); alla fine dei dieci cicli viene rieseguita la routine iniziale di accensione e spegnimento dei display, e quindi inizia un nuovo loop; se l'audio è stato attivato i simboli sui display si fermano e si sente uno dei suoni delle combinazioni vincenti previste dal firmware. Inizialmente l'audio è disattivato, per attivarlo bisogna premere il pulsante 1 durante il loop, nella fase in cui i caratteri dei display sono in "movimento"; per spegnere nuovamente l'audio è sufficiente premere ancora il pulsante 1, sempre durante il loop.

Per poter descrivere in modo comprensibile tutto il restante codice del nostro firmware, a questo punto conviene iniziare a giocare!

ORA FACCIAMO UNA PARTITA!

In qualsiasi momento della fase di loop appena descritta è possibile inserire una o più monete, anche di diverso taglio; in tal caso il loop terminerà, i display alfanumerici si spegneranno (questa operazione viene effettuata dalla routine "NO_SYMBOLS" localizzata nella scheda "DISPLAY") mentre quelli numerici mostreranno il credito derivante dalle monete inserite nella gettoniera della slot machine. In tal caso la conversione degli impulsi delle monete in credito partita viene effettuata dalla routine "VALUE_CONVERT" localizzata nella scheda "COINACC", mentre la visualizzazione del credito è gestita dalla routine "CREDIT_DISPLAY" e attivata dalla routine "TLC_ASSIGN", entrambe localizzate nella scheda "DISPLAY". Poiché il credito delle monete si somma con quello acquisito con le giocate vincenti, ovviamente abbiamo previsto un limite per l'inserimento delle monete, quindi se si raggiunge (sia in fase di loop che durante la partita) il numero di 50 crediti la gettoniera verrà inibita (accensione del LED rosso) ed eventuali successive monete saranno rifiutate e restituite al giocatore; l'attivazione e inibizione della gettoniera sono gestite rispettivamente dalle routine COIN_ENABLE e COIN_DISABLE, localizzate nella scheda "COINACC".

Questo numero permette di partire già con un credito molto elevato, volendo rendere più difficile la partita lo si può abbassare agendo sulla costante "max_amount". Dal momento in cui si inserisce la prima moneta non sarà più possibile cambiare lo stato dell'audio, per cui se è disattivo tale resterà (escluso il caso di vincita, come vedremo), mentre se è attivo si potrà sempre abbassare o azzerare il volume dall'apposito potenziometro.

La partita inizia premendo il pulsante START (nel nostro caso è quello rosso); ogni giocata costa un credito, che viene immediatamente scalato dal totale. La routine "FOUR_SYMBOLS", localizzata nella scheda "DISPLAY" genera una sequenza assolutamente casuale di quattro simboli sui display alfanumerici. Ci soffermiamo su questa operazione, a scopo meramente didattico, perché l'ottima tecnica da noi adottata per ottenere la casualità potrebbe tornare utile in molte altre applicazioni, dove sia necessario ottenere dei valori realmente casuali e quindi la parte di firmware corrispondente la potrete reimpiegare in molti altri sistemi, come ad esempio dei token per generare password casuali o chiavi di accesso di vario genere.

La tecnica consiste nel leggere i quattro pin analogici AO+A3 (pin fisici 26-24-25-26) del microcontrollore, ed utilizzare i quattro valori letti per inizializzare il generatore di numeri random, facendolo partire da una posizione qualsiasi della sequenza casuale che, benché lunga e realmente casuale, è sempre la stessa; questa operazione, suggerita dal Reference del sito ufficiale di Arduino (<https://www.arduino.cc/en/Reference/RandomSeed>) seguita dal comando "random", permette di ottenere quattro simboli iniziali realmente casuali ad ogni nuova partita; il comando random è naturalmente limitato a 19 valori (da 0 a 18) dalla costante "n_symgr" e selezionerà quindi quattro simboli dalla matrice già descritta.

Naturalmente, proprio perché casuali, questi simboli potranno essere anche uguali tra loro, il che fornirà già un punteggio iniziale.

Nella descrizione dello schema elettrico accennavamo al fatto di aver volutamente omesso il condensatore da 100 nF tra il pin 21 (AREF) del micro U1 e la massa; tale condensatore serve per garantire stabilità all'ADC interno del micro, mentre a noi serve la maggior instabilità possibile per avere la massima casualità nelle letture.

La valutazione della combinazione uscita viene effettuata dalla routine "POINT_CALC" localizzata nella scheda "SCORE", e l'eventuale punteggio acquisito viene istantaneamente sommato al cre-



dito attuale. Questa routine è molto importante in quanto è su di essa che si dovrà agire per aumentare o diminuire le possibilità di combinazioni vincenti, mentre per variare semplicemente i punti attribuiti alle varie combinazioni sarà sufficiente modificare i valori delle costanti PT1÷PT4.

In caso di combinazione vincente, se l'audio è attivo verrà prodotto quello associato al punteggio ottenuto, ad opera della routine "POINT_SOUND" localizzata nella scheda "AUDIO".

Se la combinazione vincente dovesse riguardare tutti e quattro i display (per esempio due coppie o un poker di simboli uguali), si potrà passare subito alla prossima giocata premendo nuovamente il pulsante START. In caso contrario, per sostituire (una o più volte) uno o più simboli, si potrà agire sui pulsanti P1÷P4, fino ad un massimo di quattro tentativi totali; quindi per esempio, se si realizza un tris di simboli, si può usare il pulsante del simbolo da sostituire fino a quattro volte per cercare di realizzare un poker di simboli.

Se invece si realizza una coppia, si possono usare i due pulsanti dei simboli da sostituire due volte ciascuno o tre volte uno ed una volta l'altro, sempre se necessario. Infine se la combinazione è perdente si possono usare i quattro pulsanti a piacimento per tentare di ottenere una qualsiasi combinazione vincente. La **Tabella 2** mostra le combinazioni vincenti con i relativi punteggi, impostati attualmente nel nostro firmware, ma eventualmente modificabili a

Combinazione simboli uguali	Punteggio
coppia	2
due coppie	5
tris	10
poker	20

Tabella 2
Le combinazioni vincenti della SLOT MACHINE.

piacimento, agendo sulle opportune variabili e sulla routine "POINT_CALC", già descritta.

Ad ogni pressione di uno dei 4 pulsanti corrisponde la sostituzione casuale del relativo simbolo, che viene sostituito nella combinazione mediante la routine "ONE_SYMBOL" localizzata nella scheda "DISPLAY"; tale routine in realtà viene eseguita all'interno di una routine più complessa, la "REPLACE" (localizzata nella scheda "SCORE"), che dopo aver eseguito la "ONE_SYMBOL", lancia la "POINT_CALC", già descritta in precedenza. Subito dopo, una volta aggiornato il credito, si verifica se esso raggiunge o meno il "max_amount" al fine di disabilitare o abilitare la gettoniera.

Se si ottiene la combinazione desiderata prima di aver usato tutti i quattro tentativi a disposizione, basterà premere il pulsante START per avviare una nuova giocata; in caso contrario, al quarto tentativo effettuato i pulsanti P1÷P4 verranno disattivati e tutte le variabili saranno resettate, la condizione di fine giocata sarà mostrata con lo spegnimento dei display, per qualche istante, per poi mostrare



Fig. 3
La Slot Machine completa.

Fig. 4
La Slot Machine
vista dall'interno del
contenitore.



nuovamente l'ultima combinazione ottenuta. A questo punto è possibile premere il tasto START per iniziare una nuova sessione di gioco. La partita terminerà quando il credito diventerà zero oppure raggiungerà o supererà il valore 95 (costante "win"). Nel primo caso l'indicazione sui display numerici avverrà ad opera della routine "ZERO_CREDITS", la slot machine entrerà nuovamente in loop e sarà riabilitata la possibilità di gestire l'audio; inoltre si potranno inserire nuove monete per ricominciare a giocare. Nel secondo caso, a prescindere dalla condizione della sezione audio (attivata oppure disattivata...) verrà eseguito il ciclo sonoro previsto in caso del verificarsi della vincita (nel nostro caso sarà l'esecuzione dell'intera registrazione contenuta nel chip audio), quindi la routine "PULL_OPENING" provocherà l'apertura motorizzata del cassetto delle monete (routine "MOTOR_ON"), che resterà aperto 5 secondi (routine "MOTOR_OFF") durante i quali il giocatore dovrà cercare di prelevare il maggior numero di monete possibile. Subito dopo la routine "PULL_CLOSING" farà girare il motore in senso contrario ed il cassetto si richiuderà, arrestandosi nel momento in cui la sua parete

posteriore premerà lo switch di finecorsa; naturalmente il giocatore dovrà togliere le mani non appena inizierà la chiusura, sia per non farsi male, sia per non danneggiare la meccanica di azionamento del cassetto ed anche, perché no, per correttezza. Le ultime routine descritte sono tutte localizzate nella scheda chiamata "PULLER". Dopo la chiusura del cassetto, il credito verrà azzerato e la slot-machine tornerà nella modalità LOOP: sarà quindi pronta per farvi cominciare una nuova partita, se lo vorrete.

CONCLUSIONI

Bene, a questo punto abbiamo terminato la descrizione del nostro progetto e del suo funzionamento caratteristico; come abbiamo più volte sottolineato, il firmware, che è fornito in versione sorgente e che abbiamo commentato adeguatamente affinché possa comprenderlo a fondo, può essere personalizzato in alcune costanti, ma anche fortemente modificato, qualora decideste di apportare modifiche importanti all'hardware o al funzionamento da noi previsto, per ottenere una versione tutta vostra del gioco.

Inoltre la versatilità del firmware e la ricca dotazione hardware, potrebbero permettere a chi volesse cimentarsi nel farlo, di trasformare la slot machine qui proposta in qualcosa di differente: per esempio in un gioco a gettoni ma dal funzionamento differente, in un erogatore, in uno strumento didattico e in altro ancora.

Buon lavoro e buon divertimento!

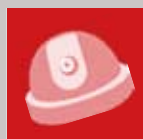


Cosa occorre?

Il materiale presentato in questo articolo è disponibile presso Futura Elettronica. L'alimentatore switching 12V 3A (cod. ADL-40-12) è in vendita al prezzo di Euro 8,50, il modulo step-down 5V (cod. MP1584ENDCDC) è disponibile a Euro 2,50, la gettoniera programmabile (cod. GETTON1) costa Euro 69,00, il modulo audio ISD1820 (cod. YB390) è in vendita al prezzo di Euro 4,50, l'amplificatore audio 3+3 W (cod. PAM8403POT) è disponibile a Euro 6,50, la coppia di altoparlanti (cod. AP5243W) costa Euro 2,90 caduno, il motore passo-passo (cod. STEPMOT03) è in vendita al prezzo di Euro 18,00, la staffa per NEMA17 (cod. SMXNEMA17) è disponibile a Euro 3,50, il giunto per NEMA17 (cod. GIUNTO8X5NEW) costa Euro 9,50 caduno, la barra filettata diametro 8mm (cod. BARRA8) è in vendita al prezzo di Euro 8,00. 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>



Sfoggia on-line i nostri cataloghi

**FUTURA
ELETTRONICA®**



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.

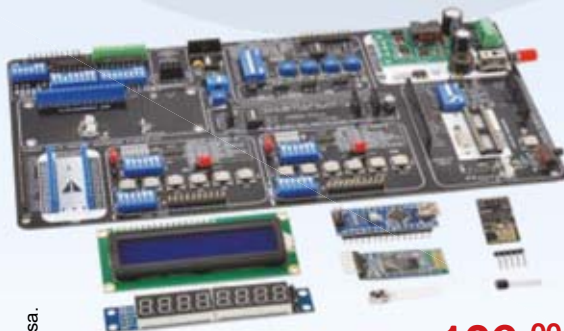
BOARD ATTINY85 USB



€ 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

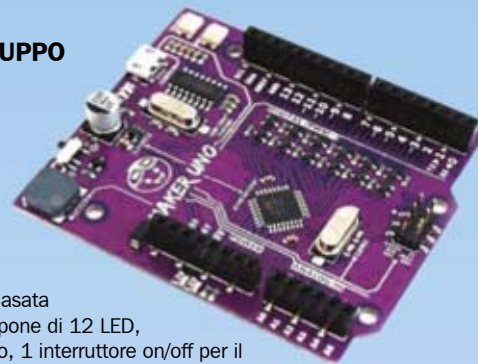


€ 106,00
cod. ARDUINODEMO

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



Workstation con tutto quello che serve per imparare a utilizzare Arduino, sviluppare attività software e hardware impiegando un unico supporto PCB. Fornisce diverse soluzioni hardware per connettere gli accessori Arduino, senza alcuna necessità di saldature. Si collega direttamente al PC tramite cavo USB e si programma attraverso l'IDE Arduino.

Prezzi IVA inclusa.

SCOPRI TUTTE LE BOARD SU WWW.FUTURASHOP.IT

IoT WATER LEVEL

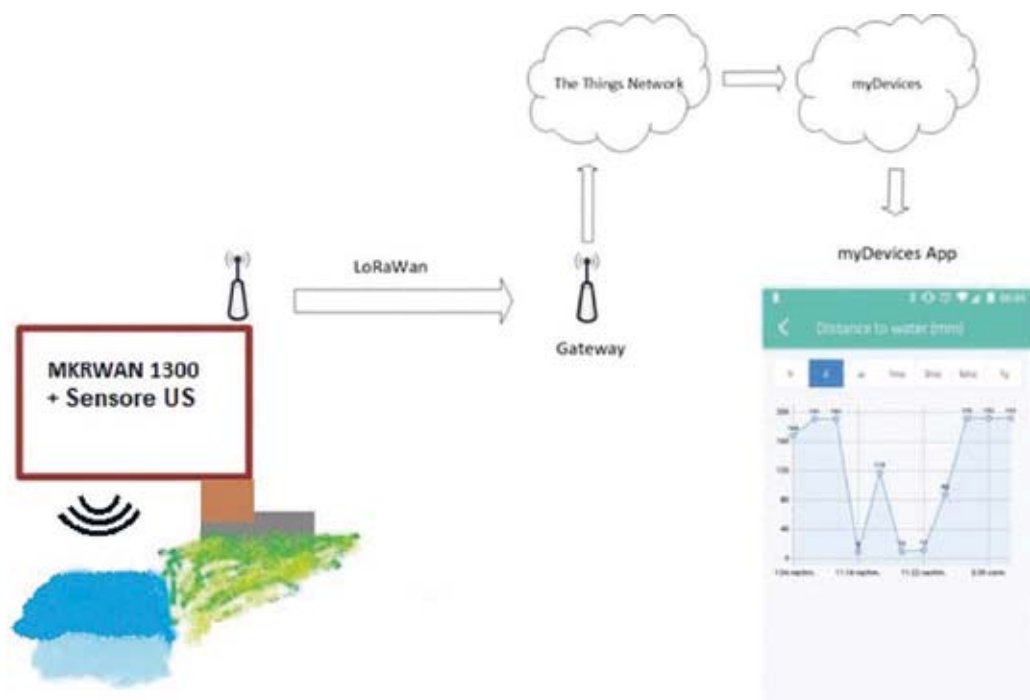


di ING. MIRCO SEGATELLO

Dispositivo IOT con tecnologia LoRaWAN per il monitoraggio del livello dell'acqua di un laghetto o di un corso d'acqua

Nel Corso dedicato all'IoT, abbiamo fatto una panoramica su diverse tecnologie di interazione degli oggetti mediante connessioni wireless e in particolare ci siamo soffermati, nella terza puntata, sul link chiamato LoRaWAN. Questa sistema di trasmissione consente di realizzare reti su portante a radiofrequenza capaci di operare con potenze ridottissime, riuscendo però a coprire distanze consistenti, a tutto vantaggio della riduzione dei consumi dei dispositivi. Nella stessa puntata del corso IoT vi abbiamo esposto l'impiego pratico della tecnologia wireless LoRaWAN utilizzando, allo scopo, una scheda Arduino MKR WAN 1300 per implementare un'applicazione dimostrativa. Nell'esempio proposto ci siamo soffermati su una semplice

Fig. 1
Schema di cablaggio
dei componenti
utilizzati.

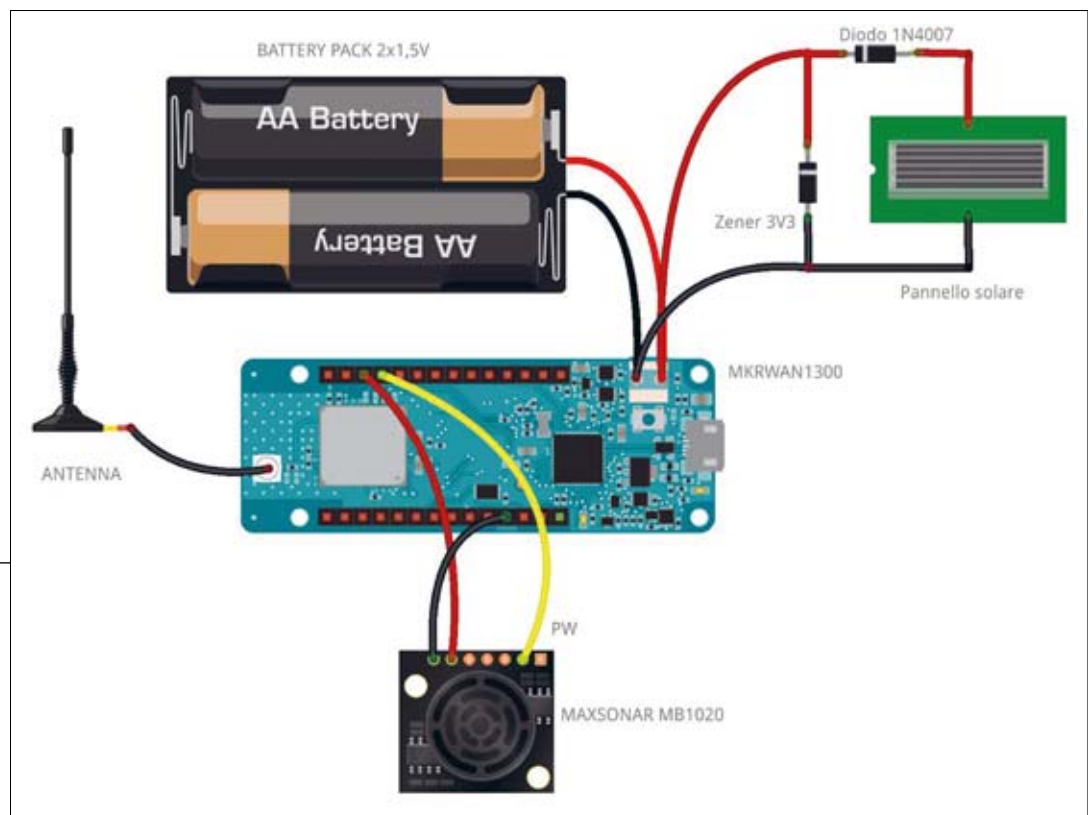


applicazione che prevedeva la lettura di un segnale analogico, successivamente inviato al portale di TheThingsNetwork, in modo da poterlo visualizzare da remoto con qualsiasi smartphone connesso a Internet.

IL NOSTRO PROGETTO

Vorremmo partire ora da quell'esempio, per costruire una nuova applicazione che permettesse il monitoraggio del livello dell'acqua in un corso d'acqua o in un lago, con la possibilità di lanciare

Fig. 2
Schema di cablaggio hardware
con il sensore
MB1020.



un allarme nel caso il livello dovesse scendere al di sotto o superare una soglia preimpostata. Lo schema di cablaggio visibile in **Fig. 1** prevede che il nostro device periferico IoT sia posto in un punto rialzato dell'invaso e tramite un sensore ad ultrasuoni, misuri la distanza tra il dispositivo ed il livello dell'acqua; nota la distanza tra il sensore il fondo dell'invaso, sarà possibile determinare l'altezza del liquido. Tale tecnica viene oggi utilizzata in molte applicazioni in cui è richiesta la rilevazione del livello in un serbatoio o una cisterna ed è preferita al classico galleggiante che aziona un potenziometro.

Il valore misurato sarà inviato ad un gateway LoRa WAN, che a sua volta instraderà il dato verso il network server, che nel nostro caso è TheThingsNetwork (TTN). La visualizzazione dei dati su TTN è limitata, inoltre non è possibile configurare degli avvisi, molto importanti se si monitorizzano variabili importanti. Per questo motivo dobbiamo appoggiarci ai servizi di un Application Server che sia compatibile con TTN come ad esempio il servizio offerto da Cayenne, che avevamo già utilizzato in una delle puntate del corso IoT.

Per quanto riguarda l'hardware (**Fig. 2**) avremo bisogno di una scheda di sviluppo Arduino MKRWAN1300 e di un sensore ad ultrasuoni di buona qualità che ci permetta un elevato range di misura con bassi consumi, come ad esempio il modello MB1020 della MaxSonar oppure il model-

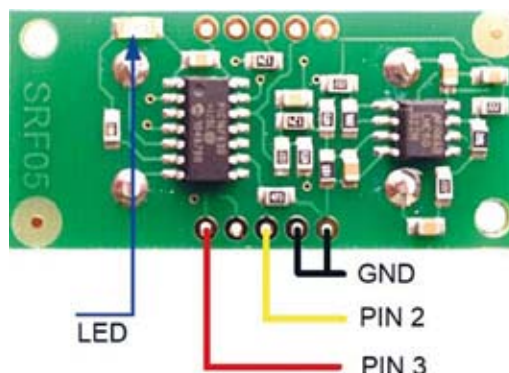


Fig. 3
Schema di
montaggio del
sensore SRF05.

lo SRF05 distribuito dalla ditta Futura Elettronica con codice 7300-SRF05 (**Fig. 3**). La scheda MKR WAN 1300 può essere alimentata con una tensione massima di 6 volt tramite il pin Vin che però fa capo ad uno stabilizzatore di tensione che ne incrementa il consumo.

Nel caso volessimo alimentare la scheda tramite delle batterie, sarà meglio optare per la soluzione proposta dalla stessa Arduino, ovvero due batterie stilo da 1,5 volt poste in serie tali da garantire circa 3V di alimentazione. La soluzione con una cella Li-Po è da scartare visto che a piena carica può raggiungere i 4,2 volt. Vista l'applicazione all'aperto, se optiamo per due batterie ricaricabili Ni-MH potremmo sfruttare il fatto che esse sopportano la carica continua (purché di valore limitato) ed abbinarci un piccolo pannello solare che possa garantire un'alimentazione praticamente illimitata.

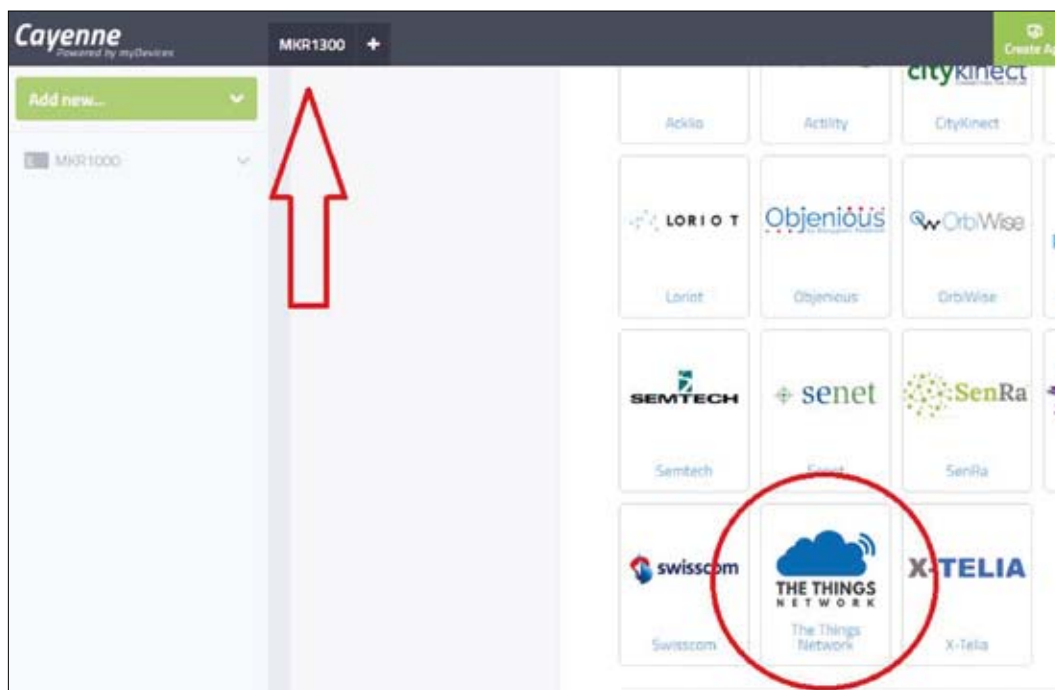


Fig. 4
Aggiunta di un
dispositivo TTN al
progetto Cayenne.

Fig. 5
Inserimento del riferimento per il dispositivo TTN.

Fig. 6
Inserimento dei riferimenti del dispositivo.

In questo caso non sono necessari circuiti di ricarica, basterà semplicemente aggiungere un diodo contro l'inversione di polarità nel pannello solare ed un diodo Zener che garantisca che alla scheda Arduino non giungano più di 3,3 volt.

L'architettura Arduino non prevede un hardware che possa funzionare in ultra low-power mode, pertanto non sarà possibile il funzionamento esclusivamente a batteria per diversi mesi, come invece potrebbe essere per il solo modem LoRa. Per quanto riguarda il sensore ad ultrasuoni abbiamo previsto che la sua alimentazione giunga da un pin digitale della scheda in modo da comandarne l'accensione nel solo momento in cui serve la lettura della distanza.

Qui di seguito elenchiamo il materiale occorrente alla realizzazione del progetto.

- Arduino MKRWAN 1300 con antenna;
- Sensore ultrasuoni SRF05 oppure MB1020;
- Portabatterie per 2 batterie formato AA;
- Batterie Ni-MH formato AA;
- Pannellino solare 5,5 volt 0,5 watt;
- Diodo 1N4007;
- Diodo Zener da 3,3volt.

Prima di procedere a livello software è necessario rendere operativo l'esempio riportato nella terza puntata del corso IOT, seguendo tutte le indicazioni riportate, compreso l'installazione delle librerie per Arduino e la configurazione dell'account TTN. Per verificare che tutto funzioni correttamente vi consigliamo di caricare sulla scheda MKR l'esempio **LoRa_Level_US.ino** che consente di visualizzare su TTN il valore misurato (in centimetri) del sensore ad ultrasuoni. Ricordatevi di inserire, nella tab payload format, la riga seguente:

```
decoded.level=(bytes[1]<<8) + bytes[0]
```

Accediamo a <https://mydevices.com> e se non lo avessimo già fatto creiamo un nuovo account. Il successivo passaggio è aggiungere un nuovo progetto ed al progetto aggiungere un dispositivo di tipo The Things Network, come indicato in **Fig. 4**. Successivamente andremo a specificare che questo nuovo dispositivo comunicherà in modalità Cayenne LPP; dovremo inserire il nome del dispositivo (MKR1300 nel nostro caso) e il device EUI ricavato dalla risposta su Serial Monitor in corrispondenza della riga **Serial.println(modem.deviceEUI)**, come appare in **Fig. 5**.

A questo punto il sistema rimane in attesa di ricevere i dati dal nuovo dispositivo, pertanto dovremo configurare TTN usufruendo della scheda **Integrations**. Cliccate su **Add integration** e selezionate Cayenne (sappiate comunque che è anche disponibile il servizio IFTTT), vi verrà chiesto di inserire il Process ID che troverete nell'indirizzo Internet del

progetto Cayenne, invece per l'Access Key potete cliccare su **Default Key** (Fig. 6). Confermate l'integrazione del dispositivo e verificate che lo Status sia in Running, allorché vi apparirà una schermata come quella visibile in Fig. 7.

PREPARIAMO IL FIRMWARE

Non ci resta che scrivere lo sketch per Arduino e per fare questo è necessario includere la libreria CayenneLPP (Fig. 8). La sigla LPP sta per Low Power Payload ovvero una modalità nella quale i dati da trasmettere sono codificati nel minor numero di byte al fine di ridurre il bit rate nella trasmissione, requisito fondamentale in una rete LoRa WAN. Questa libreria svolge il compito di rendere compatibile il dato inviato dalla scheda Arduino con Cayenne Low Power Payload, che è il formato scritto da myDevices per l'integrazione dei nodi in LoRaWAN. Ci servirà anche la libreria Arduino Low Power per sfruttare la modalità a basso consumo (deep sleep) delle schede MKR, come indicato in Fig. 9. La dashboard di Cayenne si aggiornerà in automatico visualizzando gli ultimi valori acquisiti, come visibile in Fig. 10; sarà anche possibile personalizzare i dati ricevuti (Fig. 11) e visualizzare un grafico con l'andamento temporale dei valori, dove in automatico sono aggiunti i campi relativi all'intensità e alla qualità del segnale, come visibile

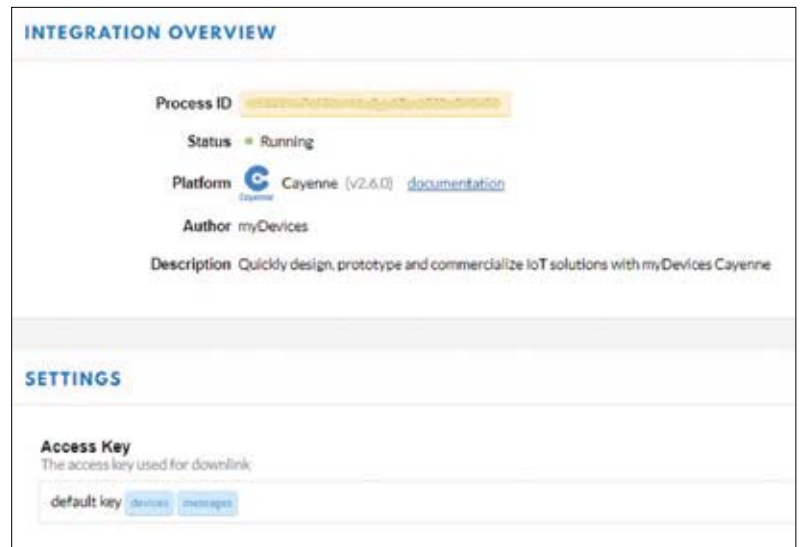


Fig. 7
Procedura di integrazione del dispositivo Cayenne completata.

in Fig. 12. Alcune limitazioni della libreria e della dashboard di Cayenne non permettono di modificare l'unità di misura della grandezza visualizzata, il che non ci permette di inserire quelle nostrane. Un altro punto importante da affrontare riguarda il consumo in corrente della board MKR in previsione di alimentarla tramite delle batterie. La scheda MKR1300 è predisposta per funzionare con un'alimentazione esterna fornita da una batteria che bypassa lo stabilizzatore di tensione ed alimen-



Fig. 8
Installazione libreria CayenneLPP.

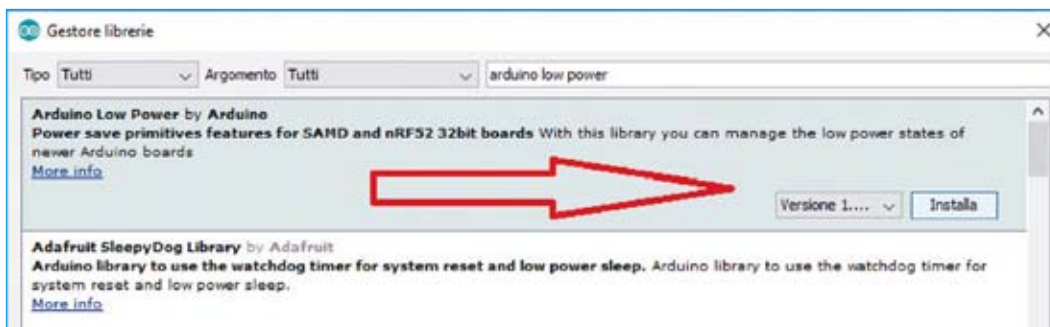


Fig. 9
Installazione libreria Arduino Low Power.

→ **Fig. 10**
Elenco delle
trasmissioni con TTN
(tab Application Data).

APPLICATION DATA						
Filters						
	uplink	downlink	activation	ack	error	
time	counter	port				
▲ 18:31:11	3	2	dev id:	mkrwan1300-01	payload:	01 65 00 44 02 02 01 2D
▲ 18:26:10	2	2	dev id:	mkrwan1300-01	payload:	01 65 00 44 02 02 01 1E
▲ 18:21:10	1	2	dev id:	mkrwan1300-01	payload:	01 65 00 44 02 02 01 1E
▲ 18:16:08	0	2	dev id:	mkrwan1300-01	payload:	01 65 00 44 02 02 01 1E
★ 18:16:00			dev id:	mkrwan1300-01	dev addr:	26 01 22 6B app eui: 70 B3D57ED0018BCA

ta direttamente la CPU, con il vantaggio di non avere componenti che assorbano corrente anche in standby. Il LED di alimentazione è connesso ai +5volt e rimane spento quando la scheda è alimentata a batteria, mentre rimangono alimentati il modem LoRa e l'integrato ECC508 (Crypto-Authentication) e il microcontrollore SAM21 che

fortunatamente in standby assorbono circa 1mA. Per impostare la scheda in modalità sleep ci viene incontro la libreria Arduino Low Power specificatamente scritta per le schede con MCU SAM, la quale si appoggia alla libreria RTCzero per temporizzare il tempo di risveglio, in questo modo la scheda è attiva per il solo tempo necessario a leggere il sensore, la tensione della batteria ed inviare i dati al server. Considerando che potremmo inviare i dati ogni ora la CPU sarà in sleep per il 99,9% del tempo. Siamo ora pronti a caricare lo sketch definitivo di nome *LoRa_Level_US_Cayenne_LowPower.ino*, possiamo farlo tranquillamente con la scheda connessa via USB al PC ed il pacco batterie collegato. Ricordate che durante la fase di sleep della scheda la seriale è inibita pertanto non sarà possibile procedere alla programmazione, in questo caso è necessario procedere al reset della scheda prima di avviare il download dello sketch azionando due volte consecutivamente il pulsante di reset nella scheda. Se apriamo il tab Application Data nel portale TTN potremmo vedere il primo collegamento della scheda (c'è un piccolo simbolo di fulmine sulla sinistra) e poi via via i dati ricevuti ad intervalli di un minuto, il payload è costituito da soli 8 byte

→ **Fig. 11**
Personalizzazione dei
dati ricevuti.

Water Level

General

Widget Name

Water Level

Channel

1

Channel Widget

Value

Channel Unit

Tank

Channel Color

Celsius

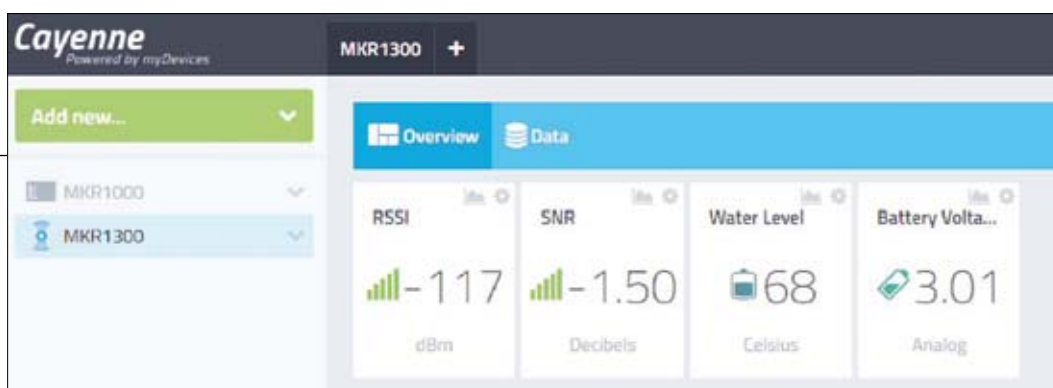
Number of decimals

0

Remove

Save

→ **Fig. 12**
Schermata di Cayenne
con i dati ricevuti da
MKR1300.



Timestamp	Device	Chan...	Sensor Name	Sensor ID	Data	Unit	Values
2019-03-13 7:24:38	MKR1300	101	SNR	570x560-432b-11e9...	snr	db	2.25
2019-03-13 7:24:38	MKR1300	100	RSSI	571a20f0-432b-11e9-a...	rssi	dbm	-117
2019-03-13 7:24:38	MKR1300	101	SNR	570x560-432b-11e9...	snr	db	2.25
2019-03-13 7:24:38	MKR1300	1	Water Level	57177280-432b-11e9...	lum	lux	68
2019-03-13 7:24:38	MKR1300	2	Battery Voltage	42c54650-4355-11e9...		null	3.01
2019-03-13 7:24:38	MKR1300	100	RSSI	571a20f0-432b-11e9-a...	rssi	dbm	-117
2019-03-13 6:24:20	MKR1300	2	Battery Voltage	42c54650-4355-11e9...	analog_v...	null	3.01
2019-03-10 6:51:16	MKR1300	1	Water Level	57177280-432b-11e9...	lum	lux	68

Fig. 13
Tabella con i
dati raccolti da
Cayenne.

pur riportando quattro valori numerici. In questo sketch abbiamo aggiunto la variabile zeroLevel che rappresenta la distanza del sensore dal fondo dell'invaso e in base a questo valore calcoleremo l'altezza (sempre positiva) del liquido. L'invio dei dati ogni minuto è necessario solo per una diagnostica del sistema, successivamente l'invio dei dati ogni ora andrà più che bene e non incapperemo nella limitazione del massimo numero di byte giornalieri. Sarà possibile visualizzare dei grafici personalizzati (ad esempio riportanti i dati relativi all'intensità del segnale RF) oppure l'elenco dei dati ricevuti per l'esportazione come in Fig. 13. E' anche possibile aggiungere un'allerta quando si verifica una determinata condizione, ad esempio il livello dell'acqua è troppo basso. Su Cayenne, dal menù in alto a sinistra, clicchiamo sulla voce **Add new Trigger** e, dalla schermata che si apre così facendo, nella sezione **IF** selezioniamo il dispositivo e la variabile associata al trigger, il valore di soglia e se l'evento dovrà verificarsi al di sopra o al di sotto di questa soglia (Fig. 14). Nella sezione **then** selezioniamo il tipo di notifica che vogliamo ricevere tra un SMS e/o una e-mail. Se avete selezionato l'invio di una e-mail (ma è lo stesso con gli SMS) riceverete l'avviso non appena la variabile indicata, nel nostro caso Water level, dovesse scendere sotto la soglia impostata; gli avvisi si susseguono ad ogni arrivo di dato aggiornato che dovesse rispettare l'IF, come visibile in Fig. 15.

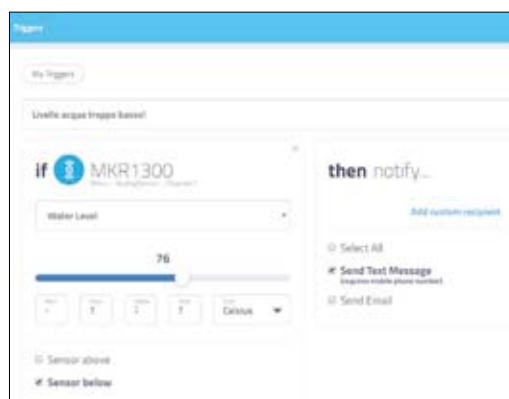


Fig. 14
Impostazioni del
trigger.

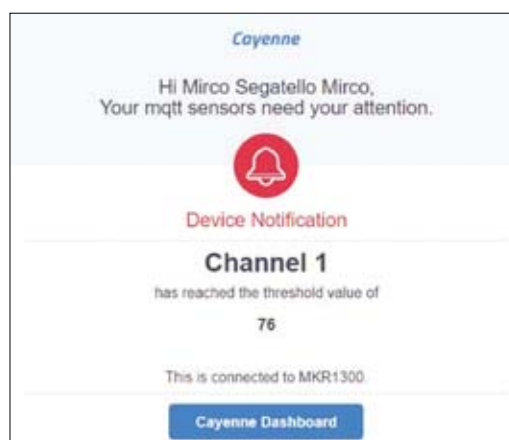


Fig. 15
Ricezione di e-mail
di avviso da parte di
Cayenne.

CONCLUSIONI

Bene, con questo abbiamo terminato la descrizione del progetto del rilevatore livello liquidi per IoT che necessita di qualche attenzione per poter essere portato a termine, e come spesso accade con articoli di questo tipo, lasciamo all'inventiva di ciascuno il compito di personalizzare hardware e software per altre specifiche applicazioni nel campo dell'IoT.



Cosa occorre?

Il materiale presentato in questo articolo è disponibile presso Futura Elettronica. La scheda Arduino MKR WAN 1300 (cod. MKR WAN 1300) è disponibile a Euro 44,00, l'antenna a dipolo (cod. ANTENNADIPOL) costa Euro 6,00, il sensore ad ultrasuoni (cod. SRF05) è in vendita a Euro 25,00, il portabatterie per 2 batterie formato AA (cod. PORTA2AACOVER) è disponibile a Euro 1,00, le batterie Ni-MH formato AA (cod. PAN2700AA) costano Euro 11,00, il pannellino solare 5,5 volt 0,5 watt (cod. SOLPAN05W) è in vendita a Euro 4,50.

Il materiale va richiesto a:

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



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

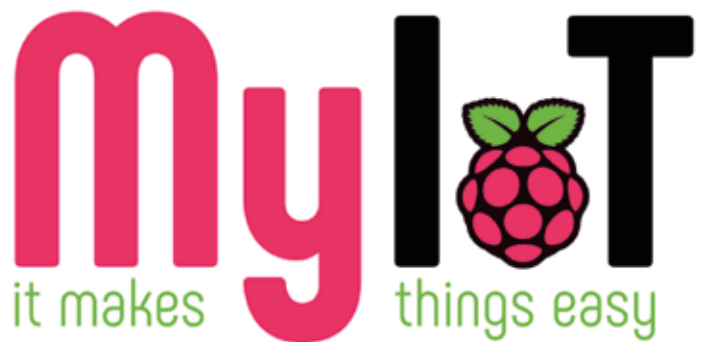
Telecomando universale, programmabile da PC, per controllare qualunque dispositivo di qualsiasi marca e modello. Include tutte le funzioni dell'originale, configurazione facile e veloce. Mai obsoleto grazie ad aggiornamenti gratuiti. Oltre 260.000 modelli in continuo aggiornamento, può comandare fino 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.





di **MARCO MAGAGNIN**

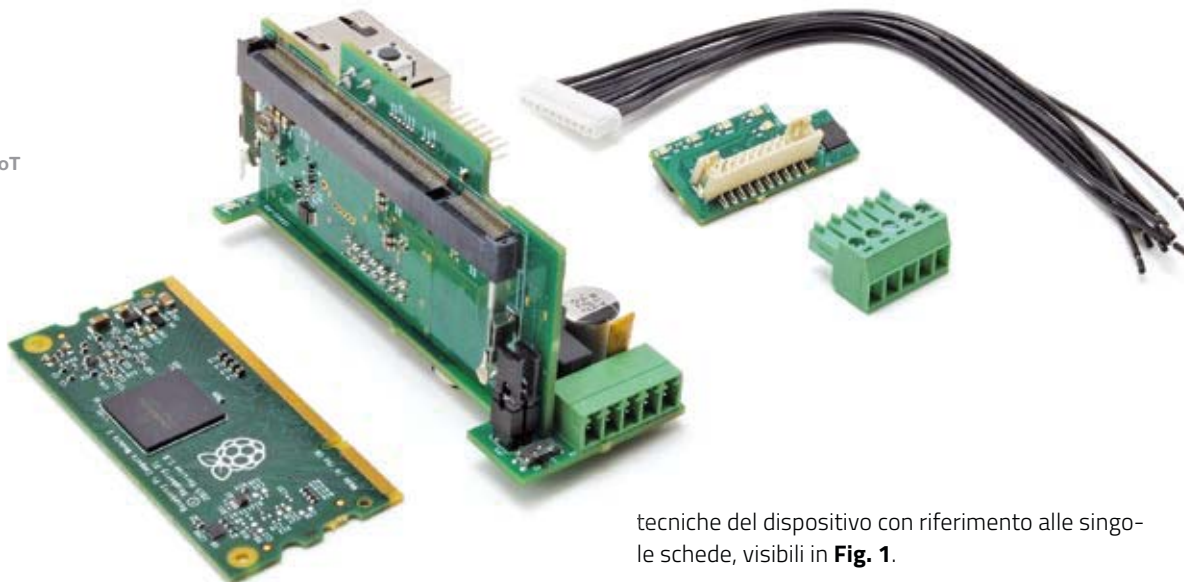
Un "dispositivo" in grado di rendere Raspberry Pi adatto ad essere utilizzato in ambienti "professionali", basato sul "Compute Module 3".



La gamma di microcomputer della famiglia Raspberry Pi rappresenta ormai una realtà consolidata nel mondo delle soluzioni, destinate alla didattica, alla formazione ed anche alle applicazioni professionali ed "industriali". A dicembre 2018 risultavano 23 milioni di "pezzi" venduti. E la storia continua con la messa in vendita da Giugno 2019 del nuovo Raspberry Pi 4 B, un vero e proprio PC. Stesse dimensioni e prezzo invariato, secondo la politica della Fondazione Raspberry Pi. Ma le prestazioni schizzano grazie alla tecnologia a 28nm del processore e alla possibilità di portare la RAM fino a 4 GB. Una nuova conferma della legge di Moore?

In questo articolo vi presentiamo il prototipo del modulo MyIoT prodotto dalla società Italsensor (azienda specializzata in automazione industriale dal 1976), destinato ad applicazioni "professionali" nel campo dell'automazione industriale e della domotica. Il dispositivo è costruito attorno al modulo "Compute Module 3", la versione "professionale" di Raspberry Pi ed è composto da una serie di "basette" che permettono di esporre le principali periferiche disponibili sul CM3, e non solo. Il tutto inserito in un contenitore compatto barra DIN da due moduli. Per esempio è prevista la conversione dell'interfaccia seriale RS232 nell'interfaccia

Fig. 1
Basette che
compongono il
dispositivo MyloT



“industriale” RS485, con la possibilità di inserire un terminatore da 100 Ohm e/o le resistenze di pull-up e pull-down sulle linee A e B dell’interfaccia RS485.

Nella **Fig. 1** è possibile vedere il dispositivo, estratto dal contenitore a barra DIN e scomposto nelle sue parti principali. Per “aprire” il contenitore del dispositivo, agendo con delicatezza ed aiutandosi con un piccolo cacciavite, si rimuove il gancio di blocco per guida DIN dal fondello di chiusura della custodia. Per facilitare l’operazione successiva vi consigliamo di rimuovere il pannellino frontale del dispositivo, facendo leva con delicatezza sulle due piccole fessure a destra ed a sinistra del pannellino, circa a metà altezza. Ora, si deve rimuovere il retro del dispositivo, facendo leva sulla parte in alto per sbloccare i due fermi a scatto. Lo stesso nella parte inferiore. Raccomandiamo sempre la massima delicatezza per evitare danni irreparabili. Una volta tolta la parte posteriore del dispositivo, è possibile estrarre tutta l’elettronica dalla custodia, che è possibile scomporre come visibile in **Fig. 1**. Questa operazione è necessaria per l’inserimento della Micro SD Card e della batteria per il modulo RTC. È però comunque possibile richiedere il modulo già equipaggiato della micro SD Card montata e precaricata con sistema operativo Raspbian “Stretch” e della batteria RTC. Nel caso dobbiate inserire voi la batteria RTC, di tipo CR1025, fate attenzione a orientare il polo negativo della batteria a contatto con il circuito stampato ed il polo positivo in contatto con l’alloggiamento metallico della batteria. Dato che abbiamo disassemblato l’elettronica del dispositivo è agevole elencare le caratteristiche

tecniche del dispositivo con riferimento alle singole schede, visibili in **Fig. 1**.

1. Modulo “Compute Module 3”. È il microcontrollore della famiglia Raspberry Pi, in pratica il “cervello” dell’intero sistema. Nella tabella 1 sono indicati gli utilizzi dei pin GPIO del Compute Module 3.
2. Micro SD Card da 8 GB, classe 10 che contiene il sistema operativo Raspbian “Stretch” ed i programmi preinstallati; La configurazione iniziale prevede:
 - IP statico preconfigurato: 192.168.2.228 (file / etc/dhcpd.conf);
 - SSH abilitato anche per utente root (user: root, password: ital);
 - server web Apache2, PHP7 (v 7.0.33) e database Mariadb (v 10.1.37) già caricati ed operativi;
 - interfaccia seriale ttyAMA0 già configurata per utilizzo diretto dal Compute Module (disattivato console output alla partenza del sistema), l’uscita è sulla porta RS485;
 - utility rpim3extended precaricata e subito utilizzabile per la gestione del modulo;
 - esempi di utilizzo (uscite aggiuntive, gestione watchdog, uso RS485, ...) scritti in Python;
3. Sulla “scheda” principale sono stati implementati:
 - n° 1 porta Ethernet LAN 10/100 conforme allo standard IEEE802.3/802.3u ;
 - n° 2 porte USB2.0;
 - n° 2 porte USB3.0;
 - n° 1 porta micro USB interna per la programmazione della memoria eMMC relativa al modulo CM3 (se disponibile);
 - RTC già configurato ed operativo (batteria CR1025, Li/MnO2, già installata);
 - Connettore a vite - Alimentazione DC 9÷28V

GPIO COMPUTE MODULE	DIREZIONE	FUNZIONE
GPIO2	-	Linea I2C (SDA) utilizzata per modulo RTC
GPIO3	-	Linea I2C (SCL) utilizzata per modulo RTC
GPIO23	-	Riservato: utilizzo futuro
GPIO24	-	Riservato: utilizzo futuro
GPIO14	OUT	TXD – Linea trasmissione seriale
GPIO15	IN	RXD – Linea ricezione seriale
GPIO16	OUT	1: led LD1 ON / 0: led LD1 OFF
GPIO34	OUT	1: led LD2 ON / 0: led LD2 OFF
GPIO25	IN	1: RST attivato / 0: RST a riposo
GPIO18	OUT	1: attiva procedura di reset del Compute Module dopo 60"
GPIO22	OUT	1: attiva funzionalità watchdog hardware
GPIO27	OUT	Segnale di heartbeat per watchdog hardware, richiesto almeno un cambio di stato ogni 60"
GPIO17	IN	Linea forzata ad 1 dal microcontrollore quando, con hardware watchdog attivato (GPIO22), il microcontrollore non rileva almeno un cambio di stato della linea GPIO27 entro un intervallo di tempo di 60", dopo un'ulteriore attesa di 60" viene eseguito il reset del sistema forzando a zero la linea RUN del Compute Module.
GPIO11	OUT	1: attiva uscite linee 1,2,3,4 / 0: linee 1,2,3,4 in 3-state
GPIO21	OUT	1: attiva uscite linee 5,6,7,8 / 0: linee 5,6,7,8 in 3-state

Tabella 1
Utilizzo pin
GPIO Compute
Module 3.

(protezione contro inversione polarità attraverso fusibile ripristinabile);

- Connettore a vite - Interfaccia RS485 half-duplex da 1200 a 115200 baud con gestione automatica della direzione di trasmissione e meccanismo fail-safe;
- Pulsante di "Reset" del sistema;

4. Schedina di supporto del connettore:

- n°8 porte di uscita con driver push-pull non optoisolati, funzione 3-state attivabile su due gruppi di 4 linee, protezione contro sovra-temperatura, corrente di uscita massima per canale 50 mA, frequenza massima in uscita 50 kHz. Alimentazione interfaccia a +5V, lo 0V è in comune con lo 0V dell'alimentazione del modulo e deve essere necessariamente collegato esternamente al modulo stesso. **ATTENZIONE:** attualmente non è implementata la protezione contro inversione di polarità sull'alimentazione (nelle future versioni sarà disponibile un'interfaccia push-pull con le stesse caratteristiche di base, ma opto isolata e con alimentazione estesa a 5÷30V).

In **Fig. 2** è visibile l'elettronica assemblata, prima di essere inserita nella custodia a barra DIN.

In **Fig. 3** è visibile l'alloggiamento della Micro SD Card del tipo a blocco meccanico. Per inserirla è necessario spingerla il più possibile all'interno del connettore, un breve ritorno conferma il blocco della micro SD Card. Per estrarla è necessario spingere nuovamente la Micro SD Card verso l'interno dell'alloggiamento in modo da sbloccare il meccanismo di ritenzione.

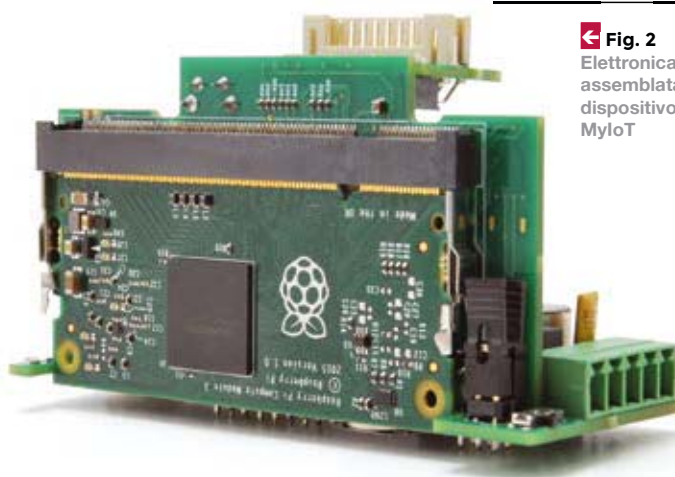


Fig. 2
Elettronica
assemblata
dispositivo
MyIoT

In **Fig. 4** è visibile il dettaglio dell'alloggiamento della batteria per il clock RTC.

In **Fig. 5** è visibile il dettaglio del connettore di alimentazione ed interfaccia RS485. Il connettore DIS è il riferimento isolato della linea RS485, non è collegato alla massa (0V) del dispositivo. L'apertura (foro) a sinistra del connettore permette di accedere ad un interruttore, sulla scheda principale, che consente di inserire, se spostato verso il connettore di alimentazione, un terminatore da 100 ohm tra le linee A e B dell'interfaccia RS485. All'interno del modulo, attraverso i due jumper, visibili in **Fig. 6**, è possibile inserire o disinserire i resistori di pull-up e pull-down da 620 ohm sulle linee A e B dell'interfaccia RS485. In questo modo è possibile gestire tutte le possibili configurazioni di un collegamento RS485.

In **Fig. 7** è visibile il frontalino del dispositivo. Di seguito il significato dei singoli elementi:

TX: led verde (trasmissione dati su RS485);

RX: led rosso (ricezione dati su RS485);

ON: led blu (attività Compute Module);

L1: led giallo (utilizzo utente);

L2: led giallo (utilizzo utente);

RST: accesso per pulsante RST, l'impiego di questo pulsante è descritto in seguito;

CN2: connettore espansione n°8 uscite aggiuntive (1...8). Nella Tabella 2 sono indicate le corrispondenze tra i pin del connettore CN2 ed i pin del GPIO del Compute Module 3; gli ultimi due sono connettori di alimentazione 0 e +5 V.

Se desiderate utilizzare il connettore CN2 per le linee di uscita aggiuntive è bene eseguire questa operazione con l'elettronica estratta dalla custodia, in quanto, come visibile in **Fig. 2**, che mostra l'elettronica assemblata, la schedina con il connettore per le uscite aggiuntive è montata "a sbalzo" rispetto alla scheda principale. Inserire il connettore del cavo piatto rischierebbe di rovinare i contatti della schedina stessa. È necessario sfilare, o sostenere la schedina durante l'inserimento del connettore del cavo piatto. La linea dello 0V sul connettore CN2 è in comune con l'alimentazione del modulo, i driver di uscita non separano galvanicamente i GPIO del Compute Module dalle linee di uscita, ma realizzano un'interfaccia in grado di lavorare a 5V, protetta termicamente, con una corrente massima di 50 mA per canale e

Fig. 3
Alloggiamento
micro SD Card.

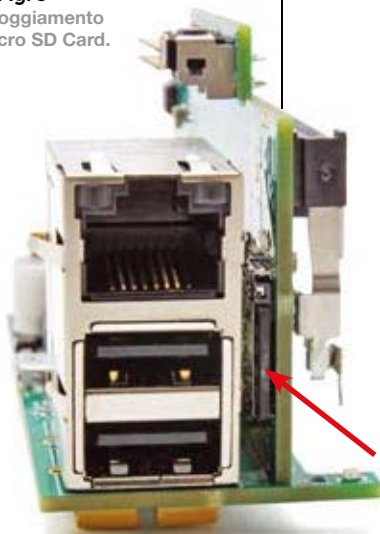


Fig. 4
Alloggiamento batteria
clock RTC.

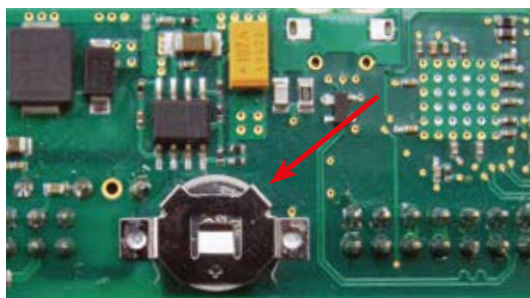
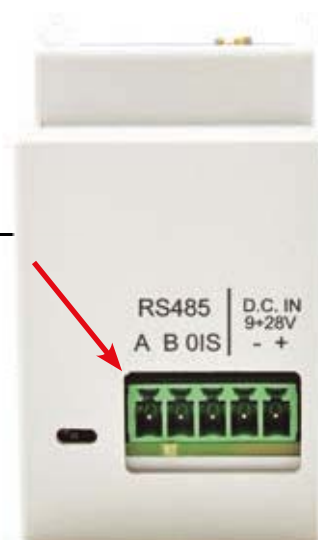


Fig. 5
Connettore
alimentazione ed
interfaccia RS485.



PIN CN2	GPIO COMPUTE MODULE	DIREZIONE	FUNZIONE
1	GPIO36	OUT	Impostazione stato uscita linea 1
2	GPIO20	OUT	Impostazione stato uscita linea 2
3	GPIO37	OUT	Impostazione stato uscita linea 3
4	GPIO35	OUT	Impostazione stato uscita linea 4
5	GPIO38	OUT	Impostazione stato uscita linea 5
6	GPIO13	OUT	Impostazione stato uscita linea 6
7	GPIO12	OUT	Impostazione stato uscita linea 7
8	GPIO26	OUT	Impostazione stato uscita linea 8
9	-	-	Linea alimentazione uscita espansione: 0V
10	-	-	Linea alimentazione uscite espansione: +Vin=5V

Tabella 2
Corrispondenza
pin connettore
CN2 e pin
GPIO Compute
Module 3.

frequenza di commutazione fino a 50 kHz.
Terminata la descrizione "statica" del dispositivo MyIoT possiamo riassembalarlo.

Per rimontare il modulo si eseguono le seguenti operazioni:

- Si introduce il modulo nella custodia facendo prima passare (se utilizzati) i cavetti della scheda di espansione attraverso l'apertura presente in cima alla custodia, in seguito si inserisce e si blocca il pannello serigrafato di chiusura.
- Si inserisce il fondello del modulo chiudendo così la custodia.
- Si inserisce nel fondello il gancio per l'innesto sulla guida DIN.

A questo punto possiamo procedere alla prima accensione del dispositivo MyIoT; per l'alimentazione è necessario utilizzare un alimentatore in grado di erogare tra i 9 ed i 26 V. I terminali positivo e negativo vanno collegati ai rispettivi morsetti a vite del connettore verde nella parte superiore del dispositivo.

Collegate il cavo di rete della vostra rete locale all'apposito connettore sul dispositivo MyIoT, alimentate il dispositivo collegando un alimentatore in grado di erogare tra i 9 ed i 28 V in corrente continua ed almeno 2A di corrente. Inizialmente il led blu (ON) presente sulla scheda di uscita inizierà a lampeggiare (lampeggia anche nel caso in cui il modulo venga alimentato senza che sia inserita la scheda Compute Module). Se dopo alcuni secondi il led blu (ON) smetterà di lampeggiare, significa che il processo di boot è andato a buon fine, la scheda Compute Module può quindi essere raggiunta attraverso connessione LAN (ad esempio utilizzando Putty per la connessione SSH ed accedere alla finestra di terminale del Compute Module oppure WinSCP per gestire il trasferimento di file da e verso il dispositivo). Per approfondire le modalità di connessione e di utilizzo dei microcomputer Raspberry Pi, e quindi del Compute Module, vi rimandiamo alle informazioni contenute nel libro "Entra nel mondo di

Fig. 6
Impostazione jumper
interfaccia RS485

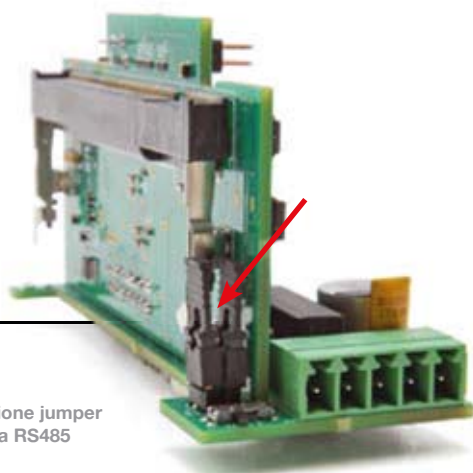


Fig. 7
Frontalino
dispositivo MyIoT.



Fig. 8
Configurazione
indirizzo IP.

```

pi@raspberrypi: ~
GNU nano 2.7.4 File: /etc/dhcpd.conf

#interface eth0
#static ip_address=192.168.0.10/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
#static routers=192.168.0.1
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1

interface eth0
static ip_address=192.168.2.228/24
static routers=192.168.0.1
static domain_name_servers=192.168.2.1
static domain_search=

^G Get Help ^O Write Out ^W Where Is ^R Cut Text ^J Justify ^C Cur Pos
^X Exit ^B Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

```

Raspberry Pi 3+” ed ai numerosi articoli apparsi sulla rivista.

Per poter accedere al sistema operativo è necessario conoscere le credenziali di accesso. Per il dispositivo MyIoT sono disponibili due utenti predefiniti: l’utente generico “pi” con password “raspberrypi”, che è il classico utente predefinito per il sistema operativo Raspbian e l’utente di amministrazione “root” con password “ital”, predisposto dalla società costruttrice per le operazioni di amministrazione. Dato che l’indirizzo IP predefinito per il dispositivo MyIoT può non combaciare con la sottoclasse di rete della vostra rete locale, potete assegnare al PC che utilizzate per il collegamento SSH un indirizzo IP fisso compatibile con la sottoclasse di rete del dispositivo MyIoT. Con questo artificio potete accedere temporaneamente al dispositivo ed eseguire le prime prove e le eventuali configurazioni che ritenete opportune, per esempio modificare l’indirizzo IP. Dopo esservi collegati con Putty in SSH al dispositivo MyIoT, per modificare l’indirizzo IP potete modificare il file di configurazione /etc/dhcpd.conf. Suona tutto strano, vi raccomandiamo di approfondire sul libro

citato in precedenza. Digitate il comando:

```
sudo nano /etc/dhcpd.conf
```

Andate alla sezione visibile di seguito (**Fig. 8**) e modificate la configurazione secondo le vostre esigenze. Salvate con <CTRL-X>, e confermate con “Y”.

```

interface eth0
static ip_address=192.168.2.228/24
static routers=192.168.0.1
static domain_name_servers=192.168.2.1
static domain_search=

```

Sulla micro SD Card fornita con il dispositivo sono presenti alcuni programmi utili per verificare le funzionalità del dispositivo. Si trovano nella cartella:

/home/test

In questa cartella sono presenti degli esempi scritti in Python utilizzabili come base di partenza per la scrittura delle proprie applicazioni.

Tabella 3
Elenco programmi di
esempio.

PROGRAMMA (/home /test)	UTILIZZO
testout.py	Demo utilizzo uscite interfaccia espansione e led
tstGPIO18.py	Portando GPIO18 ad 1 attiva la procedura di reset dopo 60s
tstGPIO25.py	Lettura stato linea GPIO25 (stato del pulsante RST)
tstHWWDT.py	Esempio di utilizzo del watchdog hardware
tstserial_ma.py	Test comunicazione seriale su RS485 (master)
tstserial_sl.py	Test comunicazione seriale su RS485 (slave)



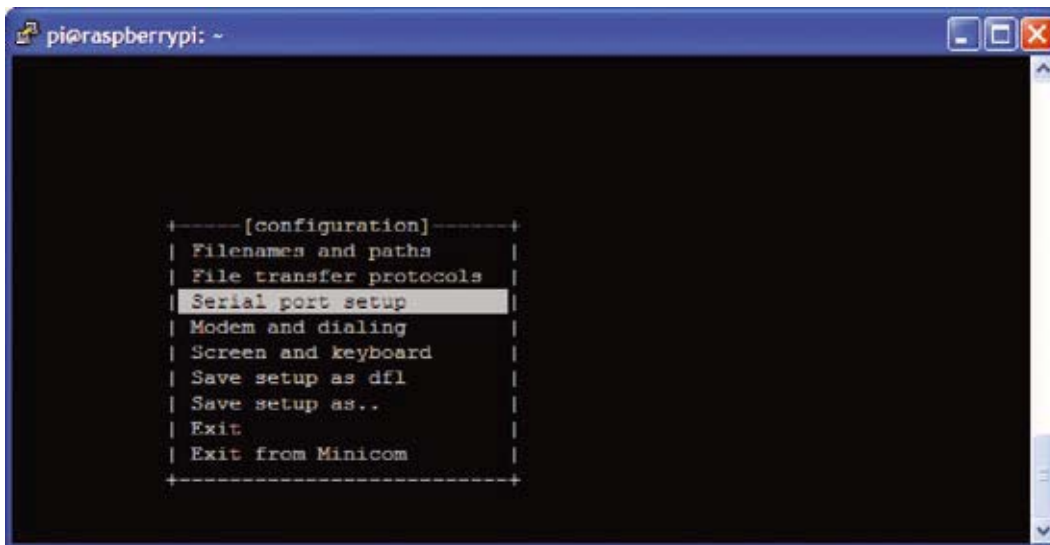


Fig. 9
Menu minicom.

Tutte le funzionalità del modulo possono essere gestite attraverso un'apposita applicazione richiamabile da linea di comando. Per visualizzare la guida in linea digitate:

```
rpim3extended
```

Per la gestione delle uscite sul modulo di espansione si hanno le seguenti opzioni:

- led: accende (on) o spegne (off) il led L1
- led1: accende (on) o spegne (off) il led L2
- out<n>: imposta a "1" (5V) o a "0" (0V) la linea n (n=1...8) di uscita
- outall: imposta a "1" (5V) o a "0" (0V) tutte le linee di uscita
- outen<n>: abilita (on) disabilita (off) il primo o il secondo gruppo delle 4 linee in uscita
- outenall: abilita (on) oppure disabilita (off) tutte le uscite ponendole in alta impedenza (3-state)

Nella **tabella 3** è presente la lista dei programmi di esempio disponibili ed il loro utilizzo: Ogni programma di esempio può essere interrotto durante l'esecuzione con la combinazione di tasti ^C (CTRL+C). Maggiori dettagli nei commenti contenuti nei file.

Il pin GPIO25, come visibile analizzando la **tabella 1**, viene attivato premendo il pulsante RST. Più precisamente, con il pulsante premuto il pin GPIO25 assume il livello alto ("1") mentre con il pulsante rilasciato il pin GPIO25 rimane al livello basso ("0"). Al pulsante RST sono associate anche altre funzionalità. Potete verificare questo comportamento eseguendo il programma "tstGPIO25.py, digitando il comando:

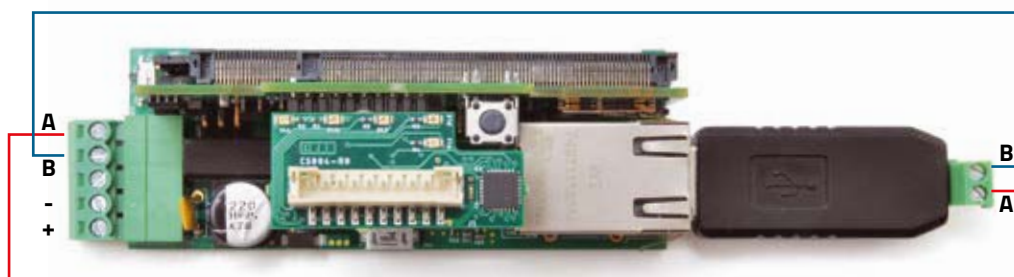
```
cd /home/test
python tstGPIO25.py
```

Tenete conto dei tempi di latenza nella risposta e non tenetelo premuto per un tempo superiore ai quattro secondi, in quanto, se RST è premuto per un tempo superiore a quattro secondi il led blu (ON) inizia una sequenza di due lampeggi, se dopo questo periodo di tempo il pulsante viene rilasciato il microcontrollore attende 60 secondi prima di iniziare la sequenza di reset del Compute Module. Se il pulsante RST continua a rimanere premuto per almeno altri 10s il led blu lampeggia velocemente ed il rilascio del pulsante provocherà il reset immediato del Compute Module.

Per quanto riguarda la comunicazione seriale RS485 il baud rate è compreso tra 1200 e 115200 baud. Interfaccia isolata e protetta con meccanismo di fail-safe, la direzione di trasmissione è gestita automaticamente dal microcontrollore presente nel modulo. Sono disponibili due programmi di esempio tstserial_ma.py e tstserial_sl.py come da elenco in tabella 3. La disponibilità di una comunicazione in modalità RS485 permette di integrare un dispositivo con questa architettura, fatte salve alcune considerazioni che faremo alla fine, in un sistema di comunicazione industriale, per esempio come elemento di integrazione tra il sottosistema di produzione, basati su protocolli industriali, come appunto l'RS485 ed i sottosistemi di controllo e gestione, che utilizzano applicazioni orientate agli utenti e protocolli di comunicazione TCP/IP.

Per provare la comunicazione RS485 tra un modulo master ed almeno uno slave, potevamo

➔ **Fig. 10**
Schema
collegamento
RS485 tra modulo
USB-RS485 e
MyIoT



utilizzare due dispositivi MyIoT, configurati uno come master ed uno come slave. Abbiamo a disposizione un solo prototipo e di conseguenza l'unica possibilità di eseguire il test in modo semplice era di rovistare nel solito cassetto "magico" dove abbiamo trovato un adattatore convertitore USB-RS485. Oltre che nel cassetto lo trovate a catalogo presso Futura Elettronica, con il codice 1606-CONVUSBR5485. Inserito in una porta USB del dispositivo MyIoT, verrà riconosciuto in modo automatico ed apparirà in /dev come ttyUSB0 (almeno nel nostro caso).

Lo schema di collegamento tra modulo USB-RS485 e MyIoT è visibile in **Fig. 10**.

Siccome il modulo USB-RS485 viene visto come porta seriale virtuale, per utilizzarlo è sufficiente utilizzare un programma di emulazione seriale, come "minicom". Con piacevole sorpresa l'abbiamo trovato già installato e quindi lo possiamo configurare eseguendolo in modalità "setup" con il comando (**Fig. 10**):

```
sudo minicom -s
```

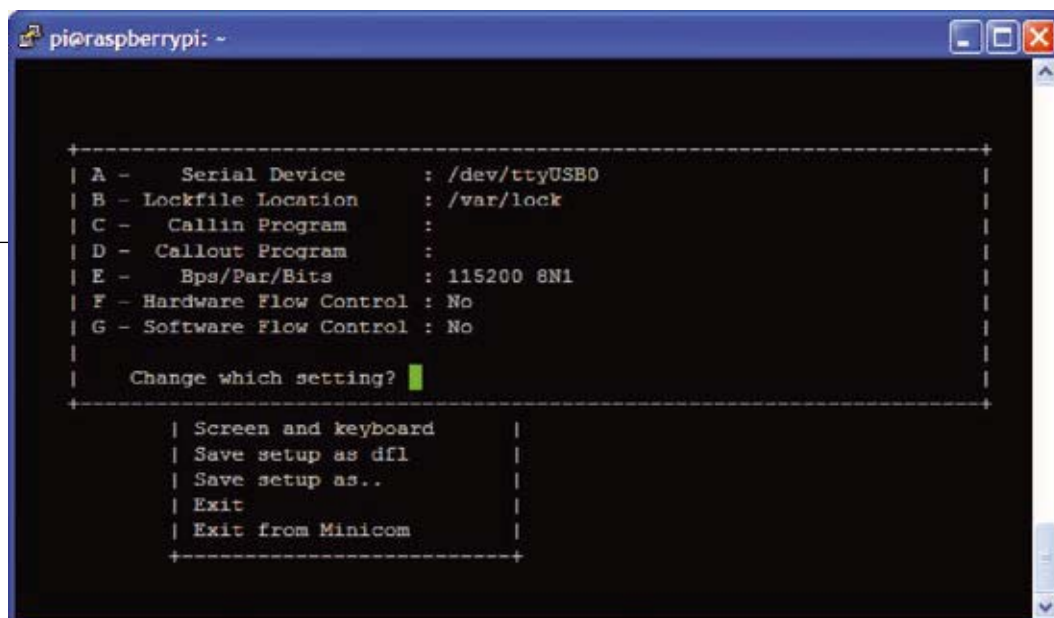
Selezionate la voce di menu "Serial port Setup" e configurate la porta come visibile in **Fig. 11**.

Con <INVIO> tornate al menu principale dove selezionate "Exit" per entrare nella finestra di emulazione seriale. Prima che restiate intrappolati, per uscire dalla finestra di emulazione seriale digitate

<CTRL-A> e poi <X> e "Yes".

Per eseguire il codice dimostrativo della porta RS485 prima di tutto dovete collegare con due cavi la porta "A" del dispositivo MyIoT con la porta "A" del convertitore USB. Lo stesso con le porte B. Ora dovete aprire una seconda finestra di terminale, con PuTTY, dove far eseguire i programmi di test della comunicazione RS485. Per entrambi i programmi di esempio è disponibile un help di utilizzo richiamabile invocando lo script senza argomenti. Per avere una trasmissione/ricezione

➔ **Fig. 11**
Configurazione
emulazione porta
seriale.



```
pi@raspberrypi: /home/test
(0) TX (val): 65(char): A
(1) TX (val): 65(char): A
(2) TX (val): 65(char): A
(2) RX (val): 32(char): SPACE
(2) RX (val): 66(char): B
(3) TX (val): 65(char): A
(3) RX (val): 66(char): B
(4) TX (val): 65(char): A
(4) RX (val): 118(char): v
(5) TX (val): 65(char): A
(6) TX (val): 65(char): A
```

Fig. 12
Output
programma
tstserial_ma.py
(RS485 master).

```
pi@raspberrypi: ~
Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Apr 22 2017, 09:14:19.
Port /dev/ttyUSB0, 11:12:41

Press CTRL-A Z for help on special keys

AAAAAA
```

Fig. 13
Output
emulatore porta
seriale (RS485
slave).

coerente si deve impostare lo stesso valore di baud rate per entrambi gli script. I programmi di esempio relativi alla trasmissione/ ricezione seriale RS485 sono stati scritti pensando ad una gestione dei dati di tipo binario, non solo quindi per la trasmissione di dati relativi ai codici ASCII dei caratteri visualizzabili.

Esempio 1: Il dispositivo MyIoT funge da master ed il convertitore USB da Slave. Il master trasmette in continuo un singolo carattere con una pausa di cinque secondi tra due invii successivi. Il modulo slave può a sua volta inviare caratteri. Nella prima finestra di terminale deve essere in esecuzione il programma “minicom” mentre nella seconda devono essere digitati i comandi:

```
cd /home/test
python tstserial_ma.py 115200 65 65 5
```

Ogni tanto digitate un carattere nella finestra con l'emulazione di comunicazione seriale. Nelle **Fig. 12 e 13** vedete il risultato della comunicazione bidirezionale.

Esempio 2: In questo secondo esempio il dispositivo MyIoT funge da Slave ed il convertitore USB da Master. Lo Slave attende l'invio di caratteri da parte del Master e li ritrasmette in una sorta di Loop Pack. Nella prima finestra di terminale deve essere in esecuzione il programma “minicom” mentre nella seconda devono essere digitati i comandi:

```
pi@raspberrypi: ~
Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Apr 22 2017, 09:14:19.
Port /dev/ttyUSB0, 10:05:28

Press CTRL-A Z for help on special keys

A B C D
```

Fig. 14
Output
emulatore porta
seriale (RS485
master).

Fig. 15
Output programma
tstserial_sl.py
(RS485 slave).

```
pi@raspberrypi: /home/test
Waiting for incoming messages...
(0) RX (val):65(char): A
(1) TX (val):65(char): A
(1) RX (val):32(char): SPACE
(2) TX (val):32(char): SPACE
(2) RX (val):66(char): B
(3) TX (val):66(char): B
(3) RX (val):32(char): SPACE
(4) TX (val):32(char): SPACE
(4) RX (val):67(char): C
(5) TX (val):67(char): C
(5) RX (val):32(char): SPACE
(6) TX (val):32(char): SPACE
(6) RX (val):68(char): D
(7) TX (val):68(char): D
```

```
cd /home/test
python tstserial_sl.py 115200
```

Digitate una serie di caratteri nella finestra con l'emulazione di comunicazione seriale e li vedrete riapparire ritrasmessi nella stessa finestra. Nelle **Fig. 14 e 15** vedete il risultato della comunicazione bidirezionale.

Per mantenere aggiornate la data e l'ora di sistema anche nei periodi di disconnessione dalla rete e/o mentre il dispositivo MyIoT è spento, nel circuito è presente un modulo RTC basato sull'integrato MCP79410.

RTC (real Time Clock) è un orologio hardware separato dall'ora di sistema e che permette di mantenere le informazioni di data ed ora anche a modulo non alimentato. Si interfaccia al Compute Module attraverso le linee I2C del modulo stesso. Se volete verificare che il Compute Module "veda" correttamente l'orologio RTC sul bus I2C provate a digitare il comando:

```
i2cdetect -y 1
```

Otterrete il risultato visibile in **Fig. 16**. All'indirizzo 57 dovreste vedere i caratteri "57" che corrispondono all'indirizzo I2C dell'orologio RTC. Verificato che l'orologio RTC è funzionante, data ed ora verranno aggiornati automaticamente e potrete verificarlo con i comandi standard di gestione di data ed ora in ambiente GNU/Linux. Per verificare il corretto funzionamento della funzione RTC eseguite il processo descritto di seguito, dopo aver verificato che il Compute Module sia connesso alla rete internet. Prima di tutto controllate la data e l'ora impostata nel sistema con il comando:

```
date
```

Il comando "date" riporta la data e l'ora di sistema, che "dovrebbe" essere stata aggiornata automaticamente in fase di avvio del sistema operativo, per mezzo di un accesso ad uno dei server internet che forniscono il servizio "ntp" (network time protocol). In **Fig.17** è possibile vedere un possibile risultato.

Ora verificate data ed ora memorizzate nell'orologio RTC del dispositivo MyIoT con il comando:

```
sudo hwclock -r
```

In **Fig.18** è possibile vedere un possibile risultato. Se la data ed ora restituite non sono corrette, o è stato restituito un messaggio di errore, aggiornate l'orologio RTC con l'ora di sistema digitando il comando:

```
sudo hwclock -w
```

Il parametro "-w" comanda l'operazione di scrittura di data ed ora nell'orologio RTC. Ora rieseguite il test precedente e verificate se i due orologi sono sincronizzati.

Sempre dal punto di vista delle funzionalità installate in modo predefinito, sono presenti il server web Apache 2 ed il database MariaDB, utilizzando i quali possono essere sviluppate le proprie applicazioni. Nella configurazione predefinita è presente un esempio di interfacciamento tra il mondo web e l'hardware del modulo, esempio facilmente estendibile al mondo esterno al modulo attraverso la porta RS485 in esso integrata. Nella cartella /var/www/html/tornado è disponibile un esempio di webserver basato su



```

pi@raspberrypi: ~
pi@raspberrypi:~$ i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  57  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
pi@raspberrypi:~$

```

Fig. 16
Dispositivi
presenti sul bus
I2C.

```

pi@raspberrypi: ~
pi@raspberrypi:~$ date
Tue  2 Jul 09:45:51 CEST 2019
pi@raspberrypi:~$

```

Fig. 17
Risultato
comando date.

```

pi@raspberrypi: ~
pi@raspberrypi:~$ sudo hwclock -r
2019-07-02 09:47:52.934306+0200
pi@raspberrypi:~$

```

Fig. 18
Lettura data
e ora orologio
RTC.

websocket, il codice del server è scritto in python. Nella **Tabella 4** sono elencati i file disponibili per l'applicazione.

L'esempio permette di realizzare una comunicazione bidirezionale persistente tra il browser (lato utente) ed il sistema MyIoT implementando una connessione basata su websocket gestita attraverso il web server Tornado nel file *py3torex1.py*, il codice contenuto in questo file, opera da "ponte" tra la parte web e l'hardware del MyIoT, e dovrà quindi essere mandato in esecuzione prima di accedere alla pagina web, diversamente non sarà possibile connettersi al MyIoT dalla pagina web ed inviare e ricevere messaggi attraverso la

pagina stessa. Di seguito si riportano il comando da utilizzare per mandare in esecuzione il web server Tornado:

```
sudo python /var/www/html/tornado/py3torex1.py
```

Alla partenza sarà visibile l'indirizzo IP del modulo che, se fornito con microSDHC oppure eMMC pre-programmata, risulta fissato a 192.168.2.228. Nel nostro caso, visibile in **Fig. 19**, è visibile un indirizzo diverso perchè abbiamo adattato il modulo MyIoT alla nostra rete locale. Sempre in **Fig. 19** è visibile la sequenza dei messaggi di log generati dalla comunicazione.

FILE (/var/www/html/tornado)	UTILIZZO
jquery.min	File di supporto per la pagina web
pysocket.php	Pagina web con javascript per comunicazione su websocket
py3torex1.py	File python web server tornado per gestione websocket lato Raspberry

Tabella 4
Cartella
programmi per
server Web.

Nella pagina web è stata implementata una finestra di Log in cui sono visualizzati i messaggi ricevuti dal modulo MyIoT, una casella di testo in cui scrivere i messaggi che si desiderano trasmettere al MyIoT (con QUIT si ferma l'applicativo che implementa il server web tornado, in modo pulito, senza essere costretti a forzare l'uscita con CTRL+C nella finestra della shell) e due pulsanti che hanno il compito rispettivamente di accendere e spegnere il led LD1 presente sulla scheda di uscita del modulo MyIoT.

Per accedere alla pagina web sarà sufficiente aprire un browser che supporti la comunicazione attraverso websocket (Mozilla oppure Chrome tanto per citare i più noti) e digitare nella barra indirizzi:

`http://<indirizzo.IP-MyIoT>/tornado/pysocket.php`

alla pressione del tasto Invio/Enter si otterrà in risposta la pagina web trasmessa dal modulo MyIoT al browser attraverso il web server Apache2, visibile in **Fig. 20**. La comunicazione vera e propria attraverso il protocollo websocket sarà attivata dal codice javascript presente nel file `pysocket.php` (che utilizza la libreria jquery presente nella cartella dell'esempio) il quale si occupa di stabilire e gestire la connessione persistente con il server Tornado mandato in esecuzione attraverso lo script python lanciato in precedenza.

L'esempio riportato, nonostante la semplicità, permette da subito di instaurare una comunicazione bidirezionale tra la pagina web ed il modulo MyIoT, comunicazione che può prevedere la trasmissione di messaggi testuali che possono essere visti come comandi trasmessi dall'utente al modulo MyIoT ai quali è possibile associare determinate operazioni (in questo caso è gestito solo il messaggio QUIT) oppure la gestione di eventi associati alla pressione di pulsanti sulla pagina web mediante il codice javascript presente nel file `pysocket.php`. La comunicazione implementata si basa solo su caratteri ASCII, ma nessuno vieta la trasmissione di strutture dati più complesse attraverso ad esempio il protocollo JSON. Questo potrà essere oggetto di futuri articoli su possibili applicazioni pratiche di questo versatile modulo.

Nel caso si utilizzi un Compute Module 3 dotato di memoria eMMC è possibile installare il sistema operativo Raspbian sulla eMMC stessa. Per la programmazione della eMMC si utilizza l'utility `rpiboot.exe` reperibile sul sito web della Raspberry

Pi Foundation.

Una volta scaricato questo tool (disponibile per sistema operativo Windows e Linux) si collega il modulo MyIoT ad una porta libera del PC attraverso la porta microUSB interna al modulo, si alimenta il modulo ed in seguito si esegue l'utility `rpiboot`, dopo alcuni istanti il sistema configura il Compute Module affinché possa essere visto dal sistema come disco esterno. A questo punto utilizzando un software di masterizzazione per SD come Etcher oppure Win32DiskImager si esegue la programmazione della memoria. Completata la programmazione si rimuove il cavo di programmazione e l'alimentazione del modulo. Riapplicando la tensione di alimentazione si avrà il boot del sistema dalla eMMC programmata.

Per ora ci fermiamo qui e ci permettiamo un paio di considerazioni, che valgono per il prototipo che abbiamo avuto a disposizione per la nostra recensione. La prima riguarda l'impiego della micro SD Card. Per un dispositivo di utilizzo professionale questo rappresenta un punto di possibile debolezza data la vita "limitata", in termini di scritture, propria delle SD Card. Per un utilizzo in campo fotografico, questo non costituisce un problema, ma se l'utilizzo prevede numerose riscritture in tempi ravvicinati nelle stesse locazioni della memoria della micro SD Card, come nel caso di scrittura di dati di log o di utilizzo con un sistema di gestione database, allora la vita di questa si riduce notevolmente. In questi casi è meglio scegliere di memorizzare il sistema operativo ed i dati su una memoria eMMC (embedded Multi Media Card). Nella famiglia Compute Module sono disponibili soluzioni con eMMC di diverse capacità. Una seconda considerazione è legata alla continuità di alimentazione. Vi consigliamo



Cosa occorre?

Il controller industriale basato su Raspberry Pi CM3 (cod. MyIoT-RPICM3) è in vendita a Euro 249,00 (il modulo CM3 non è compreso). 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>**



```
pi@raspberrypi: ~  
pi@raspberrypi:~$ sudo python /var/www/html/tornado/py3torex1.py  
Websocket Server Started at: 192.168.178.55  
New connection was opened from 192.168.178.10  
Path: /ws  
Incoming message: LEDON  
Incoming message: LEDOFF  
New connection was opened from 192.168.178.10  
Path: /ws  
Connection was closed...
```

Fig. 19
Log server
Tornado

di prevedere un sistema di UPS a batteria per evitare il crash del sistema in caso di caduta di tensione, con possibile corruzione della memoria di massa. In questo caso è opportuno utilizzare il circuito di watchdog hardware, opportunamente previsto nel dispositivo MyIoT Ultima considerazione riguarda la gestione del calore. Forse anche perché stiamo scrivendo questo articolo con quasi quaranta gradi di temperatura, abbiamo notato che il Compute Module 3 scalda parecchio. Alcuni

fori di aereazione nel case, un dissipatore sul SoC ed un programma di controllo della temperatura in grado di avvisare in caso di condizioni critiche ed eventualmente di spegnere il dispositivo potrebbe essere utile. Anche in questo caso il circuito di watchdog hardware costituisce un aiuto prezioso. Per il momento, buona sperimentazione. Da parte nostra seguiremo con attenzione le future evoluzioni di questi prodotti e relative soluzioni. ■

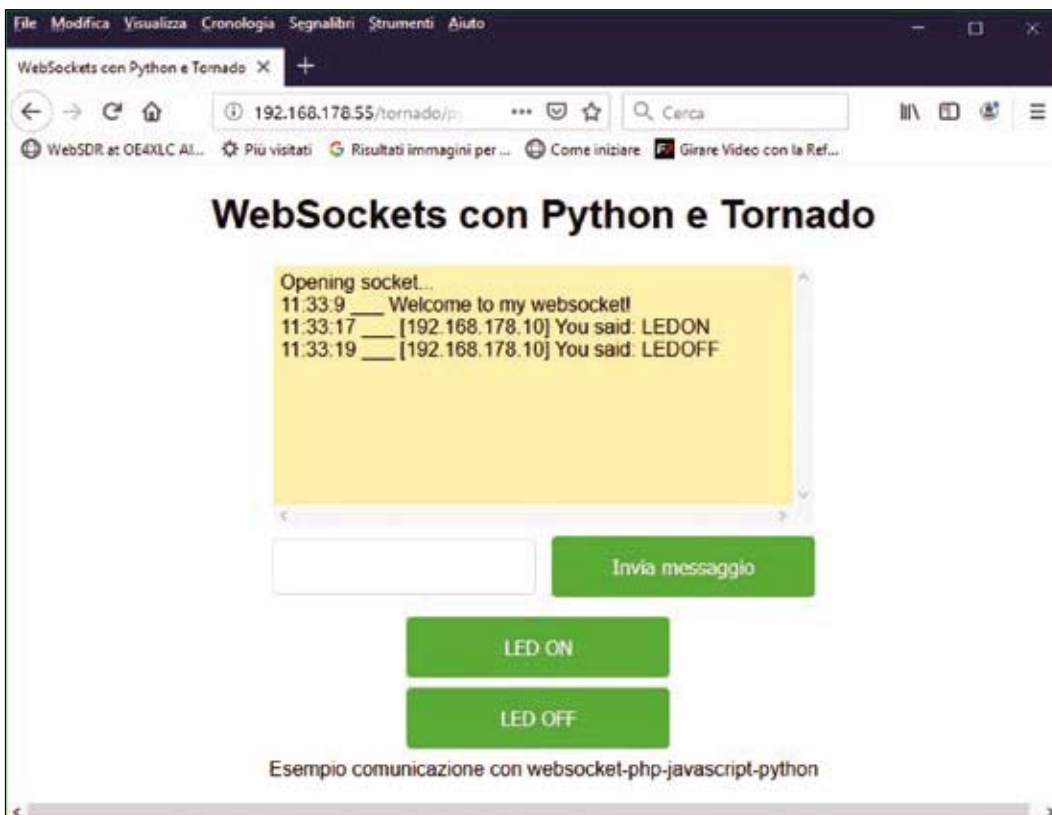


Fig. 20
Pagina web di
esempio per
controllo MyIoT

GESTISCI E CONTROLLA LA TUA CASA CON I TELECONTROLLI GSM

ATTIVA E CONTROLLA DA REMOTO QUALSIASI DISPOSITIVO
ELETTRICO O ELETTRONICO: DALLA CENTRALINA ANTIFURTO
ALL'IMPIANTO DI IRRIGAZIONE, DALLE LUCI DI CASA
AL SISTEMA DI CONDIZIONAMENTO E RAFFRESCAMENTO.
GESTIONE TRAMITE CHIAMATA VOCALE O MESSAGGI SMS.

cod. TDG133

€ 79,00

Telecontrollo GSM bidirezionale 2 IN / 2 OUT

- 2 uscite a relè con funzionamento monostabile o bistabile e 2 ingressi d'allarme optoisolati.
- Dimensioni 103 x 67 x 28 mm
- Alimentazione 9÷32 Vdc/1 A
- Completo di antenna GSM esterna



**Disponibili
anche moduli
con funzione di:**

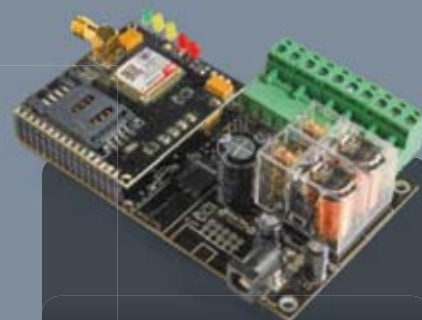
APRICANCELLO

cod. TDG134 • € 75,00

GSM

TERMOSTATO CON CONTROLLO GSM

cod. TDG139 • € 79,00



cod. TDG140 • € 88,00

TELECONTROLLO GSM CON COMANDI DTMF

Tutti i telecontrolli vengono forniti già montati e collaudati e sono facilmente configurabili mediante SMS o in locale tramite PC (per il collegamento al PC è necessario il modulo opzionale d'interfaccia USB cod. FT782M • € 13,00).



cod. TDG138 • € 79,00

TELEALLARME GSM CON ANTI JAMMER

Contenitore in ABS appositamente forato per ospitare i telecontrolli GSM: TDG133, TDG138 e TDG140. Cod. BOXTDG • € 16,90

Disponibile anche per TDG134. Cod. BOXTDG134 • € 16,90

Disponibile anche per TDG139. Cod. BOXTDG139 • € 16,90



Prezzi IVA inclusa.

**FUTURA
ELETTRONICA®**

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

Caratteristiche tecniche
e vendita on-line su:

www.futurashop.it

Il mondo dell'



dell'Ing. MIRCO SEGATELLO



Sperimentiamo l'IOT con l'impiego della rete a basso consumo SigFox, alternativa alle reti wifi e GSM. Quarta e ultima puntata.

Durante questo corso, dopo aver spiegato in cosa consiste l'IoT e quali siano i suoi utilizzi, ci siamo occupati di alcune implementazioni facenti uso di schede Arduino MKR appositamente progettate per lo scopo. Nella seconda lezione abbiamo implementato un dispositivo IOT su rete WiFi, mentre nella terza lezione è stata usata una scheda MRX 1300 su rete LoraWAN. Quando si parla di IoT sembra imprescindibile parlare di reti LPWAN (Low Power Wide Area Network) ovvero di quella tecnologia che permette trasmissioni utilizzando moduli radio a bassissima potenza ma con lungo raggio di copertura, anche se a discapito

della quantità e velocità dei dati trasmessi. Questo è valido in quanto a livello europeo la banda di frequenze comprese tra 865MHz e 870MHz è di libero utilizzo a patto di non utilizzare trasmettitori con potenza superiore ai 25mW e con un tempo di utilizzo inferiore all'1% ovvero 36 secondi ogni ora. In questa ultima lezione ci soffermiamo sulla soluzione fornita dalla compagnia francese SigFox che dispone di una rete di telecomunicazioni specificatamente progettata per l'IOT, attualmente presente in Europa occidentale, San Francisco, ed in fase test in Sud America e Asia. La rete SigFox utilizza la tecnologia LPWAN ed è stata progettata per l'invio

di messaggi di ridotte dimensioni in cui la comunicazione viene sempre avviata dal dispositivo finale, il quale non dovrà rimanere costantemente attivo in attesa di una ricezione. Questa modalità assicura un'elevata efficienza energetica che permette ad un dispositivo SigFox di durare anni con una batteria standard, come in apparecchiature per il rilievo di parametri fisici o ambientali (lettura dei contatori di acqua e gas) che non richiedono in invio continuo dei dati.

La **Fig. 1** schematizza la rete SigFox.

La comunicazione SigFox utilizza una modulazione Ultra Narrow Band (UNB) senza alcuna negoziazione dei parametri della connessione tra il dispositivo e la base station ricevente, permettendo di ridurre la quantità di dati scambiati senza rinunciare all'immunità alle interferenze. Il dispositivo semplicemente emette nella banda di frequenza disponibile ed il segnale viene rilevato dalle base station più vicine, decodificato e trasmesso al back-end della rete, dove i messaggi vengono poi inoltrati all'applicazione dell'utente e resi accessibili tramite le API di SigFox. Ogni messaggio viene autenticato utilizzando un meccanismo hash, e una chiave privata specifica del dispositivo garantendo l'impossibilità di hackerare i dati trasmessi; vediamo le caratteristiche più salienti.

È possibile inviare fino a 12 byte (96 bit) per messaggio ed il protocollo include già un timestamp (data e ora) della trasmissione e l'ID unico del dispositivo nei suoi metadata.

Il numero massimo di messaggi che possono essere inviati ogni giorno è di 140, limitazione dovuta al rispetto delle normative del settore in quanto, la normativa europea che disciplina la banda 868 MHz consente un duty cycle massimo di trasmissione del 1%. Un singolo dispositivo non potrà pertanto trasmettere per più dell' 1% del tempo in un'ora.

Giusto per completezza vediamo quali sono le differenze tra le due tecnologie trattate in questo corso, LoRaWAN e SigFox. SigFox usa uno protocollo di trasmissione radio

chiamato BPSK (Binary Phase Shift Keying) che utilizza una porzione molto piccola dello spettro radio permettendo di minimizzare l'effetto del rumore sulla comunicazione ed utilizzare apparati (endpoint) più semplici ed economici, ma richiede l'utilizzo di gateway/basestation molto più sofisticati e costosi. LoRa invece utilizza una tecnologia chiamata Spread Spectrum con un maggiore utilizzo di banda (oltre 125 kHz) che introduce molto più rumore, pertanto l'hardware è tendenzialmente più costoso ma può essere utilizzato indifferentemente su tutti i dispositivi (non vi è differenza tra endpoint e basestation) permettendo di avere gateway più economici e semplici da realizzare rispetto SigFox.

Le grosse differenze tra le due reti riguardano invece la gestione commerciale. LoRa è open source e di libero utilizzo ed i punti di accesso possono essere creati da tutti, anche privati, non ci sono costi per accedere alla rete, pertanto è considerata la rete delle persone ed è in continua espansione proprio grazie agli utenti stessi che desiderano installare un punto di accesso per la loro zona. L'unico aspetto negativo è che i moduli radio Lora sono prodotti da un'unica azienda (solo recentemente sono stati stipulati accordi commerciali per la produzione su licenza) con l'effetto di mantenere elevati i prezzi di vendita.

SigFox invece si comporta come un operatore telefonico noleggiando l'accesso alla rete tramite i suoi gateway, oppure gateway di aziende partner con le quali ha stipulato contratti commerciali. Il costo è di circa 1 euro al mese per singolo device. Di contro la produzione dell'hardware è demandata a molti produttori in regime di libero mercato permettendo prezzi di vendita decisamente contenuti.

Nettrotter è l'unico operatore Italiano in grado di implementare e gestire la rete SigFox sul territorio nazionale Italiano, avendo il ruolo di SNO (SigFox Network Operator). Questa iniziativa fa parte di un più ampio progetto, coordinato da



Fig. 1 - Struttura della rete SigFox.

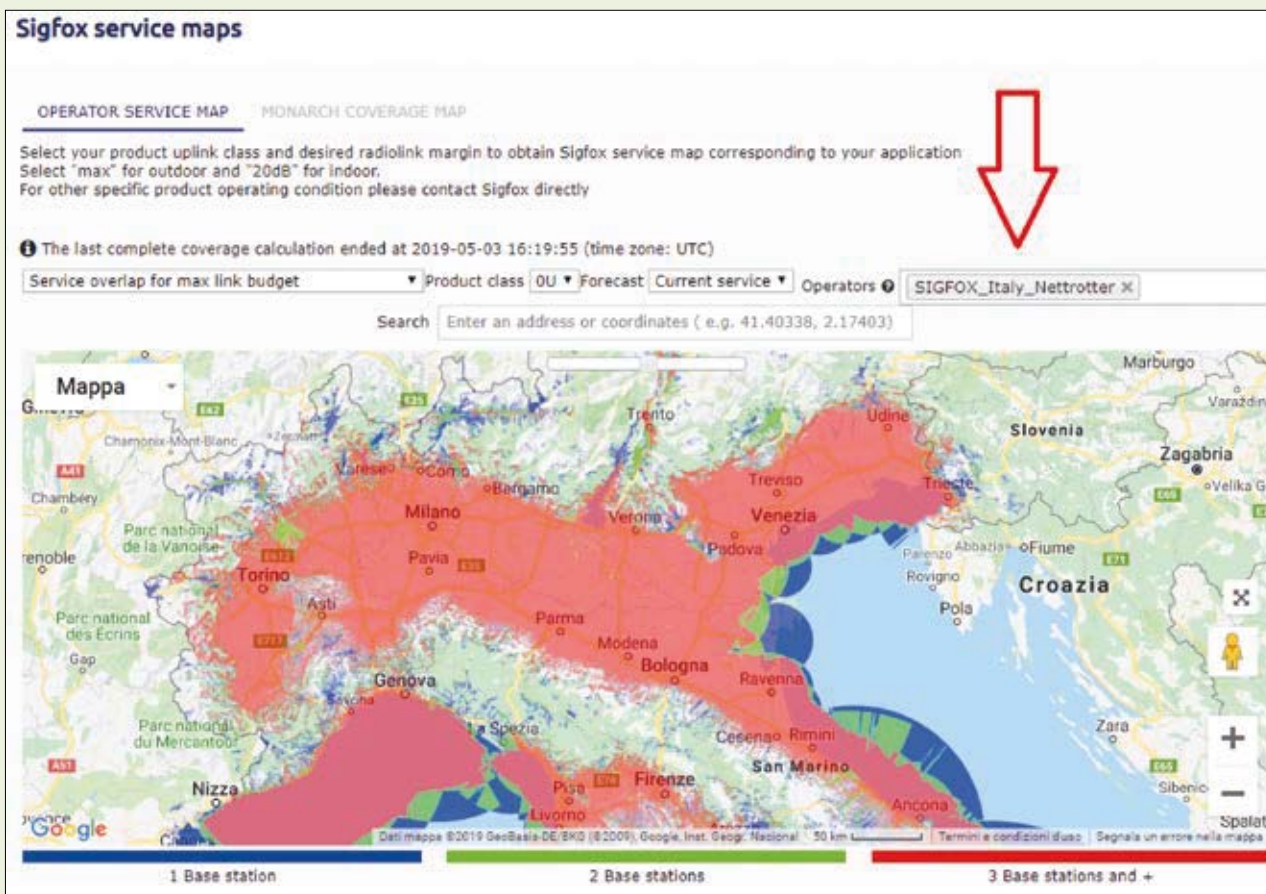


Fig. 2 Copertura della rete SigFox.

SigFox, che ha l'obiettivo di sviluppare un network globale per l'IoT a supporto di una ampia gamma di applicazioni. La rete offerta da Nettrotter sfrutta le esistenti torri TLC e televisive e prevede di raggiungere una copertura nazionale, con quasi 1.000 Base Station SigFox già installate.

Già coperte oltre 40 tra le principali città italiane, fra cui Roma, Milano, Torino, Bologna, Firenze, Napoli, Bari, Reggio Calabria, Palermo e già servita l'80% della popolazione. La Nettrotter fornisce il sistema Telenet Water, prima soluzione Italiana su rete SigFox per lo smart metering idrico, caratterizzata da bassi consumi ed alta sensibilità in ricezione del segnale trasmesso dall'apparato. La soluzione consente, previa applicazione del prodotto sul contatore dell'acqua, la trasmissione in real time dei dati caratterizzanti la distribuzione dell'acqua potabile alle utenze domestiche ed industriali con una durata della batteria fino a 15 anni. Per conoscere la copertura nella vostra zona dovete accedere al portale (<https://backend.sigfox.com/auth/login>), cliccare sul logo in alto a sinistra, selezionare service maps e selezionare come provider SIGFOX_Italy_Nettrotter, come visibile nella **Fig. 2**.

Vediamo ora come implementare praticamente un dispositivo che sfrutti le potenzialità della rete SigFox e lo facciamo

utilizzando la scheda Arduino MKR FOX 1200 (**Fig. 3**), appositamente progettata allo scopo. Come già detto nelle precedenti puntate le MKR sono schede basate su MCU SAM D21E (Microchip ARM Cortex M0) operanti alla frequenza di 48MHz che includono 256K di memoria Flash e 32K di memoria SRAM, quindi decisamente più preformanti di una scheda Arduino UNO. La parte di comunicazione SigFox è demandata ad un IC ATA8520 della Microchip, il quale comunica con la



Fig. 3 - La scheda Arduino MKR FOX 1200 con antenna

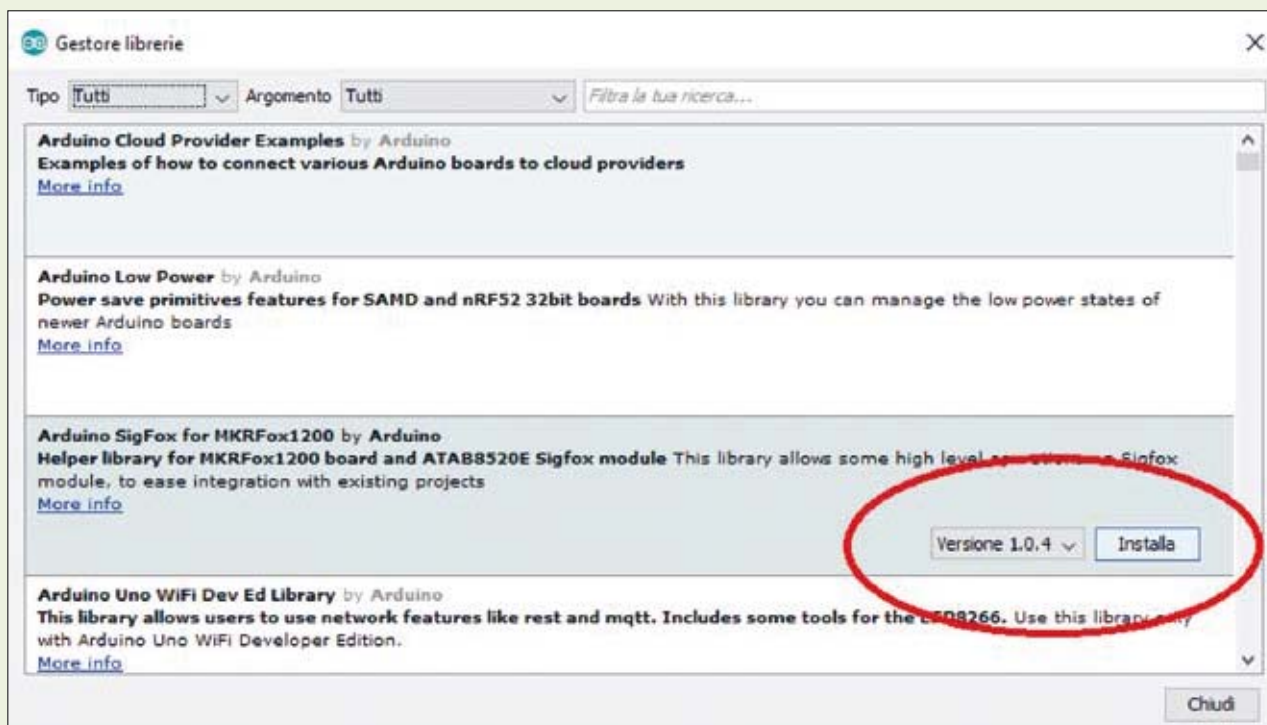


Fig. 4 - Installazione della libreria per l'applicazione SigFox.

CPU principale tramite bus SPI. Il prezzo di acquisto comprende un abbonamento per due anni ai servizi SigFox. L'utilizzo del modulo radio e l'accesso alla rete SigFox è notevolmente semplificato grazie a librerie scritte appositamente per lo scopo. Sarà quindi necessario installare le librerie: Arduino SAMD Boards, Arduino RTCZero, Arduino Low-Power, Arduino SigFox for MKRFox1200.

Aprire e caricare l'esempio di nome *FirstConfiguration.ino* contenuto in File-Esempi-Arduino SigFox for MKRFox 1200, su Serial Monitor potete leggere la versione del firmware del modulo radio e i due parametri univoci che identificano l'hardware ID e PAC (**Fig. 5**). La **Fig. 4** propone l'installazione della libreria per l'applicazione SigFox. Il prezzo della scheda MKRFox1200 include un abbonamento

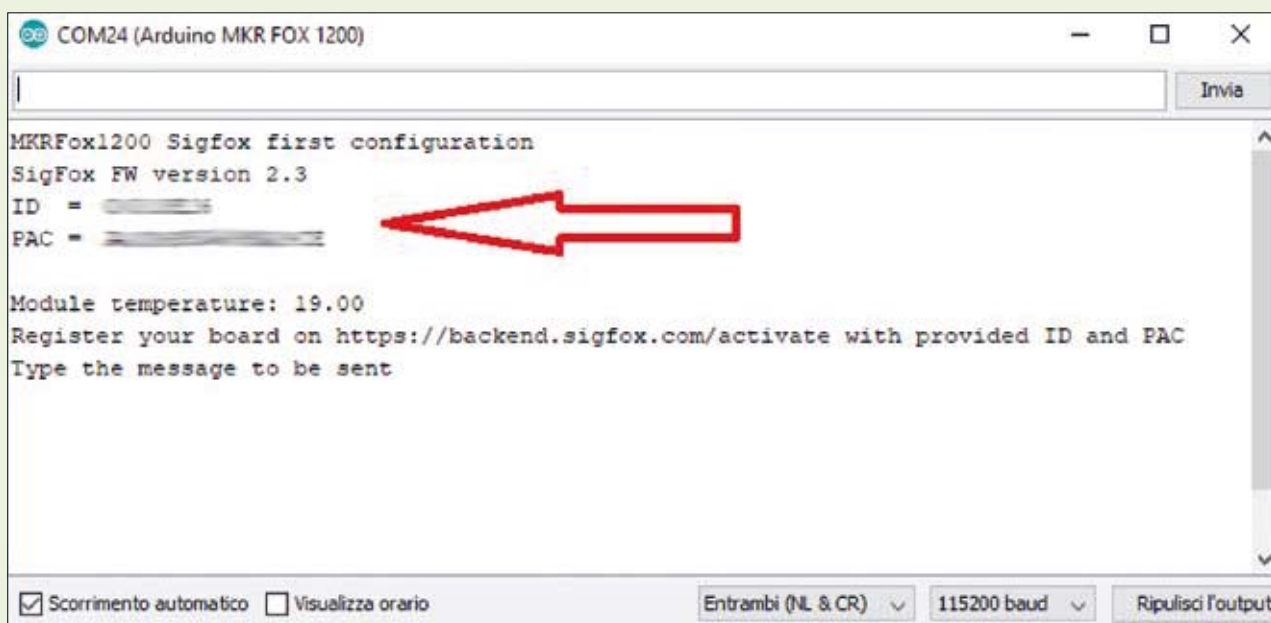


Fig. 5 Schermata di Serial monitor relativa all'esempio *FirstConfiguration.ino*

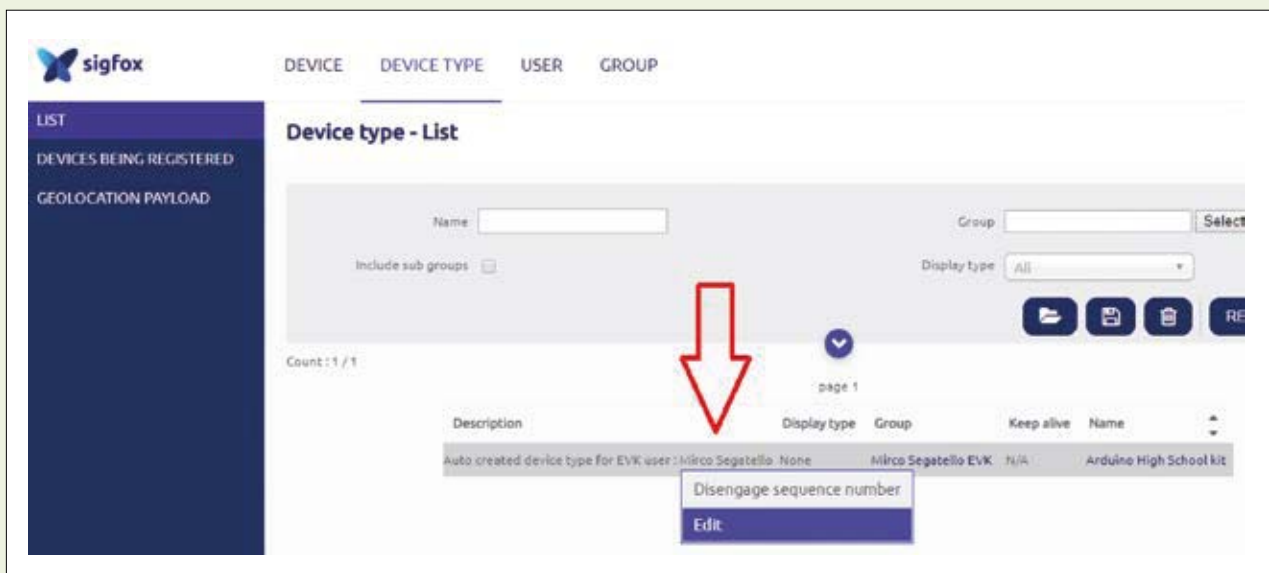


Fig. 6 Apertura del menu per la configurazione della scheda.

alla rete SigFox per un anno. Il piano verrà attivato automaticamente dopo l'invio del quarto messaggio, con il limite di 140 messaggi al giorno. Affinché la scheda MKR1200 possa accedere alla rete SigFox è necessario attivarla tramite il portale

<https://buy.sigfox.com/activate>. Il primo passo è selezionare il paese (Italia), il secondo è selezionare il tipo di hardware (Arduino), successivamente la procedura chiede di specificare a quale provider volete connettervi (Nettrotter). Nei passi successivi dovrete inserire i dati della scheda (ID e PAC) e i

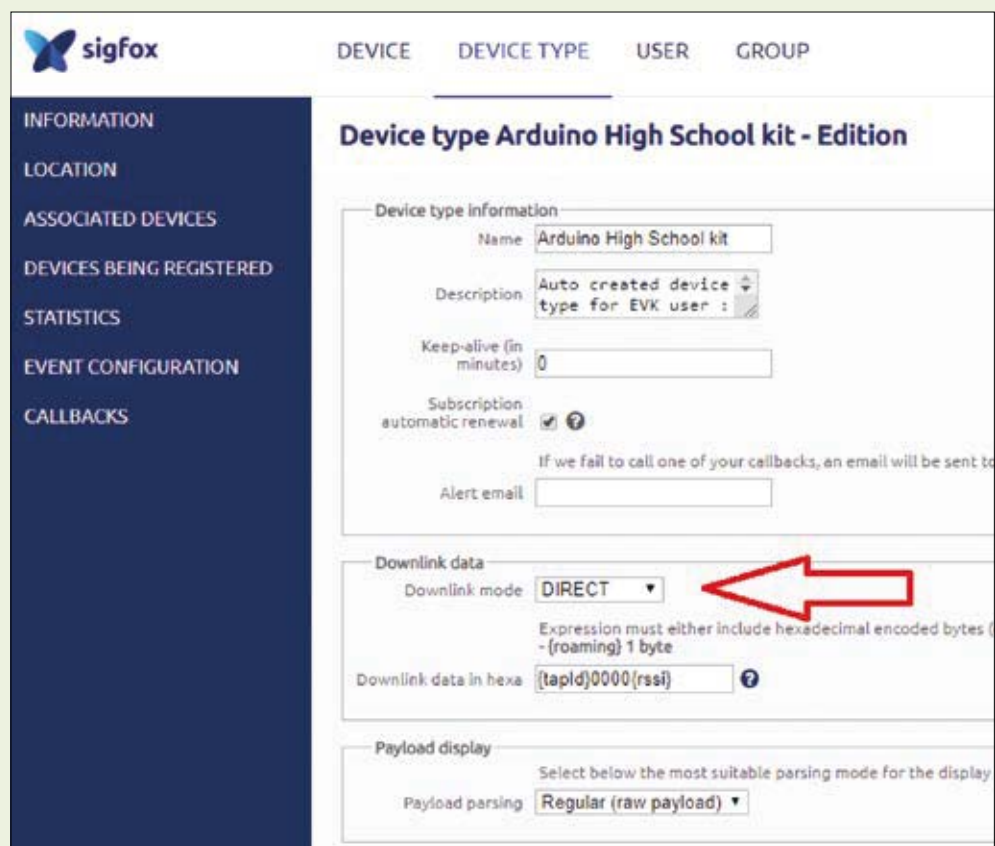


Fig. 7 Configurazione della scheda.

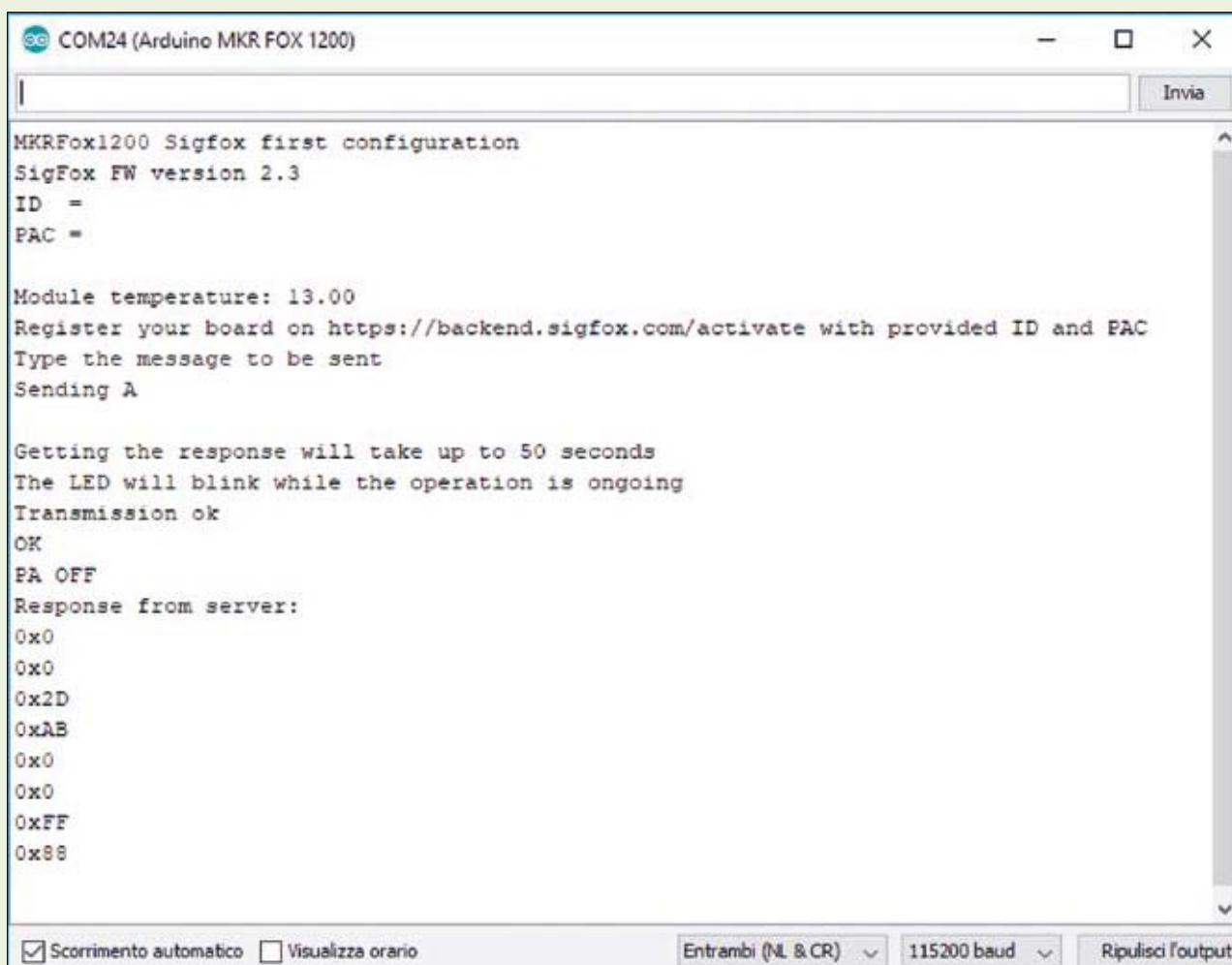


Fig. 8 - Schermata di Serial monitor con invio andato a buon fine

vostri dati personali, a procedura conclusa riceverete una mail di conferma che contiene un link che vi condurrà all'in-

serimento di una password per l'accesso al portale; conclusa la registrazione vi ritroverete una schermata come quella

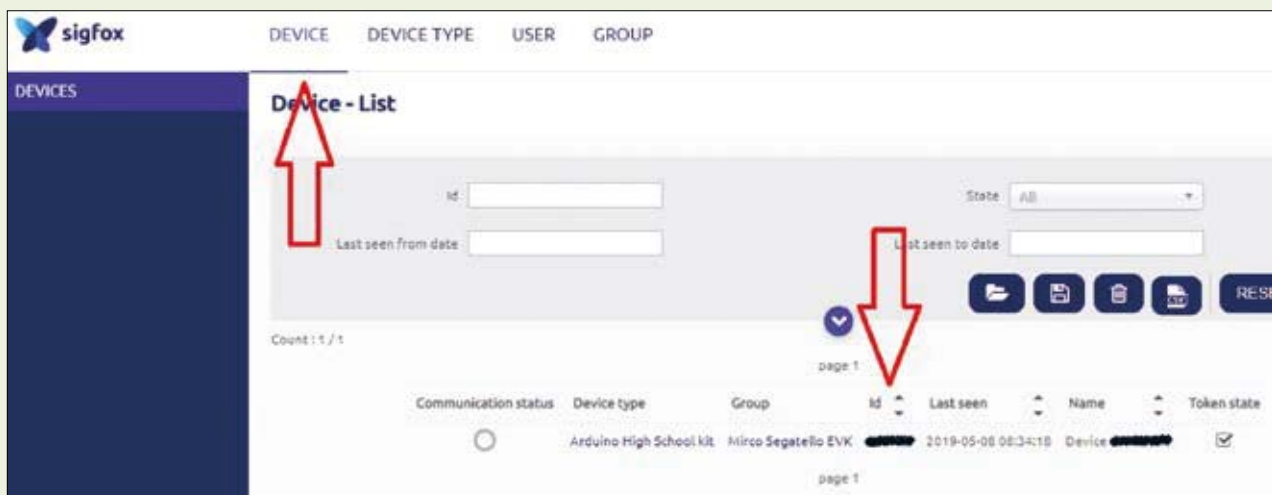


Fig. 9 - Selezione dei dispositivi dal portale SigFox.

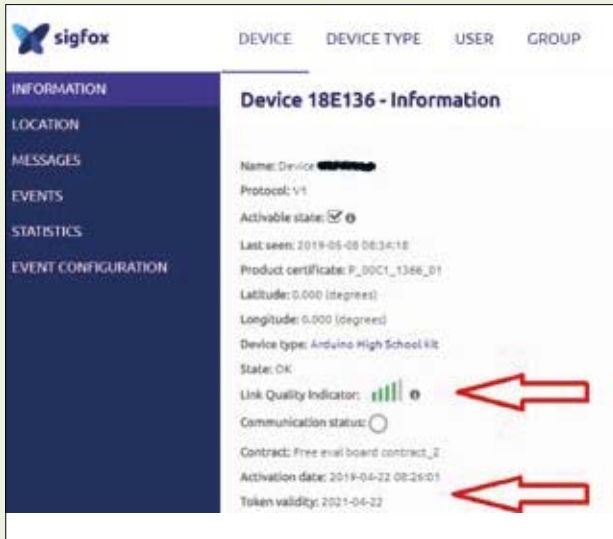


Fig. 10 Schermata con le informazioni della scheda

visibile in Fig. 6.

Il menu in alto permette di accedere alle informazioni delle vostre schede associate e ai dati del vostro account: se cliccate su Device type e successivamente sulla vostra scheda (sotto la voce description), si aprirà un menu a tendina e selezionando il comando Edit aprirete il menu di configurazione della scheda (Fig. 7). Da questa schermata potete modificare il nome associato alla scheda, l'indirizzo email per la funzionalità Call Back (esportazione dei dati verso altri portali) ma soprattutto potremmo abilitare la risposta automatica ai messaggi ricevuti, permettendo di avere conferma di ricezione; alla voce "download link mode" selezioniamo DIRECT. Sempre da questo menu, è possibile avere la statistica dei messaggi cliccando sulla voce STATISTICS.

Adesso è tutto pronto per la nostra prima comunicazione, quindi avviamo nuovamente lo sketch *FirstConfiguration.ino* e questa volta inseriamo il messaggio da inviare, nel nostro caso il semplice carattere **A** maiuscolo. Ricordiamo che SigFox limita fortemente il transito dei dati per cui è sempre necessario ridurre al massimo i bytes inviati.

La schermata di risposta fornita da Serial Monitor di Arduino è visibile nella Fig 8; nel caso la risposta dal server tardasse ad arrivare Serial Monitor ne darà notifica consigliando di trovare una posizione della scheda con più segnale. I dati di risposta contengono la data, l'ora e l'intensità del segnale

page 1

Time	Data / Decoding	LQI	Callbacks	Location
2019-05-08 08:33:41	41			
2019-05-08 07:55:45	T177			

Fig. 11 - Elenco dei messaggi ricevuto dal portatore SigFox.

Listato 1

```
int sensorValue;
#include <SigFox.h> // inclusione libreria sigfox
#define INTERVAL 10 //intervallo di tempo dei

void setup()
{
  Serial.begin(115200); // abilita la seriale per il debug
  while (!Serial) {}; // aspetta che la seriale sia avviata

  if (!SigFox.begin()) // verifica la presenza del modem sigfox
  {
    Serial.println("Shield error or not present!");
    return;
  }

  SigFox.debug(); // Abilita il debug del LED
  SigFox.reset(); //resetta il modem e la libreria
  delay(100);
  SigFox.end(); // pone il modem in modalità di basso consumo
}

void loop()
{
  SigFox.begin(); // avvia il modulo
  delay(100); // attende che il modulo sia inizializzato

  // legge la temperatura interna del modem
  int8_t value = (int8_t)SigFox.internalTemperature();
  Serial.print("Temp=");
  Serial.println(value); // visualizzo il valore acquisito

  SigFox.beginPacket(); // prepara il modem per l'invio
  SigFox.write(value); // invia il valore

  // chiude la comunicazione e verifica la corretta trasmissione
  int ret = SigFox.endPacket(false);
  if (ret > 0) {
    Serial.println("No transmission");
  } else {
    Serial.println("Transmission ok");
  }
  delay(100);
  SigFox.end(); // pone il modem in modalità di basso consumo
  delay(100);
  Serial.println("Attendo...\n");
  delay(INTERVAL*60*1000);
}
```

(codificati in esadecimale), così come specificato nella voce download data. Noterete che la temperatura del modulo è di 13°C, questo perché è stato posizionato all'esterno dell'abitazione al fine di garantire una buona intensità del segnale, non sempre garantita in ambienti interni.

Dal portale SigFox cliccate sul menu DEVICES e, nella lista dei dispositivi che vi viene presentata, cliccate sull'ID della vostra scheda (Fig. 9): a questo punto si aprirà un nuovo menu visibile in Fig. 10 con alcuni dati (compresa la data di scadenza del contratto); a sinistra selezionate MESSAGE per visualizzare i dati ricevuti.

Come potete vedere dalla Fig. 11, nella schermata corrispondente vengono riportati data e ora del messaggio, il nostro carattere inviato in formato ASCII codificato in esadecimale e l'intensità del segnale radio.

Fig. 12 - Impostazione dei parametri in DEVICE TYPE.

COSTRUIAMO LA NOSTRA APPLICAZIONE CON SIGFOX

Vediamo ora come possiamo realizzare un'applicazione sfruttando la rete SigFox, tenendo anche in considerazione il fatto che l'invio di dati verso la nostra scheda è limitata a circa un messaggio ogni dieci minuti (almeno con il contratto base). Per prima cosa reimpostiamo la comunicazione in modalità unidirezionale in modo da evitare di dover lasciare attiva la scheda per ricevere il messaggio di ritorno (che può impiegare anche 50 secondi) e successivamente impostiamo il dato di arrivo di tipo intero, visualizzabile come temperatura. Dalla schermata principale clicchiamo su DEVICE TYPE e poi sopra il nome della scheda, nel menu che si aprirà clicchiamo su edit ed impostiamo i parametri come visibile nella **Fig. 12**. Non ci resta che caricare lo sketch identificato dal nome *MKRfox1200_TempMonitor.ino* (**Listato 1**) per iniziare l'invio dei dati. Vediamo brevemente, esaminando il **Listato 1**,

quali sono i punti salienti di questo programma: la prima riga riguarda l'inclusione della libreria SigFox mentre la seconda specifica, tramite il parametro INTERVAL, ogni quanti minuti sarà inviato il dato. Nella sezione setup viene semplicemente configurata la comunicazione con il PC e verificato il corretto funzionamento del modem.

La parte più importante, quella che gestisce l'invio del dato, è tutta contenuta nel loop, semplificato per garantirne la massima chiarezza. In questa parte si abilita il modem e si legge la temperatura fornita dello stesso (ha un sensore interno) che successivamente viene inviata a Serial Monitor per la visualizzazione. Ovviamente in questa sezione potrete inserire la lettura di qualsiasi altro valore, fermo restando che sarà un valore ad otto bit. Le righe successive si occupano di inviare il pacchetto di dati e chiudere la comunicazione, viene anche verificata se la trasmissione è andata a buon fine ma non viene controllata la corretta ricezione dei dati trasmessi, in quanto abbiamo volutamente disabilitato dal portale l'invio della ricevuta di ritorno. Esiste la possibilità di implementare la funzionalità di deep sleep della scheda e del modulo al fine di ridurre drasticamente il consumo di corrente, funzione che però in alcuni PC, interrompe definitivamente la funzionalità di invio dati a Serial Monitor. In questa prima fase ci sembra utile mantenere attivi i messaggi di debug al fine di verificare che tutto funzioni correttamente.

Cliccando su MESSAGE nel portale SigFox potrete vedere i dati ricevuti (**Fig. 13**) e cliccando sul simbolo a goccia rovesciata di colore blu potete visualizzare una mappa con la posizione della vostra scheda. Siccome le coordinate sono determinate per triangolazione con le varie base station conoscendo solo il segnale RSSI, l'area a volte è troppo vasta per poter essere usata come localizzatore.

CONCLUSIONI

Con la descrizione del sistema SigFox concludiamo questo corso dedicato all'IoT. Prossimamente pubblicheremo progetti applicativi sia con la scheda MKR1200 per la rete SigFox, sia con hardware per LoRa. A presto!

Time	Data / Decoding	LQI	Callbacks	Location
2019-05-09 15:49:57	16 Temperatura: 22			
2019-05-09 15:39:46	13 Temperatura: 19			
2019-05-09 15:29:34	12 Temperatura: 18			

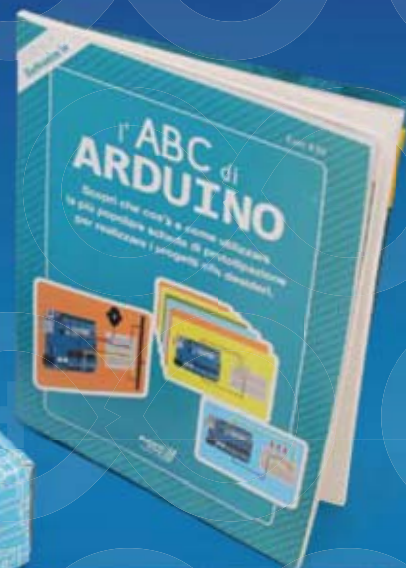
Fig. 13
Letture dei messaggi ricevuti con l'esempio del listato 1

Programma e realizza applicazioni elettroniche con ARDUINO!

**IL PIÙ
COMPLETO!**

SET PER ARDUINO UNO REV3

Set contenente tutto il necessario per sviluppare applicazioni con ARDUINO UNO Rev3. Il kit comprende: il libro "l'ABC di ARDUINO", la board Arduino UNO Rev3, shield di prototipazione, breadboard, connettori, contenitore plastico, LED, buzzer, pulsanti, display vari, interruttori, fotoresistenze, potenziometro, jumper, sensori, ricevitore infrarossi, telecomando IR, resistenze, cavetti, display LCD 16 caratteri, servo, motore passo-passo, scheda driver, chip 74HC595.



€ 65,00

Cod. **ARDUKITV6**

Utilizzalo anche con il kit di 37 sensori

Moduli con sensori per Arduino, Raspberry, robotica ecc.

La confezione contiene i seguenti moduli:

- Joystick da C.S.
- Sensore di fiamma
- Pulsazioni cardiache
- LED e interruttore al mercurio
- Sensore effetto di Hall
- Relè da 5 V
- Sensore effetto Hall lineare
- LED RGB SMD
- LED cambia colore
- Sensore di Tilt
- Sensore di temperatura DS18B20
- Rilevatore di suono con microfono grande
- Touch sensor
- LED bicolore 5 mm
- LASER
- Sensore di Tilt a inclinazione
- Sensore di temperatura analogico
- Rilevatore di suono con microfono piccolo
- Sensore di temperatura digitale
- LED bicolore 3 mm
- Pulsante N.A.
- Fotoresistenza
- LED trasmettitore IR
- Sensore di linea bianca e nera
- Buzzer con elettronica
- Interruttore reed e interfaccia digitale
- Interruttore a vibrazione
- Sensore di temperatura e umidità
- Interruttore reed
- Ricevitore IR a 3 pin
- Rilevatore di ostacoli
- Buzzer senza elettronica
- Encoder rotativo
- Sensore effetto di Hall analogico
- Sensore urto
- Mini photo interruttore a forcella

€ 29,90

Cod. **SENSORKIT37**



Prezzi IVA inclusa

Bolle di idrogeno: un nuovo meccanismo per generare luce dai semiconduttori

Uno studio condotto dalla Sapienza, in collaborazione con il Cnr e la Australian National University, ha rivelato che attraverso la formazione di bolle di idrogeno ad alta pressione è possibile modificare le proprietà fisiche e strutturali di semiconduttori innovativi per l'optoelettronica e il fotovoltaico.

I risultati sono stati pubblicati sulla rivista Advanced Materials

I materiali del futuro saranno bidimensionali e con eccezionali proprietà: più resistenti dell'acciaio, più leggeri dell'alluminio, flessibili come la plastica. Dopo anni di intensa e fruttuosa attività sul grafene, l'interesse della comunità scientifica va progressivamente spostandosi verso la ricerca di altri materiali bidimensionali, in grado di sostituire i semiconduttori convenzionali e particolarmente adatti per

la realizzazione di dispositivi compatti, a basso consumo e flessibili.

Un nuovo studio, coordinato dai ricercatori del Dipartimento di Fisica della Sapienza in collaborazione con il Centro di ricerca per le nanotecnologie applicate all'ingegneria (Cnis) della Sapienza, l'Istituto di Fotonica e Nanotecnologie del Cnr e la Australian National University, ha individuato un metodo innovativo per

Il Boeing MQ-25 Unmanned Aerial Refueler completa il suo primo test flight



Boeing e la Marina americana hanno completato con successo il primo test flight del MQ-25 unmanned aerial refueler.

Il test asset MQ-25, denominato T1, ha completato il volo autonomo di due ore sotto la direzione dei piloti Boeing da una stazione di controllo a terra all'aeroporto MidAmerica St. Louis Airport a Mascoutah, Illinois, dove è basato il programma

dei test. L'aereo ha completato in modo autonomo un rullaggio e un decollo per poi volare su una rotta predefinita al fine di validare con la stazione di controllo a terra le funzioni di volo base dell'aereo e le operazioni.

L'MQ-25 fornirà alla Marina la capacità di rifornimento aereo unmanned di cui più ha bisogno. Consentirà un migliore utilizzo

dei combat strike fighter che attualmente svolgono il ruolo di rifornimento e amplierà il range dello squadrone del vettore.

"Questo volo test è un traguardo entusiasmante e significativo per il nostro programma e per la Marina", ha dichiarato il Program Manager Capt. Chad Reed del Navy's Unmanned Carrier Aviation (PMA-268). *"Il volo di questo test asset due anni*



creare cristalli a due dimensioni altamente deformati. La possibilità di indurre deformazioni meccaniche controllate in questi materiali può essere sfruttata per modellarne a piacimento le proprietà elettroniche, ottiche e di trasporto. I risultati di questo studio sono stati pubblicati sulla rivista Advanced Materials.

www.uniroma1.it/

Presentata la nuova Starship di Elon Musk

Elon Musk ha svelato tutte le caratteristiche della nuova astronave "Starship" di SpaceX, destinata a trasportare l'uomo sulla Luna e su Marte. *"È il Sacro Graal dei viaggi spaziali - ha detto, parlando a Boca Chica, in Texas, il miliardario visionario - rappresenta la svolta verso la trasformazione della nostra civiltà in una civiltà spaziale, rendendo i viaggi interplanetari una normalità, come i viaggi aerei".*

Alla folta platea intervenuta, Musk ha mostrato i suggestivi video in graphic

animation sui futuribili viaggi verso la Luna e verso Marte. La Starship è alta 50 metri e larga 9 ed è in grado di portare in orbita uomini e materiali fino a 150 tonnellate di peso. Partirà in cima a un razzo Super Heavy, alto 68 metri. I primi test di volo atmosferico del prototipo denominato Starhopper sono già cominciati con buoni risultati, a breve, invece, la Starship, che nella forma ricorda l'astronave dei fumetti di Tin Tin, sarà pronta a effettuare i primi voli spaziali, prima senza, e poi con equipaggio umano.

www.spacex.com



prima dell'arrivo del nostro primo MQ-25 rappresenta il primo grande passo di una serie di opportunità di apprendimento che ci stanno aiutando a progredire verso la consegna di capacità rivoluzionarie per lo squadrone dell'operatore e per i comandanti del gruppo d'attacco".

www.boeing.com

ExaNeSt, il supercomputer europeo che consuma un decimo degli altri

È stato testato con successo il precursore di un calcolatore elettronico ad elevate prestazioni tutto "made in Europe" che consuma fino a un decimo dell'energia richiesta dalle analoghe piattaforme di calcolo oggi operative. L'Italia è coinvolta nel progetto con l'INFN e l'Istituto Nazionale di Astrofisica (INAF). La costruzione di un supercomputer in grado svolgere l'incredibile cifra di un miliardo di miliardi di calcoli al secondo (exascale) rappresenta una delle più grandi sfide scientifiche e tecnologiche dei nostri giorni. L'Europa sta da tempo lavorando per rendere realtà la realizzazione di un supercomputer di classe Exascale grazie anche ai consorzi finanziati dal programma europeo Horizon 2020 in progetti come ExaNeSt che, in poco più di 3 anni dalla fine del 2015, ha costru-



ito il primo prototipo di super computer, interamente realizzato in Europa a partire dalla progettazione dell'hardware sino alle applicazioni che lo usano. Si tratta di un prototipo interamente funzionante basato su tecnologie di rete e storage europee, sviluppato, testato e ottimizzato a partire dalle applicazioni scientifiche e tecnologiche che per la prima volta hanno contribuito sin dall'inizio alla realizzazione dell'hardware e del software. Il prototipo realizzato da ExaNeSt non solo ha delle elevate prestazioni di calcolo, ma ha anche una ottima efficienza energetica: l'energia consumata per risolvere un problema computazionale su questa nuova piattaforma è da 3 a 10 volte inferiore a quella richiesta da piattaforme HPC tradizionali.

www.inaf.com

Nanorisonatori per pesare le molecole

"Large-scale parallelization of nanomechanical mass spectrometry with weakly-coupled resonators" è il titolo della ricerca che è stata pubblicata su Nature Communications: la ricerca è di Stefano Stassi e Carlo Ricciardi, ricercatore e docente del Dipartimento di Scienza Applicata e Tecnologia del Politecnico di Torino, in collaborazione con l'Istituto Italiano di Tecnologia e l'Università di Melbourne. Lo studio affronta il problema dell'identificazione delle molecole: ad oggi la metodologia più utilizzata è la spettrometria di massa, in uso ad esempio negli aeroporti per la rilevazione di sostanze stupefacenti o per le analisi di contaminanti chimici. La tecnica, analizzando un campione, mette in movimento e "spacchetta" le molecole atomo ad atomo per il riconoscimento. Tecnica che però presenta alcuni limiti quando applicata alle analisi

del sangue: in primo luogo l'identificazione delle biomolecole, che essendo più grandi e più pesanti, oltre a "far fatica" a muoversi, una volta divise perdono la loro funzionalità biologica; e in secondo luogo la grandezza e la velocità di elaborazione degli strumenti che non possono essere "portatili" e a basso costo. La ricerca dimostra che i nanorisonatori possono essere un'alternativa più veloce e ad ampio spettro: si tratta di minuscole travi di silicio che vibrano a una propria frequenza, frequenza che, entrando in contatto con una determinata molecola che viene "pesata" come fosse su una bilancia, subisce una variazione.

Grazie a questa variazione, che è diversa da molecola a molecola ed è tanto più ampia quanto più piccolo è il risonatore-bilancia, si può effettuare una identificazione diretta, senza necessità di "spacchettare" la molecola.

www.polito.it

Il Cnr nel primo volo suborbitale di ricerca 'human-tended' con Virgin Galactic

Virgin Galactic ha firmato un contratto con l'Aeronautica Militare Italiana per un volo di ricerca sul velivolo suborbitale SpaceShipTwo. È la prima volta che un'istituzione governativa finanzia un volo di ricerca a carattere umano su un velivolo spaziale commerciale. La missione, che avrà luogo nel 2020, farà volare a bordo della SpaceShipTwo tre Space Flight Participants, due dell'Aeronautica Militare e uno del Cnr, oltre a un rack per il payload.

Durante il volo suborbitale le condizioni di microgravità, pur se di pochi minuti, permettono infatti di effettuare esperimenti multidisciplinari con risultati e scoperte importanti, altrimenti difficilmente realizzabili, relativi alla scienza dei materiali, alla fisica dei fluidi, alla scienza della combustione, alla biologia, con ciò migliorando la tecnologia dello sviluppo di nuovi materiali, trovando nuove soluzioni ai problemi della loro realizzazione, scoprendo fenomeni fisici, chimici e biologici ancora sconosciuti.

Il Dta e il Diitet sono i due dipartimenti che coordineranno le attività Cnr sul volo suborbitale.

Un team composto da personale dei due dipartimenti coinvolti sta valutando una serie di sperimentazioni e progettazione di payload sperimentali idonei alla tipologia di missione, in collaborazione con gli SFP dell'Aeronautica Militare e il Team 'Payload' di Virgin Galactic. Il volo è previsto per dicembre 2020.

www.cnr.it



Tutto sotto controllo ovunque e in ogni momento!



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

TELECAMERA IP Wi-Fi 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 Wi-Fi 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.



Scarabeus, il termodinamico che abbatte i costi

Il Progetto Horizon 2020 SCARABEUS, coordinato dal Politecnico di Milano, ha l'obiettivo di ridurre del 32% i costi di installazione delle centrali solari termodinamiche e del 40% quelli di produzione dell'energia elettrica. Un risparmio di risorse che si tradurrebbe anche nell'abbattimento del costo finale dell'energia elettrica arrivando potenzialmente a 96€/MWh (-30%). Sarà così colmato il divario tra il termodinamico e le altre tecnologie rinnovabili.

Tecnicamente questa efficienza sarà raggiunta tramite lo sviluppo di un

innovativo ciclo di conversione dell'energia termica in energia elettrica dove l'acqua è sostituita con miscele di CO₂. L'aggiunta di piccole quantità di elementi selezionati a CO₂ pura (ovvero composti inorganici e fluorocarburi), nota come miscelazione di CO₂, consentirà di migliorare l'efficienza del ciclo di potenza riducendone contemporaneamente le dimensioni e, appunto, i costi. Oltre alla validazione teorica del concetto, nel progetto SCARABEUS verranno caratterizzate le proprietà termo-fisiche delle miscele di CO₂ analizzate e verrà dimostrata la loro stabilità termica in tipiche condizioni operative per 2.000 ore.

Inoltre, i componenti necessari per il nuovo fluido di lavoro, come ad esempio gli scambiatori di calore, saranno testati in condizioni CSP tipiche per 300 ore in un ciclo di prova di 300

Lo smartphone si ricarica al sole

Il gruppo di ricerca 'NewPV' dell'Istituto dei materiali per l'elettronica ed il magnetismo del Consiglio nazionale delle ricerche (Cnr-Imem) di Parma, ha sviluppato dei sistemi di alimentazione portatili basati sul fotovoltaico (PV) per la ricarica di utenze elettriche a bassa potenza. Le celle solari di alimentazione sono flessibili, e quindi applicabili anche su superfici curve come ad esempio quelle degli ombrelloni e lettini da mare, ecc.

"L'idea di questi sistemi nasce dal bisogno sempre più urgente di avere smartphone e tablet sempre funzionanti e connessi alla rete mobile. Il sovrautilizzo di questi dispositivi riduce l'autonomia della batteria, che necessita quindi di essere ricaricata più volte al giorno."

Il sistema, chiamato NewPV-3, permette di utilizzare l'energia solare liberamente, in qualsiasi luogo ci si trovi, sfruttando un mini-modulo fotovoltaico opportunamente ingegnerizzato per caricare fino a 6 smartphone con-

kWth, valutando anche la compatibilità dei materiali e potenziali problemi relativi ai materiali. Infine, saranno sviluppati progetti di turbomacchine per centrali elettriche di grandi dimensioni. SCARABEUS durerà 48 mesi, iniziato ad aprile 2019 terminerà nel marzo 2023.

www.polimi.it

temporaneamente, senza il bisogno di essere connessi alla rete elettrica, a costo zero ed a zero emissioni di CO₂. I mini-moduli possono essere trasportati, installati e disinstallati facilmente, non necessitano di alcun tipo di manutenzione, sono resistenti all'acqua e a limitate deformazioni meccaniche. Ogni sistema è dotato di un mini-modulo solare collegato ad un'unità operativa, a cui si connettono i dispositivi da caricare mediante prese USB e accendisigari.

www.cnr.it



Materiali organici cristallini per la progettazione delle celle solari

Sebbene siano stati fatti notevoli progressi nei sistemi fotovoltaici organici, il controllo della loro struttura nanometrica si è rivelato sfuggente. Lo sviluppo di sistemi modello con reti periodiche ben definite e interpenetranti di fasi di donatori e accettori di elettroni rappresenterebbe un'importante svolta.

Il progetto ECOF, finanziato dall'UE, ha affrontato questa sfida creando sistemi modello altamente definiti per aumentare la comprensione della relazione tra i parametri elettronici e strutturali e le conseguenti dinamiche dei portatori di carica indotte dalla luce.

I ricercatori hanno basato il loro lavoro sulle strutture organiche covalenti (COF, covalent organic frameworks), una classe di materiali cristallini organici al-

tamente porosi scoperta di recente i cui elementi costitutivi molecolari sono tenuti insieme da legami covalenti. "Questi polimeri a strati bidimensionali e tridimensionali promettono un controllo strutturale e una comprensione decisamente maggiori delle proprietà fisiche", afferma il

prof. **Thomas Bein**, coordinatore del progetto.

Questo straordinario progetto interdisciplinare prevedeva una sintesi organica di alto livello con nanoscienze avanzate e caratterizzazione fisica approfondita.

"Essere in grado di assemblare architetture COF su nanoscala e microscala è di fondamentale importanza per ottenere nuove funzionalità", spiega il prof. Bein.

<https://cordis.europa.eu>



EVA+: attive in Italia e Austria tutte le 200 stazioni di ricarica del progetto

Sono attive tutte le 200 stazioni di ricarica veloce da 50kW installate nell'ambito del progetto EVA+, pari a 400 punti di ricarica disponibili per i clienti finali presso le principali strade a lunga percorrenza di Italia e Austria.

EVA+, Electric Vehicles Arteries, è il progetto europeo per lo sviluppo della mobilità elettrica nato nel 2016, che prevedeva entro tre anni l'installazione di 200 infrastrutture di ricarica veloce, 180 in

Italia e 20 in Austria, lungo le strade a lunga percorrenza extraurbane dei due Paesi. Le infrastrutture di ricarica di EVA+ presenti sul territorio italiano sono interamente sviluppate e installate da Enel X, e grazie a una potenza fino a 50kW, permettono di fare un pieno di energia a due veicoli contemporaneamente in circa 30 minuti. Ogni stazione è compatibile con tutti i modelli di auto elettriche attualmente sul mercato e rispetta gli standard

internazionali nell'ambito dell'e-mobility: CCS, CHAdeMO e Type 2. Le stazioni di ricarica veloci EVA+ fanno parte di una rete capillare presente sul territorio italiano e austriaco che, da oggi, permette agli utenti di affrontare un viaggio in auto elettrica dalla Sicilia all'Austria con soste di breve durata e senza paura di rimanere senza energia.

www.enel.com



Velivoli senza pilota per una migliore gestione marina

La protezione e la gestione sostenibile delle risorse marine è un obiettivo politico importante per l'UE, come indicato nella sua Strategia per la ricerca marina e marittima. Le vaste coste e aree economiche dell'Europa che si estendono fino agli oceani Atlantico e Artico sono difficili da monitorare e gestire. Per questo scopo vengono generalmente impiegati grandi navi da ricerca, boe, satelliti e aeromobili con equipaggio che, oltre a fornire una copertura limitata, comportano anche costi finanziari elevati. Oceani e coste risultano quindi relativamente sottocampionati con una conseguente carenza di dati con le informazioni richieste su spazio, tem-

po e caratteristiche. I sistemi autonomi di velivoli senza pilota (UAV, Unmanned Aerial Vehicle) vengono sempre più utilizzati come soluzione. Essi offrono elevata resistenza, costi ridotti, flessibilità, rapidità di implementazione e maggiore accuratezza/risoluzione, con rischi ridotti per l'uomo e per l'ambiente. Il progetto MarineUAS (Autonomous Unmanned Aerial Systems for

Marine and Coastal Monitoring), finanziato dall'UE, ha formato 16 ricercatori all'inizio della carriera in diversi paesi, creando un ambiente collaborativo che ha beneficiato della vasta conoscenza, esperienza sul campo e strutture dei partner per migliorare le tecnologie. I ricercatori hanno costruito e testato sperimentalmente prototipi per migliorare la misurazione del vento e l'elaborazione intelligente, nonché metodi per l'atterraggio sicuro di UAV ad ala fissa su piccole imbarcazioni e la capacità degli UAV di resistere a condizioni meteorologiche avverse.

www.marineuas.eu/



Il primo viaggio commerciale del traghetto completamente elettrico

Nel quadro degli intensi sforzi di decarbonizzazione del trasporto marittimo e con il sostegno del progetto E-ferry, finanziato dall'UE, un traghetto completamente elettrico che può trasportare circa 30 veicoli e 200 passeggeri ha navigato tra i porti di Soebye (Søby) e Fynshav, situati sulle isole danesi di Aerø (Ærø) e Als. Afferma Anil Srivastava, amministratore delegato di Leclanché, società partner del progetto, "Con la sua capacità di 4,3 MWh, l'E-ferry rappresenta un nuovo traguardo

nella propulsione navale commerciale. Su un periodo di un anno, eviterà il rilascio nell'atmosfera di 2.000 tonnellate di CO₂, 42 tonnellate di NOX (ossido di azoto), 2,5 tonnellate di particolati e 1,4 tonnellate di SO₂ (diossido di zolfo)".

Leclanché ha fornito il sistema di batteria agli ioni di litio al traghetto elettrico Ellen, "il precursore di una nuova era nel

settore marittimo commerciale", aggiunge Srivastava. "Tale progetto dimostra che oggi è possibile sostituire i motori a propulsione termica che usano combustibile fossile con energia pulita, contribuendo quindi alla lotta contro il riscaldamento globale e l'inquinamento, per il benessere delle nostre comunità". Ellen può percorrere fino a 22 miglia nautiche fra una ricarica e l'altra, "ossia una distanza di sette volte maggiore rispetto a quanto precedentemente possibile per un traghetto elettrico."

<http://e-ferryproject.eu/>



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.

INDISPENSABILI IN LABORATORIO!



L'alimentatore è lo strumento più importante di ogni laboratorio elettronico da cui si esige prima di tutto la possibilità di regolare con buona risoluzione la tensione e la corrente di uscita, requisiti soddisfatti da questi nuovi alimentatori. Mantengono i valori impostati in modo affidabile e indipendentemente dalle fluttuazioni della tensione di alimentazione o del carico collegato.

ALIMENTATORE LINEARE CON DOPPIA USCITA REGOLABILE 0-30 VDC / 0-5 A

Alimentazione: 230 VAC - 50 Hz

Uscita duale regolabile:

- Tensione: da 0 a 30 VDC

- Corrente: da 0 a 5 A

Uscita fissa: 5 VDC / 3 A

Tensione di ripple: < 1 mV

Dimensioni: 360x265x165 mm

Peso: 10 kg



€ 54,00

Cod. **AL1502**

ALIMENTATORE SWITCHING A SINGOLA USCITA REGOLABILE 0-15 VDC / 0-2 A

Alimentazione: 230 VAC - 50 Hz

Uscita regolabile:

- Tensione: da 0 a 15 VDC

- Corrente: da 0 a 2 A

Dimensioni: 240x150x95 mm

Peso: 3,5 kg



DISPONIBILE ANCHE NELLA
VERSIONE DA 0-30 VDC / 0-3 A
COD. AL2X3003 € 189,00



€ 228,00

Cod. **AL2X3005**

Prezzi IVA inclusa.